

# Universidad de las Ciencias Informáticas

## Facultad 2



**Título: Módulo de Administración SIRIO Mobile Admin.**

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

**Autores:** David Rodríguez Rodríguez

Rigoberto Pérez Rajadel

**Tutores:** Ing. Arian Sulueta Casal

Ing. Yunisel Viera Vargas

## **DECLARACIÓN DE AUTORÍA**

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Entidad Procyon de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
David Rodriguez Rodriguez  
Firma del Autor

\_\_\_\_\_  
Rigoberto Pérez Rajadel  
Firma del Autor

\_\_\_\_\_  
Ing. Arian Zulueta Casal  
Firma del Tutor

\_\_\_\_\_  
Ing. Yunisel Viera Vargas  
Firma del Tutor

## **AGRADECIMIENTOS**

*A nuestros tutores Arian y Yunisel que nos apoyaron siempre, a Abel que nos guió en nuestro trabajo.*

*A todos los trabajadores de Procyon, fue una tremenda experiencia trabajar a con ustedes.*

*Y a todos los que contribuyeron a que este sueño fuera posible.*

*A la Universidad por formarnos como profesionales.*

**David y Rigo**

*A mami (Alina) por quererme y malcriarme tanto, a papi (David) por confiar siempre en mí y mostrarme con su ejemplo como ser el mejor padre del mundo, a mi hermano (Dainer) por ser el*

*mejor regalo que me ha dado la vida, a mi novia (Denyse) por su amor,*

*que le da a mi espíritu, la savia que necesita.*

**David**

*A mis padres Milieydís y Rafe, a mi hermano Andy, a mis cuatro abuelos.*

**Rigo**

## DEDICATORIA

*A mis padres, a mi hermano y al amor de mi vida.*

*A mis amigos, a los compañeros de aula en estos 5 años, y a todos los que me han apoyado o ayudado durante la carrera.*

*David*

*A mis padres, mi hermano, mis abuelos, que siempre me apoyaron y me ayudaron para vencer a mis metas, a mis amigos, mis compañeros de grupo con los que siempre pude contar.*

*Rigo*

## **RESUMEN**

El amplio desarrollo alcanzado por la telefonía celular ha propiciado un indiscutible aumento en la producción de aplicaciones informáticas destinadas a ellos, constituyendo actualmente una de las ramas de mayor crecimiento mundial.

En Procyon, entidad cubana dedicada al desarrollo de aplicaciones para móviles, se lleva a cabo un amplio trabajo en el área de juegos para celulares. Producto a la gran aceptación que poseen, principalmente los del tipo multi-usuario, se desarrolló SIRIO, plataforma que maneja la gestión de estos juegos entre teléfonos móviles.

En su primera versión, SIRIO brinda un Módulo de Administración que ofrece la posibilidad de monitorear el estado del servidor de juegos mediante una interfaz Web. Como parte de un proceso de mejoras fueron identificadas deficiencias en cuanto a: ubicuidad, interoperabilidad con diferentes aplicaciones y seguridad en la transferencia de información.

Partiendo de este análisis se desarrolló SIRIO Mobile Admin, un nuevo Módulo de Administración que fue incorporado a la plataforma y está destinado a teléfonos móviles. Este facilita el control del servidor y proporciona herramientas necesarias para su constante monitoreo.

Además, asegura la interoperabilidad con diferentes aplicaciones con la incorporación de un Servicio Web, que sirve de intermediario entre las aplicaciones clientes y el servidor de la plataforma. Incorpora elementos de seguridad, con el establecimiento de un canal seguro para la comunicación.

Con el objetivo de cubrir una amplia gama de dispositivos, fueron desarrolladas dos aplicaciones cliente, en dos de las plataformas más utilizadas actualmente en el mundo del software para móviles.

### **Palabras Claves**

GPRS, Módulo, Administración, Servicio Web, Teléfonos Celulares, Seguridad.

## TABLA DE CONTENIDOS

AGRADECIMIENTOS .....	II
DEDICATORIA .....	III
RESUMEN .....	IV
INTRODUCCIÓN .....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 INTRODUCCIÓN.....	5
1.2 TELEFONÍA CELULAR O MÓVIL, COMUNICACIONES INALÁMBRICAS .....	5
1.2.1 <i>Evolución de telefonía celular o móvil. Generaciones</i> .....	5
1.2.2 <i>Comunicaciones inalámbricas</i> .....	7
1.2.2.1 GPRS.....	8
1.2.3 <i>Plataformas para el soporte de juegos</i> .....	9
1.3 ARQUITECTURA CLIENTE SERVIDOR.....	11
1.4 METODOLOGÍA DE DESARROLLO EMPLEADA EN EL PROYECTO .....	11
1.4.1 <i>Proceso Unificado de Desarrollo de Software</i> .....	12
1.5 PLATAFORMAS DE DESARROLLO. ....	12
1.5.1 <i>¿Por qué J2ME?</i> .....	12
1.5.2 <i>¿Por qué J2EE?</i> .....	14
1.5.3 <i>¿Por qué .NET COMPACT FRAMEWORK?</i> .....	14
1.6 LENGUAJES DE PROGRAMACIÓN UTILIZADOS .....	15
1.6.1 <i>¿Por qué Java?</i> .....	15
1.6.2 <i>¿Por qué C#?</i> .....	15
1.7 SERVIDOR WEB .....	16
1.8 SERVICIOS WEB .....	16
1.8.1 <i>Protocolo de Comunicación</i> .....	17
1.9 PROTOCOLO SEGURO DE TRANSFERENCIA DE HIPERTEXTO (HTTPS).....	17
1.9.1 <i>Certificados Digitales</i> .....	18
1.10 HERRAMIENTAS DE DESARROLLO .....	19
1.10.1 <i>J2ME Wireless Toolkit 2.2</i> .....	19
1.10.2 <i>ActiveSync</i> .....	20
1.10.3 <i>Key and Certificate Management Tool (keytool)</i> .....	20
1.10.4 <i>Entornos de Desarrollo Integrado</i> .....	20
1.10.4.1 Eclipse.....	21
1.10.4.2 Visual Studio 2005 Professional Edition.....	21
1.11 LENGUAJE UNIFICADO DE MODELACIÓN .....	21
1.12 HERRAMIENTAS DE MODELADO UML .....	22
1.12.1 <i>Rational Rose Enterprise Edition 2003</i> .....	22
1.13 HERRAMIENTAS PARA EL CONTROL DE VERSIONES .....	22
1.13.1 <i>SVN</i> .....	23
1.13.2 <i>Cliente de Subversion utilizado</i> .....	23

1.14 CONCLUSIONES.....	24
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....</b>	<b>24</b>
2.1 INTRODUCCIÓN.....	24
2.2 OBJETO DE AUTOMATIZACIÓN .....	25
2.3 PROPUESTA DEL SISTEMA.....	25
2.4 MODELO DE NEGOCIO.....	27
2.5 MODELO DE DOMINIO.....	27
2.5.1 <i>Descripción de los conceptos del domino</i> .....	28
2.6 REQUERIMIENTOS .....	29
2.6.1 <i>Requerimientos funcionales</i> .....	29
2.6.2 <i>Requerimientos no funcionales</i> .....	30
2.7 MODELO DEL SISTEMA. DEFINICIÓN DE CASOS DE USO .....	32
2.7.1 <i>Definición de actores</i> .....	32
2.7.2 <i>Casos de uso del sistema</i> .....	32
2.7.3 <i>Descripción detallada de los casos de uso del sistema</i> .....	34
2.8 CONCLUSIONES .....	35
<b>CAPITULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA .....</b>	<b>36</b>
3.1 INTRODUCCIÓN.....	36
3.2 FLUJO DE TRABAJO ANÁLISIS Y DISEÑO.....	37
3.3 MODELO DE ANÁLISIS .....	37
3.3.1 <i>Diagrama de clases del análisis</i> .....	38
3.4 MODELO DE DISEÑO .....	38
3.4.1 <i>Diagrama de clases del diseño</i> .....	39
3.4.2 <i>Diagrama de Secuencia</i> .....	40
3.5 DECISIONES DE DISEÑO EMPLEADOS EN EL DESARROLLO DEL SISTEMA.....	42
3.6 TRATAMIENTO DE ERRORES.....	45
3.7 SEGURIDAD .....	46
3.8 INTERFAZ .....	48
3.9 CONCLUSIONES .....	49
<b>CAPITULO 4: IMPLEMENTACIÓN Y PRUEBA .....</b>	<b>50</b>
4.1 INTRODUCCIÓN.....	50
4.2 MODELO DE IMPLEMENTACIÓN .....	50
4.2.1 <i>Diagrama de Componentes</i> .....	50
4.2.2 <i>Diagrama de Despliegue</i> .....	51
4.3 MODELO DE PRUEBAS.....	52
4.3.1 <i>Métodos de Prueba</i> .....	53
4.3.2 <i>Diseño de Casos de Prueba. Prueba de Caja negra</i> .....	53
4.4 CONCLUSIONES.....	58
<b>CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD.....</b>	<b>59</b>
5.1 INTRODUCCIÓN.....	59
5.2 ESTIMACIÓN POR PUNTOS DE CASOS DE USO .....	59

<b>5.3 PLANIFICACION BASADA EN CASOS DE USO</b> .....	60
<b>5.4 BENEFICIOS TANGIBLES E INTANGIBLES</b> .....	66
<b>5.5 ANÁLISIS DE COSTOS Y BENEFICIOS</b> .....	67
<b>5.6 CONCLUSIONES</b> .....	68
<b>CONCLUSIONES</b> .....	<b>69</b>
<b>RECOMENDACIONES</b> .....	<b>70</b>
<b>BIBLIOGRAFÍA</b> .....	<b>71</b>
<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	<b>72</b>
<b>ANEXO</b> .....	<b>75</b>
<b>GLOSARIO</b> .....	<b>95</b>



## **INTRODUCCIÓN**

La telefonía celular brinda ubicuidad e inmediatez, convirtiéndose en una de las tecnologías más utilizadas a nivel mundial. En el período 2005-2010 se prevé un crecimiento medio anual del 11,9%, alcanzando, para el año 2010, 3.9 mil millones de clientes celulares de una población mundial estimada en 7 mil millones de habitantes. [1]

Los celulares actuales, incluyen funciones cada vez más versátiles, convirtiéndose prácticamente en una pequeña computadora de bolsillo. Puesto que son impulsados por procesadores cada vez más potentes, memorias con mayor capacidad y poseen compatibilidad con estándares abiertos, se están convirtiendo en sólidas plataformas de aplicaciones para usos empresariales. [2] Su conectividad, comodidad y su capacidad de comunicación prometen nuevos aumentos de productividad dentro de estos entornos.

En Procyon, entidad que se encuentra enmarcada dentro de la infraestructura productiva (IP) de la Universidad de las Ciencias Informáticas (UCI), se fomenta el uso de las nuevas tecnologías de comunicación en función de la producción de aplicaciones para móviles. Producto al alto crecimiento en la demanda de los juegos en red se desarrolló la plataforma SIRIO, para la gestión de estos, que tiene como característica fundamental el soporte para múltiples juegos.

La plataforma está conformada por tres módulos, el Servidor Web, el API de Conexión y el Módulo de Administración Web, este último presenta gran importancia ya que permite el monitoreo del estado del servidor [3], a continuación se abunda en algunas de sus características.

Para su uso se necesita una computadora con acceso a la plataforma SIRIO. No posee flexibilidad para migraciones futuras a otro tipo de aplicaciones o diferentes dispositivos.

El tráfico de información, entre el navegador y el servidor es inseguro ya que los datos transferidos viajan como texto plano por la red, estando propensos a ser interceptados o modificados por intrusos, esto puede causar inestabilidad en el sistema, insatisfacción y pérdida de clientes, lo que reportaría disminución en los ingresos.

Como parte del proceso de mejoras de la primera versión de la plataforma SIRIO, fueron identificadas funcionalidades necesarias a incluir en el Módulo de Administración.

- Se necesita una mayor capacidad de uso del módulo de administración, de forma tal que sea accesible desde casi cualquier lugar a través de teléfonos celulares.
- Requiere mayor flexibilidad para la adaptación del módulo a nuevos dispositivos y plataformas.
- Garantizar la confidencialidad en el intercambio de información.

Por lo que se dio inicio a una nueva investigación con el fin de resolver el siguiente **problema**:

¿Cómo desarrollar un nuevo módulo de administración de la plataforma SIRIO, destinada a dispositivos móviles, que brinde mayor interoperabilidad y provea seguridad en la transferencia de datos?

El **objeto de estudio** de la investigación se enmarca en la telefonía móvil, las comunicaciones inalámbricas, el paradigma Cliente – Servidor, la tecnología de Servicios Web y la seguridad a través del protocolo de comunicación HTTPS (del inglés Hypertext Transfer Protocol Secure). Siendo su **campo de acción**, la interoperabilidad, seguridad y movilidad en sistemas de administración de plataformas de juegos para móviles que se comuniquen vía GPRS.

Con el fin de obtener una solución de alta calidad se plantean los siguientes **objetivos**.

### **Objetivo General:**

- Definir una solución para un nuevo módulo de administración de la plataforma SIRIO, que incorpore elementos claves como la ubicuidad, la interoperabilidad con diferentes aplicaciones y la seguridad en la transferencia de información.

### **Objetivos Específicos:**

- Obtener una aplicación orientada a dispositivos que presenten Windows Mobile<sup>1</sup> como sistema operativo, para el monitoreo y administración de la plataforma SIRIO.
- Obtener una aplicación orientada a dispositivos J2ME (del inglés Java 2 Micro Edition), para el monitoreo y administración de la plataforma SIRIO.

---

<sup>1</sup> Sistema operativo compacto destinado a dispositivos móviles.

- Crear un Servicio Web para la interacción con la plataforma SIRIO, que brinde las funcionalidades necesarias para su administración y garantice una mayor interoperabilidad.
- Garantizar la confidencialidad de la información en el proceso de comunicación.

Para el desarrollo del proyecto, se tuvieron en cuenta las siguientes **preguntas de científicas**:

- ¿Cuáles son las principales características de GPRS (del inglés General Packet Radio Service)?
- ¿Cómo es el funcionamiento de la plataforma SIRIO?
- ¿Cómo lograr un alto nivel de seguridad en la transferencia de información?
- ¿Qué ventajas ofrece el desarrollo de aplicaciones de administración para la plataforma SIRIO destinadas a dispositivos móviles?
- ¿Qué ventajas ofrece la utilización de Servicios Web en aplicaciones informáticas?

Con el propósito de darle cumplimiento a los objetivos trazados se realizaron las siguientes **tareas de la investigación**:

- Análisis del funcionamiento de los tres módulos de la plataforma SIRIO, especialmente el Servidor y el Módulo de Administración Web.
- Estudio y selección de la tecnologías disponibles para el desarrollo de aplicaciones para teléfonos móviles.
- Estudio del proceso de comunicación mediante el protocolo HTTPS.
- Estudio y selección de las principales metodologías de desarrollo establecidas a nivel mundial.
- Análisis e implementación de una arquitectura que satisfaga los objetivos planteados.

- Estudio e implementación de patrones de diseño para lograr un producto robusto, flexible y acabado.

Para lograr una mejor organización del contenido, se decidió que el documento estuviera estructurado en capítulos, los cuales se describen a continuación.

- Capítulo 1: Fundamentación Teórica: Describe los principales conceptos involucrados en la realización de la aplicación, el estado del arte del tema tratado a nivel internacional, nacional y de la Universidad así como las principales tendencias, técnicas, tecnologías, metodologías y software usados en la solución del problema.
- Capítulo 2: Características del Sistema: Se describe una propuesta del sistema basada en los aspectos planteados anteriormente y se lleva a cabo un análisis comparativo con las soluciones similares encontradas durante la investigación.
- Capítulo 3: Análisis y Diseño del Sistema: Se describen todos los artefactos relacionados con el análisis y diseño de la aplicación.
- Capítulo 4: Implementación y Prueba: Se describen todos los artefactos relacionados con la implementación y prueba de la aplicación.
- Capítulo 5: Estudio de Factibilidad: Se lleva a cabo el estudio de factibilidad del proyecto desarrollado, se describen los beneficios tangibles e intangibles obtenidos y se analizan los costos.

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

### **1.1 Introducción**

En el presente capítulo se describen los principales aspectos y conceptos de relevancia que han sido objeto de análisis a lo largo de la investigación. Se realiza un acercamiento al estado en que se encuentran actualmente las aplicaciones para la administración de servidores de juegos en red desde teléfonos celulares, así como las principales tendencias, plataformas de desarrollo, tecnologías, metodologías y herramientas que hicieron posible la realización del trabajo.

### **1.2 Telefonía celular o móvil, comunicaciones inalámbricas**

En este epígrafe se abordarán temas relacionados con el mundo la telefonía celular y su evolución. Además se mostrarán los principales beneficios de las comunicaciones inalámbricas y su clasificación de acuerdo al alcance geográfico.

#### **1.2.1 Evolución de telefonía celular o móvil. Generaciones**

La industria de las comunicaciones móviles ha evolucionado en tres etapas, siendo cada generación más fiable y flexible que la anterior. En cada una de ellas según grado de evolución técnica resulta fácilmente reconocible un conjunto de características que las conforman.

La primera generación (1G) fue analógica y limitada en capacidad de roaming<sup>2</sup>, permitía solamente llamadas de voz con baja calidad. Debido a que los estándares existentes solo definían el protocolo de acceso de radio entre una estación móvil y su correspondiente estación base, y no cubrían aspectos relativos a la gestión de la movilidad, surgieron toda una serie de estándares diferentes en diversos países, que trajeron consigo los correspondientes problemas de incompatibilidad.

Posteriormente surgió una segunda generación (2G) digital que significó un incremento en la capacidad de la red, una mejora en la calidad de voz, capacidades de roaming mejoradas, incorporación de la transmisión de datos a bajas velocidades, reducción de tarifas y los primeros servicios de valor añadido, como son los mensajes cortos SMS (del inglés Short Message Service) y

---

<sup>2</sup> Capacidad de un dispositivo para moverse de una zona de cobertura a otra, hacer y recibir llamadas en redes móviles fuera del área de servicio local.

además conformó las bases para la prestación de otros servicios como la transmisión de datos a altas velocidades. [4]

La segunda generación de sistemas de comunicación móvil fue pensada fundamentalmente para la transmisión de voz. Por ello, de la concepción original del estándar se derivan una serie de limitaciones para las aplicaciones basadas en la transmisión de datos, como son: lento establecimiento de la conexión (sistema orientado a conexión), insuficiencia a la hora de transmisión de grandes volúmenes de información a altas velocidades entre terminales inalámbricos y la red fija, poco ancho de banda, simetría del enlace (mismo ancho de banda para ambos sentidos de la comunicación), coste excesivo (facturación por tiempo de conexión). Para suplir estas necesidades en los servicios, surgió una generación previa a la tercera, denominada 2.5. Una de estas tecnologías es GPRS, basada en la transmisión de paquetes y donde los canales de comunicación se comparten entre los distintos usuarios de forma dinámica. [5]

GPRS coexiste con GSM<sup>3</sup> (del inglés Global System for Mobile Communication), compartiendo gran parte de su infraestructura. Ofrece al usuario un servicio portador más eficiente para las comunicaciones de datos, especialmente en el caso de los servicios de acceso a redes IP (del inglés Internet Protocol), como Internet. Es adecuada para muchas aplicaciones e integra WAP (del inglés Wireless Application Protocol), MMS (del inglés Multimedia Messaging Service) y juegos móviles SMS. [6]

Muchos de los proveedores de servicios de telecomunicaciones se moverán a las redes 2.5G antes de entrar masivamente a la 3G. Debido a que la tecnología 2.5G es más rápida, y más económica para actualizar que la tercera generación.

La 3G promete ser la respuesta a los problemas existentes al ofrecer servicios de voz, dato y video a altas velocidades, y quizás el don de la ubicuidad en cuanto a comunicaciones móviles. Pensada para roaming global, transmisión de datos de alta velocidad a través de técnicas avanzadas de conmutación de circuitos y de paquetes, soporta tecnología IP y ATM (del inglés Asynchronous Transfer Mode), lo que posibilita el acceso a Internet, y en general aplicaciones multimedia móviles, con servicios personalizados y basados en la localización de los usuarios. El nuevo modelo de negocio es radicalmente distinto del actual y entran en juego nuevos agentes, como son los proveedores de

---

<sup>3</sup> Tecnología correspondiente a la segunda generación de telefonía móvil (2G).

contenidos y los proveedores de aplicaciones. Esta tecnología 3G está recogida en Europa por el estándar UMTS (del inglés: Universal Mobile Telecommunications System). [7]

### **1.2.2 Comunicaciones inalámbricas**

Las comunicaciones inalámbricas están presentes en muchas de las actividades que se realizan diariamente, su uso ha llegado a ser tan común, que se pierde la percepción de lo útil e indispensable que pueden llegar a ser. Al igual que el telégrafo evolucionó con el devenir del tiempo a teléfonos móviles, las redes locales necesitaban un proceso evolutivo similar, de aquí que surgiera la idea de crear Redes Locales Inalámbricas, WLAN (del inglés Wireless Local Area Network). [3]

En la comunicación inalámbrica se utiliza la modulación de ondas electromagnéticas, las cuales se propagan por el espacio sin un medio físico que comunique cada uno de los extremos de la transmisión. En ese sentido, los dispositivos físicos sólo están presentes en los emisores y receptores de la señal, como por ejemplo: antenas, laptops, PDAs<sup>4</sup>, teléfonos celulares, etc.

Las redes inalámbricas traen consigo un conjunto considerable de ventajas tales como:

- Mayor disponibilidad y acceso de la información, permite establecer conexión desde cualquier lugar dentro del área de influencia de la red.
- Facilita la ampliación y el mantenimiento, es más económico cuando se requiere aumentar el número de máquinas en una red debido a que no es necesario rediseñar toda la estructura ya existente, reduciendo considerablemente tanto la inversión económica como el tiempo de reestructuración.
- Comodidad, facilidad de instalación y costes mínimos.
- Flexibilidad, permite que la red vaya donde no puede llegar el cable.
- Escalabilidad, pueden configurarse en varias topologías para satisfacer las necesidades de usos específicos. [8]

Las redes inalámbricas se clasifican de atendiendo a su alcance geográfico en los siguientes grupos.

---

<sup>4</sup> Ordenador de pequeño tamaño cuya principal función era en principio mantener una agenda electrónica.

- Redes inalámbricas tipo PAN (del inglés Personal Area Network): útiles cuando la conectividad debe alcanzar un radio de acción de pocos metros de distancia (por ejemplo, teléfonos móviles, computadoras y otros dispositivos como impresoras y cámaras fotográficas), alguna de las tecnologías que utiliza son Bluetooth<sup>5</sup>, IEEE 802.15<sup>6</sup> y HomeRF<sup>7</sup>.
- Redes inalámbricas tipo LAN (del inglés Local Area Network): válidas en edificios o estructuras que pueden estar dispersas en un radio de acción de entre 150 metros y un kilómetro y medio, según la penetrabilidad de los muros y suelos de los edificios, la tecnología de comunicación inalámbrica más utilizada en este caso es WIFI<sup>8</sup> (del inglés Wireless Fidelity).
- Redes inalámbricas tipo WAN/MAN (del inglés Wireless Wide Area Network/Wireless Metropolitan Area Network): abarcan un área geográfica relativamente extensa, cubren desde decenas hasta miles de kilómetros, ejemplos de tecnologías que implementan este tipo de redes podemos encontrar a GSM, GPRS y UMTS. [9]

Partiendo del hecho que la aplicación a realizar debe acceder a datos y estadísticas alojados en un servidor a una gran distancia de los posibles clientes (celulares), el abanico de tecnologías a utilizar se reduce a las que faciliten la conexión a largas distancias. Entre ellas encontramos: GSM, GPRS y UMTS.

La solución a obtener deberá ser desarrollada sobre GPRS, debido a que la misma constituye un módulo de la plataforma SIRIO, la cual utiliza dicha tecnología para el establecimiento de la comunicación. A continuación se ampliarán sus principales características.

#### **1.2.2.1 GPRS**

La tecnología Servicio General de Paquetes de Radio, GPRS es la evolución natural de la actual GSM, reutilizando parte de su infraestructura y corrigiendo alguna de sus deficiencias. Utiliza el concepto de

---

<sup>5</sup> Especificación industrial para redes Pan que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia.

<sup>6</sup> El Estándar IEEE 802.15 se enfoca básicamente en el desarrollo de estándares para redes tipo PAN o redes inalámbricas de corta distancia. <http://www.conocimientosweb.net/portal/html.php?file=cursos/redes/Conectividad.htm>

<sup>7</sup> HomeRF es una especificación que permite la interconexión de dispositivos inalámbricos en un área pequeña. <http://www.conocimientosweb.net/portal/html.php?file=cursos/redes/Conectividad.htm>

<sup>8</sup> Conjunto de estándares para redes inalámbricas basados en las especificaciones IEEE 802.11. <http://www.aulaclie.es/articulos/wifi.html>



Comunicación por Paquetes, en vez de la tradicional por circuitos empleada por GSM y utiliza su rango de frecuencias.

Entre sus principales características cabe destacar que los canales de comunicación se comparten entre los distintos usuarios dinámicamente, de modo que un usuario tiene asignado un canal cuando se está realmente transmitiendo datos.

Las principales ventajas que aporta GPRS respecto a GSM son, además de una mayor velocidad de transmisión, la conexión permanente y la tarificación por datos transmitidos, convirtiéndolo en el portador ideal para los servicios WAP, el acceso a Internet y a intranets de empresas. Tradicionalmente la transmisión de datos con la utilización de un canal GSM, se ha venido realizando a una velocidad máxima de 9.6 Kbps, con GPRS la velocidad de transmisión de datos se ve aumentada desde un mínimo 40 Kbps a un máximo de 115 Kbps por comunicación. [10]

### **1.2.3 Plataformas para el soporte de juegos**

A nivel mundial empresas informáticas han realizado plataformas y servidores que dan soportes a juegos en red desde teléfonos móviles, entre ellas encontramos:

- **Neutrón**, plataforma para el desarrollo de aplicaciones multiusuario producida por Exit Games, en San Francisco, Estados Unidos, actualmente se encuentra desplegada y brinda servicios a nivel global, sirviendo de alojamiento a una gran cantidad de juegos multiusuario para teléfonos celulares. [11]
- **Deus**, plataforma producida y puesta en marcha en el 2003 por Exmachina, que radica en Ámsterdam, Holanda. Basada en la arquitectura empresarial de Java. Es capaz de manejar miles de jugadores de forma simultánea en un servidor, es escalable a cualquier capacidad necesaria. [12]
- La plataforma **Gex** desarrollada por GameFederation, de Estocolmo, Suecia, funciona como un intermediario que provee una solución estandarizada para proveedores de servicios, operadores de telecomunicación y desarrolladores. Gex provee importantes beneficios al cliente permitiéndole a los proveedores distribuir los juegos multiusuario en sus teléfonos y habilitando los juegos multi-jugadores en línea. La plataforma Gex posibilita a los usuarios

registrarse en los juegos, obtener nombres de usuarios, subir sus puntuaciones, participar en torneos y recibir estadísticas.[13]

En este tipo de soluciones informáticas siempre es necesario contar con herramientas que permitan a los administradores tener control sobre los servidores donde se encuentran alojados los juegos. El hecho de que a los usuarios finales les sea indiferente todo lo concerniente a la administración de los servidores, hace que no se divulgue mucho sobre estas aplicaciones en las plataformas mencionadas. A pesar de esto, pudimos constatar algunas características administrativas, por ejemplo, la plataforma **Neutrón** posee aplicaciones que monitorean las conexiones con el servidor, controlando las cuentas de usuarios y cada tipo de juego a los que provee soporte, por otro lado **Gex** brinda la posibilidad de darle seguimiento a las actividades que realicen sus usuarios, ya sea procesos de descarga, de búsqueda o pago a través de la plataforma, de igual forma tienen un control de los torneos establecidos, que pueden servir para atraer la actividad de los suscriptores.

### **SIRIO GPRS**

SIRIO GPRS es la plataforma servidora desarrollada entre el año 2006 y 2007 en la entidad Procyon, su principal función es gestionar, de forma eficiente, las peticiones de los clientes, organizando de forma interna los usuarios conectados por canales correspondientes al juego desde el cual se encuentran autenticados en el sistema. [3]

La plataforma cuenta actualmente con tres módulos fundamentales:

- **Cliente API:** Para el desarrollo de las aplicaciones cliente.
- **Servidor Web:** Es la aplicación encargada de gestionar las peticiones de las aplicaciones y de generar la respuesta correcta a cada acción enviada.
- **Módulo de Administración Web:** Brinda la posibilidad de administrar el servidor a través de una interfaz Web, desde la que se pueden acceder a una serie de funcionalidades, tales como registrar usuario, listar canales, listar usuarios y puntuaciones. Permite al administrador del sistema enviar mensajes de avisos, a todos los usuarios que se encuentren conectados. Facilita la interacción con el servidor y proporciona datos referentes al estado del mismo. [3]

Es importante destacar que el propósito de la entidad es entrar en el mercado y competir con las plataformas descritas anteriormente. Una mayor facilidad de uso y movilidad al utilizar el Módulo de

Administración, se traduciría en un producto mejorado, con un incremento de la capacidad de control y monitoreo del servidor. Para lograrlo, se pretende desarrollar aplicaciones para los teléfonos móviles que presenten Windows Mobile como sistema operativo y para los que soporten aplicaciones J2ME, una en cada caso. Estas ofrecerán nuevas funcionalidades, además de las ya brindadas por el Módulo de Administración Web, incorporarán elementos de seguridad y potenciarán la interoperabilidad con diferentes aplicaciones.

### **1.3 Arquitectura Cliente Servidor**

La arquitectura Cliente/Servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en elementos independientes que cooperan entre sí para intercambiar información, servicios o recursos.

Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

### **1.4 Metodología de desarrollo empleada en el proyecto**

El desarrollo de software es un proceso muy complejo requiriendo de una metodología eficiente y sistemática. Su misión fundamental es brindarnos un entorno de trabajo bien definido y estandarizado, que facilite la obtención de productos que aseguren los requerimientos de calidad, que cubran las exigencias de los clientes y cumpla con el tiempo estimado de desarrollo. Para la realización de la plataforma se decidió el uso de RUP (del inglés Rational Unified Process).

### **1.4.1 Proceso Unificado de Desarrollo de Software**

El Proceso Unificado de Desarrollo de Software (RUP), es la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Constituye una forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo (quién hace qué, cuándo y cómo), y tiene como objetivo asegurar la producción de software de calidad dentro de plazos y presupuestos predecibles. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). [14]

## **1.5 Plataformas de desarrollo.**

Aunque hoy, los teléfonos celulares cuentan con capacidades y prestaciones que hace algunos años eran impensables, aun resultan insuficientes principalmente para los desarrolladores que siempre intentan lograr con estos escasos recursos, aplicaciones que cumplan las expectativas del cliente. Debido a esto se utilizan lenguajes que facilitan un grupo de bibliotecas o paquetes de clases reducidos, que sean capaces de ejecutarse en dichos dispositivos.

El desarrollo de dos aplicaciones similares, en plataformas diferentes, es parte de los requisitos del cliente, para así cubrir una gama más alta de teléfonos. Teniendo en cuenta esto, fueron escogidas las dos plataformas más empleadas actualmente en el desarrollo de software para móviles, J2ME y .Net Compact Framework<sup>9</sup>.

Para el desarrollo de las aplicaciones, se decidió el uso de los lenguajes Java y C#. Además utilizaremos la plataforma J2EE (del inglés Java 2 Enterprise Edition) para la incorporación de un Servicio Web que brinde las funcionalidades de administración del servidor de la plataforma SIRIO.

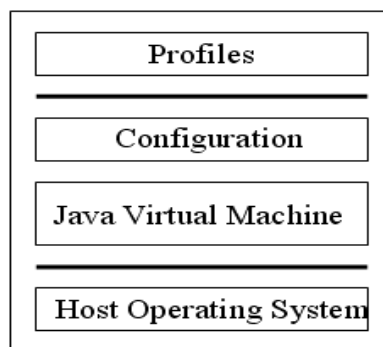
### **1.5.1 ¿Por qué J2ME?**

---

<sup>9</sup> El .NET Compact Framework es una versión "reducida" del .NET Framework y se utiliza en los computadoras de bolsillo (Pocket PC) , o en los equipos que utilicen el Windows CE o el Windows CE .NET.

La plataforma J2ME es una familia de especificaciones que definen varias versiones minimizadas de la plataforma Java 2; estas versiones minimizadas pueden ser usadas para programar en dispositivos electrónicos; desde teléfonos celulares, PDAs, hasta en tarjetas inteligentes, etc. Los cuales no disponen de abundante memoria ni mucha capacidad de procesamiento.

Una de las principales ventajas de J2ME es que presenta una arquitectura modular, figura #1, que se adapta a las limitaciones de los diferentes dispositivos en las que se quiere integrar. Se definen tres capas:



**Figura. # 1 Arquitectura J2ME**

- Máquina virtual: En la actualidad J2ME soporta dos máquinas virtuales: la Java Virtual Machine que se emplea en ediciones J2SE y en J2EE para los dispositivos con procesadores de 32 bit, y la KVM (del inglés Kernel Based Virtual Machine) para arquitecturas de 16/32 bits pero con capacidades de almacenamiento limitado.
- Configuraciones: Definen una serie de bibliotecas Java que están disponibles para un conjunto de dispositivos, con similares capacidades de procesamiento y memoria. J2ME soporta varias configuraciones, en la actualidad existen dos estandarizadas:
  1. Configuración de Dispositivo Limitada por la Conectividad CLDC (del inglés Connected Limited Device Configuration), que engloba en general a dispositivos personales móviles.
  2. Configuración de Dispositivos Conectados, CDC (del inglés Connected Device Configuration), que engloba en general a dispositivos fijos. Por motivos de compatibilidad es un súper conjunto de CLDC.

- Perfiles: Definen un conjunto de API's que pueden emplearse para desarrollar aplicaciones para una familia particular de dispositivos. Este perfil maneja el trabajo con la red, tratamiento con la interfaz de usuario y el almacenamiento de datos locales en Sistema de Administración de Registros, RMS (del inglés Record Manager System).

J2ME provee un robusto ambiente para el desarrollo de aplicaciones, incluye interfaces flexibles, seguridad confiable, protocolos y soporte para redes y aplicaciones que pueden ser descargadas dinámicamente. Es sin duda la plataforma más utilizada para el desarrollo de juegos para dispositivos móviles actualmente y su implementación por parte de los fabricantes en los dispositivos es casi de obligado cumplimiento, debido a la gran demanda del mercado. La elección de J2ME reduce en gran parte el tiempo de aprendizaje y ofrece unas prestaciones aceptables en cuanto al aprovechamiento de recursos se refiere y abarca un amplio número de clientes. [15]

### **1.5.2 ¿Por qué J2EE?**

J2EE (del inglés Java 2 Platform, Enterprise Edition) es, según la definición de Sun Microsystems, un conjunto de estándares y especificaciones para el desarrollo de aplicaciones empresariales basado en la tecnología Java.

La plataforma J2EE ofrece un modelo de aplicaciones distribuidas multinivel, componentes reutilizables, un modelo de seguridad unificado, control flexible de transacciones, y soporte para Servicios Web a través del intercambio integrado de datos con estándares y protocolos basados en XML (del inglés Extensible Markup Language). Usando XML los datos son totalmente portables, y con su adecuado empleo en aplicaciones J2EE se pueden construir sistemas de software robustos y flexibles. J2EE reduce el costo y la complejidad de desarrollo, lo cual se traduce en rapidez. Es totalmente libre. [16]

### **1.5.3 ¿Por qué .NET COMPACT FRAMEWORK?**

.NET Compact Framework aporta la eficacia del entorno de programación .NET Framework a los dispositivos móviles. Es un entorno independiente del hardware para la ejecución de programas en dispositivos de computación con limitaciones de recursos, entre los que se encuentran los asistentes

de datos personales (PDA), teléfonos móviles, decodificadores de televisión y dispositivos de computación para automóviles, que están integrados en el sistema operativo Windows Mobile.

.NET Compact Framework ofrece las siguientes funciones principales:

- Admite protocolos de red comunes y se conecta perfectamente con servicios XML Web.
- Proporciona a los desarrolladores un modelo para orientar sus aplicaciones y componentes ya sea a una amplia gama de dispositivos o a una categoría específica de éstos.
- Facilita el diseño y la optimización de los recursos de sistema limitados. [17]

## **1.6 Lenguajes de programación utilizados**

En el presente epígrafe se llevará a cabo un análisis de los lenguajes de programación que se utilizaron para en el desarrollo del sistema.

### **1.6.1 ¿Por qué Java?**

Java es un lenguaje orientado a objetos que alcanzó su madurez con la popularización de Internet. Su expansión entre la comunidad de programadores ha sido vertiginosa y se ha impuesto como el paradigma de los lenguajes de programación orientados a objetos. Es un lenguaje portable, robusto, estable, independiente de la plataforma, sencillo de aprender para programadores que hayan trabajado previamente con lenguajes orientados a objetos.

### **1.6.2 ¿Por qué C#?**

Para el desarrollo de la aplicación para los teléfonos Windows Mobile utilizamos el lenguaje C#, conocido como el lenguaje estrella de la plataforma .Net.

C# presenta entre otras características:

- Sencillez y Modernidad
- Orientación a objetos y a componentes

- Gestión automática de memoria
- Seguridad de tipos
- Instrucciones seguras
- Sistema de tipos unificado
- Versionable, eficiente y compatible. [18]

### **1.7 Servidor Web**

Un servidor web podría definirse como un proceso capaz de interpretar peticiones que le llegan mediante el Protocolo de Transferencia de Hipertexto HTTP (del inglés Hypertext Transfer Protocol), procesar dichas peticiones y emitir una respuesta HTTP. Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. El uso más habitual de estos es devolvernos páginas web que los usuarios piden introduciendo una dirección URL (del inglés Uniform Resource Locator) a través de un navegador.

Otro uso posible para los servidores web es devolver el resultado de la ejecución de un método que se ha dado de alta en el servidor web y al que ha pedido su ejecución un programa cliente remoto. Las peticiones y respuestas se realizan mediante el protocolo HTTP, pero aquí en vez de que personas pidan páginas web, son los programas los que piden la ejecución de métodos a la máquina servidora.

Para la publicación y alojamiento de los Servicios Web se usará el servidor Web Apache -Tomcat el cual constituye el proyecto de software libre más famoso escrito en Java, puede integrarse fácilmente a muchos sistemas y servidores de forma eficiente, además de ser utilizado por la plataforma SIRIO. [19]

### **1.8 Servicios Web**

A grandes rasgos, un Servicio Web es un conjunto de métodos que realizan una funcionalidad que se exponen al resto de las aplicaciones, XML es el formato estándar para el intercambio de información.



Permiten la comunicación entre aplicaciones o componentes de aplicaciones de forma estándar a través de protocolos comunes (como HTTP) y de manera independiente al lenguaje de programación, plataforma de implantación, formato de presentación o sistema operativo. Presentan independencia del protocolo de transporte.

Los cambios en los métodos de los Servicios Web son transparentes para el programador. El almacenamiento y ejecución de los programas no recae en una sola máquina sino que está repartida por todas las máquinas de red de las que se hace uso. Fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento. [20]

### **1.8.1 Protocolo de Comunicación**

SOAP (del inglés Simple Object Access Protocol) es el protocolo utilizado por los Servicios Web en el intercambio de información. Es un protocolo de comunicación basado en XML, considerado como la columna vertebral de una nueva generación de aplicaciones de computación distribuida, sin tener la dependencia de plataformas, modelos de desarrollo y lenguajes de programación usados, denominada Servicios Web.

No está asociado con ningún lenguaje, los desarrolladores involucrados en nuevos proyectos pueden elegir desarrollar con el último y mejor lenguaje de programación que exista. No se encuentra fuertemente asociado a ningún protocolo de transporte. No está atado a ninguna infraestructura de objeto distribuido. Aprovecha los estándares existentes en la industria, los principales contribuyentes a la especificación SOAP evitaron, intencionadamente, reinventar las cosas, optaron por extender los estándares existentes para que coincidieran con sus necesidades. Por ejemplo, el uso de XML para la codificación de los mensajes.

Permite la interoperabilidad entre múltiples entornos, se desarrolló sobre los estándares existentes de la industria, por lo que las aplicaciones que se ejecuten en plataformas con dicho estándares pueden comunicarse mediante mensaje SOAP con aplicaciones que se ejecuten en otras plataformas. [21]

### **1.9 Protocolo Seguro de Transferencia de Hipertexto (HTTPS)**

Versión segura del protocolo HTTP. HTTPS (del inglés Hypertext Transfer Protocol Secure) utiliza un cifrado basado en Secure Socket Layers (SSL) para crear un canal más apropiado para el tráfico de información sensible.

Funciona gracias a la criptografía, que permite asegurar mediante una serie de funciones matemáticas el origen y la ocultación de los datos.

Cada servidor que quiera implementar HTTPS debe disponer de un certificado, una especie de identidad digital, que indica todos sus datos y la clave que permite cifrar todo lo que enviamos y recibimos del servidor. Estos certificados son emitidos por una serie de compañías a nivel mundial, que garantizan la identidad del propietario del certificado.

### **1.9.1 Certificados Digitales**

Existen tres tipos principales de certificados digitales: certificados autofirmados, certificados generados por infraestructura de llave pública (PKI) de Windows y certificados de terceros.

#### **Certificados autofirmados:**

Un certificado autofirmado lo firma el propio emisor. Estos se deben copiar manualmente al almacén de certificados de raíz de confianza en el equipo cliente o el dispositivo móvil. Cuando un cliente se conecta a un servidor por SSL y el servidor presenta un certificado autofirmado, el mismo lo comprueba en su almacén de certificados de confianza. Si el certificado del servidor está dentro de los que el cliente tiene registrados, este debe confiar explícitamente en la entidad emisora. [22]

#### **Certificados de infraestructura de clave pública de Windows:**

El segundo tipo de certificado es un certificado generado por la PKI de Windows. Una PKI es un sistema de certificados digitales, entidades de certificación y entidades de registro que comprueban y autentican la validez de cada parte implicada en una transacción electrónica, mediante el uso de criptografía con claves públicas. Todo esto, sin embargo, conlleva un coste asociado al tener que implementar servidores e infraestructuras adicionales para crear y administrar los certificados generados por la PKI de Windows. [22]

#### **Certificados de confianza de terceros:**

Los certificados comerciales o de terceros son generados por una entidad certificadora externa o comercial, adquiridos para usarlos en sus servidores de red. La mayoría de los certificados de entidad certificadora comerciales son de confianza porque el certificado ya reside en el almacén de certificados raíz de confianza. Dado que el emisor es de confianza, el certificado también lo es. El uso de certificados de terceros simplifica notablemente la implementación. [22]

Para las organizaciones de mayor tamaño o las que deben implementar certificados de forma pública, el uso de certificados comerciales o de terceros es la mejor solución, a pesar del coste que llevan asociados. Los certificados comerciales pueden no ser la mejor solución para las organizaciones de pequeño o mediano tamaño, para las que puede preferir usar alguna de las otras opciones de certificados disponibles.

En el caso que se está tratando, no se cuenta con la posibilidad de acceder a una entidad certificadora reconocida para obtener un certificado digital, que sería la solución más idónea, ya que mayoría de los dispositivos móviles tienen varios certificados comerciales de confianza de terceros preinstalado y en algunos casos los fabricantes y operadores móviles deshabilitan el acceso al almacén de certificados de raíz de confianza. Esto provoca que no se puedan instalar certificados autofirmados, ni certificados de una PKI de Windows en el dispositivo móvil.

Después de una investigación, se arribó a la conclusión de que el más apropiado para la solución necesaria es el certificado autofirmado ya que este cuenta con un coste mínimo y una configuración sencilla, lo que se ajusta a las condiciones de escasez de presupuesto y falta de experiencia del equipo de desarrollo. Independientemente de esto, presenta como desventaja la dificultad al establecer una infraestructura para la administración del ciclo de vida y la renovación de certificados.

## **1.10 Herramientas de desarrollo**

En el presente epígrafe se llevará a cabo un análisis de las principales herramientas y entornos de desarrollo que se disponen en la actualidad para producir aplicaciones para móviles.

### **1.10.1 J2ME Wireless Toolkit 2.2**

Plataforma ideada por Sun Microsystems cuya última versión salió al mercado a finales de octubre de 2004. Es un conjunto de herramientas que hace posible crear aplicaciones para teléfonos móviles y otros dispositivos inalámbricos.

Componentes:

- **Ktoolbar:** automatiza la mayor parte de las tareas involucradas en la creación de aplicaciones MIDP. Es el centro de la aplicación y se puede usar para construir aplicaciones, lanzar el emulador y empezar las utilidades.
- **Emulador:** se trata de una simulación de un teléfono móvil. Sobre él se evalúan las aplicaciones MIDP.
- **Utilities:** una colección de utilidades que proporciona otros servicios, como pueden ser funciones de criptografía o una consola de mensajes.

### **1.10.2 ActiveSync**

Es el módulo que permite la conexión desde una computadora con un dispositivo móvil Windows Mobile y puede descargarse de forma gratuita desde la Web. A través de esta herramienta se puede probar directamente las aplicaciones en los dispositivos a los cuales va dirigida.

### **1.10.3 Key and Certificate Management Tool (keytool)**

Se encuentra en la instalación del kit de desarrollo estándar de Java. Brinda la posibilidad de crear parejas de claves pública/privada, autofirmar certificados, y manejar el almacén de llaves (keystore). Además, protege cada clave privada con una propia contraseña. [23]

Es una herramienta de sencilla utilización, perfecta para equipos de trabajo sin mucha experiencia. Es libre y no requiere grandes recursos para su ejecución.

### **1.10.4 Entornos de Desarrollo Integrado**

#### **1.10.4.1 Eclipse**

Eclipse es una plataforma universal para integrar herramientas de desarrollo que corren sobre un amplio rango de sistemas operativos, con una arquitectura abierta y basada en plug-ins. La característica clave del Eclipse es su extensibilidad a través de los plug-ins que no son más que la mínima unidad de la plataforma que puede ser desarrollado por separado y que le aporta una nueva funcionalidad. [24]

Además conserva el registro de las versiones, genera y mantienen la documentación de cada etapa del proyecto. Gestiona los proyectos realizados en espacios de trabajo, de forma que resulta fácil crear, modificar y eliminar recursos de la estación de trabajo desde el entorno de desarrollo. Compila el código en tiempo real, lo que significa que a medida que se va programando el sistema muestra los errores que se van cometiendo optimizando el tiempo de desarrollo.

Para el desarrollo del proyecto se hará uso del plug-in Eclipse ME para el desarrollo de aplicaciones en J2ME. Este plug-in se integra con J2ME Wireless Toolkit permitiendo un cúmulo de facilidades entre las cuales se pueden destacar, la facilidad de depurar las aplicaciones y emular las mismas en una amplia gama de dispositivos.

#### **1.10.4.2 Visual Studio 2005 Professional Edition**

El entorno de desarrollo Visual Studio 2005 fue el escogido para el desarrollo de la aplicación en .Net Compact Framework ya que se amplían las capacidades de desarrollo para dispositivos móviles (PDA y Smartphone<sup>10</sup>, ahora una misma plataforma unificada), y se presentan nuevas herramientas que nos facilitaran el trabajo. Además .NET Compact Framework está totalmente integrado en Visual Studio .NET 2005 y versiones anteriores. Durante el desarrollo de una aplicación, no es necesario implementarla en un teléfono real para realizar las pruebas, ya que se dispone de un emulador integrado. El comportamiento y la velocidad del emulador ofrece una indicación precisa sobre cómo funcionaría la aplicación en un teléfono móvil real. [25]

### **1.11 Lenguaje Unificado de Modelación**

---

<sup>10</sup> Teléfono móvil que ofrece funcionalidades avanzadas que van más allá de las de un teléfono tradicional y están más cercanas a las de una computadora.

UML (del inglés Unified Modeling Language) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Permite la modelación de sistemas con tecnología orientada a objetos. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática. Incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.

### **1.12 Herramientas de modelado UML**

La creciente complejidad de los sistemas informáticos representa un reto importante para los ingenieros y arquitectos del software. Los modelos proporcionan un mayor nivel de abstracción, permitiendo trabajar con sistemas mayores y más complejos, y facilitando el proceso de codificación e implementación del sistema de forma distribuida y en distintas plataformas. Para el modelado de los artefactos y diagramas generados a lo largo del ciclo de vida del proyecto se decidió el uso de Rational Rose Enterprise Edition 2003, ya que su uso está muy estandarizado a nivel mundial y constituye una herramienta más madura y acabada, además el grupo de desarrollo se encuentra familiarizado con el uso de la misma.

#### **1.12.1 Rational Rose Enterprise Edition 2003**

Es una herramienta de diseño orientada a objetos, que da soporte al modelado visual, es decir, que permite representar gráficamente el sistema, permitiendo hacer énfasis en los detalles más importantes, centrándose en los casos de uso y enfocándose hacia un software de mayor calidad, empleando un lenguaje estándar común que facilita la comunicación. A través del uso de la notación UML, permite la creación y refinamiento de vistas dentro de modelo general del dominio del sistema de software. Este modelo global está compuesto por clases, casos de uso, objetos, paquetes y otros componentes, así como relaciones que posibilitan describir las características más detalladas del sistema. Además nos brinda un desarrollo en paralelo de los modelos posibilitando su descomposición en unidades más pequeñas y permitiendo la integración con sistemas de control de versiones.

### **1.13 Herramientas para el control de versiones**

Una versión, revisión o edición de un producto, es el estado en el se encuentra en un momento dado en su desarrollo o modificación. Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas (por ejemplo, para algún cliente específico). Un sistema de control de versiones proporciona mecanismos de almacenamiento y posibilita la edición para cada objeto bajo control, además permite almacenamiento de la línea de cambios por los que ha pasado el objeto, de forma tal que posibilite ir a una versión anterior en condiciones determinadas.

### **1.13.1 SVN**

Subversion es uno de los sistemas de control de versiones más modernos y utiliza un sistema con repositorio centralizado y fue diseñado como reemplazo del sistema más utilizado hasta la fecha, CVS (del inglés Concurrent Versions System).

Algunas de sus principales ventajas son:

- Se sigue la historia de los archivos y directorios a través de copias y renombrados.
- Las modificaciones son automáticas.
- Se envían sólo las diferencias en ambas direcciones.
- Puede ser servido, mediante Apache. Esto permite que sea accedido desde la web.
- Maneja eficientemente archivos binarios.
- Permite selectivamente el bloqueo de archivos. Se usa en archivos binarios que, al no poder fusionarse fácilmente, conviene que no sean editados por más de una persona a la vez. [26]

### **1.13.2 Cliente de Subversion utilizado**

Tortoise SVN es un cliente gratuito de código abierto para el sistema de control de versiones Subversion. Maneja ficheros y directorios a lo largo del tiempo. Los ficheros se almacenan en un

repositorio central. El repositorio es prácticamente lo mismo que un servidor de ficheros ordinario, salvo que recuerda todos los cambios que se hayan hecho a sus ficheros y directorios. Esto permite que pueda recuperar versiones antiguas de sus ficheros y examinar la historia de cuándo y cómo cambiaron sus datos, y quién hizo el cambio. [27]

#### **1.14 Conclusiones.**

Durante la descripción del capítulo se detallaron y argumentaron los principales conceptos y aspectos tratados a lo largo de la investigación y que están relacionados con la solución propuesta. Se hizo especial énfasis en el análisis de las soluciones brindadas por otras empresas a la problemática planteada, señalando sus principales características. Se abordaron temas relacionados con la seguridad en la transferencia de información, así como las principales tendencias y herramientas que existen actualmente a nivel mundial.

## **CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA**

### **2.1 Introducción**



En el presente capítulo se tratan los temas referentes al objeto de estudio, haciendo énfasis en los procesos involucrados en el campo de acción, así como un análisis de la ejecución del mismo. Se describen las características del sistema que fueron obtenidos durante la modelación del dominio y que recogen los temas referentes a las necesidades de los clientes, así como las acciones que se deben llevar a cabo para su solución y que finalmente conllevan a que se realice una propuesta del sistema.

## **2.2 Objeto de automatización**

En la entidad Procyon con el desarrollo de la plataforma SIRIO se avanzó un escalón más en el desarrollo de juegos multiusuario, permitiendo conectarse desde largas distancias a través de la red GPRS. Para hacer uso del Módulo de Administración Web que propone esta plataforma, es necesario contar con la presencia física de una computadora, de esta forma se limita sus capacidades de uso, con el objetivo de facilitar el monitoreo constante del servidor, se decidió el desarrollo de un sistema orientado a teléfonos celulares que brinde las funcionalidades de dicho módulo, que provea seguridad en las transferencias de datos e interoperabilidad con diferentes aplicaciones.

## **2.3 Propuesta del sistema**

### **Descripción general.**

**Módulo de Administración Móvil:** Brinda la posibilidad de administrar el servidor a través de un teléfono celular, se pueden acceder a una serie de funcionalidades, tales como, listar canales, listar juegos y usuarios. Permite al administrador del sistema enviar mensajes de avisos, a todos los usuarios que se encuentren conectados. Posibilita además la visualización de estadísticas de un juego determinado y funciones para la gestión de cuentas de usuarios con permisos de administración. Facilita la interacción con el servidor y proporciona a los administradores todos los datos referentes al estado del servidor.

Con el objetivo de cubrir el mayor número posible de teléfonos se decidió el desarrollo de dos aplicaciones, una para los dispositivos que soporten Java y otra para los que presenten Windows Mobile como sistema operativo.

Se desarrolló además, un Servicio Web que brinda la información referente al estado del servidor que presenta relevancia para la aplicación cliente.

Para garantizar la seguridad en la comunicación entre el Servicio Web y las aplicaciones que lo van a consumir se decidió el uso del protocolo HTTPS.

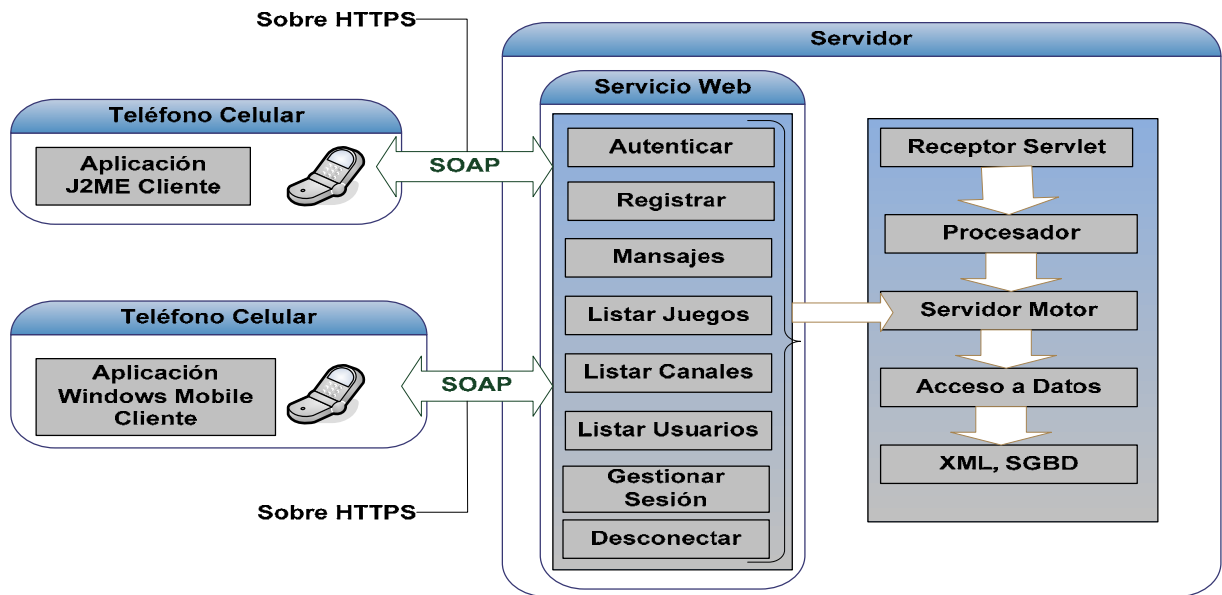


Figura. # 2 Propuesta de Sistema.

### Análisis comparativo

Anteriormente las aplicaciones administrativas de la plataforma SIRIO intercambiaban información directamente con el servidor de juegos a través de Servlets (código en Java que se ejecuta en un servidor web), estos contienen la lógica de programación y aunque mantienen asegurado el código, este no es independiente de la interfaz. No definía métodos para la aseguración de las transferencias de datos en la red, provocando falta de seguridad y confianza.

La nueva propuesta está basada en una idea novedosa, la creación de un Servicio Web para servir de intermediario entre las aplicaciones y el servidor. Proporcionará las funcionalidades anteriormente implementadas, además provee un sistema que brinda interoperabilidad, permitiendo a cualquier tipo de aplicación utilizar el servicio, convirtiéndolo en una solución más madura y adaptable a las necesidades del mundo actual, en este tipo de aplicaciones. El nuevo Módulo de Administración constará de dos aplicaciones orientadas a teléfonos móviles para el monitoreo del estado del servidor, de esta forma se incrementa la ubicuidad del sistema, al poder hacer uso del mismo en todo lugar

donde el celular tenga cobertura. Se propone además, la utilización de un protocolo seguro, como HTTPS, para asegurar la integridad y la confidencialidad de los datos en el intercambio información entre el Servicio Web y las aplicaciones que lo consumen.

El sistema propuesto subsana aspectos que quedaron inconclusos en la primera versión del Módulo de Administración, ganando en interoperabilidad, ubicuidad y seguridad. Convirtiéndolo en un producto superior y mejorado, con más atractivo para los posibles clientes y de este modo con mayores posibilidades de generar ingresos.

## **2.4 Modelo de Negocio**

Al dividir un problema en pequeñas piezas, se reduce su complejidad y se aumenta la facilidad para resolverlo. Esas piezas se pueden representar a través de modelos que permitan abstraer sus características esenciales. En el campo del software también resulta útil la creación de modelos que organicen y representen los detalles importantes de problemas reales que se vinculan con el sistema informático a construir. Estos modelos deben cumplir una serie de propiedades, entre ellas la de ser coherentes y relacionados. Uno de los modelos útiles previo al desarrollo de un software es el modelo del negocio.

Dependiendo de la situación o escenario que se presente, hay varias alternativas de desarrollar este proceso, no siempre es necesaria o posible, la completa realización de un modelo de negocio. En ese caso se realizará lo que se conoce como un modelo del dominio. [28]

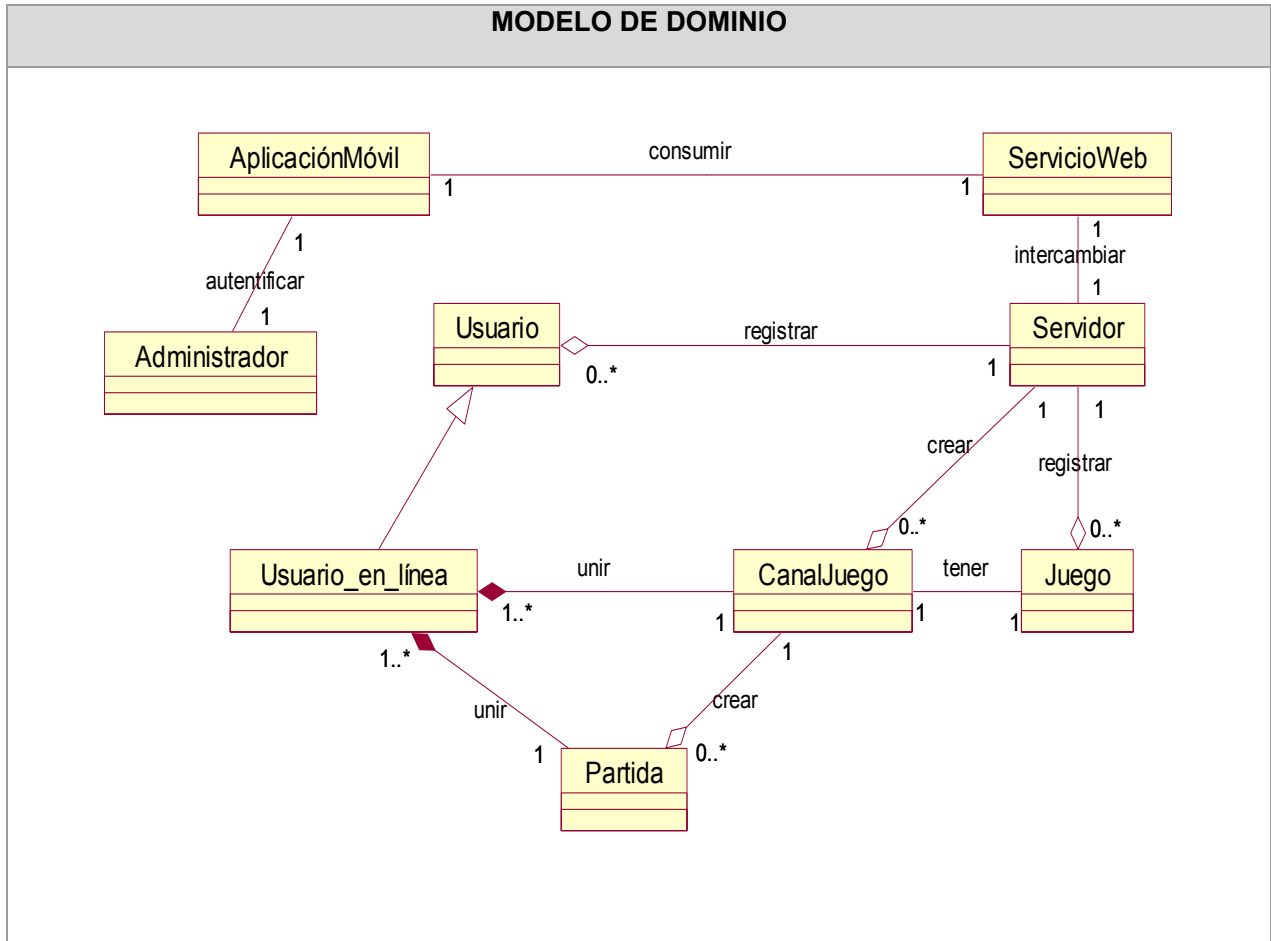
## **2.5 Modelo de Dominio**

El modelo de dominio muestra las clases conceptuales significativas en el dominio del problema, captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema. Es considerado un subconjunto del llamado modelo de objetos del negocio.

Durante el desarrollo de la aplicación no se pudo contactar procesos bien definidos en el entorno del negocio. Se hizo difícil determinar los elementos más importantes del sistema y sus interconexiones, así como el establecimiento de las reglas de funcionamiento. Sin embargo podemos identificar personas, eventos, transacciones y objetos involucrados en ese entorno que no está bien delimitado,

por lo que se hizo necesario un modelado del dominio perteneciente a la solución [28], mostrado en la tabla 1.

Tabla 1. Modelo de Dominio



### 2.5.1 Descripción de los conceptos del dominio

Administrador: Define actor que engloba a las acciones relacionadas con el administrador del sistema.

AplicaciónMóvil: Define aplicación informática destinada a teléfonos celulares.

ServicioWeb: Define responsable de manejar el intercambio de información entre las aplicaciones y el servidor.

Servidor: Define responsable de recibir y procesar todas las peticiones de los clientes así como controlar usuarios conectados y el intercambio de información entre los mismos.

Usuario: Define usuario que posee una cuenta en el servidor.

Partida: Define entendimiento entre 2 ó más jugadores para competir en determinado juego.

Usuario\_en\_línea: Define usuario que ha pasado por el proceso de autenticación y sus credenciales son correctas.

CanalJuego: Define espacio en el servidor donde se posibilita el intercambio de mensajes entre jugadores de un mismo juego.

Juego: Define juego que ha sido gestionado en el servidor al menos una ocasión.

## **2.6 Requerimientos**

Los requerimientos son la condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo, y las propiedades o cualidades que debe ser reunida o poseída por un sistema o componente del sistema para satisfacer un contrato, especificación, estándar, u otra documentación formal.

### **2.6.1 Requerimientos funcionales**

Después de un análisis de las principales necesidades de los clientes y las características que el sistema debe cumplir se definió un listado de las capacidades o condiciones que deben estar presentes en él.

RF1- Gestionar Usuario

RF.1.1- Autenticar Usuario.

RF.1.2- Registrar Usuario.

RF.1.3- Cambiar Contraseña.

RF.1.4- Desconectar usuario.

RF2- Enviar Mensaje de texto.

RF.2.1 Enviar Mensaje de texto a un usuario determinado.

RF.2.2 Enviar Mensaje de texto a todos los usuarios en un canal de juego.

RF.2.2 Enviar Mensaje de texto a todos los usuarios conectados.

RF3- Gestionar Reportes.

RF.3.1- Listar usuarios conectados a un canal.

RF.3.2- Listar usuarios que se encuentran jugando un juego.

RF.3.3- Listar canales habilitados en el servidor.

RF.3.4- Listar juegos disponibles en un canal.

RF.3.5- Listar juegos iniciados en un canal.

RF.3.6- Listar estadísticas de un juego.

RF4- Gestionar Sesión.

RF.4.1- Crear sesión.

RF.4.2- Actualizar sesión.

RF.4.3- Verificar sesión.

RF.4.4- Desconectar sesión.

## **2.6.2 Requerimientos no funcionales**

Con el propósito de satisfacer al máximo las exigencias de los clientes así como la calidad del mismo se definió un listado de las propiedades o cualidades que el producto debe tener, descritas a continuación.

- Usabilidad: La aplicación está orientada a los responsables de la administración y control del estado del servidor, los cuales deben poseer conocimientos de manejo de la tecnología celular.
- Soporte: Se requiere una documentación apropiada que describa todas las funcionalidades del sistema desarrollado así como una guía para su uso.
- Portabilidad: El sistema debe ser portable en una amplia gama de dispositivos móviles.
- Seguridad: El sistema debe proteger la integridad de la información que se maneja y no permitir el acceso no autorizado, exigiendo autenticación antes de realizar cualquier acción.
- Disponibilidad: El sistema debe contar con una disponibilidad ininterrumpida mientras el servidor de juegos este desplegado.
- Software: Debe utilizarse el contenedor Web Apache Tomcat, además de la Máquina virtual Java JDK 6.
- Hardware: Mínimo 256 MB de memoria RAM y procesador 1 GHz

Requerimientos no funcionales para los teléfonos que soporten Java.

- Software: Los dispositivos deben poseer configuración CLDC 1.2, Profile MIDP 2.0 como mínimos y soporte para mensajes SOAP.
- Hardware: Los dispositivos deben pertenecer a la generación 2.5 o superior a esta.

Requerimientos no funcionales para los teléfonos que presenten Windows Mobile como sistema operativo.

- Software: Los dispositivos deben poseer alguna de las siguientes versiones del sistema operativo: Windows Mobile 2003, Windows Mobile 2003 Second Edition, Windows Mobile 5.0, Windows Mobile 6.

## **2.7 Modelo del Sistema. Definición de casos de uso**

Una de las situaciones más difíciles a la hora de construir un sistema es precisamente su modelación. El modelo de casos de uso del sistema representa las funcionalidades deseadas y el entorno del sistema a través de actores y casos de uso. Sienta las bases necesarias para el desarrollo del análisis y el diseño del sistema.

La identificación de los casos de uso es precisamente la guía del Ingeniero de Software que lleva adelante el desarrollo de un sistema informático.

### **2.7.1 Definición de actores**

Los actores del sistema constituyen personas o sistemas que fueron trabajadores del negocio y que interactúan de alguna forma con el sistema y que están asociadas al cumplimiento de los requisitos funcionales o procesos que responden a las funcionalidades definidas en los mismos.

**Tabla 2. Definición de actores del sistema.**

<b>Actores</b>	<b>Justificación</b>
Administrador	Define actor que engloba a las acciones relacionadas con el administrador de la plataforma.
Reloj	Define actor que determina cada cierto tiempo la verificación de la validez de las sesiones de usuarios.

### **2.7.2 Casos de uso del sistema**

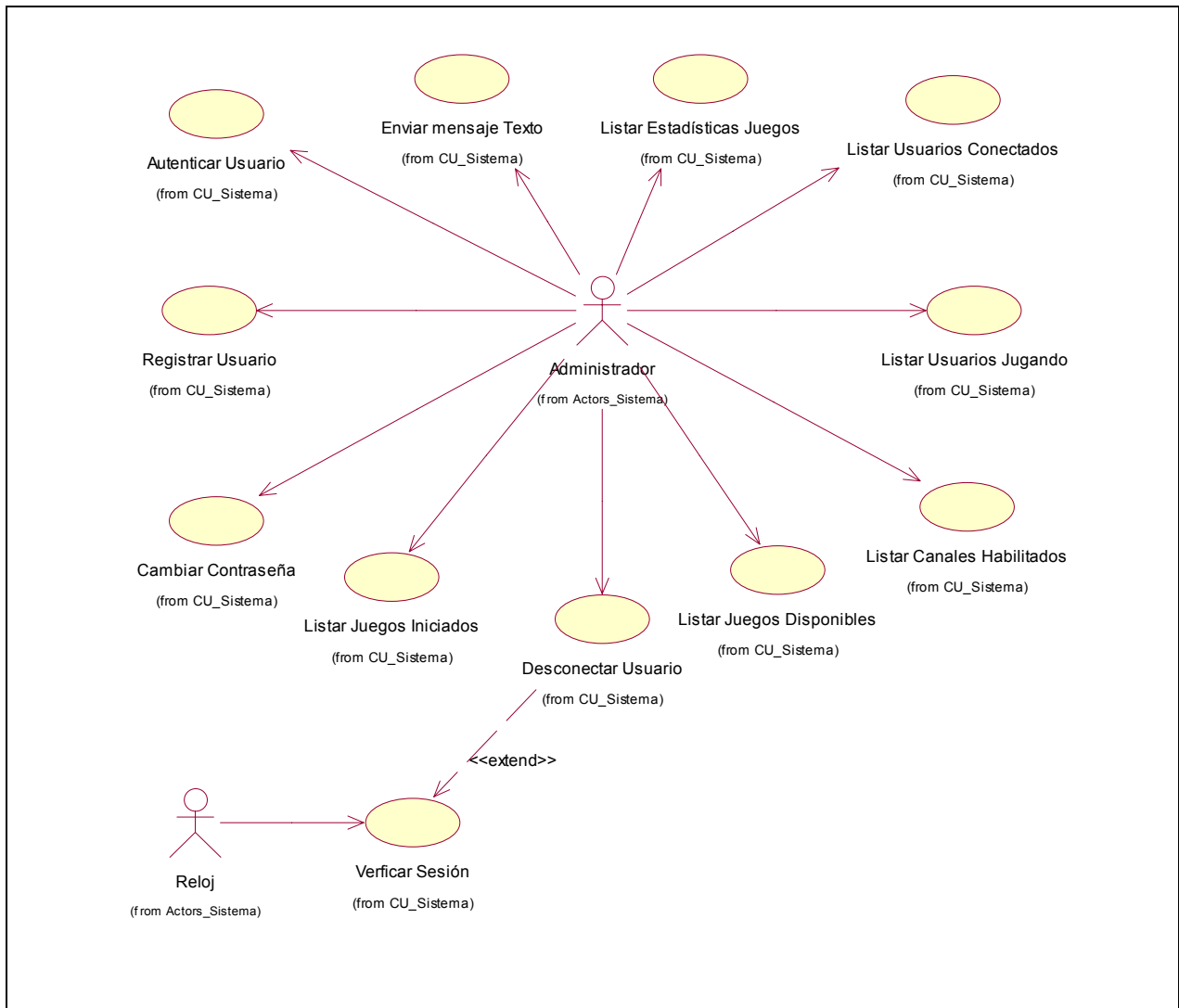
Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema.



Para la realización del módulo se definieron un conjunto de acciones que deben ser ejecutas por distintos actores y que desencadenan un conjunto de operaciones. Las interrelaciones entre las acciones y los actores del sistema son agrupadas en el diagrama de casos de uso que se muestran a continuación.

**Tabla 3. Diagrama de casos de uso del sistema**

<b>DIAGRAMA DE CASOS DE USO DEL SISTEMA</b>
---



### 2.7.3 Descripción detallada de los casos de uso del sistema

Para comprender la funcionalidad que engloba cada caso de uso, no basta con la representación gráfica. Debe ser elaborada una descripción textual que abarque las principales características de los mismos. El formato de descripción que se propone en el anexo #1, recoge los elementos más importantes que detallan el flujo de eventos de los caso de uso.

A continuación se muestra la descripción del caso de uso: **Listar Canales Habilitados.**

Tabla # 4. Descripción detallada del CUS: Listar Canales Habilitados.

<b>Nombre del Caso de Uso</b>		Listar Canales Habilitados.
<b>Actores</b>	Administrador	
<b>Propósito</b>	Listar todos los canales que se encuentran activos en el servidor.	
<b>Resumen</b>	El caso de uso se inicia cuando el administrador solicita listar todos los canales que se encuentran habilitados en un momento dado.	
<b>Referencias</b>	RF.3.3, RF.4.2, RF.4.3	
<b>Precondiciones</b>	Administrador previamente autenticado en el servidor.	
<b>Poscondiciones</b>	Lista de usuarios en línea.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>		<b>Respuesta del Sistema</b>
1- El administrador solicita al sistema un listado de los canales de juego que se encuentren habilitados.		1.1- El sistema recibe la petición, verifica que el actor posea una sesión válida, de ser positivo disminuye tiempo de inactividad del usuario.
		1.2- El sistema muestra el listado.
<b>Flujo Alternativo:</b>		
Acción 1.1:		1.1a Si la sesión del usuario ha expirado, se llama al caso de uso: Autenticar Usuario.
<b>Prioridad:</b>	Primaria	

## 2.8 Conclusiones

Durante la descripción del capítulo se argumentaron los principales aspectos relacionados con las características más generales del sistema, se desarrolló un profundo análisis de los principales

procesos que se llevan a cabo dentro de la entidad Procyon. Se modelaron los conceptos más importantes dentro del entorno, se describieron las funcionalidades que deben cumplir el sistema y sus capacidades.

## **CAPITULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA**

### **3.1 Introducción**

En este capítulo se describen los procesos que se llevan a cabo en el flujo de trabajo de análisis y diseño. Se elaboran los diagramas que describen el análisis y diseño de las clases, así como los que representan la interacción entre objetos, siguiendo el criterio de distribución por caso de uso. Además se crea un conjunto de acciones para mitigar los posibles errores en el proceso de intercambio de información con el servidor.

### **3.2 Flujo de Trabajo Análisis y Diseño**

El objetivo principal de este flujo de trabajo es transformar los requerimientos a una especificación que describa cómo implementar el sistema. El análisis, fundamentalmente consiste en obtener una visión que se preocupa de ver que hará el sistema de software a desarrollar, por tal motivo este se interesa en los requerimientos funcionales. Por otro lado, el diseño es un refinamiento que toma en cuenta los requerimientos no funcionales, por lo cual se centra en como el sistema cumple sus objetivos. [28]

Los objetivos específicos del análisis y diseño son:

- Transformar los requerimientos al diseño del futuro sistema.
- Desarrollar una arquitectura para el sistema.
- Adaptar el diseño para que sea consistente con el entorno de implementación.

### **3.3 Modelo de análisis**

El modelo de análisis se centra en el tratamiento de requisitos funcionales y pospone los no funcionales para el diseño. Esboza cómo llevar a cabo la funcionalidad dentro del sistema, incluidas las funcionalidades significativas para la arquitectura, sirve como una primera aproximación del diseño y es el resultado de la actividad de analizar los casos de uso.

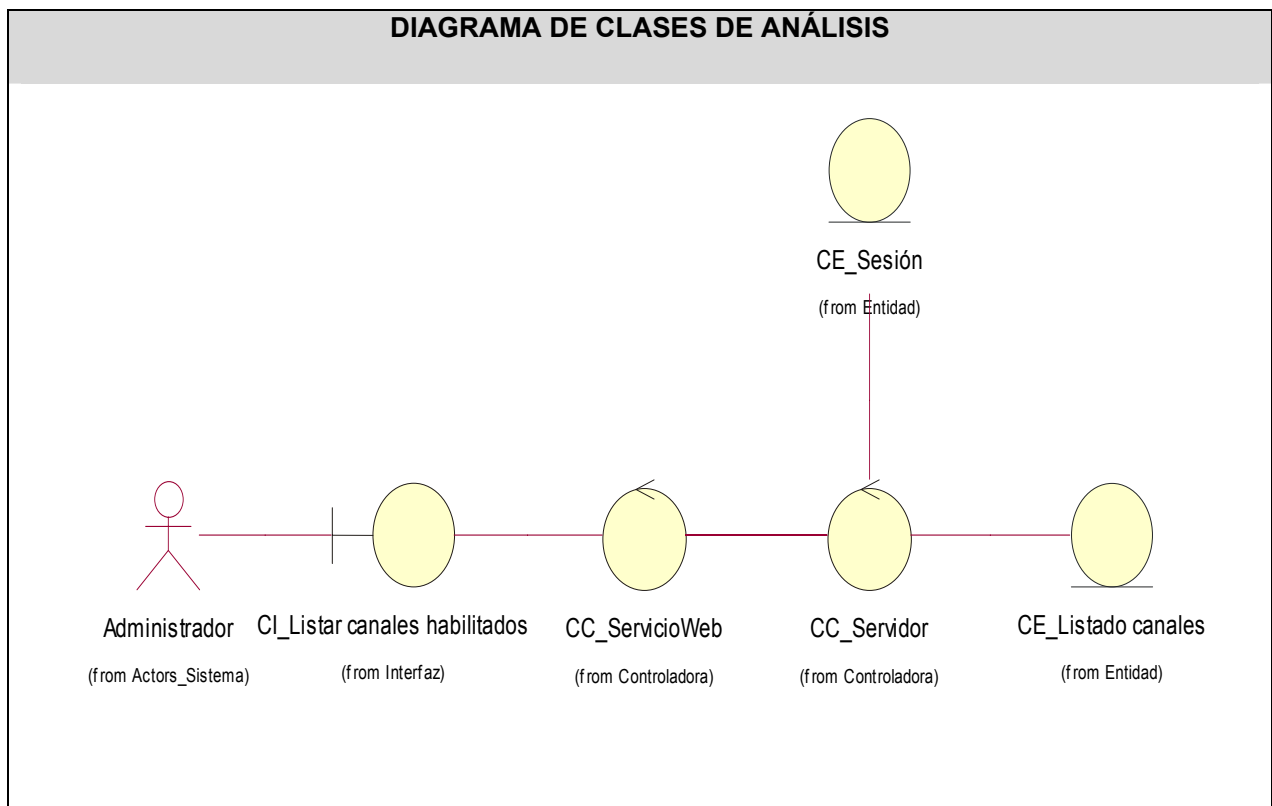
En la construcción del modelo de análisis se tienen que identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el diagrama de clases del análisis, estas representan los conceptos en un dominio del problema, sin llegar al nivel de detalle necesario que se persigue en el diseño, constituyendo una primera aproximación de este. Una clase de análisis representa una abstracción de una o varias clases

y/o subsistemas del diseño y según RUP siempre encajan en alguno de los estereotipos: interfaz, control y entidad. [28]

Para el desarrollo del sistema se realizó de un diagrama de clases del análisis para cada caso de uso. A continuación se ilustra el diagrama de clases del análisis del caso de uso Listar Canales Habilitados.

### 3.3.1 Diagrama de clases del análisis

Tabla 5. Diagrama de clases del análisis. CUS Listar Canales Habilitados.



### 3.4 Modelo de Diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso. Se centra en cómo los requisitos funcionales y no funcionales tienen impacto en el sistema a desarrollar.

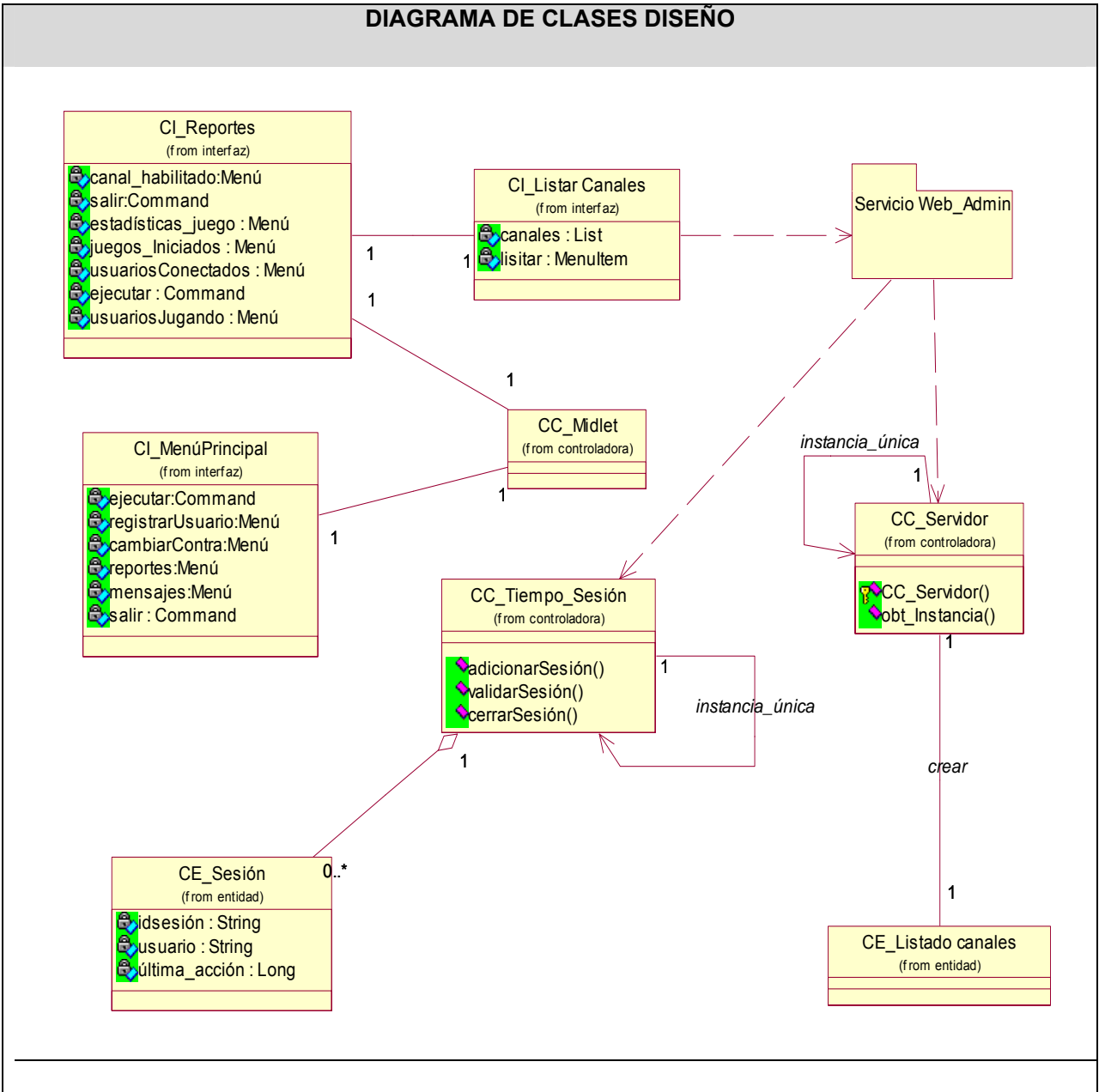
Dentro de sus propósitos están: crear una entrada apropiada y un punto de partida para la implementación, descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo. Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia.

La realización de caso de uso del diseño contiene una descripción de flujos de eventos textual, diagramas de clases y diagramas de interacción. Los diagramas de clases son los artefactos más utilizados en el modelado de sistemas orientados a objetos. Un diagrama de clases representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas. Las clases de diseño se especifican utilizando la sintaxis del lenguaje de programación elegido. [28]

Para el desarrollo del sistema se llevó a cabo la realización de un diagrama de clases del diseño y un diagrama de secuencia para cada caso de uso. En el sistema se cuentan con dos diseños diferentes, debido a que se desarrollarán aplicaciones con funcionalidades similares en dos plataformas distintas, con el objetivo de cubrir una gama más amplia de móviles. La figura que se muestra a continuación, ilustra el diagrama de clases del diseño del caso de uso Listar Canales Habilitados para la plataforma J2ME. En el anexo #3 están reflejados los diagramas de clases diseños más importantes correspondientes al sistema, incluidos los destinados a teléfonos móviles que presenten Windows Mobile como sistema operativo.

### **3.4.1 Diagrama de clases del diseño**

**Tabla 6. Diagrama de clases del diseño (J2ME). CUS Listar Canales Habilitados**



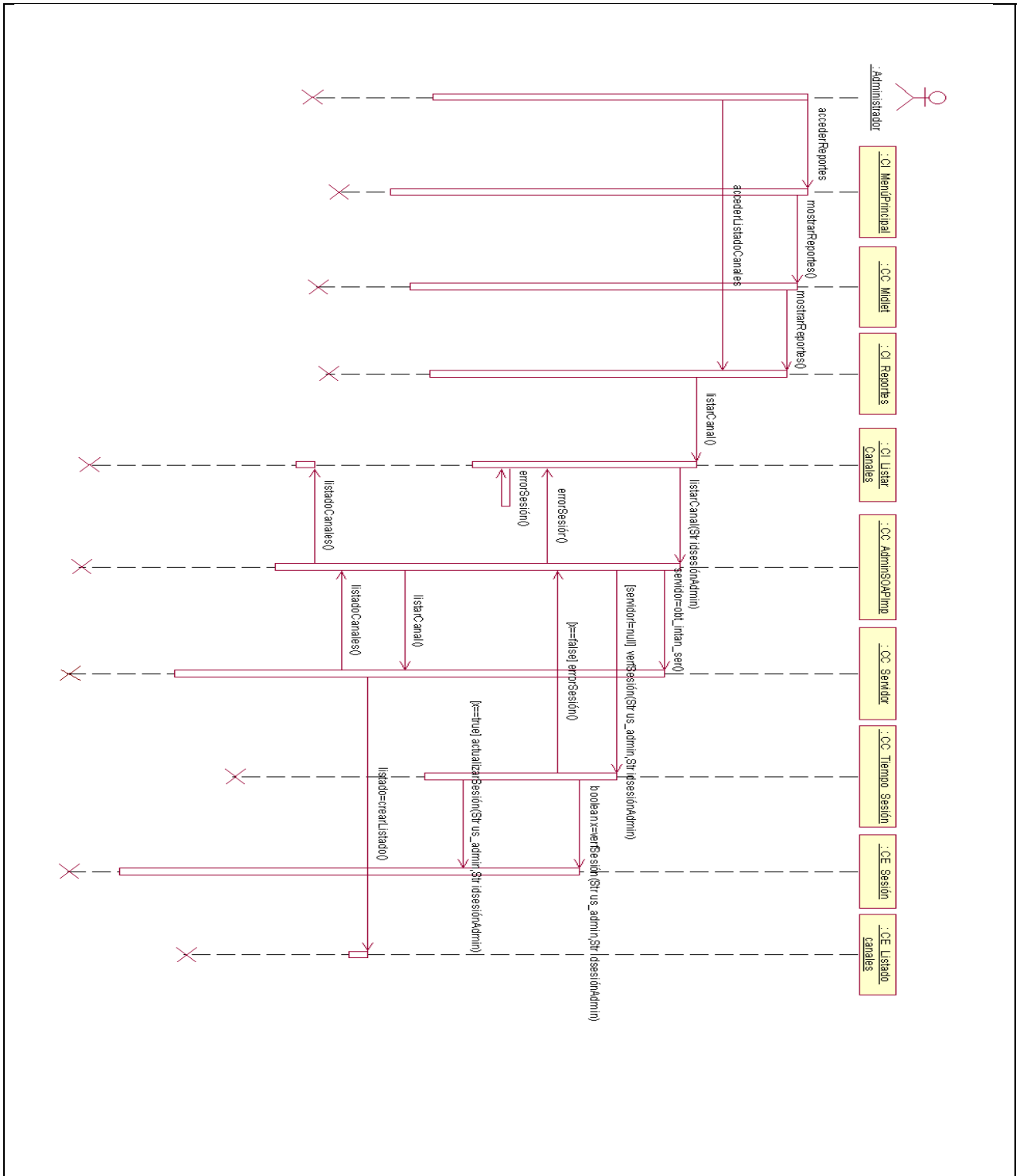
### 3.4.2 Diagrama de Secuencia



Un diagrama de secuencia muestra las interacciones entre objetos, ordenadas en secuencia temporal durante un escenario concreto. A continuación, se ilustra el diagrama de secuencia del caso de uso Listar Canales Habilitados.

**Tabla 7. Diagrama de secuencia (J2ME). CUS Listar Canales Habilitados**

**DIAGRAMA DE SECUENCIA**



3.5 Decisiones de diseño empleados en el desarrollo del sistema

La calidad del diseño de la interacción entre los objetos y la asignación de responsabilidades, presentan gran variación, decisiones poco acertadas, dan origen a sistemas y componentes frágiles y difíciles de mantener, entender, reutilizar o extender. Los diseñadores expertos en orientación a objetos van formando un amplio repertorio de principios generales y de expresiones que los guían al crear software, a algunos podemos asignarles el nombre de patrones, estos se codifican en un formato estructurado que describe el problema y su solución.

"...describa un problema que sucede una y otra vez en nuestro entorno, y luego describa el núcleo de la solución a ese problema, de tal forma que pueda utilizar esa solución un millón de veces más, sin siquiera hacerlo dos veces de la misma manera." - Christopher Alexander

Actualmente el uso de los patrones de diseño se ha generalizado, son utilizados y están presentes en casi todos los sistemas informáticos de importancia a nivel mundial.

Un patrón de diseño es una abstracción de una solución en un nivel alto que solucionan problemas que existen en muchos niveles de abstracción. En la terminología de objetos, el patrón es una descripción de un problema y su solución, que recibe un nombre y que puede emplearse en otros contextos. [29]  
[30]

En el desarrollo de la solución planteada, se utilizaron varios patrones de diseño con el objetivo de lograr un software más acabado, robusto, flexible y de mayor calidad. Entre ellos podemos encontrar los patrones GRASP (del inglés: General Responsibility Assignment Software Patterns) que describen los principios fundamentales de la asignación de responsabilidades a objetos y los pertenecientes al grupo o pandilla de cuatro, GOF (del inglés: Gang of Four).

A continuación se describen las principales características de los patrones utilizados.

#### **GRASP:**

**Experto:** Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

Durante el diseño orientado a objetos, cuando se definen las interacciones entre los objetos, tomamos decisiones sobre la asignación de responsabilidades a las clases. Si se hacen en forma adecuada, los sistemas tienden a ser más fáciles de entender, mantener y ampliar, y se presenta la oportunidad de reutilizar los componentes en futuras aplicaciones.

El uso de este patrón trae como beneficios la conservación del encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento. Además el comportamiento se distribuye entre las clases que cuentan con la información requerida, alentando con ello definiciones de clases sencillas y cohesivas que son más fáciles de comprender y de mantener. Así se brinda soporte a una alta cohesión.

**Creador:** Asignar a una clase B la responsabilidad de crear una instancia de una clase A.

La creación de objetos es una de las actividades más frecuentes en un sistema orientado a objetos. En consecuencia, conviene contar con un principio general para asignar las responsabilidades concernientes a ella. El diseño, bien asignado, puede soportar un bajo acoplamiento, una mayor claridad, el encapsulamiento y la reutilización de código.

Con su utilización se brindó soporte a un bajo acoplamiento (que se describirá más adelante), lo cual supone menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización.

**Bajo Acoplamiento:** Asignar la menor cantidad de responsabilidades a cada clase y así evitar que esta recurra a muchas otras.

El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. El Bajo Acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad. No puede considerarse en forma independiente de otros patrones como Experto o Alta Cohesión, sino que más bien ha de incluirse como uno de los principios del diseño que influyen en la decisión de asignar responsabilidades.

**GOF:**

**Cadena de responsabilidad:** Enviar una petición a través de la cadena de objetos relacionados hasta que llegue a la clase que contenga la información necesaria para su realización. Permite a un sistema determinar, en tiempo de ejecución, el objeto que se encargará de un mensaje. Cada objeto de la cadena puede manejar el mensaje o pasarlo al siguiente objeto de la cadena.

**Única Instancia:** Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Por lo tanto el mismo objeto es siempre compartido por distintas partes del código.

Los Patrones de Diseño posibilitan la reutilización del conocimiento, su utilización como guía para el desarrollo de software garantiza un producto robusto y acabado, su buen uso trae aparejado facilidades de reutilización, mantenimiento y comprensión, por estas razones es recomendado que sean estudiados y aplicados con frecuencia en los proyectos informáticos.

### 3.6 Tratamiento de errores

El Módulo de Administración Móvil, hace uso del sistema de control para el tratamiento de los posibles errores que puedan presentarse durante la ejecución de alguna acción en el servidor o en el proceso de comunicación con el mismo que posee la plataforma SIRIO, además consta de un tratamiento para ellos en las aplicaciones clientes de los teléfonos móviles.

En el desarrollo del sistema se han definido una serie de acciones que pueden provocar un mal funcionamiento del servidor, para cada una de ellas se definió un mecanismo de recuperación, que asegurará una respuesta adecuada para garantizar cierto grado de estabilidad. Los mensajes de error llegan al usuario en un lenguaje común, con el objetivo de facilitar la comprensión.

En la siguiente tabla se muestran algunos de los principales errores posibles:

**Tabla 8. Descripción de errores**

<b>Error</b>	<b>Causa</b>	<b>Respuesta del Sistema</b>
ERROR_USUARIO_DUPLICADO	El cliente intenta acceder al sistema con un usuario que se encuentra en línea.	Notificar al usuario que intenta acceder al sistema, que las credenciales se encuentran en uso.
ERROR_USUARIO_DESCONOCIDO	El cliente trata de acceder al sistema con un usuario	Notificar al usuario que no se encuentra

	desconocido, que no ha sido registrado en el sistema.	registrado en el servidor.
ERROR_CONTRASEÑA _ EQUIVOCADA	El cliente trata de acceder al sistema con una contraseña que no concuerda con la suya.	Notificar al usuario que la contraseña introducida no se corresponde con la almacenada en el servidor.
ERROR_CONEXION	Puede suceder por múltiples causas que involucran principalmente problemas en la red del cliente que se afecte la conexión con el servidor.	Notificar al cliente que ha perdido la conexión con el servidor.
ERROR_SISTEMA	Error grave del sistema que impide que su correcto funcionamiento.	El sistema intenta recuperar la información y restablecer los servicios.

### **3.7 Seguridad**

Cuando se utilizan servicios web la seguridad es algo primordial, debido a que los mensajes SOAP son transmitidos en texto plano por la red, estos pueden ser interceptados, poniendo en peligro la confidencialidad de la información. Para evitar esto, fueron creadas una serie de acciones que fortalecen todos los mecanismos de seguridad definidos en la plataforma SIRIO.

Para mitigar este problema surgió la idea de utilizar HTTPS para asegurar la transferencia de información. El proceso de establecimiento de una comunicación segura utilizando SSL se describe a continuación.

Se utilizó la opción de certificado autofirmado, una vez creado se le incluye en la raíz de certificados de confianza de los teléfonos móviles. Al crear un certificado, se genera un par de claves en el servidor: una pública y una privada. La clave privada se instala en su servidor y es de crucial importancia que nadie más tenga acceso a la misma. La clave pública concordante se instala en el servidor web como parte del certificado digital. Ambas claves, públicas y privadas, se relacionan matemáticamente, pero no son idénticas. [31]

Cuando el cliente envíe una petición al servidor, este responderá con el certificado digital, en ese momento el cliente chequea en su lista de certificados de confianza, una vez comprobada la identidad digital del servidor, se procede a enviar una clave de sesión encriptado con la llave pública del servidor, la cual solo podrá ser descifrada con la llave privada que posee el mismo. En cada petición al HTTPS se generan nuevas claves de sesión y se repite el proceso. Una vez establecida una clave de sesión se inician las comunicaciones seguras entre el cliente y el servidor. [31]

El proceso es perfecto, de modo que al usuario le son invisibles estos pasos.

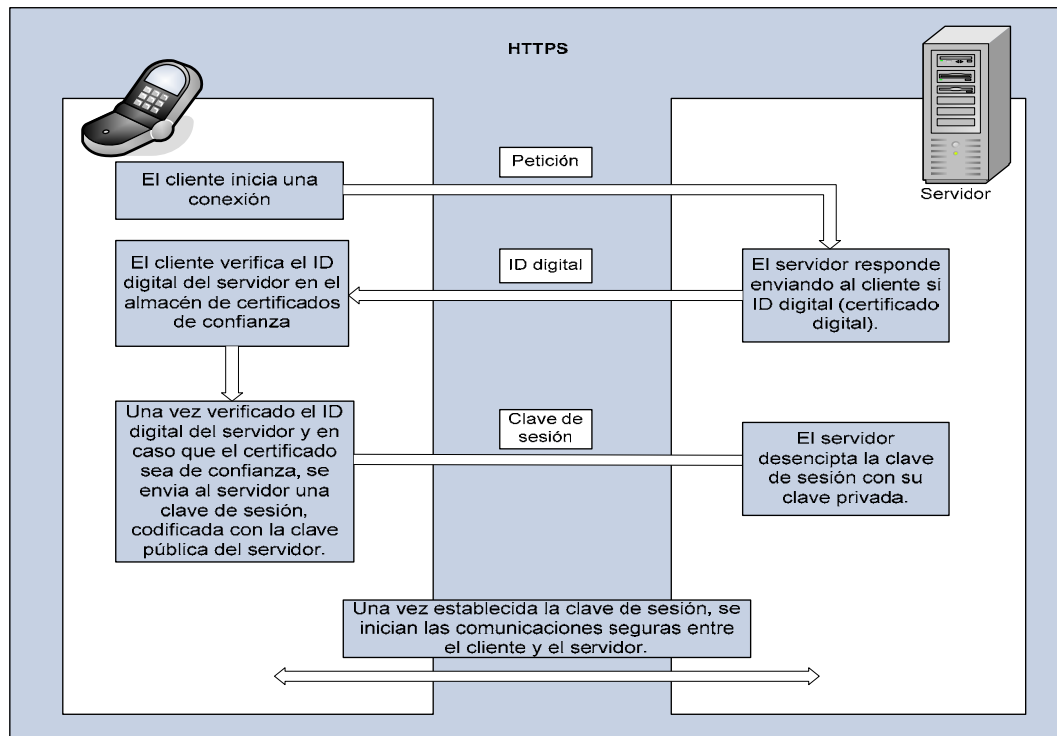


Figura. # 3 Pasos que tienen lugar cuando se establece una sesión SSL.

Otro aspecto importante es el trabajo con las sesiones de usuarios, estas son asignadas en el momento de la autenticación y cuentan con un tiempo de expiración, que está marcado por la inactividad del usuario, este parámetro es configurable. El sistema periódicamente chequea las sesiones y cierra aquellas cuyo tiempo inactivo es igual al de expiración. Con esto se prevee que una aplicación que accidentalmente halla quedado abierta, pueda ser utilizada por personas no autorizadas.

Además se comprueba la identidad del usuario antes de realizar cualquier acción en el servidor. Para evitar que datos sensibles como el usuario y la contraseña viajen con frecuencia por el canal de comunicación, se le otorga al usuario durante su autenticación, un identificador de sesión único compuesto por 28 caracteres, el mismo se genera automáticamente y tiene una vigencia igual al tiempo que se esté conectado, es decir, cada usuario contará con un identificador de sesión diferente cada vez que se autentique, después de esto, solo se comprobará su identidad a través de su identificador de sesión.

### **3.8 Interfaz**

En el desarrollo de las aplicaciones móviles, se decidió el uso de una de interfaz de usuario agradable y sencilla, con un sistema de navegación que oriente al usuario, para ello, en la parte superior de cada interfaz, se especifica el nombre que la identifica, además de un menú en la parte inferior que contiene las opciones válidas para cada una de ellas.

El uso de imágenes no fue tenido en cuenta con el objetivo de aumentar el tiempo de respuesta a las peticiones realizadas. En el caso de la aplicación destinada los dispositivos Windows Mobile, se utilizaron diferentes tonalidades del color azul, siguiendo los principios de diseño del Módulo de Administración Web y aprovechando las bondades que ofrece el entorno de desarrollo en cuanto a diseño, como se muestra en la figura. #4. Las aplicaciones J2ME fueron diseñadas siguiendo los mismos principios, independientemente de esto cuentan con una menor vistosidad, debido a que los componentes brindados por la plataforma están menos trabajados y con un menor grado de configuración, como se muestra en la figura. #5.





Figura. #4 (izquierda-derecha) Pantalla de autenticación, pantalla de menú principal, pantalla de reportes (Windows Mobile).



Figura. #5 (izquierda-derecha) Pantalla de autenticación, pantalla de menú principal, pantalla de reportes (J2ME).

### 3.9 Conclusiones

En el capítulo se confeccionaron los principales artefactos correspondientes al flujo de trabajo de análisis y diseño, haciendo énfasis en los diagramas de clases, así como los diagramas de secuencia y colaboración pertenecientes a cada caso de uso. Se describieron los patrones de diseño usados a lo largo de la aplicación y se llevaron a cabo acciones para mitigar posibles errores en la comunicación con el servidor. Se realizó la descripción de las características y estándares de diseño empleados en el desarrollo de la interfaz de usuario.

## **CAPITULO 4: IMPLEMENTACIÓN Y PRUEBA**

### **4.1 Introducción**

En el presente capítulo se describen los temas referentes a la implementación y prueba del sistema, se hace referencia a las principales tareas que se llevan a cabo en este flujo de trabajo, así como la descripción de los artefactos que se generan en el mismo.

### **4.2 Modelo de Implementación**

El flujo de trabajo de implementación comienza a partir del resultado del diseño y la principal actividad que se lleva a cabo es la implementación del sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. Además se describen cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

El modelo de implementación está conformado por los diagramas de despliegue y componentes, estos describen los componentes a construir y su organización y dependencia entre nodos físicos en los que funcionará el sistema.

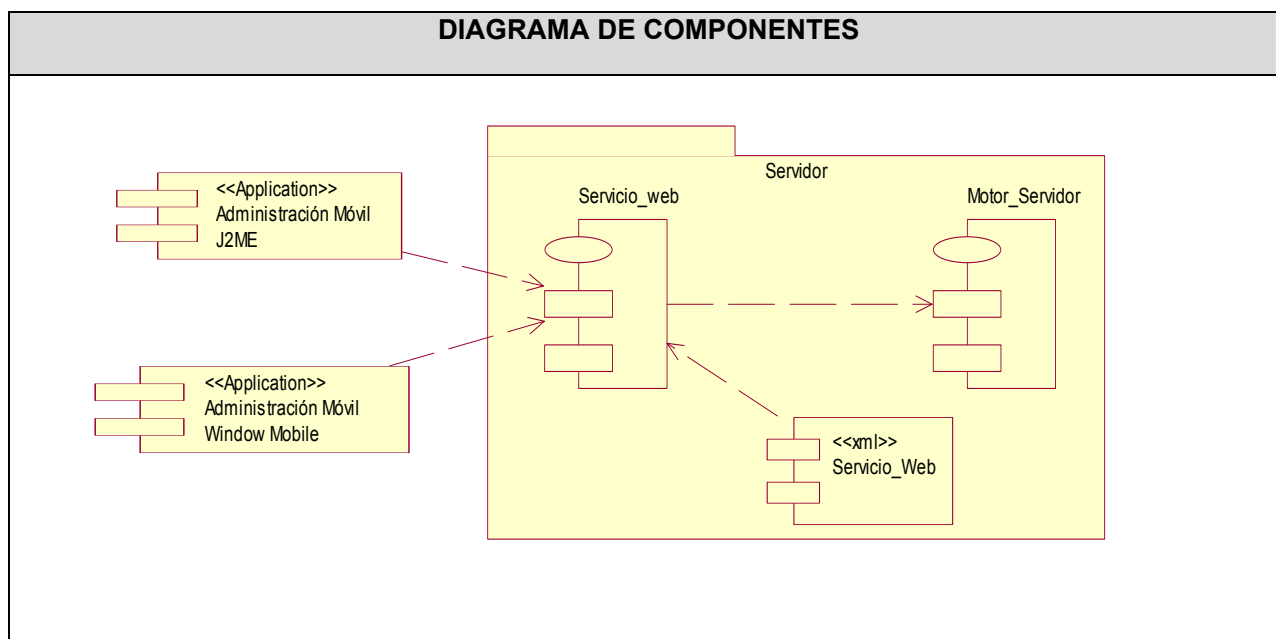
#### **4.2.1 Diagrama de Componentes**

Un diagrama de componentes es un grafo que une componentes a través de relaciones, estas pueden ser de compilación o de ejecución, y además se pueden representar las interfaces de esos componentes.

Son usados para estructurar el modelo de implementación en términos subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. Es otra forma de representar una vista estática del sistema, que representa la organización y dependencia entre los componentes físicos que se necesitan para ejecutar la aplicación, sean éstos componentes de código fuente, librerías, binarios o ejecutables. [28]

En el desarrollo de las soluciones propuestas se crearon una serie de componentes, a continuación se ilustran en el diagrama correspondiente.

Tabla 9. Diagrama de Componente



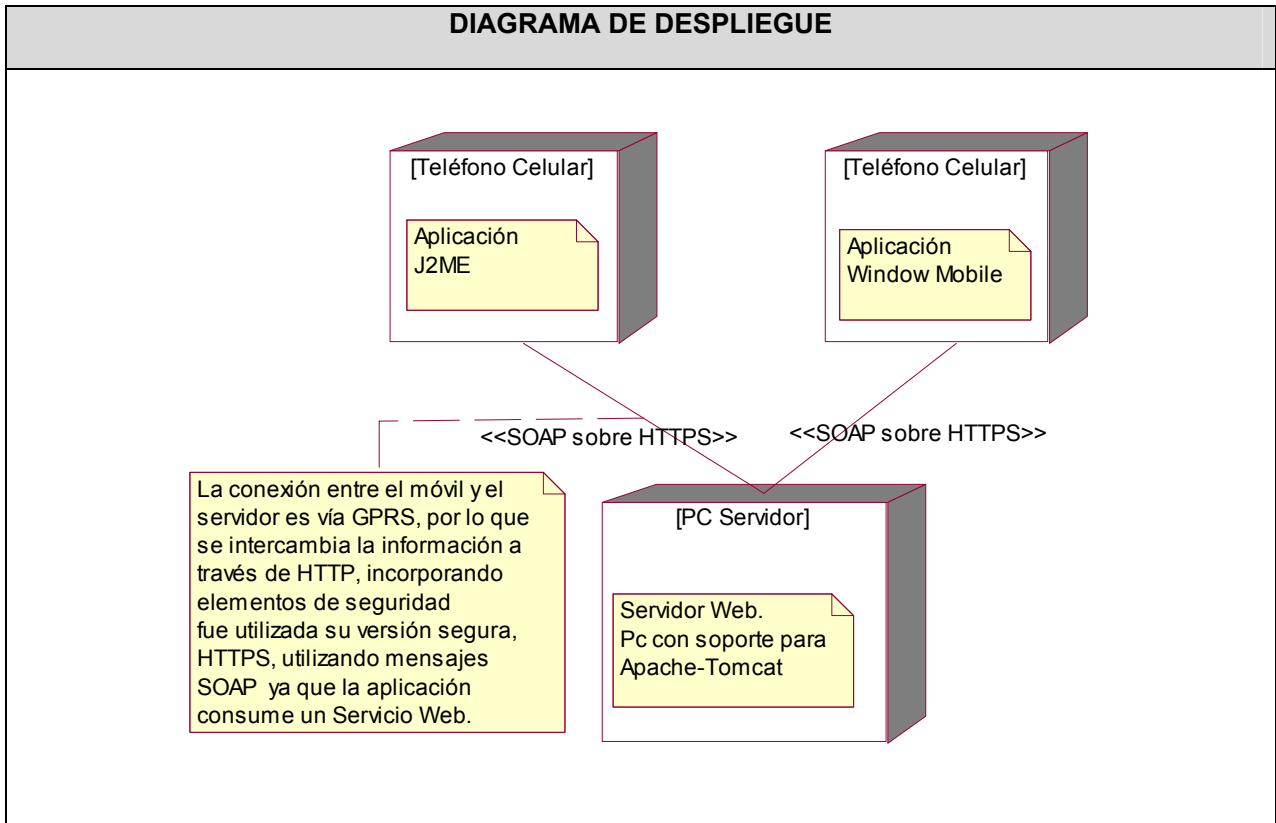
#### 4.2.2 Diagrama de Despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos).

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo se representa como un cubo y es el elemento donde se ejecutan los componentes, representan el despliegue físico de estos. [28]

Teniendo en cuenta las necesidades del sistema antes expuesto, se confeccionó el diagrama de despliegue que se muestra a continuación.

Tabla 10. Diagrama de despliegue



### 4.3 Modelo de pruebas

Uno de los objetivos de la fase de pruebas es verificar que el comportamiento del sistema satisfaga los requisitos establecidos por los clientes y futuros usuarios. A medida que los sistemas sean más complejos y aumente la demanda de calidad, se hacen necesarios procesos y métodos que la garanticen.

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. La prueba de software es un elemento crítico para lograr obtener mejoras en la calidad y representa una revisión final de las especificaciones del diseño y de la codificación.

### **4.3.1 Métodos de Prueba**

Cualquier producto de ingeniería puede ser probado de una de estas formas:

1. Conociendo la funcionalidad específica para la cual fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa.
2. Conociendo el funcionamiento del producto se pueden desarrollar pruebas que aseguren que “todas las piezas encajen”, o sea, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada.

El primer enfoque se denomina Prueba de Caja Negra y el segundo Prueba de Caja Blanca.

#### **Prueba de caja negra**

- Pruebas que se llevan a cabo sobre la interfaz de usuario, se centra principalmente en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software.

#### **Prueba de caja blanca**

- Las pruebas de caja blanca realizan un seguimiento del código fuente según va ejecutando los casos de prueba, de manera que se determinan de manera concreta las instrucciones, bloques, etc. en los que existen errores. Requieren del conocimiento de la estructura interna del programa y son derivadas a partir de las especificaciones internas de diseño o el código.

### **4.3.2 Diseño de Casos de Prueba. Prueba de Caja negra**

Con el objetivo de comprobar el funcionamiento del sistema se realizaron diferentes casos de prueba, a continuación se describen dos de ellos.

**Caso de Uso:** Registrar Usuario.

**Caso de Prueba:** CPR 1 Registrar Usuario.

**Descripción del flujo**

1- Se introducen los datos necesarios para registrar un usuario:

- Nombre de usuario.
- Contraseña del usuario.
- Contraseña del usuario (confirmación).
- Lenguaje.

**Precondiciones:**

- Ejecutar el caso de uso Autenticar Usuario.

**Tabla 11. Caso de uso: Enviar mensaje de texto**

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba
Se introducen los siguientes datos: <b>Nombre de usuario:</b> David <b>Contraseña:</b> David <b>Contraseña(confirmación):</b> David <b>Lenguaje:</b> Es		Se registra el usuario	Satisfactorio

	<p>Se introducen los siguientes datos:</p> <p><b>Nombre de usuario:</b> David</p> <p><b>Contraseña:</b> David</p> <p><b>Contraseña(confirmación):</b> David1s</p> <p><b>Lenguaje:</b> Es</p>	<p>Se dispara error de contraseñas discordantes.</p>	<p>Satisfactorio</p>
	<p>Se introducen los siguientes datos:</p> <p><b>Nombre de usuario:</b> Rigo</p> <p><b>Contraseña:</b> David</p> <p><b>Contraseña(confirmación):</b> David</p> <p><b>Lenguaje:</b> Es</p>	<p>Se dispara error de nombre de usuario en uso.</p>	<p>Satisfactorio</p>
	<p>Se introducen los siguientes datos:</p> <p><b>Nombre de usuario:</b> David</p> <p><b>Contraseña:</b> David</p> <p><b>Contraseña(confirmación):</b> David</p> <p><b>Lenguaje:</b> vacío</p>	<p>Se dispara error de lenguaje inválido.</p>	<p>Satisfactorio</p>

**Caso de Uso:** Autenticar Usuario.

**Caso de Prueba:** CPR 2 Autenticar Usuario.

**Descripción del flujo**

1- Se introducen las credenciales para la autenticación del usuario:

- Nombre de usuario.
- Contraseña del usuario.

**Precondiciones:**

- Usuario previamente registrado en el servidor.

**Tabla 12. Caso de Prueba Autenticar usuario.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba
Se introducen los siguientes datos: <b>Nombre de usuario:</b> David <b>Contraseña:</b> David		Autenticación exitosa, el usuario ha entrado al sistema.	Satisfactorio
	Se introducen los siguientes datos: <b>Nombre de usuario:</b> Pedro <b>Contraseña:</b> David	Se dispara error de usuario desconocido.	Satisfactorio



	Se introducen los siguientes datos: <b>Nombre de usuario:</b> David <b>Contraseña:</b> David1s	Se dispara error de contraseña incorrecta.	Satisfactorio
	Se introducen los siguientes datos: <b>Nombre de usuario:</b> vacío <b>Contraseña:</b> David	Se dispara error de nombre de usuario invalidado.	Satisfactorio
	Se introducen los siguientes datos encontrándose el usuario David autenticado en el sistema : <b>Nombre de usuario:</b> David <b>Contraseña:</b> David	Se dispara error de usuario duplicado.	Satisfactorio

El flujo de trabajo de pruebas absorbe una buena porción de los costes de desarrollo de software. Además, se muestra renuente a un tratamiento matemático o, simplemente, automatizado. Su ejecución se basa en metodología (reglas que se les dan a los encargados de probar) que se va desarrollando con la experiencia. Requiere una buena dosis de mala intención, y provoca difíciles reacciones humanas.

Aunque se han desarrollado miles de herramientas de soporte de este flujo, todas han limitado su éxito a entornos muy concretos, frecuentemente sólo sirviendo para el producto para el que se desarrollaron. Sólo herramientas muy generales como analizadores de complejidad, sistemas de ejecución simbólica y medidores de cobertura han mostrado su utilidad en un marco más amplio. Pero al final sigue siendo imprescindible un artista humano que sepa manejarlas.

La etapa de prueba es tan o más importante que todas las realizadas hasta el momento puesto que en ella se refleja la calidad con que ha sido llevada a cabo la proyección del sistema.

#### **4.4 Conclusiones**

Durante la realización del capítulo se describieron los principales artefactos generados en los flujos de trabajo de implementación y prueba. Se ilustraron los diagramas de despliegue y componentes así como las pruebas realizadas al sistema.

## **CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD**

### **5.1 Introducción.**

En la realización de un proyecto se hace necesaria la planificación y el control del esfuerzo, costo y tiempo que tomará llevarlo a cabo. Con la utilización de métodos de estimación de costos, se puede determinar una aproximación de los recursos necesarios, así como el total de tiempo que gastaría una persona o un equipo, en el desarrollo de un producto de software específico. Además se determina la viabilidad económica, ambiental, técnica y de mercado.

### **5.2 Estimación por Puntos de Casos de Uso**

Existen varios métodos para calcular la factibilidad de un proyecto: COCOMO, COCOMO II, Análisis de Punto de función, Análisis de Puntos de Casos de Uso, entre otros. El método de Puntos de Caso de Uso se definió como extensión del Punto de Función, en este trabajo se decidió usarlo de acuerdo a las siguientes consideraciones:

- La alternativa es simple y similar a los Puntos de Función.
- Las otras alternativas no demostraron tener aceptación general ni se han utilizado ampliamente.
- No existen ejemplos de distintas fuentes respecto al uso de otros métodos.
- Algunas herramientas presumen usarlas exclusivamente para la tarea de cálculo.
- El método de Puntos de Casos de Uso es un método de estimación y cálculo de tamaño del software basado en cuentas hechas sobre los casos de uso para un sistema de software.

El método exige la existencia de un modelo de casos de uso, por lo que la labor deberá ser hecha cuando exista algún entendimiento del dominio del problema o cuando se esté realizando las labores de arquitectura y dimensionamiento del tamaño del sistema. Por lo general, estas condiciones están dadas al término de las actividades de análisis. [32]

En términos simples, el método requiere de casos de uso en modo textual y gráfico sólo en términos de mayor claridad y se realizan los siguientes pasos:

### 5.3 Planificación basada en Casos de Uso

**Paso 1.** Cálculo de Puntos de Casos de Uso sin ajustar.

Variables a utilizar:

**UUCP:** Puntos de Casos de Uso sin ajustar

**UAW:** Factor de Peso de los Actores sin ajustar

**UUCW:** Factor de Peso de los Casos de Uso sin ajustar

Se calcula a partir de la siguiente ecuación:

$$\mathbf{UUCP = UAW + UUCW}$$

#### 1.1 Factor de Peso de los Actores sin ajustar (UAW)

Este valor se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema. Los criterios se muestran en la siguiente tabla:

**Tabla 13. Factor de Peso de los Actores sin ajustar.**

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	1	1
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz	3	1	3

	gráfica.			
Total			1	4

$$UAW = \sum \text{cantActores} * \text{peso}$$

$$UAW = 4$$

### 1.2 Factor de Peso de los Casos de Uso sin ajustar (UUCW)

Este valor se calcula mediante un análisis de la cantidad de casos de uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los casos de uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia. Los criterios se muestran en la siguiente tabla:

**Tabla 14. Factor de Peso de los Casos de Uso sin ajustar.**

Tipo de CU	Descripción	Factor de peso	Cantidad CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	12	60
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	0	0
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0
Total			12	60

$$UUCW = \sum \text{cantCU} * \text{peso}$$

$$UUCW = 60$$

$$UUCP = UAW + UUCW$$

$$UUCP = 4+60$$

**UUCP = 64**

**Paso 2.** Cálculo de los Puntos de Casos de Uso ajustados.

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

Variables a utilizar:

**UCP:** Puntos de Casos de Uso ajustados

**UUCP:** Puntos de Casos de Uso sin ajustar

**TCF:** Factor de complejidad técnica

**EF:** Factor de ambiente

Se calcula a partir de la siguiente ecuación:

$$\mathbf{UCP = UUCP \times TCF \times EF}$$

### 2.1 Factor de complejidad técnica (TCF)

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores:

**Tabla 15. Factor de complejidad técnica.**

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema Distribuido	2	5	10
T2	Tiempo de Respuesta	1	3	3
T3	Eficiencia del Usuario Final	1	4	4
T4	Funcionamiento Interno Complejo	1	4	4
T5	El código debe ser reutilizable	1	5	5

T6	Facilidad de instalación	0,5	4	2
T7	Facilidad de uso	0,5	5	2.5
T8	Portabilidad	2	4	8
T9	Facilidad de cambio	1	4	4
T10	Concurrencia	1	3	3
T11	Incluye objetivos especiales de seguridad	1	4	4
T12	Provee acceso directo a terceras partes	1	4	4
T13	Se requieren facilidades especiales de entrenamientos de usuario	1	1	1
Total				54.5

$$TCF = 0.6 + 0.01 * \Sigma(\text{peso} * \text{valor asignado})$$

$$TCF = 0.6 + 0.01 * 54.5$$

$$TCF = 1.145$$

## 2.2 Factor de ambiente (EF)

El factor de ambiente (EF) está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema. Cada factor se cuantifica con un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

**Tabla 16. Factor de ambiente.**

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1.5	3	4.5

E2	Experiencia en la aplicación	0.5	3	1.5
E3	Experiencia en la Orientación a Objetos	1	5	5
E4	Capacidad del analista líder	0.5	5	2.5
E5	Motivación	1	5	5
E6	Estabilidad de requerimientos	2	4	8
E7	Personal Part-Time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	2	-2
Total				24.5

$$EF = 1.4 - 0.03 * \Sigma(\text{peso} * \text{valor asignado})$$

$$EF = 1.4 - 0.03 * 24.5$$

$$EF = 0.665$$

$$UCP = UUCP * TCF * EF$$

$$UCP = 64 * 1.145 * 0.665$$

$$UCP = 48.7312$$

**Paso 3.** Estimación de esfuerzo a través de los Puntos de Casos de Uso

Variables a utilizar:

**E:** Esfuerzo estimado en horas hombres.

**UCP:** Punto de casos de usos ajustados.

**CF:** Factor de conversión.



Se calcula a partir de la siguiente ecuación:

$$E = UCP * CF$$

Para obtener el factor de conversión (CF) se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por encima de la media para los restantes (E7, E8). Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de Uso. Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de Uso. Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso es demasiado alto.

En este caso:

$$CF = 20 \text{ Horas-Hombre / Punto de Casos de uso.}$$

$$E = 48.7312 * 20$$

$$E = 974.624$$

**Paso 4.** Calcular esfuerzo de todo el proyecto

**Tabla 17. Estimación del esfuerzo total del proyecto.**

Actividad	Pocentaje %	Horas – Hombres
Análisis	20	487.312
Diseño	20	487.312
Implementación	40	974.624
Pruebas	10	243.656
Sobrecarga(otras actividades)	10	243.656
Total	100	2436.56

Si ET = 2436.56 horas-hombre y se estima que cada mes tiene como promedio 192 horas laborables, eso daría un ET = 12.6904167 mes-hombre.

Esto quiere decir que 1 persona puede realizar el problema analizado en 12 meses y medio aproximadamente.

### **Costo del Proyecto**

Se asume como salario promedio mensual \$100.00.

CH: Cantidad de hombres.

Tiempo: Tiempo total del proyecto.

CH = 2 hombres

CHM = 2 \* Salario Promedio

CHM = 200.00 \$/mes

Costo = CHM \* ET / CH

Costo = 200.00 \* 12.6904167 / 2

Costo = \$ 1269.04167

Tiempo = ET / CH

Tiempo = 12.6904167 / 2

Tiempo = 6.34520833 ≈ 6 meses

De los resultados obtenidos se interpreta que con 2 hombres trabajando en el proyecto el mismo se desarrolla en 6 meses y su costo total se estima que sea \$1269 aproximadamente.

### **5.4 Beneficios tangibles e intangibles**

El Módulo de Administración Móvil, SIRIO Mobile Admin, perteneciente a la plataforma SIRIO Mobile, es un producto con un fin comercial, su principal objetivo es facilitar la administración del servidor a través de un teléfono móvil. Con este nuevo módulo, se incluye la posibilidad de un constante monitoreo al estado del servidor. Además se incluye el uso de Servicios Web, garantizando así, la posibilidad de la creación de nuevas aplicaciones orientadas a distintos dispositivos.

Se utiliza el protocolo HTTPS en el intercambio de información, garantizando la confidencialidad de datos sensibles.

El sistema propone dos aplicaciones que poseen las mismas funcionalidades, una de ellas orientada a los teléfonos que soporten Java y la otra a los que presenten Windows Mobile como sistema operativo, logrando abarcar una amplia gama de teléfonos.

Por tanto, los beneficios inmediatos son generalmente intangibles:

- Aumento de la experiencia y conocimientos del equipo de desarrollo.
- Eliminación de la dependencia de una computadora para hacer uso de las funciones de administración que brinda la plataforma.
- Aumenta la conectividad y el control, debido a que el teléfono móvil podrá establecer comunicación con el servidor siempre que posea cobertura y el servicio de GPRS sea brindado por el operador.
- Incorporación de seguridad en la transferencia de información, aumentando la confidencialidad.
- Aumento de la flexibilidad para incluir nuevas aplicaciones orientadas a diferentes dispositivos, debido al uso de Servicios Web.
- Incrementa las funcionalidades de la plataforma SIRIO Mobile, sirviendo así como mejora y punto atrayente para la inserción en el mercado de la misma.

### **5.5 Análisis de costos y beneficios**

El desarrollo de un producto de software, lleva consigo el consumo de diferentes recursos. La puesta en marcha de proyectos casi siempre esta condicionada por los beneficios que reportaría su implantación y utilización. El sistema que se propone, es parte de la plataforma SIRIO Mobile perteneciente a la entidad Procyon, esta posee fines comerciales y por lo tanto su mayor beneficio es del orden económico.

Una vez implantado, el sistema contribuirá a aumentar la eficiencia en el monitoreo del estado del servidor, permitiendo al administrador, con su teléfono móvil, desde casi cualquier lugar y a cualquier hora, hacer uso de las funciones que el módulo SIRIO Mobile Admin brinda.

Plataformas existentes en el mundo, similares a SIRIO, como es el caso de Unkasoft, presentan un valor comercial que oscila entre 6 mil y 18 mil euros, en dependencia de las funcionalidades que se soliciten. [33]

El costo total estimado de la plataforma SIRIO sumado al del Módulo de Administración Móvil es de \$2527.24.

Analizando el costo del proyecto, los numerosos beneficios que reporta detallados con anterioridad y comparando el valor comercial de productos similares con el costo estimado de SIRIO, se puede concluir que su implementación es realmente factible.

## **5.6 Conclusiones**

En este capítulo se describió el estudio de factibilidad realizado correspondiente al sistema propuesto, teniendo en cuenta el costo estimado y los beneficios que reportará al ser implantado.

El sistema propuesto reportará beneficios significativos e importantes para la administración de la plataforma SIRIO Mobile, al contribuir a incrementar la capacidad de uso del mismo, lo que indica que es factible su implementación.

## **CONCLUSIONES**

Al finalizar la investigación se cumple con el objetivo planteado de definir una solución para un nuevo módulo de administración de la plataforma SIRIO, que incorpore elementos claves como la ubicuidad, la interoperabilidad con diferentes aplicaciones y la seguridad en la transferencia de información.

Para su cumplimiento se desarrolló un Servicio Web que fue incorporado en el servidor de la plataforma, este brinda las funcionalidades necesarias para su administración y garantiza flexibilidad a la hora de realizar nuevas aplicaciones en diferentes plataformas de desarrollo o dispositivos. Fueron implementadas además, dos aplicaciones que consumirán este servicio, una destinada a los teléfonos celulares que soporten J2ME y otra a los que presenten Windows Mobile como sistema operativo, de esta forma se aprovechan las capacidades de movilidad y uso de los móviles. Se llevaron a cabo mecanismos para asegurar la confidencialidad en el envío de información con el uso del Protocolo Seguro de Transferencia de Hipertexto (HTTPS).

Para la arribar a una solución que cumpliera con el objetivo se realizó un análisis de la Telefonía Móvil y las Comunicaciones Inalámbricas, se investigó sobre el uso de los teléfonos celulares en entornos empresariales para la administración de procesos, ubicando a SIRIO Mobile Admin en las tendencias más actuales. También se hizo un análisis de la seguridad en este tipo de aplicaciones y particularmente el uso de HTTPS en los dispositivos móviles de poca capacidad de procesamiento. Se empleó RUP como metodología para organizar el proceso de desarrollo del software. Fueron realizadas las pruebas suficientes para hacer entrega a la entidad Procyon de un producto confiable y robusto.

Toda la documentación del módulo de administración SIRIO Mobile Admin está acompañada de una bibliografía actual y variada, se hizo uso de varias publicaciones confiables de internet y de otras en formato duro.

Como resultado de los estudios realizados y el desarrollo de la solución expuesta se puede concluir que fueron cumplidos cabalmente los objetivos plateados y se recomiendan para futuras versiones algunos aspectos que mejorarán e incrementarán las funcionalidades de administración en la Plataforma SIRIO Mobile.

## **RECOMENDACIONES**

- Establecimiento de un sistema de avisos automatizados desde el servidor al Módulo de Administración Móvil, que alerte sobre posibles acciones que pongan en peligro el buen funcionamiento de la plataforma.
- Adicionar la posibilidad de no permitir nuevas conexiones en el servidor o eliminar las ya existentes.
- Se recomienda además la adquisición de un certificado digital firmado por una Entidad Certificadora reconocida nivel internacional para hacer más eficiente el uso de HTTPS.

## BIBLIOGRAFÍA

1. **Pressman, Roger S.** *Ingeniería de software.un enfoque práctico.* 2005.
2. **JACOBSON, IVAR, BOOCH, GRADY y RUMBAUGH, JAMES.** *EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE.* 2002.
3. **Larman, Craig.** *UML y Patrones.* s.l. : Prentice Hall, 2002.
4. **Gamma, Erich, y otros.** *Design Patterns.* 1994.
5. **Thawte.** Transferencia Segura de Datos en Línea con SSL. [En línea] 2008. [https://www.thawte.com/ssl-digital-certificates/free-guides-whitepapers/pdf/ssl\\_es.pdf](https://www.thawte.com/ssl-digital-certificates/free-guides-whitepapers/pdf/ssl_es.pdf).
6. **Stevei, Graham, y otros.** *Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI.* 2004. pág. 816.
7. **Bequet, Henry, y otros.** *Beginning Java Web Services.* s.l. : Wrox Press , 2002. pág. 411.
8. **Badani Hernández, Sigifredo E.** UCENTRAL. [En línea] septiembre de 2002. [http://eiec.ucentral.cl/ftp/material/apuntes/iec61/Diseno/Metodo\\_PCU.doc](http://eiec.ucentral.cl/ftp/material/apuntes/iec61/Diseno/Metodo_PCU.doc).
9. **Microsoft Corporation.** MSDN. [En línea] 2007. [http://msdn.microsoft.com/es-es/library/ms172489\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/ms172489(VS.80).aspx).
10. **Chappell, David A y Jewell, Tyler.** *Java Web Services.* s.l. : O'Reilly Media, 2002. pág. 276.
11. **Englander, Robert.** *Java and SOAP.* [ed.] O'Reilly. 2002. pág. 276. 0-596-00175-4.
12. **José, Valle.** Definición arquitectura cliente servidor. *Monografias.com.* [En línea] 2005. <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml?monosearch>.
13. **Letelier, Patricio y Penadés, M<sup>a</sup> Carmen.** Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [En línea] <http://www.willydev.net/descargas/masyxp.pdf>.
14. **Martínez Álvarez, Francisco.** *DESARROLLO DE APLICACIONES JAVA PARA MÓVILES: J2ME Y HERRAMIENTAS DE DESARROLLO.* Sevilla : s.n., 2006.
15. **Méndez, Justo.** Monografias.com. [En línea] 2004. <http://www.monografias.com/trabajos/lengprog/lengprog.shtml>.
16. **Microsoft Corporation.** MDSN. [En línea] 2007. [http://msdn.microsoft.com/es-es/library/ms172489\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/ms172489(VS.80).aspx).
17. **González, Benjamín.** DesarrolloWeb.com. [En línea] <http://www.desarrolloweb.com/articulos/1557.php>.

18. *Ayuda de la herramienta Case Rational Rose.*
19. **Rodríguez García, Yoanis.** Monografías.com. [En línea] 2007.  
<http://www.monografias.com/trabajos44/telefonía-movil-cuba/telefonía-movil-cuba.shtml>.
20. **Mendívil, Ignacio.** *El abc de los documentos electrónicos seguros.* 1997.
21. **Sánchez Wevar, Juan Andrés.** *Análisis y Estudio de Redes GPRS.* 2005.

## Referencias Bibliográficas

1. **Observatorio Europeo de Tecnologías de la Comunicación.** n-economía. [En línea] 2006. [http://www.n-economía.com/fichas\\_neconomía/pdf/gr2/2\\_6.pdf](http://www.n-economía.com/fichas_neconomía/pdf/gr2/2_6.pdf).
2. **Einarson, Jeff.** Administración y aprovisionamiento de los dispositivos móviles de la empresa. [En línea] 2007.  
<http://www.intel.com/espanol/technology/magazine/archive/2004/jun/mo06041.pdf>.
3. **Buedo Idalgo, Dennys, Vieira, Yunisel.** *Plataforma Sirio Mobile.* 2007.
4. **Ramos Pascual, Francisco.** Radiocomunicaciones y fibra óptica. [En línea]  
[http://www.radióptica.com/Radio/telefonía\\_movil.asp](http://www.radióptica.com/Radio/telefonía_movil.asp).
5. **Velázquez Pastrana, Lourdes.** Enter@te en línea. [En línea] 2002.  
<http://www.enterate.unam.mx/Articulos/2002/agosto/telefccl.htm>.
6. **Berbel Navarro, Genís.** UOC. [En línea] 2001. <http://www.uoc.edu/web/esp/art/uoc/0105021/berbel.html>.
7. **WebMovilGSM.** webmovilgsm. [En línea] <http://www.webmovilgsm.com/umts.htm>.
8. *Revista Digital de las Tecnologías de la Información y las Comunicaciones.* **Cutiño, Ruth Yurina Vega.** s.l. : DEPARTAMENTO DE TELEMÁTICA DEL INSTITUTO SUPERIOR POLITÉCNICO JOSÉ ANTONIO ECHEVERRÍA., 2003, Vol. 40.
9. **ConocimientosWeb.** conocimientosweb.net. [En línea] 2007.  
<http://www.conocimientosweb.net/portal/html.php?file=cursos/redes/Conectividad.htm>.
10. **Móvil10.** MelodiasMóviles.com. [En línea] 2006. <http://www.melodiasmoviles.com/documentacion/intro-gprs.php>.
11. **ExitGames.** ExitGames. [En línea] 2006. <http://www.exitgames.com/>.
12. **BV Ex Machina.** Ex Machina. [En línea] 2005. <http://www.exmachina.nl/sp.html>.



13. **Gamefederation.** Gamefederation. [En línea] 2005. [Citado el: 5 de 11 de 2007.] <http://www.gamefederation.com/?pageID=14>.
14. **Pressman, Roger S.** *Ingeniería de software un enfoque práctico*. 2005.
15. **Villalobos.** Ciberaula. [En línea] 2006. [http://java.ciberaula.com/articulo/introduccion\\_j2me/](http://java.ciberaula.com/articulo/introduccion_j2me/).
16. **Armstrong, Eric y Ball, Jennifer.** java.sun.com. [En línea] 2005. <http://java.sun.com/j2ee/1.4/docs/tutorial/doc/>.
17. **Corporation Microsoft.** msdn. [En línea] 2007. [http://msdn.microsoft.com/es-es/library/f44bbwa1\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/f44bbwa1(VS.80).aspx).
18. **Aragon, Adrian.** Devjoker. [En línea] 2007. [Citado el: 10 de enero de 2008.] [http://www.devjoker.com/asp/ver\\_contenidos.aspx?co\\_contenido=125](http://www.devjoker.com/asp/ver_contenidos.aspx?co_contenido=125).
19. **Foundation Apache Software.** tomcat.apache.org. [En línea] 2007. [Citado el: 15 de 12 de 2007.] <http://tomcat.apache.org/>.
20. **Bailey, Thomas.** *Getting started with web services*. s.l. : Sony Ericsson, 2006. EN/LZT 108 8985 R1B.
21. **González, Benjamín.** DesarrolloWeb. [En línea] 2004. <http://www.desarrolloweb.com/articulos/1557.php>.
22. **Microsoft Corporation.** MicrosoftTechNet. [En línea] 06 de 04 de 2007. [Citado el: 23 de febrero de 2008.] [http://technet.microsoft.com/es-es/library/bb430767\(EXCHG.80\).aspx](http://technet.microsoft.com/es-es/library/bb430767(EXCHG.80).aspx).
23. **IBM.** IBM. [En línea] 2005. <http://publib.boulder.ibm.com/html/as400/v5r1/ic2931/index.htm?info/rzaha/keytool.htm>.
24. **Foundation Eclipse.** Eclipse. [En línea] 2007. <http://www.eclipse.org/>.
25. **Microsoft.com.** MSDN. [En línea] 2005. <http://www.microsoft.com/spanish/msdn/vs2005/editions/pro/default.mspix>.
26. **Tigris.org.** Tigris.org. [En línea] 2006. <http://subversion.tigris.org/>.
27. —. Tigris.org. [En línea] <http://tortoisesvn.tigris.org/>.
28. **JACOBSON, IVAR, BOOCH, GRADY y RUMBAUGH, JAMES.** *EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE*. 2002.
29. **Larman, Craig.** *UML y Patrones*. s.l. : Prentice Hall, 2002.
30. **Gamma, Erich, y otros.** *Design Patterns*. 1994.
31. **Thawte.** Transferencia Segura de Datos en Línea con SSL. [En línea] 2008. [https://www.thawte.com/ssl-digital-certificates/free-guides-whitepapers/pdf/ssl\\_es.pdf](https://www.thawte.com/ssl-digital-certificates/free-guides-whitepapers/pdf/ssl_es.pdf).

32. **Badani Hernández, Sigifredo E.** UCENTRAL. [En línea] septiembre de 2002. [http://eiec.ucentral.cl/ftp/material/apuntes/iec61/Diseno/Metodo\\_PCU.doc](http://eiec.ucentral.cl/ftp/material/apuntes/iec61/Diseno/Metodo_PCU.doc).
33. **Unkasoft.** Unkasoft. [En línea] 2007. <http://www.unkasoft.com/es>.

## ANEXO

### Anexo 1: Descripción detallada de los Casos de Uso.

Tabla 18. Descripción textual del Caso de Uso: Autenticar Usuario.

<b>Nombre del Caso de Uso</b>		Autenticar Usuario.
<b>Actores</b>	Administrador	
<b>Propósito</b>	Controlar el acceso al servidor.	
<b>Resumen</b>	El caso de uso se inicia cuando un administrador previamente registrado en el servidor intenta acceder al sistema, el mismo responde con una orden de autenticación, el administrador introduce sus credenciales y las envía al sistema el cual se encarga de procesarlas.	
<b>Referencias</b>	RF1.1, RF4.1	
<b>Precondiciones</b>	Poseer una cuenta en el servidor con privilegios de administración.	
<b>Poscondiciones</b>	El usuario es introducido en el sistema, listo para realizar cualquier acción administrativa.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1-El administrador ejecuta la aplicación móvil.	1.1- El sistema devuelve una petición de autenticación.	
2-El administrador introduce sus credenciales (usuario y contraseña) y la envía al sistema.	2.1- El sistema verifica la validez de los datos recibidos.	
	2.2- El sistema verifica el estado del canal de administración, en caso de que no exista, se crea y se adiciona el usuario a dicho canal, de lo contrario se adiciona directamente.	
	2.3- El sistema permite la entrada del administrador al sistema.	

	2.4- El sistema crea una sesión y se la asigna al usuario.
<b>Flujo Alternativo:</b>	
Acción 2.1:	2.1a Si los datos no son correctos el sistema muestra un mensaje de error y solicita que los datos sean llenados nuevamente.
<b>Prioridad:</b>	Primaria

**Tabla 19. Descripción textual del Caso de Uso: Listar Estadísticas de Juego.**

<b>Nombre del Caso de Uso</b>	Listar Estadísticas de Juego.
<b>Actores</b>	Administrador
<b>Propósito</b>	Listar las estadísticas de un juego que se halla establecido en el servidor.
<b>Resumen</b>	El caso de uso se inicia cuando el actor solicita ver las estadísticas de un juego determinado.
<b>Referencias</b>	RF.3.6, RF.4.2, RF.4.3
<b>Precondiciones</b>	Actor previamente autenticado en el servidor.
<b>Poscondiciones</b>	Lista de estadísticas.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1- El administrador solicita al sistema las estadísticas de un juego determinado.	1.1- El sistema verifica que el actor posea una sesión válida, de ser positivo disminuye tiempo de inactividad del usuario.
	1.2- El sistema recibe la petición y muestra el listado de todos los canales que se encuentran registrados en el servidor y tienen al menos un juego terminado.

2- El actor selecciona el canal del juego donde solicita las estadísticas.	2.1- El sistema muestra el listado con las de partidas establecidas, el usuario ganador y perdedor para cada una de ellas.
<b>Flujo Alternativo:</b>	
Acción 1.1:	1.1a Si la sesión del usuario ha expirado, se llama al caso de uso: Autenticar Usuario.
<b>Prioridad:</b>	Primaria

**Tabla 20. Descripción textual del Caso de Uso: Enviar Mensaje de Texto**

<b>Nombre del Caso de Uso</b>		Enviar Mensaje de Texto
<b>Actores</b>	Administrador	
<b>Propósito</b>	Enviar mensajes a usuarios conectados.	
<b>Resumen</b>	El caso de uso se inicia cuando un administrador autenticado decide enviar un mensaje de texto algún usuario que se encuentre autenticado en el servidor.	
<b>Referencias</b>	RF. 2, RF.4.2, RF.4.3	
<b>Precondiciones</b>	El usuario debe estar previamente autenticado.	
<b>Poscondiciones</b>	Mensaje enviado.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>		<b>Respuesta del Sistema</b>
1- El administrador solicita al servidor enviar mensaje de texto.		1.1- El sistema muestra tres opciones: Mensaje Privado, Mensaje Público a Usuarios en un Canal y Mensaje Público a Todos los Usuarios del Servidor.
<b>Escenario 1: Mensaje Privado</b>		
1- El actor elige la opción de Mensaje Privado.		1.1- El sistema verifica que el actor posea una sesión válida, de ser positivo disminuye tiempo de inactividad del usuario.

	1.2- El sistema recibe la petición y muestra un listado de los canales que se encuentran activos.
2- El actor selecciona el canal donde se encuentra el usuario al cual necesita enviarle un mensaje.	2.1- El sistema muestra el listado de los usuarios en línea en el canal seleccionado.
3- El actor escoge el usuario.	3.1- El sistema muestra la interfaz para introducir el texto del mensaje.
4- El actor introduce mensaje y e inicia la acción de enviarlo.	4.1- El sistema procede a enviar el mensaje al usuario seleccionado.
<b>Flujo Alternativo:</b>	
Acción 1.1:	1.1a Si la sesión del usuario ha expirado, se llama al caso de uso: Autenticar Usuario.
<b>Escenario 2: Mensaje Público a Usuarios en un Canal</b>	
1- El actor elige la opción de mensaje público a un canal.	1.1- El sistema verifica que el actor posea una sesión válida, de ser positivo disminuye tiempo de inactividad del usuario.
	1.2- El sistema recibe la petición y muestra un listado de los canales que se encuentran activos.
2- El actor escoge el canal deseado.	2.1- El sistema muestra la interfaz para introducir el texto del mensaje.
3- El actor introduce mensaje y e inicia la acción de enviarlo.	3.1- El sistema procede a enviar el mensaje al usuario seleccionado.
<b>Flujo Alternativo:</b>	
Acción 1.1:	1.1a Si la sesión del usuario ha expirado, se llama al caso de uso: Autenticar Usuario.
<b>Escenario 3: Mensaje Público a Todos los Usuarios Conectados</b>	

1- El actor elige la opción de mensaje público todos los usuarios.	1.1- El sistema verifica que el actor posea una sesión válida, de ser positivo disminuye tiempo de inactividad del usuario.
	1.2- El sistema muestra la interfaz para introducir el texto del mensaje.
2- El actor introduce mensaje y e inicia la acción de enviarlo.	2.1- El sistema procede a enviar el mensaje al usuario seleccionado.
<b>Flujo Alternativo:</b>	
Acción 1.1:	1.1a Si la sesión del usuario ha expirado, se llama al caso de uso: Autenticar Usuario.
<b>Prioridad:</b>	Secundaria

**Tabla 21. Descripción textual del Caso de Uso: Listar Juegos Iniciados.**

<b>Nombre del Caso de Uso</b>		Listar Juegos Iniciados.
<b>Actores</b>	Administrador	
<b>Propósito</b>	Listar todos los juegos que han sido iniciados.	
<b>Resumen</b>	El caso de uso se inicia cuando el actor decide listar todos los juegos que se encuentran iniciados, en un canal determinado, el sistema recibe la petición muestra el listado.	
<b>Referencias</b>	RF.3.5, RF.4.2, RF.4.3	
<b>Precondiciones</b>	Actor previamente autenticado en el servidor.	
<b>Poscondiciones</b>	Lista de juegos iniciados.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1- El actor solicita al sistema un listado de los juegos que se encuentran iniciados en un canal dado.	1.1- El sistema verifica que el actor posea una sesión válida, de ser positivo disminuye tiempo de inactividad del usuario.	

	1.2- El sistema recibe la petición y muestra el listado de todos los canales que se encuentran activos en el momento de la petición.
2- El actor selecciona el canal deseado.	2.1- El sistema muestra todos los juegos iniciados en ese canal.
<b>Flujo Alternativo:</b>	
Acción 1.1:	1.1a Si la sesión del usuario ha expirado, se llama al caso de uso: Autenticar Usuario.
<b>Prioridad:</b>	Primaria

**Tabla 22. Descripción textual del Caso de Uso: Listar Usuarios Conectados.**

<b>Nombre del Caso de Uso</b>		Listar Usuarios Conectados.
<b>Actores</b>	Administrador	
<b>Propósito</b>	Listar los usuarios que se encuentran conectados en un canal de juego determinado.	
<b>Resumen</b>	El caso de uso se inicia cuando el actor solicita ver los usuarios que se encuentran conectados a un canal en un momento dado.	
<b>Referencias</b>	RF.3.1, RF.4.2, RF.4.3	
<b>Precondiciones</b>	Actor previamente autenticado en el servidor.	
<b>Poscondiciones</b>	Lista de usuarios conectados.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>		<b>Respuesta del Sistema</b>
1- El actor solicita al sistema un listado de los usuarios conectados en un canal de juego determinado.		1.1- El sistema verifica que el actor posea una sesión válida, de ser positivo disminuye tiempo de inactividad del usuario.



	1.2- El sistema recibe la petición y muestra el listado de todos los canales que se encuentran activos en el momento de la petición.
2- El actor selecciona el canal deseado.	2.1- El sistema muestra todos los usuarios conectados en ese canal.
<b>Flujo Alternativo:</b>	
Acción 1.1:	1.1a Si la sesión del usuario ha expirado, se llama al caso de uso: Autenticar Usuario.
<b>Prioridad:</b>	Primaria

**Anexo 2: Diagramas de Clases del Análisis.**

**Tabla 23. Diagrama de Clases del Análisis CU: Autenticar Usuario**

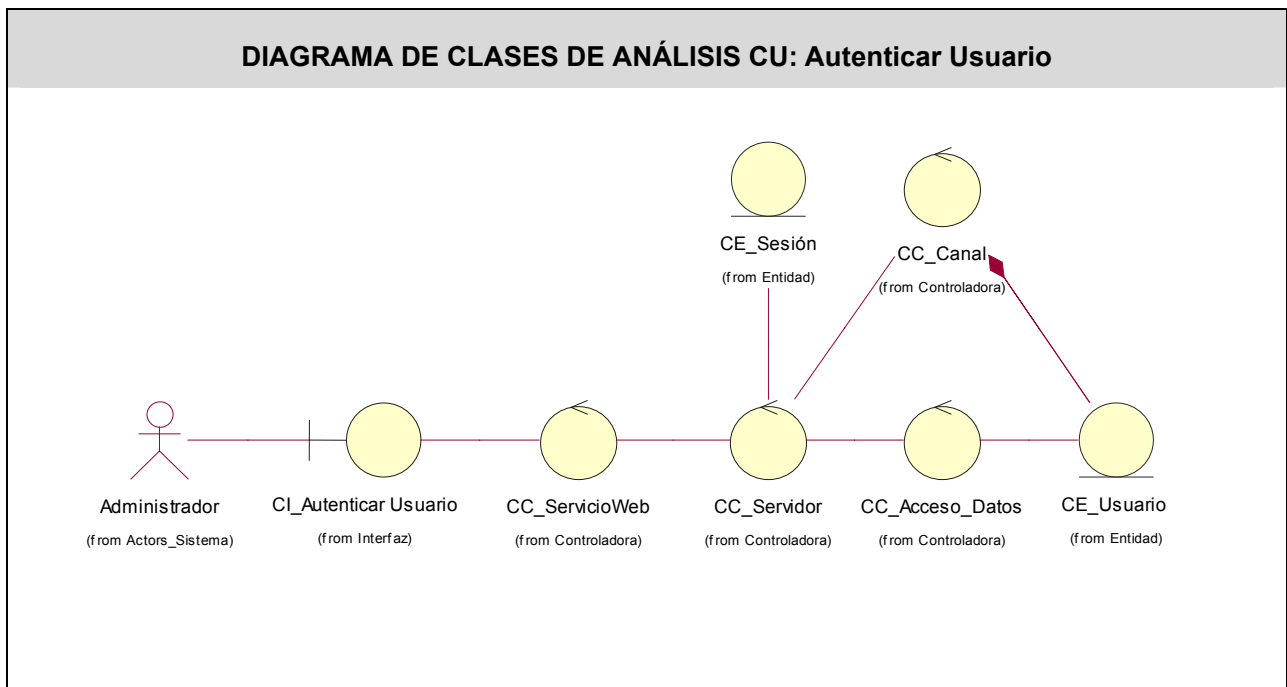


Tabla 24. Diagrama de Clases del Análisis CU: Registrar Usuario.

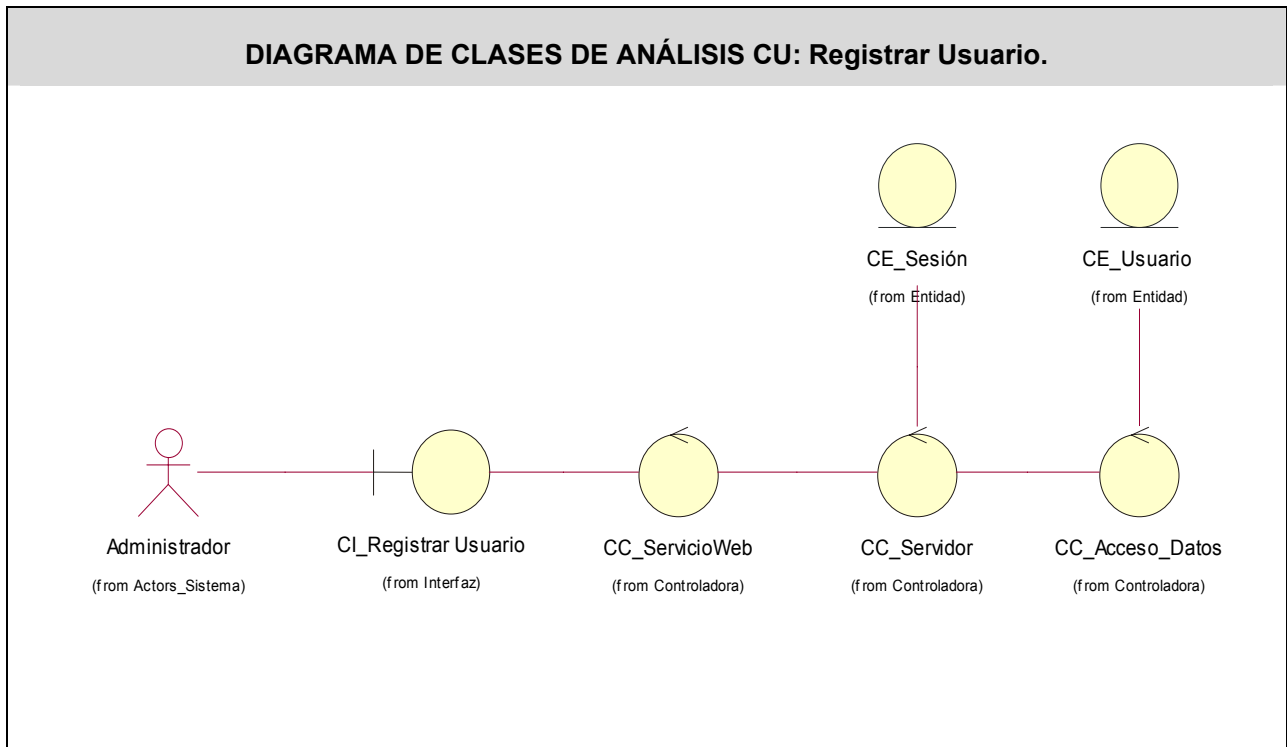


Tabla 25. Diagrama de Clases del Análisis CU: Listar Estadísticas de Juego.

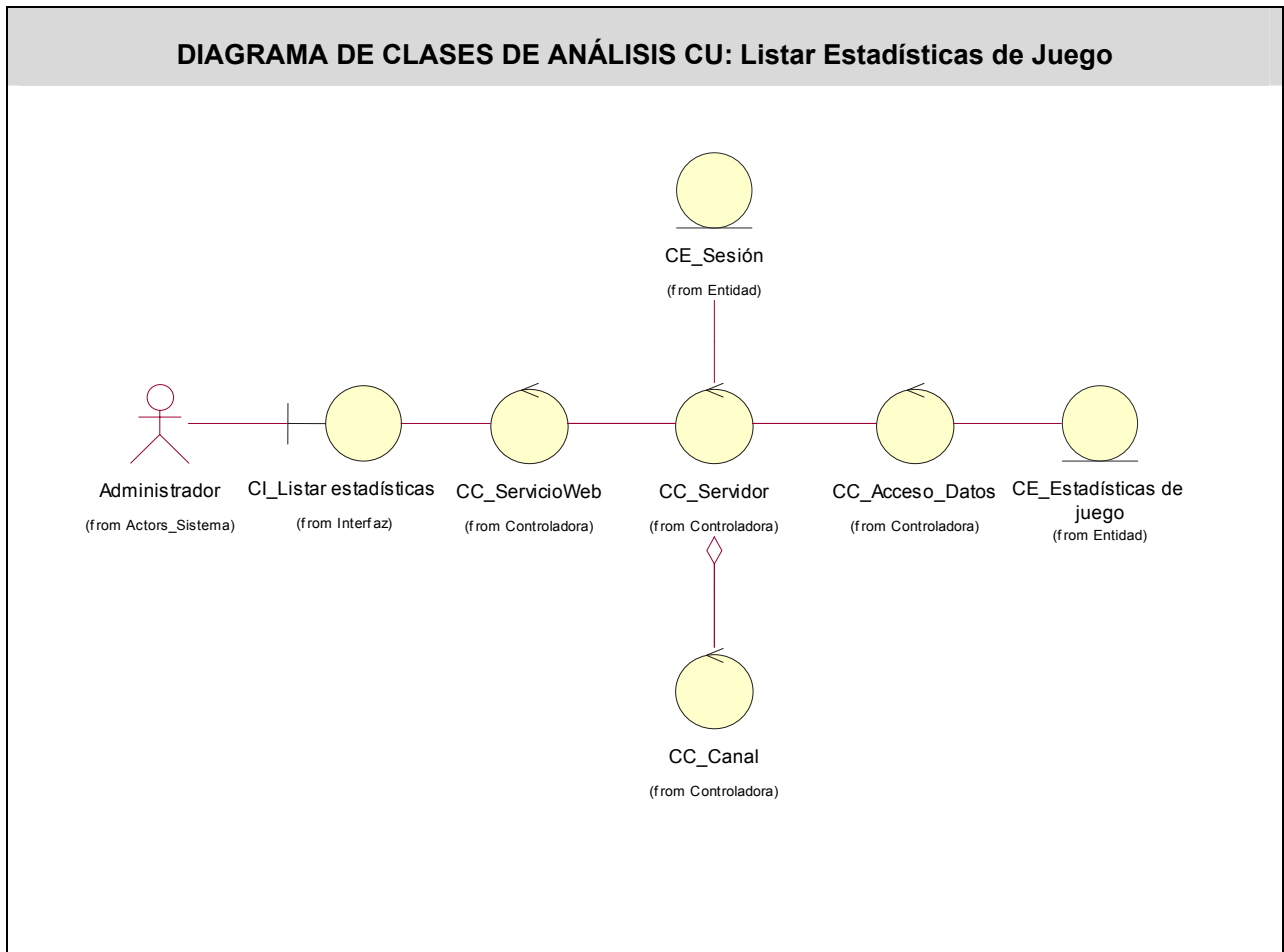
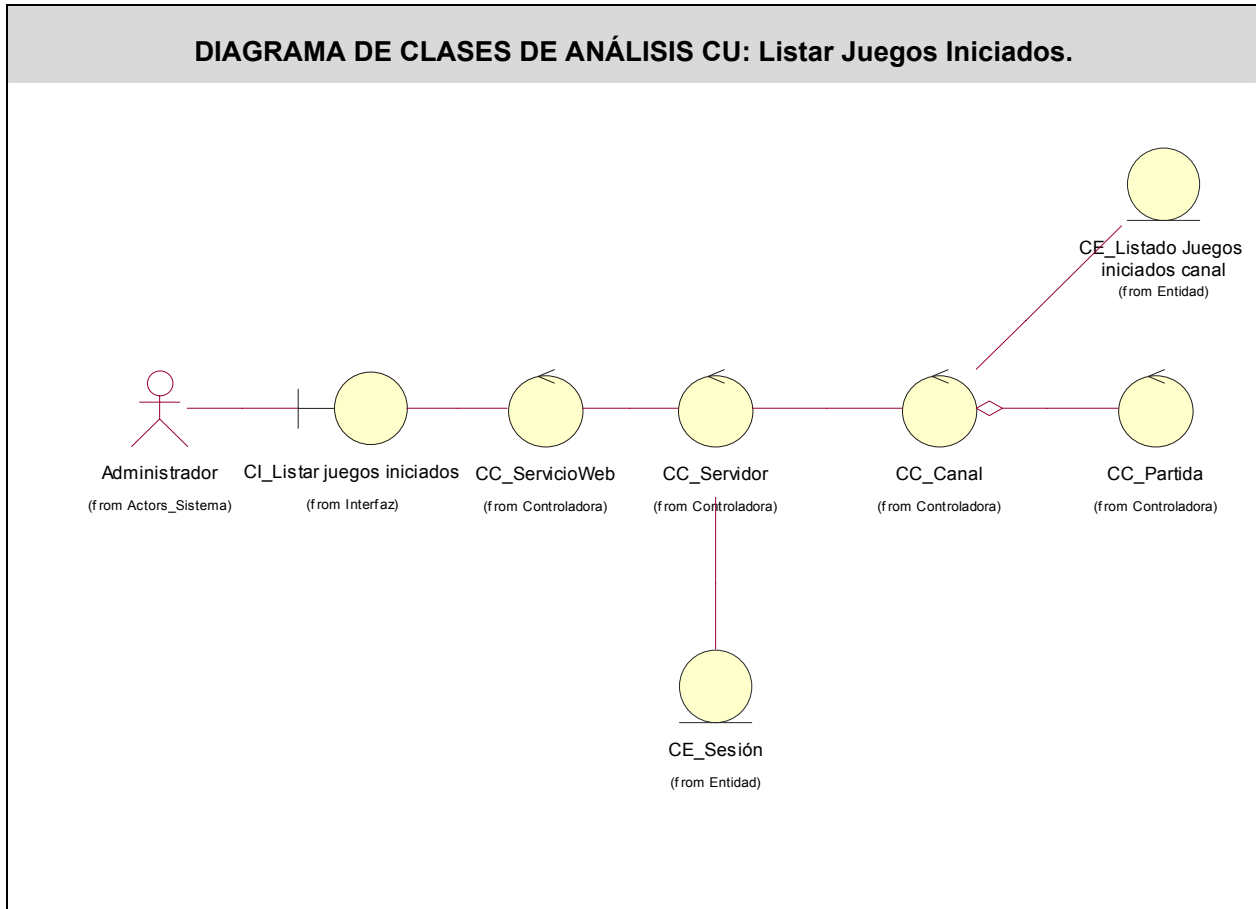


Tabla 26. Diagrama de Clases del Análisis CU: Listar Juegos Iniciados.



**Anexo 3: Diagramas de Clases del Diseño.**

Tabla 27. Diagramas de Clases del Diseño del CU: Autenticar Usuario (J2ME).

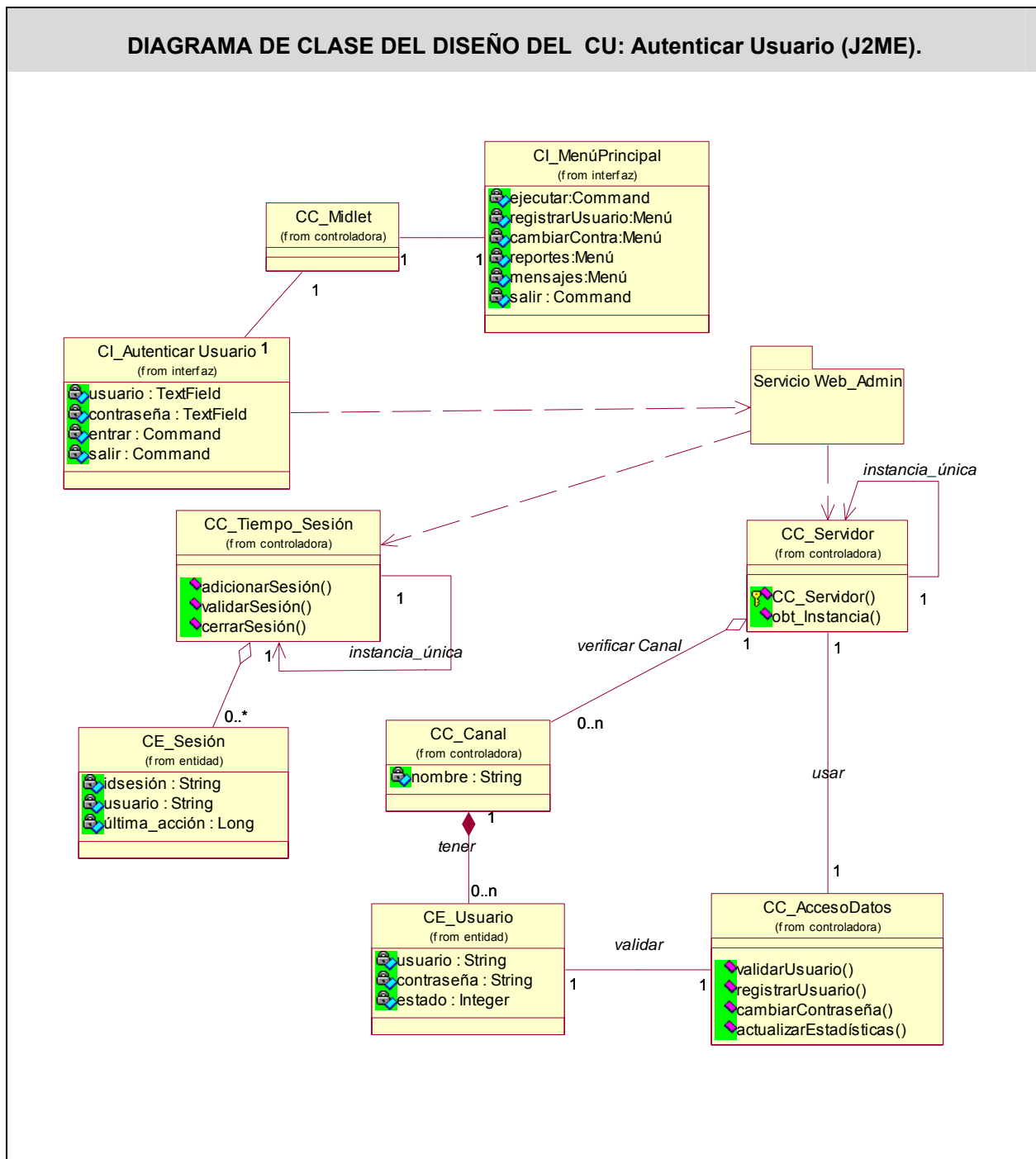


Tabla 28. Diagramas de Clases del Diseño del CU: Autenticar Usuario (Windows Mobile).

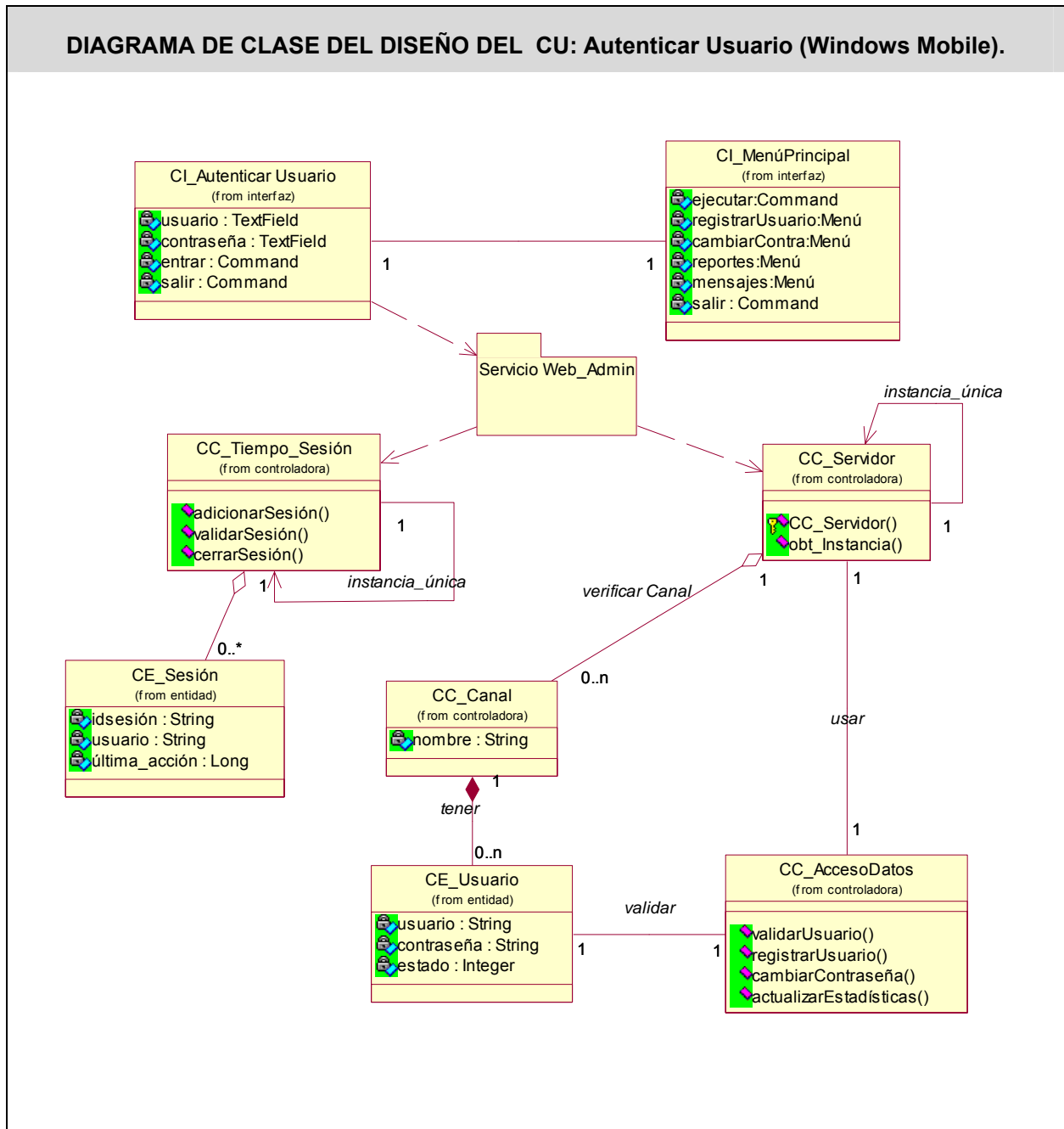


Tabla 29. Diagramas de Clases del Diseño del CU: Listar Juegos Iniciados (J2ME).

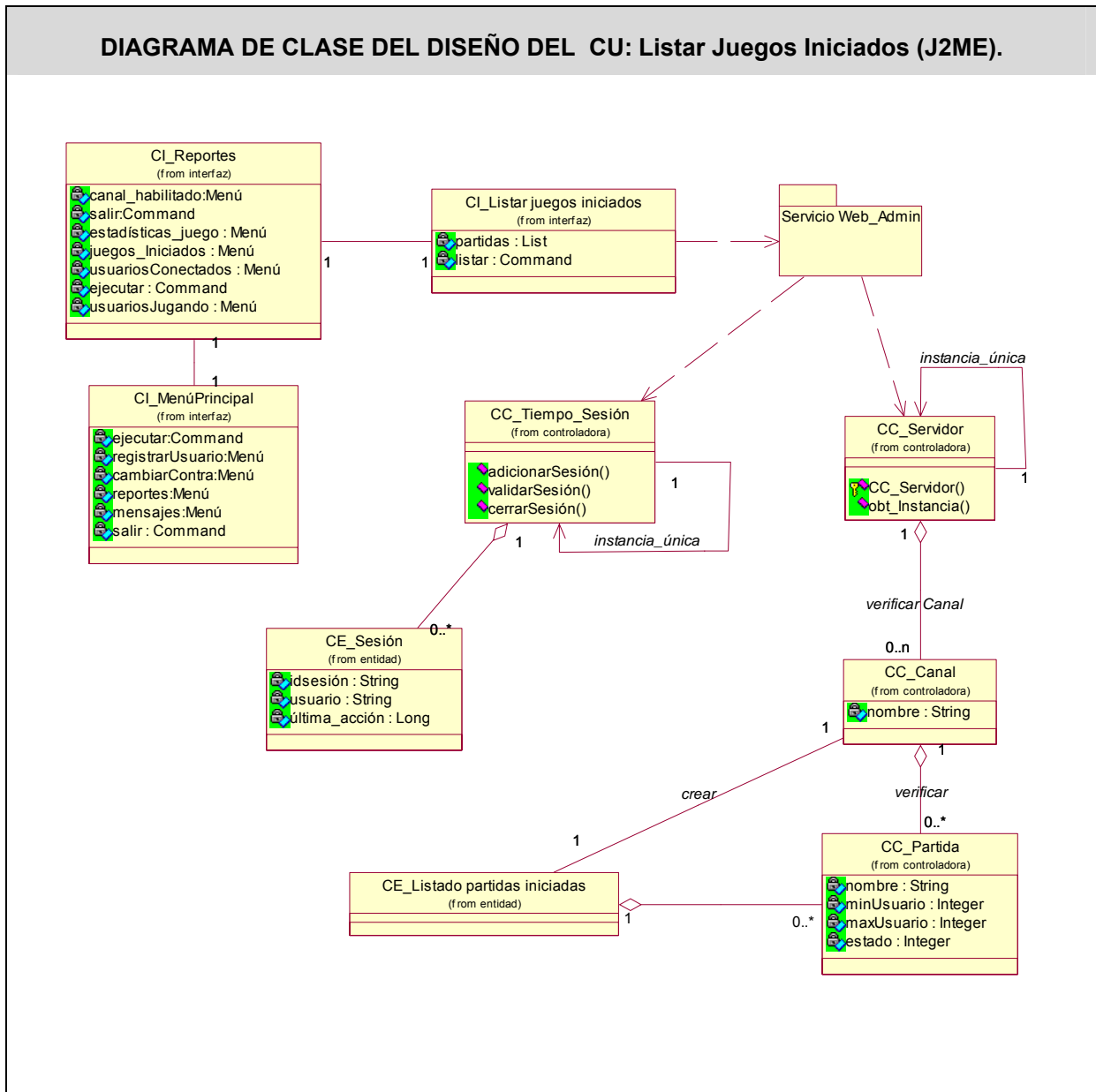


Tabla 30. Diagramas de Clases del Diseño del CU: Listar Juegos Iniciados (Windows Mobile).

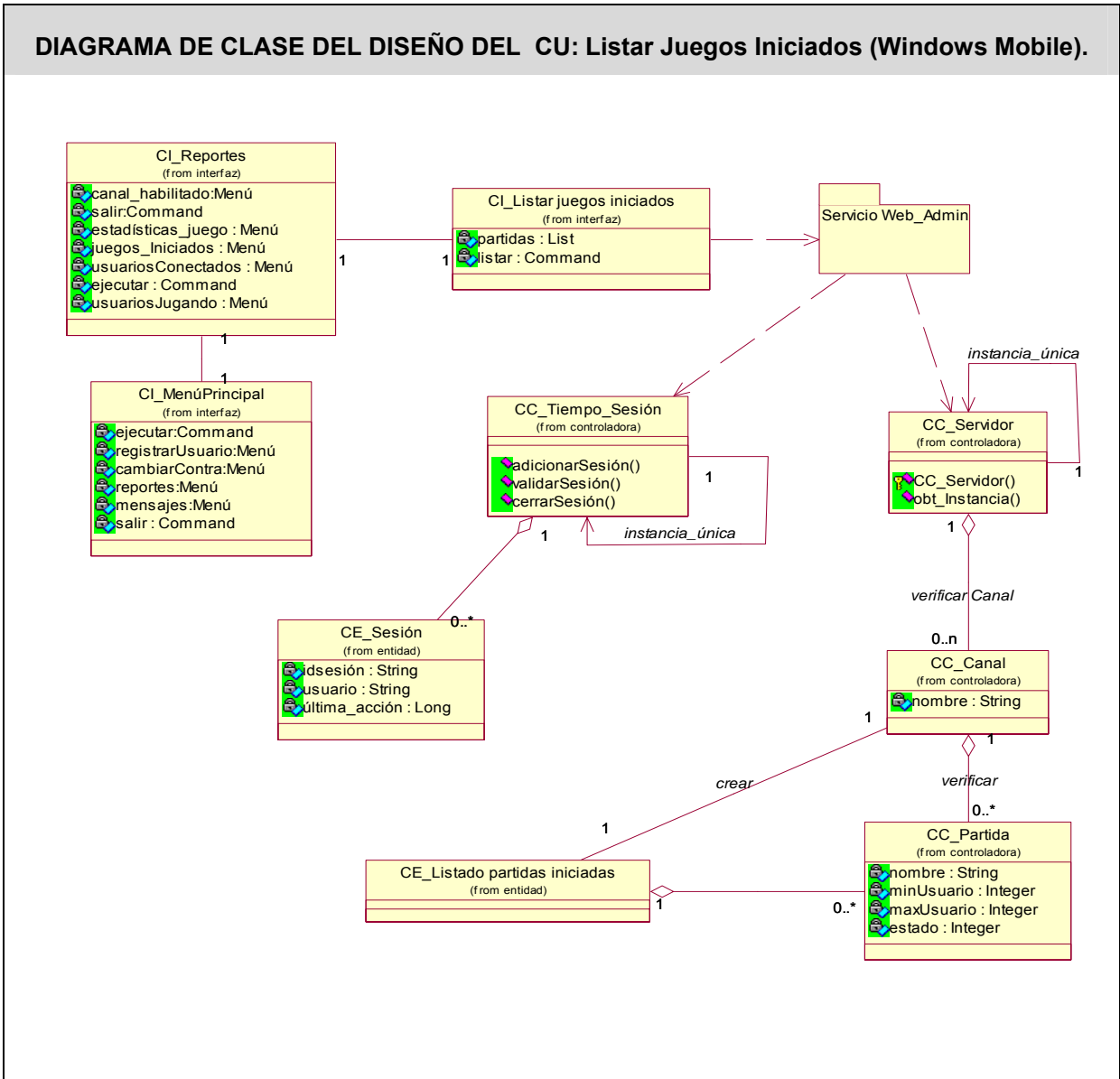
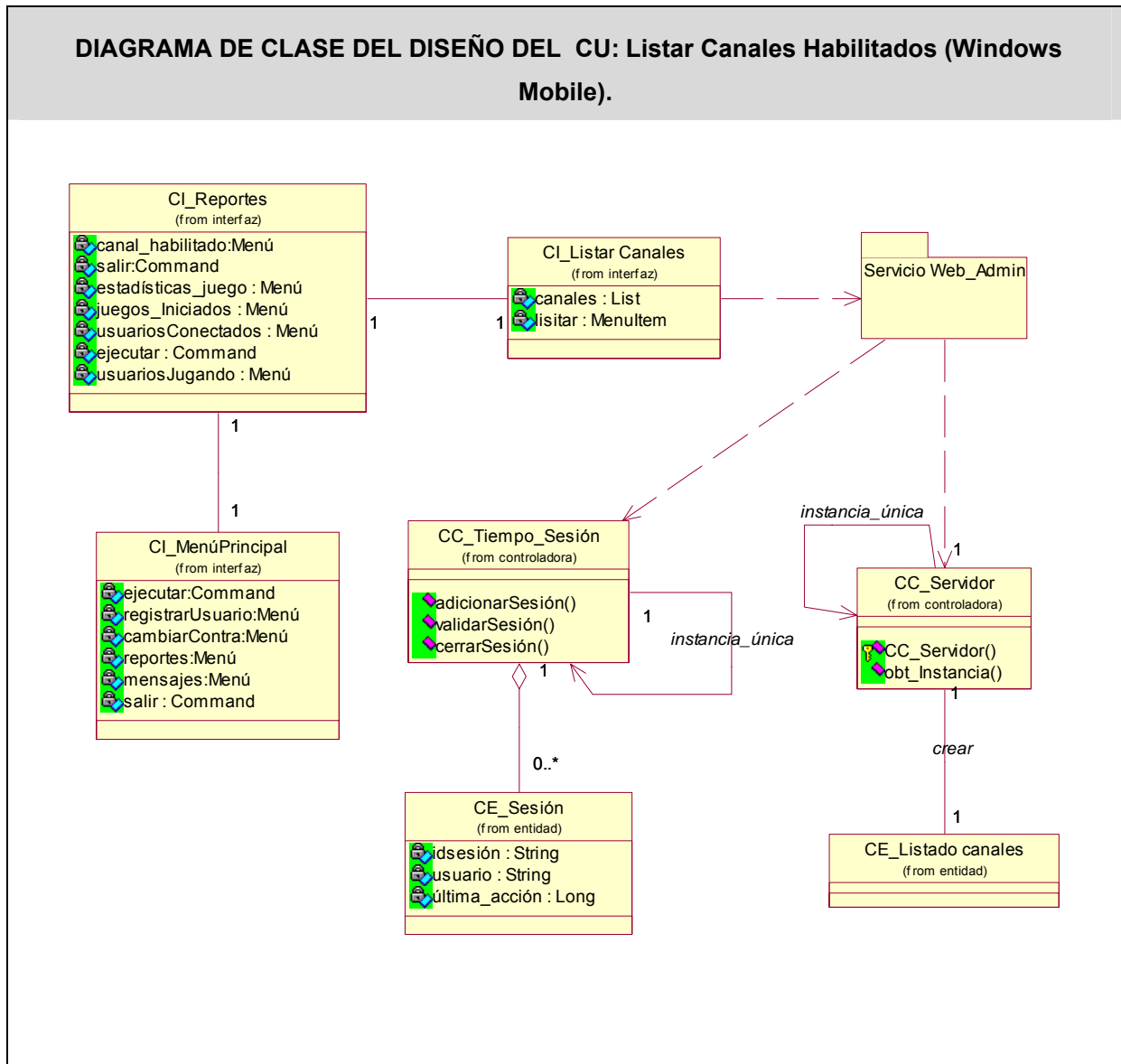


Tabla 31. Diagramas de Clases del Diseño del CU: Listar Canales Habilitados (Windows Mobile).





**Anexo 4: Diagramas secuencias del diseño**

**Tabla 32. Diagrama de secuencia del diseño CU: Autenticar Usuario (J2ME).**

DIAGRAMA DE SECUENCIA DEL DISEÑO CU: Autenticar Usuario (J2ME).

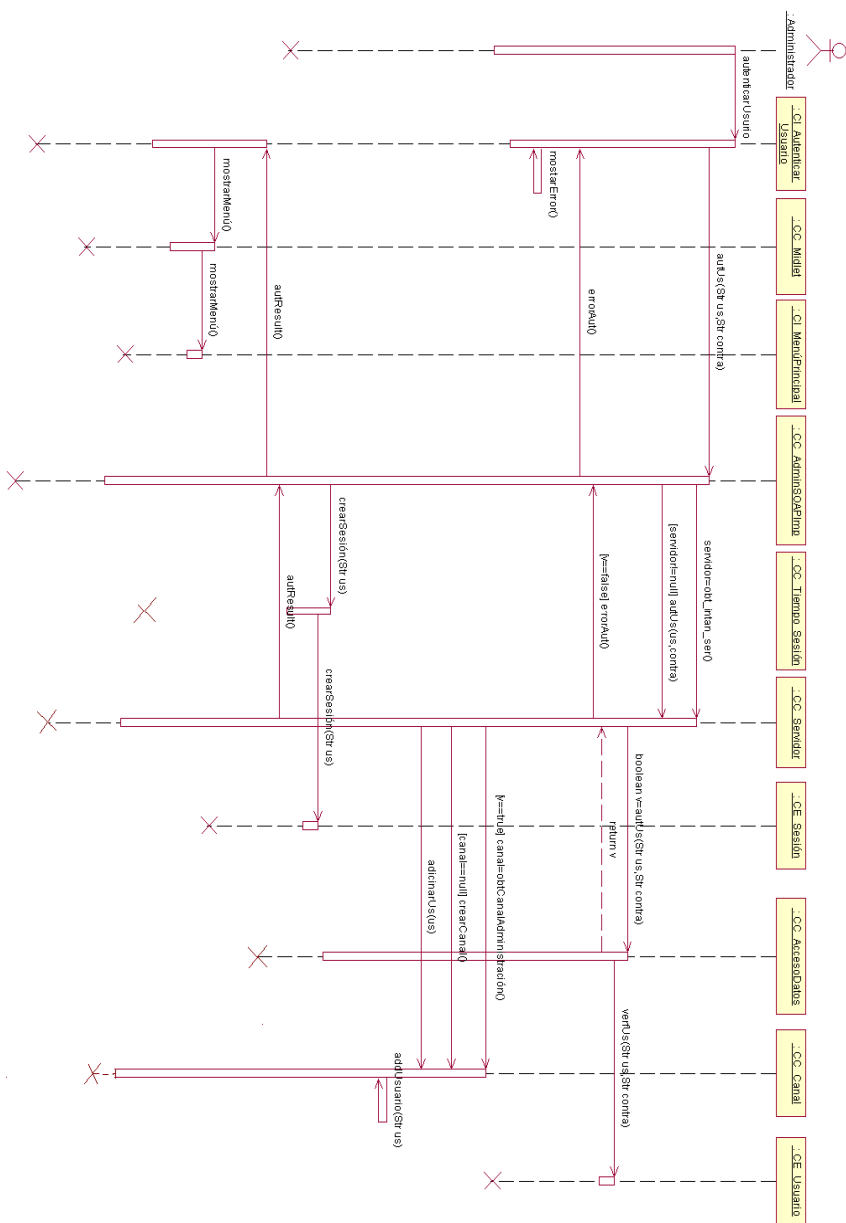


Tabla 33. Diagrama de secuencia del diseño CU: Autenticar Usuario (Windows Mobile).

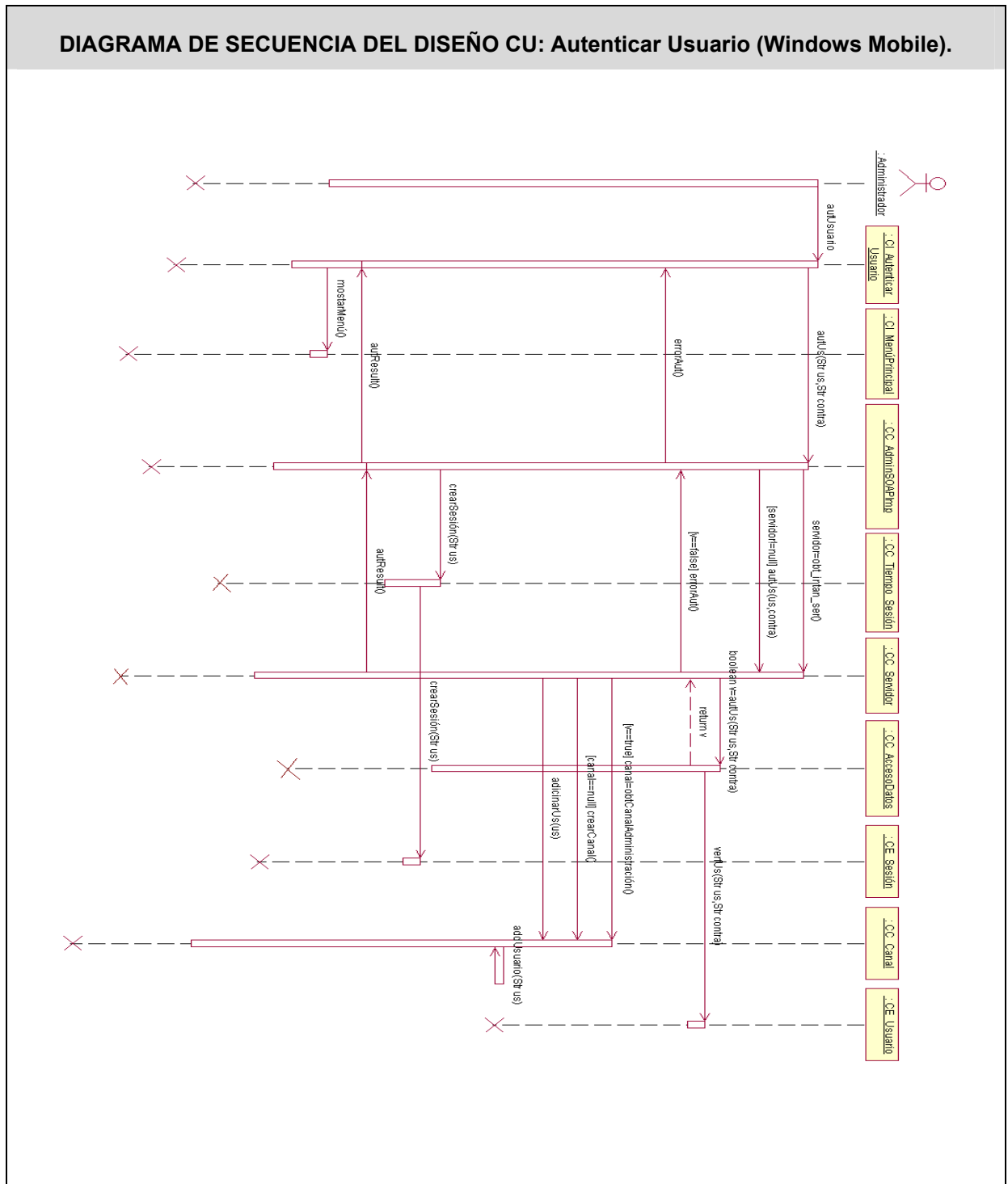


Tabla 34. Diagrama de secuencia del diseño CU: Listar Juegos Iniciados (J2ME).

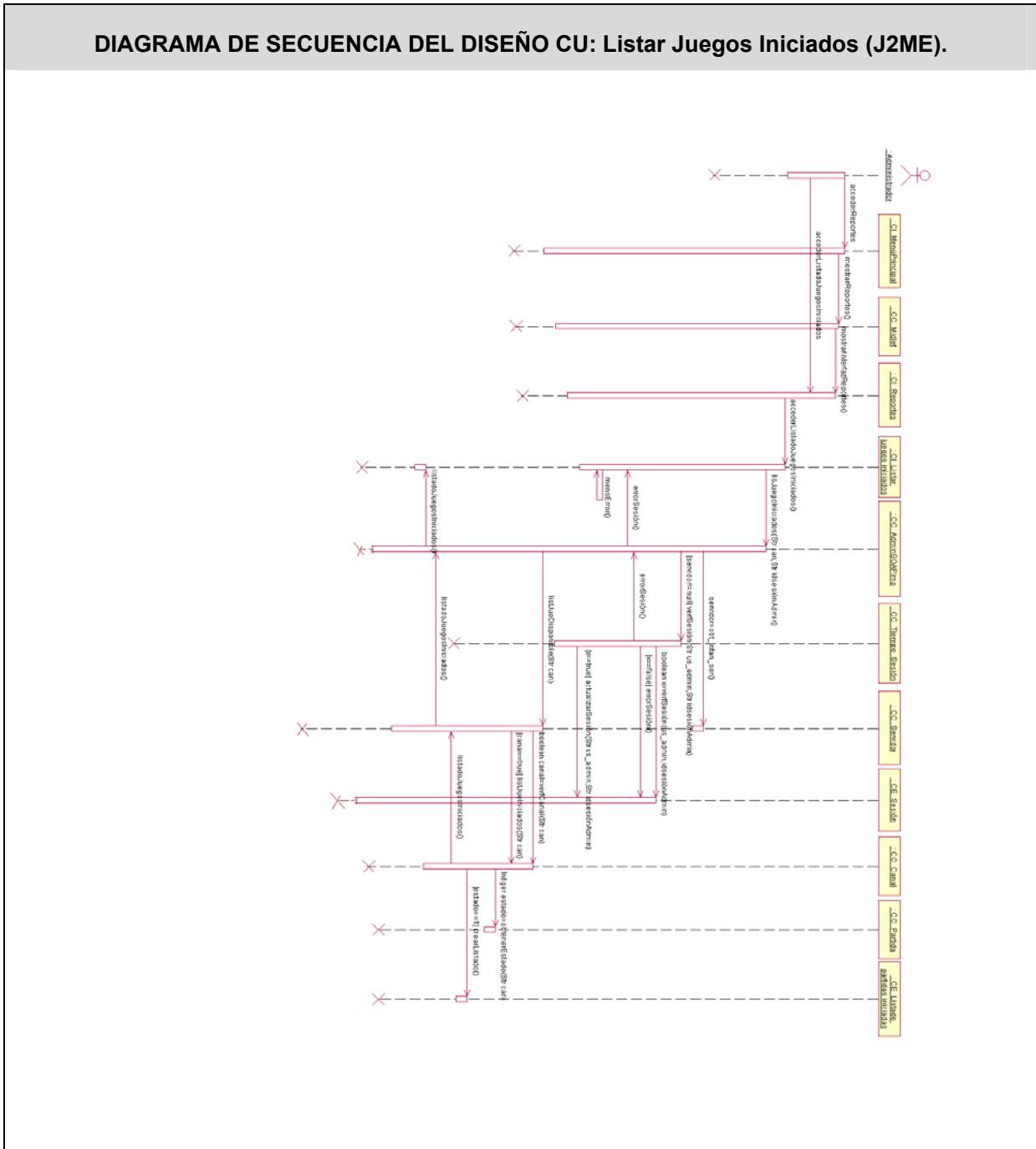


Tabla 35. Diagrama de secuencia del diseño CU: Listar Juegos Iniciados (Windows Mobile).

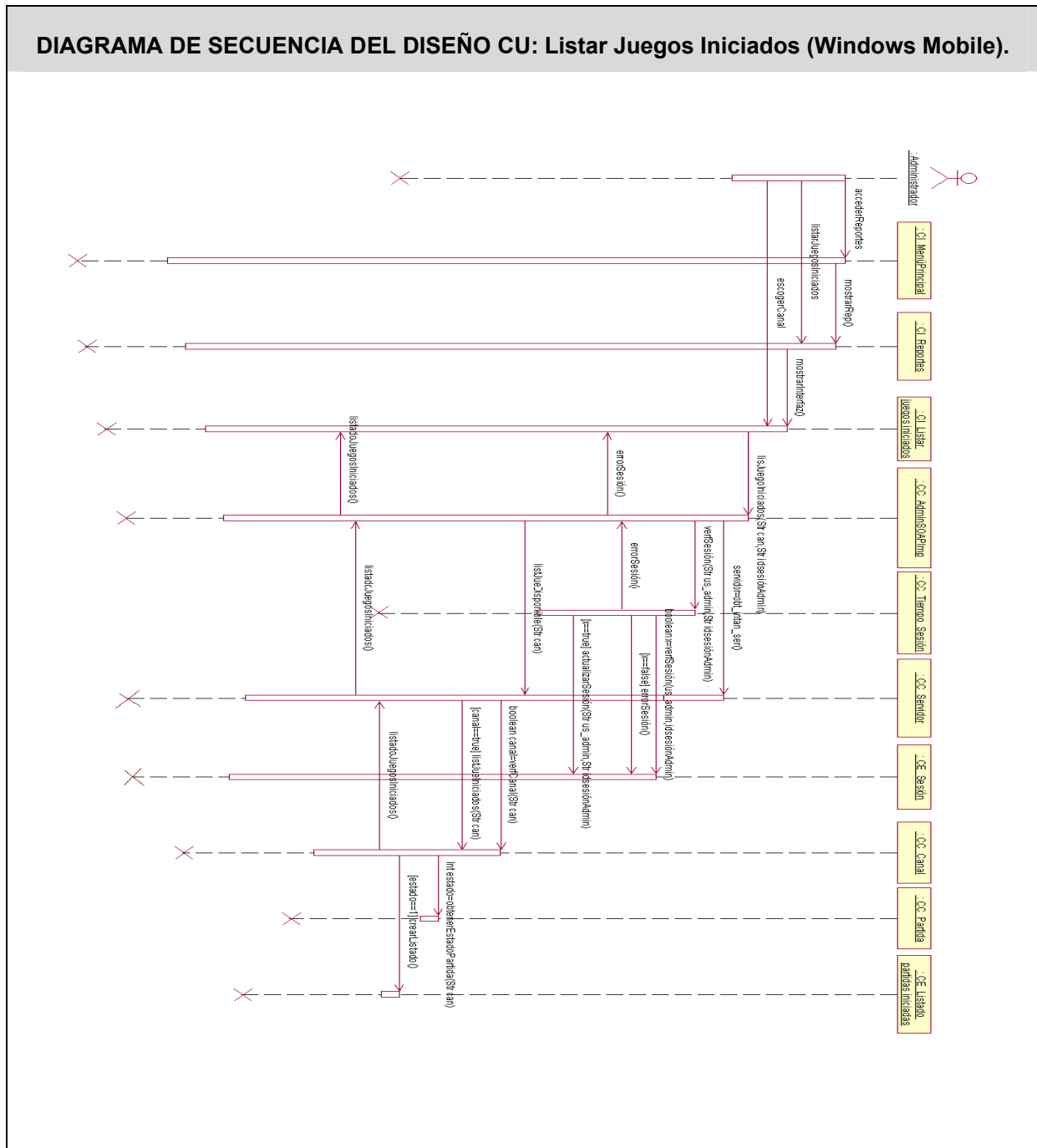
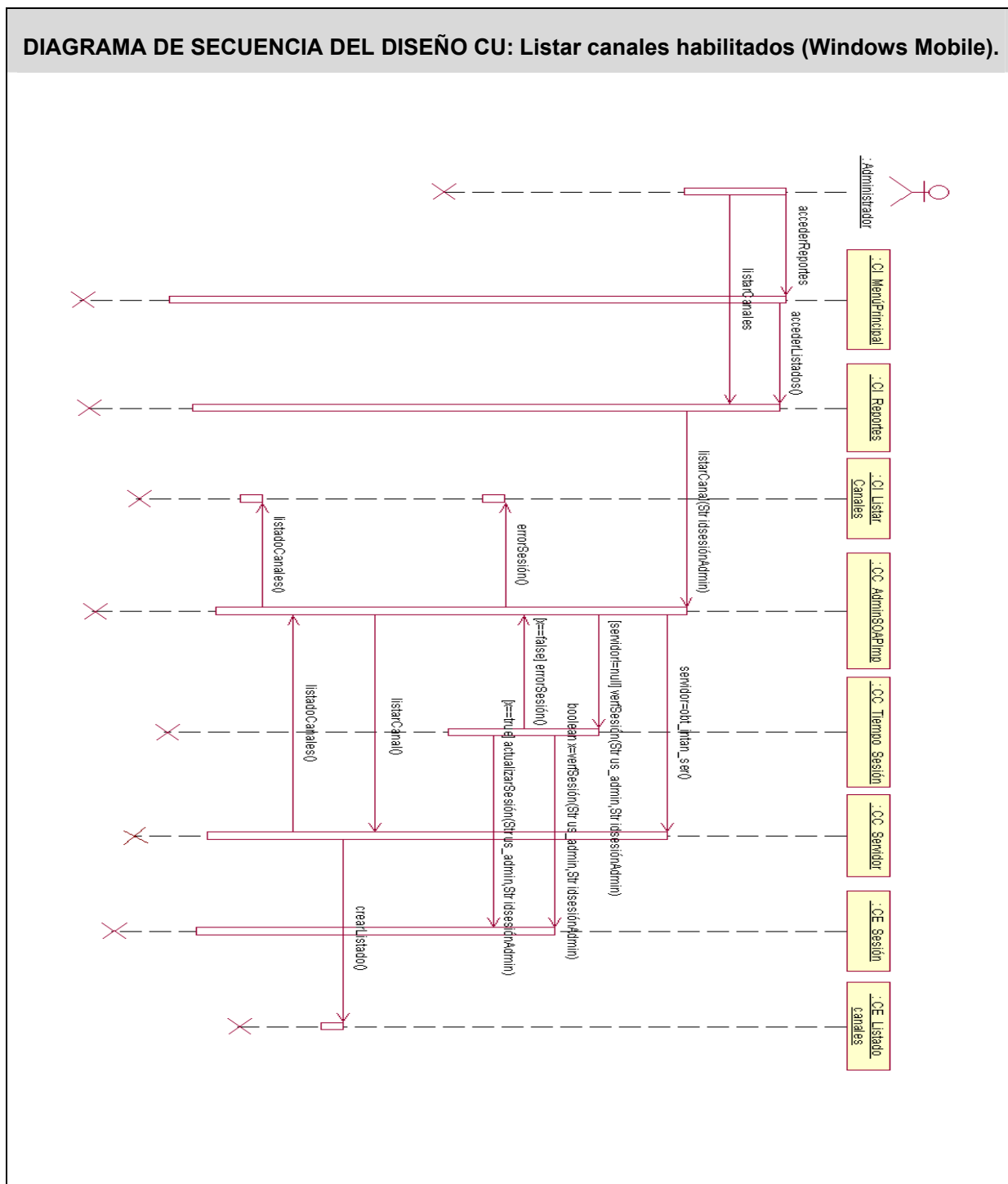


Tabla 36. Diagrama de secuencia del diseño CU: Listar Canales Habilitados (Windows Mobile).



## GLOSARIO

**API (Application Programming Interface):** Conjunto de rutinas, protocolos, y de herramientas para construir aplicaciones de software. Un buen API hace más fácil desarrollar un programa, proporcionando todos los bloques de la construcción. El programador luego pone esos bloques juntos.

**Bluetooth:** Sistema de comunicación inalámbrica que permite la interconexión de diferentes dispositivos electrónicos (PCs, teléfonos fijos o móviles, agendas electrónicas, auriculares, etc.).

**CDC (Connected Device Configuration):** Es una configuración desarrollada para dispositivos con 2 MB o más de memoria disponible para la plataforma, incluyendo RAM y memoria flash o ROM.

**CDMA (Code Division Multiple Access):** Norma de transferencia de información por teléfonos inalámbricos.

**CLDC (Connected, Limited Device Configuration):** Es una configuración diseñada para dispositivos con conexiones de red intermitentes, procesadores lentos y memoria limitada como teléfonos móviles y asistentes personales.

**Conmutación de Paquetes:** Paradigma de comunicaciones mediante el cual cada paquete de un mensaje, recorre una ruta entre sistemas anfitriones (hosts), sin que esa ruta (path) esté previamente definida.

**ETSI (European Telecommunications Standards Institute):** Es una organización de estandarización de la industria de las telecomunicaciones, fabricante de equipos y operadores de redes de Europa, con proyección mundial. ETSI ha tenido gran éxito al estandarizar el sistema de telefonía móvil GSM.

**Firewall:** Elemento de hardware o software utilizado en una red de computadoras para controlar las comunicaciones, permitiéndolas o prohibiéndolas según las políticas de red que haya definido la organización responsable.

**GPRS (General Packet Radio Service):** Servicio que permite enviar paquetes de datos a través de las redes GSM.

**GRASP (General Responsibility Assignment Software Patterns):** Describe los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable.

**GSM (Global System for Mobile Communications):** Es un sistema telefónico digital difundido en Europa usado especialmente por telefonía móvil. Puede funcionar en todo el mundo y en EE.UU. se sitúa en la banda de 1900 MHz

**HomeRF:** Especificación que permite la interconexión de dispositivos inalámbricos en un área pequeña.

**HTTP (Hypertext Transfer Protocol):** Protocolo de Transferencia de Hipertexto utilizado en la WWW para transmitir las páginas de información entre el programa navegador y el servidor. Se destaca que el HTTP seguro es un protocolo HTTP mejorado con funciones de seguridad con clave simétrica.

**HTTPS (Hypertext Transfer Protocol Secure):** Versión segura del protocolo HTTP, utiliza un cifrado basado en SSL para crear un canal cifrado más apropiado para el tráfico de información sensible que el protocolo HTTP.

**IEEE 802.15:** Estándar que define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal con tasas bajas de transmisión de datos.

**IP (Internet Protocol):** Conjunto de reglas que regulan la transmisión de paquetes de datos a través de Internet.

**J2EE (Java 2 Enterprise Edition):** Grupo de *especificaciones* diseñadas por Sun que permiten la creación de aplicaciones empresariales, brindando múltiples funcionalidades.

**J2ME (Java 2 Micro Edition):** Versión desarrollada por la Sun Microsystems de Java, destinada a dispositivos de recursos limitados como PDAs, teléfonos móviles.

**JDK (Java Development Kit):** Software que provee herramientas de desarrollo para la creación de programas en java.



**JSP (Java Server Pages):** Tecnología orientada a crear páginas web con programación en Java. Las páginas JSP están compuestas de código HTML/XML mezclado.

**MIDP (Mobile Information Device Profile):** Es el perfil para dispositivos de información móviles que combina con la configuración CLDC para proporcionar un entorno de ejecución para dispositivos móviles.

**MMS (Multimedia Mobile Service):** Es un estándar de mensajería que le permite a los teléfonos móviles enviar y recibir contenidos multimedia, incorporando sonido, video, fotos, etc.

**PKI (Microsoft Public Key Infrastructure):** Proporciona una infraestructura integrada de clave pública que permite proteger e intercambiar información con un elevado nivel de seguridad y fácil administración.

**PocketPC:** Ordenador de bolsillo, también llamado PDA (Personal Digital Assistant). Se trata de un pequeño ordenador, diseñado para ocupar el mínimo espacio y ser fácilmente transportable que ejecuta el sistema operativo Windows CE de Microsoft entre otros, el cual le proporciona capacidades similares a los PCs de escritorio.

**RMS (Record Management System):** Es una base de datos simple definida por J2ME en MIDP con el principal objetivo de poder almacenar información una vez que el MIDlet finalice.

**Roaming:** Capacidad de un dispositivo para moverse de una zona de cobertura a otra, hacer y recibir llamadas en redes móviles fuera del área de servicio local.

**Servicios Web:** Es un sistema software diseñado para soportar la interoperabilidad máquina - máquina a través de una red. Un servicio web es una comunicación por medio de mensajes SOAP entre diferentes equipos a través de una red.

**Servlet:** Pequeño programa que corre en un servidor. Por lo general son aplicaciones Java que corren en un entorno de servidor web. Esto es análogo a una aplicación Java que corre en un navegador.

**SIM (Single Identification Module):** Tarjeta que identifica y da servicio a un usuario, utilizado en teléfonos GSM.

**Smartphone:** Teléfono móvil que ofrece funcionalidades avanzadas que van más allá de las de un teléfono tradicional y están más cercanas a las de una computadora.

**SMS (Short Message Service):** Es un servicio de mensajería por teléfonos celulares. Con este sistema se puede enviar o recibir mensajes entre celulares y, luego, a través de internet.

**SSL (Secure Sockets Layer):** Protocolo criptográfico que proporciona comunicaciones seguras en Internet. SSL proporciona autenticación y privacidad de la información entre las computadoras sobre Internet mediante el uso de criptografía.

**SVN (Subversion):** Aplicación para el control de versiones que permite gestionar los cambios y versiones que realizamos en nuestros desarrollos de una forma sencilla.

**TDMA (Time Division Multiple Access):** Tecnología inalámbrica de segunda generación, que distribuye las unidades de información en ranuras alternas de tiempo, dando acceso múltiple a un número reducido de frecuencias. TDMA permite dar servicios de alta calidad de voz y datos.

**UML (Unified Modeling Language):** Lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

**UMTS (Universal Mobile Telecommunications System):** Estándar de telefonía móvil celular de banda ancha y alta velocidad (de 2 Mbps en adelante). Se trata de un sistema de tercera generación que permite la conexión a Internet. Sustituirá a los sistemas GSM y GPRS.

**URL (Uniform Resource Locator):** Secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

**WAP (Wireless Application Protocol):** Protocolo de aplicación de telefonía inalámbrica que permite a los usuarios de teléfonos móviles el acceso a servidores web.

**WiFi (Wireless Fidelity):** Conjunto de estándares para redes inalámbricas basado en las especificaciones IEEE 802.11 (especialmente la 802.11b), creado para redes locales inalámbricas, pero que también se utiliza para acceso a internet.

**Windows Mobile:** Sistema operativo compacto, con una suite de aplicaciones básicas para dispositivos móviles basados en la API Win32 de Microsoft.

**WLAN: (Wireless Local Area Network):** Red de comunicación inalámbrico por radio frecuencia alternativo a las LAN con cables.

**XML (Extensible Markup Language):** Desarrollado por el World Wide Web Consortium (W3C). Su objetivo es conseguir páginas web más semántica. XML separa la estructura del contenido y permite el desarrollo de vocabularios modulares. Se trata de un formato abierto.