

Universidad de las Ciencias Informáticas

Facultad 2



Título: SIRIO Mobile 2.0, Plataforma para la gestión de juegos multiusuario desde dispositivos móviles vía GPRS.

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

Autores: Liane Figueroa Hernández.
Erick Pérez Castillo.

Tutores: Ing. Denys Buedo Hidalgo.
Ing. Yunisel Viera Vargas.

Consultor: MSc. Pedro Angel Romero.

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 2008.

Liane Figueroa Hernández
Firma del Autor

Erick Pérez Castillo
Firma del Autor

Ing. Denys Buedo Hidalgo
Firma del Tutor

Ing. Yunisel Viera Vargas
Firma del Tutor

AGRADECIMIENTOS

A mi familia, el tesoro más grande que tengo, sin ellos nada hubiese sido posible, gracias a su amor infinito:

A mimá, por tanta entrega, por permitirme ser su Gotica. A abu, por ayudarme tanto cuando estoy en aprietos, por estar siempre a mi lado. A mi mamita linda, por ser la mejor del mundo. A mi tata, por quererme como a una hija. A mi papá por siempre tenerme en su corazón. A mi hermanito, por su sonrisa. A mis tíos, tías y primos, los más lindos del mundo.

A Pedro, por todo su amor y ayuda, por la felicidad que tengo a su lado.

A mis amigos, Mage, Alela, Jie, Odalys, los que siempre están. A los nuevos, a los que me ayudaron a seguir, Roci, Yeyita, Naty, Leyi, Norbe, Franklin, Marcos, Carlos, fue bello tenerlos a todos. A los chitis por momentos tan divertidos.

A mis profes, los especiales, los que sembraron muchas cosas en mi, Fernando, Maykel y José Carlos, los quiero mucho. A Abdel por ayudarme tanto. A Cao por su paciencia y apoyo. A Ramsés por todo el conocimiento que me aportó.

A Procyon, por darme la posibilidad de aprender tantas cosas, me encantó formar parte de ese proyecto, en especial a York.

A Erick, por aguantarme, por ser tan responsable y noble, te quiero.

A la UCI por darme la posibilidad de hacer este sueño realidad.

A todos los que me han ayudado y han estado a mi lado, gracias.

Liane

A los tutores por estar siempre disponibles, y brindarnos toda la ayuda necesaria.

A mi familia y en especial a mi madre que no se encuentra en Cuba por estar cumpliendo misión internacionalista, a mi padre que me guió siempre por el buen camino y a mi hermano querido.

A la UCI y a todos los profesores que de una forma u otra han aportado en mi formación como un profesional.

A Procyon y a todos los trabajadores, por darme la posibilidad de aprender tanto en todo el tiempo que estuve con ellos. A Orlando por ser tan comprensivo con los problemas que me daba la computadora.

A mis compañeros de proyecto, a la Riky por darme seis razones para no olvidarme de ellos, a Liane por ser mi compañera en los buenos y malos momentos que pasamos, un beso, a Yeilin por encontrar el símbolo del grupo (el pantalón “kiss”), al yuma por ser el terror del RapidShare, a Marquette por ser el niño del grupo, al flaco por ser el abusador, a Joan por estar al tanto de lo último en cuanto a fútbol, películas y series, a David por ser la conciencia de los varones, a Kmix por gustarle el rosado, a EJB por ser de Pinar y saber tanto y a Rigo porque con él aprendí muchas cosas, a todos gracias por ser capaces de aguantarme tanto tiempo.

A mis antiguos compañeros de los grupos 2108 y 2202 que tan buenos momentos me hicieron pasar, a todos los que por una razón u otra no se encuentran en la UCI.

A Yaimara por ser el eje inspirador y motivo de orgullo de todos mis logros, por ser tan buena a lo largo de todos estos años, por haber logrado en mi tanto en tan poco tiempo, por aguantar los malos momentos que pasamos y por ser como es.

A todos los que de una forma u otra me han ayudado y por descuido se me han olvidado mencionar, a todos ustedes mil gracias y espero que sepan que nunca los olvidaré.

Erick

DEDICATORIA

A mi familia linda, merecedora de todos los méritos,

A mis amigos y al amor de mi vida.

A mi Comandante.

Liane

A toda mi familia que le debo todo lo que soy.

A mi madre por estar cumpliendo misión internacionalista.

A todos los buenos amigos que he tenido en estos años en la UCI.

A los amigos de toda la vida que de una forma u otra siempre me han acompañado.

Quiero dedicársela a Liane por todos los trabajos que hemos pasado y por ser tan buena amiga.

Al amor de mi vida que siempre ha estado ahí para apoyarme en los momentos que más la necesité.

Erick

RESUMEN

El adelanto de la ciencia y la tecnología ha permitido incrementar las capacidades de hardware y software de los teléfonos celulares, posibilitando el desarrollo de aplicaciones destinadas a ellos, la producción de juegos constituye una de las áreas de mayor importancia.

En Procyon se desarrolló una plataforma llamada SIRIO Mobile, destinada a la gestión de juegos multiusuario en dispositivos móviles vía GPRS. Se plantearon algunas insuficiencias en la misma y se desarrolló mediante esta investigación una segunda versión que incorpora las nuevas funcionalidades de salvar el estado de las aplicaciones y de iniciar partidas con jugadores virtuales.

Se amplió un API de conexión con el servidor, desarrollado en la primera versión donde se añaden nuevas características para cumplir con los objetivos planteados. También se obtuvo un API de conexión para servidores de aplicaciones que gestionen la lógica de los juegos y permitan persistir los datos de las partidas salvadas

Para ello se realizó un estudio de la Telefonía Celular y las Comunicaciones Inalámbricas, además se analizaron productos similares de plataformas de desarrollo de juegos para móviles, así como las herramientas necesarias para la extensión de la Plataforma.

Se mantuvieron, mediante el ciclo de vida del producto las mejores prácticas de la Ingeniería de Software, obteniendo los artefactos que se deben generar. Se tuvieron en cuenta aspectos de seguridad que garantizan la confianza de la Plataforma.

Como parte del proceso de pruebas se implementó un juego por turno denominado TicTacToe, mediante el cuál se puede comprobar el correcto funcionamiento del producto.

PALABRAS CLAVES

GPRS, Plataforma, Multiusuario, API, Teléfonos Celulares, Juegos.

TABLA DE CONTENIDOS

AGRADECIMIENTOS.....	III
DEDICATORIA.....	V
RESUMEN.....	VI
PALABRAS CLAVES.....	VI
INTRODUCCIÓN.....	5
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	8
1.1 INTRODUCCIÓN.....	8
1.2 TELEFONÍA CELULAR O MÓVIL.....	8
1.2.1 Funcionamiento.	8
1.2.2 Usos del Teléfono Móvil.....	9
1.2.3 Generaciones de comunicación móvil	10
1.3 COMUNICACIONES INALÁMBRICAS.....	12
1.3.1 GSM.....	13
1.3.2 GPRS.....	13
1.3.3 UMTS.....	14
1.4 COMUNICACIÓN POR SOCKET.....	16
1.5 PROTOCOLO DE COMUNICACIÓN	17
1.5.1 HTTPS.....	18
1.6 SIRIO MOBILE 1.0.....	19
1.7 PLATAFORMAS DE DESARROLLO DE JUEGOS PARA MÓVILES.....	20
1.7.1 Productos similares.....	21
1.8 ARQUITECTURA CLIENTE – SERVIDOR.....	24
1.9 PLATAFORMA DE DESARROLLO Y LENGUAJE DE PROGRAMACIÓN.....	25
1.10 SERVIDOR WEB.....	26
1.11 PERSISTENCIA DE DATOS.....	27
1.11.1 XML.....	27
1.11.2 MySQL Server.	28
1.12 METODOLOGÍA DE DESARROLLO EMPLEADA.....	29
1.13 HERRAMIENTAS DE DESARROLLO EMPLEADAS.....	30
1.13.1 Herramientas de modelado de UML.....	30
1.13.2 Entorno Integrado de Desarrollo.....	32
1.13.3 Herramienta para el control de versiones.	32
1.14 CONCLUSIONES.....	34
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	35
2.1 INTRODUCCIÓN.....	35
2.2 COMPARACIÓN ENTRE LAS PLATAFORMAS DE DESARROLLO DE JUEGOS PARA MÓVILES Y SIRIO MOBILE.....	35
2.3 PROPUESTA DEL SISTEMA.....	36
2.4 MODELO DEL DOMINIO.....	37

2.4.1	Descripción de los conceptos del dominio.....	39
2.5	LEVANTAMIENTO DE REQUISITOS.....	40
2.5.1	Requisitos funcionales.....	40
2.5.2	Requisitos no funcionales.....	41
2.6	MODELO DE CASOS DE USO DEL SISTEMA.....	42
2.6.1	Actores del sistema.....	42
2.6.2	Casos de uso del sistema.....	43
2.6.3	Descripción de los Casos de uso del sistema.....	46
2.7	CONCLUSIONES.....	48
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....		49
3.1	INTRODUCCIÓN.....	49
3.2	MODELO DE CLASES DEL ANÁLISIS.....	49
3.2.1	Diagrama de clases de análisis.....	49
3.2.2	Diagramas de colaboración del análisis.....	50
3.3	MODELO DE CLASES DEL DISEÑO.....	51
3.3.1	Diagramas de clases del diseño.....	51
3.3.2	Diagramas de secuencia del diseño.....	53
3.4	DECISIONES DE DISEÑO EMPLEADAS EN EL DESARROLLO DE LA APLICACIÓN.....	54
3.5	DISEÑO DE BASE DE DATOS.....	57
3.5.1	Diagrama de Clases Peristentes.....	58
3.5.2	Diagrama Entidad Relación.....	59
3.6	TRATAMIENTO DE ERRORES.....	59
3.7	SEGURIDAD.....	61
3.8	CONCEPCIÓN DE LA AYUDA.....	62
3.9	CONCLUSIONES.....	62
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.....		63
4.1	INTRODUCCIÓN.....	63
4.2	MODELO DE IMPLEMENTACIÓN.....	63
4.2.1	Diagrama de Componentes.....	63
4.2.2	Diagrama de Despliegue.....	64
4.3	MODELO DE PRUEBAS.....	65
4.3.1	Métodos de Prueba.....	65
4.3.2	Diseño de casos de prueba. Prueba de Caja Negra.....	66
4.4	CONCLUSIONES.....	69
CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD.....		70
5.1	INTRODUCCIÓN.....	70
5.2	PLANIFICACIÓN BASADA EN CASOS DE USO.....	70
5.3	BENEFICIOS TANGIBLES E INTANGIBLES.....	77
5.4	ANÁLISIS DE COSTOS Y BENEFICIOS.....	78
5.5	CONCLUSIONES.....	79
CONCLUSIONES.....		80
RECOMENDACIONES.....		81

BIBLIOGRAFÍA.....	82
ANEXOS	85
GLOSARIO	98

INTRODUCCIÓN

Actualmente la portabilidad e inmediatez que otorga la telefonía celular, la convierte en una de las tecnologías más aceptadas a nivel mundial. Su uso asciende anualmente, llegando a los 3.000 millones de dispositivos, lo cuál equivale a la mitad de la población de la Tierra. (Penalva, 2007)

Los celulares se han convertido prácticamente en computadoras de bolsillo. El desarrollo de la ciencia y la tecnología ha permitido disminuir su tamaño y peso, obtener baterías más pequeñas y de mayor duración, pantallas más nítidas y de colores, e incorporación de software más amigable.

Este avance ha hecho que estos dispositivos incorporen funciones que no hace mucho parecían futuristas. Entre las posibilidades más atractivas que brindan se encuentra la incorporación de juegos, lo cuál ha propiciado el surgimiento de empresas dedicadas al desarrollo de este tipo de aplicaciones, con el fin de explotar al máximo el factor entretenimiento.

En Procyon, entidad que se encuentra enmarcada dentro de la infraestructura productiva de la Universidad de las Ciencias Informáticas (UCI), se fomenta el uso de las nuevas tecnologías de comunicación y la producción de aplicaciones y contenidos para móviles; en ella se desarrollan juegos, de tipo simple y multiusuario. Estos últimos utilizan distintas vías de comunicación, la tecnología Bluetooth¹, para distancias cortas, y la tecnología de servicio general de paquetes de radio, GPRS (del inglés: General Packet Radio Service), para mayores distancias.

La Plataforma SIRIO Mobile 1.0, es el medio que se usa en la entidad Procyon para que dos o más celulares jueguen por vía GPRS, estableciendo, mediante una sala de chat, las reglas del juego.

Sin embargo existen una serie de insuficiencias en el modelo actual. El jugador tiene que esperar a que otro esté en línea para poder hacer uso de una aplicación multiusuario, ya que la Plataforma funciona de intermediario entre dos jugadores conectados que tengan en sus dispositivos la misma aplicación. Además una vez que alguno se ausente del juego se pierde el estado que había alcanzado, imposibilitando que lo recupere en otro momento.

¹ Es un estándar inalámbrico que utiliza la tecnología de radio para conectar todo tipo de dispositivo. Se utiliza para distancia cortas, entre 10 y 30 m.

Por lo que se hace necesario llevar a cabo la puesta en marcha de la investigación con el fin de resolver el problema de la entidad Procyon de necesitar que la Plataforma SIRIO Mobile inicie partidas con jugadores virtuales y permita mantener el estado de las aplicaciones que maneja.

El objeto de estudio de la investigación se enmarca en la Telefonía Móvil, las Comunicaciones Inalámbricas, y el paradigma Cliente – Servidor. Siendo su campo de acción las plataformas de desarrollo de juegos para móviles, que se comuniquen vía GPRS, y como un caso particular, la primera versión de SIRIO Mobile.

Si se realiza un estudio de la Telefonía Móvil, las Comunicaciones Inalámbricas, la arquitectura Cliente – Servidor, así como de las tendencias actuales de las plataformas de desarrollo de juegos para celulares y particularmente de SIRIO Mobile 1.0, y atendiendo a dichos análisis se extiende la Plataforma incorporando: partidas con jugadores virtuales y el mantenimiento del estado de las aplicaciones; entonces los usuarios no se verán obligados a esperar en línea por otros jugadores y no perderán el estado de sus aplicaciones al ausentarse.

El objetivo general de este trabajo es extender la Plataforma SIRIO Mobile con el fin de iniciar partidas con jugadores virtuales y mantener el estado de sus aplicaciones; haciendo uso de las mejores prácticas de Ingeniería del Software y de las herramientas más adecuadas.

Tributando al cumplimiento del mismo se trazan los siguientes objetivos específicos:

- Documentar las características de la nueva Plataforma, así como las decisiones de diseño y su proceso de obtención.
- Implementar una aplicación con el fin de probar los resultados de la Plataforma SIRIO Mobile 2.0.

Para llegar a estos objetivos se ha de cumplir con las siguientes tareas:

1. Estudiar y documentar el estado del arte en el campo de la Telefonía Celular y las Comunicaciones Inalámbricas para valorar el desarrollo que han alcanzado y enfocar esta investigación en las tecnologías más idóneas.
2. Estudiar y documentar las tendencias de las plataformas de desarrollo de juegos para móviles para obtener buenas prácticas e identificar deficiencias.

3. Analizar la Plataforma SIRIO Mobile 1.0 con el fin de obtener una nueva versión que rehúse sus mejores prácticas y corrija sus deficiencias.
4. Estudiar las principales metodologías de desarrollo y herramientas para obtener un producto que se cumpla con las características de una plataforma de desarrollo de juegos multiusuario para móviles vía GPRS.
5. Diseñar e implementar una arquitectura que satisfaga los objetivos planteados.
6. Aplicar los 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 por cinco capítulos, descritos a continuación:

Capítulo 1: Fundamentación teórica: Se realiza un estudio del estado del arte de las Comunicaciones Inalámbricas y la Telefonía Móvil. Se analizan los productos similares de plataformas de desarrollo de juegos para móviles, así como las herramientas necesarias para la extensión de la Plataforma SIRIO Mobile 1.0.

Capítulo 2: Características del Sistema: Se describen los requerimientos de la Plataforma SIRIO Mobile 2.0 y se determinan los casos de uso más importantes. Se realiza una propuesta del Sistema.

Capítulo 3: Análisis y Diseño del Sistema: Se generan los artefactos relacionados con el análisis y diseño, se determinadas las decisiones de diseño. Se manejan los posibles errores de la Plataforma y se hace un análisis de la seguridad de SIRIO Mobile 2.0

Capítulo 4: Implementación: Se generan los artefactos relacionados con la implementación. Se determinan los casos de prueba.

Capítulo 5: Estudio de Factibilidad: Se realiza un análisis de la factibilidad del Sistema y de los costos que genera la Plataforma a desarrollar. Se hace un análisis de los beneficios que genera.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 INTRODUCCIÓN

En el presente capítulo se hace un estudio acerca de la Telefonía Móvil y las Comunicaciones Inalámbricas, ubicando la investigación en las más actuales tecnologías, pudiendo hacer uso de ellas para lograr cumplir el objetivo fundamental de este trabajo. Se analizan los productos similares de plataformas de desarrollo de juegos para móviles con el fin de acoplar experiencias que puedan ser incorporadas en la aplicación a desarrollar. Además se investigan las herramientas más adecuadas para obtener la segunda versión de la Plataforma.

1.2 TELEFONÍA CELULAR O MÓVIL

Los teléfonos celulares han revolucionado el área de las comunicaciones por sus ventajosas prestaciones. Su funcionamiento se ha desarrollado hasta convertirse en productos de consumo masivo y su aceptación se hace evidente a medida que transcurren los años.

1.2.1 Funcionamiento.

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 móviles) que permiten el acceso a dicha red.

El teléfono móvil es un dispositivo de comunicación electrónico con las mismas capacidades básicas de un teléfono de línea telefónica convencional. Además de ser portátil es inalámbrico, al no requerir cables conductores para su conexión a la red telefónica.

La Red de Telefonía Celular es una red de radiofrecuencia compuesta por áreas definidas (Células), cubiertas por un transmisor fijo (Radio Base), y enlazada a una red telefónica pública, con el objetivo de proporcionar un servicio de telefonía móvil totalmente dúplex² y una serie de aplicaciones adicionales de voz y datos. La Figura 1 muestra la forma en que se distribuyen las células o también llamadas celdas.

² Utilizado en las telecomunicaciones para definir a un sistema que es capaz de mantener una comunicación bidireccional, enviando y recibiendo mensajes de forma simultánea.

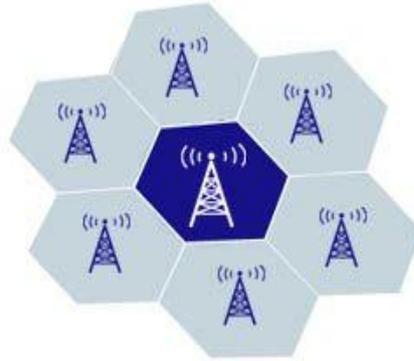


Figura 1: Distribución de Celdas.

Cada celda tiene una estación base que consiste de una torre y un pequeño edificio que contiene el equipo de radio.

Cuando el usuario desea realizar una llamada, el teléfono celular envía un mensaje a la torre, solicitando una conexión a un número de teléfono específico. Si la torre dispone de los suficientes recursos para permitir la comunicación, un dispositivo llamado "switch"³ conecta la señal del teléfono celular a un canal de la red de telefonía pública. La llamada en este momento toma un canal inalámbrico así como un canal en la red de telefonía pública que se mantendrán abiertos hasta que la llamada se concluya. (Lizama, 2007)

1.2.2 Usos del Teléfono Móvil

El teléfono celular surge en primera instancia como un medio para resolver la necesidad básica de la comunicación a distancia de una manera práctica, permitiendo la movilidad del instrumento. Sin embargo, se ha transformado en un objeto personal, y más aún, en un accesorio de moda. Además de la comunicación telefónica, el celular ofrece a su propietario una amplia gama de propuestas y servicios tendientes a satisfacer diferentes tipos de ansiedades. En este contexto, la creciente demanda dentro de la población y la gran cantidad de servicios móviles que proponen las empresas que proveen los servicios conducen a crear una nueva identidad personal de los usuarios. El teléfono es además de un instrumento de comunicación, un objeto de entretenimiento en el marco de los últimos adelantos tecnológicos.

Algunos de los usos alternativos del teléfono móvil son:

³ Administrador inteligente del ancho de banda.

Agenda, despertador, calculadora, cronómetro, cámara de fotos y video, servicio de localización, televisión en tiempo real, Internet, entre otros.

La finalidad del teléfono celular se ha trasladado de un medio de comunicación a un centro de entretenimiento, y su adquisición no siempre implica el simple hecho de comunicarse, sino también de beneficiarse con las ventajas tecnológicas que posee. (Lizama, 2007)

1.2.3 Generaciones de comunicación móvil

Los sistemas de telefonía móvil se han ido desarrollando, clasificándose en distintas generaciones dependiendo del grado de evolución técnica de los mismos.

La generación 0 (0G) es la radiocomunicación de dos vías, usando el método para hablar en línea, PTT (del inglés: Push to Talk) y el método de comunicación MTS (del inglés: Mobile Telephone System).

La Primera Generación (1G) hizo su aparición en 1979 y representa el surgimiento de la telefonía móvil. Opera bajo una plataforma analógica y es estrictamente para voz, las medidas preventivas no formaban parte de esta primitiva celular. La tecnología predominante de esta generación era AMPS (del inglés: Advanced Mobile Phone System).

Al ser insuficiente el espectro de frecuencia para soportar la calidad de servicio que se requería, surgió en 1990 la Segunda Generación (2G). Representando la transición de la telefonía móvil analógica a la telefonía móvil digital con todos sus valores agregados (mensajería, datos, etc.) El sistema 2G utiliza protocolos de codificación más sofisticados y se emplea en los sistemas de Telefonía Celular actuales. Las velocidades de información por voz son más altas, pero limitados en comunicación de datos. Con respecto a la anterior, la mayoría de los protocolos de 2G ofrecen diferentes niveles de encriptación.

Esta tecnología se clasifica de acuerdo al método en que se establece la comunicación y el acceso al canal. Primeramente surge TDMA (del inglés: Time Division Multiple Access) en la que varios usuarios utilizan el mismo canal dividiendo y transmitiendo la información en diferentes segmentos de tiempo. Luego CDMA (del inglés: Code Division Multiple Access), que divide la información en distintas secuencias pseudo-aleatorias de códigos para cada usuario. Por lo tanto, este método no está en función del tiempo ni de la frecuencia. Puede manejar mayor cantidad de usuarios y llamadas por radio base respecto a TDMA.

De las tecnologías más utilizadas a nivel mundial esta GSM (del inglés: Global System for Mobile Communications) de 2G, basada en TDMA. Las características principales de GSM es la utilización del SIM (del inglés: Subscriber Identity Module), calidad de voz digital y el envío de mensajes de texto.

Una vez que la 2G se estableció, las limitantes de algunos sistemas en lo referente al envío de información se hicieron evidentes. El uso de los canales de comunicación se hizo más eficiente y se hizo necesaria la tecnología 2.5G, por su rapidez y economía, ofreciendo características extendidas, como GPRS, tecnología basada en la transmisión por paquetes de datos, usada para el envío de mensajes cortos de datos (SMS, del inglés: Short Message Service), multimedia (MMS, del inglés: Multimedia Message Service), acceso a Internet (WAP, del inglés: Wireless Application Protocol) y otras aplicaciones. En este modelo 2.5G, se cobra por la cantidad de datos transferidos y no por tiempo de conexión.

Este desarrollo da paso a la Tercera Generación (3G), caracterizada por la capacidad de transferir de forma simultánea voz y datos (mensajes instantáneos, email, descarga de archivos) con velocidades notablemente superiores a las disponibles en la 2G (GSM / GPRS). Brinda la posibilidad de realizar videoconferencias en tiempo real, siendo este su mayor atractivo comercial. Hace uso del estándar, UMTS (del inglés: Universal Mobile Telecommunications System).

Por último aparece la Generación 4 (4G), que integra una serie de tecnologías para manejar velocidades desde 100 Mbps (en redes de telefonía celular) hasta 1 Gbps en redes inalámbricas WLAN (del inglés: Wireless Local Area Network) haciendo uso de Wi-MAX (del inglés: World Wide Interoperability for Microwave Access). Aunque aún no existe ningún operador que haya lanzado este tipo de redes a nivel comercial. (Jiménez, 2004; Martín Inzaurrealde, 2006; Syscom, 2005; Radioptica, 2007).

La tecnología se desarrolla a medida que transcurren los años, los móviles se han convertido en dispositivos esenciales en la vida cotidiana, por su uso sencillo y las posibilidades que brinda a partir de su portabilidad. Ese proceso evolutivo ha contribuido al surgimiento de cuatro generaciones, esta investigación se enfoca en la 2.5, donde el uso de los canales de comunicación se hace de forma eficiente y ofrece características extendidas, como GPRS, que usa en la primera versión de SIRIO Mobile para el envío de mensajes desde el dispositivo móvil y se sigue ajustando a las necesidades de la presente investigación.

1.3 COMUNICACIONES INALÁMBRICAS

Hasta este momento para navegar en Internet, usar el correo o acceder a cualquier otro tipo de información se necesita estar conectados a una red física, con la presencia de una toma de red y un cable para conectar un ordenador.

La necesidad de estar siempre conectados a Internet y sobre todo la movilidad, está haciendo que las redes inalámbricas vayan apareciendo en todas las esferas. Estas redes, WLAN⁴ (del inglés: Wireless Local Area Network), permiten que una persona pueda moverse, sin necesidad de una conexión física.

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

- Disponibilidad permanente a la información, lo cuál 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: los dispositivos necesitan estar a corta distancia dependiendo de la tecnología que se emplee, por ejemplo IrDA⁵ (del inglés: Infrared Data Association) y Bluetooth.
- Wi-Fi⁶: redes que pueden llegar a alcanzar longitudes de 350 metros en zonas abiertas y que están destinadas a la creación de redes de trabajo.
- Wi-Max: redes de más ancho de banda y mayor alcance.

(Denys Buedo, Yunisel Viera, 2007)

⁴ Sistema de comunicación de datos inalámbrico flexible, muy utilizado como alternativa a las redes LAN cableadas o como extensión de éstas

⁵ Define un estándar físico en la forma de transmisión y recepción de datos por rayos infrarrojo.

⁶ Conjunto de estándares para redes inalámbricas basados en las especificaciones IEEE 802.11. Creado para ser utilizado en redes locales inalámbricas.

Para comunicarse a larga distancia existen diferentes tecnologías como son GSM, GPRS y UMTS.

1.3.1 GSM.

GSM, Sistema Global para las Comunicaciones Móviles, o también conocida como Grupo Especial Móvil, es un estándar mundial para teléfonos móviles digitales. Es abierto, no propietario y evolutivo (aún en desarrollo). Su primera funcionalidad es la transmisión de voz, aunque también permite la transmisión de datos a baja velocidad: 9,6 kbit/s.

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, es decir, una vez establecida la conexión física entre dos usuarios, dedica los recursos propios hasta que no es solicitado expresamente el establecimiento de la conexión, independientemente del hecho de que los dos usuarios se intercambien datos durante todo el tiempo de conexión.

Esta modalidad de transferencia es óptima sólo en el caso en que los dos usuarios tengan que intercambiar una cantidad significativa de datos (transferencia de ficheros o archivos); resulta ineficiente en cuanto los datos a intercambiarse son de pequeña entidad o bien, en el caso más frecuente, el tráfico de datos es de tipo interactivo o transitorio, es decir, el tiempo de uso efectivo de los recursos de la red supone sólo una parte con respecto al tiempo total de conexión (como, por ejemplo, la navegación en Internet a través de la World Wide Web). Para establecer conexión demora de 15 a 30 segundos, y en este caso se paga a partir de ese momento hasta que el usuario se desconecte.

A pesar de esto GSM ofrece una compatibilidad superior con sistemas, equipos y aplicaciones desarrolladas alrededor del mundo. (Navarro, 2005)

1.3.2 GPRS.

GPRS, perteneciente la G 2.5 es una evolución de GSM, no conlleva grandes inversiones y reutiliza parte de sus infraestructuras. Maneja el concepto de comunicación por paquetes, en vez de la tradicional por circuitos. Su velocidad de transferencia es de hasta 144 Kbps, la conexión es permanente y el tiempo de establecimiento de la misma inferior al segundo.

Los nuevos terminales GPRS se pueden conectar a ordenadores portátiles o de sobremesa como cualquier módem, pero, evidentemente, con las ventajas de ser inalámbrico.

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

- 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 para usos empresariales a través del servicio Protocolo para Aplicaciones Inalámbricas.

Cuando se accede a servicios directamente desde el dispositivo, la velocidad de transferencia, a diferencia de lo que puede parecer, no es el factor determinante. Tener una conexión permanente a Internet (lo que se traduce en acceder a servicios de manera casi instantánea) o el hecho de no estar pagando por tiempo de conexión son factores más relevantes que la velocidad de transmisión. (Enlys Molina, 2002; García, 2007).

1.3.3 UMTS.

La tecnología UMTS es un sistema de 3G, que juega un papel principal en las telecomunicaciones multimedia inalámbricas de alta calidad, con un pronóstico de alcanzar hasta 2000 millones de usuarios en todo el mundo en el año 2010.

Es una tecnología apropiada para una gran variedad de usuarios y tipos de servicios, y no solamente para usuarios muy avanzados, UMTS ofrece:

- Facilidad de uso y bajos costos: servicios de uso fácil y adaptable para abordar las necesidades y preferencias de los usuarios, amplia gama de terminales para realizar fácil acceso a los distintos servicios con bajo costo para asegurar un mercado masivo.
- Nuevos y mejorados servicios: Los servicios vocales mantendrán una posición dominante durante varios años. Los usuarios exigirán a UMTS servicios de voz de alta calidad junto con servicios de datos e información. Las proyecciones muestran una base de abonados de servicios multimedia en fuerte

crecimiento hacia el año 2010, con alta calidad en áreas carentes de estas posibilidades en la red fija, como zonas de difícil acceso.

- Acceso rápido: La principal ventaja de UMTS sobre la 2G, es la capacidad de soportar altas velocidades de transmisión de datos de hasta 144 kbit/s sobre vehículos a gran velocidad, 384 kbit/s en espacios abiertos de extrarradios y 2 Mbit/s con baja movilidad (interior de edificios). Esta capacidad sumada al soporte inherente del Protocolo de Internet (IP), se combinan poderosamente para prestar servicios multimedia interactivos y nuevas aplicaciones de banda ancha, tales como servicios de video telefonía y video conferencia.
- Transmisión de paquetes de datos y velocidad de transferencia de datos a pedido: UMTS ofrece la transmisión de datos en paquetes y por circuitos de conmutación de alta velocidad debido a la conectividad virtual a la red en todo momento y a las formas de facturación alternativas (por ejemplo, pago por byte, por sesión, tarifa plana, ancho de banda asimétrico de enlace ascendente / descendente) según lo requieran los variados servicios de transmisión de datos que están haciendo su aparición.

(Buedo, Viera, 2007; García, 2006)

Se trata de un avance a lo inevitable, ya que esta tecnología será la revelación del universo móvil. Una tecnología capaz de enviar datos a gran velocidad puede ser de gran utilidad en un mundo donde el tiempo real se valora cada día más. La empresa, el ocio y muchos otros sectores se verán beneficiados gracias a UMTS.

La Plataforma SIRIO Mobile 1.0 emplea la vía de comunicación GPRS, ya que es una de las más difundidas a nivel mundial, con velocidad de transmisión mayor a la de GSM. La baja velocidad de transferencia de GSM limita la cantidad de servicios que Internet ofrece. Por ejemplo, a 9,6 Kbps no se puede navegar por Internet de una manera satisfactoria. Si además, se tiene en cuenta que se está pagando por tiempo de conexión, los costos se disparan. Estos factores negativos hacen que GSM sea una tecnología mayoritariamente utilizada para la voz y no para los datos. La conexión permanente y el pago por información transmitida, son factores decisivos para utilizar la tecnología, ya que para los juegos, tipo de aplicaciones que soporta la plataforma SIRIO, se precisa estar conectado permanentemente; además el operador se beneficia pues para el jugador es necesario el intercambio de información, siendo esta la forma de pago de GPRS. La tecnología UMTS es muy joven y sólo

soportada por dispositivos de 3G, tecnología que no abunda actualmente. En la versión 2.0 se continúa trabajando sobre GPRS beneficiándose de todas sus características.

1.4 COMUNICACIÓN POR SOCKET

Los Sockets son semejantes a conexiones telefónicas entre dos ordenadores remotos, la comunicación es continua y finaliza cuando uno de los dos ordenadores cierra su conexión, los mismos permiten implementar una arquitectura Cliente - Servidor. La clase `ServerSocket` es una clase incluida en `java.net` que sirve para atender peticiones de conexiones, lo cual es útil para de crear un servidor, donde existe un bucle infinito que llama al método `accept` de la clase `ServerSocket` quedando a la espera de aceptar una conexión. Cuando esta conexión se produce esta clase devuelve un objeto de tipo socket que es el que utiliza el servidor para comunicarse con el cliente, de este socket se consigue una corriente de entrada y otra de salida.

A continuación, en la Figura 2 se muestra un diagrama de lo que ocurre en el lado del cliente y del servidor:

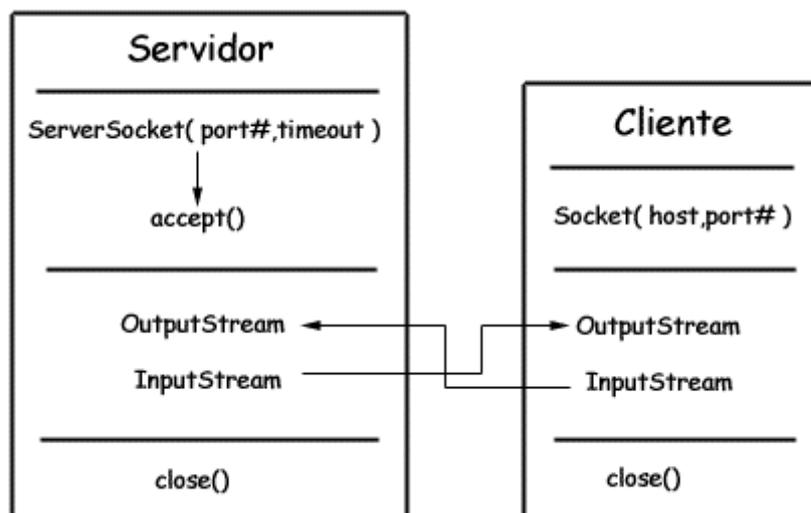


Figura 2: Modelo Comunicación Socket

El modelo de sockets más simple es:

- El servidor establece un puerto y espera durante un cierto tiempo (segundos), a que el cliente establezca la conexión. Cuando el cliente solicite una conexión, el servidor abrirá la conexión socket con el método `accept`.

- El cliente establece una conexión con la máquina host a través del puerto que se designe en puerto#
- El cliente y el servidor se comunican con manejadores InputStream y OutputStream.

Las propiedades de un socket dependen de las características del protocolo en el que se implementan. El protocolo más utilizado es TCP (del inglés: Transmission Control Protocol), aunque también es posible utilizar UDP (del inglés: User Datagram Protocol) o IPX (del inglés: Internetwork Packet Exchange), este último casi en desuso desde la aparición de TCP/IP (del inglés Transmission Control Protocol/Internet Protocol). Gracias al protocolo TCP, los sockets tienen las siguientes propiedades:

- Orientado a conexión.
- Se garantiza la transmisión de todos los octetos sin errores ni omisiones.
- Se garantiza que todo octeto llegará a su destino en el mismo orden en que se ha transmitido.

Estas propiedades son muy importantes para garantizar la corrección de los programas que tratan la información.

El protocolo UDP es un protocolo no orientado a la conexión. Sólo se garantiza que si un mensaje llega, llegue bien. En ningún caso se garantiza que llegue o que lleguen todos los mensajes en el mismo orden que se mandaron. Esto lo hace adecuado para el envío de mensajes frecuentes pero no demasiado importantes, como por ejemplo, mensajes para las actualizaciones de gráficos. (Sepulveda,2005; V, 2005)

1.5 PROTOCOLO DE COMUNICACIÓN

El protocolo de transferencia de hipertexto HTTP (del inglés: HyperText Transfer Protocol) es un protocolo del nivel de aplicación usado para la transferencia de información entre sistemas, de forma clara y rápida.

El protocolo HTTP se basa en un paradigma de peticiones y respuestas. Un cliente envía una petición en forma de método, y el servidor contesta con una línea de estado que incluye la versión del protocolo y un código que indica éxito o error.

Generalmente es el cliente el que inicia la comunicación HTTP y consiste en la petición de un recurso del servidor. Puede hacerse de forma directa al servidor o a través de intermediarios. (Guardado, 2005)

1.5.1 HTTPS

El protocolo HTTPS (del inglés Hypertext Transfer Protocol Secure) es una versión segura del protocolo HTTP que implementa un canal de comunicación cifrado y basado en SSL (Secure Socket Layers) entre el navegador del cliente y el servidor HTTP, para el tráfico de información sensible, como contraseñas, pin de cuentas de bancos. A diferencia de HTTP, HTTPS antes de enviar los datos realiza algunas acciones previas.

Para hacer esta negociación, el cliente, envía al servidor las opciones de cifrado, compresión y versión de SSL junto con algunos bytes aleatorios llamados Challenge de Cliente. El servidor, elige las opciones de cifrado, compresión y versión de SSL entre las que ha ofertado el cliente y le envía su decisión y su certificado. Ambos negocian la clave secreta llamada master secret y usando esta clave, la Challenge de Cliente y las opciones pactadas se envían la información encriptada de tal manera que de ser interceptada no se puede descifrar.

El certificado como un documento que certifica el interlocutor, en este caso el servidor HTTPS, es quien dice ser, esto se hace para evitar que un atacante pueda suplantar la identidad del servidor y recibir la comunicación segura en su lugar. Existen tres tipos de certificados digitales:

- **Certificados autofirmados:** lo firma la aplicación que lo creó. Estos se deben copiar manualmente al almacén de certificados de raíz de confianza en el equipo cliente o el dispositivo móvil. Cuando un cliente se conecta a un servidor por SSL y el servidor presenta un certificado auto firmado, se pedirá al cliente que compruebe en su almacén de certificados de confianza. Si el certificado del servidor está dentro de los que el cliente tiene registrados, este debe confiar explícitamente en la entidad emisora.
- **Certificados de infraestructura de clave pública de Windows:** este tipo es generado por la infraestructura de llave pública (PKI) de Windows. Una PKI es un sistema de certificados digitales, entidades de certificación y entidades de registro que comprueban y autentican la validez de cada parte implicada en una transacción electrónica, mediante el uso de criptografía con claves públicas. Todo esto, sin embargo, conlleva un coste asociado al tener que implementar servidores e infraestructuras adicionales para crear y administrar los certificados generados por la PKI de Windows.
- **Certificados de confianza de terceros:** son certificados generados por una entidad certificadora externa o comercial, adquiridos para usarlos en sus servidores de red. La mayoría de los certificados

de entidad certificadora comerciales son de confianza porque el certificado ya reside en el almacén de certificados raíz de confianza. Dado que el emisor es de confianza, el certificado también lo es. El uso de certificados de terceros simplifica notablemente la implementación.

(Iturralde, 2001; Bonet; Díaz)

1.6 SIRIO MOBILE 1.0.

Los grandes descubrimientos de la ciencia han dotado a los celulares de numerosas herramientas convirtiéndolos en objetos imprescindibles. La incorporación de la tecnología GPRS a los celulares ha impulsado el desarrollo de juegos en línea, los más disfrutados a nivel mundial.

Producto del alto crecimiento en la demanda de los juegos en red, la entidad Procyon llevó a cabo una investigación que se trazó como meta: gestionar las peticiones desde diferentes teléfonos celulares de manera simultánea y permitir la comunicación entre usuarios desde su móvil vía GPRS para poder hacer uso de aplicaciones de tipo multiusuario.

Producto de ello se desarrolla la Plataforma SIRIO Mobile 1.0, en este epígrafe se hace una breve descripción de sus características para ubicar esta investigación en sus bases y a partir de ellas cumplir el objetivo de extenderla a su próxima versión.

Algunos de los requisitos funcionales trazados en la primera versión de la Plataforma fueron:

1. **Gestionar Usuario:** Define todas las funcionalidades relacionados con el control de las cuentas de usuarios.
2. **Enviar Mensaje de texto:** Define el intercambio de mensajes entre los clientes conectados al servidor.
3. **Enviar Mensaje del servidor:** Define un mensaje enviado por el administrador del sistema a uno o varios usuarios.
4. **Enviar paquete de juego:** Define el envío de información relacionada con algún cambio en una partida.
5. **Gestionar Partida:** Define todas las funcionalidades relacionadas con la gestión de las partidas.

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

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 del producto, donde se tuvieron en cuenta la usabilidad, el rendimiento, soporte, portabilidad, seguridad, así como los requisitos de software y hardware.

La aplicación servidora hizo uso de J2EE (del inglés: Java 2 Enterprise Editions) y para el despliegue y alojamiento de la plataforma se usó un servidor Web Apache Tomcat.

La plataforma desarrollada tiene como principal aporte la gestión de múltiples juegos, sin ser significativo su tipo, el control del intercambio de información y recurso entre los usuarios conectados al servidor, distribuyendo éstos en dependencia del canal en el que se encuentren. Además 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 API⁷ (del inglés: Application Programming Interface) de conexión con el servidor orientado a los desarrolladores de juegos, el cuál recoge las acciones relacionadas con el funcionamiento de las aplicaciones clientes.

1.7 PLATAFORMAS DE DESARROLLO DE JUEGOS PARA MÓVILES.

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 Comunicaciones Inalámbricas y de la Telefonía Celular 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).

⁷ Conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Representa un interfaz de comunicación entre componentes software

Para dar soporte a estos 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, las cuales 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 estos sin ser significativo su tipo. Almacenan grandes volúmenes de información y numerosas conexiones simultáneas de usuarios.

(Buedo, Viera, 2007)

En Cuba 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 donde se realiza un amplio trabajo en la producción de juegos para teléfonos celulares. Obteniendo 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 para gestionar los juegos en red a largas distancias, llamada SIRIO Mobile 1.0. Con esta investigación se persigue realizar una extensión de dicha Plataforma, incorporándole la presencia de jugadores virtuales y el mantenimiento del estado de las aplicaciones que maneja. Se hace necesario evaluar las tendencias mundiales de este tipo de plataforma de desarrollo de juegos para móviles con el objetivo de identificar las mejores prácticas y hacer uso de ellas en esta investigación.

1.7.1 Productos similares.

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. Los desarrolladores de juegos podrán crear fácilmente juegos Java con un componente para juegos móviles online multiusuario, que permitirán a los usuarios jugar online con otros jugadores de todo el mundo. Los operadores de redes ya pueden

desplegar rápidamente y de forma rentable, comunidades de juego móvil online multiusuario para sus clientes de juegos Java, permitiéndoles generar nuevos ingresos por cada partida online. SNAP Mobile se basa en la tecnología middleware⁸ Scalable Network Application Package (SNAP) de Nokia y en el ampliamente generalizado lenguaje Java para terminales móviles. El middleware SNAP Mobile de Nokia estará disponible en el entorno operativo Solaris de Sun para facilitar la alta disponibilidad y capacidad requerida por las sesiones multiusuario. Permite además un acercamiento entre plataformas para juegos móviles multiusuario. 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. (Nokia, 2004)

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 (Exmachina, 2005).

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. (ExitGames, 2006)

Caso de estudio # 4: Gex

⁸ El Middleware es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas.

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 multiusuario en línea a través de múltiples tipos de dispositivos móviles. La plataforma Gex posibilita a los usuarios registrarse, obtener nombres de usuarios, subir sus puntuaciones, participar en torneos y recibir estadísticas de juegos.

Permite 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. (GameFederation, 2005)

Caso de estudio # 5: JIVE™

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 su transmisión, 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. (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, servicios de mensaje multimedia, MMS, 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)

1.8 ARQUITECTURA CLIENTE – SERVIDOR.

El cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al Servidor. Normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de una red.

El Servidor es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos.

Las características básicas de una arquitectura Cliente - Servidor son:

- Combinación de un cliente que interactúa con el usuario, y un servidor que interactúa con los recursos compartidos. El proceso del cliente proporciona la interfaz entre el usuario y el resto del sistema. El proceso del servidor actúa como un motor de software que maneja recursos compartidos tales como bases de datos, impresoras, módems, etc.
- Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad y capacidades del disco, y dispositivos de entrada y salida. Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.
- La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a recursos compartidos.

- Los clientes corresponden a procesos activos en cuanto a que son estos los que hacen peticiones de servicios a los servidores. Estos últimos tienen un carácter pasivo ya que esperan las peticiones de los clientes.
- No existe otra relación entre clientes y servidores que no sea la que se establece a través del intercambio de mensajes entre ambos. El mensaje es el mecanismo para la petición y entrega de solicitudes de servicio.
- El ambiente es heterogéneo. La plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma. Precisamente una de las principales ventajas de esta arquitectura es la posibilidad de conectar clientes y servidores independientemente de sus plataformas.

Uno de los aspectos que más ha promovido su uso, es la existencia de plataformas de hardware cada vez más baratas. Esta constituye a su vez una de las más palpables ventajas de este esquema, la posibilidad de utilizar máquinas considerablemente más baratas que las requeridas por una solución centralizada, basada en sistemas grandes. Además, se pueden utilizar componentes, tanto de hardware como de software, de varios fabricantes, lo cuál contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones. (Catarina, 2006)

En la versión 1.0 se hace uso de la arquitectura Cliente - Servidor, siendo el Cliente el teléfono móvil y el Servidor la Plataforma SIRIO Mobile 1.0, en esta investigación se continúa trabajando sobre este paradigma, con la misma estructura. Los Clientes le envían peticiones serializadas a SIRIO mediante el API Cliente – Juego, estableciendo un lenguaje común, el recibe las mismas, las interpreta y le devuelve la respuesta.

En el caso de la segunda versión la Plataforma también es Cliente cuando interactúa con los servidores de aplicaciones, mediante el API SIRIO para Servidor Aplicación.

1.9 PLATAFORMA DE DESARROLLO Y LENGUAJE DE PROGRAMACIÓN.

En la versión 2.0 de SIRIO Mobile se continúa haciendo uso de la plataforma J2EE que ofrece un conjunto de especificaciones y técnicas que proporcionan soluciones completas, seguras, estables y escalables para el desarrollo, despliegue y gestión de aplicaciones en múltiples niveles de funcionalidad basadas en servidores. J2EE reduce el costo y la complejidad de desarrollo, lo cuál redundará en rapidez de desarrollo. Además su uso es política de la entidad Procyon.

La plataforma J2EE define un estándar para el desarrollo de aplicaciones de múltiples niveles (servidores Web, de aplicaciones y base de datos). Gracias a que su funcionamiento se basa en componentes modulares que incluyen un conjunto de servicios predefinidos, se simplifica la tarea de la producción de sistemas. J2EE extiende las ventajas de la plataforma Java 2 (como por ejemplo, seguridad, la portabilidad de programas y el acceso a las bases de datos).

El desarrollo de la aplicación se realizará sobre el lenguaje Java, el cuál se caracteriza por ser simple y con una funcionalidad potente sin características confusas, es orientado a objetos, por lo que incorpora el encapsulamiento, la herencia y el polimorfismo. Además es robusto, comprueba los punteros, los límites de arrays y maneja las excepciones. Es seguro, separa el espacio de nombres local de los recursos procedentes de la red. Las clases se buscan primero entre las locales (luego en exteriores) para evitar los caballos de Troya, hace una apertura de ficheros y arranque de otras aplicaciones en máquina remota, no en local. Una de sus características fundamentales es que es dinámico, si se ejecuta una aplicación Java sobre red y encuentra una pieza que no sabe manejar, es capaz de traer automáticamente del servidor cualquiera de las piezas que necesita para funcionar. Es multihilos por lo que tiene mejor rendimiento interactivo y facilidad de desarrollo. (Buedo, Viera, 2007)

1.10 SERVIDOR WEB.

Un servidor Web es un programa que implementa el protocolo HTTP. Este protocolo está diseñado para transferir lo que se llama hipertextos, páginas Web o páginas HTML (del inglés: HyperText Markup Language): 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 6.0.14, el cuál constituye el proyecto de software libre más famoso escrito en Java, puede integrarse fácilmente a muchos sistemas y servidores de forma eficiente. Fue usado en la primera versión de SIRIO Mobile por ser su uso, política de la entidad Procyon.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta

disponibilidad. Dado que fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java. (Apache Software Foundation, 2006) En la segunda versión se continúa usando.

1.11 PERSISTENCIA DE DATOS.

1.11.1 XML

La versión 2.0 de SIRIO Mobile dará la posibilidad al desarrollador de integrar su juego, para ello se necesita almacenar la información necesaria con el fin de hacer un reconocimiento de la aplicación, en este caso se va a usar un fichero de configuración XML (del inglés: Extensible Markup Language).

XML es un metalenguaje extensible de etiquetas desarrollado, que permite definir la gramática de lenguajes específicos. No es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Propone un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo, entre otras.

Es una tecnología sencilla que tiene a su alrededor otras que la complementan. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Ventajas de XML:

- Es extensible, lo que quiere decir que una vez diseñado un lenguaje y puesto en producción, igual es posible extenderlo con la adición de nuevas etiquetas de manera de que los antiguos consumidores de la vieja versión todavía puedan entender el nuevo formato.
- El analizador es un componente estándar, no es necesario crear un analizador específico para cada lenguaje. Esto posibilita el empleo de uno de los tantos disponibles. De esta manera se evitan errores y se acelera el desarrollo de la aplicación.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarlo. Mejora la compatibilidad entre aplicaciones.
- La tecnología XML busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible. Que la información sea estructurada quiere decir que se

compone de partes bien definidas, y que esas partes se componen a su vez de otras partes. Entonces se tiene un árbol de pedazos de información. (W3, 2006)

- En la presente investigación se desarrolla la versión 2.0 de la Plataforma SIRIO Mobile, en su primera variante se almacenaban los datos persistentes en un fichero XML, en función de mejorar la eficiencia del sistema, se persigue aliviar la memoria del Servidor, guardando los datos necesarios para el funcionamiento de la plataforma en una base de datos, teniendo en cuenta que es mayor, esta vez, el número de elementos que deben persistir, por incorporar SIRIO Mobile, las funcionalidades de guardar el estado de las aplicaciones que maneja e iniciar partidas con jugadores virtuales.

1.11.2 MySQL Server.

Una base de datos es una colección estructurada de datos. Para añadir, acceder, y procesar los datos almacenados se necesita un sistema de gestión de base de datos como por ejemplo MySQL Server. Al ser las computadoras excelentes en tratar grandes cantidades de datos, los sistemas de gestión de bases de datos juegan un papel fundamental en aplicaciones autónomas o como parte de otras aplicaciones. MySQL es un sistema de gestión de base de datos relacional, pues almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén, lo que añade velocidad y flexibilidad, la parte SQL de "MySQL" se refiere a lenguaje estructurado de consultas, (del inglés: Structured Query Language). SQL es el lenguaje estandarizado más común para acceder a bases de datos y está definido por el estándar ANSI/ISO SQL. Es un sistema de código abierto, lo que significa que es posible usar y modificar el software en toda la comunidad de desarrollo, se puede bajar el mismo desde internet y usarlo gratis, en esta investigación se hace uso de la versión 5.1.22, la cuál posee licencia GPL (del inglés: General Public License). MySQL tiene una característica significativa, es muy rápido, fiable y fácil de usar, ha sido aplicado con éxito en entornos de producción de alto rendimiento durante varios años. MySQL Server ofrece hoy en día una gran cantidad de funciones, su conectividad, velocidad, y seguridad lo hacen ser altamente apropiado para acceder a bases de datos. (Burbano, 2006)

El equipo de desarrollo se encuentra familiarizado con este sistema gestor de base de datos y además su uso es política de la entidad Procyon.

1.12 METODOLOGÍA DE DESARROLLO EMPLEADA.

El Proceso Unificado es un proceso de software genérico que puede ser utilizado para varios sistemas de software, en diferentes áreas de aplicación, tipos de organizaciones, niveles de competencia y tamaños de proyectos.

RUP (del inglés: Rational Unified Process) es un proceso basado en la documentación, importante elemento para futuras actualizaciones o versiones de SIRIO Mobile, además aporta diferentes elementos de planificación, con los que se controla el desarrollo del software. A través de un predefinido esquema de escalabilidad y gestión de riesgos, se pueden reconocer previamente problemas y fallos de forma temprana. Define en cada momento del ciclo de vida del proyecto, qué artefactos, con qué nivel de detalle, y qué rol, se deben crear. La versión 1.0 de SIRIO Mobile, hizo uso de RUP, en esta investigación se continúa trabajando sobre él por la documentación que aporta para la investigación y las ventajas que ofrece.

Entre ellas, provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Asegura la producción de software de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible.

Además posee tres conceptos claves que lo hacen único: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental

Un caso de uso es una pieza en la funcionalidad del sistema que le da al usuario un resultado de valor. Juntos constituyen el modelo de casos de uso que describe la funcionalidad completa del sistema. Esta es una forma de añadir funcionalidades a SIRIO Mobile poco a poco, aportando un ejecutable en cada iteración que cumple un conjunto de requerimientos, en vez de añadirlas todas cada intervalos de tiempos mayores.

La arquitectura surge de las necesidades de la empresa. Es la vista del diseño completo con las características fundamentales, dejando los detalles de lado. El valor de la arquitectura depende del personal al cuál se le asigne la tarea, pues el criterio define los aspectos más importantes y este se enriquece con la experiencia, es una ventaja que el equipo de desarrollo de esta investigación se encuentre familiarizado con el proceso y centre su trabajo en la arquitectura mitigando los riesgos en todo el ciclo de vida.

Desarrollar un producto de software comercial es una tarea enorme que puede continuar por varios meses o años, siendo este el caso en cuestión; es práctico dividir el trabajo en pequeños pedazos o mini-proyectos, los cuales constituyan una iteración que finaliza en un incremento. Las iteraciones se refieren a pasos en el flujo de trabajo, y los incrementos a crecimiento del producto.

En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean el diseño usando la arquitectura como guía, implementan el diseño en componentes y verifican que satisfagan los casos de uso. Si una iteración cumple sus metas, el desarrollo continúa con la siguiente, sino se prueba un nuevo enfoque. Esta estrategia se ajusta a la forma de trabajo que se adopta en el desarrollo de la versión 2.0 de la Plataforma, ya que aporta aumentos del producto que pueden ser medidos en cada etapa e incrementados en las posteriores.

1.13 HERRAMIENTAS DE DESARROLLO EMPLEADAS

Mediante las herramientas de desarrollo y sus prestaciones, se obtienen productos robustos, que pueden ser modelados, desarrollados y controlados. Se hace un estudio sobre las herramientas CASE⁹ (del inglés: Computer Aided Software Engineering) y se hace referencia a una de las más poderosas herramientas de modelado visual, Rational Rose Enterprise Edition 2003. En este epígrafe se estudia el Eclipse como Entorno Integrado de Desarrollo y se define la herramienta de control de versiones a usar durante el proceso de la investigación, TortoiseSVN.

1.13.1 Herramientas de modelado de UML

A medida que los sistemas que hoy se construyen se tornan más complejos, las herramientas de modelado con UML¹⁰ (del inglés: Unified Modeling Language) ofrecen muchos beneficios para todos los involucrados en un proyecto. Las herramientas CASE de modelado con UML permiten aplicar la metodología de análisis y diseño orientados a objetos y abstraernos del código fuente, en un nivel donde la arquitectura y el diseño se tornan más obvios y más fáciles de entender y modificar. Cuanto más grande es un proyecto, es más importante utilizar una herramienta CASE.

Al usar las herramientas CASE (Apexnet, 2005):

⁹ Las Herramientas CASE son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

¹⁰ UML es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

- Los Analistas de Negocio/Sistema pueden capturar los requisitos del Negocio/Sistema con un modelo de casos de uso.
- Los Diseñadores/Arquitectos pueden producir el modelo de diseño para articular la interacción entre los objetos o los subsistemas de la misma o de diferentes capas (los diagramas UML típicos que se crean son los de clases y los de interacción).
- Los Desarrolladores pueden transformar rápidamente los modelos en una aplicación funcionando, y buscar un subconjunto de clases, métodos y asimilar el entendimiento de cómo lograr interfaces con ellos.

El modelo actúa como el plano y guiará finalmente la construcción del sistema. De manera semejante, la administración es capaz de ver, puntualmente y desde un alto nivel, una representación del diseño y comprender lo que está sucediendo.

Por estas razones, las herramientas CASE de UML acompañadas con metodologías, brindan una forma de representar sistemas demasiados complejos para comprenderlos a través de su código fuente subyacente y permiten desarrollar la solución de software correcta más rápido y más económicamente.

Para el modelado de los artefactos y diagramas generados a lo largo del ciclo de vida del proyecto se decidió utilizar Rational Rose Enterprise Edition 2003, pues su uso está muy estandarizado a nivel mundial y es conocido por su madurez y acabado. Además, el grupo de desarrollo se encuentra familiarizado con el uso de esta herramienta.

Es una de las más poderosas herramientas de modelado visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo. Cubre todo el ciclo de vida de un proyecto:

- Concepción y formalización del modelo.
- Construcción de los componentes.
- Transición a los usuarios.
- Certificación de las distintas fases.

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.

1.13.2 Entorno Integrado de Desarrollo.

Un IDE (del inglés: Integrated Development Environment) es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse a un solo lenguaje de programación, inclusive a varios proveyendo un marco de trabajo amigable para la mayoría. Es empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI (del inglés: Graphical User Interface).

En la versión 2.0 de SIRIO Mobile se continúa haciendo uso de Entorno Integrado de Desarrollo, Eclipse 3.3.2 con el cuál el equipo de desarrollo ya tiene experiencias y conoce de sus facilidades.

Eclipse es una plataforma abierta para el desarrollo de aplicaciones que corren sobre un amplio rango de sistemas operativos. 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 cuál 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.

El plugin Tomcat se diseñó para el desarrollo de aplicaciones Web y el despliegue hacia un servidor específico.

1.13.3 Herramienta para el control de versiones.

Una versión, revisión o edición de un producto, es el estado en el se encuentra en un momento dado en su desarrollo o modificación. Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas.

El control de versiones se realiza principalmente en la industria informática para controlar las distintas versiones del código fuente. Sin embargo, los mismos conceptos son aplicables a otros ámbitos como documentos, imágenes, sitios web, entre otros.

Un sistema de control de versiones proporciona un mecanismo de almacenaje de los elementos que gestiona. Da la posibilidad de realizar cambios sobre estos. Además hace un registro histórico de las acciones realizadas con cada elemento o conjunto de elementos, normalmente pudiendo volver o extraer un estado anterior del producto.

Aunque un sistema de control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión, como es el caso del sistema de control de versiones Subversion de tipo centralizado, pues existe un repositorio que contiene el código. Se facilitan las tareas administrativas a cambio de reducir la potencia y flexibilidad, pues todas las decisiones fuertes necesitan la aprobación del responsable.

Es un software libre, otra característica importante de Subversion es que, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, 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.

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.

Existen varias interfaces a Subversion, ya sea programas individuales como interfaces que lo integran en entornos de desarrollo. En esta investigación se hace uso del TortoiseSVN, el cuál provee integración con el explorador de Windows y es la interfaz más popular en este sistema operativo.

1.14 CONCLUSIONES.

En este capítulo se hizo un análisis de la Telefonía Móvil, que ofrece nuevas prestaciones a los usuarios, los cuales demandan aplicaciones más amigables y entretenidas. A este adelanto lo acompaña la evolución de las Comunicaciones Inalámbricas, GPRS, se escoge como tecnología para llevar a cabo la extensión de SIRIO Mobile, por sus facilidades de velocidad de transmisión y pago por información transmitida. Se realiza un análisis de las plataformas de desarrollo para juegos similares en el mundo, sustrayendo experiencias con el fin de incorporarle mayores servicios a la versión 2.0 de la Plataforma. Queda definida una Arquitectura Cliente – Servidor, la persistencia de datos con XML y MySQL, plataforma de desarrollo J2EE y lenguaje de programación Java. Se deciden las herramientas necesarias, modelado de la aplicación con Rational Rose 2003, el desarrollo sobre Eclipse y TortoiseSVN para el control de versiones.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 INTRODUCCIÓN.

Procyon Soluciones provee soluciones y servicios en el mundo de las tecnologías de la información, generando un alto valor y satisfacción a sus clientes. La rapidez, calidad, sentido de innovación, y la responsabilidad con que asumen sus compromisos es una de las primicias de la organización. Mantiene constantemente su infraestructura tecnológica actualizada a fin de garantizar a sus clientes una capacidad inmediata de respuesta que satisfaga tanto distintos modelos de negocios, como las nuevas exigencias que puedan presentar operadores, clientes y proveedores de contenido.

Con la Plataforma SIRIO Mobile 1.0, Procyon permite la comunicación simultánea entre usuarios conectados a un juego desde su teléfono celular vía GPRS. La segunda versión incorpora nuevas funcionalidades, surgidas a partir de deficiencias existentes, como son: iniciar partidas con jugadores virtuales y mantenimiento del estado de sus aplicaciones. Características importantes para poder competir con las soluciones similares desarrolladas a nivel mundial.

2.2 COMPARACIÓN ENTRE LAS PLATAFORMAS DE DESARROLLO DE JUEGOS PARA MÓVILES Y SIRIO MOBILE.

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

Nokia desarrolló el SNAP Mobile's, el cuál tiene 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, al igual que SIRIO Mobile 1.0. Dicha API de Nokia 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 estudiadas en la presente investigación: 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. SIRIO Mobile 1.0 cuenta con un módulo de Administración Web para todo lo referido a crear cuentas de usuarios, obtener estadísticas de jugadores y juegos on line desarrollado por otro equipo de trabajo.

En general las plataformas de desarrollo de juegos para móviles son capaces de manejar varias conexiones de manera simultánea, siendo en su mejor caso escalable a cualquier capacidad necesaria. Tienen la posibilidad de alojar gran cantidad de juegos, incluso en ocasiones de distintos tipos. Semejante a SIRIO Mobile 1.0, que gestiona múltiples juegos, sin ser significativo su tipo, y 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. La Plataforma Gex, posibilita a los usuarios registrarse, obtener nombres, subir puntuaciones, participar en torneos y recibir estadísticas de juegos; otra característica similar a la desarrollada en Procyon, la cuál incorpora una sala de amigos donde se establece quienes jugarán una partida, también brinda un sistema de registro de usuarios, almacenados de forma segura, así como las estadísticas de las partidas desarrolladas.

SIRIO Mobile 2.0 amplía sus funcionalidades añadiendo la posibilidad de iniciar partidas con jugadores virtuales, y permitiendo guardar el estado de las aplicaciones que maneja. Características de primer orden en las plataformas analizadas anteriormente, y que no incorporaba en la primera versión, es por ello que se hacen necesarias y se evalúan como deficiencias comparándola con los productos similares desarrollados a nivel mundial. Además se incorpora la internacionalización de la Plataforma, permitiendo a los jugadores recibir los mensajes del servidor en el lenguaje que estime.

2.3 PROPUESTA DEL SISTEMA.

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

1. Cliente API Juego: Para el desarrollo de los juegos se brinda este API de conexión que se comunica con el Servidor mediante un lenguaje común, vía GPRS. En dicha API se encuentran agrupadas y definidas las posibles acciones que pueden ser enviadas y recibidas hacia o desde el Servidor. De esta forma los desarrolladores de juegos multiusuario disminuyen el tiempo de desarrollo de sus aplicaciones.

2. Servidor: Es la aplicación encargada de gestionar las peticiones de las aplicaciones y de generar las respuestas a cada acción enviada. Es capaz de manejar la concurrencia, seguridad y persistencia de los datos en el servidor. Cuenta con servicios para controlar las pérdidas de conexión de algún usuario con el servidor y la detección de cambios en el fichero de configuración.

3. API SIRIO para Servidor Aplicación: Este API es el encargado de manejar la comunicación entre la Plataforma SIRIO Mobile 2.0 y el Servidor de Aplicaciones mediante un lenguaje común vía Socket,

de esta forma se garantiza la homogeneidad de la transmisión de los datos, siendo esta clara, eficiente y fácil de comprender. Permite gestionar la lógica de los juegos y salvar el estado de las aplicaciones.

La Figura 3 presenta la propuesta del Sistema a desarrollar en esta investigación, representando cada una de las partes descritas anteriormente:

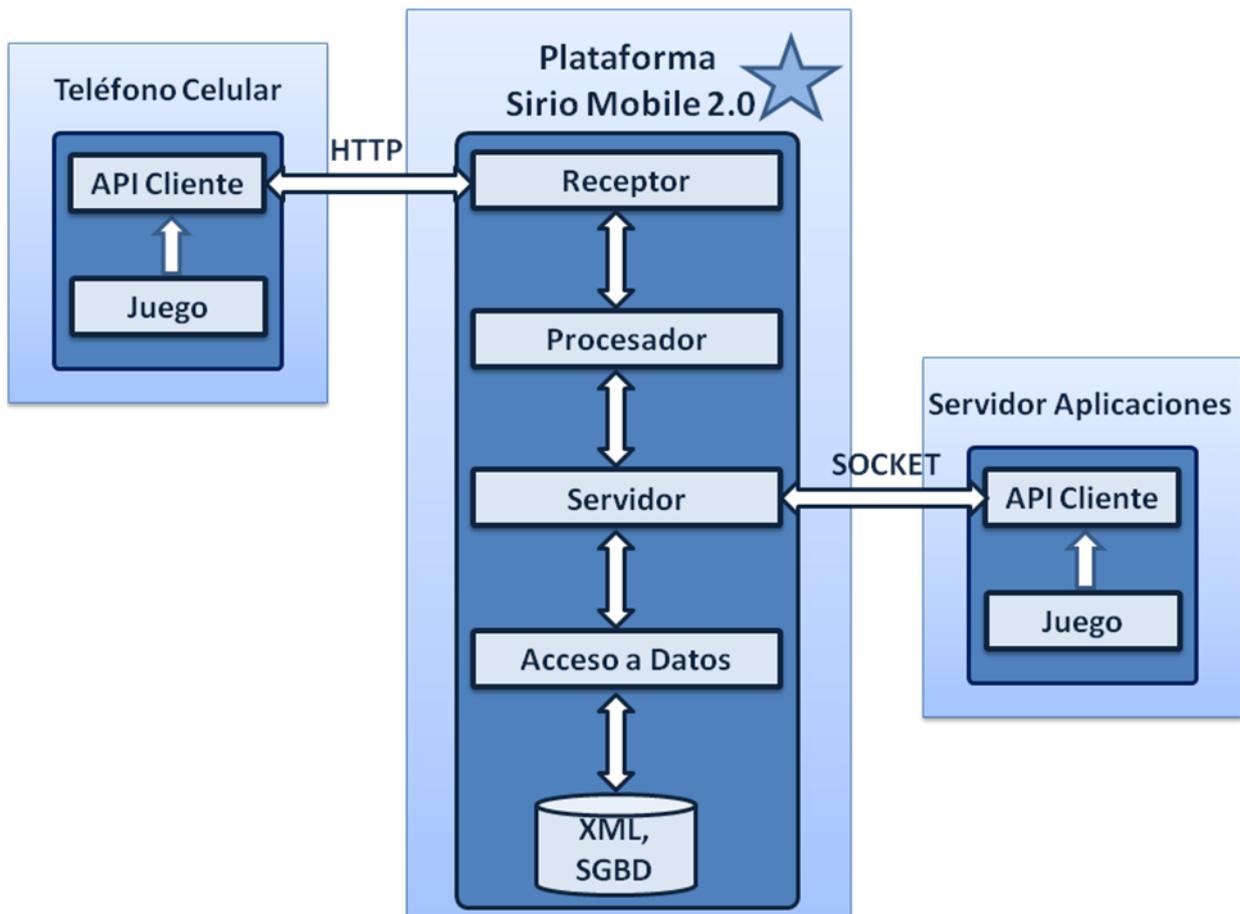


Figura 3: Propuesta del Sistema SIRIO Mobile 2.0.

2.4 MODELO DEL DOMINIO.

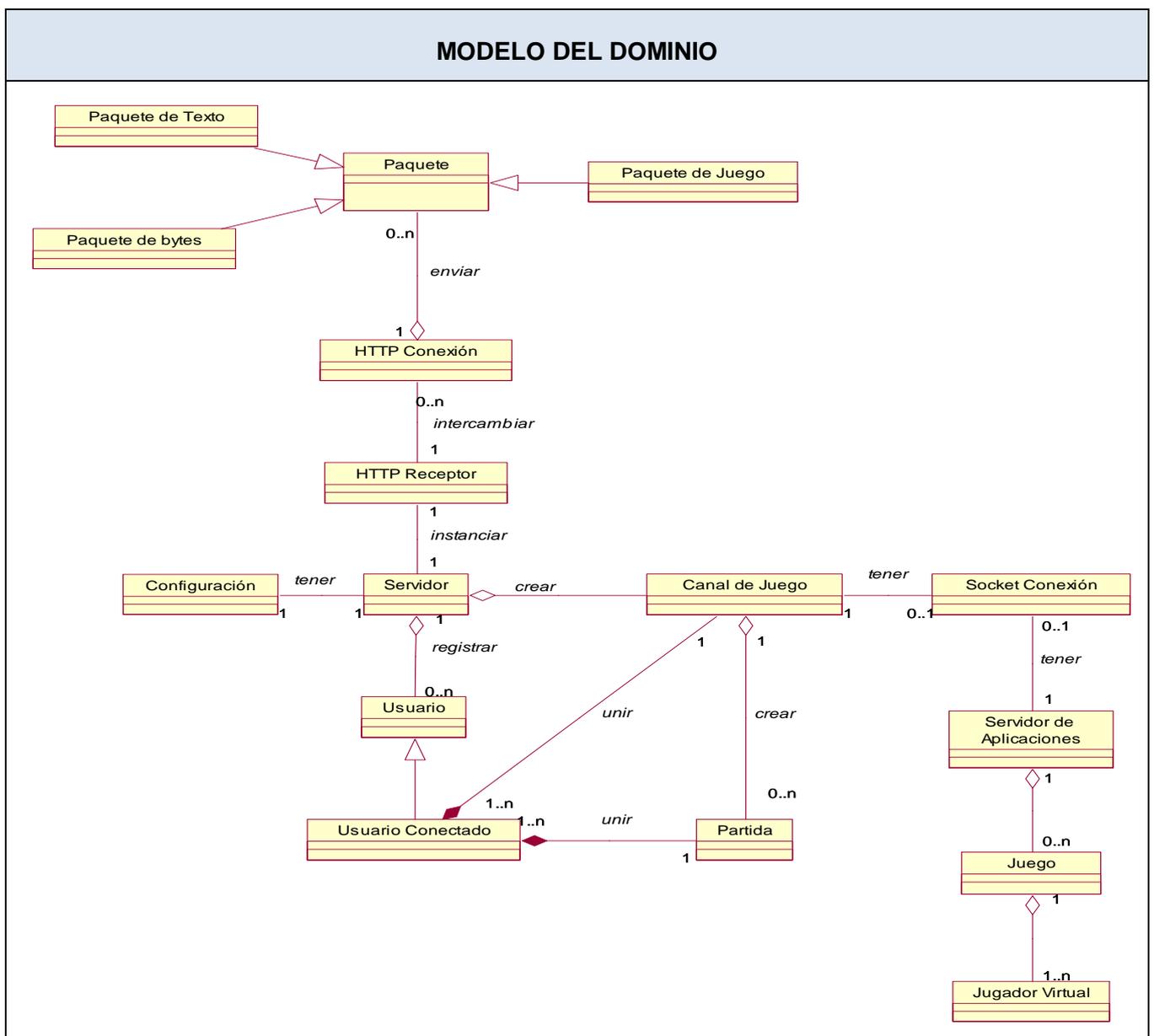
El modelo del dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema. El modelo del dominio se considera un subconjunto del llamado modelo de objetos del negocio.

Durante el desarrollo de SIRIO Mobile 1.0 se detectaron problemas en la visibilidad de los procesos. 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 un modelado del dominio. (Denys Buedo, Yunisel Viera, 2007)

La primera versión de la Plataforma describe un Modelo del Dominio donde se manejan conceptos que describen como se llevaba a cabo el proceso en aquellos momentos. SIRIO Mobile 2.0 continúa haciendo uso de algunos de esos conceptos e incorpora elementos necesarios para llevar a cabo el cumplimiento de las funcionalidades que se requieren, por lo que se hace un Modelo del Dominio.

Tabla 1: Modelo del Dominio SIRIO Mobile 2.0:



2.4.1 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).

HTTP Conexión: Define al responsable de manejar todo el proceso de envío y recepción de información de cualquier índole con el servidor.

HTTP Receptor: 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.

Configuración: Define la configuración que va a tener la plataforma SIRIO Mobile 2.0 en cuanto a juegos incorporados, idioma.

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

Partida: Define el entendimiento entre dos 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 de Juego: Define al espacio en el servidor donde se posibilita el intercambio de mensajes entre jugadores de un mismo juego.

Socket Conexión: Define el intercambio de información entre Canal de Juego y Servidor de Aplicaciones.

Servidor de Aplicaciones: Define al responsable de recibir mensajes desde el Servidor, procesar y almacenar la lógica de los juegos.

2.5 LEVANTAMIENTO DE REQUISITOS.

Un requerimiento es una condición o capacidad que necesita un usuario para resolver un problema o que tiene que ser alcanzada por un componente de un sistema o sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.

Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tuvieron acerca de lo que debe hacer el sistema, han sido analizadas como candidatas a requisitos. Los requisitos se pueden clasificar en: funcionales y no funcionales.

2.5.1 Requisitos funcionales.

RF1- Gestionar Partida: Define todas las funcionalidades relacionadas con la gestión de las partidas con jugadores virtuales o jugadores en línea.

RF1.1-Iniciar Partida: Define la posibilidad que tiene un cliente de iniciar una partida.

RF1.2- Salvar Partida: Define la posibilidad que tiene un cliente de salvar una partida.

RF1.3- Listar Partidas: Define la posibilidad que tiene un cliente de listar sus partidas salvadas.

RF1.4- Cargar Partida: Define la posibilidad que tiene un cliente de continuar una partida anteriormente salvada.

RF1.5- Abandonar Partida: Define la posibilidad que tiene un cliente de abandonar una partida.

RF2- Gestionar Paquete: Define todas las funcionalidades relacionadas con el intercambio de paquetes de juego de texto entre la Plataforma y el Servidor de Aplicaciones.

RF2.1- Enviar Paquete: Define todos los procedimientos para el envío de un paquete de juego.

RF2.2-Recepcionar Paquete: Define todos los procedimientos para la recepción de un paquete de juego.

RF3- Actualizar configuración del Servidor: Define la funcionalidad relacionada con la configuración de la Plataforma SIRIO Mobile 2.0, siendo necesario que el Servidor no detenga su funcionamiento.

RF4- Enviar contenido: Define la funcionalidad de envío de flujo de bytes entre los usuarios registrados a la Plataforma SIRIO Mobile.

RF5- Internacionalizar SIRIO Mobile: Define la funcionalidad de enviar mensajes desde la Plataforma al Cliente dependiendo del idioma con que el mismo se haya registrado.

2.5.2 Requisitos 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:

Requisitos no funcionales para la Plataforma SIRIO Mobile 2.0:

- **Usabilidad:** La aplicación estará orientada a desarrolladores de aplicaciones para móviles que deben tener un conocimiento básico en la materia. Las características de la misma 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 su uso en casi toda la totalidad de dispositivos móviles de generación 2.5 o más.
- **Seguridad:** Se necesita conocer la identidad del cliente cada vez que realice una petición a la Plataforma, además se le debe incorporar la opción de poder usar certificados digitales, permitiendo enviar los datos confidenciales vía HTTPS.
- **Software:** Debe utilizarse un contenedor Web que tenga soporte para servlets versión 2.4 y JSP versión 2.0.
- **Hardware:** Mínimo: Procesador de 1.6GHz Pentium: 256MB RAM. Recomendado: Procesador de 3.0GHz Pentium: 512MB RAM.

Requerimientos no funcionales para el juego cliente:

- **Software:** Debe ser desarrollado usando J2ME, los dispositivos para los que desarrollarán los juegos deben poseer configuración CLDC¹¹ (del inglés: Connected Limited Device Configuration) 1.0 y Profile MIDP¹² (del inglés: Mobile Information Device Profile) 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.

Requerimientos no funcionales para el servidor de juegos:

- **Software: Debe ser desarrollado usando JAVA.**
- **Hardware:** Las computadoras donde se decida establecer el servidor debe tener la maquina virtual de java instalada.

2.6 MODELO DE CASOS DE USO DEL SISTEMA.

En esta sección se identifican los actores y casos de uso del sistema, quedando determinado el Diagrama de Casos de Uso del Sistema. Se describen además los casos de uso del sistema.

2.6.1 Actores del sistema.

Los actores del sistema no son parte de él, pero sí intercambian información con él. Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado. En esta investigación se encuentran los siguientes actores:

Tabla 2. Definición de Actores del Sistema.

Actores	Descripción
Cliente	Define al cliente final de los posibles juegos a desarrollar que interactúan con el servidor.

¹¹ Es un framework específico para aplicaciones Java ME dirigidos a los dispositivos con recursos muy limitados como teléfonos móviles.

¹² Es una versión de J2ME, integrada en el hardware de celulares relativamente modernos que permite el uso en estos de aplicaciones Java, tales como juegos, aplicaciones u otros.

Reloj	Determina cada cierto tiempo la validación del fichero de configuración de la Plataforma SIRIO Mobile.
--------------	--

2.6.2 Casos de uso del sistema.

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.

En esta versión de SIRIO Mobile se manejan casos de uso ya descritos en la anterior versión, pero que se modifican en esta con el propósito de incorporar nuevas funcionalidades. Se incorporan algunos nuevos en dependencia de los requisitos funcionales analizados.

Los casos de uso determinados en la versión 2 de SIRIO Mobile son:

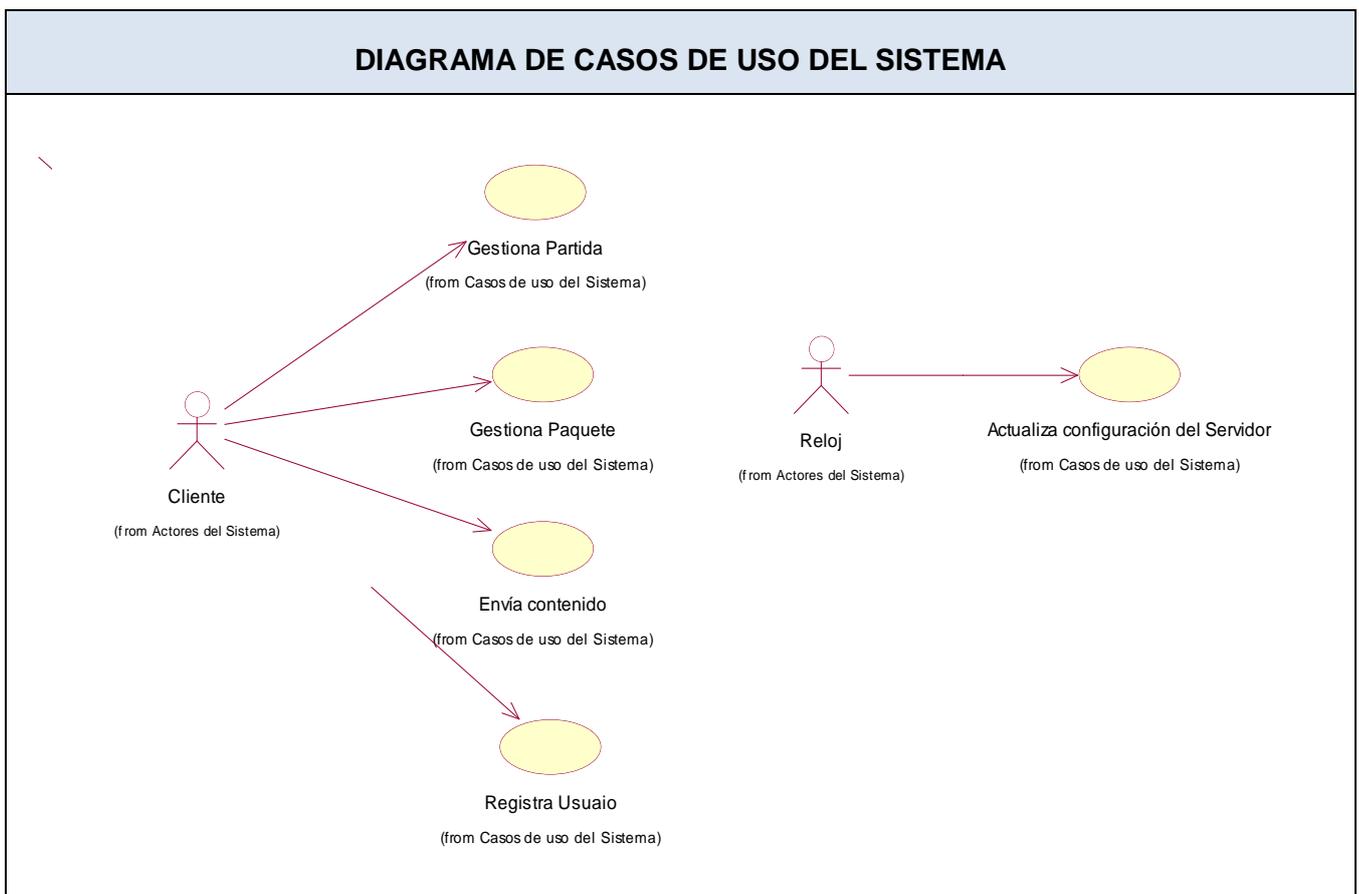
Tabla 3. Definición de los Casos de Uso del Sistema.

CU_1	Gestionar Partida
Actor	Cliente
Descripción	<p>Es el caso de uso que engloba todas las funcionalidades que tienen que ver con la gestión de una partida:</p> <ul style="list-style-type: none"> • Iniciar Partida. • Salvar Partida. • Busca Partida. • Cargar Partida. • Abandonar Partida.
Referencia	RF1,CU_2

CU_2	Gestionar Paquete de Juego
Actor	Cliente
Descripción	Es el caso de uso que permite la gestión de un Paquete de Juego: <ul style="list-style-type: none"> • Enviar Paquete de Juego. • Recibir Paquete de Juego.
Referencia	RF2
CU_3	Configurar la Plataforma SIRIO Mobile 2.0.
Actor	Reloj
Descripción	Es el caso de uso que permite la actualización de la configuración de SIRIO Mobile 2.0.
Referencia	RF3
CU_4	Enviar Contenido
Actor	Cliente
Descripción	Es el caso de uso que permite el envío de flujo de bytes entre clientes registrados en la Plataforma SIRIO Mobile.
Referencia	RF4
CU_5	Registrar Usuario
Actor	Cliente
Descripción	Es el caso de uso que permite el registro de

	un usuario en la Plataforma SIRIO Mobile, descrito en la versión 1.0, y que esta vez se le incorpora la opción de escoger el idioma.
Referencia	RF5

Tabla 4. Diagrama de Casos de Uso del Sistema



2.6.3 Descripción de los Casos de uso del sistema.

A continuación se muestra la descripción del Caso de Uso: Gestionar Partida y de su Sección: Salvar Partida. En el Anexo 1 se encuentren descritas las demás secciones de este caso de uso.

Tabla 5. Descripción textual del Casos de Uso Gestionar Partida. Sección: Salvar Partida.

Caso de Uso	Gestionar Partida	
Actor	Cliente	
Resumen	El Caso de Uso comienza cuando el Cliente inicia el Juego y selecciona la opción a realizar.	
Precondiciones	El usuario debe estar previamente autenticado en el Servidor.	
Referencias	RF1, RF1.1, RF-1.2, RF-1.3, RF-1.4, RF-1.5.	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Cliente inicia el juego.	2. El sistema le muestra las opciones que puede realizar: <ul style="list-style-type: none"> • Iniciar Partida. • Salvar Partida. • Listar Partidas. • Cargar Partida. • Abandonar Partida. 	

<p>3. Si el Cliente selecciona la opción Iniciar Partida, ir a la sección “Iniciar Partida”.</p> <p>4. Si el Cliente selecciona la opción Salvar Partida, ir a la sección “Salvar Partida”.</p> <p>5. Si el Cliente selecciona la opción Listar Partidas, ir a la sección “Listar Partidas”.</p> <p>6. Si el Cliente selecciona la opción Cargar Partida, ir a la sección “Cargar Partida”.</p> <p>7. Si el Cliente selecciona la opción Abandonar Partida, ir a la sección “Abandonar Partida”.</p>	
<p>Sección “Salvar Partida”.</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
<p>4.1 El cliente trasmite al sistema la petición de salvar el estado de la partida.</p>	<p>4.2 El sistema recibe la petición y busca los datos de la partida.</p>
	<p>4.3 Los datos de la partida y la solicitud son transmitidos al Servidor de Aplicaciones.</p>
	<p>4.4 Si no existe error en la transmisión se salva la partida.</p>
<p>Flujos Alternos</p>	
	<p>4.4 Si existe algún error en la transmisión se le informa al cliente.</p>
<p>Poscondiciones</p>	<p>La partida es salvada.</p>

2.7 CONCLUSIONES.

En este Capítulo se compara a SIRIO Mobile 2.0 con las plataformas de desarrollo de juegos para móviles, se analiza la necesidad de la extensión de SIRIO Mobile 1.0, y se realiza una propuesta del nuevo sistema a realizar. Se estructura un Modelo del Dominio definiéndose los conceptos tratados en el mismo, donde se mantienen algunos de la primera versión de la Plataforma, los cuales serán usados y modificados en esta investigación y se incorporan nuevos conceptos necesarios para llevar a cabo los objetivos planteados. Posteriormente se hace un levantamiento de requisitos necesario para la continuidad y claridad del proceso de desarrollo del software. Quedan definidos los casos de uso del sistema, sus actores, así como la descripción textual de cada uno de los casos de uso, siendo de vital importancia para próximas iteraciones.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 INTRODUCCIÓN.

Para crear una aplicación de software hay que describir el problema y las necesidades o requerimientos: en qué consiste el conflicto y qué debe hacerse. El Análisis se centra en una investigación del problema, no en la manera de definir una solución. Para desarrollar una aplicación, también es necesario contar con descripciones detalladas y de alto nivel de la solución lógica y saber como se satisfacen los requerimientos y las restricciones. El Diseño pone de relieve una solución lógica: cómo el sistema cumple con los requerimientos. La esencia del análisis y el diseño orientados a objetos consiste en situar el dominio de un problema y su solución lógica dentro de la perspectiva de los objetos.

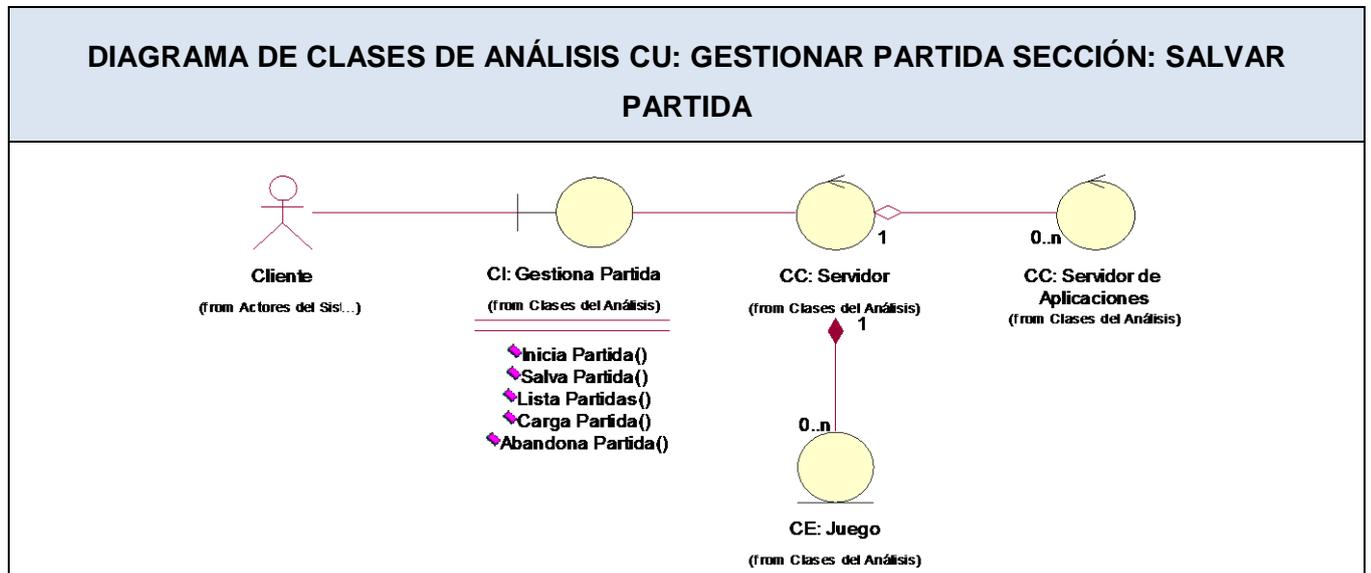
3.2 MODELO DE CLASES DEL ANÁLISIS.

El Modelo de Análisis describe la realización de los casos de uso analizada anteriormente, y es una abstracción del Modelo de Diseño que se adopte. Contiene los resultados de los casos de análisis, y contiene las clases del análisis. Para un buen diseño se recomienda que se realice un Modelo de Análisis bien desrito en la fase de Elaboración, siendo el propósito de este epígrafe, para luego ser refinado en la fase de Construcción y que el tránsito al diseño sea más claro.

3.2.1 Diagrama de clases de análisis.

Las Clases del Análisis mostradas a continuación representan un modelo conceptual que captura las principales responsabilidades del sistema, y a la vez representa el primer paso en la abstracción del mismo. En la presente investigación se realiza un diagrama de clases de análisis por casa caso de uso, a continuación se presenta el del Caso de Uso: Gestionar Partida Sección: Salvar Partida. Los restantes diagramas de clases del análisis de las secciones de Gestionar Partida se encuentran en el Anexo 2.

Tabla 6. Diagrama de Clases del Análisis del CU: Gestionar Partida Sección: Salvar Partida.

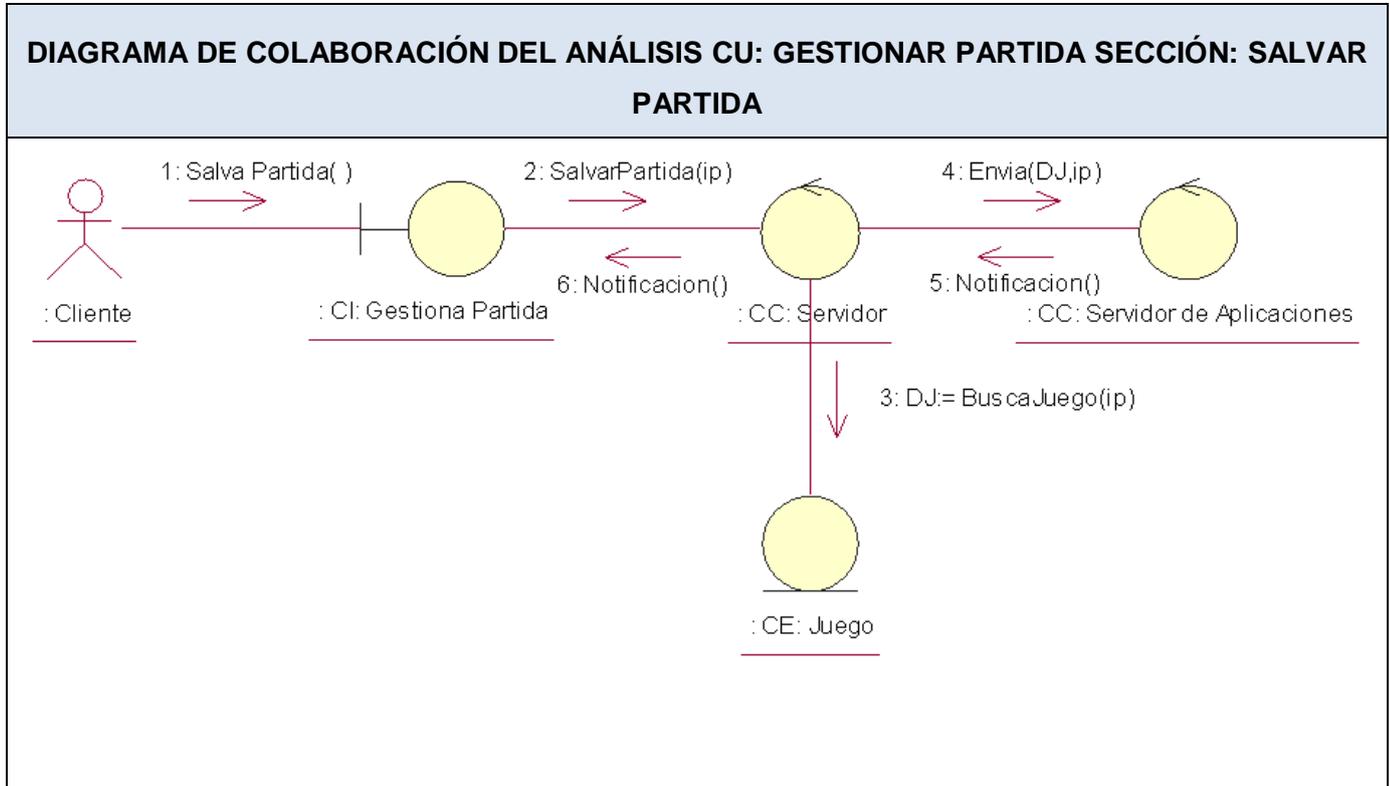


3.2.2 Diagramas de colaboración del análisis.

Los diagramas de interacción se utilizan para modelar aspectos dinámicos de un sistema. Constan de un conjunto de objetos y sus relaciones, incluyendo mensajes que se pueden enviar entre ellos.

En este caso se utiliza un diagrama de colaboración, que es un tipo de diagrama de interacción, porque destaca la organización de los objetos que participan en una interacción. Se construye colocando en primer lugar los objetos que participan en la colaboración como nodos de un grafo, a continuación se representan los enlaces que conectan esos objetos como arcos del grafo. Los cuales se adornan con mensajes que envían y reciben los objetos. Los mismos tienen dos características distintivas, el camino para indicar como se enlaza un objeto a otro, y el número de secuencia para indicar la organización temporal de los mensajes. A continuación se presenta el diagrama de colaboración del Caso de Uso: Gestionar Partida Sección: Salvar Partida. Los demás diagramas de colaboración de las secciones de este caso de uso se encuentran en el Anexo 3.

Tabla 7. Diagrama de Colaboración del Análisis del CU: Gestionar Partida Sección: Salvar Partida.



3.3 MODELO DE CLASES DEL DISEÑO.

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, que ya han sido analizados, junto a otras restricciones, relacionadas con el entorno de implementación tienen impacto en el sistema a considerar. En el modelo de diseño los casos de uso son realizados por las clases del diseño y sus objetos. (Ivar Jacobson, 1999)

3.3.1 Diagramas de clases del diseño.

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Normalmente contiene, clases, asociaciones y atributos; interfaces con sus operaciones y constantes; métodos; información sobre los tipos de atributos; navegabilidad y dependencias. Un diagrama de este tipo contiene las definiciones de las entidades de software.

Se muestra a continuación el Diagrama de Clases del Diseño del CU: Gestionar Partida de la Sección: Salvar Partida. Los restantes diagramas de clases del diseño de las secciones del caso de uso en cuestión se encuentran en el Anexo 4.

Tabla 8. Diagrama de Clases del Diseño del CU: Gestionar Partida Sección: Salvar Partida.

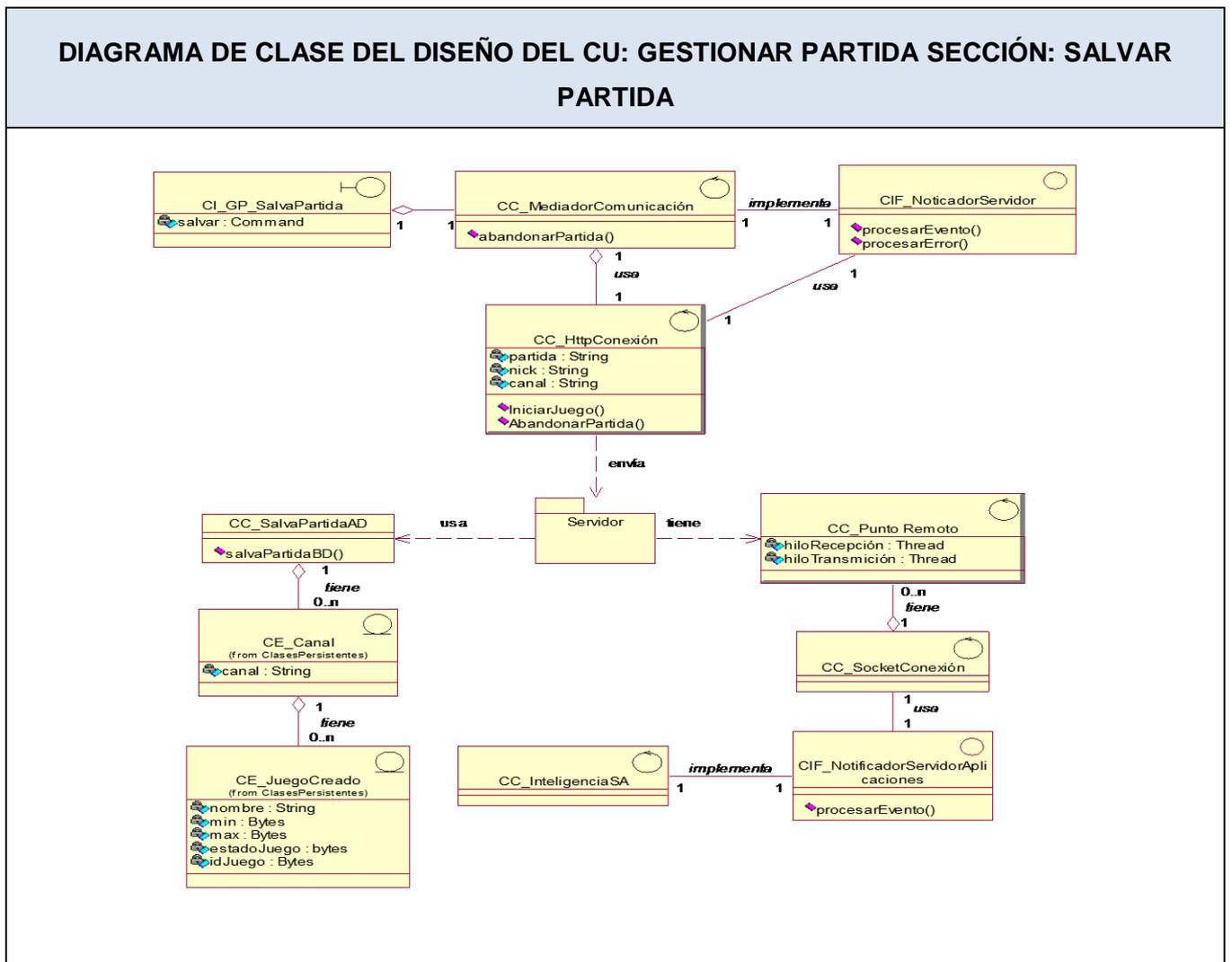
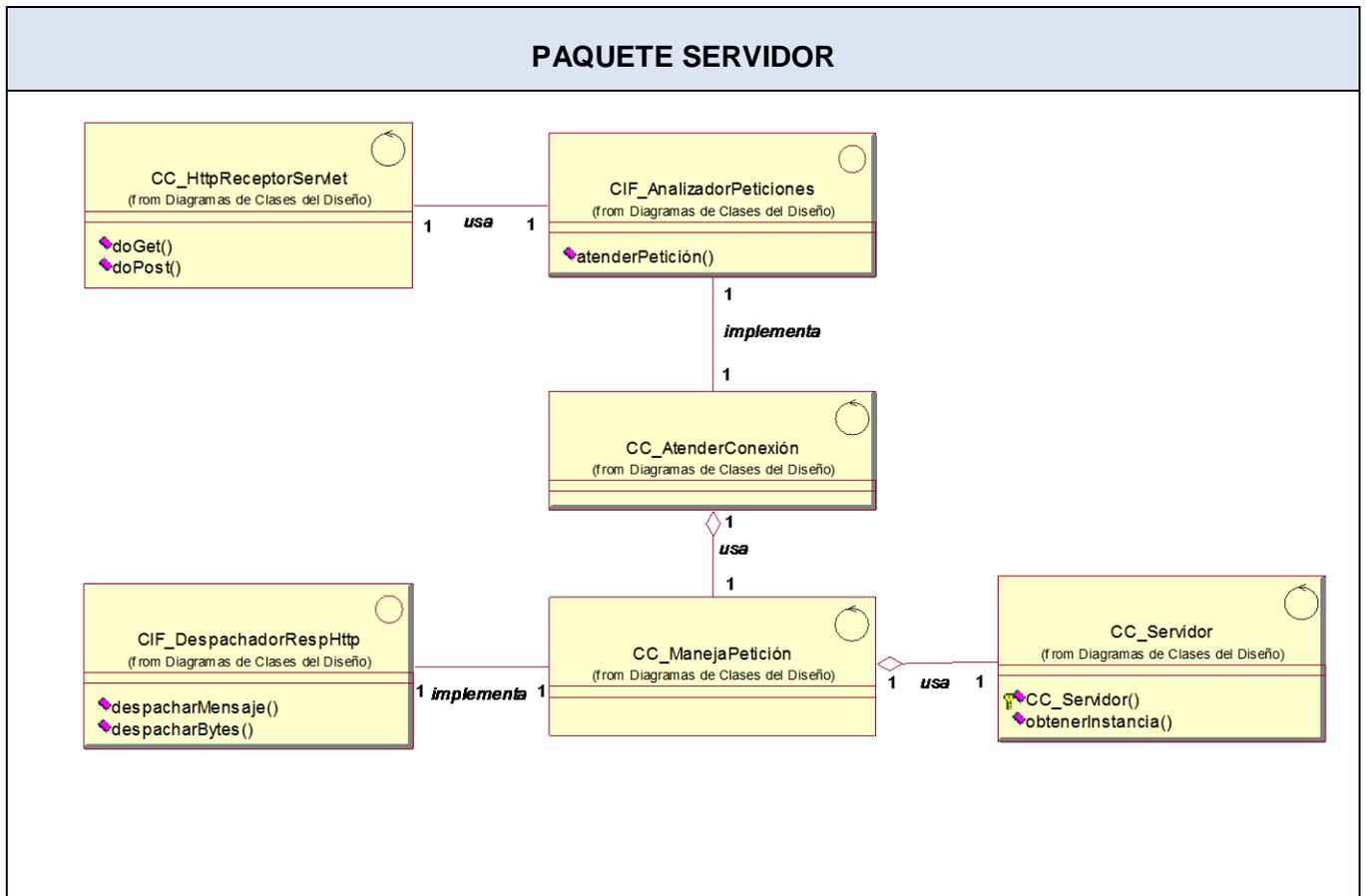


Tabla 9. Descripción del Paquete Servidor.

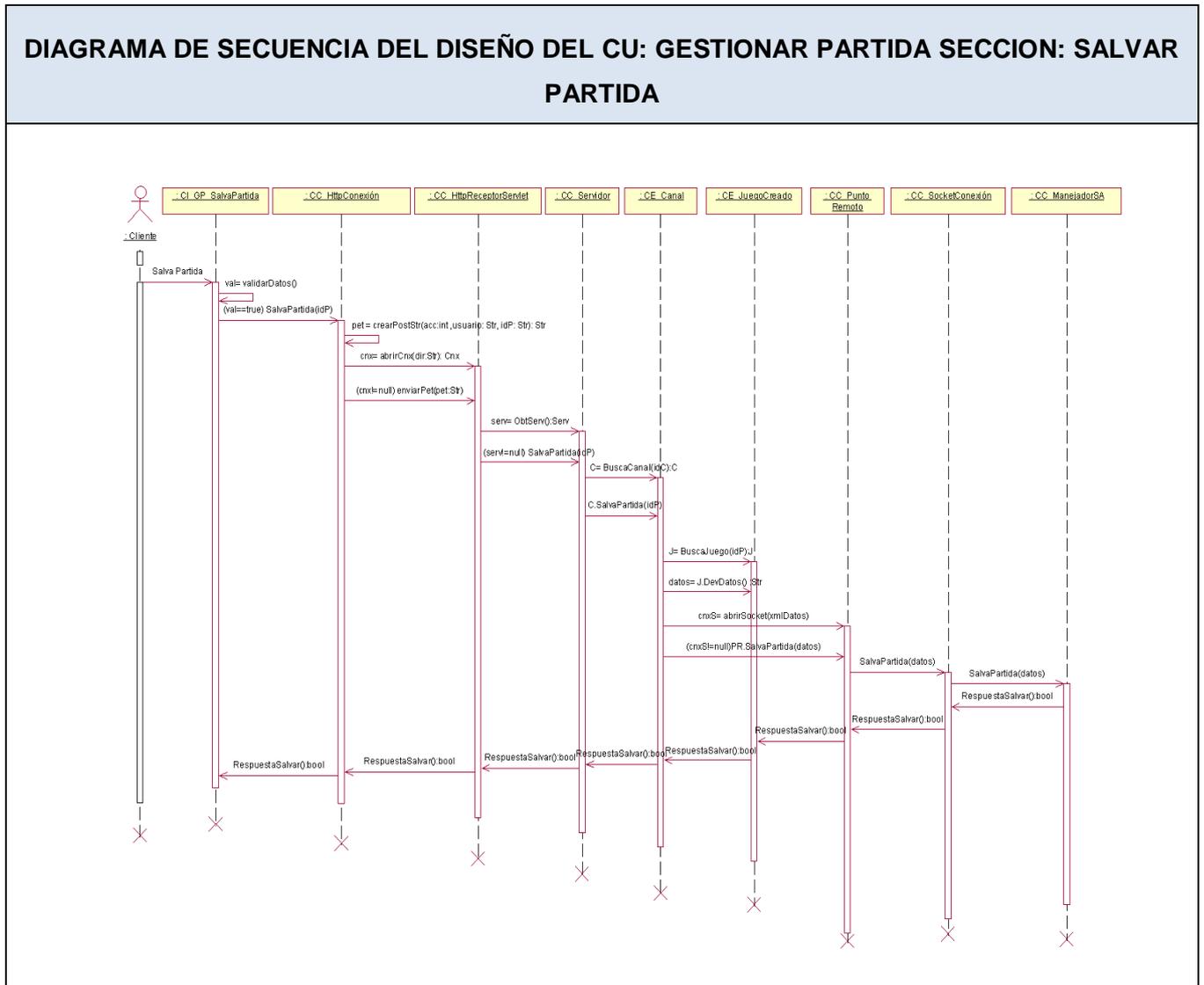


3.3.2 Diagramas de secuencia del diseño.

Un diagrama de secuencia es también un diagrama de interacción que destaca la ordenación temporal de los mensajes. Se forma colocando en primer lugar los objetos que participan en la interacción en la parte superior del diagrama, a lo largo del eje X. Normalmente se coloca a la izquierda el objeto que inicia la interacción, y los subordinados a la derecha. A continuación se ubican los mensajes que estos objetos se envían y reciben a lo largo del eje Y, en orden de sucesión de tiempo, desde arriba hacia abajo. Tienen dos características distintivas, la línea de vida, que representa la existencia de un objeto, a lo largo de un período de tiempo y el foco de control, que representa el período de tiempo durante el cuál un objeto ejecuta una acción. (Larman, 1999)

A continuación se muestra el Diagrama de Secuencia de la sección Salvar Partida del caso de uso más significativo Gestionar Partida.

Tabla 10. Diagrama de Secuencia del Diseño del CU: Gestionar Partida Sección: Salvar Partida.



3.4 DECISIONES DE DISEÑO EMPLEADAS EN EL DESARROLLO DE LA APLICACIÓN.

Un sistema orientado a objetos se compone de objetos que envían mensajes a otros objetos para que lleven a cabo las operaciones. La calidad de diseño de la interacción de los objetos y la asignación de responsabilidades presentan gran variación. Las decisiones poco acertadas dan origen a sistemas y componentes frágiles y difíciles de mantener, entender, reutilizar o extender. La habilidad más

importante en el análisis y el diseño orientados a objetos es asignar eficientemente las responsabilidades a los componentes del software, por ser una actividad ineludible y que repercute en la solidez, capacidad de mantenimiento, y en la reutilización de los componentes de software.

Los diseñadores expertos en orientación a objetos, según se experiencia, han ido formando un amplio repertorio de principios generales y de expresiones que los guían al crear software. A los cuales se les asigna el nombre de patrones, si se codifican en un formato estructurado que describe el problema y su solución, y si se les asigna un nombre sugerente de acuerdo con su descripción.

Los patrones **GRASP** (del inglés: General Responsibility Assignment Software Patterns) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Un diseñador de software sin mucha experiencia en la tecnología de objetos debe dominarlos cuanto antes: constituyen el fundamento de como se diseñará el sistema. (Larman, 1999). En el desarrollo de este Capítulo se han tenido en cuenta los siguientes:

Experto: Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Se usa más que cualquier otro patrón, al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Con el no se pretende designar una idea oscura ni extraña; expresa simplemente la "intuición" de que los objetos hacen cosas relacionadas con la información que poseen. Poniéndolo en práctica se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer los que se les pide. Soporta un bajo acoplamiento, lo que favorece que el sistema sea robusto y de fácil entendimiento.

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

Bajo Acoplamiento: El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo acoplamiento no depende de muchas otras. Si todas las clases dependen de todas, ¿cuánto software se puede extraer de un modo independiente y reutilizarlo en otro proyecto? El Bajo Acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que

acrecientan la oportunidad de una mayor productividad. Es la meta principal que es preciso tener presente en el diseño y un patrón evaluativo que el diseñador aplica al juzgar sus decisiones.

Alta Cohesión: Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable. Es un principio a tener en cuenta en toda decisión de diseño, el desarrollador valora la importancia que le va atribuir. Este patrón mejora la claridad y facilidad con que se entiende el diseño, a menudo genera bajo acoplamiento, proporciona una mayor reutilización, ya que las clases altamente cohesivas tienen funciones específicas, pudiendo ser utilizadas con diversos propósitos.

Controlador: Asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades que el controlador no realiza, las delega en otras clases con las que mantiene un modelo de alta cohesión.

Fabricación Pura: Asignar un conjunto altamente cohesivo de responsabilidades a una clase artificial que no representa nada en el dominio del problema. Entre sus beneficios está brindar soporte a una Alta Cohesión porque las responsabilidades se dividen en una clase de granularidad fina que se centra exclusivamente en un conjunto muy específico de tareas afines. Además puede aumentar el potencial de reutilización debido a que desempeñan funciones específicas, que pueden utilizarse en otras aplicaciones.

Indirección: Asignar la responsabilidad a un objeto intermedio para que medie entre otros componentes o servicios, y éstos no terminen directamente acoplados. El motivo de la Indirección casi siempre es el Bajo Acoplamiento, se agrega un intermediario con el fin de desacoplar otros componentes o servicios.

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. En esta investigación se hace uso de varios patrones pertenecientes al grupo o pandilla de cuatro, **GOF** (del inglés: Gang of Four) A continuación se describen:

Agente Remoto: Cuando un sistema necesita comunicarse con un servicio externo, sugiere crear una clase de software local que represente el servicio externo y asignarle la responsabilidad de contactar con el servicio real.

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

Única Instancia: Este patrón está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Garantiza que una clase sólo tenga una instancia y proporciona un punto de acceso global a ella. Se implementa creando un método que crea una instancia del objeto de una clase sólo si todavía no existe alguno. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor.

(Erich Gamma, 1994).

Un patrón arquitectónico define a una familia de sistemas informáticos en términos de su organización estructural. Un estilo arquitectónico describe componentes y las relaciones entre ellos con las restricciones de su aplicación, la composición asociada y el diseño para su construcción.

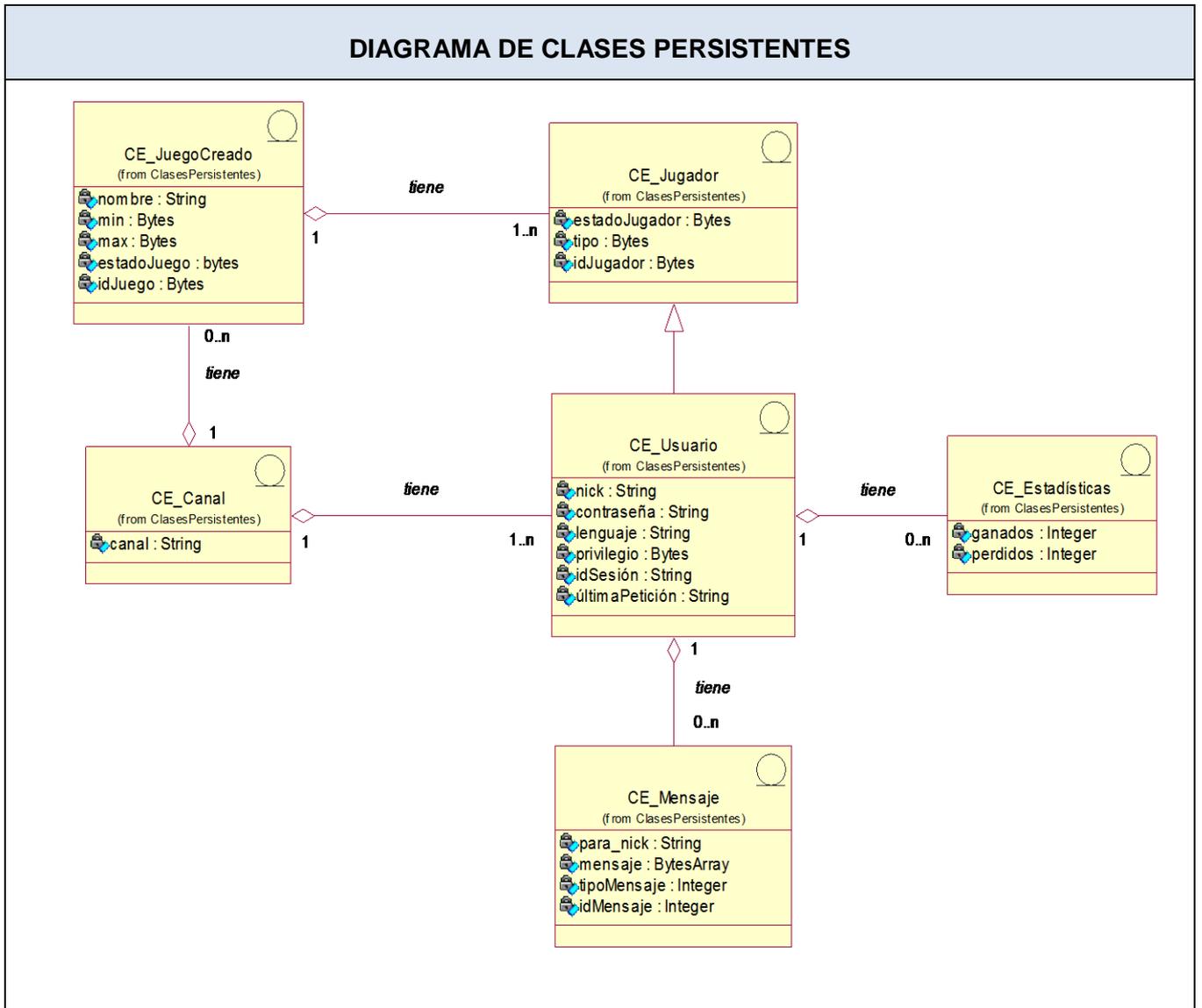
Existen varios patrones de arquitectura, en esta investigación se hace uso de Arquitectura N capas, que consiste en aislar la lógica de la aplicación para convertirla en una capa intermedia bien definida y lógica del software. En la capa de presentación se realiza relativamente poco procesamiento de la aplicación; las ventanas envían a la capa intermedia peticiones de trabajo. Y este se comunica con la capa de almacenamiento del extremo posterior.

3.5 DISEÑO DE BASE DE DATOS.

Para el almacenamiento de la información que maneja la Plataforma SIRIO Mobile, se decidió crear una Base de Datos haciendo uso del sistema de gestión MySQL Server. Luego de escoger las clases que deben persistir, y las relaciones entre ellas, se generó el Diagrama Entidad Relación que refleja las tablas y los campos que contendrá la Base de Datos. De esta manera el Servidor tiene un funcionamiento más eficiente, y su carga de memoria disminuye considerablemente. En la versión anterior los elementos que persistían se guardaban en un fichero XML, por ser este, un sistema que puede aceptar múltiples peticiones, este fichero podía crecer, resultando desfavorable para la velocidad de respuesta a los usuarios. Por lo que en esta versión se usa una base de datos, con el objetivo de mejorar el procesamiento y obtener búsquedas más eficientes.

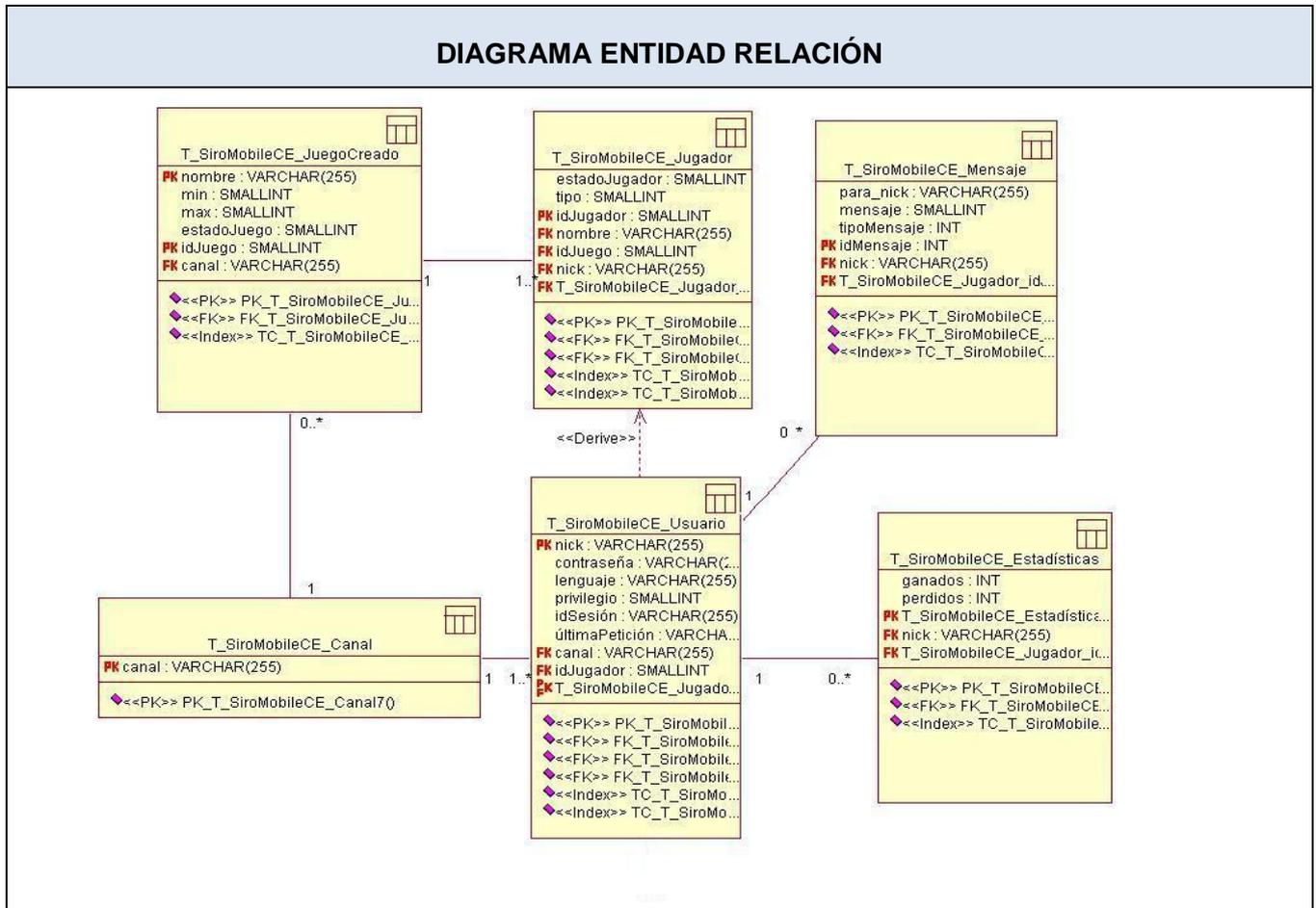
3.5.1 Diagrama de Clases Peristentes.

Tabla 11. Diagrama de Clases Peristentes



3.5.2 Diagrama Entidad Relación

Tabla 12. Diagrama Entidad Relación.



3.6 TRATAMIENTO DE ERRORES.

La Plataforma SIRIO Mobile 1.0 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. En la presente versión se hace uso de este sistema, y se incorporan nuevas posibles defensas frente a acciones incorrectas, sobre todo por la integración del Servidor de Aplicaciones, que conlleva a posibles errores en su conexión con SIRIO Mobile o en la trasmisión de datos entre ellos. A continuación se describen algunos de los posibles errores de la primera versión y otros que se incorporan en la versión 2.0

Tabla 13. Descripción de errores.

ERROR	CAUSA	RESPUESTA DEL SISTEMA
ERROR_ SISTEMA	Error grave del sistema que impide su correcto funcionamiento.	El sistema intenta recuperar la información y restablecer los servicios.
ERROR_ ARGUMENTOS _NO_VÁLIDOS	El usuario realiza una petición con datos vacios o nulos.	Notificar al cliente que existen errores en los datos enviados.
ERROR_USUARIO_AUTENTIFICADO	El usuario decide autentificarse en más de una ocasión.	Notificar al cliente que ya se encuentra autentificado.
ERROR_DESTINATARIO_DESCONOCIDO	El usuario decide enviar un mensaje a un usuario desconocido.	Notificar al cliente que al usuario que desea enviar el mensaje a alguien desconocido.
ERROR_USUARIO_FUERA_LÍNEA	El usuario decide enviar un mensaje a otro usuario que no se encuentra en línea.	Notificar al cliente que el otro usuario no se encuentra.
ERROR_LÓGICA_NO_DISPONIBLE	El usuario decide jugar un juego que tenga la lógica del lado del servidor pero esta no se encuentra disponible.	Notificar al cliente que la lógica del juego no se encuentra disponible.

3.7 SEGURIDAD.

La Plataforma SIRIO Mobile es un sistema distribuido, que necesita que sus términos de cómputo se comuniquen, es importante que la información confidencial que se transmite entre ellos, viaje de forma segura. En el caso de la primera versión de la plataforma, desde el cliente móvil, se envían todos los datos, incluyendo la contraseña usando el protocolo HTTP, que es soportado por todos los browser, sin embargo los datos no están encriptados, por lo que corren el riesgo de ser corrompidos.

En la segunda versión, al iniciar el servidor se carga un fichero XML, donde se le informa a la plataforma qué protocolo usar para el envío de las contraseñas de los usuarios. En caso de que la misma tenga un certificado validado por una entidad certificadora reconocida, este proceso se haría mediante HTTPS. En caso contrario se le informa al cliente que la información que va a introducir al loguearse no es segura, dándole la posibilidad de decidir si desea enviarla. De esta manera SIRIO tiene la opción de usar certificados digitales para garantizar la seguridad de sus datos confidenciales, los restantes se enviarían siempre mediante HTTP, para garantizar un mejor desempeño, y respuestas más rápidas a los clientes. Para almacenar en la Base de Datos las contraseñas de los usuarios se utiliza el algoritmo de encriptación MD5¹³ (Message-Digest Algorithm 5), por ser muy potente y estandarizado a nivel mundial.

La comunicación entre SIRIO Mobile y los servidores de aplicaciones, donde se encuentra la lógica de los juegos, se realiza mediante Socket, la información que se transmite no es secreta por lo tanto no es interesante garantizar la seguridad de estos datos, y sí enfocar el desarrollo en el desempeño de la aplicación, evitando el proceso de encriptación de los datos.

En la versión 2.0 además, se define una variable identificadora de sesión, la cuál se asigna al usuario en el momento de autenticarse en la Plataforma, pudiendo ser verificada su identidad en las posteriores peticiones que realice.

Estas medidas han sido tomadas en función de evitar la interceptación, transformación, suplantación, desvío o pérdida de la información, que puede ser vital para el usuario o para la integridad de la plataforma.

¹³ Es un algoritmo de reducción criptográfico de 128 bits ampliamente usado. La codificación del MD5 de 128 bits es representada típicamente como un número de 32 dígitos hexadecimal.

3.8 CONCEPCIÓN DE LA AYUDA.

La concepción de la ayuda consta de una correcta documentación de las nuevas clases e interfaces añadidas al API Cliente – Juego, explicando claramente su forma de uso, permitiendo a quien implemente una aplicación para ser incorporada a la Plataforma, sepa los mínimos requisitos de integración que debe cumplir y las opciones que tiene para desarrollar su juego.

Igualmente se documenta le descripción del API SIRIO para Servidor Aplicación, por ser de gran importancia para los desarrolladores en el momento de acoplar su inteligencia artificial a la Plataforma SIRIO Mobile. Se explica claramente la forma de conexión, y de intercambio de información entre SIRIO Mobile y el Servidor de Aplicaciones, se describen también las clases e interfaces.

3.9 CONCLUSIONES

En este capítulo se ha podido razonar sobre los aspectos internos del sistema, dándole forma en su totalidad, persiguiendo construirlo como un todo mantenible. Se generaron los artefactos relacionados con el análisis y diseño, mostrando en este documento los Diagramas de Clases y de Interacción. Se definen las clases persistentes y se genera el Diagrama Entidad Relación. Quedan determinadas las decisiones de diseño, definiendo los patrones a utilizar. Además se manejan los posibles errores de la Plataforma y se hace un análisis de la seguridad de SIRIO Mobile 2.0. A partir de este momento están creadas las bases para comenzar la implementación del sistema.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.1 INTRODUCCIÓN.

En la implementación se empieza con el resultado del diseño y se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, ficheros de código binario, ejecutables y similares. La mayor parte de la arquitectura del sistema es capturada durante el diseño, siendo el propósito fundamental de la implementación el desarrollar la arquitectura y el sistema como un todo. Entre los propósitos que persigue se encuentran, planificar las integraciones del sistema necesarias en cada iteración, distribuir el sistema asignando componentes ejecutables en los nodos del diagrama de integración. Además se implementan las clases encontradas durante el diseño y se prueban individualmente todos los componentes, integrándolos, compilándolos y enlazándolos en uno o más ejecutables. Se realizan los release y las pruebas necesarias al sistema una vez concluida todas sus funcionalidades.

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.

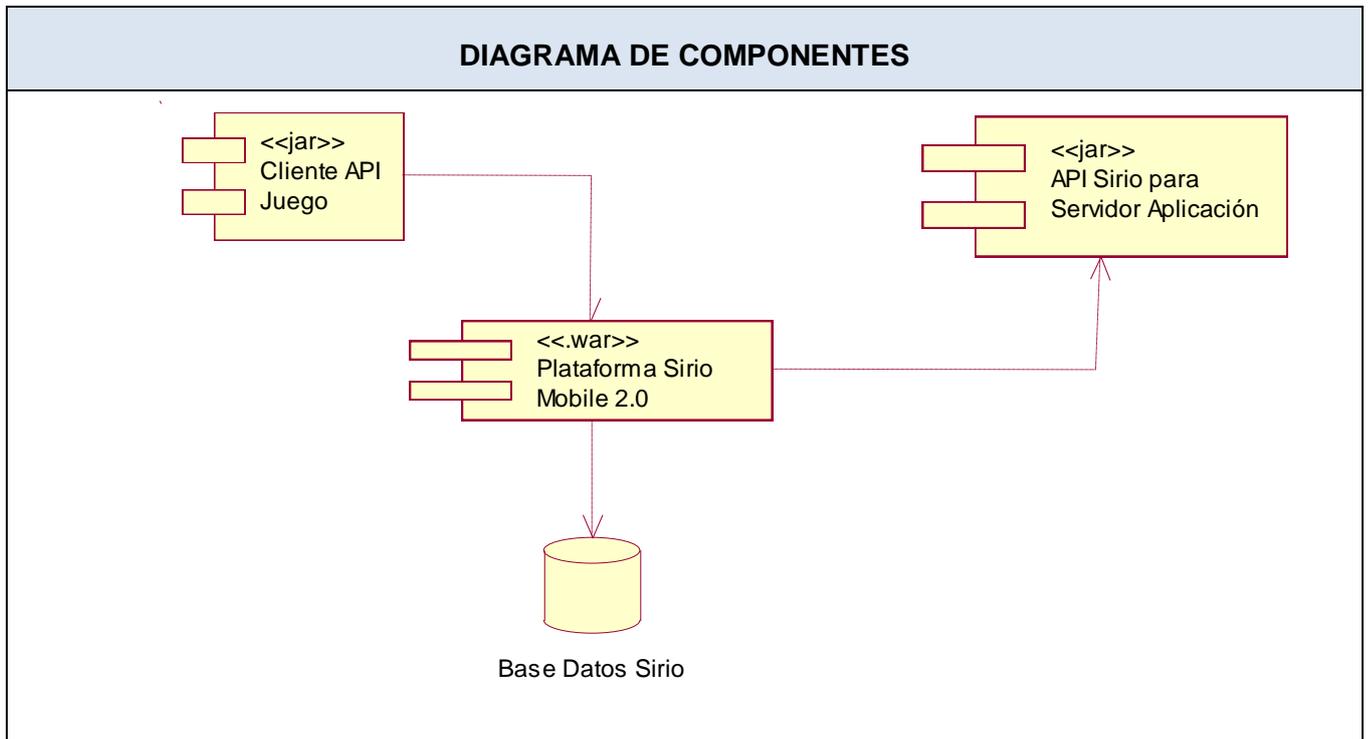
4.2.1 Diagrama de Componentes.

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. (Ivar Jacobson, 1999)

Para dar solución a la propuesta del sistema se desarrollaron una serie de componentes, los cuales se reflejan en el Diagrama de Componentes que se presenta a continuación:

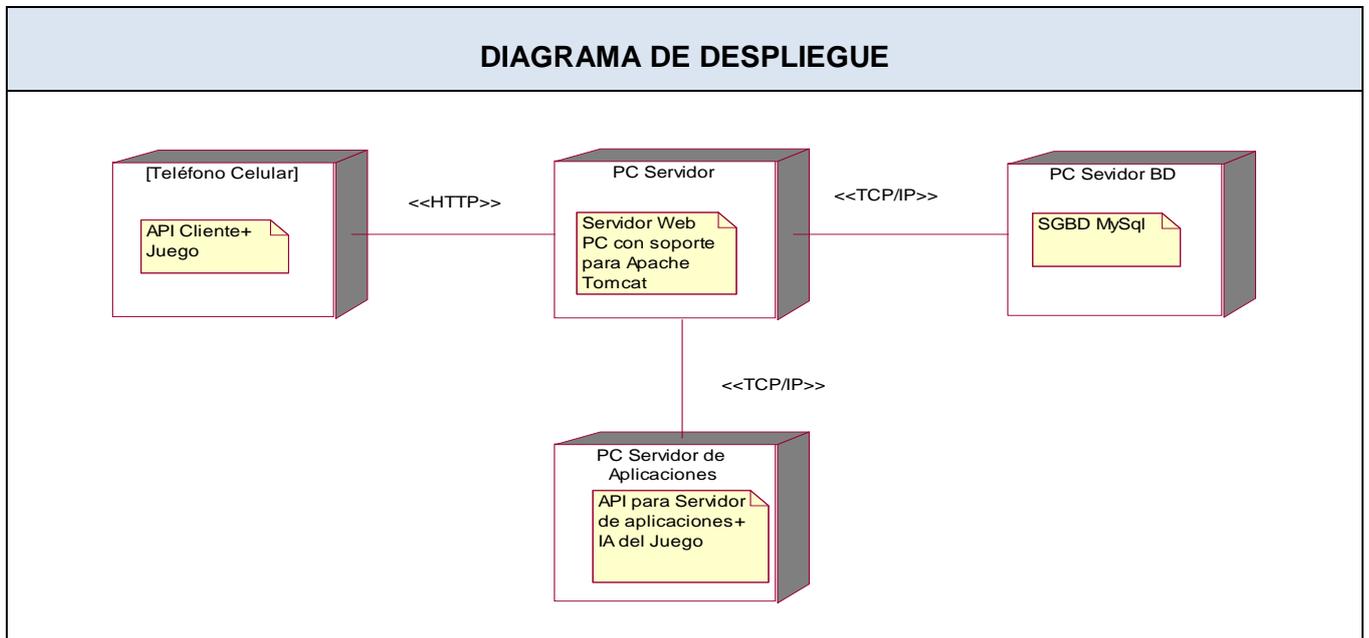
Tabla 14. Diagrama de Componentes



4.2.2 Diagrama de Despliegue.

El Modelo de Despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se construye la funcionalidad entre nodos de cómputo. Se utiliza como entrada fundamental en las actividades de diseño e implementación, debido a que la distribución del sistema tiene una influencia principal en su diseño.

Tabla 15. Diagrama de Despliegue



4.3 MODELO DE PRUEBAS.

El modelo de pruebas describe principalmente como se prueban los componentes ejecutables en el modelo de implementación con pruebas de integración y de sistema. El modelo de pruebas puede describir también como deben ser probados aspectos específicos del sistema. Es una colección de casos de pruebas, procedimientos y componentes de pruebas. (Ivar Jacobson, 1999)

4.3.1 Métodos de Prueba.

Cualquier producto de software puede ser probado de dos formas diferentes. Una es conociendo la funcionalidad específica para la cuál fue diseñado el producto, pudiendo llevar a cabo pruebas que demuestren que cada función es completamente operativa. La otra forma requiere que se conozca el funcionamiento del producto, permitiendo desarrollar pruebas que se aseguren que las operaciones internas se ajustan a las especificaciones y que los componentes se comprueben de forma adecuada.

El primer enfoque se refiere al Método de Prueba de Caja Negra, que se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta a la forma adecuada y que se

produce una salida correcta, así como que la integridad de la información externa se mantiene. En este modelo no se tiene en cuenta la estructura interna del software.

El segundo se basa en el minucioso examen de los detalles procedimentales, conocido como Método de Prueba de Caja Blanca. Se comprueban los caminos lógicos del software proponiendo casos de prueba que examinen que estén correctas todas las condiciones y/o bucles para determinar si el estado real coincide con el esperado. Esto genera gran cantidad de caminos posibles, por lo que hay que dedicar esfuerzos a la determinación de las condiciones de prueba que se van a verificar.

4.3.2 Diseño de casos de prueba. Prueba de Caja Negra.

Con el fin de probar el correcto funcionamiento de la Plataforma SIRIO Mobile, se desarrolló una aplicación de prueba, el juego TicTacToe, conocido como el juego de las cruces, donde participan dos jugadores, sobre él se llevaron a cabo los siguientes casos de prueba con la rigurosidad pertinente:

Caso de Uso: Gestionar Partida Sección: Iniciar Partida

Caso de Prueba: CPR 1: Iniciar Partida

Descripción del flujo

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

Si es el caso de una partida virtual:

- Nombre de la partida
- Tipo de Jugador

Si es el caso de una partida no virtual:

- Nombre de la partida
- Tipo de Jugador
- Nombre de Jugador

Tabla 16. Caso de Prueba Iniciar Partida

Clases Válidas	Clases Inválidas	Resultado esperado	Resultado
<p>Se introducen los siguientes datos:</p> <p>Nombre de la partida: MiTic</p> <p>Tipo de jugador: virtual</p>		Se inicia la partida.	Satisfactorio
<p>Se introducen los siguientes datos:</p> <p>Nombre de la partida: MiTic</p> <p>Tipo de jugador: no virtual</p> <p>Nombre Jugador: Erick (se encuentra disponible en el canal)</p>		Se inicia la partida.	Satisfactorio
	<p>Se introducen los siguientes datos:</p> <p>Nombre de la partida: vacío</p> <p>Tipo de jugador: virtual</p>	Se lanza el error: Argumento no válido.	Satisfactorio
	<p>Se introducen los siguientes datos:</p> <p>Nombre de la partida: MiTic</p> <p>Tipo de jugador: vacío</p>	Se lanza el error: Argumento no válido.	Satisfactorio

Caso de Uso: Registrar Usuario

Caso Prueba: CPR 2: Registrar Usuario

Descripción del flujo

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

- Nombre de usuario
- Contraseña
- Idioma

Tabla 17. Caso de Prueba Registrar Usuario

Clases Válidas	Clases Inválidas	Resultado esperado	Resultado
Se introducen los siguientes datos: Nombre de usuario: Liane Contraseña: Lia Idioma: Español		Se resgistra el usuario.	Satisfactorio.
Se introducen los siguientes datos: Nombre de usuario: Liane Contraseña: Lia Idioma: vacío		Se resgistra el usuario y se le asigna el idioma por defecto de la Plataforma.	Satisfactorio.

	<p>Se introducen los siguientes datos:</p> <p>Nombre de usuario: vacío</p> <p>Contraseña: Lia</p> <p>Idioma: Español</p>	<p>Se lanza el error: Argumento no válido.</p>	Satisfactorio.
	<p>Se introducen los siguientes datos:</p> <p>Nombre de usuario: Liane</p> <p>Contraseña: vacío</p> <p>Idioma: Español</p>	<p>Se lanza el error: Argumento no válido.</p>	Satisfactorio.

Para preparar los casos de prueba se comprobaron un número de datos que ayudaron a la ejecución de los casos de prueba anteriores, que permitieron la ejecución del sistema en todas sus variantes, estos datos pueden ser válidos o inválidos para el programa según si lo que se desea probar es una funcionalidad del sistema o hallar un error. Los datos se escogieron atendiendo a especificaciones del problema, sin importar los detalles internos del programa, a fin de comprobar su correcto funcionamiento. se realizaron pruebas similares a las otras funcionalidades del sistema.

4.4 CONCLUSIONES.

En este capítulo se describieron los artefactos que se generan en el flujo de trabajo implementación y prueba. Se presenta el diagrama de componentes que refleja la forma en que se acoplan los distintos ficheros obtenidos a lo largo de esta investigación y el diagrama de despliegue que especifica en que nodos se encuentran desplegados estos componentes. Se definieron y llevaron a la práctica las pruebas necesarias, utilizando el Método de Prueba de Caja Negra.

CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

5.1 INTRODUCCIÓN.

Un estudio de factibilidad se hace necesario para auxiliar a la entidad Procyon a lograr los objetivos trazados en esta investigación y cubrir las metas con los recursos actuales. Es una investigación altamente enfocada al proyecto de diseño propuesto, se determina la viabilidad económica, financiera, ambiental, técnica y de mercado del proyecto y el esfuerzo a realizar por sus desarrolladores en un período de tiempo.

5.2 PLANIFICACIÓN BASADA EN CASOS DE USO.

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

Se calcula a partir de la siguiente ecuación:

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

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

1.1. Factor de Peso de los Actores sin ajustar (**UAW**)

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

Tabla 18. Factor de peso de los actores sin ajustar.

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	1	1

Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	1	3
Total			2	4

UAW = Σ cantActores * peso

UAW = 4

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

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

Tabla 19. Factor de peso de los casos de uso sin ajustar.

Tipo de CU	Descripción	Factor de peso	Cantidad CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	2	10
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	2	20
Complejo	El caso de uso tiene más de 8 transacciones.	15	1	15
Total			5	45

$$\mathbf{UUCW} = \Sigma \text{cantCU} * \text{peso}$$

$$\mathbf{UUCW} = 45$$

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

$$\mathbf{UUCP} = 4+45$$

$$\mathbf{UUCP} = 49$$

Paso 2. Cálculo de los Puntos de casos de uso ajustados. Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

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

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

2.1 Factor de complejidad técnica (TCF)

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

Tabla 20. Factor de complejidad técnica.

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema Distribuido	2	5	10
T2	Tiempo de Respuesta	1	4	4
T3	Eficiencia del Usuario Final	1	5	5
T4	Funcionamiento Interno Complejo	1	4	4
T5	El código debe ser reutilizable	1	5	5
T6	Facilidad de instalación	0.5	4	2
T7	Facilidad de uso	0.5	4	2
T8	Portabilidad	2	5	10
T9	Facilidad de cambio	1	4	4
T10	Concurrencia	1	4	4
T11	Incluye objetivos especiales de seguridad	1	5	5
T12	Provee acceso directo a terceras partes	1	4	4
T13	Facilidades especiales de entrenamientos.	1	4	4
Total				63

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

$$\text{TCF} = 0.6 + 0.01 * 63$$

$$\text{TCF} = 1.23$$

2.2 Factor de ambiente (EF)

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

Tabla 21. 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 Objetos	1	4	4
E4	Capacidad del analista líder	0.5	5	2.5
E5	Motivación	1	5	5
E6	Estabilidad de requerimientos	2	4	8
E7	Personal Part-Time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	1	-1
Total				24

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

$$EF = 1.4 - 0.03 * 24$$

$$EF = 0.68$$

$$UCP = UUCP * TCF * EF$$

$$UCP = 49 * 1.23 * 0.68$$

$$UCP = 40.9836$$

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:

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

$$E = 40.9836 * 20$$

$$E = 819.672$$

Paso 4. Calcular esfuerzo de todo el proyecto

Tabla 22. Estimación del esfuerzo total del proyecto.

Actividad	Pocentaje %	Horas – Hombres
Análisis	20	409.836
Diseño	20	409.836
Implementación	40	819.672
Pruebas	10	204.918
Sobrecarga(otras actividades)	10	204.918
Total	100	2049.18

Si $E_T = 2049.18$ horas-hombre y se estima que cada mes tiene como promedio 192 horas laborables, eso daría un $E_T \approx 10.67$ mes-hombre.

Esto quiere decir que 1 persona puede realizar el problema analizado en 10 meses y medio aproximadamente. Entre dos, el proyecto consume 5 meses aproximadamente.

-Costo del Proyecto.

Se asume como salario promedio mensual \$100.00

CH: Cantidad de hombres.

Tiempo: Tiempo total del proyecto.

CH = 2 hombres

CHM = 2 * Salario Promedio

CHM = 100.00 \$/mes

Costo = CHM * E_T / CH

Costo = 200.00 * 10.6728125 / 2

Costo = \$ 2134.5625 ≈ \$ 2134.56

Tiempo = E_T / CH

Tiempo = 10.6728125 / 2

Tiempo = 5.33640625 ≈ 5.34 meses

De los resultados obtenidos se interpreta que con 2 hombres trabajando en el proyecto el mismo se desarrolla en 5.34 meses y su costo total se estima que sea \$2134.56

5.3 BENEFICIOS TANGIBLES E INTANGIBLES.

La entidad Procyon posee en estos momentos de una Plataforma que gestiona las peticiones de distintos clientes móviles, permitiéndoles jugar vía GPRS. A partir de esta investigación se aportan un Cliente API Juego y un API SIRIO para Servidor Aplicación El equipo de desarrollo hace entrega de un sistema completamente funcional y robusto, al cuál se le ha aplicado una correcta Ingeniería del Software. A continuación se describen los beneficios fundamentales que aporta SIRIO Mobile 2.0, estrechamente vinculados a sus tres módulos principales, que contienen funcionalidades específicas:

1. Se cuenta con un Servidor que maneja todas las peticiones desde los dispositivos móviles de forma dinámica, y brindándoles una rápida respuesta. De esta forma la entidad cuenta con el medio para transmitir datos, como mensajes y jugadas a otro dispositivo móvil.
2. Mediante el Cliente API Juego se permiten desarrollar juegos de tipo multiusuario que se pueden integrar con el Servidor, mediante sus interfaces, permitiendo añadirle las funcionalidades de conexión vía GPRS y disminuyendo el tiempo de desarrollo de este tipo de aplicaciones.

3. En esta etapa también se obtuvo un API SIRIO para Servidor Aplicación, que permite a los desarrolladores implementar un servidor de aplicaciones donde persistan la lógica de sus juegos y los datos de los mismos. Facilitándoles la integración con el Servidor de SIRIO Mobile 2.0 mediante el uso de interfaces vía Socket.

Otros beneficios que se obtienen 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 sólo 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 realizó un estudio de la factibilidad del sistema desarrollado en esta investigación. Mediante el método de análisis de casos de uso se determina el esfuerzo y el tiempo invertido en el proyecto, se analizan además los costos y los beneficios que reportará el producto.

CONCLUSIONES

Después de culminar la investigación se le da solución al objetivo planteado de incorporarle funcionalidades a la Plataforma SIRIO Mobile, la misma permite salvar el estado de las aplicaciones que maneja e iniciar partidas con jugadores virtuales, haciéndola más funcional y atractiva. Para su cumplimiento se ofrece el Cliente API Juego y un API SIRIO para Servidor Aplicación, además de la Plataforma que maneja las peticiones que se hacen mediante los APIs mencionados, haciendo posible que dos o más jugadores puedan jugar vía GPRS. Para la consecución de este objetivo se realizó un estudio de la Telefonía Móvil y las Comunicaciones Inalámbricas, se realizó un profundo estudio del estado del arte de las plataformas de desarrollo de juegos para móviles, ubicando a SIRIO Mobile en las tendencias más actuales. También se hizo un análisis de la seguridad de este tipo de aplicación.

Se determinaron las herramientas más idóneas, Rational Rose Enterprise Edition 2003 para el modelado de la aplicación, Eclipse como entorno de desarrollo, TortoiseSVN para el control de versiones, lenguaje de programación Java, el sistema gestor de base de datos MySql Server, como contenedor web el Apache Tomcat. Haciendo uso de la metodología RUP se generaron todos los artefactos necesarios para la comprensión y reutilización del sistema mediante las mejores prácticas de la Ingeniería del Software. Se realizaron las pruebas suficientes para hacer entrega a la entidad Procyon de un producto fiable.

La aplicación está acompañada de una bibliografía actual y muy variada, se hizo uso de varias publicaciones confiables de internet y de otras en formato duro. También se desarrolló un juego de prueba llamado TicTacToe para probar las funcionalidades de la Plataforma.

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

RECOMENDACIONES

Se recomienda para próximas versiones permitir incorporar redes de amigos a los juegos que contenga la Plataforma. Además brindar la posibilidad de crear torneos para los usuarios.

BIBLIOGRAFÍA

Apexnet. 2005. <http://www.apexnet.com.ar>. *http://www.apexnet.com.ar*. [Online] Junio 25, 2005. [Cited: Noviembre 20, 2007.] <http://www.apexnet.com.ar/index.php/news/main/38/event=view>.

Aurora. 2002. <http://www.aurora.com.do>. *http://www.aurora.com.do*. [Online] Enero 2002. [Cited: Octubre 16, 2007.] <http://www.aurora.com.do/gsm.htm>.

Bonet, Esteban. *Creación y administración de certificados de seguridad.*

Burbano, Diego Javier. 2006. <http://www.mysql-hispano.org>. [Online] Mayo 5, 2006. <http://www.mysql-hispano.org/articulos/num43/analisis-comparativo.pdf>.

Catarina. 2006. <http://catarina.udlap.mx>. *http://catarina.udlap.mx*. [Online] julio 2006. [Cited: Noviembre 9, 2007.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf.

Copyright © 1999-2006, Apache Software Foundation. 2006. <http://tomcat.apache.org/>. *http://tomcat.apache.org/*. [Online] 2006.

Díaz, Andrés. *Gestión de Certificados.*

Denys Buedo, Yunisel Viera. 2007. *SIRIO, Plataforma para la gestión de juegos multiusuario en dispositivos móviles vía GPRS.* Ciudad de la Habana : s.n., 2007.

Enlys Molina, Juan Carlos. 2002. <http://www.monografias.com>. *http://www.monografias.com*. [Online] 2002. [Cited: Octubre 18, 2007.] <http://www.monografias.com/trabajos13/gpts/gpts.shtml?monosearch>.

Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. 1994. *Design Patterns* . 21 : 10, 1994.

ExitGames. 2006. <http://www.exitgames.com/>. *http://www.exitgames.com/*. [Online] 2006. [Cited: Octubre 25, 2007.] <http://www.exitgames.com/product>.

Exmachina. 2005. <http://www.exmachina.n>. *http://www.exmachina.n*. [Online] 2005. <http://www.exmachina.nl/sp.html>.

GameFederation. 2005. <http://www.gamefederation.com>. *http://www.gamefederation.com*. [Online] 2005. [Cited: Octubre 28, 2007.] <http://www.gamefederation.com/?nav=products.products..>

García, Belarmino. 2006. <http://www.coit.es>. *http://www.coit.es*. [Online] 2006. [Cited: Octubre 18, 2007.] <http://www.coit.es/publicaciones/bit/bit147/1%20UMTS.pdf>.

García, Yoany Rodríguez. 2007. <http://www.monografias.com>. *http://www.monografias.com*. [Online] Enero 2007. [Cited: Octubre 18, 2007.] <http://www.monografias.com/trabajos44/telefonía-movil-cuba/telefonía-movil-cuba2.shtml#red>.

Guardado, Jacobo Pérez. <http://www.lfcia.org>. [Online] <http://www.lfcia.org/openprojects/camllets/doc/html/node16.html>.

Ivar Jacobson, Gardy Booch, James Rumbaugh. 1999. *El proceso unificado de desarrollo del software*. 1999.

Iturralde, Juan Luis. 2001. *El Certificado Digital*. 2001.

Jiménez, Jose Juan. 2004. <http://www.ilustrados.com>. *http://www.ilustrados.com*. [Online] Febrero 6, 2004. [Cited: 10 15, 2007.] <http://www.ilustrados.com/publicaciones/EpZyEkkuVlwxnCdrUU.php>.

Larman, Craig. *UML y Patrones*. —. 1999. *UML y Patrones*. 1999.

Lizama, Rafael Ricardo Sánchez. 2007. <http://www.monografias.com>. *http://www.monografias.com*. [Online] 2007. [Cited: Octubre 15, 2007.] <http://www.monografias.com/trabajos30/telefonía-celular-universitarios/telefonía-celular-universitarios.shtml>.

Martín Inzaurrealde, Jorge Isi, Javier Garderes. 2006. <http://www.cabinas.net>. *http://www.cabinas.net*. [Online] 2006. [Cited: Octubre 15, 2007.] http://www.cabinas.net/monografias/tecnología/generaciones_de_la_telefonía_celular.asp.

Navarro, Genís Berbel. 2005. <http://www.webmovilgsm.com>. *http://www.webmovilgsm.com*. [Online] 2005. [Cited: Octubre 18, 2007.] <http://www.webmovilgsm.com/gprs.htm>.

Nokia. 2004. <http://www.nokia.com.co>. *http://www.nokia.com.co*. [Online] marzo 22, 2004. <http://www.nokia.com.co/A4600316?newsid=-9978>.

Penalva, Javier. 2007. <http://www.xatakamovil.com>. [Online] Junio 27, 2007. <http://www.xatakamovil.com>.

PixelPlay. 2006. <http://www.pixelplay.com>. *http://www.pixelplay.com*. [Online] 2006. [Cited: Noviembre 3, 2007.] <http://www.pixelplay.com/technology/index.html>.

Radioptica. 2007. <http://www.radioptica.com>. *http://www.radioptica.com*. [Online] 2007. [Cited: Octubre 15, 2007.] http://www.radioptica.com/Radio/telefonía_movil.asp.

Sepulveda, Daniel. <http://www.wikilearning.com>. *http://www.wikilearning.com*. [Online] http://www.wikilearning.com/curso_gratis/protocolos_seguros_para_el_web-ssl_secure_socket_layer_y_tls_transport_layer_secure/6091-4.

Syscom. 2005. <http://www.syscom.com.mx>. *http://www.syscom.com.mx*. [Online] 2005. [Cited: Octubre 9, 2007.] <http://www.syscom.com.mx/PPT/ampcelnextel.pdf>.

Unipier. 2005. <http://www.unipier.com>. *http://www.unipier.com*. [Online] 2005. <http://www.unipier.com/?id=138>.

V, Héctor Tejada. 2005. <http://www.fismat.umich.mx>. [Online] Agosto 12, 2005. <http://www.fismat.umich.mx/mn1/manual/node24.html>.

W3. 2006. <http://www.w3.org>. *http://www.w3.org*. [Online] Septiembre 29, 2006. [Cited: Noviembre 20, 2007.] <http://www.w3.org/TR/REC-xml/>.

ANEXOS

Anexo 1: Descripción detallada de los casos de uso.

Tabla 23. Descripción textual del Caso de Uso Gestionar Partida Sección: Iniciar Partida, Listar Partidas, Cargar Partida, Abandonar Partida.

Sección “Iniciar Partida”.	
Acción del Actor	Respuesta del Sistema
3.1 El cliente trasmite al sistema la petición de dar inicio a una partida virtual.	3.2 El sistema recibe la petición, y la envía al Servidor de Aplicaciones.
	3.3 Se crea la partida virtual y se le informa al Cliente que puede dar comienzo a la partida.
Poscondiciones	Se inicia la partida.
Sección “Listar Partidas”	
Acción del Actor	Respuesta del Sistema
5.1 El cliente trasmite al sistema la petición de listar partidas.	5.2 El sistema recibe la petición y verifica que existan partidas
	5.3 Si existen partidas salvadas se le muestran al cliente.
Flujos Alternos	Respuesta del Sistema
	5.4 Si no existen partidas salvadas se le muestra un mensaje de información el cliente.
Poscondiciones	Las partidas son listadas.
Sección “Cargar Partida”	

Acción del Actor	Respuesta del Sistema
6.1 El cliente envía los datos de la partida que desea reanudar.	6.2 El sistema recibe la petición y le envía toda la información referente a dicha partida.
Poscondiciones	La partida es reanudada.
Sección “Abandonar Partida”	
Acción del Actor	Respuesta del Sistema
7.1 El cliente solicita abandonar la partida.	7.2 El sistema recibe la petición y notifica a todos los participantes
Poscondiciones	La partida es abandonada.

Tabla 24. Descripción textual del Caso de Uso Gestiona Paquete Sección: Envía Paquete, recibe Paquete.

Caso de Uso	Gestiona Paquete
Actor	Cliente
Resumen	El caso de uso se inicia cuando el Cliente intercambia información con el jugador virtual.
Precondiciones	Partida iniciada.
Referencias	RF2,RF2.1,RF2.2
Prioridad	Secundaria
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

1. Si el Cliente realiza una petición, ir a la sección “Envía Paquete”.	
2. Si el Cliente espera una respuesta, ir a la sección “Recibe Paquete”.	
Sección “Envía Paquete”	
Acción del Actor	Respuesta del Sistema
1.1 El Cliente realiza una jugada.	1.2 El sistema recibe la petición y actualiza el estado del juego.
	1.3 Verifica que haya comunicación con el Servidor de Aplicaciones.
	1.4 Si hay comunicación con el Servidor de Aplicaciones envía la petición.
Flujos Alternos	
	1.5 Si no hay Comunicación con el Servidor de Aplicaciones, se establece, y luego se envía la petición.
Poscondiciones	Paquete enviado.
Sección “Recibe Paquete”	
Acción del Actor	Respuesta del Sistema
2.2 El cliente recibe la información y la procesa.	2.1 La Plataforma recibe información del Servidor de Aplicaciones y la transmite al cliente.
Poscondiciones	Paquete recibido.

Tabla 25. Descripción textual del Caso de Uso Actualiza configuración del Servidor.

Caso de Uso	Actualiza configuración del Servidor	
Actor	Reloj	
Resumen	El caso de uso se inicia cuando un pulso de Reloj del propio sistema, da inicio al proceso de lectura del fichero de configuración de la Plataforma SIRIO Mobile, si se ha modificado, se actualizan los cambios en el Servidor.	
Referencias	RF3	
Prioridad	Secundaria	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Reloj envía un aviso de verificación.	2. El sistema lee el fichero de configuración, si se percata de que han habido cambios, actualiza su configuración.	
Precondiciones	Se actualiza la Configuración de la Plataforma.	

Tabla 26. Descripción textual del Caso de Uso Enviar Contenido.

Caso de Uso	Enviar Contenido	
Actor	Cliente	
Resumen	El caso de uso se inicia cuando un cliente desea enviar una imagen o video a otro, mediante la Plataforma SIRIO Mobile.	
Referencias	RF4	
Prioridad	Secundaria	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Cliente envía un contenido.	2. El sistema recibe la petición y direcciona el flujo de bytes hacia el destinatario.	
Precondiciones	Se envía el flujo de bytes.	

Tabla 27. Descripción textual del Caso de Uso Registrar Usuario.

Caso de Uso	Registrar Usuario	
Actor	Cliente	
Resumen	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. Es un caso de uso analizado en la versión anterior, en esta se amplía su descripción incorporándole la opción al usuario de escoger el idioma que prefiera.	
Referencias	RF4	
Prioridad	Secundaria	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Cliente solicita registrarse en la Plataforma enviando sus datos.	2. El sistema recibe la petición y valida los datos recibidos.	
	2.1 Si no existen anomalías se registra al usuario.	
	2.2 Si existen errores en los datos se le envía un mensaje al usuario, indicándole que repita la operación.	
Precondiciones	El usuario es registrado en el servidor permitiéndosele acceder a los distintos juegos con las credenciales proporcionadas.	

Anexo 2: Diagramas de Clases del Análisis.

Tabla 28. Diagrama de Clases del Análisis CU: Gestionar Partida Sección: Iniciar Partida

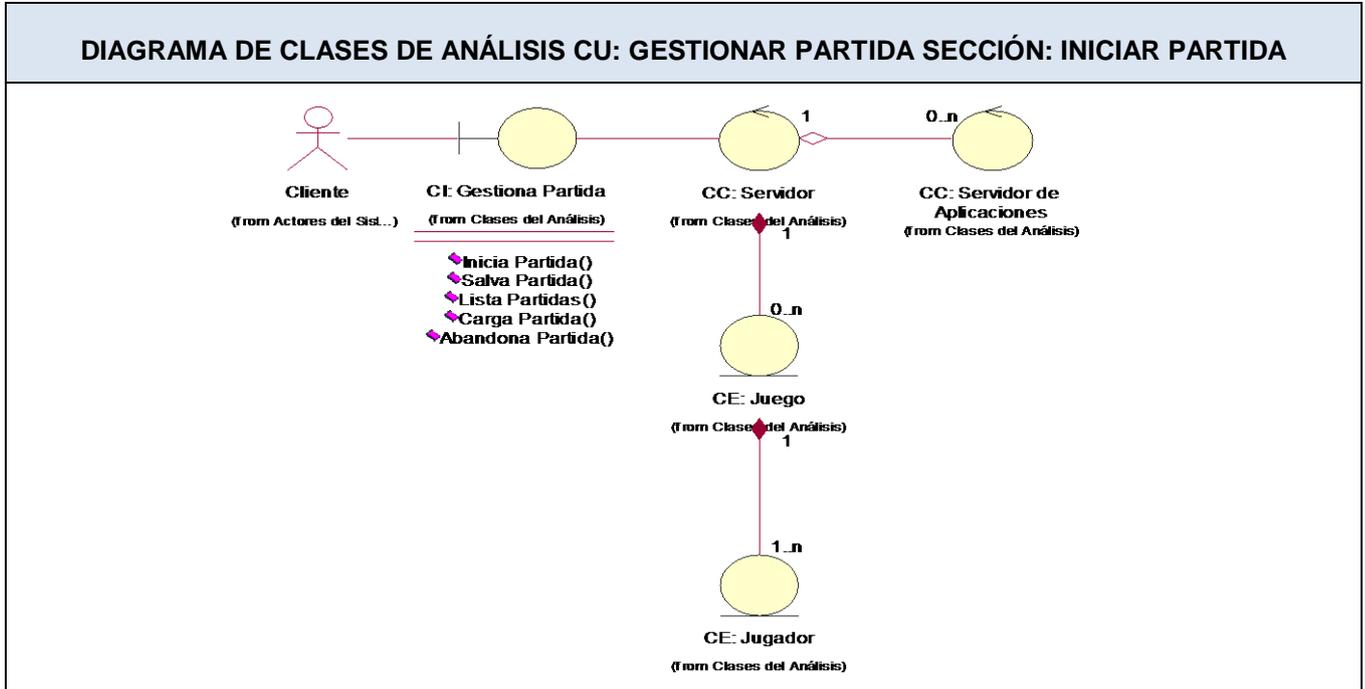


Tabla 29. Diagrama de Clases del Análisis CU: Gestionar Partida Sección: Listar Partidas.

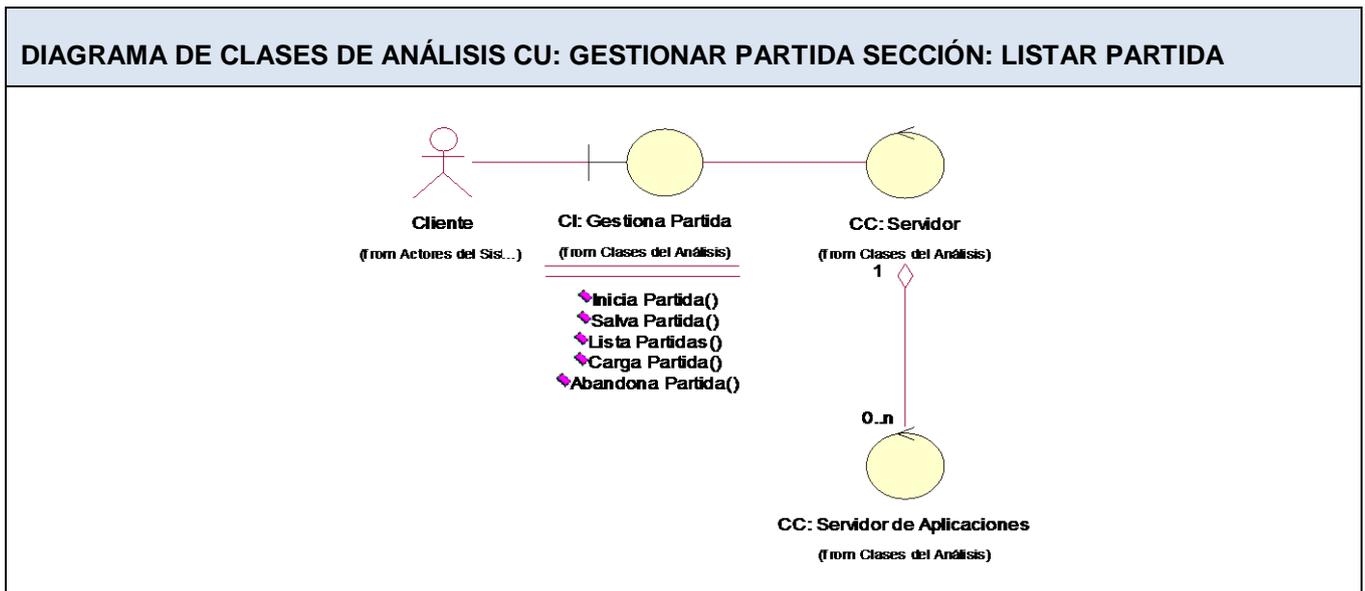


Tabla 30. Diagrama de Clases del Análisis CU: Gestionar Partida Sección: Cargar Partida.

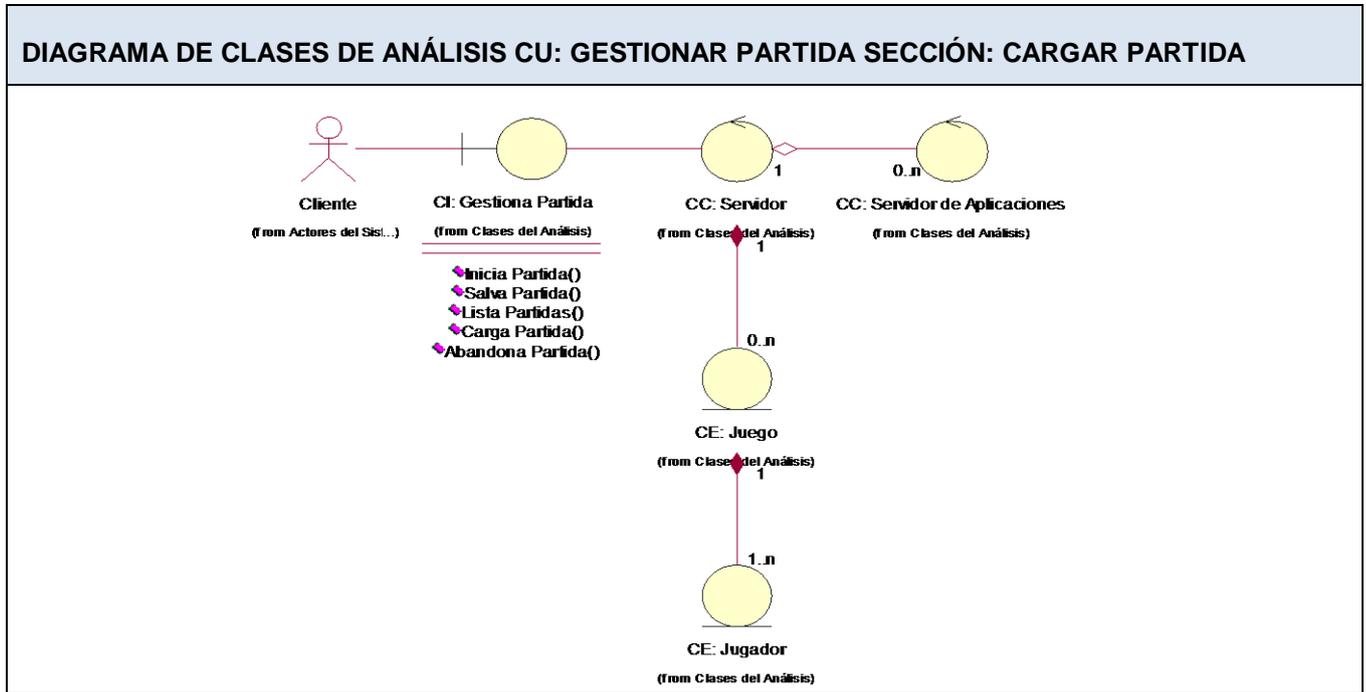
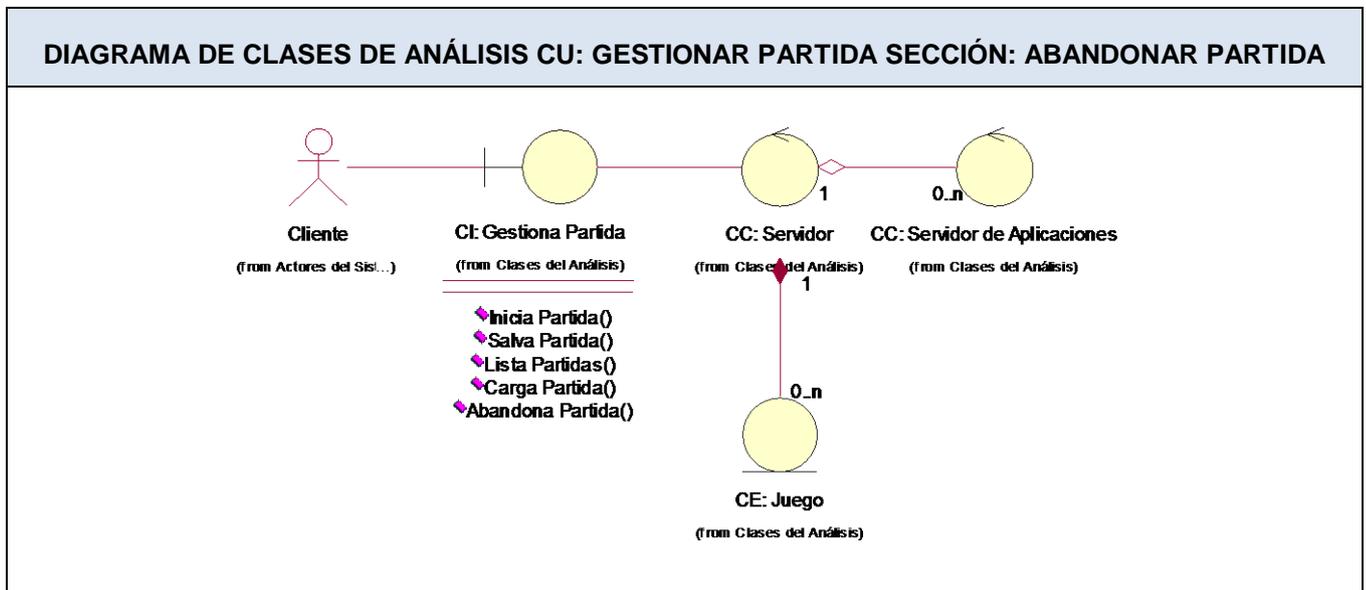


Tabla 31. Diagrama de Clases del Análisis CU: Gestionar Partida Sección: Abandonar Partida.



Anexo 3: Diagramas de Colaboración del Análisis.

Tabla 32. Diagrama de Colaboración del Análisis del CU: Gestionar Partida Sección: Iniciar Partida.

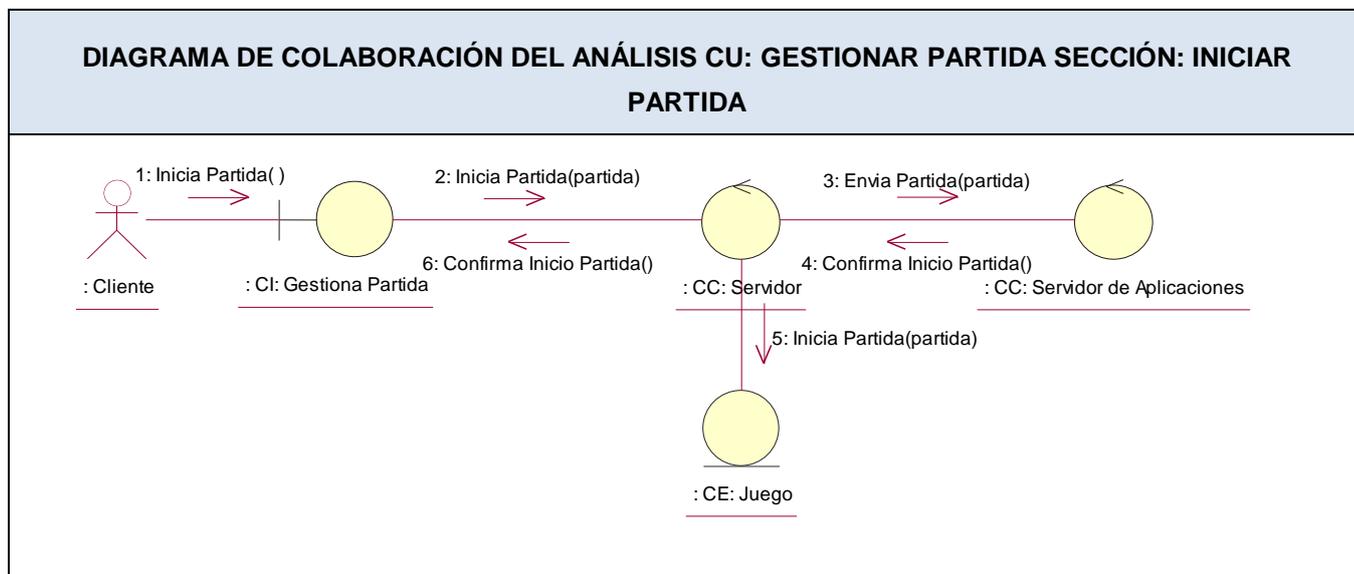


Tabla 33. Diagrama de Colaboración del Análisis del CU: Gestionar Partida Sección: Listar Partidas.

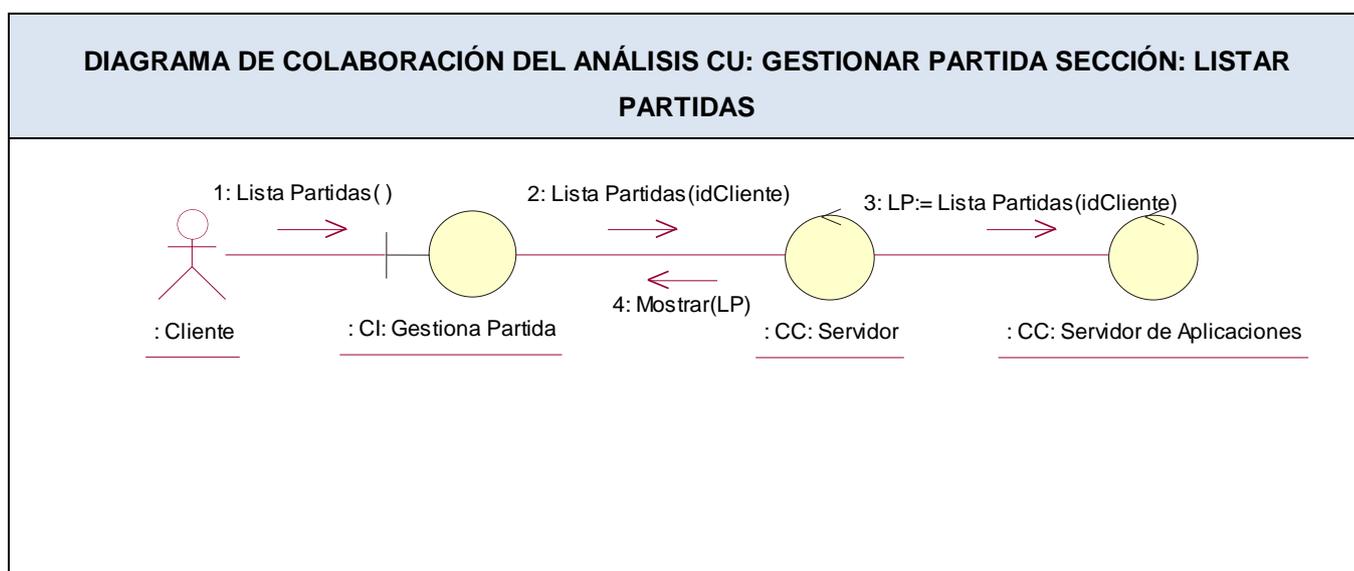


Tabla 34. Diagrama de Colaboración del Análisis del CU: Gestionar Partida Sección: Cargar Partida.

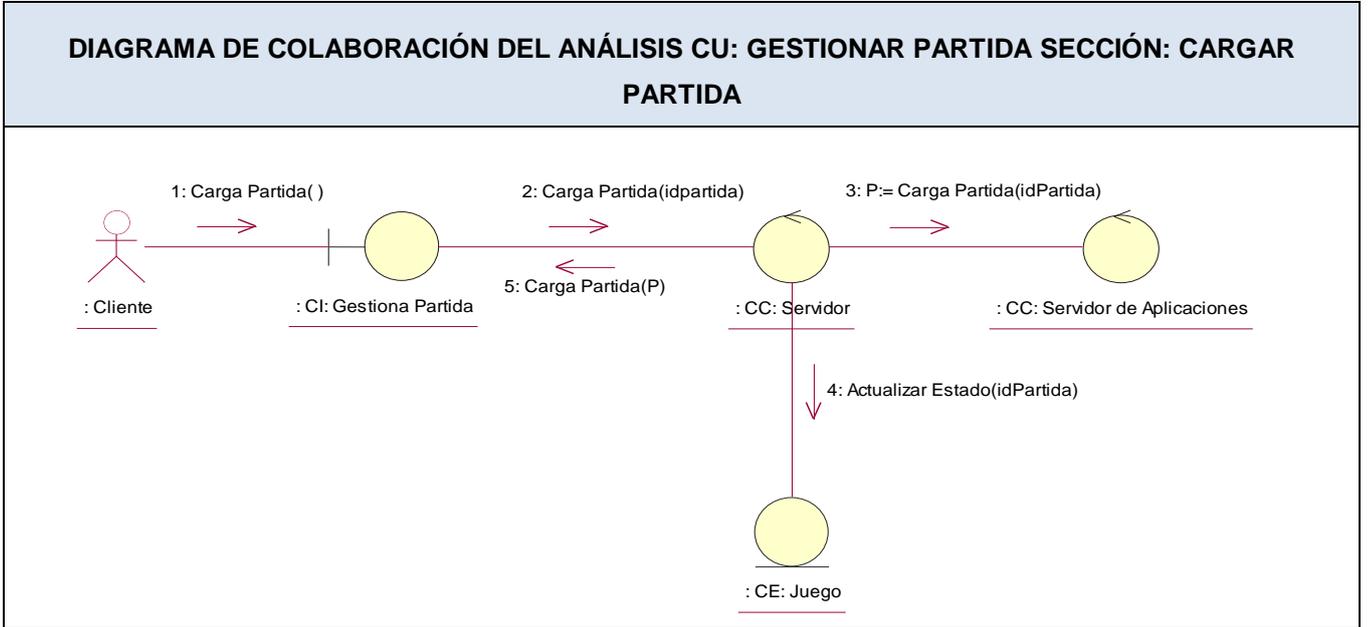
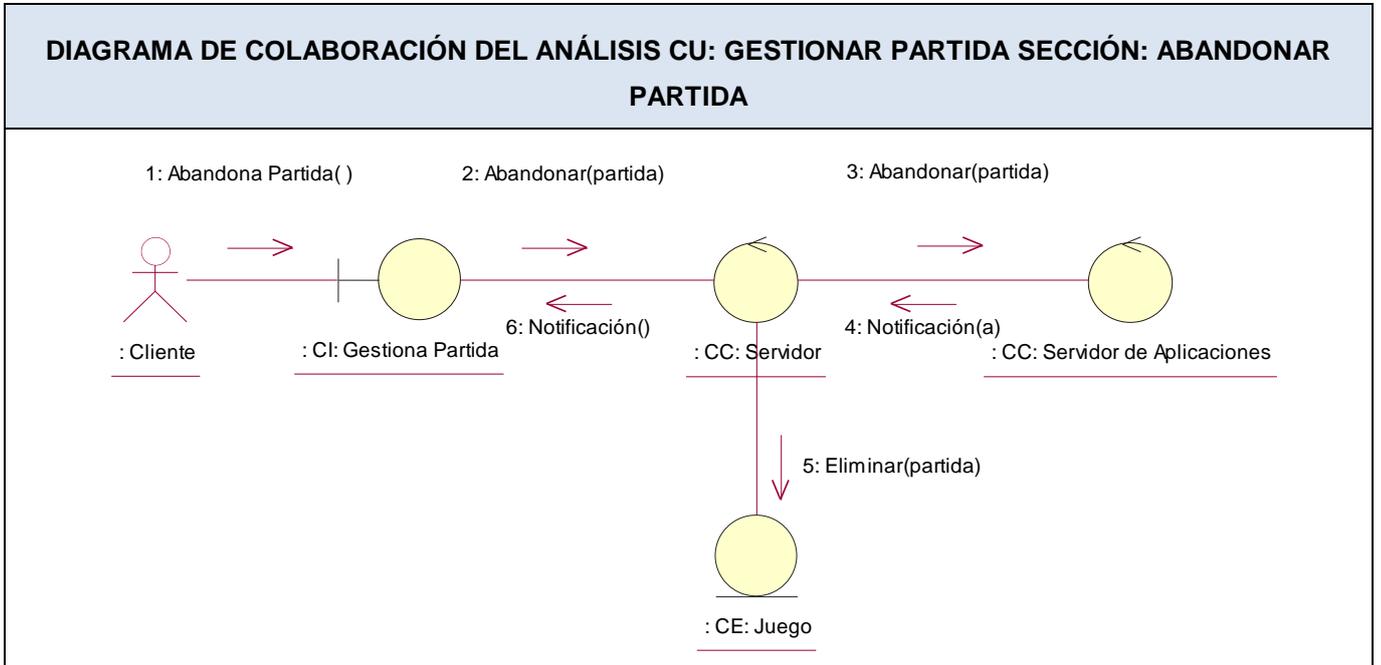


Tabla 35. Diagrama de Colaboración del Análisis del CU: Gestionar Partida Sección: Abandonar Partida.



Anexo 4: Diagramas de Clases del Diseño.

Tabla 36. Diagramas de Clases del Diseño del CU: Gestionar Partida Sección: Iniciar Partida.

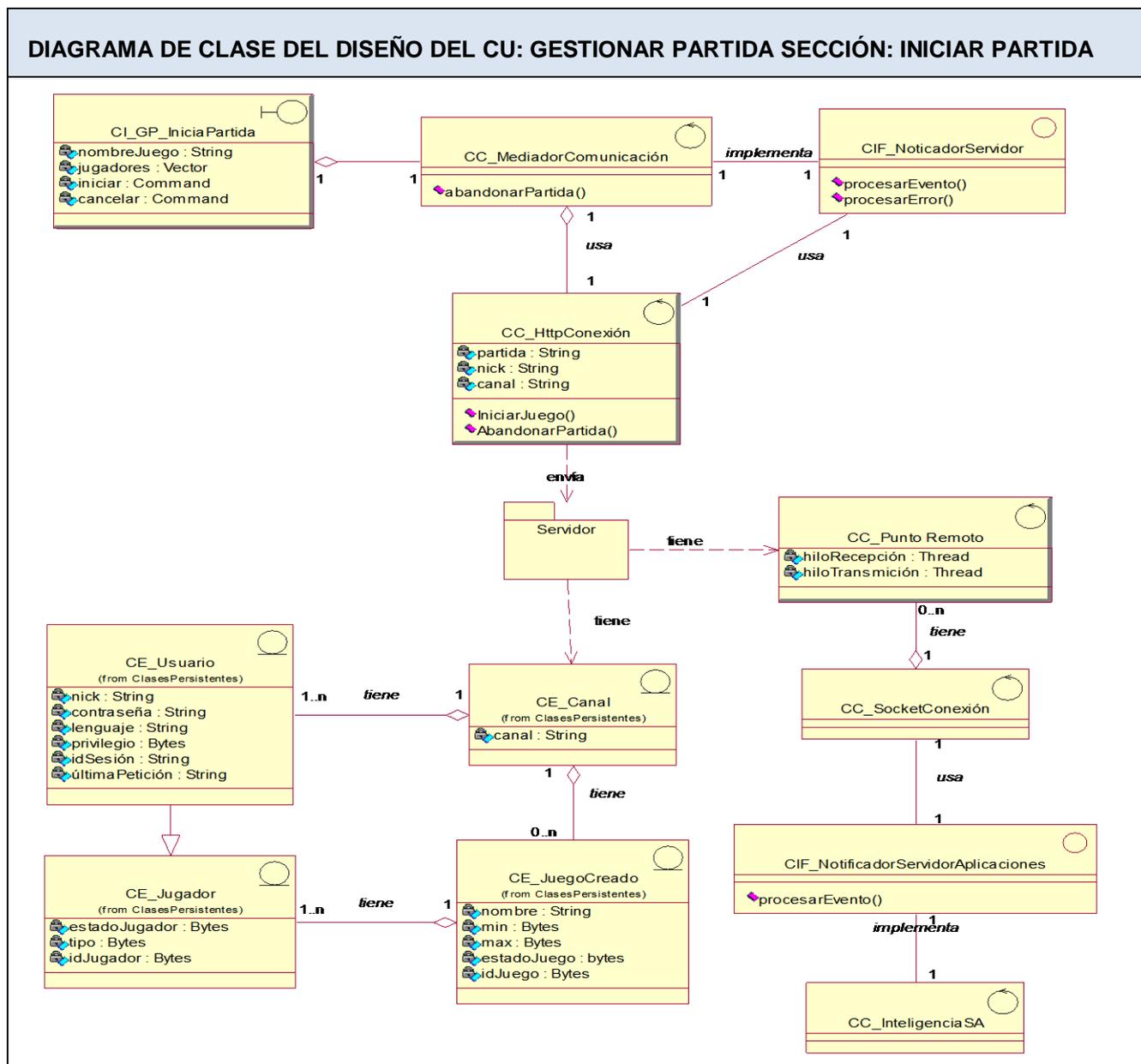
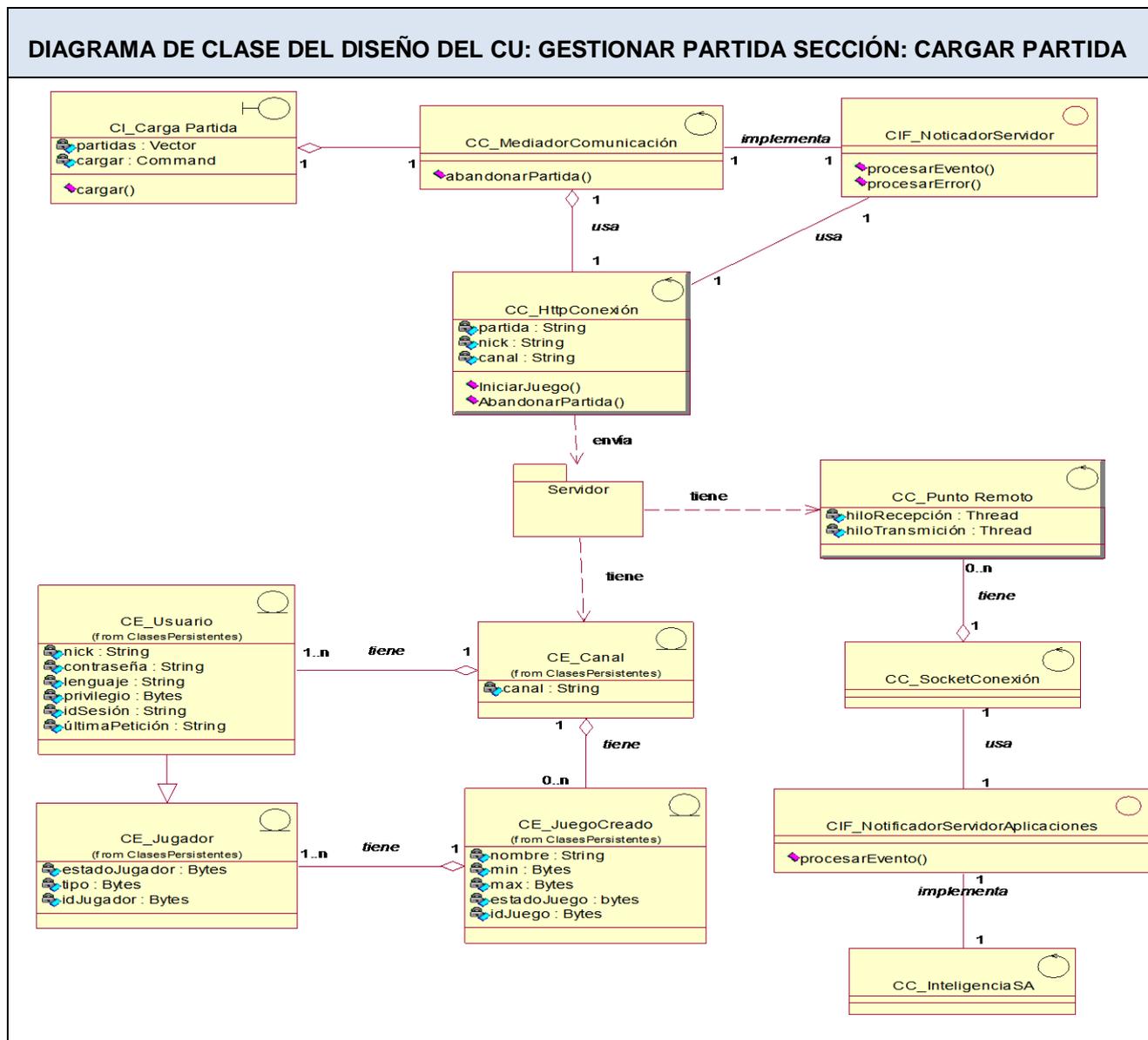


Tabla 37. Diagramas de Clases del Diseño del CU: Gestionar Partida Sección: Cargar Partida.



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.).

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.

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 para 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 usado 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.

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, utilizando 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. Permite dar servicios de alta calidad de voz y datos.

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

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 inalámbrica 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.