

**Universidad de las Ciencias Informáticas  
Facultad 2**



**Título: SIRIO, Plataforma para la gestión de juegos  
multiusuario en dispositivos móviles vía GPRS**

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

**Autores:** Denys Buedo Hidalgo

Yunisel Viera Vargas

**Tutor:** Mijail Del Toro Céspedes

**Co-tutor:** York Figueroa Valdés

Ciudad de la Habana, junio del 2007

# 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 \_\_\_\_\_.

\_\_\_\_\_  
Yunisel Viera Vargas  
Firma del Autor

\_\_\_\_\_  
Denys Buedo Hidalgo  
Firma del Autor

\_\_\_\_\_  
Firma del Tutor  
Mijail Del Toro Céspedes

\_\_\_\_\_  
Firma del Co-Tutor  
York Figueroa Valdés

## AGRADECIMIENTOS

A la máxima dirección de la Revolución Cubana, a nuestro Comandante Fidel Castro y a la UCI, por permitirnos formar parte de este proyecto futuro y contribuir a nuestra formación como profesionales revolucionarios. Nuestro más sincero agradecimiento a nuestro tutor Mijail (Chuchi), York, Tanía, Damián (el viejo), Abel y a todos los trabajadores de Procyon que de una forma u otra nos prestaron una mano amiga.

Para todos ellos, Muchas Gracias

**Yunisel y Denys**

A mis amistades: Osleydis, Dariena, Yudier, Onel, Jose Carlos, Plá, y todos los que me han ayudado de algún modo.

A mis hermanos, mi cuñada, y mi familia que siempre me han apoyado.

A mis padres, Teresa y Modesto a quienes debo agradecerles haber llegado hasta aquí.

**Yunisel**

A mis padres por siempre alumbrarme el camino y apoyarme en todo momento, a mi hermano que es el mejor del mundo, a mis tíos de la Habana (Ramón y Pura, Lázaro e Idalma) y a mi familia de Holguín que siempre me apoyaron y ayudaron.

A mi novia Yelenys por tener siempre una sonrisa para mí.

A mis compañeros de aula, en especial a Jose y Damián que me aclararon las dudas cuando la cosa estaba fea, a Darién por siempre estar dispuesto a ayudarme y ser un hermano.

A todas las personas que de una forma u otra han hecho posible este sueño.

**Denys**

## DEDICATORIA

A mis hermanos, Yunior, Yesenia y a mi cuñada Isabel, que siempre están a mi lado, aunque no sea físicamente. A mis tíos y al resto familia. A mis mejores amigas de la Universidad, Osleydis y Dariena quienes han estado siempre a mi lado. A todos los que me han acompañado, mis amigos, compañeros de aula y todos los que me han ayudado a lo largo de estos cinco años.

En especial, a mis padres, Teresa y Modesto, quienes me han apoyado y guiado siempre, de manera incondicional.

**Yunisel**

A mis padres y toda mi familia que siempre me ayudaron e hicieron posible este sueño.

A mi novia Yelenys que me apoyó siempre.

A mis compañeros de aula y profesores, en fin a todas las personas que he conocido en éstos 5 años y que han aportado su granito de arena.

**Denys**

## RESUMEN

El aumento en la capacidades de software y hardware de los teléfonos celulares ha provocado que el número de aplicaciones destinadas a ellos tome niveles elevados, actualmente constituyen una de las áreas de mayor crecimiento en el mercado a nivel mundial. Los juegos son los contenidos para móviles más descargados por los abonados de todo el planeta, destacándose entre ellos los juegos del tipo multi-jugador.

En Procyon, entidad cubana que dedica al desarrollo de aplicaciones para móviles se lleva a cabo un amplio trabajo en el área de los juegos para celulares, principalmente en los de un solo usuario, aunque se está incursionado actualmente en los juegos para múltiples jugadores.

Producto a la gran aceptación que poseen los juegos multi-jugador se decidió el desarrollo de Sirio, plataforma destinada a la gestión de juegos multiusuario en dispositivos móviles vía GPRS. La aplicación permite a los clientes registrarse en el sistema, crear, iniciar, unirse, pausar, reanudar y abandonar partidas, enviar mensajes y paquetes de juegos, así como solicitar información relacionado con usuarios, partidas y canales habilitados en el servidor.

Se desarrolló un API de conexión con el servidor, donde están plenamente definidas las funcionalidades permitidas a los clientes y que disminuye el tiempo de desarrollo de los juegos.

Como parte del proceso de pruebas del sistema se desarrollaron los juegos por turnos TicTacToe y Ping Ball obteniéndose los resultados esperados.

# TABLA DE CONTENIDOS

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>5</b>
1.1 INTRODUCCIÓN .....	5
1.2 TELEFONÍA CELULAR O MÓVIL .....	5
1.3 COMUNICACIONES INALÁMBRICAS .....	8
1.3.1 GSM .....	9
1.3.2 GPRS .....	11
1.3.3 UMTS .....	12
1.3.4 Conclusiones del epígrafe.....	13
1.4 PLATAFORMAS PARA JUEGOS EN RED DESDE TELÉFONOS CELULARES. PRINCIPALES TENDENCIAS. ....	13
1.4.1 Descripción de soluciones existentes a nivel mundial .....	15
1.4.2 Conclusiones del epígrafe.....	18
1.5 ARQUITECTURA CLIENTE SERVIDOR .....	19
1.6 METODOLOGÍA DE DESARROLLO EMPLEADA .....	19
1.6.1 Proceso Unificado de desarrollo de Software .....	20
1.7 PLATAFORMA DE DESARROLLO Y LENGUAJE DE PROGRAMACIÓN EMPLEADOS.....	20
1.7.1 ¿Por qué J2ME? .....	20
1.7.2 ¿Por qué J2EE? .....	22
1.8 JSP Y SERVLET .....	23
1.9 SERVIDOR WEB UTILIZADO .....	24
1.10 HERRAMIENTAS DE DESARROLLO EMPLEADAS .....	24
1.10.1 Herramientas de modelado UML .....	24
1.10.2 Entorno Integrado de desarrollo.....	25
1.10.3 Herramienta para el control de versiones .....	26
1.11 CONCLUSIONES .....	28
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA</b> .....	<b>29</b>
2.1 INTRODUCCIÓN .....	29
2.2 FLUJO DE LOS PROCESOS INVOLUCRADOS EN EL CAMPO DE ACCIÓN .....	29
2.3 ANÁLISIS CRÍTICO DE SOLUCIONES EXISTENTES A NIVEL MUNDIAL.....	30
2.4 OBJETO DE AUTOMATIZACIÓN .....	30
2.5 PROPUESTA DEL SISTEMA .....	31
2.6 MODELO DE NEGOCIO.....	33
2.6.1 Modelo de dominio.....	33
2.7 REQUERIMIENTOS .....	36
2.7.1 Requerimientos funcionales.....	36
2.7.2 Requerimientos no funcionales.....	38
2.8 MODELO DEL SISTEMA. DEFINICIÓN DE CASOS DE USO .....	40
2.8.1: Definición de actores.....	40
2.8.2 Casos de uso del sistema .....	41
2.8.3 Descripción detallada de los casos de uso del sistema.....	43
2.9 CONCLUSIONES .....	43

<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA .....</b>	<b>45</b>
3.1 INTRODUCCIÓN .....	45
3.2 MODELO DE ANÁLISIS .....	45
3.2.1 <i>Diagrama de clases del análisis</i> .....	45
3.3 MODELO DE DISEÑO .....	46
3.3.1 <i>Diagrama de clases del diseño</i> .....	47
3.3.2 <i>Diagrama de secuencia</i> .....	49
3.4 PATRONES DE DISEÑO EMPLEADOS EN EL DESARROLLO DE LA APLICACIÓN .....	50
3.5 TRATAMIENTO DE ERRORES.....	51
3.6 SEGURIDAD .....	53
3.7 INTERFAZ.....	53
3.8 CONCEPCIÓN DE LA AYUDA.....	55
3.9 CONCLUSIONES .....	56
<b>CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA .....</b>	<b>57</b>
4.1 INTRODUCCIÓN .....	57
4.2 MODELO DE IMPLEMENTACIÓN.....	57
4.2.1 <i>Diagrama de despliegue</i> .....	58
4.3 MODELO DE PRUEBAS .....	59
4.3.1 <i>Métodos de Prueba</i> .....	60
4.3.2 <i>Diseño de Casos de Prueba. Prueba de Caja negra</i> .....	60
4.4 CONCLUSIONES .....	64
<b>CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD .....</b>	<b>65</b>
5.1 INTRODUCCIÓN .....	65
5.2 PLANIFICACIÓN BASADA EN CASOS DE USO .....	65
5.3 BENEFICIOS TANGIBLES E INTANGIBLES.....	71
5.4 ANÁLISIS DE COSTOS Y BENEFICIOS .....	72
5.5 CONCLUSIONES .....	73
<b>CONCLUSIONES .....</b>	<b>74</b>
<b>RECOMENDACIONES.....</b>	<b>75</b>
<b>BIBLIOGRAFÍA.....</b>	<b>76</b>
<b>ANEXO .....</b>	<b>79</b>
<b>GLOSARIO.....</b>	<b>98</b>

# INTRODUCCIÓN

En la actualidad el uso de las comunicaciones inalámbricas a nivel mundial, ha aumentado significativamente, producto a las ventajas en cuanto a conectividad, movilidad, reducción de costos de infraestructura, rapidez y facilidad de despliegue que ofrecen. Esto ha propiciado el desarrollo y surgimiento de novedosas tecnologías de comunicación, la telefonía celular ocupa un lugar cimero en esta era tecnológica.

Desde sus inicios a finales de los 70, los teléfonos celulares han revolucionado las actividades que se realizan diariamente y se han convertido en una herramienta primordial para todos, existen actualmente más de 2839 millones de abonados celulares distribuidos por todo el planeta (**3G Americas, 2007**). Con el paso de los años los teléfonos celulares han incrementado sus capacidades de hardware y software, lo que ha posibilitado el desarrollo de aplicaciones destinadas a ellos, la producción de juegos constituye una de las áreas de mayor importancia.

Los juegos para celulares no se han limitado a brindar entretenimiento a un solo jugador, sino que se han vinculado al desarrollo de las comunicaciones, actualmente se programan juegos en red para múltiples usuarios concurrentes y de mucha aceptación por todos. Este tipo de juegos es muy solicitado, ya que incorpora el aspecto competitivo entre los jugadores, convirtiéndose en una forma de entretenimiento más emotiva. Empresas informáticas de importancia a escala mundial como Nokia, Sony Ericsson y Motorola han invertido sus recursos en ellos.

La producción de juegos en red para teléfonos celulares trae consigo el desarrollo de otras aplicaciones informáticas que permiten su funcionamiento, tal es el caso de los servidores, que tienen la importante tarea de gestionar el intercambio de información y recurso entre los clientes conectados, así como la gestión de las cuentas de usuarios y la lógica de los juegos. Existen actualmente 2 tendencias principales en el desarrollo de estos servidores, por un lado están los que solo gestionan la lógica de un juego determinado y por otro, los que alojan y atienden a varios juegos a la vez.

En Procyon, entidad que se encuentra enmarcada dentro de la infraestructura productiva de la universidad (UCI), se fomenta el uso de las nuevas tecnologías de comunicación y la producción de aplicaciones y contenidos para móviles. La producción de juegos constituye una de las ramas más

experimentadas. Se han obtenido resultados, fundamentalmente en los juegos de tipo simple aunque se han desarrollado algunas variantes multiusuario usando tecnología Bluetooth para la comunicación.

Producto del alto crecimiento en la demanda de los juegos en red se decidió desarrollar una plataforma para la gestión de juegos multiusuario, con la característica fundamental de brindar soporte para múltiples juegos. Se decidió el uso de la tecnología de servicio general de paquetes de radio, GPRS (del Inglés: General Packet Radio Service) para la comunicación debido a las grandes ventajas que la misma ofrece.

Se llevó a cabo la puesta en marcha de la investigación con el fin de resolver el siguiente problema:

¿Cómo desarrollar una plataforma que funcione como servidor para juegos, que permita conexiones simultáneas desde teléfonos celulares a través de GPRS y maneje eficientemente la concurrencia de las mismas?

Para ello se plantea como objeto de estudio, el desarrollo de una plataforma para juegos en línea desde teléfonos celulares sobre GPRS. El campo de acción del problema es el desarrollo de aplicaciones informáticas para teléfonos celulares.

Con el fin de obtener un software de alta calidad se plantean los siguientes objetivos.

Objetivos Generales:

- Analizar, diseñar e implementar la plataforma para la gestión de las peticiones de diferentes teléfonos celulares de manera simultánea.

Objetivos Específicos:

- Permitir comunicación simultánea entre usuarios conectados a un juego desde su teléfono celular vía GPRS.
- Desarrollar una aplicación servidor encargada de distribuir la información entre teléfonos celulares con juegos que tengan conexión con el servidor.

- Desarrollar un API (del inglés: Application Program Interface) cliente de comunicación que facilite el transporte de información entre teléfonos celulares con el servidor vía GPRS.
- Desarrollar una aplicación Web para el monitoreo y administración de grupos de usuarios que comparten datos entre juegos clientes de un mismo tipo.

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

- ¿Cuáles son las principales características de GPRS?
- ¿Qué ventajas ofrece el desarrollo de un API para el cliente?
- ¿Cómo manejar eficientemente las peticiones de los clientes de forma concurrente?
- ¿Cómo gestionar juegos de diferentes tipos?

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

- Análisis de las tecnologías existentes a nivel mundial que permiten el establecimiento de comunicaciones inalámbricas entre dispositivos móviles.
- Estudio y selección de la tecnologías disponibles para el desarrollo de aplicaciones para teléfonos celulares.
- 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, descritos 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 el que se encuentran actualmente los servidores para juegos en red para teléfonos celulares a diferentes escalas, así como las principales tendencias, plataformas de desarrollo, tecnologías, metodologías y herramientas que hicieron posible la realización del presente trabajo.

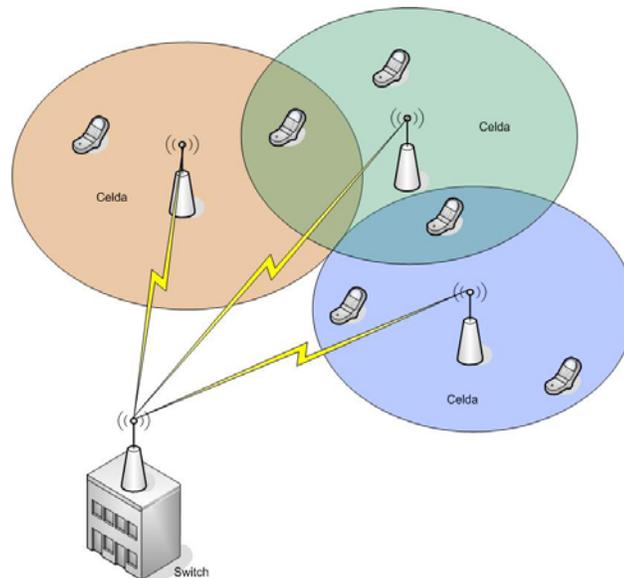
## **1.2 Telefonía celular o móvil**

La telefonía móvil, también llamada telefonía celular, básicamente está formada por dos grandes partes: una red de comunicaciones (o red de telefonía móvil) y los terminales (o teléfonos celulares) que permiten el acceso a dicha red.

Los teléfonos celulares, por sofisticados que sean, no dejan de ser radio transmisores personales. Siendo un sistema de comunicación telefónica totalmente inalámbrico, los sonidos se convierten en señales electromagnéticas, que viajan a través del aire, siendo recibidas y transformadas nuevamente en mensaje a través de antenas repetidoras o vía satélite.

La genialidad del teléfono celular reside en que una ciudad puede ser dividida en pequeñas "células" (o celdas), que permiten extender la frecuencia por toda una ciudad. Esto es lo que hace posible que millones de usuarios utilicen el servicio en un territorio amplio sin tener problemas.

He aquí cómo funciona, se puede dividir un área (como una ciudad) en células o celdas. Cada célula es típicamente de un tamaño de 10 millas cuadradas (unos 26Km<sup>2</sup>). En la figura # 1 se muestra la división de un territorio en celdas.



**Figura. # 1: Funcionamiento de la telefonía celular**

El tamaño de las células puede variar, dependiendo del lugar en que se encuentre. Las estaciones de base se separan de 1 a 3 Km. en zonas urbanas, aunque pueden llegar a separarse por más de 35Km en zonas rurales. En zonas muy densamente pobladas o áreas con muchos obstáculos (como edificios altos), las células pueden concentrarse a distancias cada vez menores.

En el devenir de los últimos 30 años, la telefonía celular ha transitado por distintas generaciones y en cada una de ellas resulta plenamente reconocible un conjunto de características que las definen.

La primera generación analógica de sistemas de comunicación móvil se desarrolló en los años ochenta, fue seguida por una segunda generación digital que proporcionó avances significativos en cuanto al número de suscriptores asignados a una frecuencia dada, la seguridad y calidad de la voz, y además conformó las bases para la prestación de otros servicios como la transmisión de datos (SUD, 2005).

La 1ra y 2da generación de sistemas de comunicación móvil tuvieron como objetivo primordial dar soporte a la comunicaciones de voz, y aunque pueden ser usadas para transmitir datos a baja velocidad no satisfacen los requerimientos de transmisión de grandes volúmenes de información a altas velocidades

entre terminales inalámbricos y la red fija, necesarios para aplicaciones como videoconferencia, conexión a Internet, gestión multimedia y correo con video y audio. Por otra parte, existe la necesidad de proporcionar capacidad de roaming internacional a usuarios de teléfonos móviles que actualmente pierden la conexión o tienen que recurrir a complicados procedimientos cuando viajan a otros países debido al intrincado laberinto de normas móviles en uso por diferentes países.

Una vez que la segunda generación se estableció, las limitantes de algunos sistemas en lo referente al envío de información se hicieron evidentes. Muchas aplicaciones para transferencia de información eran vistas a medida que el uso del ordenador portátil y del propio Internet se fue popularizando. Si bien la tercera generación (3G) estaba en el horizonte, algunos servicios se hicieron necesarios previa a su llegada, por lo que se le denominó a este periodo generación 2.5. GPRS fue desarrollado para el sistema global para las comunicaciones móviles, GSM (del inglés: Global System for Mobile Communication), fue de los primeros en ser visto. Hasta este momento, todos los circuitos eran dedicados en forma exclusiva a cada usuario. Este enfoque es conocido como "Circuit Switched", donde, por ejemplo, un circuito es establecido para cada usuario del sistema. Esto era ineficiente cuando un canal transfería información sólo en un pequeño porcentaje. El nuevo sistema permitía a los usuarios compartir un mismo canal, dirigiendo los paquetes de información del emisor al receptor. Esto permite el uso más eficiente de los canales de comunicación, lo que habilita a las compañías proveedoras de servicios a cobrar menos por ellos (Velázquez Pastrana, 2002).

2.5G provee algunos de los beneficios de 3G (por ejemplo conmutación de datos en paquetes) y puede usar algo de la infraestructura utilizada por 2G en las redes GSM y división por código de acceso múltiple, CDMA (del inglés: Code Division Multiplex Access). La tecnología más comúnmente conocida de 2.5G es GPRS, que provee transferencia de datos a velocidad moderada. Mientras los términos "2G" y "3G" están definidos oficialmente, no lo está "2.5G". Fue inventado con fines únicamente publicitarios. Muchos de los proveedores de servicios de telecomunicaciones se moverán a las redes 2.5G antes de entrar masivamente a la 3G. La tecnología 2.5G es más rápida y más económica para actualizar que la 3G.

La tercera generación promete ser la respuesta a estos problemas planteados al ofrecer servicios de voz, dato y video a altas velocidades, y quizás el don de la ubicuidad en cuanto a comunicaciones móviles (Jiménez, 2006).

### 1.3 Comunicaciones Inalámbricas

Uno de los mayores avances de la computación a nivel mundial vino en 1970 con el desarrollo de Ethernet, una red que fue capaz de conectar a varias máquinas en una red local de forma eficiente. Esta idea fue creciendo, mejores métodos de transmisión aparecieron, aunque todos ellos limitados por la necesidad de una conexión física (cable) para poder comunicarse entre distintos dispositivos. Una nueva forma de transmisión era necesaria, una que eliminara los problemas que acarreaban una reestructuración de un local de trabajo o una aplicación del mismo. Al igual que el telégrafo evolucionó con el devenir del tiempo a teléfonos móviles los cuales eliminaron el cableado tradicional, las redes locales necesitaban un proceso similar, de aquí que surgiera la idea de crear redes locales inalámbricas, WLAN (del inglés: Wireless Local Area Network) partiendo de que el concepto de eliminar la conexión física no era nuevo sino que databa de principios del siglo pasado (Miguel Blázquez, 2006).

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

- Disponibilidad permanente a la información, lo cual se traduce en eficiencia, pues un trabajador podría optimizar su tiempo de trabajo ya que puede acceder a la información desde cualquier lugar dentro de la empresa a cualquier hora.
- Facilita la ampliación de redes existentes en locales, puesto que en la mayoría de los casos resulta más económico que rediseñar toda la estructura ya existente, reduciendo considerablemente tanto la inversión económica como el tiempo de reestructuración.
- Comodidad, facilidad de instalación y mantenimiento.

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

- De corto alcance: es en la que los dispositivos necesitan estar a corta distancia dependiendo de la tecnología que se emplee, en este caso pueden citarse al IrDA y Bluetooth.
- Wi-Fi: redes que pueden llegar a alcanzar longitudes de 350 metros en zonas abiertas y que están destinadas a la creación de redes de trabajo.

Para comunicarse a larga distancia existen otras tecnologías como son GSM, GPRS y el sistema universal para las telecomunicaciones móviles, UMTS (del inglés: Universal Mobile Telecommunications System). Partiendo del hecho que la aplicación a realizar debe estar alojada en un servidor a una gran distancia de los posibles clientes (teléfonos celulares) el abanico de opciones se reduce al uso de las tecnologías que faciliten dicha conexión. A continuación se ampliará en las posibles variantes: GSM, GPRS y UMTS.

### **1.3.1 GSM**

GSM es el estándar creado por el Instituto Europeo de Estándares para las Telecomunicaciones, ETSI (del inglés: European Telecommunication Standard Institute) para los sistemas móviles celulares. La primera fase fue terminada a finales de los 80, sin embargo, no fue hasta el año 1992 que los primeros sistemas comerciales salieron al mercado. Constituye el sistema celular digital más extendido del mundo, su principal ventaja es que está completamente estandarizado, lo que permite la interoperabilidad entre suministradores (Contreras, 2003).

Los sistemas GSM fueron diseñados para transmitir voz, con el devenir del tiempo adquirieron la posibilidad de operar en modo de transferencia de datos. Los terminales operan por conmutación de circuitos, de forma tal que ambos extremos de la conexión necesitan estar conectados para que exista transmisión de información. Esto conlleva largos tiempos de espera puesto que la llamada se mantiene abierta, aun cuando no existe transferencia de datos. Uno de sus principales problemas consiste en el hecho de no soportar el protocolo de internet, IP (del inglés: Internet Protocol), lo que impide el acceso directo a Internet.

Las discusiones sobre el modelo de transferencia de datos para el GSM comenzaron en el período 1992/93, incidiendo los estudios sobre la necesidad de un sistema que fuese basado en la transmisión de datos por paquetes IP. En 1998 el ETSI, la entidad reguladora de las telecomunicaciones europeas, concluyó sus estudios sobre la definición de las normas del nuevo sistema denominado GPRS. Se planteó que el GPRS debería suavizar muchos de los problemas que afectaban la Internet móvil, al permitir una mayor capacidad de transmisión de datos.

La arquitectura abierta de GSM ofrece una compatibilidad superior con sistemas, equipos y aplicaciones desarrolladas alrededor del mundo. Las principales ventajas y beneficios son (Aurora, 2006):

- Seguridad, privacidad y flexibilidad: a través de la tarjeta SIM (del inglés: Subscriber Identity Module) que lleva cada teléfono, el usuario obtiene una mayor flexibilidad en la manera como usa su teléfono sin perder la seguridad y privacidad de sus comunicaciones.
- Innovación constante: los avances más populares e importantes en la comunicación así como en los servicios de valor agregado más exitosos se han dado en las redes GSM.
- Roaming universal: en el futuro los fabricantes de teléfonos construirán unidades que puedan operar en todas las cuatro frecuencias brindándole una verdadera experiencia de roaming universal.
- Terminales: los modelos GSM ofrecen más beneficios, funciones y diseños más atractivos. Estos celulares son más económicos comparados con los TDMA (del inglés: Time Division Multiple Access) y CDMA, gracias a las economías de escala que otorga fabricar para el 70% de los suscriptores de telefonía celular del mundo.

### 1.3.2 GPRS

La tecnología GPRS es capaz de proporcionar una velocidad de transferencia de datos mayor que GSM. Utiliza el concepto de comunicación por paquetes, en vez de la tradicional por circuitos empleada por GSM y comparte su mismo rango de frecuencias (Blanco, 2002).

Entre sus principales características cabe destacar que los canales de comunicación se comparten entre los distintos usuarios dinámicamente, de modo que un usuario sólo tiene asignado un canal cuando se está realmente transmitiendo datos. Para utilizar GPRS se precisa de un teléfono que soporte esta tecnología. La mayoría de estos terminales soportarán también GSM, por lo que podrá realizar sus llamadas de voz utilizando la red GSM de modo habitual y sus llamadas de datos tanto con GSM como con GPRS. Además, se obtiene mayor velocidad y eficiencia de la red, tradicionalmente la transmisión de datos inalámbrica se ha venido realizando utilizando un canal dedicado a GSM a una velocidad máxima de 9.6 Kbps. Con el GPRS no sólo la velocidad de transmisión de datos se ve aumentada en el rango de 40 Kbps hasta un máximo de 115 Kbps por comunicación, sino que además la tecnología utilizada permite compartir cada canal por varios usuarios, mejorando así la eficiencia en la utilización de los recursos de red.

GPRS proporciona al cliente una serie de servicios que la hacen popular, en algunos casos son equivalentes a tener un PC conectado a Internet. Los servicios son:

- Acceder en movilidad a Internet y al correo electrónico. GPRS permite acceder en movilidad a todas las facilidades de Internet usando el terminal GPRS como módem.
- Acceder con facilidad a la intranet corporativa.
- Acceso a cuentas de correo corporativas (intranet).
- Acceso a bases de datos y aplicaciones corporativas desde un dispositivo móvil.
- Acceso GPRS a aplicaciones WAP (del inglés: Wireless Application Protocol) para usos empresariales a través del servicio Protocolo para Aplicaciones Inalámbricas.

- Acceso a servicios de información (a través del servicio WAP).

Las principales ventajas que ofrece GPRS tanto para el cliente como para los proveedores de servicios telefónicos son:

- Siempre conectado: un usuario puede estar conectado todo el tiempo que desee, puesto que no hace uso de recursos de red, mientras no esté transmitiendo datos.
- Tarificación por datos transmitidos, no por tiempo.
- Mayor velocidad de transmisión, pues se pueden tener varios canales asignados para la transmisión.
- Posibilidad de recibir o realizar llamadas mientras se está conectado o utilizando cualquiera de los servicios disponibles.
- Modo de transmisión asimétrico, tendrá cuatro veces mayor capacidad de transmisión de bajada que de subida.
- Uso eficiente de los recursos de red, los usuarios ocupan los recursos de la red en el momento en que están transmitiendo datos y además se pueden compartir los canales de comunicación entre los diferentes usuarios (Movil 10, 2006).

### **1.3.3 UMTS**

UMTS es el sistema de telecomunicaciones móviles de tercera generación, que evoluciona desde GSM pasando por GPRS hasta UMTS. De aquí que UMTS sea una realidad y tenga un papel principal en las telecomunicaciones multimedia inalámbricas de alta calidad (WebmovilGsm, 2006).

UMTS ofrece otras ventajas como son el roaming y cobertura a nivel mundial, tanto vía enlace radio terrestre o vía satélite y es altamente estandarizado como interfaz única de red. Es una tecnología

apropiada para una amplia gama de servicios y usuarios. La misma ofrece facilidad de uso y adaptación para abordar las necesidades y preferencias de los usuarios, un gran número de terminales para facilitar el acceso a los distintos servicios y un bajo costo de los mismos para asegurar un mercado masivo. Otra de las ventajas es la capacidad de soportar altas velocidades de transmisión de datos hasta 144 Kbps sobre vehículos a gran velocidad a diferencia de la segunda generación, 384 Kbps en espacios abiertos de extrarradios y 2 Mbps en interiores de edificios.

### **1.3.4 Conclusiones del epígrafe**

Como se ha planteado con anterioridad, se empleará una arquitectura cliente servidor para el desarrollo de la aplicación, donde la parte servidor estará alojada a largas distancias del cliente lo cual obliga a emplear una tecnología de largo alcance. UTMS es una tecnología muy joven y solo está soportada en dispositivos de tercera generación y puesta en marcha en algunos países desarrollados. Por lo tanto, la tecnología a emplear en el proyecto será GPRS por ser la más difundida actualmente en el mundo y a la que se tiene acceso en el entorno de desarrollo.

## **1.4 Plataformas para juegos en red desde teléfonos celulares. Principales tendencias.**

La gran aceptación que poseen los juegos en red en la actualidad y el crecimiento de la demanda en el mercado de los mismos, ha provocado que importantes empresas informáticas a nivel mundial se hayan proyectado en este sentido. El desarrollo de las tecnologías inalámbricas y de comunicación se ha convertido en un factor determinante, su uso ha propiciado, a los programadores, la posibilidad de desarrollar juegos en red más robustos y acabados, que abarquen mayor número de usuarios.

Mundialmente se han creado paradigmas para estandarizar la creación de juegos multiusuario tales como el Punto-a-Punto, donde se establece una conexión directa entre los dos clientes y no existe un servidor entre ellos. Otro modelo es el Cliente/Servidor, donde una entidad, ya sea un servidor dedicado a esta

actividad, o uno de los clientes participantes en el juego, actúa como servidor y los demás como aplicaciones clientes (Price, 2007).

Existen diversos tipos de juegos multiusuario entre los que se pueden destacar:

- Multi-jugador a solas (del inglés: Multi-player solo-play): Cada jugador se enfrenta a un juego de un solo usuario y se comparan las puntuaciones al final.
- Por turnos (del inglés: Round-robin): Cada jugador toma su turno por orden, de forma que cada uno espera su turno para jugar.
- Movimiento simultáneo (del inglés: Simultaneous movement): También es considerado un juego basado en turnos. Cada jugador hace sus movimientos independientemente de los otros, y envía la orden al servidor.
- Actuar cuando quiera (del inglés: Act-whenever): El juego persiste por un largo período de tiempo, y los jugadores se pueden autenticar por períodos cortos de tiempo para realizar sus acciones en el juego.
- Actualización lenta (del inglés: Slow-update): El juego opera de forma constante en el servidor, y los jugadores pueden entrar para cambiar el comportamiento de su(s) carácter(es).
- Adaptado a la demora (del inglés: Latency-adapted): Juegos en tiempo real, el tiempo de transferencia puede ser ocultado detrás del razonamiento del juego.

Para dar soporte a éstos juegos, actualmente se desarrollan plataformas y servidores que sirven para alojar los mismos, se encargan de controlar el tráfico de información entre los clientes y el manejo de cuentas de usuarios y estadísticas por juego. Existen dos tendencias fundamentales en la producción de estas aplicaciones y que se describen a continuación.

- Servidor para un juego determinado: manejan solo el juego para el que fue diseñado, la mayoría de ellos manejan la lógica del juego y reducen el procesamiento de la aplicación cliente.

- Servidor para múltiples juegos: mantienen el control y alojamiento de éstos sin ser significativo su tipo. Esto trae consigo, que generalmente la lógica del juego se maneje en el cliente, además, almacenan grandes volúmenes de información y numerosas conexiones simultáneas de usuarios.

En Cuba en los últimos años se ha desarrollado un proceso de informatización a distintos niveles de la sociedad, creándose centros e instituciones dedicadas a fomentar la producción de aplicaciones informáticas de valor en el mercado.

En el marco de este proceso surge la entidad Procyon ubicada en la Infraestructura Productiva de la Universidad de las Ciencias Informáticas. Esta entidad se dedica a la elaboración de aplicaciones destinadas a las telecomunicaciones. Actualmente se realiza un amplio trabajo en la producción de juegos para teléfonos celulares. Se han obtenido resultados en el desarrollo de juegos en red, usando tecnología Bluetooth para la comunicación entre 2 jugadores concurrentes a cortas distancias, donde el dispositivo móvil que crea el juego actúa como servidor. Con el objetivo de entrar a competir en el mercado se decidió desarrollar una plataforma que gestione los juegos en red a largas distancias.

### **1.4.1 Descripción de soluciones existentes a nivel mundial**

A nivel mundial otras empresas informáticas han realizado plataformas y servidores que dan soporte a juegos en red desde teléfonos móviles. A continuación se describen algunas que fueron objeto de estudio, resaltando sus principales características y funcionalidades.

#### **Caso de estudio # 1: SNAP (Scalable Network Application Package)**

Es una solución desarrollada por la compañía NOKIA, de Finlandia, que posibilita conectarse a juegos desde móviles a través de una comunidad de servicios. SNAP Mobile's es una tecnología para juegos multiusuario a través del uso de servicios Web (del inglés: Webservices), mensajes instantáneos y presencia de servicios (IMPS por sus siglas en inglés). El SDK cliente para SNAP Mobile's provee

herramientas para crear juegos para móviles que cuenten con el perfil de información del dispositivo móvil, MIDP 2.0 (del inglés: Mobile Information Device Profile) y Configuración de Dispositivo Limitada por la Conectividad, CLDC 1.0 (del inglés: Connected Limited Device Configuration). El SDK (del inglés: Software Development Kit) incluye las librerías, una aplicación para la emulación de las aplicaciones, un API para realizar pruebas y documentación para el desarrollo de los juegos. El ambiente de emulación simula el comportamiento de los clúster de la comunidad de servicios y posibilita el desarrollo, depuración y realización de pruebas antes de la compilación de juegos sin la necesidad de una conexión a un servidor real. Es importante destacar que esta aplicación es de propósito comercial y su futura distribución del API cliente será incluida en las nuevas versiones de los teléfonos Nokia (Nokia, 2005).

### **Caso de estudio # 2: Deus+**

Deus es una plataforma para juegos multiusuario producida y puesta en marcha en el 2003 por la compañía Exmachina, que radica en Ámsterdam, Holanda. Basada en la arquitectura empresarial de Java, Deus+ es capaz de manejar miles de jugadores de forma simultánea en un solo servidor, es escalable a cualquier capacidad necesaria. La plataforma soporta ancho de banda de 9600 bps y negocia con redes de alta velocidad. Se integra fácilmente con sistemas de abastecimiento de pago por paquetes, pago por minutos y modelos de suscripción (Ex Machina, 2003).

### **Caso de estudio # 3: Neutrón**

Neutrón es una plataforma para el desarrollo de aplicaciones multiusuario producida por ExitGames, en San Francisco, Estados Unidos, actualmente se encuentra desplegada y usada a nivel global y sirve de alojamiento de una gran cantidad de juegos multiusuario para teléfonos celulares.

La plataforma ofrece a los clientes una gran cantidad de prestaciones y servicios. Facilidad de desarrollo de juegos multiusuario, cubre todas las conexiones con el servidor, logrando así que los desarrolladores solo se centren en el juego. Soporta grandes cantidades de cuentas de usuarios y alberga complejas

comunidades de aplicaciones, Neutrón provee soporte para múltiples tipos de juegos. Poseen interfaces directamente con los operadores de los sistemas finales para habilitar las suscripciones así como el cobro de los servicios (Exit Games, 2006).

### **Caso de estudio # 4: Gex**

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 editores.

Gex provee importantes beneficios al cliente permitiéndole a los proveedores distribuir los juegos multiusuario en los dispositivos clientes y habilitando los juegos multi-jugador en línea a través de múltiples tipos de dispositivos móviles. La plataforma Gex posibilita a los usuarios registrarse en los juegos, obtener nombres de usuarios, subir sus puntuaciones, participar en torneos y recibir estadísticas de juegos.

Posibilita de manera similar, a los usuarios finales, buscar, descargar y pagar por juegos y otros contenidos a través de la infraestructura Gex. Los juegos pueden ser evaluados y revisados para obtener futuras versiones que mejoren la experiencia del usuario y su satisfacción.

Los jugadores poseen redes de amigos y oponentes con los que se divierten jugando. La infraestructura Gex posibilita el mantenimiento de estas relaciones a través de las comunicaciones entre ellos, y el arreglo de torneos, haciendo posible presentar una serie de contenidos o campeonatos que pueden servir para atraer la actividad de los suscriptores (Game Federation, 2005).

### **Caso de estudio # 5: Gex**

La plataforma patentada JIVE™ de Pixel Play, producto de una asociación entre empresas de Nueva York y Gran Bretaña, ha sido actualizada a través de un proceso de generación de versiones que ha servido para proveer una estabilidad y mejoramiento continuo en los servicios que oferta. Permite a los usuarios

jugar desde Web y dispositivos móviles, posibilita actualización de contenidos y transmisión del mismo, tiene muchas facilidades de compras y permite la administración de torneos. Mantiene control sobre las cuentas de usuarios, así como la administración de canales, controla el acceso a la plataforma e integra múltiples servicios. Posee un sistema de configuración, monitoreo y reportes (remotos y desde el sitio). Actualmente se encuentra desplegado en más de 15 operadores a nivel mundial (PixelPlay, 2006).

### **Caso de estudio # 6: Unipier Entertainment Suite**

Unipier Entertainment Suite desarrollada por la empresa Unipier de Netanya, Israel, provee a los operadores una rápida entrega de avances renovados de última generación en cuanto a servicios de juegos hospedados en un modelo ASP (del inglés: Active Server Page). Este modelo proporciona a los operadores de móviles una solución integrada, de bajo costo, para el usuario final con reducido costo de acceso y en la mayoría de los casos, con un tiempo extremadamente pequeño de configuración.

Como principales características de Unipier Entertainment Suite se puede decir que brinda una solución integral de juegos para servicios de mensajes cortos, SMS (del inglés: Short Message Service), servicios de mensaje multimedia, MMS (del inglés: Multimedia Mobile Service), WAP, java para ediciones pequeñas, J2ME (del inglés: Java 2 Micro Edition), juegos de un solo usuario y multiusuario. Es considerado un mobiliario centralizado y adaptable de entretenimientos con fácil integración a plataformas externas para juegos, con habilidad para crear comunidades combinando perfiles de administración de usuarios con avanzados servicios de canales, grupos de usuarios y listas de amigos. Posee capacidades mejoradas para la administración de torneos y herramientas complementarias para administración de usuarios y reportes (Unipier, 2005).

### **1.4.2 Conclusiones del epígrafe**

Es importante destacar que el propósito de la empresa es entrar en el mercado y competir con las plataformas descritas anteriormente. La investigación realizada sirvió para obtener una amplia visión de

las principales tendencias, prestaciones y servicios que ofrecen los servidores para juegos en red desde teléfonos celulares y hacia ese sentido se encaminó el desarrollo de la solución propuesta.

Con este trabajo, se pretende desarrollar una aplicación que funcione como servidor común para múltiples juegos, gestionando la concurrencia de los usuarios, dejando a los desarrolladores la implementación de la lógica del juego, en la que también se brinda una librería de clases para facilitar a los desarrolladores la comunicación con el servidor.

### **1.5 Arquitectura Cliente Servidor**

La arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que, definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información. Se establece así un enlace de comunicación transparente entre los elementos que conforman la estructura. Esta arquitectura tiene entre sus características que el servidor presenta a todos sus clientes una interfaz única y bien definida y éste 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. Los cambios en el servidor implican pocos o ningún cambio en el cliente.

### **1.6 Metodología de desarrollo empleada**

En la actualidad el uso de metodologías es un factor importante en el desarrollo de sistemas informáticos. El principal propósito es contar con un marco de trabajo claramente definido y estandarizado, que permita obtener productos que garanticen los requerimientos de calidad, que cumplan con las expectativas del cliente y se desarrollen en el tiempo estimado y bajo los costos presupuestados. Para la realización de la plataforma se decidió el uso del proceso unificado de desarrollo de software, RUP (del inglés: Rational Unified Process).

### **1.6.1 Proceso Unificado de desarrollo de Software**

RUP 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. Dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental. El proceso de ciclo de vida de RUP se divide en cuatro fases. Esas fases se dividen en iteraciones, cada una de las cuales produce una pieza de software demostrable (Pressman, 2001).

### **1.7 Plataforma de desarrollo y lenguaje de programación empleados**

A pesar del aumento que han experimentado las capacidades de los teléfonos celulares, aún son efímeras, de aquí que no se pueda programar en los lenguajes convencionales sino, en aquellos que faciliten un grupo de bibliotecas o paquetes de clases reducidos, que sean capaces de ejecutarse en dichos dispositivos.

Para el desarrollo de la aplicación y atendiendo a que la misma está compuesta por 2 partes o módulos principales se decidió el uso de la plataforma J2EE (del inglés: Java 2 Enterprise Edition) para la elaboración del servidor, J2ME para el API que se distribuirá en los clientes. El desarrollo de ambas partes se realizará sobre el lenguaje de programación Java.

#### **1.7.1 ¿Por qué J2ME?**

J2ME provee un robusto y flexible ambiente para el desarrollo de aplicaciones que serán ejecutadas en teléfonos celulares y otros dispositivos móviles como los asistentes personales digitales, PDA (del inglés: Personal Digital Assistant). Incluye interfaces flexibles, seguridad robusta, protocolos de redes incorporados y soporte para redes y aplicaciones que pueden ser descargadas dinámicamente, forma parte en dispositivos donde la plataforma base es .NET Compact Framework de la Microsoft. J2ME es una

colección de especificaciones desarrolladas por la comunidad de procesos de Java, JCP (el inglés Java Community Process) que define cuáles de estas especificaciones puede contener un dispositivo en específico. El subconjunto de especificaciones para un dispositivo en particular es definido por uno o más perfiles y hereda las funcionalidades básicas de una configuración (Pradilla Cerón, 2007).

- Una configuración es una clasificación que se define para un rango de dispositivos diseñados con un conjunto de características tales como: tipo y cantidad de memoria, velocidad del procesador y tipo de conexión de red que soporta el dispositivo. Actualmente existen dos tipos de configuraciones: CLDC diseñado para dispositivos pequeños con bajos recursos y configuración de dispositivos conectados, CDC (del inglés: Connected Device Configuration) diseñado para dispositivos pequeños con altos recursos.
- Un perfil complementa una configuración adicionando clases opcionales que proveen características apropiadas a un dispositivo en particular. El perfil más común es MIDP para dispositivos con configuración CLDC. Este perfil maneja el trabajo con la red, tratamiento con la interfaz de usuario y el almacenamiento de datos locales en el sistema de administración de registros, RMS (del inglés: Record Manager System).

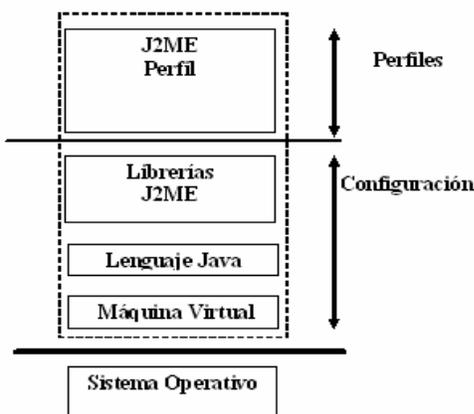


Figura # 2. Estructura de la plataforma j2me en un teléfono celular.

J2ME 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. Por política de la empresa las aplicaciones para teléfonos celulares se desarrollan sobre esta plataforma, abarcando un mayor número de clientes.

### 1.7.2 ¿Por qué J2EE?

J2EE define un estándar para el desarrollo de aplicaciones empresariales multi-capa diseñado por Sun Microsystems. J2EE simplifica las aplicaciones empresariales basándolas en componentes modulares y estandarizados, proveyendo un completo conjunto de servicios a estos componentes, y manejando muchas de las funciones de la aplicación de forma automática, sin necesidad de una programación compleja (Molpeceres y otros, 2002).

J2EE está basada en el lenguaje de programación Java y la concepción de sistema de información adquiere nuevas dimensiones de portabilidad y confiabilidad. El lenguaje extensible de etiquetas, XML (del inglés: Extensible Markup Language) es una poderosa tecnología para procesamiento y almacenamiento de datos, así como para mensajería y configuración de componentes. Usando XML los datos son totalmente portables, y con su adecuado empleo en aplicaciones J2EE se pueden construir sistemas de software robustos y flexibles (Armstrong, 2005).

Los Servlets y las páginas servidoras Java, JSP (del inglés: Java Server Pages) son el pilar de funcionamiento de cualquier sistema que pretenda prestar servicios Web, estos permiten el desarrollo de aplicaciones para las cuales los clientes no requieren nada más allá de un navegador para interactuar con ellas.

Es totalmente libre y puede ser descargada gratuitamente, además, en la actualidad existen muchas herramientas de código abierto disponible para extender la plataforma o para simplificar el desarrollo.

Es importante destacar que el uso de esta plataforma está muy ligada a las aplicaciones para teléfonos celulares, mundialmente es la más usada en este rama. El equipo de desarrollo posee experiencia en su uso, además es una política establecida en la institución.

### 1.8 JSP y Servlet

Un **servlet** es un objeto que se ejecuta en un servidor o contenedor J2EE, fue especialmente diseñado para ofrecer contenido dinámico desde un servidor Web, generalmente el protocolo de transferencia de hipertexto, HTTP (del inglés: Hypertext Transfer Protocol).

Los servlets forman parte de J2EE, que es una ampliación de la Edición Estándar de Java, J2SE (del inglés: Java 2 Standard Edition). Un servlet es un objeto Java que implementa la interfaz `javax.servlet.Servlet` o hereda alguna de las clases más convenientes para un protocolo específico (ejemplo: `javax.servlet.HttpServlet`). Al implementar esta interfaz el servlet es capaz de interpretar los objetos de tipo `HttpServletRequest` y `HttpServletResponse` quienes contienen la información de la página que invocó al servlet.

Entre el servidor de aplicaciones (o contenedor Web) y el servlet existe un contrato que determina cómo han de interactuar. La especificación de éste se encuentra en los JSR (del inglés: Java Specification Requests) del JCP.

Los **JSP** constituyen una tecnología que permite a los programadores generar contenido dinámico para Web, en forma de documentos que usen el protocolo HTML, XML o de otro tipo. Las JSPs permiten al código Java y a algunas acciones predefinidas ser incrustadas en el contenido estático del documento Web.

### **1.9 Servidor Web utilizado**

Un servidor Web es un programa que implementa el protocolo 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. Un servidor Web se encarga de mantenerse en espera de las peticiones HTTP llevadas a cabo por un cliente. Este realiza una petición al servidor, quien le responde con el contenido solicitado.

Para el despliegue y alojamiento de la plataforma 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 (Apache Tomcat, 2005).

Apache Tomcat brinda soporte para servlets y JSPs. Incluye herramientas internas que compilan los JSPs convirtiéndolos en servlets, además, su uso está estandarizado mundialmente por parte de muchos entornos de desarrollo, tanto libres como comerciales.

### **1.10 Herramientas de desarrollo empleadas**

El elevado número de prestaciones que ofrecen las herramientas de desarrollo en la actualidad posibilitan obtener un producto de mayor calidad. Proporcionan un alto grado de personalización, lo que da lugar a un conjunto mayor de soluciones escalables. Las avanzadas funcionalidades de depuración de códigos permiten desarrollar sistemas informáticos más acabados y en menores tiempos de desarrollo.

#### **1.10.1 Herramientas de modelado UML**

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, pues su uso está muy estandarizado a nivel

mundial y constituye una herramienta muy madura y acabada, además, el grupo de desarrollo se encuentra familiarizado con el uso de esta herramienta.

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

Recomienda el uso de vistas dinámicas y estáticas de los modelos lógicos y físicos para capturar los productos en proceso que sean orientados a objetos de los distintos flujos descritos por RUP. A través del uso de la notación del lenguaje unificado de modelado, UML (del inglés, Unified Modeling Language), la aplicación permite la creación y refinamiento de estas vistas dentro del modelo general del dominio de la aplicación y el 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. Estos componentes poseen una representación visual a través de iconos, con una serie de atributos predefinidos, dependiendo del tipo de modelo, diagrama u artefacto que se esté construyendo (Thoth, 2003).

Para soportar la idea de los analistas, arquitectos y desarrolladores de software, el Rational Rose provee:

- Desarrollo en paralelo de los modelos, brindando la posibilidad de descomponer los modelos en unidades más pequeños, denominados unidades controladas.
- Posibilita al equipo administrar sus modelos en consecuencia con artefactos del proyecto a través de la integración con sistemas de control de versiones.

### **1.10.2 Entorno Integrado de desarrollo**

Los entornos integrados de desarrollo, IDE (del inglés: Integrated Development Environment) constituyen programas compuestos por un conjunto de herramientas orientadas a programadores. Proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación. Es posible que un mismo IDE

pueda funcionar con varios lenguajes de programación. Este es el caso del Eclipse, que mediante plugins se le puede añadir soporte para lenguajes adicionales.

Para el desarrollo de la aplicación se decidió usar Eclipse, debido a las múltiples funcionalidades que ofrece, además, el equipo de desarrollo se encuentra familiarizado con su uso.

### **Eclipse**

Eclipse es una plataforma abierta para el desarrollo de aplicaciones que corren sobre un amplio rango de sistemas operativos. Eclipse brinda la facilidad de asociar disímiles editores para cada tipo de ficheros además de posibilitar editarlos con los programas asociados por el sistema operativo. 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, lo cual optimiza el tiempo de desarrollo. Eclipse permite la integración de plugins, que no son más que agregados que enriquecen la plataforma. Algunos plugins hacen posible que Eclipse sirva para programar en distintos lenguajes permitiéndole al desarrollador lograr un entorno personalizado.

Eclipse ME es el plugin para el desarrollo de aplicaciones en J2ME. Este plugin 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.

El plugin Tomcat se diseñó para el desarrollo de aplicaciones Web y el despliegue hacia un servidor específico. Es una herramienta muy útil dado que facilita la creación de los Servlets y JSP que componen una aplicación Web (Eclipse Foundation, 2000).

### **1.10.3 Herramienta para el control de versiones**

El control de versiones es el proceso mediante el cual se pueden gestionar todas las versiones de los elementos de configuración que forman la línea base de un producto o configuración del mismo. Los

sistemas de control de versiones facilitan la administración de los cambios que se realizan sobre los componentes.

El control de versiones es utilizado principalmente en la industria de la informática para el control de las versiones por las que transita el código fuente de una aplicación, aunque el concepto es aplicable a otras partes del software y del mundo informático, como documentos, imágenes, entre otras (Izquierdo, y otros, 2005).

Este proceso puede ser desarrollado de forma manual, aunque en la actualidad existen varias herramientas que facilitan el proceso.

Un sistema de control de versiones proporciona:

- Mecanismo de almacenamiento de cada objeto bajo control de versiones.
- Posibilidad de edición para cada objeto bajo control.
- 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.

Es aconsejable la generación de informes que almacenen los cambios producidos entre versiones, informes de estado, especificando un nombre que identifique a la persona que realizó los cambios.

Para el desarrollo de aplicación se decidió el uso de SVN, herramienta capaz de automatizar todas las funcionalidades relacionadas con el control de versiones, es de libre acceso y el grupo de desarrollo cuenta con experiencia en su uso.

### **SVN**

Software diseñado para mantener el control de versiones sobre los sistemas. Una característica importante de Subversion es que todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo (Tigris, 2006).

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.
- Permite la creación de ramas y etiquetas.
- 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.

### **1.11 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 de éstas sus principales características. Se abordaron temas relacionados con los juegos en red y tecnologías usadas en el desarrollo de los mismos, 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 de la aplicación, 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 de llevar a cabo para su solución y que tributan a que se realice una propuesta del sistema.

## 2.2 Flujo de los procesos involucrados en el campo de acción

Los juegos que se han creado en la entidad han sido para un solo usuario y solo algunos multiusuario que utilizan la tecnología bluetooth para comunicarse entre ellos. Para lograr una comunicación es necesario estar en un rango determinado de distancia del celular que funcionará como servidor, por lo que de ésta forma disminuye el número de usuarios a alcanzar. La idea principal es desarrollar una aplicación que esté disponible en el proveedor y permita a los usuarios conectarse desde largas distancias a través de la red GPRS.

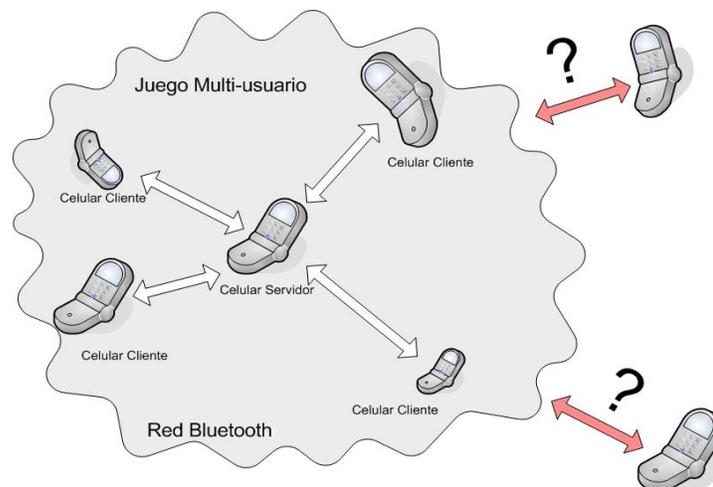


Figura # 3. Alcance de la red bluetooth

### **2.3 Análisis crítico de soluciones existentes a nivel mundial**

A nivel internacional otras empresas informáticas han desarrollado servidores y plataformas para juegos en red desde teléfonos celulares.

Nokia desarrolló el SNAP Mobile's, desarrolla un API para incluir en las aplicaciones clientes y un servidor que se encarga de gestionar todas las conexiones desde distintos clientes a disímiles juegos. Dicha API viene incluida solamente dentro de los celulares de última generación y el servidor está atado a aquellos operadores que lo hayan adquirido a través de Nokia directamente, por lo que reduce significativamente el número de usuarios posibles a alcanzar.

Las demás plataformas encontradas como parte del estudio del arte de la actual aplicación: Unipier, Gex, Deus, Neutrón y Jive, ofrecen de manera similar al Snap Mobile's, facilidades de administración para el servidor, para la manipulación de cuentas, soporte para múltiples juegos, integración con otras plataformas, así como la administración de torneos y otras características referentes a comodidad, facilidad de instalación y mantenimiento.

Es importante destacar que todas las soluciones encontradas son de propósito comercial y que con el presente trabajo, la entidad Procyon, entraría a competir con las soluciones existentes con el objetivo de lograr ingresos para nuestro país.

### **2.4 Objeto de automatización**

La entidad Procyon necesita desarrollar una plataforma servidora que gestione, de forma eficiente, las peticiones de los clientes, que garantice la organización de los usuarios conectados por canales correspondientes al juego desde el cual se encuentran autenticados en el sistema. Permitir que toda la información referente a las estadísticas de los usuarios en los distintos juegos sea persistente y esté disponible para todos.

Facilitar a los desarrolladores de juegos cliente-servidor un API que les permita abstraerse de la comunicación entre ambas partes de forma tal que puedan centrarse solamente en el desarrollo de la lógica del juego. Los desarrolladores son capaces de interactuar con el servidor a través de acciones predefinidas en el API, que incluyen toda la gama de posibles actividades a desarrollar por los juegos que se implementen.

### 2.5 Propuesta del Sistema

Para la realización de la aplicación se ha decidido dividir el proceso en los siguientes componentes:

- 1- Cliente API:** Para el desarrollo de las aplicaciones cliente se brinda un API de conexión, con la cual todas las aplicaciones clientes se comunican en un lenguaje común, que el servidor es capaz de comprender. En dicha API se encuentran agrupadas y definidas las posibles acciones que pueden ser enviadas al servidor. De manera similar, se brinda un tratamiento adecuado a las respuestas satisfactorias e insatisfactorias enviadas por el servidor.
- 2- Servidor Web:** Es la aplicación encargada de gestionar las peticiones de las aplicaciones y de generar la respuesta correcta a cada acción enviada. Es capaz de manejar la concurrencia, seguridad y persistencia de los datos en el servidor. Cuenta con servicios que se ejecutan en un tiempo definido por el desarrollador de las aplicaciones para controlar las pérdidas de conexión de algún usuario con el servidor. Periódicamente realiza comprobaciones para eliminar canales, juegos y otros recursos que no están en uso.
- 3- 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, 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 a los administradores todos los datos referentes al estado del servidor

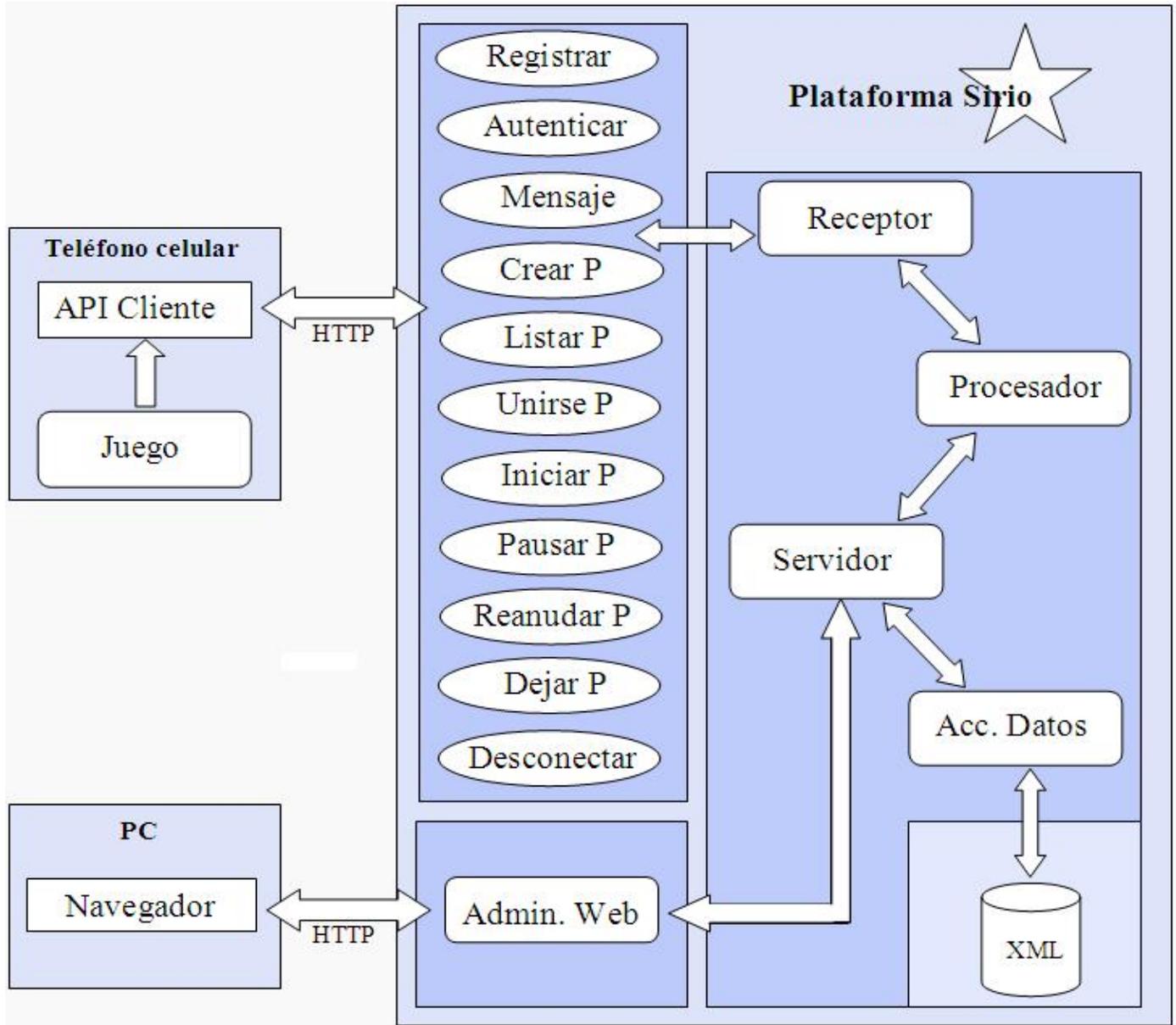


Figura # 4. Propuesta del Sistema.

### **2.6 Modelo de negocio**

Un sistema, por pequeño que sea, generalmente es complicado. Por eso se necesita dividirlo en piezas si se pretende comprenderlo y gestionar su complejidad. Esas piezas se pueden representar a través de modelos que permitan abstraer sus características esenciales. Entre estos se encuentra el modelo de negocio en el cual se refuerza la idea de que sea el propio negocio lo que determine los requisitos.

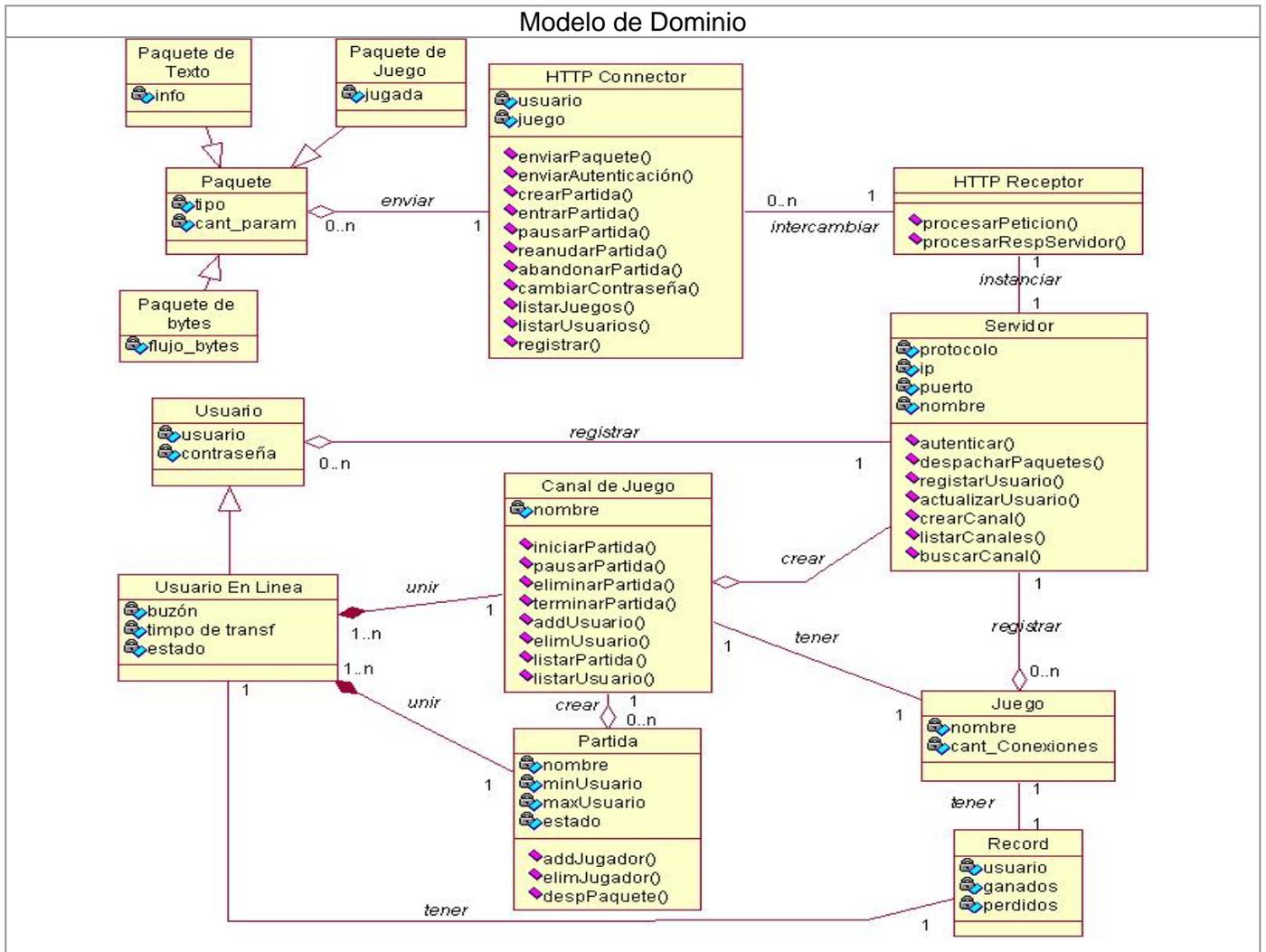
Dependiendo de la situación o escenario que se presente, hay varias alternativas de desarrollar este proceso, no siempre es necesaria o posible, la completa realización de un modelo de negocio. Si se determina que no es necesario un modelo completo del negocio se realizará lo que se conoce como un modelo del dominio.

#### **2.6.1 Modelo de dominio**

El modelo del dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema. El modelo del dominio se considera un subconjunto del llamado modelo de objetos del negocio.

Durante el desarrollo de la aplicación se detectaron problemas en el origen de los flujos de informaciones, resultando en algunos casos difuso. Se hizo difícil determinar los elementos más importantes del sistema y sus interconexiones, así como el establecimiento de las reglas de funcionamiento por lo que se hizo necesario un modelado del domino, mostrado en la tabla 1.

Tabla 1. Modelo de dominio



2.6.1.1 Descripción de los conceptos del dominio

Paquete: Define el concepto general para englobar Mensajes de Texto y Paquetes de Juego.

Paquete de Texto: Define información en texto plano que se envía entre usuarios conectados.

**Paquete de Juego:** Define información relacionado con algún cambio en un juego que se envía al servidor durante una partida.

**Paquete de Byte:** Define información relacionada con algún objeto enviado por el cliente (imagen, video, fichero).

**HTTPConector:** Define al responsable de manejar toda el proceso de envío y recepción de información de cualquier índole con el servidor.

**HTTPReceptor:** Define al responsable de recibir las peticiones de los clientes.

**Servidor:** Define al 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 Registrado:** Define al usuario que posee una cuenta en el servidor.

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

**Usuario Conectado:** Define al usuario que ha pasado por el proceso de autenticación y sus credenciales son correctas.

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

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

**Juego Estadística:** Define al responsable de hacer persistente la información relacionada con los juegos efectuados por un cliente.

### 2.7 Requerimientos

Los requerimientos constituyen la condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo y que debe ser alcanzada o poseída por un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente (Jacobson, y otros, 2000).

Para el desarrollo de la aplicación se llevó a cabo un profundo análisis de los principales conceptos y temas de importancia relacionada con su funcionamiento, el proceso de captura de requisitos arrojó un listado de las principales funcionalidades que debe cumplir el sistema, así como todos los aspectos relacionados con su funcionamiento. Para la identificación de los requisitos se tuvieron en cuenta las necesidades de los clientes, la empresa y las posibilidades reales del grupo de desarrollo.

A continuación se describen los requerimientos funcionales y no funcionales identificados, empleándose un nombre corto y una breve explicación en cada caso.

#### 2.7.1 Requerimientos funcionales

**RF1- Gestionar Usuario:** Define todas las funcionalidades relacionados con el control de las cuentas de usuarios.

**RF.1.1- Autenticar Usuario:** Define el mecanismo para controlar el acceso al sistema.

**RF.1.2- Registrar Usuario:** Define la posibilidad de crear una nueva cuenta de usuario en el servidor

**RF.1.3- Cambiar Contraseña:** Define la posibilidad de actualizar los datos referentes a los usuarios.

**RF2- Enviar Mensaje de texto:** Define el intercambio de mensajes entre los clientes conectados al servidor.

**RF3- Enviar Mensaje del servidor:** Define un mensaje enviado por el administrador del sistema a uno o varios usuarios.

**RF4- Enviar paquete de juego:** Define el envío de información relacionada con algún cambio en una partida.

**RF5- Gestionar Partida:** Define todas las funcionalidades relacionadas con la gestión de las partidas.

**RF.5.1- Crear Partida:** Define la posibilidad que tiene un cliente de crear una nueva partida antes coordinada.

**RF.5.2- Unirse Partida:** Define la posibilidad que tiene un cliente de unirse a una partida ya creada.

**RF.5.3- Iniciar Partida:** Define la posibilidad que tiene un cliente de iniciar una partida.

**RF.5.4- Pausar Partida:** Define la posibilidad que tiene un cliente de pausar una partida.

**RF.5.5- Reanudar Partida:** Define la posibilidad que tiene un cliente de reanudar una partida.

**RF.5.6- Abandonar Partida:** Define la posibilidad que tiene un cliente de abandonar una partida.

**RF6- Gestionar Reportes:** Define todas las funcionalidades relacionadas con la gestión de informes y reportes en el servidor.

**RF.6.1- Listar usuarios conectador por canal:** Define la posibilidad que tienen los clientes de listar todos los usuarios conectados al servidor.

**RF.6.2- Listar juegos disponibles por canal:** Define la posibilidad que tienen los clientes de listar todos los juegos disponibles por canal en el servidor en un momento determinado.

**RF.6.3- Listar puntuaciones:** Define la posibilidad que tienen los clientes de listar las mejores puntuaciones relacionadas con un juego determinando.

**RF.6.4- Listar juegos creados en el servidor:** Define la posibilidad que tiene el administrador del sistema de listar todos los juegos creados en el servidor.

**RF.6.5- Listar usuarios conectados al servidor:** Define la posibilidad que tiene el administrador del sistema de listar todos los usuarios conectados al servidor en un momento determinado.

**RF.6.6- Listar canales habilitados en el servidor:** Define la posibilidad que tiene el administrador del sistema de listar todos los canales habilitados en el servidor en un momento determinado.

**RF7- Enviar resultado de partida:** Define todas las funcionalidades relacionadas con el envío del resultado de las partidas.

### 2.7.2 Requerimientos no funcionales

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

- **Usabilidad:** La aplicación está orientada a desarrolladores de aplicaciones para móviles que deben tener un conocimiento básico en la materia. Las características del mismo se encuentran bien definidas y documentadas por lo que su utilización es sencilla.
- **Rendimiento:** En toda aplicación que se desarrolle siguiendo la arquitectura Cliente / Servidor es de vital importancia la tasa de transferencia entre ambos extremos. Por las características del sistema es necesario definir el tiempo límite de demora en la transferencia de datos entre el cliente y el servidor, esto está sujeto a las peculiaridades de cada juego, así como la distancia entre ambos y la tecnología utilizada en cada caso.
- **Soporte:** Se desarrollará una documentación apropiada que describa todas las funcionalidades del sistema desarrollado así como una guía para su uso.

- **Portabilidad:** El sistema será desarrollado en una plataforma de desarrollo que permitirá a la misma que sea usada en casi toda la totalidad de dispositivos móviles.
- **Seguridad:** Por la importancia que se le atribuye a este aspecto le dedicamos especial atención a la misma. Dado que el sistema mantiene constante intercambio de información con múltiples clientes simultáneos es imprescindible que sólo los clientes registrados en el sistema tengan acceso al mismo, para ello es obligatoria la autenticación. Cuando un usuario se registra en el servidor, los datos proporcionados son almacenados, para brindarle seguridad a los mismos, de forma tal que no puedan ser leídos por nadie más, éstos son codificados usando el Algoritmo de Resumen del Mensaje 5, MD5 (del inglés: Message-Digest Algorithm 5). La demanda de información puede alcanzar niveles elevados por lo que la veracidad de la misma es indispensable para el funcionamiento del sistema y tiene que estar disponible para todos los clientes en el momento solicitado.
- **Software:** Debe utilizarse un contenedor Web que tenga soporte para servlets versión 2.4 y JSP versión 2.0.
- **Hardware:** Debe poseer los requerimientos mínimos especificados por el contenedor Web seleccionado.

Requerimientos no funcionales para el juego cliente.

- **Software:** Debe ser desarrollado usando J2ME, los dispositivos para los que desarrollarán los juegos deben poseer configuración CLDC 1.0 y Profile MIDP 2.0 como mínimo. La longitud máxima de los parámetros a enviar a través de HTTP al servidor no podrá exceder los 255 caracteres.
- **Hardware:** Los dispositivos para los que se desarrollen los juegos deben ser de generación 2.5 para que brinden soporte para comunicaciones vía GPRS.

### 2.8 Modelo del sistema. Definición de casos de uso

Los casos de uso son una técnica para especificar el comportamiento de un sistema, es una secuencia de interacciones entre un sistema y alguien o algo que usa alguno de sus servicios. El modelo de casos de uso describe lo que hace el sistema para cada tipo de usuario.

#### 2.8.1: Definición de actores

Los actores del sistema constituyen personas u otros sistemas que eran 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.

Durante el desarrollo de la plataforma se definió un grupo de actores que en un momento determinado desencadenarán un grupo de acciones en el sistema, los mismos se describen a continuación:

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

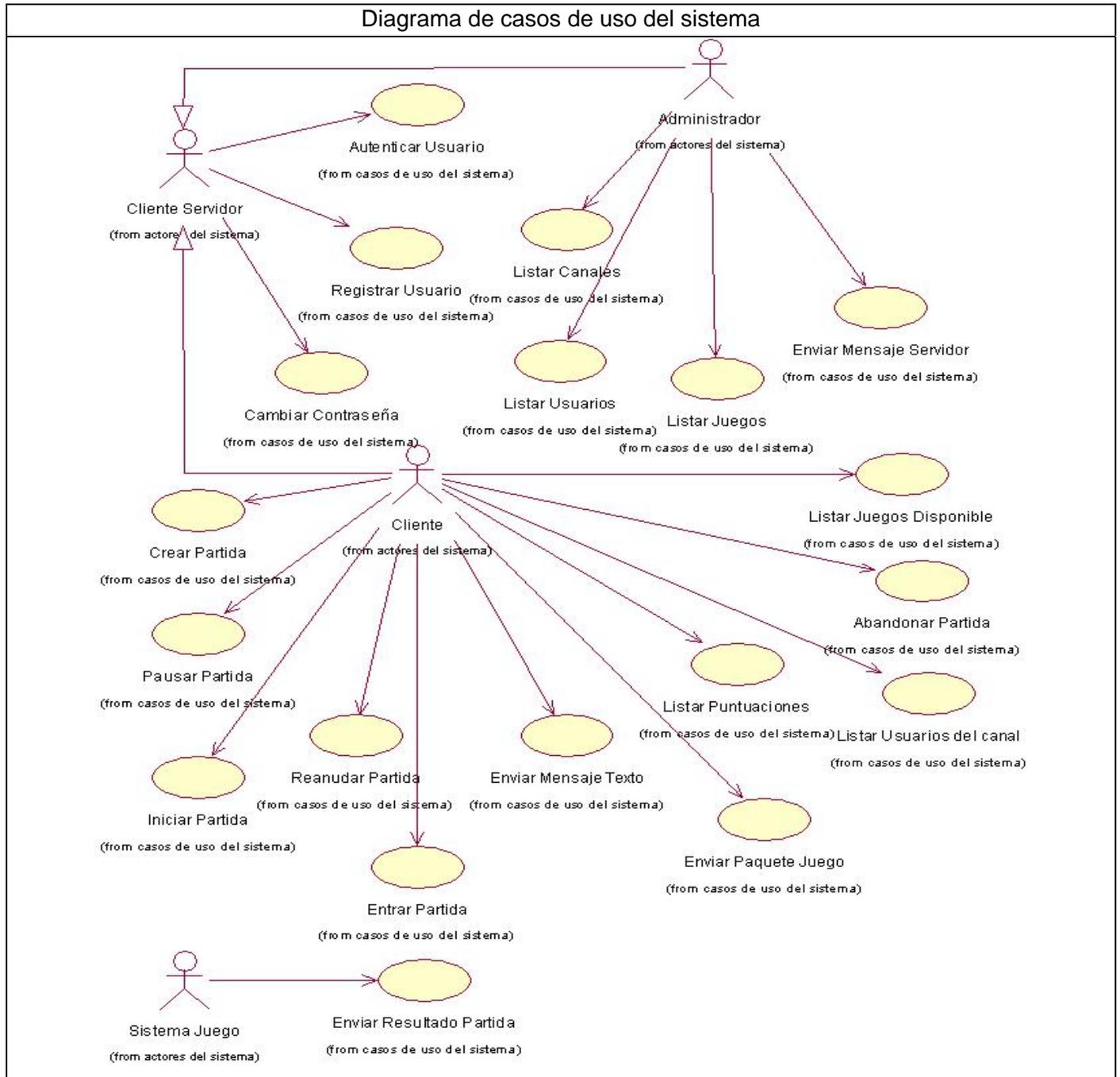
<b>Actores</b>	<b>Justificación</b>
Cliente	Engloba los actores que tienen funcionalidades comunes, relacionadas con la gestión de los usuarios.
Cliente _ jugador	Define al cliente final de los posibles juegos a desarrollar que interactúan con el servidor.
Administrador	Define al actor que engloba las acciones relacionadas con el administrador Web del sistema.
Sistema _ juego	Define al sistema cliente que interactuará con el servidor y que desencadena acciones bajo ciertas condiciones.

### **2.8.2 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 de la plataforma se definieron un conjunto de acciones que deben ser ejecutadas por distintos actores y que desencadenan un conjunto de operaciones. Las interrelaciones entre las acciones y los actores de la aplicación son agrupadas en el diagrama de casos de uso del sistema que se muestra a continuación.

Tabla 3. Diagrama de casos de usos del sistema



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

Para entender la funcionalidad asociada a cada caso de uso no es suficiente con la representación gráfica del diagrama de casos de uso. Se debe elaborar una descripción que capte las principales características de los mismos. La descripción puede ser elaborada de forma breve o extendida. A continuación se muestra la descripción del caso de uso **cambiar contraseña de usuario**.

**Tabla 4. Descripción detallada del caso de uso: Cambiar contraseña de usuario.**

<b>Nombre del Caso de Uso</b>		Cambiar contraseña de usuario
<b>Actores</b>	Cliente	
<b>Propósito</b>	Modificar contraseña de un usuario que esté autenticado en el servidor.	
<b>Resumen</b>	El caso de uso se inicia cuando un usuario autenticado decide modificar su contraseña, para ello se le envía al sistema el juego en el que está autenticado, el usuario y la nueva contraseña.	
<b>Referencias</b>	RF. 1.3	
<b>Precondiciones</b>	El usuario debe estar previamente autenticado.	
<b>Poscondiciones</b>	Contraseña modificada en el servidor.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>		<b>Respuesta del Sistema</b>
1- El cliente solicita al servidor modificar su contraseña, proporcionando para ello los datos necesarios.		1.1- El sistema recibe la petición y pasa a validar los datos recibidos.
		1.2- Si no existen anomalías en los datos del cliente se procede a ejecutar la petición.
		1.3- El sistema notifica al cliente el éxito de la acción.
<b>Puntos de Extensión:</b>		
<b>Prioridad:</b>	Secundaria	

### 2.9 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 las soluciones existentes a nivel mundial, así como 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 y capacidades que debe cumplir el sistema y se definió una propuesta del sistema.

# CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

## 3.1 Introducción

En el presente capítulo se describen los temas referentes al análisis y diseño 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.

## 3.2 Modelo de análisis

El modelo de análisis constituye un modelo que se utiliza para obtener una visión del sistema sobre los requisitos funcionales, expresados en un lenguaje técnico, es el resultado de la actividad de analizar los casos de uso. El modelo de análisis constituye la primera aproximación al modelo de diseño.

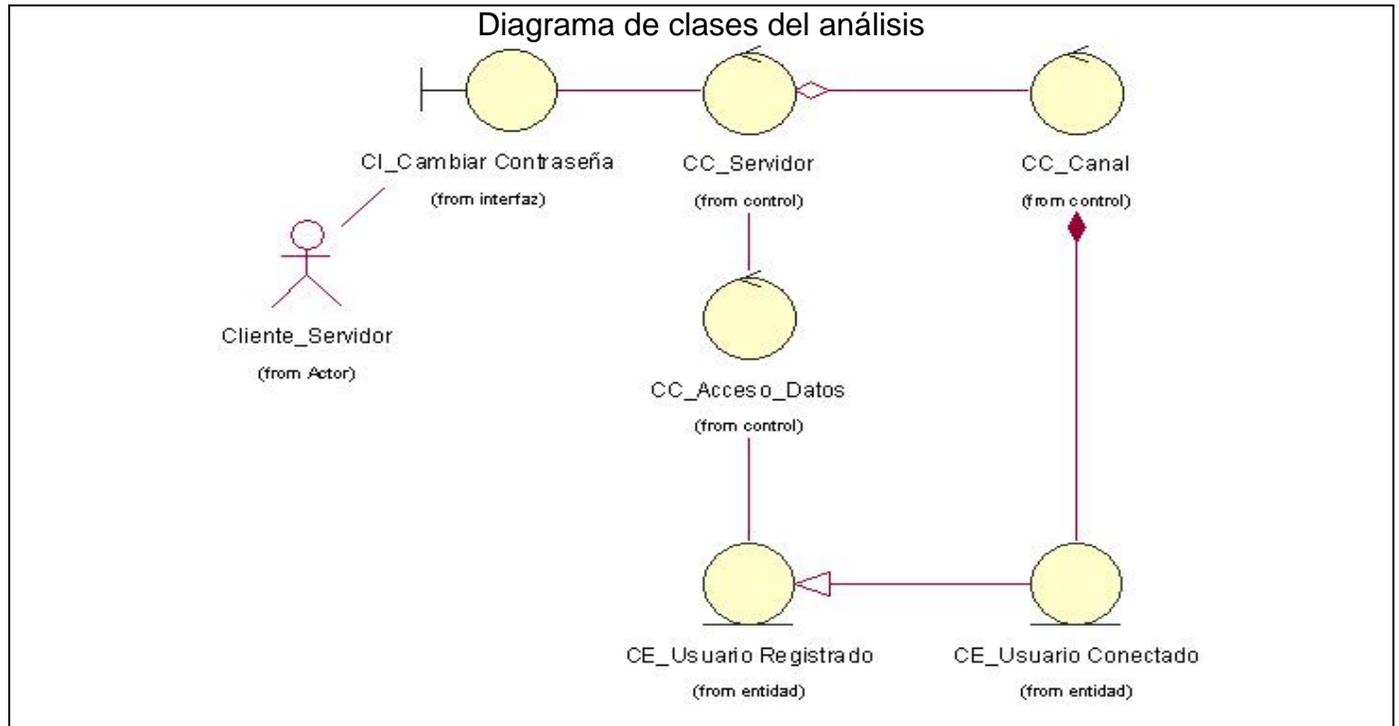
Durante la realización del modelo de análisis se precisa la elaboración de una serie de artefactos vinculados a este, el diagrama de clases del análisis constituye uno de los pilares más importantes en esta etapa desarrollo. El diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema.

Una clase de análisis representa una abstracción de una o varias clases y/o subsistemas del diseño. Se centra en el tratamiento de requisitos funcionales y pospone los no funcionales para el diseño, según RUP siempre se ajusta a alguno de los estereotipos siguientes: interfaz, control o entidad (Jacobson, y otros, 2000).

Para el desarrollo de la plataforma se realizó un diagrama de clases del análisis para cada caso de uso. La figura que se muestra a continuación ilustra el diagrama de clases del análisis del caso de uso **cambiar contraseña de usuario**.

### 3.2.1 Diagrama de clases del análisis

**Tabla 5. Diagrama de clases del análisis. CUS Cambiar Contraseña**



### 3.3 Modelo de Diseño

El modelo de diseño describe la realización física de los casos de uso centrándose tanto en los requisitos funcionales como en los no funcionales. En el diseño se modela el sistema y se confecciona su estructura (arquitectura), que sirve de soporte para todos los requisitos. Las abstracciones del modelo de diseño tienen una correspondencia directa con los elementos físicos del ambiente de implementación.

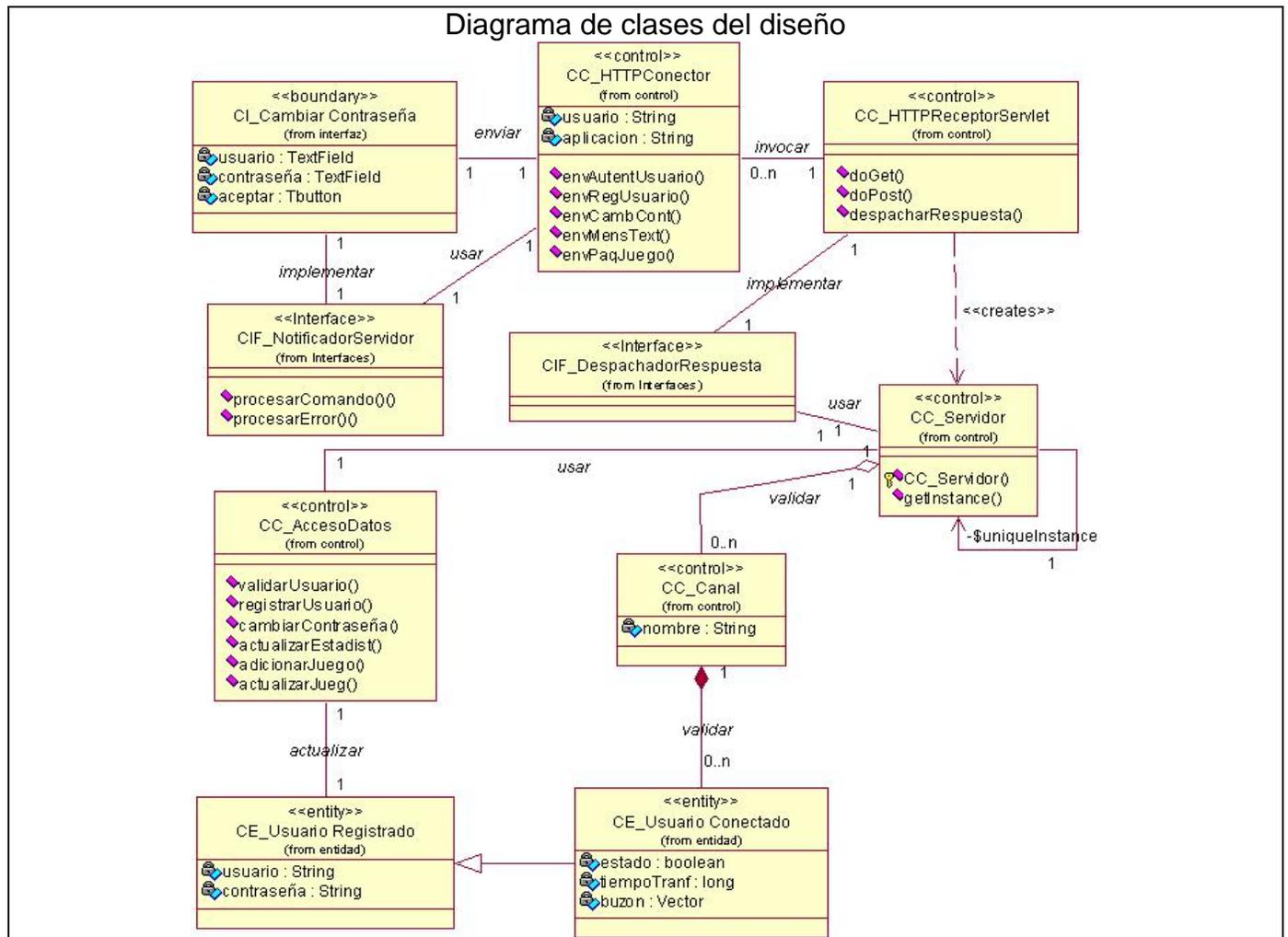
La realización de caso de uso del diseño contiene una descripción de flujos de eventos textuales, diagramas de clases y diagramas de interacción. Los diagramas de clases son los más utilizados en el modelado de sistemas orientados a objetos. Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones.

Una clase de diseño es una abstracción sin costuras con una clase o construcción similar en la implementación del sistema. Se especifican en un lenguaje de implementación (Java, C#, etc.), además se le atribuyen visibilidad a sus atributos y operaciones. **(Jacobson, y otros, 2000)**

Para el desarrollo de la plataforma se realizó un **diagrama de clases del diseño** y un **diagrama de secuencia** para cada caso de uso. La figura que se muestra a continuación ilustra el diagrama de clases del análisis del caso de uso **cambiar contraseña de usuario**.

### 3.3.1 Diagrama de clases del diseño

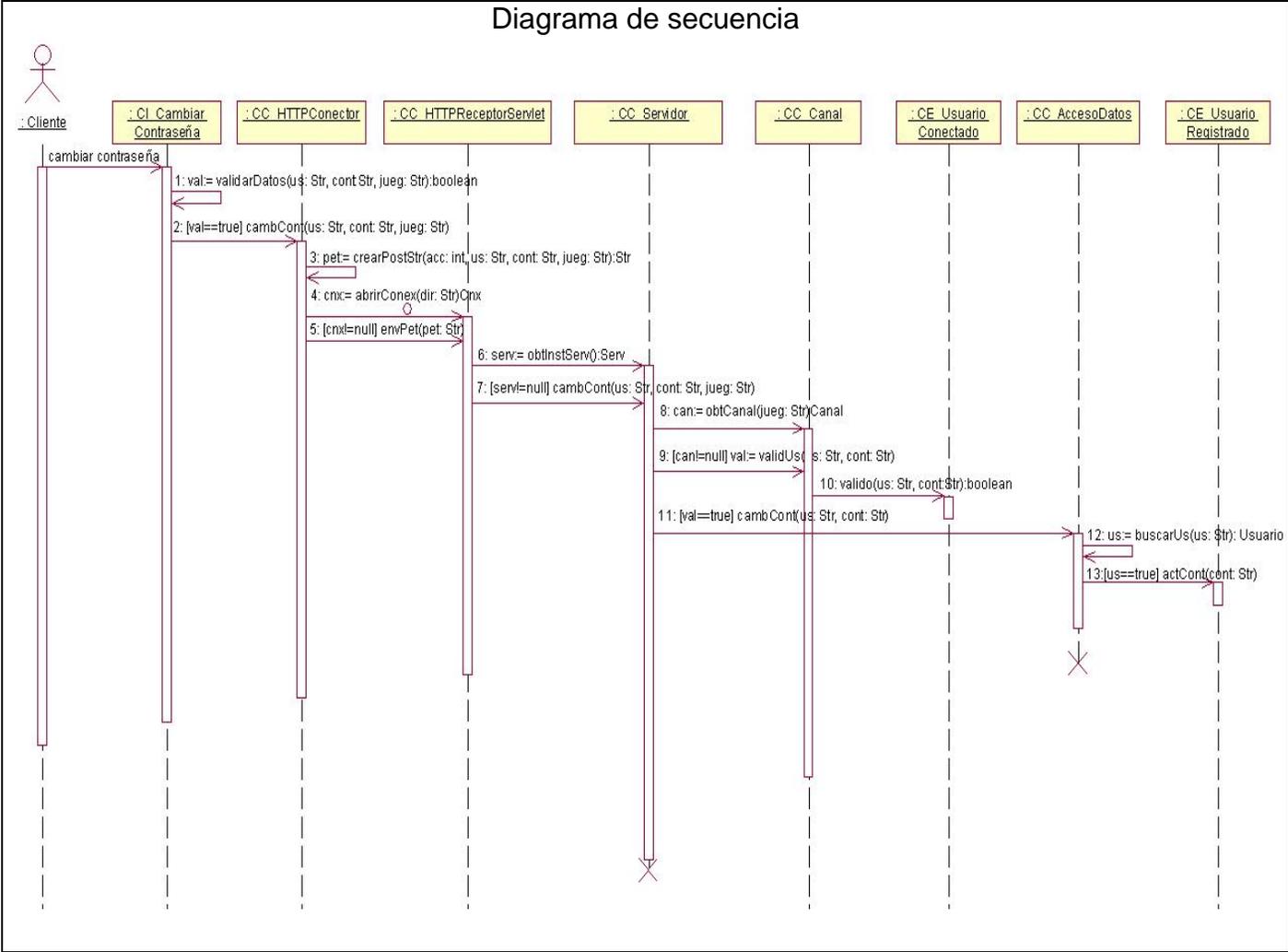
Tabla 6. Diagrama de clases del diseño. CUS Cambiar Contraseña



El diagrama de secuencia está compuesto por un conjunto de objetos y sus relaciones, donde se incluyen los mensajes que se pueden enviar entre ellos, destacando su ordenación temporal. Se utiliza para modelar los aspectos dinámicos de los sistemas. La figura que se muestra a continuación ilustra el diagrama de secuencia del caso de uso **cambiar contraseña de usuario**.

3.3.2 Diagrama de secuencia

Tabla 7. Diagrama de secuencia. CUS Cambiar Contraseña



### 3.4 Patrones de diseño empleados en el desarrollo de la aplicación

Con el desarrollo del modelo orientado a objeto, OOP (del inglés: Object Oriented Programming) el uso de los patrones de diseño ha tomado un auge importante, hoy en día son muy populares y están presentes en casi todos los sistemas informáticos a nivel mundial.

Un **patrón de diseño** constituye la descripción de clases y objetos involucrados en la solución adaptada de un problema de diseño general en un contexto particular, hace especial énfasis en la forma de comunicarse de los objetos. Identifica las clases, instancias, roles, colaboraciones, que contribuyen a lograr que el código obtenido sea flexible y que satisfaga los criterios especificados. Un patrón de diseño es una manera más práctica de describir aspectos organizacionales de la organización de un problema (Larman, 2004).

En el desarrollo de la solución planteada, se usaron varios patrones de diseño con el objetivo de lograr un software más acabado, robusto, flexible y de mayor calidad. Entre los distintos patrones se pueden citar los encargados de describir los principios fundamentales durante la asignación de responsabilidades a objetos GRASP (del inglés: General Responsibility Assignment Software Patterns) 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 usados (Gamma, y otros, 1994).

#### **GRASP:**

**Experto:** A la hora de implementar un sistema de de mucha importancia saber asignarle a cada clase el grupo de operaciones que debe llevar a cabo, el patrón experto recomienda asignarle a una clase solo aquellas responsabilidades para las que ella contenga la información necesaria para su cumplimiento. Durante el desarrollo de la aplicación se llevó a cabo el uso de este patrón de diseño con el propósito de disminuir el tiempo de respuesta a cada petición realizada por el cliente así como la organización del contenido.

**Creador:** Las peculiaridades de la plataforma j2me obligan a hacer un uso óptimo de la memoria del sistema, es determinante saber asignar la responsabilidad de crear una instancia de una clase solo aquella clase que la requiera y así evitar que se creen objetos repetidos.

**Bajo Acoplamiento:** Lograr un bajo acoplamiento en las clases j2me es de mucha importancia, la reducida memoria de los teléfonos celulares constituye un factor determinante en los sistemas orientados a éstos dispositivos. El desarrollo de un API que se anexará a los juegos conllevó a seguir plenamente las características de este patrón.

### 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.
- **Única Instancia:** Debido a las características de la plataforma desarrollada, se hizo necesario que la información requerida por cada módulo de sistema fuera totalmente confiable, para ello se decidió crear solo una única instancia la clase servidor y así garantizar que la información solicitada por 2 o más módulos fuera igual.

### 3.5 Tratamiento de errores

La plataforma posee un sistema de control interno para el tratamiento de los posibles errores que puedan ocasionarse durante la ejecución de alguna acción en el servidor o en el proceso de comunicación con el mismo, con el fin de garantizar la integridad y confiabilidad de la información que en ella se maneja. Se lleva a cabo un análisis del error ocurrido detectando la gravedad del mismo iniciando una serie de acciones y medidas en pos de mantener la integridad y funcionalidad del servidor.

En el desarrollo de la plataforma se han detectado una serie de acciones que pueden provocar un mal funcionamiento del servidor, especificando para cada una de ellas un mecanismo de recuperación. Los mensajes de errores son traducidos a un lenguaje de fácil comprensión para el cliente.

Algunos de los mensajes de error que pueden llegar a la aplicación cliente son los siguientes:

**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 autenticado.	Notificar al usuario que intenta acceder al sistema que las credenciales se encuentran en uso.
ERROR_PARTIDA_DUPLICADA	El cliente intenta crear una partida con un nombre reservado por una partida creada.	Notificar al cliente que debe seleccionar un nombre distinto para la partida.
ERROR_USUARIO_DESCONOCIDO	El cliente trata de acceder al sistema con un usuario desconocido, que no ha sido registrado en el sistema.	Notificar al usuario que el usuario proporcionado no se encuentra autenticado en el servidor.
ERROR_CANAL_DESCONOCIDO	El cliente envía un canal que no se encuentra creado en el sistema.	Notificar al cliente que el canal proporcionado no se encuentra creado en el servidor.
ERROR_PARTIDA_DESCONOCIDA	El cliente informa que está dentro de una partida que no se encuentra dentro del sistema.	Notificar al usuario que la partida proporcionada no se encuentra activa en el servidor.
ERROR_CONEXION_PERDIDA	Puede suceder por múltiples causas que involucran principalmente problemas en la red del cliente que afectan la conexión con el servidor.	Notificar al cliente que ha perdido la conexión con el servidor.
ERROR _ SISTEMA	Error grave del sistema que impide su correcto funcionamiento.	El sistema intenta recuperar la información y restablecer los servicios.

### 3.6 Seguridad

Es de vital importancia el tema de la seguridad en el desarrollo de la plataforma, para ello se definió un grupo de medidas encaminadas a evitar la transformación o suplantación de la información referente al sistema.

Una de las medidas implementadas es el proceso de encriptación de todos los datos que se hacen persistentes para la plataforma y que contienen información relacionada con los usuarios y sus estadísticas, a través de una encriptación que utiliza el algoritmo MD5.

Otra parte importante en la seguridad de la plataforma es el proceso de intercambio de información entre los clientes y el servidor, para ello se utilizaron las ventajas de seguridad brindadas por GPRS y por la plataforma JAVA, de forma tal que se detecte el arribo de información ajena al sistema.

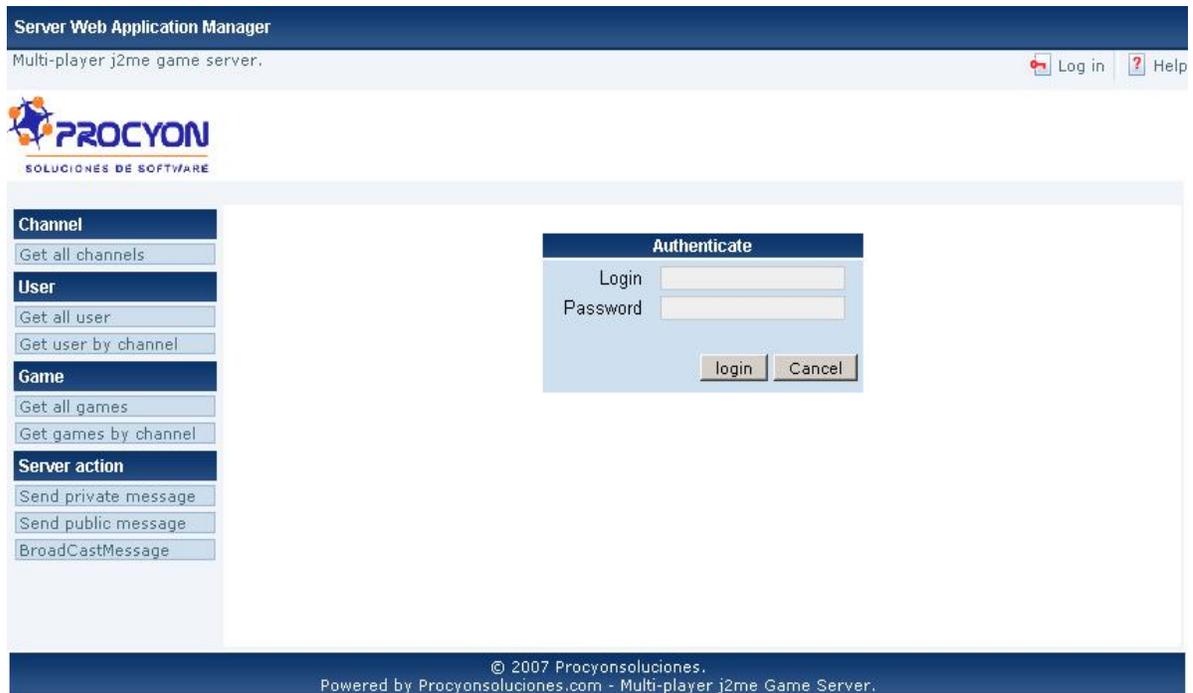
Además de las medidas hasta ahora expuestas, se utilizaron todos los mecanismos y reglas de seguridad definidas en la plataforma J2ME.

### 3.7 Interfaz

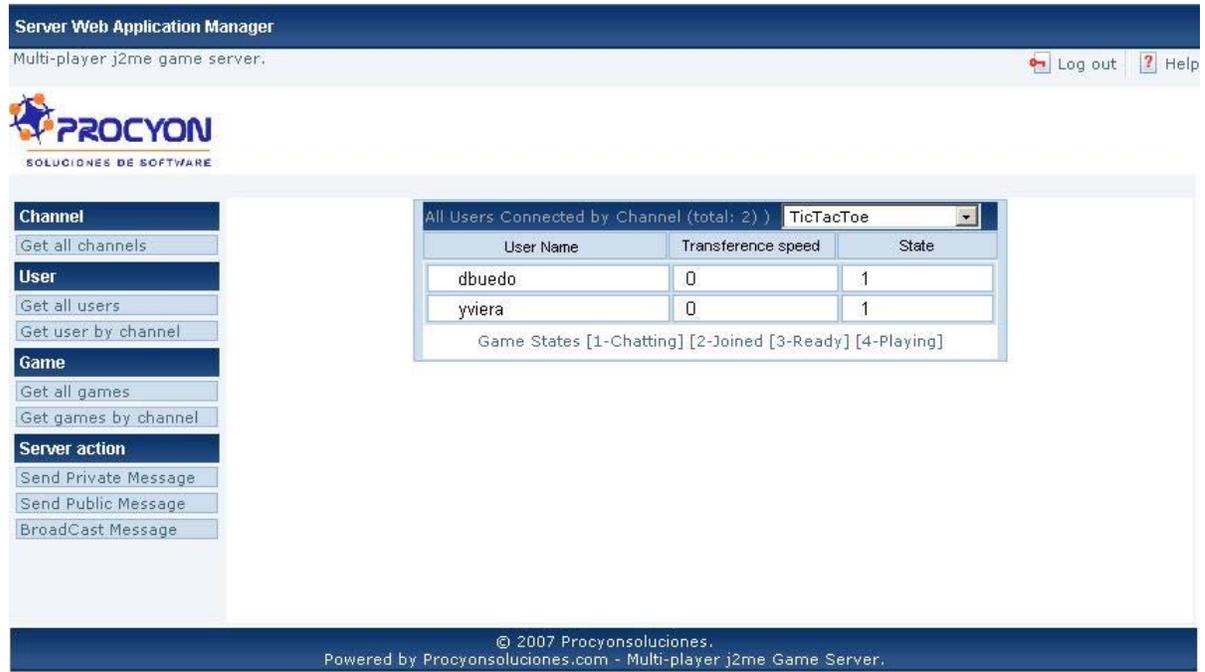
Debido a que la plataforma es una aplicación que corre en un servidor Web, la comunicación con ella se hizo mediante una aplicación J2ME que fue desarrollada por disímiles clientes. Ellos son los encargados de visualizar la información según sus necesidades, por lo que el desarrollo de las interfaces de usuarios no está en nuestro ámbito. En la versión 2.0 del API, desarrollada para la comunicación con el servidor, se incluirá un paquete para estandarizar el proceso de autenticación de usuario a través de una interfaz gráfica de usuario.

Para el desarrollo del sitio Web de administración se usó una interfaz de usuario sencilla y sin muchas imágenes, lo que aumenta el tiempo de respuesta a las peticiones realizadas. La misma cuenta con un

menú vertical, en el lado izquierdo, que facilita la navegación a través del sitio. Fue desarrollada utilizando diferentes tonalidades del color azul como se muestra en las siguientes imágenes.



**Figura #5. Pantalla de autenticación**



**Figura #6. Listado de usuarios por canal**

## 3.8 Concepción de la ayuda

Debido a las características del sistema, el proceso de concepción de la ayuda es atípico y consta de dos partes fundamentales:

- Una correcta documentación de todas las clases brindadas en el API para la comunicación entre ambas partes, en la que se abordan las características de la misma, en específico una explicación detallada de todas las clases e interfaces que la componen. Además, proporciona una descripción detallada de cómo usar las clases brindadas a la hora de desarrollar una aplicación específica, brindando además, ejemplos prácticos, previamente desarrollados, que facilitarán el proceso de aprendizaje para el uso del API.

- Referente a la aplicación servidor la documentación es más limitada en este aspecto, pues solo se especifica un grupo mínimo de características del servidor, debido a que los programadores en la presente versión no necesitarán acceder o modificar el código de la plataforma, se llevó a cabo una descripción detallada de todas las clases, sus respectivos atributos y operaciones.

### **3.9 Conclusiones**

En el capítulo se elaboraron los principales artefactos relacionados con el análisis y diseño del sistema de la solución propuesta, haciendo especial énfasis en los diagramas de clases de análisis y diseño, así como los diagramas de secuencias pertenecientes a cada caso de uso. Se describieron los patrones de diseño usados a lo largo de la aplicación, así como las medidas tomadas por parte del grupo de desarrollo para mitigar los posibles errores durante la ejecución de las acciones en el servidor. Se realizó la descripción de las características y estándares de diseño empleados en el desarrollo de la interfaz de usuario.

# CAPÍTULO 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

En el flujo de trabajo de diseño se propone crear un plano del modelo de implementación, por lo que sus últimas actividades están vinculadas a la creación del modelo de despliegue. El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

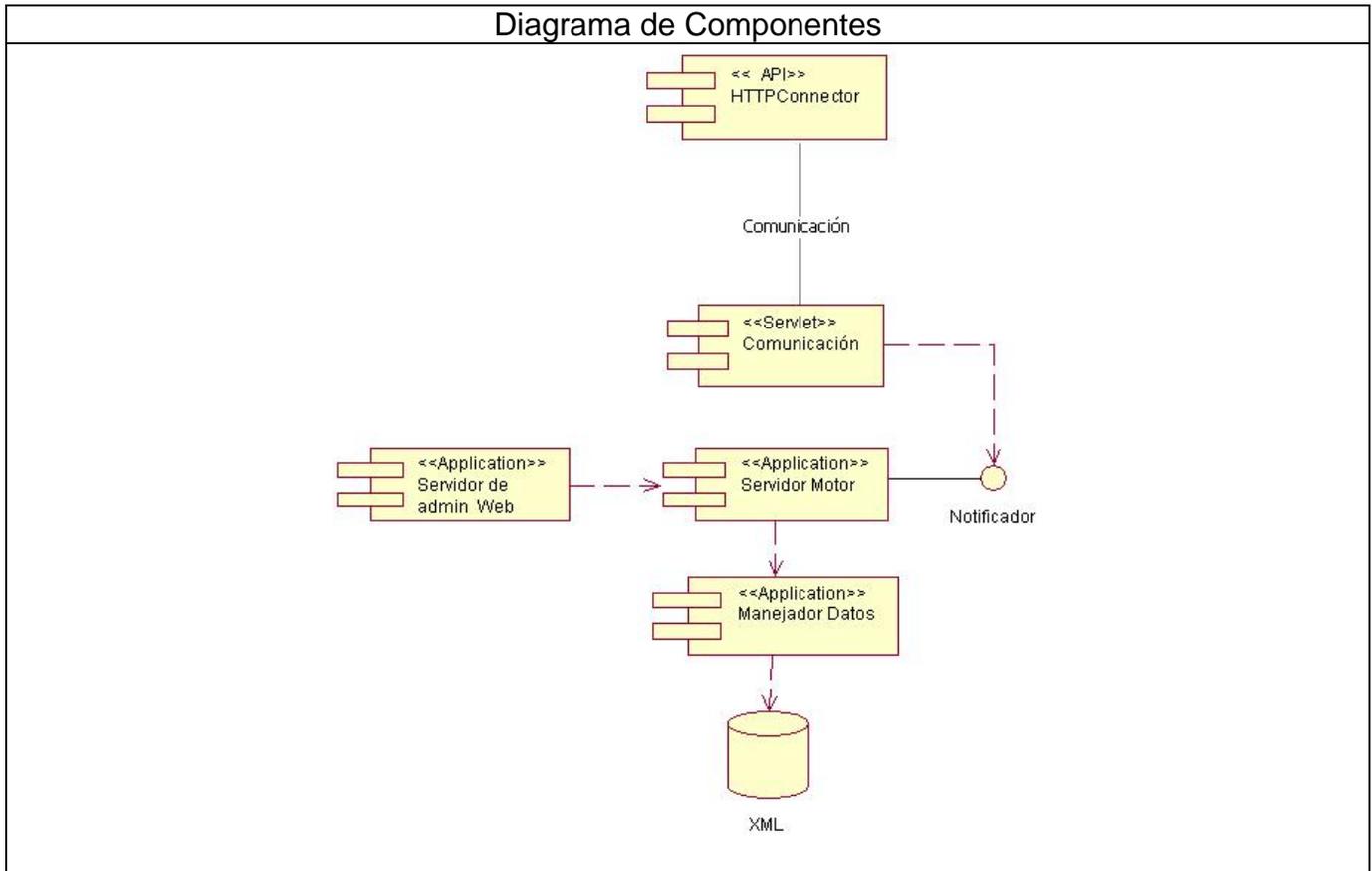
Los **diagramas de despliegue** y **componentes**, que son artefactos generados en este flujo de trabajo, conforman lo que se conoce como un **modelo de implementación**, al describir los componentes a construir y su organización y dependencia entre nodos físicos en los que funcionará la aplicación.

Los componentes constituyen la parte modular del sistema, encapsulan implementación y un conjunto de interfaces y proporciona la realización de los mismos. Un componente típicamente contiene clases y puede ser implementado por uno o más artefactos (ficheros ejecutables, binarios, etc.).

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. (Jacobson, y otros, 2000).

Para la realización de la solución propuesta se desarrollaron una serie de componentes, los cuales se encuentran referenciados en el diagrama que se ilustra a continuación.

Tabla 9. Diagrama de Componentes



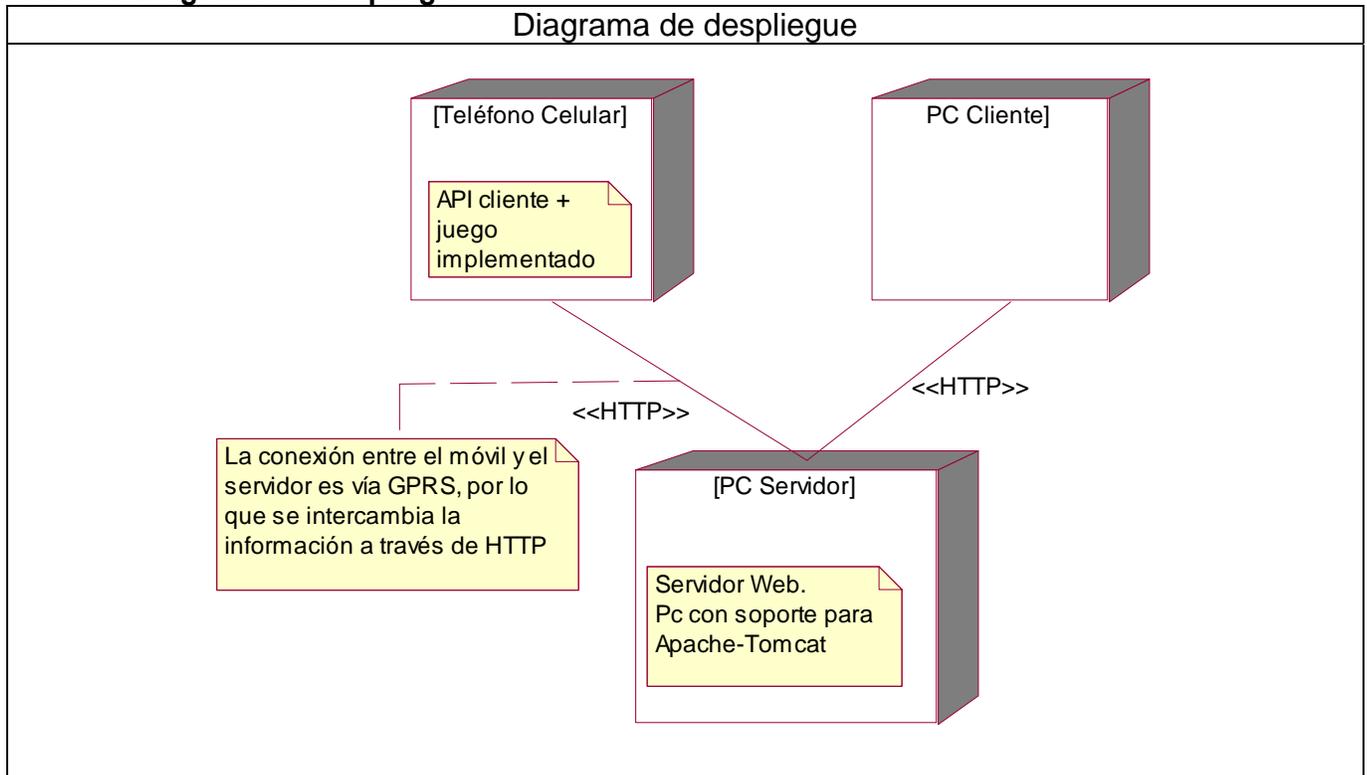
#### 4.2.1 Diagrama de despliegue

En el diagrama de despliegue se indica la situación física de los componentes lógicos desarrollados. Es decir, se sitúa el software en el hardware que lo contiene. Cada Hardware se representa como un nodo.

Un nodo se representa como un cubo, es un elemento donde se ejecutan los componentes, representa el despliegue físico de estos componentes (Jacobson, y otros, 2000).

Atendiendo a las características del sistema antes planteadas se realizó el diagrama de despliegue que se muestra a continuación, el cuál satisface todas las necesidades y exigencias de la aplicación.

**Tabla 10. Diagrama de despliegue**



### 4.3 Modelo de pruebas

El desarrollo del software implica una serie de actividades de producción en las que las posibilidades de de cometer errores son elevadas, debido a esto ha de ir acompañado de una actividad que mejore la calidad del mismo.

**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 obtener mejoras en la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación.

### 4.3.1 Métodos de Prueba

Existen actualmente 2 técnicas de pruebas fundamentales: caja blanca y caja negra.

- **Prueba de caja negra:** Pruebas que se llevan a cabo sobre la **interfaz de usuario**, se centran en chequear las funcionalidades operativas del sistema. Las pruebas de caja negra verifican que los datos de entrada al sistema sean aceptados correctamente y que produzcan los resultados esperados en cada caso, así como el chequeo de la integridad de la información externa añadida al mismo.
- **Prueba de caja blanca:** Las pruebas de caja blanca se basan en el diseño de casos de prueba, los que son diseñados a partir de la estructura de control del diseño procedimental. Requieren del conocimiento de la estructura interna del programa y son derivadas a partir de las especificaciones internas del diseño o del código. A partir de ellas se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado.

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

Para garantizar el correcto funcionamiento de la plataforma se llevaron a cabo los casos de prueba que se describen a continuación

**Caso de Uso:** Crear partida

**Caso de Prueba:** CPR 1 Crear partida

**Descripción del flujo**

1- Se introducen los datos pertinentes para la ejecución de la acción, los mismos son:

- Nombre de usuario.
- Nombre del canal o juego.
- Nombre de la partida.
- Cantidad mínima de usuarios que permite la partida.
- Cantidad máxima de usuarios que permite la partida.

**Precondiciones:**

- Ejecutar el caso de uso Autenticar usuario.

**Tabla 11. Caso de Prueba Crear partida.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado
Se introducen los siguientes datos:  <b>Nombre de usuario:</b> denys <b>Nombre del canal:</b> Tictac toe <b>Cantidad mín. de usuarios:</b> 2 <b>Cantidad máx. de usuarios:</b> 2		Se crea la partida.	Satisfactorio

	<p>Se introducen los siguientes datos:</p> <p><b>Nombre de usuario:</b> denys  <b>Nombre del canal:</b> vacío  <b>Cantidad mín. de usuarios:</b> 2  <b>Cantidad máx. de usuarios:</b> 2</p>	<p>Se dispara error de canal desconocido</p>	<p>Satisfactorio</p>
	<p>Se introducen los siguientes datos:</p> <p><b>Nombre de usuario:</b> vacío  <b>Nombre del canal:</b> Tictac toe  <b>Cantidad mín. de usuarios:</b> 2  <b>Cantidad máx. de usuarios:</b> 2</p>	<p>Se dispara error de usuario desconocido</p>	<p>Satisfactorio</p>
	<p>Se introducen los siguientes datos:</p> <p><b>Nombre de usuario:</b> denys  <b>Nombre del canal:</b> Tictac toe  <b>Cantidad mín. de usuarios:</b> vacío  <b>Cantidad máx. de usuarios:</b> 2</p>	<p>Se dispara error de cantidad mínima de usuarios inválida.</p>	<p>Satisfactorio</p>
	<p>Se introducen los siguientes datos:</p> <p><b>Nombre de usuario:</b> denys  <b>Nombre del canal:</b> Tictac toe  <b>Cantidad mín. de usuarios:</b> vacío  <b>Cantidad máx. de usuarios:</b> 2</p>	<p>Se dispara error de cantidad máxima de usuarios inválida.</p>	<p>Satisfactorio</p>

**Caso de Uso:** Enviar mensaje de texto.

**Caso de Prueba:** CPR 2 Enviar mensaje de texto.

**Descripción del flujo**

1- Se introducen los datos pertinentes para la ejecución de la acción, los mismos son:

- Nombre de usuario.
- Nombre del canal o juego.
- Mensaje.

**Precondiciones:**

- Ejecutar el caso de uso Autenticar usuario.

**Tabla 12. Caso de Enviar mensaje de texto**

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba
Se introducen los siguientes datos: <b>Nombre de usuario:</b> denys <b>Nombre del canal:</b> Tictac toe <b>Mensaje:</b> Hola Mundo		Se envía el mensaje	satisfactorio
	Se introducen los siguientes datos: <b>Nombre de usuario:</b> denys <b>Nombre del canal:</b> vacío <b>Mensaje:</b> Hola Mundo	Se dispara error de canal desconocido	satisfactorio
	Se introducen los siguientes datos: <b>Nombre de usuario:</b> vacío <b>Nombre del canal:</b> Tictac toe <b>Mensaje:</b> Hola Mundo	Se dispara error de usuario desconocido	satisfactorio
	Se introducen los siguientes datos:	Se dispara error de argumento inválido.	satisfactorio

	<b>Nombre de usuario:</b> denys <b>Nombre del canal:</b> Tictac toe <b>Mensaje:</b> vacío		
--	---	--	--

El objetivo de la prueba de software es descubrir errores y defectos. Para conseguir este objetivo se planifican y se ejecutan una serie de pasos que van revisando todos los elementos del software. En todas las fases del desarrollo del proyecto hay que probar el software que se va construyendo, aunque en la fase de construcción se centran los mayores esfuerzos de este flujo. 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 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

Para realizar un proyecto es de vital importancia el análisis del costo y los beneficios que reportará. Los principales resultados de este análisis son el tiempo de desarrollo en meses, el costo y la cantidad de personas que se necesitan para desarrollar el proyecto. En el presente capítulo se describe la estimación de costos del sistema y sus beneficios.

## 5.2 Planificación basada en casos de uso

**Paso 1.** Cálculo de los Puntos de casos de uso desajustados

$$UUCP = UAW + UUCW$$

Donde:

- **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.

**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	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	4	12
Total			4	12

$$UAW = \sum cant\ actores * peso$$

$$UAW = 12$$

**Tabla 14. Factor de peso de los casos de uso sin ajustar.**

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	19	95
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			19	95

$$UUCW = \sum cant\ CU * Peso$$

$$UUCW = 95$$

$$UUCP = 12 + 95$$

$$UUCP = 107$$

**Paso 2.** Cálculo de los Puntos de casos de uso ajustados.

$$UCP = UUCP * TCF * EF$$

Donde:

- **UCP:** Puntos de casos de uso ajustados.
- **UUCP:** Puntos de casos de uso sin ajustar.
- **TCF:** Factor de complejidad técnica.
- **EF:** Factor de ambiente.

El factor de complejidad técnica (TCF) se calcula mediante la cuantificación de varios factores que determinan la complejidad técnica del sistema. Cada factor se cuantifica en un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

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

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido	2	0	0
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	2	2
T4	Funcionamiento Interno complejo	1	4	4
T5	El código debe ser reutilizable	1	4	4
T6	Facilidad de instalación	0,5	4	2
T7	Facilidad de uso	0,5	4	2
T8	Portabilidad	2	5	10
T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	4	4
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	1	1
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1	1
Total				40

$$TCF = 0.6 + 0.01 * \sum (peso * valor asignado)$$

$$TCF = 0.6 + 0.01 * 40$$

$$TCF = 0.6 + 0.4$$

$$TCF = 1$$

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	2	1
E3	Experiencia en la orientación a objetivos.	1	5	5
E4	Capacidad del analista líder.	0,5	4	2
E5	Motivación.	1	5	5
E6	Estabilidad de requerimientos	2	5	10
E7	Personal Part–Time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	2	-2
Total				25.5

$$EF = 1.4 - 0.03 * \sum (peso * valor asignado)$$

$$EF = 1.4 - 0,03 * 25.5$$

$$EF = 1.4 - 0.765$$

$$EF = 0.634$$

$$UCP = UUCP * TCF * EF$$

$$UCP = 107 * 1 * 0.634$$

$$UCP = 67.838$$

**Paso 3.** Estimación de esfuerzo a través de los puntos de casos de uso

$$E = UCP * CF$$

Donde:

E: Esfuerzo estimado en horas hombres.

UCP: Punto de casos de usos ajustados.

CF: Factor de conversión.

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 Horas-Hombre / Punto de Casos de uso.

$$E = 67.838 * 20$$

$$E = 1356.76 \text{ Horas-Hombre}$$

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

**Tabla #17. Esfuerzo del proyecto**

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	339.725
Diseño	20	679.45
Implementación	40	1358.9
Pruebas	15	509.5875
Sobrecarga (otras actividades)	15	509.5875
Total	100	3397.25

Si  $E_T = 3397.25$  horas-hombre y se estima que cada mes tiene como promedio 192 horas laborables, eso daría un  $E_T = 12.5824074$  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.

$$CH = 2 \text{ hombres}$$

$$\text{CHM} = 2 * \text{Salario Promedio}$$

$$\text{CHM} = 100.00 \text{ \$/mes}$$

$$\text{Costo} = \text{CHM} * E_T / \text{CH}$$

$$\text{Costo} = 200.00 * 12.5824074 / 2$$

$$\text{Costo} = \$ 1258.24074 \approx \$ 1258.24$$

$$\text{Tiempo} = E_T / \text{CH}$$

$$\text{Tiempo} = 12.5824074 / 2$$

$$\text{Tiempo} = 6.2912 \approx 6.29 \text{ meses}$$

De los resultados obtenidos se interpreta que con 2 hombres trabajando en el proyecto el mismo se desarrolla en 6.29 meses y su costo total se estima que sea \$1258.24.

### 5.3 Beneficios tangibles e intangibles

La Plataforma para la gestión de juegos multiusuario en dispositivos móviles vía GPRS conjuntamente con el API desarrollado para la comunicación con el servidor es un producto con fines comerciales, su principal objetivo es proveer a los operadores de un servidor común para juegos multiusuario cliente-servidor.

El sistema ofrece dos beneficios fundamentales:

- 1- El desarrollo de una plataforma común para todos los juegos, permite que los desarrolladores, una vez que adquieran práctica en el desarrollo de aplicaciones para la plataforma, reduzcan considerablemente los tiempos de desarrollo. De forma similar cuando el sistema sea vendido a un proveedor de servicios, facilitará el proceso de instalación, mantenimiento y administración.

- 2- Por otra parte el sistema brinda un API para la comunicación entre aplicaciones clientes con el servidor, dicho API básicamente garantiza que todas las aplicaciones se comuniquen con el servidor en un lenguaje comprensible por él, con los comandos adecuados para cada acción, disminuyendo considerablemente el tiempo de desarrollo de nuevos juegos sobre la plataforma.

Algunos beneficios inmediatos son:

- Disminución del tiempo y esfuerzo que se invierte en desarrollar un juego multiusuario vía GPRS.
- Posibilidad de migración de los juegos previamente desarrollados en la empresa, a juegos multiusuario usando el sistema diseñado.
- Disminución de los gastos, pues se reduce considerablemente el tiempo de desarrollo de los juegos multiusuario.

### **5.4 Análisis de costos y beneficios**

El desarrollo de un producto informático tiene como principal costo el salario proporcionado a los desarrolladores. Debido a que el sistema implementado reduce los tiempos de desarrollo, este costo salarial disminuye, pues solo se debe desarrollar la aplicación cliente, sin necesidad de implementar la conexión al servidor ni el servidor en sí.

El sistema está enmarcado dentro de la entidad Procyon, en fase de pruebas con el objetivo de garantizar su estabilidad para su futura venta al mercado, con una gama de juegos disponibles.

Una vez instalado en el proveedor se podrán conectar nuevos juegos sin necesidad de modificar el código de la plataforma servidor, lo cual se considera un beneficio importante a considerar.

La tecnología utilizada para el desarrollo del sistema es totalmente libre, por tanto no hay que incurrir en gastos de licencias de uso. El sistema es portable por lo que un cambio de plataforma para la implantación del mismo es viable y factible.

Analizando el costo del proyecto y los numerosos beneficios que reporta, detallados con anterioridad, se puede concluir que su implementación es factible.

### **5.5 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.

La herramienta propuesta reportará beneficios significativos e importantes para el desarrollo de juegos multiusuario vía GPRS, al contribuir a mejorar los procesos de desarrollo de los juegos y reducir su tiempo de desarrollo.

## CONCLUSIONES

Con el propósito de darle cumplimientos a los objetivos trazados se desarrollaron todas las tareas planteadas para la realización del mismo. Se profundizó en el uso de las tecnologías inalámbricas, en especial GPRS y se sentaron las bases en el país para el desarrollo de futuros trabajos relacionados con la transferencia de datos entre dispositivos móviles. Se empleó RUP como metodología para organizar el proceso de desarrollo del software.

La plataforma desarrollada tiene como principal característica que gestiona múltiples juegos, sin ser significativo su tipo, es capaz de controlar el intercambio de información y recurso entre los usuarios conectados al servidor, distribuyendo éstos en dependencia del canal en el que se encuentren. Brinda un sistema de registro de usuarios, los cuales son almacenados de forma segura, así como las estadísticas de las partidas desarrolladas.

Se desarrolló un API de conexión con el servidor orientado a los desarrolladores de juegos, el cual recoge las acciones relacionadas con el funcionamiento de las aplicaciones clientes.

La aplicación está acompañada de una detallada bibliografía, un juego de prueba y un módulo de administración Web que posibilita el monitoreo del estado del servidor en todo momento.

Como resultado de los estudios realizados y el desarrollo de la soluciones expuesta se puede concluir que todos los objetivo planteados fueran cumplidos cabalmente y se recomiendan para futuros trabajos algunos aspectos que mejorarán e incrementarán la funcionalidades del mismo.

## RECOMENDACIONES

- Adicionar la posibilidad de comunicarse con el servidor vía socket.
- Ampliar las funcionalidades de la plataforma para manejar la lógica del juego en el servidor.

## BIBLIOGRAFÍA

3G Americas. 2007. 3G Americas. [Consultado] Enero del 2007.

<http://www.3gamerica.org/Spanish/Statistics/>.

Apache Tomcat. 2005. The Apache Software Foundation. [Consultado] Enero del 2007.

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

Armstrong, Eric. 2005. [Consultado] Marzo del 2007. <http://java.sun.com/j2ee/1.4/docs/tutorial/doc/>.

Aurora. 2006. Acerca de la Tecnología GSM. *Aurora Wireless Technology*. [Consultado] Febrero del 2007.

<http://www.aurora.com.do/gsm.htm>.

Blanco Oliver, Jose. 2002. GPRS (General Packet Radio Service) . *Pc-news.com*. [Consultado] 23 de Abril de 2007. <http://www.pc-news.com/detalle.asp?sid=&id=44&lda=721>.

Contreras, Daniel. 2003. Vocabulario de la telefonía móvil. *Mause Digital*. [Consultado] Enero del 2007.

<http://mouse.latercera.cl/2003/taller/05/02/index.htm>.

Larman, Craig. 2004. *UML y Patrones*. 2004. Vols. 9,10,11,12.

Eclipse Foundation. 2000. Eclipse. [Consultado] Febrero del 2007. <http://www.eclipse.org/>.

Ex Machina. 2003. Ex Machina = Mobile Multiplayer Gaming. [Consultado] Febrero del 2007.

<http://www.exmachina.nl/sp.html>.

Exit Games. 2006. Exit Games. [Consultado] Enero del 2007. <http://www.exitgames.com/>.

Game Federation. 2005. Game Federation. [Consultado] Enero del 2007.

<http://www.gamefederation.com/?nav=products.products>.

Gamma, Erich, y otros. 1994. *Gang of Four*. Abril 2007.

Izquierdo, Susana y Ahijado, Francisco Javier. 2005. Control de versiones. *Ahijado's blog*. [Consultado] Mayo del 2007. <http://www.ahijado.info/configuracion-de-subversion/>.

- Jacobson, Ivar, y otros. 2000. *El Proceso Unificado de Desarrollo de Software*. Abril del 2007.
- Jiménez, José Juan. 2006. Monografías. *Evolución e historia de la telefonía celular*. [Consultado] Febrero del 2007. <http://www.monografias.com/trabajos14/celularhist/celularhist.shtml#gener>.
- Miguel Blázquez, José. 2006. ¿Conexiones con cable o inalámbricas? – Ventajas e inconvenientes. *ConfigurarEquipos*. [Consultado] Febrero del 2007. <http://www.configurarequipos.com/doc351.html>.
- Molpeceres Touris, Alberto y Pérez Mariñán, Martín. 2002. *Arquitectura empresarial y software libre, J2EE. javaHispano*. [Consultado] Marzo del 2007.
- Movil 10. 2006. Introducción a las redes GPRS:. *MelodiasMoviles.com*. [Consultado] Febrero del 2007. <http://www.melodiasmoviles.com/documentacion/intro-gprs.php>.
- Nokia. 2005. *Snap Mobile*. [Consultado] Enero del 2007. <http://snapmobile.nokia.com/core/en/index.html>.
- PixelPlay. 2006. PixelPlay. [Consultado] Enero del 2007. <http://www.pixelplay.com/technology/index.html>.
- Pradilla Cerón, Juan Vicente. 2007. Universidad del Valle. [Consultado] Abril del 2007. <http://eisc.univalle.edu.co/materias/WWW/J2ME/IntroduccionAJ2ME.pdf>.
- Pressman, R.S. 2001. [aut. libro] Pressman R.S. *Ingeniería del Software: un enfoque práctico*. Mc Graw Hill : s.n., Mayo del 2007.
- Price, David. 2007. Multi-User MIDP Game Desing. [Consultado] Marzo del 2007. <http://fivedots.coe.psu.ac.th/Software.coe/J2ME/Nokia%20Articles/Multi-UserMIDPGameDesignv05.pdf>.
- SUD. 2005. Todo lo que siempre quiso saber sobre Teléfonos Móviles. [Consultado] Marzo del 2007. <http://www.latinsud.com/moviles/>.
- Sun microsystems. 2000. Sun Java Wireless Toolkit for CLDC. *Sun Developer Network (SDN)*. [Consultado] Enero del 2007. <http://java.sun.com/products/sjwtoolkit/>.
- Toth, Alexander. 2003. Rational Rose Enterprise Edition 2003. [Consultado] Mayo del 2007. <http://www.yz70.dial.pipex.com/Software%20Engineering.htm>.

Tigris. 2006. Subversion. *Tigris.org*. [Consultado] Febrero del 2007. <http://subversion.tigris.org/>.

Unipier. 2005. Unipier Entertainment Suite. [Consultado] Enero del 2007. <http://www.unipier.com/?id=138>.

Velázquez Pastrana, Lourdes. 2002. Tre Generaciones en la Telefonía Celular. *Enter@te*. [Consultado] Febrero del 2007. <http://www.enterate.unam.mx/Articulos/2002/agosto/telefccl.htm>.

WebmovilGsm. 2006. UMTS. *WebmovilGsm.com*. [Consultado] Marzo del 2007. <http://www.webmovilgsm.com/umts.htm>.

# ANEXO

## Anexo 1: Descripción detallada de los casos de uso

Tabla 18. Descripción detallada del CUS: Autenticar usuario.

<b>Nombre del Caso de Uso</b>	Autenticar usuario.	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Mantener controlados e identificados los usuarios en el servidor, así como limitar el acceso al mismo de personal no autorizado.	
<b>Resumen</b>	El caso de uso se inicia cuando un usuario desconocido intenta acceder al sistema, el mismo responde con una orden de autenticación, el usuario introduce sus credenciales y las envía al sistema el cual se encarga de procesarlas.	
<b>Referencias</b>	RF1.1	
<b>Precondiciones</b>	Poseer una cuenta en el servidor.	
<b>Poscondiciones</b>	El usuario es enviado a un canal asignado al juego desde el que se está conectando.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1- El cliente intenta abrir una nueva conexión con el servidor.	1.1. El sistema devuelve una petición de autenticación por parte del cliente.	
2. El cliente introduce sus credenciales y la envía al sistema.	2.1 El sistema verifica la validez de los datos recibidos en caso positivo permite la entrada del cliente al sistema.	
<b>Puntos de Extensión:</b> Si el usuario no está registrado en el servidor el sistema notifica al mismo y le ofrece la posibilidad de registrarse en el mismo.		
<b>Prioridad:</b>	Primaria	

Tabla 19. Descripción detallada del CUS: Registrar usuario.

<b>Nombre del Caso de Uso</b>	Registrar usuario.	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Registrar nuevos usuarios en el servidor que no posean cuentas en el mismo.	
<b>Resumen</b>	El caso de uso se inicia cuando un usuario desconocido decide acceder a los servicios del sistema y no se encuentra registrado en el mismo, le indica al sistema su petición y envía los datos necesarios para la ejecución de la acción.	
<b>Referencias</b>	RF 1.2.	
<b>Precondiciones</b>		
<b>Poscondiciones</b>	El usuario es registrado en el servidor permitiéndosele acceder a los distintos juegos con las credenciales proporcionadas.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1- El cliente solicita al servidor el registro del	1.1- El sistema recibe la petición y pasa a validar los	

mismo, enviándole los datos necesarios.	datos recibidos.
	1.2- Si no existen anomalías en los datos del cliente se procede a ejecutar la petición.
	1.3- El sistema notifica al cliente el éxito de la acción.
<b>Puntos de Extensión:</b> Si el usuario seleccionado por el cliente está reservado, se le notifica al cliente que debe seleccionar otro usuario y reenviar la información al servidor.	
<b>Prioridad:</b>	Primaria

**Tabla 20. Descripción detallada del CUS: Crear Partida.**

<b>Nombre del Caso de Uso</b>		Crear Partida.
<b>Actores</b>	Cliente	
<b>Propósito</b>	Crear una nueva partida.	
<b>Resumen</b>	El caso de uso se inicia cuando un usuario registrado decide crear una partida, indicándole al sistema su acción e identificando la misma con un nombre, el sistema recibe la petición y valida el nombre escogido por el cliente para identificar la partida, de no encontrarse alteración alguna se procede a realizar la acción solicitada.	
<b>Referencias</b>	RF.5.1	
<b>Precondiciones</b>	Usuario previamente autenticado.	
<b>Poscondiciones</b>	Una nueva partida es creada	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>		<b>Respuesta del Sistema</b>
1- El cliente indica al sistema que va a crear una nueva partida y le envía al mismo el nombre escogido para identificarla.		1.1- El sistema recibe la petición y pasa a validar la información recibida.
		1.2- Si no existe anomalía en la información recibida procede a realizar la acción solicitada.
<b>Puntos de Extensión:</b> Si existe otra partida que ha sido creada con un nombre similar al escogido por el cliente, el mismo sistema notifica lo ocurrido solicitando otro nombre.		
<b>Prioridad:</b>	Primaria	

**Tabla 21. Descripción detallada del CUS: Entrar Partida.**

<b>Nombre del Caso de Uso</b>		Entrar Partida.
<b>Actores</b>	Cliente	
<b>Propósito</b>	Permitir al cliente que se una a una partida que ha sido creada por otro jugador.	
<b>Resumen</b>	El caso de uso se inicia cuando un cliente registrado solicita unirse a una partida creada por otro jugador. Después de listar las mismas (partidas creadas en el servidor) selecciona una de ellas y notifica al sistema dicha acción.	
<b>Referencias</b>	RF.5.2	
<b>Precondiciones</b>	Usuario previamente autenticado en el servidor.	
<b>Poscondiciones</b>	El usuario es adicionado a la lista de jugadores de dicha partida y queda a la espera de comenzar la partida.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>		<b>Respuesta del Sistema</b>

1- El cliente indica al sistema que desea unirse a una partida que ha sido creada por otro cliente.	1.1- El sistema recibe la petición y chequea que la partida está disponible, de ser así procede a ejecutar dicha petición
<b>Puntos de Extensión:</b> Si la partida ha iniciado o está llena, el sistema notifica al cliente que la partida seleccionada no está disponible en ese instante.	
<b>Prioridad:</b>	Primaria

**Tabla 22. Descripción detallada del CUS: Iniciar Partida.**

<b>Nombre del Caso de Uso</b>	Iniciar partida.	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Darle inicio a la partida.	
<b>Resumen</b>	El caso de uso se inicia cuando un cliente registrado que se encuentra dentro de una partida que el mismo ha creado decide darle inicio a la misma, notifica al sistema la acción correspondiente y espera por la confirmación del resto de los participantes.	
<b>Referencias</b>	RF.5.3	
<b>Precondiciones</b>	Usuario previamente autenticado en el servidor y debe ser el creador de la partida.	
<b>Poscondiciones</b>	La partida es iniciada notificando a todos los usuarios del comienzo de la misma.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1- El cliente trasmite al sistema la petición correspondiente a dar inicio a una partida que el mismo ha creado.	1.1- El sistema recibe la petición y notifica a todos los jugadores participantes en la misma.	
<b>Puntos de Extensión:</b>		
<b>Prioridad:</b>	Primaria	

**Tabla 23. Descripción detallada del CUS: Pausar partida.**

<b>Nombre del Caso de Uso</b>	Pausar partida.	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Permitir al cliente que se encuentra activo en una partida pausarla.	
<b>Resumen</b>	El caso de uso se inicia cuando un jugador de una partida específica solicita pausar la misma, el sistema valida los datos proporcionados por el usuario y prosigue a notificar a los demás participantes de la partida que dicho usuario ha pausado la partida.	
<b>Referencias</b>	RF.5.4	
<b>Precondiciones</b>	Usuario previamente autenticado en el servidor.	
<b>Poscondiciones</b>	La partida es pausada.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1- El cliente indica al sistema que desea pausar la partida en la que se encuentra.	1.1- El sistema recibe la petición y chequea que la partida este activa, de ser así procede a ejecutar dicha petición y notificar a los demás usuarios que están en la misma que el juego ha sido pausado por dicho usuario.	
<b>Puntos de Extensión:</b>		

<b>Prioridad:</b>	Secundaria
-------------------	------------

**Tabla 24. Descripción detallada del CUS: Reanudar partida.**

<b>Nombre del Caso de Uso</b>	Resumir partida.	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Permitir al cliente que previamente pausó una partida resumirla nuevamente.	
<b>Resumen</b>	El caso de uso se inicia cuando un jugador que pausó una partida decide reiniciarla nuevamente.	
<b>Referencias</b>	RF.5.5	
<b>Precondiciones</b>	Usuario previamente autenticado en el servidor.	
<b>Poscondiciones</b>	La partida es resumida.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1- El cliente indica al sistema que desea reiniciar la partida que pausó previamente.	1.1- El sistema recibe la petición y chequea que la partida esté pausada y que el usuario sea quien la haya pausado, de ser así procede a ejecutar dicha petición y notificar a los demás usuarios que están en la misma que el juego ha sido reiniciado por dicho usuario.	
<b>Puntos de Extensión:</b> El usuario es notificado, en caso que no sea quien la haya pausado, que no tiene privilegios para realizar la operación solicitada.		
<b>Prioridad:</b>	Secundaria	

**Tabla 25. Descripción detallada del CUS: Abandonar partida.**

<b>Nombre del Caso de Uso</b>	Abandonar partida.	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Permitir al cliente que abandone una partida.	
<b>Resumen</b>	El caso de uso se inicia cuando un cliente registrado solicita al sistema abandonar una partida iniciada.	
<b>Referencias</b>	RF.5.6	
<b>Precondiciones</b>	Usuario previamente autenticado en el servidor.	
<b>Poscondiciones</b>	El resto de los usuarios son notificados que dicho usuario abandonó la partida.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1- El cliente indica al sistema que desea abandonar la partida en la que se encuentra.	1.1- El sistema recibe la petición y notifica a los demás jugadores la decisión tomada por el cliente.	
<b>Puntos de Extensión:</b>		
<b>Prioridad:</b>	Secundaria	

**Tabla 26. Descripción detallada del CUS: Listar puntuaciones.**

<b>Nombre del Caso de Uso</b>	Listar puntuaciones.	
<b>Actores</b>	Cliente	

<b>Propósito</b>	Permitir al cliente listar las mejores puntuaciones alcanzadas en un juego determinado.	
<b>Resumen</b>	El caso de uso se inicia cuando un cliente registrado solicita al sistema un listado de las mejores puntuaciones alcanzadas en un juego determinado. El sistema muestra el listado requerido.	
<b>Referencias</b>	R10.	
<b>Precondiciones</b>	Usuario previamente autenticado en el servidor.	
<b>Poscondiciones</b>	Lista de mejores puntuaciones.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>		<b>Respuesta del Sistema</b>
1- El cliente trasmite al sistema la petición de listar puntuaciones de un juego determinado.		1.1- El sistema recibe la petición y envía la información requerida.
<b>Puntos de Extensión:</b>		
<b>Prioridad:</b>	Secundaria	

**Tabla 27. Descripción detallada del CUS: Listar juegos disponibles por canal.**

<b>Nombre del Caso de Uso</b>	Listar juegos disponibles.	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Listar todos los juegos que han sido creados así como el estado del mismo. (Iniciado, disponible).	
<b>Resumen</b>	El caso de uso se inicia cuando un usuario registrado decide listar todos los juegos que se encuentran creados por otros usuarios, el sistema recibe la petición muestra el listado.	
<b>Referencias</b>	RF.6.2.	
<b>Precondiciones</b>	Usuario previamente autenticado en el servidor.	
<b>Poscondiciones</b>	Lista de juegos disponibles.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>		<b>Respuesta del Sistema</b>
1- El cliente solicita al sistema un listado de los juegos que se encuentran creados en el mismo.		1.1- El sistema recibe la petición y muestra el listado.
<b>Puntos de Extensión:</b>		
<b>Prioridad:</b>	Primaria	

**Tabla 28. Descripción detallada del CUS: Listar usuarios conector por canal.**

<b>Nombre del Caso de Uso</b>	Listar usuarios en línea.	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Listar todos los usuarios que se encuentran activos en un canal asignado a determinado juego.	
<b>Resumen</b>	El caso de uso se inicia cuando un usuario solicita ver todos los usuarios que se encuentran conectados dentro de su mismo canal.	
<b>Referencias</b>	RF.6.1	
<b>Precondiciones</b>	Usuario 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 cliente solicita al sistema un listado de los usuarios conectados a su canal.	1.1- El sistema recibe la petición y muestra el listado.
<b>Puntos de Extensión:</b>	
<b>Prioridad:</b>	Secundaria

## Anexo 2: Diagramas de clases del análisis

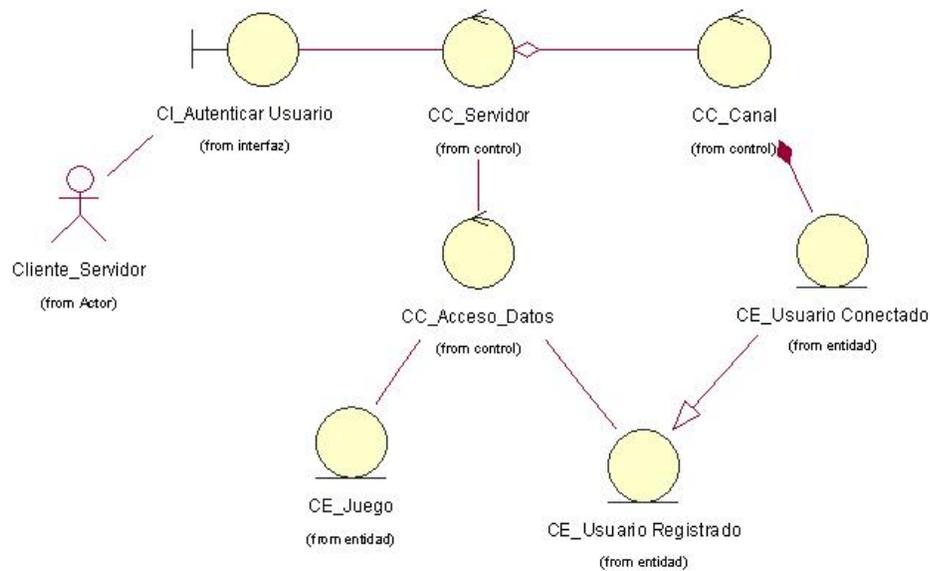


Figura # 7. Diagrama de Clases del análisis. CUS Autenticar Usuario.

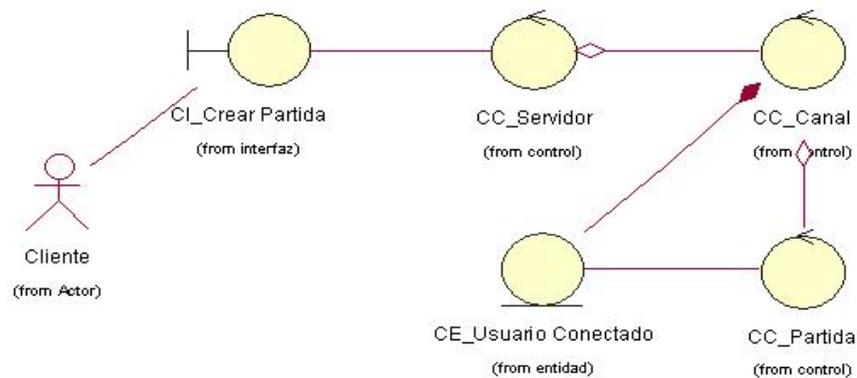


Figura # 8. Diagrama de Clases del análisis. CUS Crear Partida.

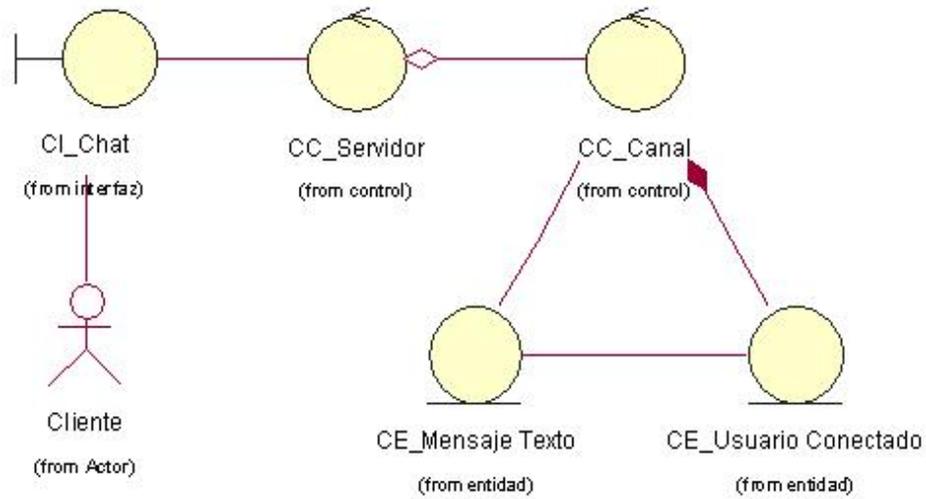


Figura # 9. Diagrama de Clases del análisis. CUS Enviar mensaje de texto.

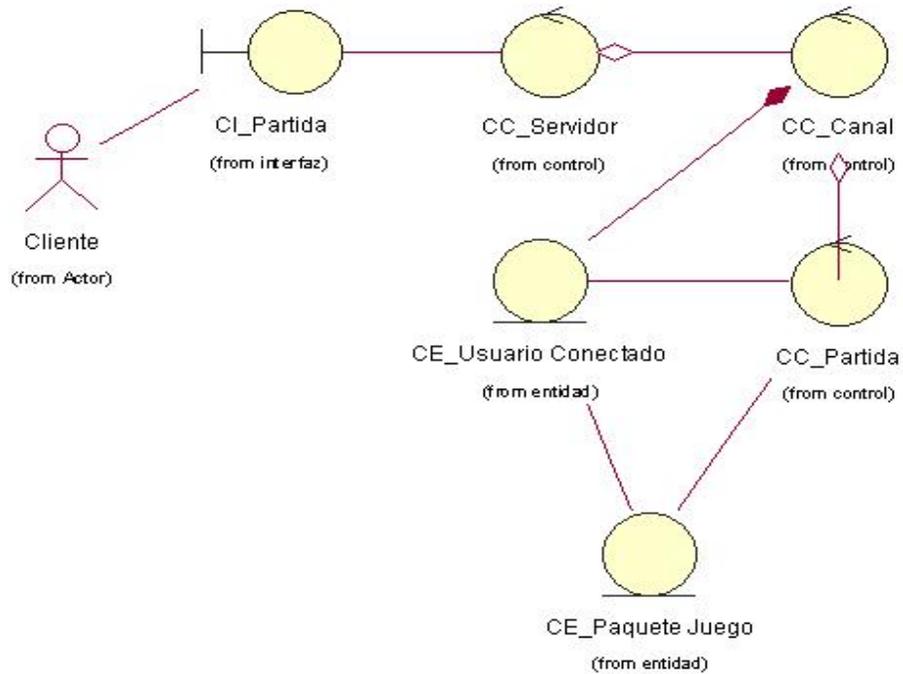
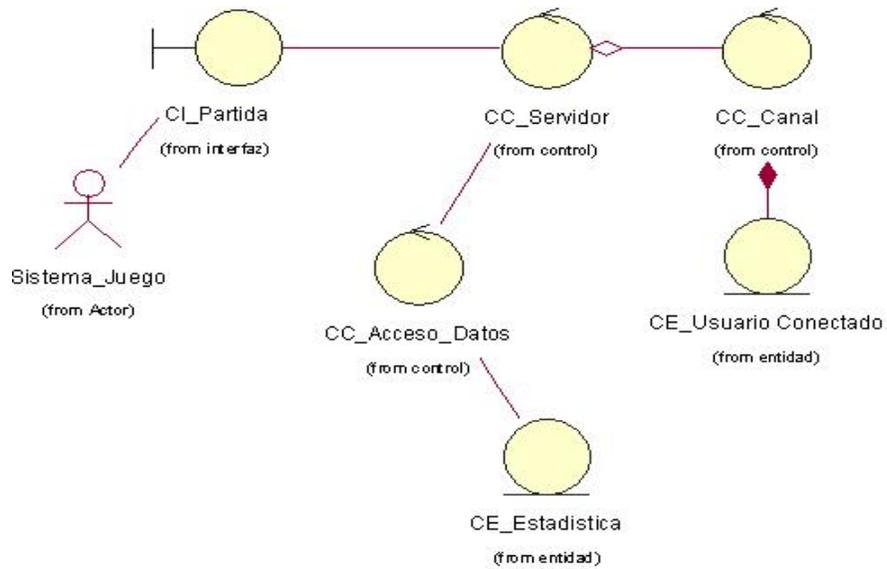


Figura # 10. Diagrama de Clases del análisis. CUS Enviar paquete de juego.



**Figura # 11 Diagrama de Clases del análisis. CUS Enviar resultado de la partida.**

Nota/ En aras de disminuir el tamaño del documento no se pusieron en el mismo todos los diagramas de clases del análisis, solo aquellos que resultaron arquitectónicamente mas significativos durante del desarrollo de la aplicación, los restantes se encuentran disponibles en formato digital.

## Anexo 2: Diagramas de clases del diseño

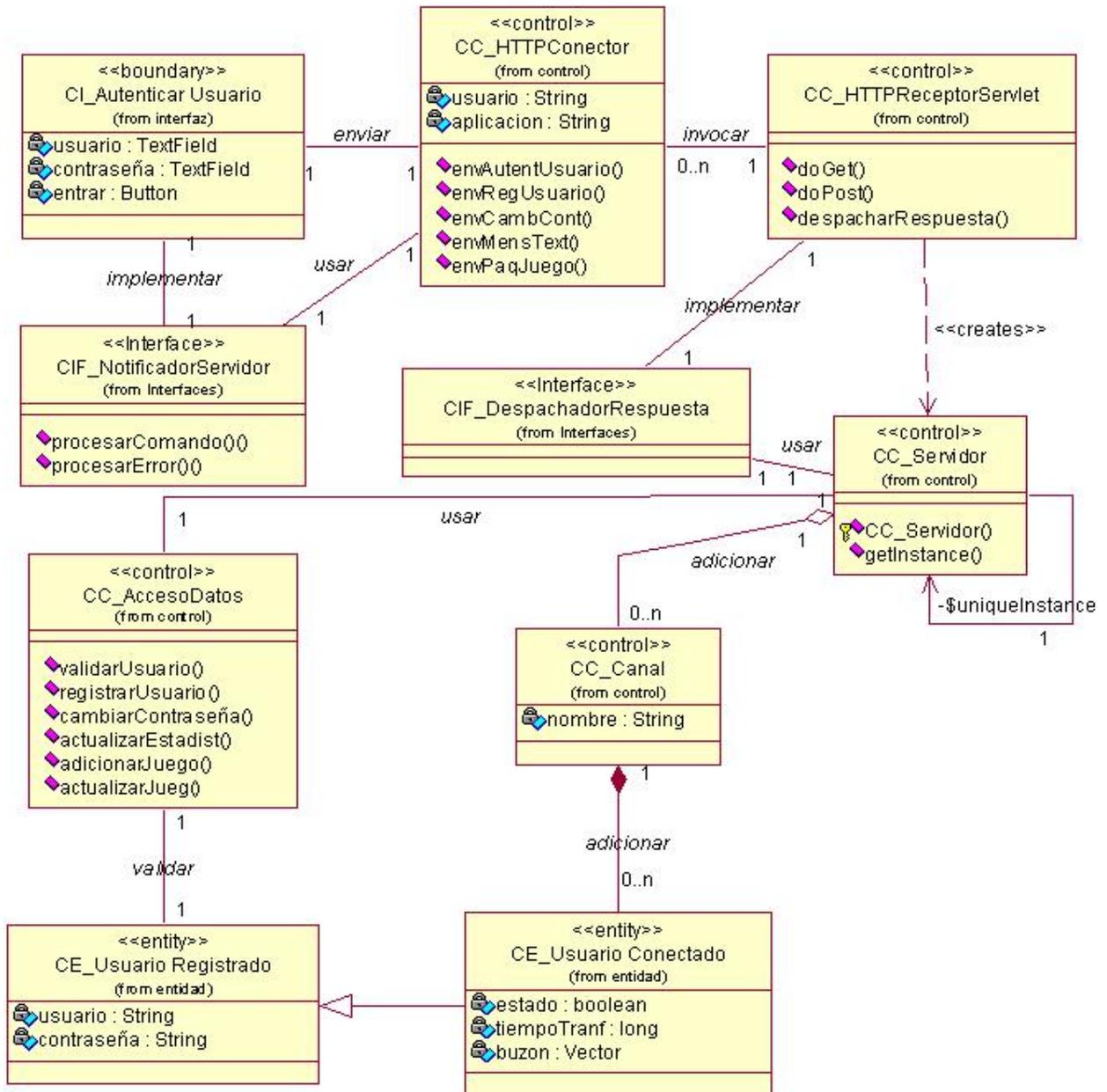


Figura # 12. Diagrama de Clases del diseño. Autenticar usuario.

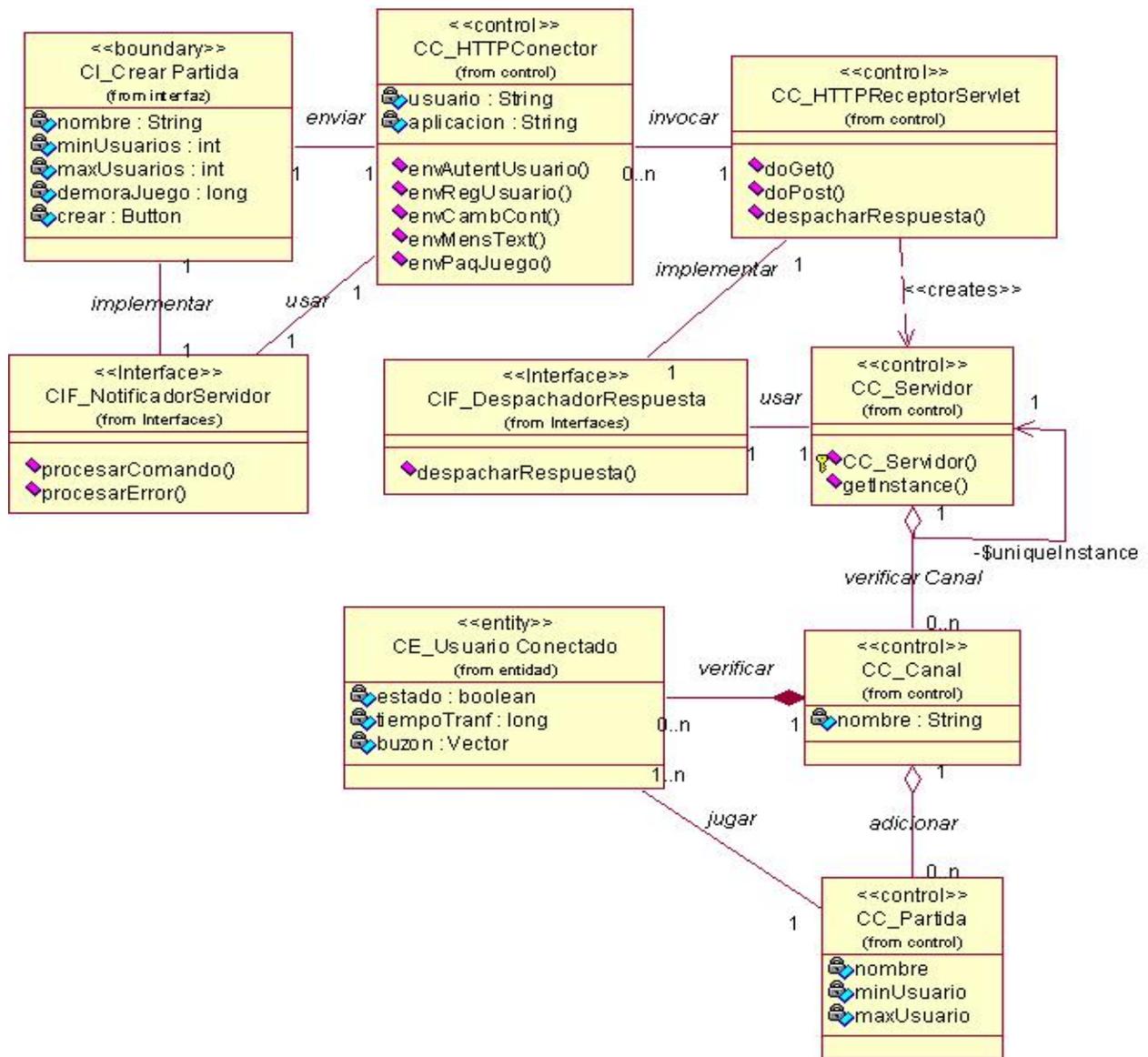


Figura # 13. Diagrama de Clases del diseño. CUS Crear partida.

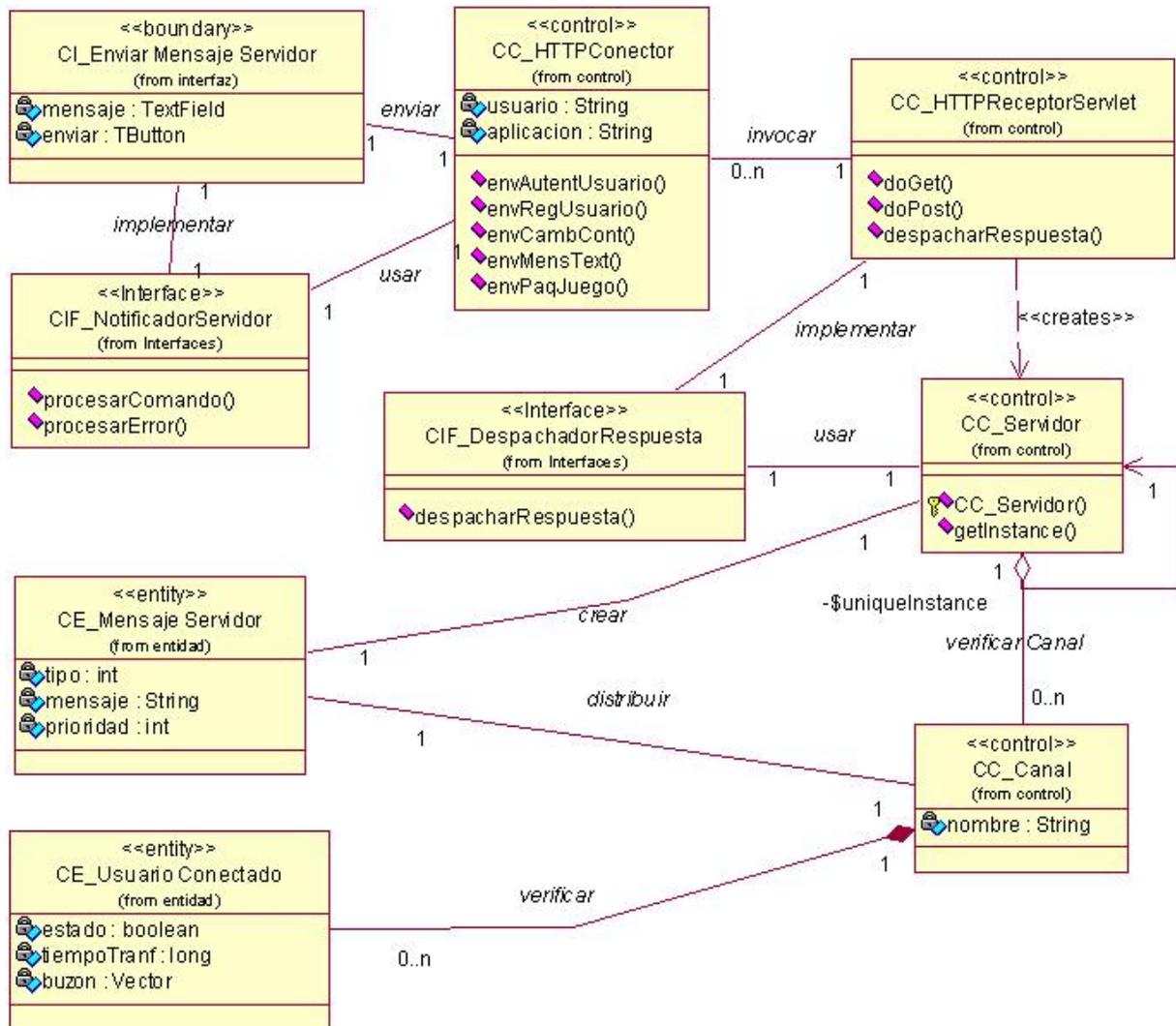


Figura # 14. Diagrama de Clases del diseño. CUS Enviar mensaje servidor.

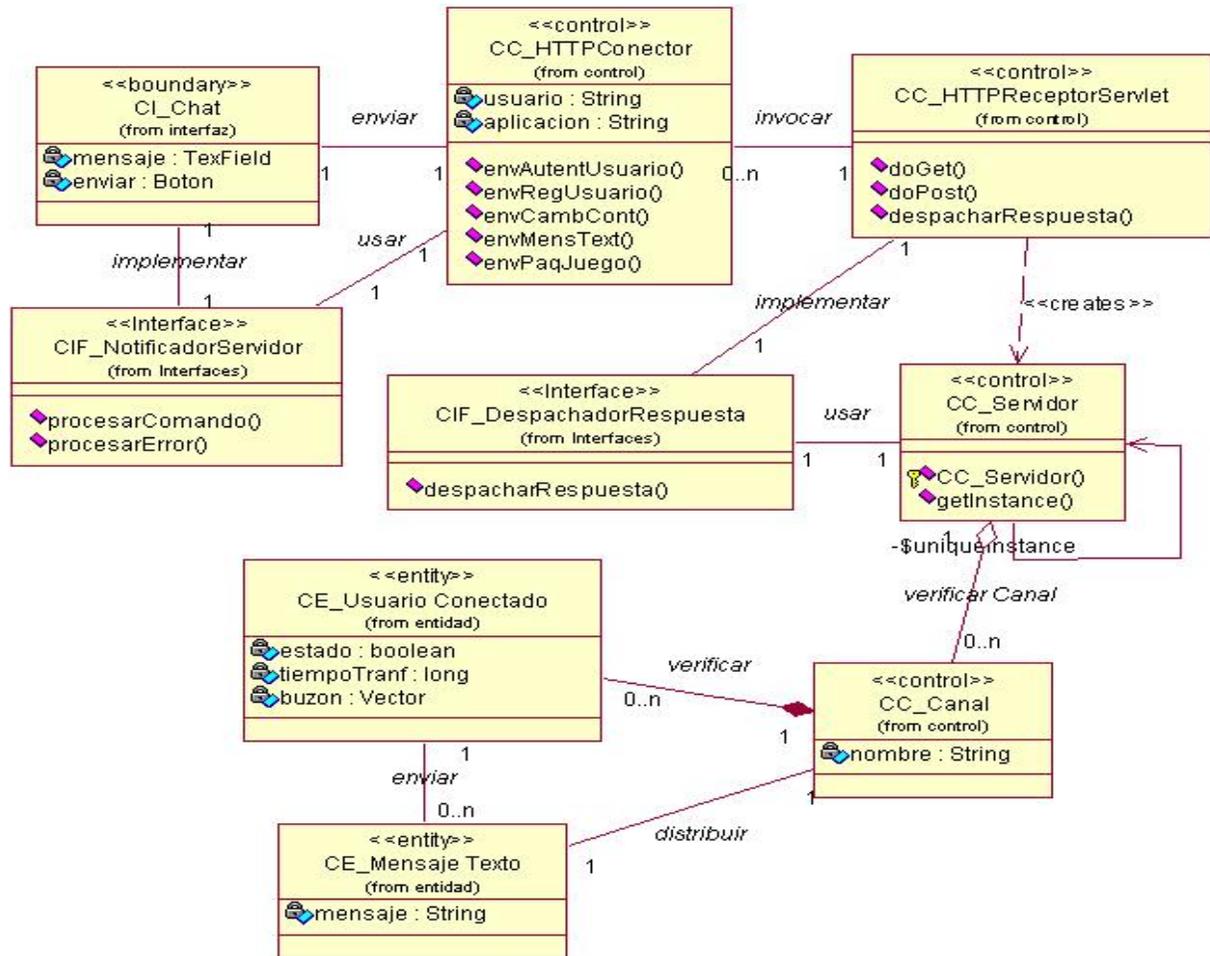


Figura # 15. Diagrama de Clases del diseño. CUS Enviar mensaje de texto.

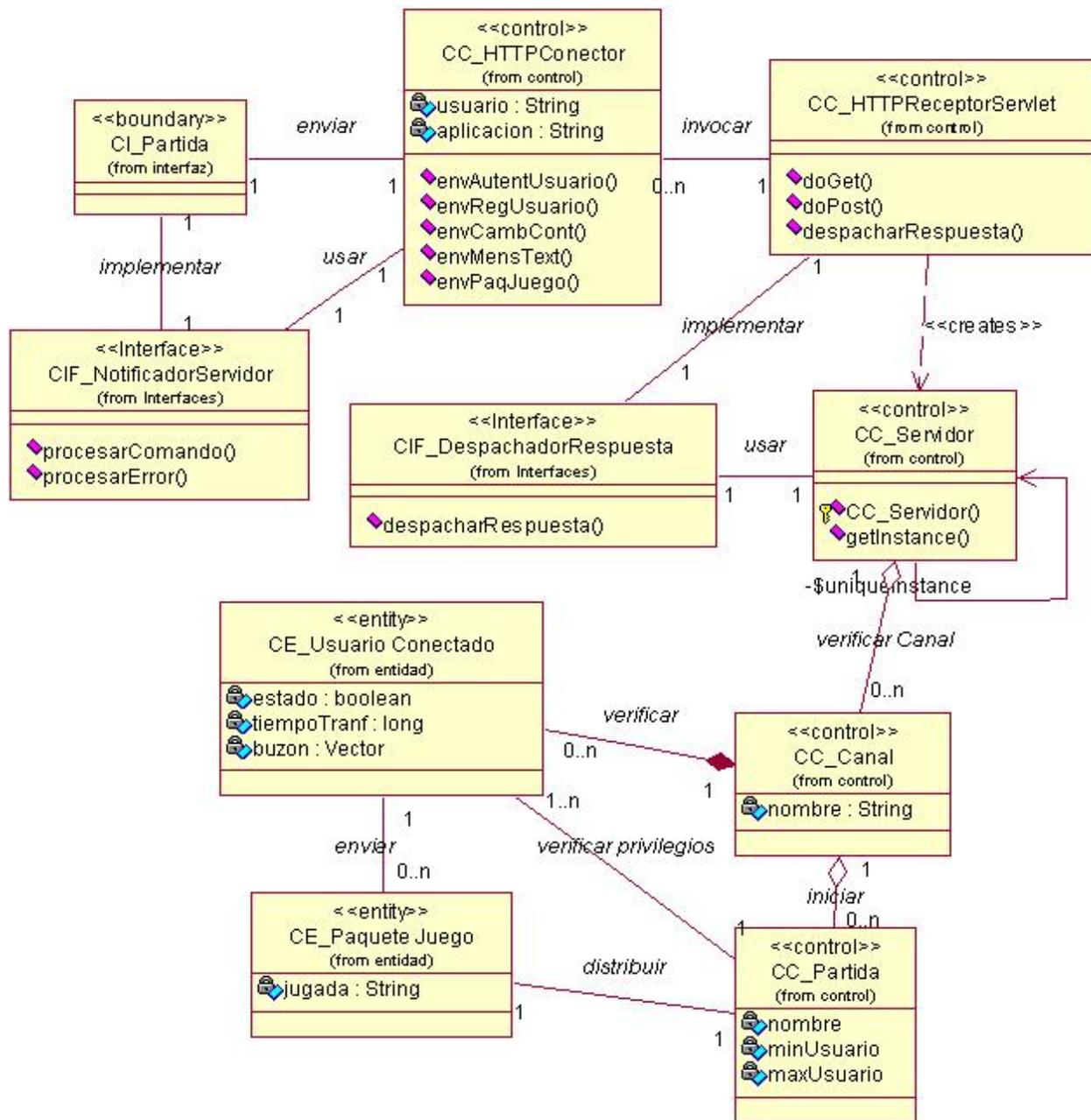


Figura # 16. Diagrama de Clases del diseño. CUS Enviar paquete de juego.

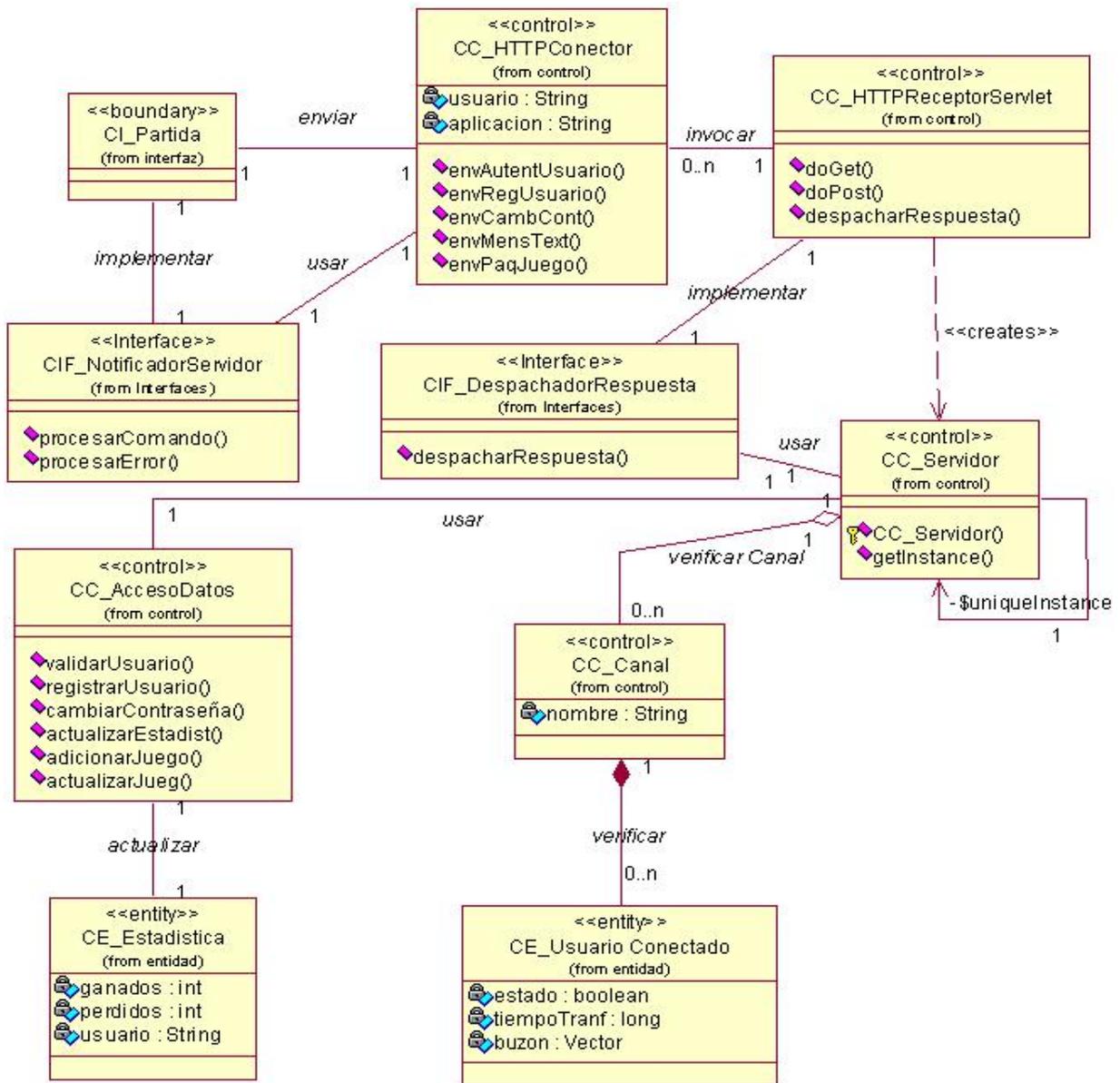


Figura # 17. Diagrama de Clases del diseño. CUS Enviar resultado de la partida.

Nota/ En aras de disminuir el tamaño del documento no se pusieron en el mismo todos los diagramas de clases del análisis, solo aquellos que resultaron arquitectónicamente mas significativos durante del desarrollo de la aplicación, los restantes se encuentran disponibles en formato digital.

## Anexo 2: Diagramas secuencias del diseño

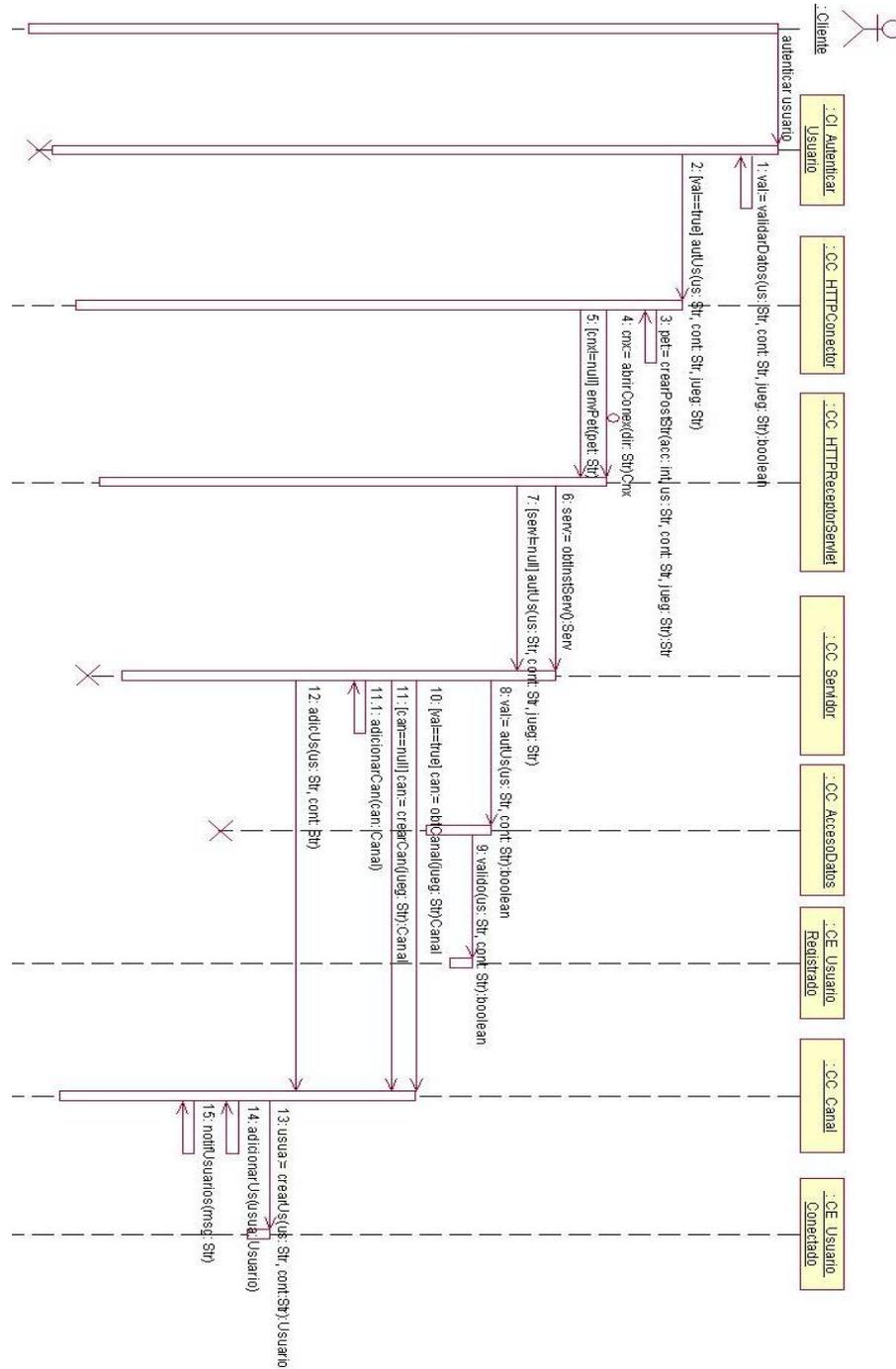


Figura # 18. Diagrama de secuencias del diseño. CUS Autenticar usuario.

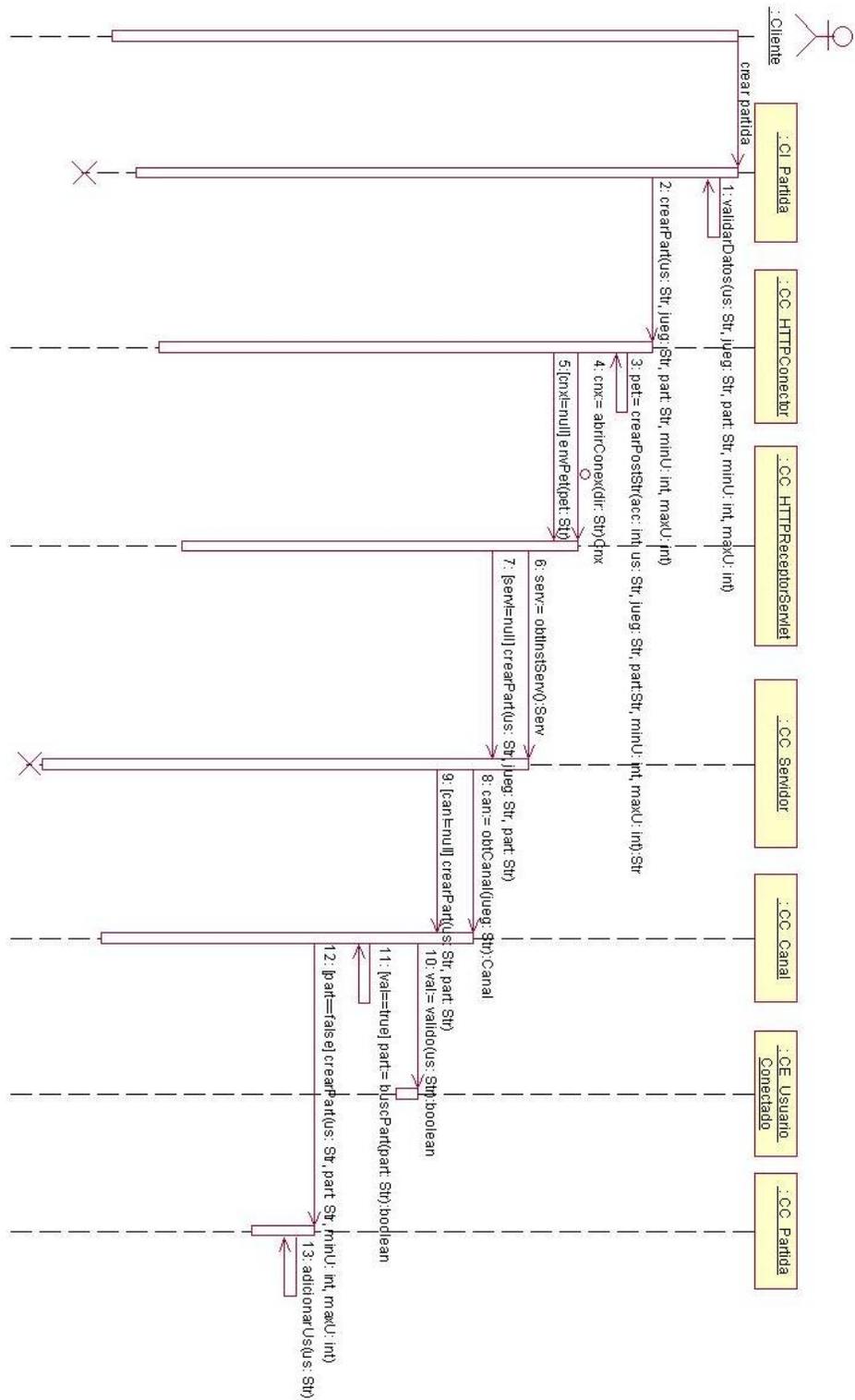


Figura # 19. Diagrama de secuencias del diseño. CUS Crear partida.

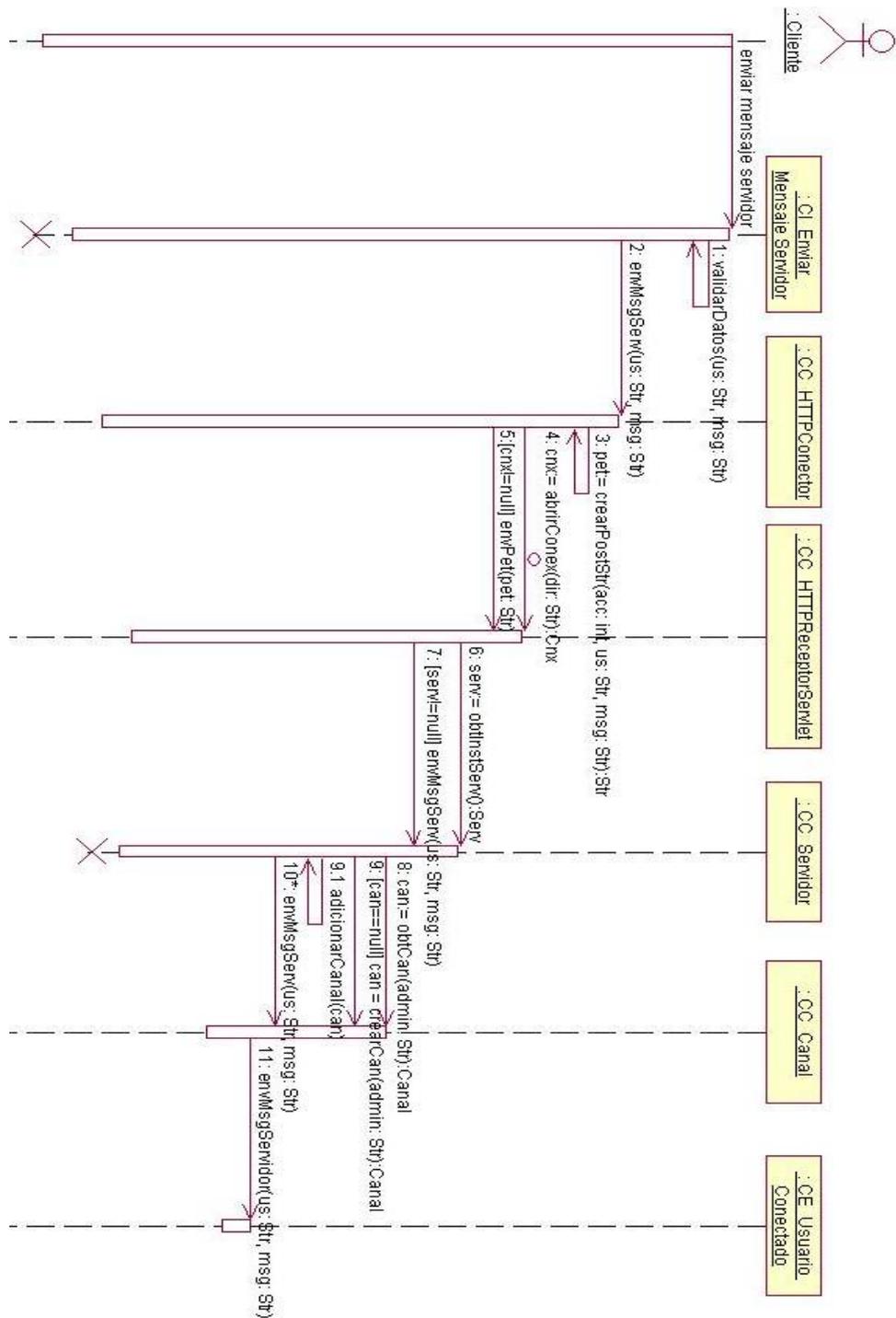


Figura # 20. Diagrama de secuencias del diseño. CUS Enviar mensaje servidor.

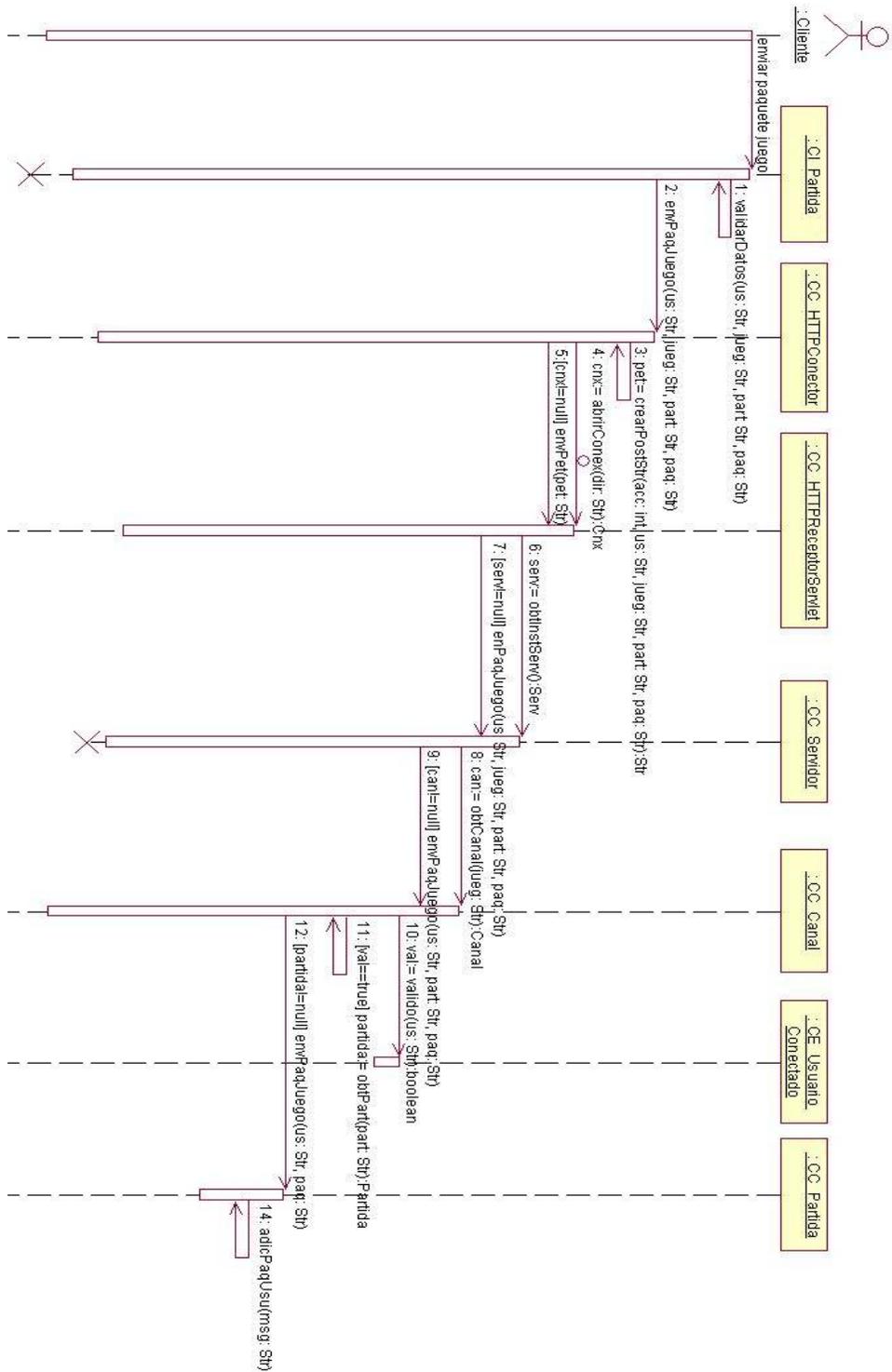


Figura # 21. Diagrama de secuencias del diseño. Enviar paquete de juego.

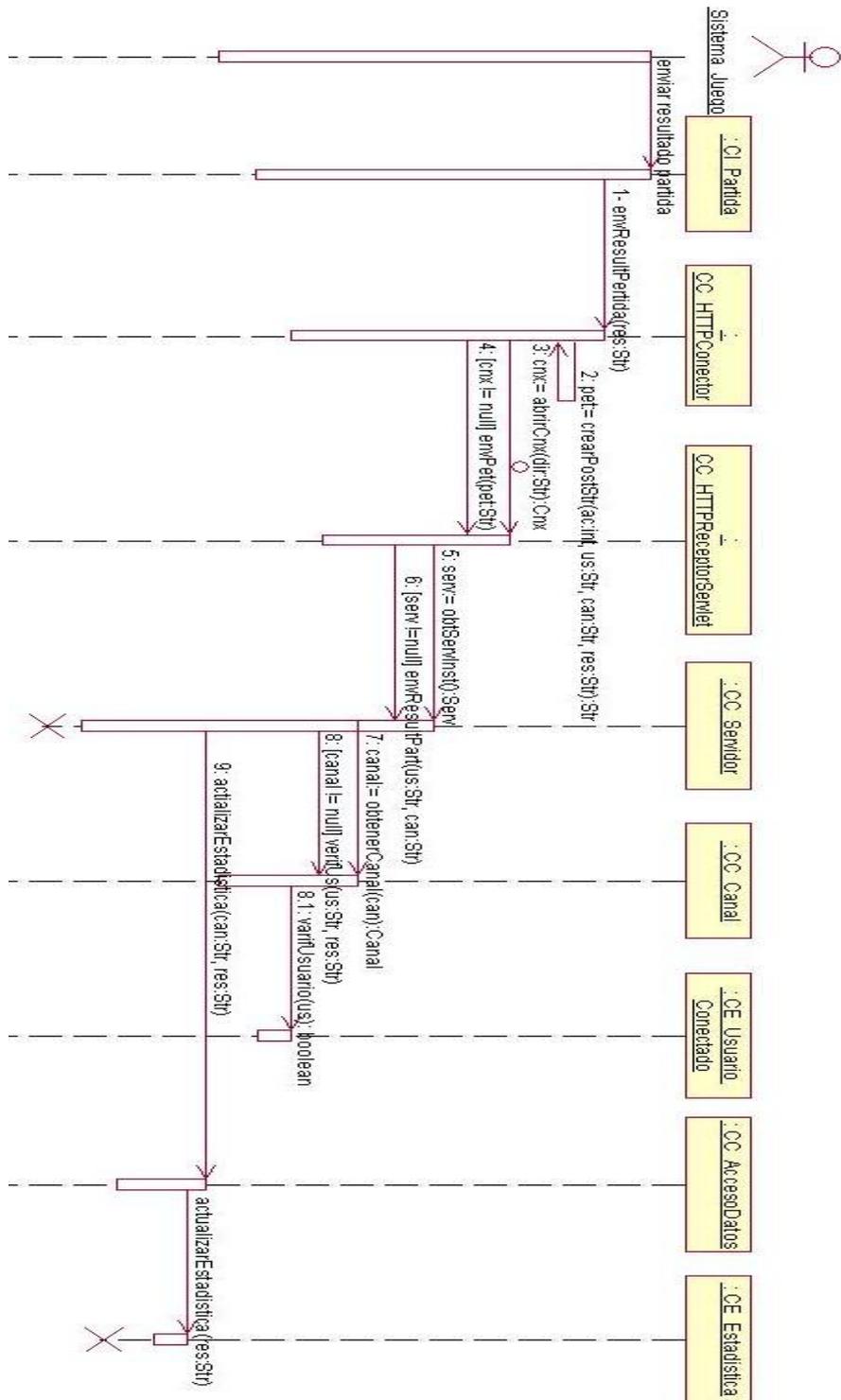


Figura # 22. Diagrama de secuencias del diseño. CUS Enviar resultado de la partida.

## 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.

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

**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

**HTML (Hyper Text Mark-up Language):** Lenguaje desarrollado por el CERN que sirve para modelar texto y agregarle funciones especiales (como hipervínculos). Es la base para la creación de páginas web tradicionales.

**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.

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

**IrDA (Infrared Data Association):** Organización con el fin de crear normas internacionales para el hardware y el software empleados en comunicaciones por infrarrojo, muy importante en comunicaciones inalámbricas.

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

**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.

**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.

**MD5:** El algoritmo toma una cadena de caracteres (por ejemplo una contraseña) y aplica una serie de transformaciones sobre esta, para producir una cadena de 32 caracteres y que es función del mensaje de entrada. Es un algoritmo irreversible.

**PDA (Personal Digital Assistant):** Ordenador de pequeño tamaño cuya principal función era, en principio, mantener una agenda electrónica. No obstante, cada vez más se va confundiendo con los ordenadores de mano y de palma.

**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.

**SDK (Software Development Kit):** Conjunto de herramientas y programas de desarrollo que permite al programador crear sus aplicaciones.

**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.

**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.

**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.

**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.

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

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

**Wi-Fi (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.

**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.