

2012

# Universidad de las Ciencias Informáticas Facultad 5



## **Título: Gestión de datos históricos en el perfil de usuario del videojuego Meteorix.**

**Autor:** Osmay Rafael Lojo Carnero

**Tutor:** MSc. Omar Correa Madrigal

**Co-tutor:** Ing. Adisleidys Mirabal Pérez

La Habana, junio 2012  
Año 54 de la Revolución



*"...Lo fundamental es hacer algo nuevo cada día y luego perfeccionar lo que se ha hecho el día anterior..."*  
*Ché*

## DECLARACIÓN DE AUTORÍA

---

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

---

**Autor:** Osmay Rafael Lojo Carnero

---

**Tutor:** MSc. Omar Correa Madrigal

---

**Co-Tutora:** Ing. Adisleidys Mirabal Pérez

### Generales del Tutor:

**Nombre y apellidos:** MSc. Omar Correa Madrigal.

**Especialidad:** Master en Informática Aplicada.

### Síntesis del tutor:

- Actual jefe del Proyecto de Investigación Juegos CNeuro.
- Categoría docente: Instructor.
- 7 años de experiencia.

**Correo electrónico:** ocorrea@uci.cu

### Generales de la Co-Tutora:

**Nombre y apellidos:** Ing. Adisleidys Mirabal Pérez

**Especialidad:** Ingeniería en Ciencias Informáticas.

### Síntesis de la co-tutora:

- Graduada en el año 2011.
- 4 años de experiencia.

**Correo electrónico:** amirabal@uci.cu

*Quiero agradecer a mis padres por ser tan especiales. A mi mamá, a mi papá por apoyarme todo el tiempo, por siempre estar dispuesto a ayudarme en todas las cosas que he necesitado. Aunque he tratado en muchas ocasiones de valerme por mi mismo para solucionar mis problemas siempre te he tenido presente y he sabido que puedo contar contigo para todo. A mi abuelita mami, por todo el trabajo que pasaste conmigo, por aguantarme todas mis malcriadeces, y ser tan preocupada con mis cosas. Mi mayor deseo es que ustedes estén orgullosos de mí.*

*Quiero también agradecer a mis hermanos Abelito, Rafelito y Rey, gracias por su cariño y apoyo, ustedes son los mejores del mundo. A mi hermana Nanita que a lo largo de la carrera ha hecho papel amiga y de madre.*

*A mi primo Pavel, gracias por tus consejos.*

*A mi familia en general.*

*A mi hermano Carlito por ser mi brazo derecho, mi apoyo, mi paño de lágrimas; por darme tanto de sí sin esperar nada a cambio.*

*A Yanet, que a pesar de haber llegado a mi vida en un momento muy difícil se ha convertido en una de las personas más importantes para mí, gracias por tu comprensión, paciencia y cariño.*

*A mis tutores Omar y Adis. Omar sé que no pudiste ayudarme como hubieras querido, pero aun así lo hiciste lo mejor que pudiste, espero que este trabajo esté a la altura de tus expectativas. Adis gracias por brindarme tu ayuda en todo momento y por guiarme en el desarrollo de la tesis.*

*A mis profes Marleodys y Yadira por regalarme además su amistad. A mi profesor Jorge Canales por ser quien me enseñó a pensar.*

*A todos ellos, y a mis amigos que no he mencionado por no cometer el error de olvidar alguno, les doy las más sinceras gracias.*

*Dedico mi trabajo de diploma a mi abuelo Fran, espero que donde esté se sienta orgulloso de mis logros tanto en el plano profesional, como en el plano de la vida en general.*

*A mi abuelita Margarita (mami), espero haber cumplido tu sueño de ser un hombre de bien.*

*A mis padres (Marilyn y Rafael), gracias por todo el amor y la confianza que me han dado todo este tiempo, espero haberlos correspondido con mi éxito en todos los sentidos.*

*A mis hermanos Rafelito, Abelito y Reynier (el negro), espero que estén orgullosos de mí.*

*A mi Familia y mis Amigos, ellos son para mí lo más importante del mundo.*

## **Resumen**

La Ambliopía es una enfermedad oftalmológica que posee una prevalencia entre el 3 y 4 % de los niños en edad preescolar y entre el 2 y 7 % de los niños en edad escolar. La forma de tratarse ha incorporado terapias no muy agradables y en ocasiones invasivas, pasando por varios tipos de métodos que van desde el uso de la corrección refractiva, penalización basada en fármacos o parche hasta la aplicación de las nuevas tecnologías de la informática y las telecomunicaciones. En Cuba la carencia de medios ha llevado a que el tratamiento de esta enfermedad se efectúe por la forma tradicional. Con el fin de mejorar esta clase de tratamientos y brindarle a los especialistas la posibilidad de hacer una terapia más humana y eficiente se desarrolló el videojuego Meteorix.

Actualmente el videojuego Meteorix no cuenta con un módulo que se encargue de gestionar los datos históricos en el perfil de usuario de forma local y distribuida. Dichos datos son definidos por los resultados obtenidos durante la terapia. Su posterior evaluación y análisis permitirá a los especialistas identificar el grado de avance del paciente o modificar, si es preciso, el tratamiento.

Para resolver esta problemática se desarrolló un módulo responsable de gestionar los datos históricos en el perfil de usuario de forma local y distribuida, garantizando su persistencia y seguridad. Para el desarrollo de la solución, fueron seleccionadas las herramientas tecnológicas necesarias, haciendo énfasis en el uso del software libre. También fue seleccionada la metodología de *software* a utilizar en el proceso de desarrollo.

### **Palabras claves:**

Ambliopía, videojuego, datos históricos, perfil de usuario, persistencia.

## Índice de contenidos

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA.....</b>	<b>5</b>
1.1 VIDEOJUEGOS .....	5
1.2 RESEÑA SOBRE VIDEOJUEGOS.....	6
1.3 JUEGOS SERIOS (SERIOUS GAMES).....	7
1.4 METEORIX .....	8
1.5 PERFIL DE USUARIO .....	10
1.6 GESTIÓN DEL PERFIL DE USUARIO .....	10
1.7 GESTIÓN DE LOS PERFILES DE USUARIOS EN DIFERENTES GÉNEROS DE VIDEOJUEGOS .....	11
1.7.1 <i>Videojuegos Shooters</i> .....	11
1.7.2 <i>Videojuegos de rol</i> .....	12
1.7.3 <i>Videojuegos Deportivos</i> .....	13
1.7.4 <i>Videojuegos de Carreras</i> .....	13
1.8 EL REGISTRO APLICADO A LOS VIDEOJUEGOS .....	14
1.9 PERSISTENCIA DE LOS DATOS.....	15
1.9.1 <i>Gestores de base de datos</i> .....	16
1.9.2 <i>Ficheros XML</i> .....	17
1.9.3 <i>Ficheros de texto</i> .....	18
1.9.4 <i>Ficheros binarios</i> .....	18
1.10 SEGURIDAD DE LOS DATOS.....	19
1.10.1 <i>Encriptación</i> .....	19
1.10.2 <i>Algoritmos de cifrado</i> .....	20
1.10.3 <i>Algoritmo de cifrado Cesar</i> .....	21
1.11 TRABAJOS RELACIONADOS EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS .....	22
1.12 METODOLOGÍA Y HERRAMIENTAS .....	23
1.12.1 <i>TinyXML</i> .....	23
1.12.2 <i>Metodología Programación Extrema (Extreme Programming, XP)</i> .....	24
1.12.3 <i>Lenguaje de modelado (UML)</i> .....	26
1.12.4 <i>Herramienta CASE (Visual Paradigm)</i> .....	26
1.12.7 <i>Librería libpq</i> .....	27
1.12.8 <i>Gestor de base de datos PostgreSQL</i> .....	27
<b>CAPÍTULO 2 DESARROLLO DE LA SOLUCIÓN .....</b>	<b>30</b>
2.1 SITUACIÓN ACTUAL .....	30
2.2 PROPUESTA DEL SISTEMA .....	30
2.3 SELECCIÓN DE LOS DATOS PERSISTENTES.....	31
2.4 SUBMÓDULO DE PERSISTENCIA DISTRIBUIDO .....	35
2.4.1 <i>Diagrama Entidad-Relación</i> .....	36
2.5 SUBMÓDULO DE PERSISTENCIA LOCAL.....	32
2.5.1 <i>Especificación de la estructura interna del fichero xml</i> .....	32
2.6 MOMENTOS DE SALVA.....	36
2.7 ANÁLISIS DEL DOMINIO .....	37



2.8	FASE DE EXPLORACIÓN.....	38
2.8.1	<i>Actores del sistema</i> .....	39
2.8.2	<i>Historias de usuario</i> .....	39
2.8.3	<i>Diseño de Casos de Pruebas</i> .....	42
2.9	PLANIFICACIÓN Y ENTREGA.....	42
2.9.1	<i>Plan de entrega</i> .....	43
2.10	DISEÑO DEL SISTEMA.....	44
2.10.1	<i>Arquitectura del sistema</i> .....	44
2.10.2	<i>Tarjetas CRC del sistema</i> .....	45
2.10.3	<i>Patrones de diseño</i> .....	48
2.10.4	<i>Estándares de codificación</i> .....	48
2.11	DESARROLLO DE ITERACIONES.....	52
2.11.1	<i>Implementación</i> .....	52
<b>CAPÍTULO 3 APLICACIÓN DE LA SOLUCIÓN.....</b>		<b>58</b>
	PRUEBAS DE ACEPTACIÓN.....	58
<b>CONCLUSIONES.....</b>		<b>62</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>		<b>64</b>
<b>GLOSARIO.....</b>		<b>67</b>

**Índice de tablas**

Tabla 1: Patients.....	35
Tabla 2: Sessions.....	35
Tabla 3: Levels.....	35
Tabla 4: Etiquetas.....	33
Tabla 5: Actores de sistema .....	39
Tabla 6: HU Crear perfil de usuario.....	40
Tabla 7: Cargar perfil de usuario .....	40
Tabla 8: Salvar sesión .....	40
Tabla 9: Salvar nivel .....	41
Tabla 10: Actualizar base de datos .....	41
Tabla 11: Plan de entrega.....	43
Tabla 12: Estimación del esfuerzo por historia de usuarios.....	43
Tabla 13: Tarjeta CRC para la clase CUser .....	45
Tabla 14: Tarjeta CRC para la clase CProfile.....	46
Tabla 15: Tarjeta CRC para la clase CProfileXml.....	46
Tabla 16: Tarjeta CRC para la clase CProfileDB .....	47
Tabla 17: HU planificadas para la primera iteración .....	53
Tabla 18: Diseñar la estructura del fichero xml.....	53
Tabla 19: Diseñar la base de datos .....	53
Tabla 20: Crear perfil de usuario .....	54
Tabla 21: Salvar perfil de usuario .....	54
Tabla 22: HU planificadas para la segunda iteración .....	54
Tabla 23: Salvar sesión .....	55
Tabla 24: Salvar nivel .....	55
Tabla 25: HU planificadas para la tercera iteración.....	55
Tabla 26: Actualizar la base de datos.....	56
Tabla 27: Caso de prueba Crear el perfil de usuario.....	58
Tabla 28: Caso de prueba Seleccionar perfil de usuario .....	59
Tabla 29: Caso de prueba Salvar sesión.....	59
Tabla 30: Caso de prueba Salvar nivel.....	60
Tabla 31: Actualizar base de datos .....	61

## Introducción

La Ambliopía es una enfermedad oftalmológica que posee una prevalencia entre el 3 y 4 % de los niños en edad preescolar y entre el 2 y 7 % de los niños en edad escolar. Clínicamente es uno de los padecimientos más frecuentes cuando se habla de discapacidades visuales de tipo funcional. La forma de tratarse ha incorporado terapias no muy agradables y en ocasiones invasivas, pasando por varios tipos de métodos que van desde el uso de la corrección refractiva, penalización basada en fármacos o parche hasta la aplicación de las nuevas tecnologías de la informática y las telecomunicaciones. A partir de que se diagnostica la ambliopía es importante valorar el hecho de que las mejoras se pueden hacer desde las edades preescolares ante la aparición del trastorno, estableciendo un margen de edad idóneo en el que resulta decisivo el aporte de la informática, específicamente en el campo de la Realidad Virtual que ha devenido como una alternativa interesante y flexible para su intervención. Su utilización ha ayudado a sensibilizar aún más la manera de realizar las terapias de rehabilitación [1].

Debido al vertiginoso crecimiento y desarrollo a nivel mundial de las nuevas tecnologías, en algunos países, durante las sesiones de terapia para el tratamiento de la ambliopía, se somete a los niños a ver una película 3D (en 3 dimensiones) y posteriormente a jugar una aplicación 3D interactiva. Dicho método ha arrojado excelentes resultados, superando los anteriores obtenidos mediante el método tradicional. Por otra parte la visión estereoscópica ha comenzado a incorporarse como otra alternativa para tratar la deficiencia visual. Tal es el caso de Fundación Visión COI, con la colaboración de Sony Computer Entertainment en España (SCEE) han comenzado desarrollar videojuegos estereoscópicos con vista a lograr un aumento de la agudeza visual con un consecuente aumento de la estereopsis [1].

En Cuba la carencia de medios ha llevado a que el tratamiento de esta enfermedad se efectúe por la forma tradicional de oclusión, mediante la cual el niño permanece más de 2 horas realizando ejercicios correctivos compensatorios, haciendo que el ojo afectado, no tapado, esté obligado a ejercitarse mediante diferentes tareas.

Con el fin de mejorar esta clase de tratamientos y brindarle a los especialistas la posibilidad de hacer una terapia más integrada y eficiente se desarrolló el videojuego Meteorix, el cual es fruto del proyecto

Investigación Juegos CNeuro, perteneciente al departamento de Visualización Gráfica y Realidad Virtual del Centro de Informática Industrial, Universidad de las Ciencias Informáticas, en colaboración con el Instituto Oftalmológico Pando Ferrer y el proyecto de Tecnología y discapacidad de la Universidad de Ciencias Pedagógicas “Enrique José Varona”.

Actualmente el videojuego Meteorix no posee un método para gestionar los datos históricos. Dichos datos son definidos por los resultados del uso de indicadores (variables de control como son: el Tiempo de Reacción, Nivel de Profundidad y Cantidad de Aciertos) que integran la actividad del sujeto durante en juego. Debido a que no existe una forma de registrar estos datos los especialistas no pueden realizar una evaluación del grado de avance del paciente ni de la eficiencia de la terapia. El videojuego también carece de un tratamiento de los perfiles de usuario, siendo estos un elemento fundamental en la identificación del paciente y su evolución.

Por lo anteriormente planteado surge el siguiente **problema científico**: ¿Cómo gestionar los datos históricos en el perfil de usuario del videojuego Meteorix?

Se hace entonces imprescindible analizar las técnicas existentes para gestionar datos históricos del perfil de usuario y garantizar la persistencia de la información así como la seguridad de la misma. Por tanto se establece como **objeto de estudio**: Gestión del perfil de usuario en los videojuegos.

Precisando el objeto de la investigación se define como **campo de acción**: Gestión de los datos históricos en el perfil de usuario del videojuego Meteorix.

Para dar solución al problema anteriormente expuesto se define como **objetivo general**: Desarrollar un módulo para la gestión de los datos históricos en el perfil de usuario en el videojuego Meteorix que garantice la persistencia y seguridad de los mismos.

Se derivan entonces los siguientes **objetivos específicos**:

- Desarrollar un submódulo que permita la gestión de los datos históricos de manera local.
- Desarrollar un submódulo que permita la gestión de los datos históricos de manera distribuida.

En consecuencia, la **idea a defender** con este trabajo es: El videojuego Meteorix, al contar con el módulo para la gestión de los datos históricos en el perfil de usuario, permitirá a los especialistas evaluar la evolución de los pacientes y la eficacia de la terapia.

Para dar cumplimiento a los objetivos se identifican las siguientes **tareas investigativas**:

- Caracterización de las técnicas existentes para la gestión del perfil de usuario en los videojuegos.
- Selección de la plataforma informática (metodologías, herramientas tecnológicas) para desarrollar el módulo que se encargue de la gestión de los datos históricos en el perfil de usuario del videojuego Meteorix.
- Elaboración del módulo de ingeniería de software como antesala al proceso de desarrollo del módulo para la gestión de datos históricos tanto local como distribuido.
- Validación del sistema mediante pruebas a los modos de persistencias desarrollados para garantizar el cumplimiento del objetivo general.

Durante el proceso investigativo se utilizan los métodos del nivel **teórico y empírico** los cuales se describen a continuación:

Nivel teórico: Se utilizan para acumular los conocimientos necesarios sobre el estado del arte del tratamiento de los datos históricos en los videojuegos.

Histórico-lógico: Se utilizan para la comprensión de los antecedentes y las tendencias actuales referidas a la evolución en el mundo de los mecanismos de gestión de datos históricos en los videojuegos.

Analítico-sintético: Se aplican con el fin de realizar el estudio de las concepciones y los conceptos referentes en la gestión de datos históricos en los videojuegos.

Nivel empírico: Posibilitan extraer de los procesos analizados las informaciones que se necesitan sobre ellos a través de observaciones y del uso de otras técnicas de recopilación de la información.

Como **aporte práctico** se espera obtener un módulo que se integre al videojuego Meteorix y que permita la gestión de los resultados del proceso terapéutico para el tratamiento de la ambliopía.

El presente documento consta de Resumen, Introducción y 3 capítulos que constituyen el cuerpo de la tesis, Conclusiones, Recomendaciones, Referencias bibliográficas, Glosario y Anexos. Los capítulos son:

## **Capítulo 1: Fundamentación Teórica**

En este capítulo se hace una investigación sobre el tratamiento de datos del perfil de usuario en los videojuegos así como sus principios y características. Se destacan las tendencias actuales sobre el tema a nivel mundial.

## **Capítulo 2: Desarrollo de la solución técnica**

En este capítulo se describe de forma conceptual las particularidades técnicas en el desarrollo del módulo y sus principales características. Así como también se describen los artefactos generados por la metodología en cada una de sus fases.

## **Capítulo 3: Pruebas funcionales**

En este capítulo se muestran las pruebas realizadas a la solución para garantizar el correcto funcionamiento de sus funcionalidades.

## Introducción

En este capítulo se hace una breve reseña sobre los videojuegos y algunas de sus clasificaciones. Se abordan algunos aspectos y características del proceso de desarrollo de un videojuego, específicamente la gestión de los datos históricos en el perfil de usuario. Se hace referencia además a las variantes empleadas para salvar información. También se realiza un breve análisis de las herramientas y tecnologías a utilizar en el desarrollo del módulo para la gestión de los datos históricos en el perfil de usuario del videojuego Meteorix. Estos aspectos contribuirán a proporcionar un mayor conocimiento sobre el tema de la gestión de los datos históricos en el perfil de usuario y posibilitarán elegir el camino más adecuado para resolver el problema de la presente investigación.

### 1.1 Videojuegos

Un videojuego es un tipo de juego con características técnicas concretas. Para poder entender el concepto de videojuego, primero se debe hacer un breve repaso la definición de juego en general.

La Real Academia Española define un juego como: "aquel ejercicio recreativo sometido a reglas, en el cual se gana o se pierde", aunque no en todos los juegos se gana o se pierde, sino que existen otras alternativas o metas.

“Reducido a su esencia formal un juego es una actividad entre dos o más jugadores independientes que buscan lograr sus objetivos en un contexto limitado. Una definición más convencional establecería que un juego es un concurso con reglas entre adversarios tratando de ganar estos objetivos.” [2]

El problema con esta definición es que no todos los juegos son concursos entre adversarios “en algunos juegos los jugadores cooperan para alcanzar una meta común en contra de alguna fuerza contraria o situación natural que en sí misma no es un jugador ya que carece de objetivo.” [2]

Concretar una definición universal de juego es algo difícil, depende, por tanto, del contexto en que se vaya a utilizar. En el mundo de la informática también se suele tener una apreciación algo similar del concepto de juego, solo que bajo otro término más específico, videojuego.

Un videojuego es un programa informático, creado en un principio para el entretenimiento [3], basado en la interacción entre una o varias personas y un dispositivo electrónico que ejecuta el videojuego. Pueden ejecutarse en equipos electrónicos de uso general (computadoras o los teléfonos celulares), en equipos de uso específico para este fin (videoconsolas). Los videojuegos recrean entornos y situaciones en los cuales el jugador puede controlar a uno o varios personajes (o cualquier otro elemento de dicho entorno), para conseguir uno o varios objetivos por medio de unas reglas determinadas.

## **1.2 Reseña sobre videojuegos**

Aproximadamente en 1947 tiene sus inicios la historia de los videojuegos, comenzó con una idea de la realización de un videojuego de misiles creada y patentada por Thomas T. Goldsmith Jr. y Estle Ray Mann. Algunos de los primeros videojuegos fueron el OXO (1952), este es un juego gráfico computarizado, derivado de la tesis doctoral de Alexander Sandy Douglas [3] y el Tennis For Two creado en 1958. Desde aquellos primeros intentos, los videojuegos han evolucionado considerablemente; si bien antes eran bastante sencillos y ordinarios en la actualidad esto ha sufrido grandes cambios debido al desarrollo de la tecnología en materia de hardware y por supuesto también a las técnicas y algoritmos implementados por los programadores. Aunque hoy en día cada uno de estos videojuegos tiene sus propias características, todos tienen un objetivo común, el entretenimiento del usuario. Para lograrlo combinan el uso de elementos de video, audio, retroalimentación táctil, basándose en tramas interesantes y que le planteen un desafío aceptable al jugador.

## **Géneros de videojuegos**

Tradicionalmente, no se ha utilizado ningún tipo de clasificación que abarque todos los posibles enfoques de géneros de videojuegos, sino que se han ido creando nuevos a medida que han ido apareciendo nuevos videojuegos que no entraban en las definiciones ya establecidas. Algunos de los criterios que normalmente se utilizan para clasificar los videojuegos son:



- Según el tipo de interacción que hay entre jugador y máquina.
- Según la representación gráfica del videojuego.
- Según las reglas que lo definen.
- Según su argumentación y discurso narrativo.

A medida que los videojuegos se han desarrollado, es más común que este pertenezca a más de un género y que cualquier clasificación que se haga de él deba considerarse como una aproximación, ya que la misma es muy difícil de mantener de manera coherente a medida que salen al mercado nuevos títulos con nuevos argumentos y formas de juego. Así mismo, las clasificaciones evolucionan a medida que salen nuevas propuestas, fusionando géneros y subgéneros que pueden estar presentes al mismo tiempo o no en un mismo videojuego.

### **1.3 Juegos Serios (Serious Games)**

El poder lúdico y de aprendizaje que tienen los juegos y, por extensión, los que se basan en las nuevas tecnologías, como los videojuegos, es uno de los principales temas de investigación por parte de los especialistas en la materia. De esta forma algunos investigadores han querido ir más allá y explorar el posible potencial de los juegos; no solo desde el punto de vista del entretenimiento sino desde otros enfoques como la educación, el entrenamiento y la información. Así, ha visto la luz una nueva generación de juegos, denominados, serious games.

Un juego serio es un videojuego diseñado de forma que junto al propio entretenimiento, transmite mensajes o ideas relevantes sobre diversos aspectos no relacionados con la industria de los videojuegos y que entran dentro del ámbito de la ética, como la política, la religión o la guerra, por poner unos ejemplos. El término «juego serio» ha existido desde mucho antes de la entrada en el mundo del entretenimiento de los dispositivos informáticos y electrónicos. En 1970, Clark Abt ya definió este término en su libro *Serious Games*, publicado por Viking Press. En este libro, Abt habla principalmente de los juegos de mesa y de los juegos de cartas, pero proporciona una definición general que puede aplicarse con facilidad a los juegos de la era informática.

“La más simple definición de videojuego serio es que estos juegos tienen un explícito y cuidadoso propósito educacional y no tienen como objetivo primario el entretenimiento, disfrute y diversión del sujeto que interactúa con él. No quiere decir que estos no sean divertidos ni entretenidos sino que estas características no son las más importantes.” [4]

### 1.4 Meteorix

Meteorix es un videojuego FPS (First Person Shooter) de corte galáctico que surge como fruto del proyecto Investigación Juegos CNeuro. Meteorix nombre con el cual se le bautizó, constituye un videojuego 3D, que fusiona su lógica con la de un tratamiento de manera que el paciente no se percata de que está siendo tratado, este equilibrio constituye uno de sus principales aportes. También es capaz de personalizar los contenidos del juego avanzando a mayores niveles de complejidad según sea el nivel de respuesta del niño (habilidades óculo manuales), lo cual es verificado por un sistema estadístico. Cuando al paciente le resulta difícil vencer un nivel, el juego le proporciona niveles de ayuda, los cuales son activados de forma automática y en orden jerárquico con la finalidad de atenuar la complejidad de las tareas. Meteorix posee 3 niveles de ayuda que son parte de la lógica del juego y su activación depende de la cantidad de energía que el jugador haya podido acumular. Estos niveles son:

Piloto automático: Orientación de la nave hacia los objetivos (el niño solo se concentra en la intercepción de los blancos).

Ralentizar tiempo: Hace que los objetos se muevan más lentos mientras el niño sigue con el mismo grado de movilidad.

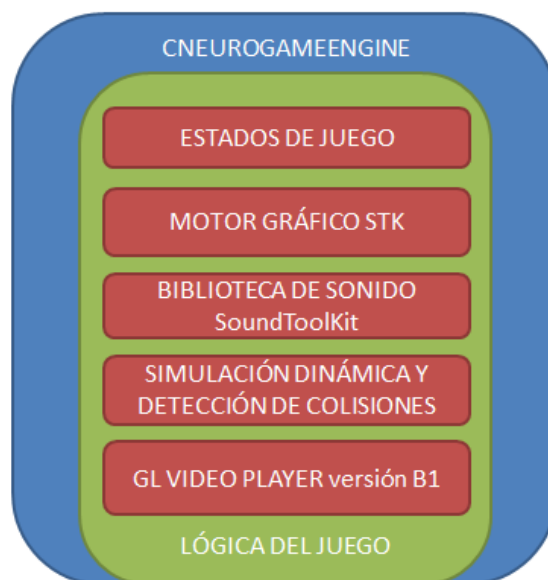
Resaltar tareas: Resaltar los objetivos gráficamente aumentando su tamaño.

Además se utilizan diferentes técnicas de proyección con vista a medir el nivel de impacto de las mismas en el tratamiento. En la *figura 1* se muestra una sesión de validación clínica [1].



**Figura 1: Técnicas de proyección**

El videojuego Meteorix está soportado por el motor de videojuegos **CNeuroGameEngine**, este fue creado con el propósito de realizar videojuegos que ayuden en la rehabilitación de algunas enfermedades [5]. En la *figura 2* se pueden apreciar los módulos que lo componen:



**Figura 2: Módulos del motor CNeuroGameEngine**

Meteorix además fue programado en lenguaje C++ utilizando como entorno de desarrollo integrado (IDE) Microsoft Visual Studio.Net 2003 debido a que la biblioteca para entornos virtuales SceneToolkit (STK) lo requiere para su funcionamiento.

## 1.5 Perfil de Usuario

Para mayor entendimiento del tema es necesario hacer referencia al concepto de perfil de usuario en una de sus acepciones. El término perfil se deriva de la psicología, dentro de esta disciplina se entiende como tal el "...conjunto de medidas diferentes de una persona o grupo, cada una de las cuales se expresa en la misma unidad de medición..." [6]

Esto significa que ciertas características de un individuo son medidas mediante pruebas que arrojan puntuaciones diferentes, estas puntuaciones constituyen su perfil, el cual es utilizado con fines diagnósticos. La presentación de las características del usuario debe ser sistematizada y consistente, a través de la precisión de su perfil.

## 1.6 Gestión del perfil de usuario

Durante la interacción del usuario con el videojuego y a medida que este va avanzando el jugador pone a prueba sus habilidades, las perfecciona y estas al mismo tiempo le permiten llegar a nuevos niveles. Entonces resulta muy útil y apreciada por los jugadores la opción de personalizar el juego a su gusto, y además, guardar los avances logrados al adentrarse en la trama del mismo. En los juegos de carros como el "Need for Speed Underground" por ejemplo, se emplea un sistema en el que el jugador debe competir en una serie de eventos para ganar dinero. A medida que avanza, podrá comprar nuevos autos o personalizar el que ya tiene para mejorar su rendimiento, así estará en condiciones de competir en circuitos de mayor dificultad. "Gran Turismo" constituye otro impresionante juego de carreras, contiene once circuitos, múltiples eventos contrarreloj y la modalidad de carreras de resistencia. El jugador tiene la posibilidad de usar varios modelos de autos. Comienza con un crédito con el que puede comprar coches nuevos o usados de las tiendas especializadas de fabricantes. A medida que avanza y gana eventos, el jugador obtiene más créditos, los que pueden ser usados en modificaciones para su auto actual o para comprar uno nuevo. El objetivo del juego "Burnout Paradise Ultimate Boxy" es obtener puntos mediante

choques espectaculares y saltos imposibles a toda velocidad, para poder seguir avanzando por la ciudad y adquirir nuevos coches por el camino. Dichos logros son guardados en registros correspondientes a cada **perfil de usuario**. Es una dinámica de juego sencilla pero muy divertida, por lo que centra la atención de muchos jugadores [7].

Precisamente la utilidad gestionar el perfil de usuario en un videojuego reside en brindarle al usuario la posibilidad de configurar el entorno a su gusto, editar las opciones que estime conveniente y de guardar los progresos alcanzados dentro del juego. Para ello puede crear su propio perfil, además de cargar o eliminar uno de los existentes. Estas características pueden contribuir a incrementar la aceptación y comercialización de un videojuego. Mientras más opciones de configuración se le brinden al usuario, más complacido estará.

### **1.7 Gestión de los perfiles de usuarios en diferentes géneros de videojuegos**

A continuación se hará una descripción de cómo se manejan estas funcionalidades en varios de los géneros de videojuegos existentes. La manera de salvar los datos de los perfiles es bastante similar en muchos de ellos, pero vale la pena tratarlos por separado, ya que las pequeñas diferencias que existen constituyen ventajas provechosas a tener en cuenta a la hora de decidir una variante para resolver el problema que se aborda en esta investigación.

#### **1.7.1 Videojuegos Shooters**

Son juegos con una temática muy sencilla, donde el jugador básicamente sólo puede moverse y disparar para cumplir su objetivo. Estos juegos también se llaman “shooters”, que es la traducción al inglés de pistoleros [3].

Dentro de este tipo de videojuegos se encuentra el “Medall of Honor” el cual es muy popular, cuenta básicamente con dos modos de juegos: multijugador, que permite la conexión en red de varios usuarios para formar equipos contrarios; y jugador individual, donde se juega contra otro usuario o en modo de misión, en el cual el jugador debe cumplir una serie de peligrosas tareas para llegar a obtener la victoria. En este último se realizan salvas cada vez que se avanza en el juego, se pasa una emboscada, se ejecuta

una explosión o se realiza una maniobra complicada, el usuario puede salvar esos logros haciendo uso de la tecla designada para este fin (F5) [8].

También se realizan salvos de información cuando el jugador configura los controles de teclado, nivel de dificultad del juego, nivel de detalle de los gráficos, el volumen de sonido o la sensibilidad del dispositivo de retroalimentación de fuerza. Además cuando se cumplen tareas claves de la misión, el mecanismo de salva automático del juego se activa y realiza esta operación. Todas las salvos de las misiones son grabadas en ficheros dentro de una carpeta del directorio del juego reservada para este fin, y dentro de ella una para cada perfil existente.

### 1.7.2 Videojuegos de rol

En este tipo de juego el jugador encarna uno o varios personajes con características propias de una personalidad, combinando una serie de características como son la fuerza, inteligencia, magia y ciertas habilidades [3].

El videojuego “World of WarCraft” de la compañía estadounidense “Blizzard Entertainment” forma parte de este género. Usa un servidor de base de datos (MySQL Admin) en el cual se encuentran todos los elementos que lo conforman, desde el total de vida de un simple personaje controlado por la computadora hasta la más complicada información de administración de permisos.

A medida que un jugador va adentrándose en las misiones, obteniendo elementos y subiendo de nivel, se le brinda la oportunidad de guardar sus progresos, de forma tal que pueda pausar el juego y posteriormente reiniciar donde se había quedado. Por otra parte, el juego presenta tolerancia a fallos, o sea, que en caso de existir un error en el servidor, o se interrumpa la electricidad y se cierre el juego bruscamente, el usuario al iniciar nuevamente la aplicación tiene salvado todos sus avances y logros.

La estrategia seguida por el administrador del servidor es programar el autoguardado del servidor cada cierto tiempo (no mayor de 3 minutos). Además dentro del juego se puede crear una macro que efectúa una salva cada vez que el usuario estime conveniente: a petición suya se envía un comando al servidor para que guarde todos los cambios hasta ese momento. La configuración de los controles de teclado, nivel

de detalle de los gráficos, volumen del ambiente del juego implica una salva de información en el servidor para garantizar estos cambios en próximos inicios de partida.

### 1.7.3 Videojuegos Deportivos

Este tipo de juegos se caracteriza por tener como núcleo argumental el tema deportivo. Se basa en deportes reales o ficticios, y se subdividen en simuladores o de menor realismo que un simulador [3].

Para explicar la gestión de los perfiles de usuarios en este género se toman de ejemplo los videojuegos de béisbol “MVP 08” y de fútbol “FIFA 08”. En ellos la estrategia de salva es similar. Al iniciar, el jugador escoge el nombre que tendrá su perfil, así como la liga y el equipo con que jugará. Luego puede hacer todo tipo de configuraciones en cuanto a la alineación del equipo, los controles, nivel de dificultad del juego, así como otros elementos relacionados con el nivel de detalle de los gráficos, el volumen del juego o la sensibilidad del dispositivo de retroalimentación de fuerza. A medida que el jugador obtiene resultados favorables, acumula puntos que le permiten desbloquear terrenos, trajes, trofeos o jugadores legendarios, a los cuales no tiene acceso inicialmente. El usuario tiene la posibilidad de salvar estos logros mediante uno de los botones del menú [9].

Casi todas las generaciones de juegos deportivos usan la estrategia de salvar los perfiles de usuario en ficheros que se generan en un directorio de fácil acceso, en esta localización se crea una carpeta con el nombre del juego, que dentro contiene otra por cada perfil con sus respectivos ficheros de configuración. Si se desea jugar en otra computadora y mantener los mismos avances, tan solo debe copiarse esta carpeta en el mismo directorio de la otra computadora.

### 1.7.4 Videojuegos de Carreras

El objetivo de este tipo de videojuegos es recrear una sensación de velocidad al jugador. Se pueden conducir todo tipo de vehículos, desde coches hasta aviones, pasando por lanchas motoras o naves espaciales. El objetivo del jugador es llegar el primero en el menor tiempo posible a la meta [3].

En el videojuego “Need for Speed Underground” que forma parte de este género, para realizar la salva de los perfiles de usuario se usa la misma idea que los videojuegos deportivos explicados anteriormente.

Cuando el usuario comienza a jugar debe elegir el nombre del perfil y el auto. En las diferentes modalidades de carreras que existen, el jugador va obteniendo dinero, con el cual puede desbloquear nuevas pistas, carros y piezas para mejorarlos [10].

También puede configurar los controles de teclado, nivel de dificultad del juego, nivel de detalle de los gráficos, el volumen del ambiente del juego o la sensibilidad del dispositivo de retroalimentación de fuerza.

Todos estos cambios son salvados en un fichero que se genera en la dirección que tiene predefinida el juego; un directorio de fácil acceso para el usuario.

## **1.8 El Registro aplicado a los videojuegos**

El término Registro se haya indisolublemente ligado a la gestión de la información, proceso de determinar la información que se precisa, recogerla, analizarla, registrarla, recuperarla, utilizarla y divulgarla. El registro es el complemento de la gestión de la información pues evita la pérdida de datos y permite que se realicen procedimientos con la información que en él se encuentra.

Según el Diccionario de la real academia cita textual registro es: Acción y efecto de registrar. Conjunto de datos relacionados entre sí, que constituyen una unidad de información en una base de datos.

El principal objetivo de los registros, es proporcionar información útil sobre la prevalencia, evolución, resultados y necesidades. Los registros deben cumplir una serie de características, tales como la validez, la exactitud, la confianza y la calidad. La utilidad de los registros depende básicamente de la calidad de los datos recogidos. Si los datos son de pobre calidad los resultados obtenidos con ellos pueden ser inciertos. Además, si la calidad de los datos no ha sido verificada, los resultados pueden ser de poca credibilidad, especialmente si se trata de datos que son inesperados. Se considera que un registro es de calidad, si proporcionan información válida tanto para una persona como para otra [11].

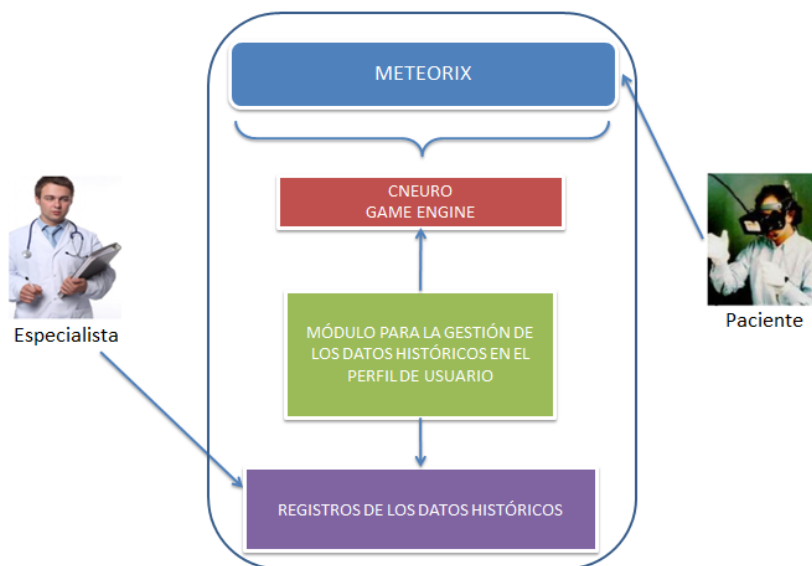
En resumen el registro es el conjunto de datos relacionados entre sí debido que toda la información que en él se encuentra están estrechamente relacionadas una con otra mediante la acción del usuario con el sistema. Es una pequeña unidad de almacenamiento destinada a contener cierto tipo de datos, teniendo



en cuenta que se trata como un área pequeña de almacenamiento de alta velocidad donde se guardan datos referentes a una persona y su desempeño en la realización de un examen.

El desarrollo de los videojuegos con fines terapéuticos aumenta constantemente, estos en su gran mayoría precisan de la utilización de los registros para llevar el control sobre la efectividad del tratamiento. Alguna de las ventajas del uso de las nuevas tecnologías de la información en este sentido se encuentra obtener registros más competentes y eficaces (los datos pueden ser introducidos continuamente sin limitaciones ni de tiempo, ni de lugar), aumentar la calidad de los mismos para permitir su inmediato procesamiento y chequeo de su consistencia y el permitir una más fácil diseminación de la información.

En la Figura 1.2 se observa el aporte de los registros en Meteorix.



**Figura 3: Utilización de los registros en Meteorix**

## 1.9 Persistencia de los datos

Persistencia es el término utilizado en informática para guardar y recuperar datos u objetos. La mayoría de las aplicaciones siguen una estructura sencilla de funcionamiento: recuperación de datos de una anterior ejecución, procesado y salvaguarda de los nuevos datos [12].

En muchas ocasiones se necesita que los datos manejados por una aplicación tengan un tiempo de vida que exceda el tiempo que dicha aplicación se mantiene en activo. Esto se conoce como persistencia de datos. Existen diversas formas de garantizar la persistencia en una aplicación informática que se analizan a continuación.

## 1.9.1 Gestores de base de datos

Un gestor de base de datos o sistema de gestión de base de datos es un software que permite introducir, organizar y recuperar la información de las bases de datos; en fin, administrarlas [13]. Permiten a los usuarios crear bases de datos y especificar su esquema por medio de algún lenguaje de definición, además de consultar los datos y modificarlos usando para ello un lenguaje de consulta y manipulación.

Una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y que ha sido diseñada para satisfacer los requerimientos de información. Son utilizadas en sistemas que requieren una interacción fluida con la aplicación que las usa. Además de los datos, también se almacena su descripción [11].

Existen videojuegos que usan las bases de datos con el fin de almacenar información para su funcionamiento y gestión, por ejemplo el videojuego “World of Warcraft” basa toda su estructura en las consultas que realiza sobre la base de datos que contiene información referente al entorno del videojuego y la información de cada jugador.

También el simulador web “Second Life” usa las facilidades de las bases de datos para almacenar todo el volumen de información que puede generar la creación de miles de personajes distintos, cada uno con diferentes características y modos de vida [14].

La ventaja del uso de base de datos en un videojuego tiene que ver directamente con el volumen de información a manejar, es recomendable usarlas cuando son muchos datos. Se emplean generalmente en videojuegos que fueron diseñados específicamente para aceptar muchos jugadores online a cualquier hora del día y en cualquier lugar del planeta, por lo que los volúmenes de datos generados son impresionantes.

### 1.9.2 Ficheros XML

XML, siglas en inglés de Extensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML (son las siglas de Standard Generalized Markup Language o "Estándar de Lenguaje de Marcado Generalizado". Consiste en un sistema para la organización y etiquetado de documentos) y permite definir la gramática de lenguajes específicos. Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Es considerado un método para introducir datos estructurados en un fichero de texto [15].

Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios, como estructurar, almacenar e intercambiar información. Consiste en varias reglas y convenciones definidas para conformar formatos de texto para almacenar datos, con el fin de crear ficheros que sean fáciles de generar y leer por una computadora. De esta manera se evitan problemas muy comunes como la falta de extensibilidad, soporte para la internacionalización y dependencia a una determinada plataforma.

Este tipo de fichero puede ser utilizado en los juegos web para la transferencia de información, ya que un cliente puede escribir en un documento los datos en texto plano que estaba manejando el formato XML sin necesidad de tener alguna aplicación en específico para este fin y trabajar con estos datos.

Algunos de los sistemas de metadatos que se han publicado en los últimos años han optado por el lenguaje XML como principal mecanismo de representación y codificación de los mismos. Las ventajas de usar XML frente a otras alternativas se encuentran en la facilidad de su intercambio y posterior procesamiento utilizando una única sintaxis común, y la opción de combinar e intercalar los metadatos dentro del texto completo de los documentos. Así, sería posible codificar un texto usando un conjunto de etiquetas o marcas relevantes para ese tipo de información y agregarle metadatos descriptivos o administrativos como parte de la cabecera de dicho documento, codificados en cualquier otro vocabulario siempre que éste también esté basado en la sintaxis XML.

## 1.9.3 Ficheros de texto

Los ficheros de texto son la manera que se utiliza en la programación informática para salvar los datos procesados por los programas. En los ficheros de texto la información se guarda como caracteres ASCII u otro estándar de codificación, lo que los hace eficaces a la hora de guardar textos y números pequeños.

Estos archivos están compuestos por bytes que representan caracteres ordinarios como letras, números y puntuación, también incluye algunos caracteres de control como tabulaciones, separaciones de líneas y fin de línea. Esta simplicidad permite que una gran variedad de programas pueda leer y editar ese contenido [16].

Es una alternativa que puede ser usada por cualquier videojuego para salvar información de la aplicación o bien de los jugadores que la usan. Solamente tendrían que plantearse algunas interrogantes acerca de la seguridad de los archivos, ya que los mismos exponen texto plano que es configurable con facilidad, además de que es entendible; razón que puede afectar el uso de ficheros de texto en caso de querer guardar información que sea necesario se mantenga oculta.

## 1.9.4 Ficheros binarios

Es un archivo que contiene información de cualquier tipo, codificada en forma binaria para el propósito de almacenamiento y procesamiento en computadoras. En el formato binario los números se guardan tal y cómo están representados en la memoria: los números enteros representan dos bytes, los reales cuatro bytes, etc [16]. El uso de ficheros binarios aumenta los niveles de confidencialidad de la información del archivo, dificultándose algún tipo de modificación efectiva fuera del entorno de modelación [17].

Las últimas dos características explicadas son aprovechadas por algunos videojuegos para salvar información que se necesita leer en el menor tiempo posible, que sea confidencial y difícil de cambiar (al menos manualmente no se conseguirán resultados coherentes) Por ejemplo, en el videojuego "MVP 08" el fichero de configuración de los perfiles de usuario está en formato binario, aumentando la confidencialidad de los datos de cada jugador y a su vez que el sistema pueda realizar una lectura rápida sin tener que hacer conversión de datos.

Habiendo analizado los objetivos y las principales formas de persistencia de los datos es posible seleccionar las más idóneas para utilizar en el presente trabajo.

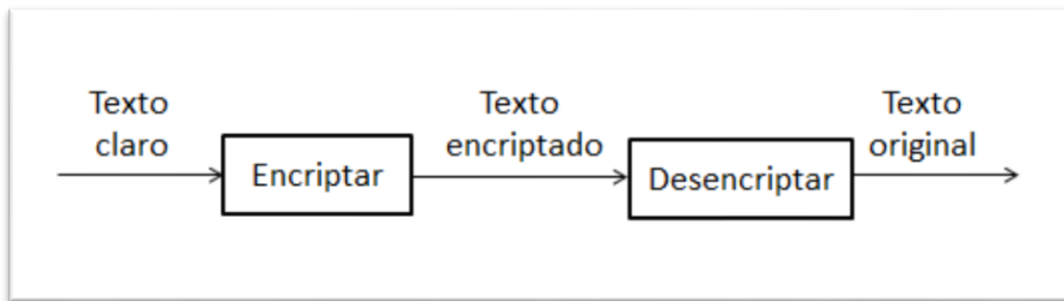
### **1.10 Seguridad de los datos**

En términos simples, la seguridad de los datos es la práctica de mantener los datos protegidos de la corrupción y el acceso no autorizado. Los datos son la forma compacta de la información almacenada en forma de columnas y filas en bases de datos, servidores de red y ordenadores personales. De aquí se deriva una amplia gama de información que abarca desde archivos personales y propiedad intelectual hasta datos de análisis de mercado y de alto secreto. Los datos podrían ser cualquier cosa de interés que puede ser leído o interpretado de otro modo en forma humana [18].

Un instrumento de seguridad de la información es el cifrado o encriptación de los datos.

#### **1.10.1 Encriptación**

Encriptación es el proceso mediante el cual cierta información o texto sin formato es cifrado de forma que el resultado sea ilegible a menos que se conozcan los datos necesarios para su interpretación. Es una medida de seguridad utilizada para que al momento de almacenar o transmitir información sensible ésta no pueda ser obtenida con facilidad por terceros. Opcionalmente puede existir además un proceso de desencriptación a través del cual la información puede ser interpretada de nuevo a su estado original, aunque existen métodos de encriptación que no pueden ser revertidos. El término encriptación es una traducción literal del inglés, la forma más correcta de utilizar este término sería cifrado.



**Figura 4: Flujo de cifrado**

Toda encriptación se encuentra basada en un Algoritmo, la función de este Algoritmo es básicamente codificar la información para que sea indescifrable a simple vista, de manera que una letra "A" pueda equivaler a "5x5mBwE" o bien a "xQE9fq", el trabajo del algoritmo es precisamente determinar cómo será transformada la información de su estado original a otro que sea muy difícil de descifrar. Una vez que la información arribe a su destino final, se aplica el algoritmo al contenido codificado "5x5mBwE" o bien a "xQE9fq" y resulta en la letra "A" o según sea el caso, en otra letra. Hoy en día los algoritmos de encriptación son ampliamente conocidos, es por esto que para prevenir a otro usuario "no autorizado" descifrar información encriptada, el algoritmo utiliza lo que es denominado llave ("key") para controlar la encriptación y desencriptación de la información.

## 1.10.2 Algoritmos de cifrado

**Cifradores Simétricos:** usa la misma llave al encriptar y desencriptar. Los cifradores simétricos también son llamados cifradores de llave privada o llave secreta.

**Cifradores Asimétricos (también llamados cifradores de llaves públicas):** involucran una llave pública, que puede ser libremente distribuida, y una llave privada, que puede ser secreta. Estas llaves siempre son generadas en parejas. Llaves públicas realmente son públicas, nadie puede violar su privacidad sin la llave privada.

A continuación se encuentran enunciados algunos algoritmos de cifrado:

- AES(Advanced Encryption Standard) es un nuevo estándar adoptado por el gobierno de U.S.A. que remplazara a DES/3DES. AES también es conocido como el algoritmo de Encryption Rijndael, el cual es muy veloz. El número después del nombre del algoritmo es el tamaño de cifrado que usa: aes128-cbc, aes192-cbc, aes256-cbc.
- DES (Data Encryption Standard) y 3DES (Triple Data Encryption Standard) fueron los primeros 2 algoritmos de cifrado disponibles. Creados por la NSA (Nacional Security Agency de USA), a principios de los de la década de los 70. Estos algoritmos han sido susceptibles a análisis criptográfico, por lo que no son muy recomendados si se tiene otros algoritmos disponibles.
- Blowfish es un algoritmo de cifrado en bloque con llave simétrica diseñado por Schneier que usa una llave de tamaño variable y un bloque de 64 bits. La llave puede ser de 32 a 448 bits. Los datos son pasados por la función de encriptación 16 veces.

La Ley Orgánica de Protección de Datos establece tres niveles para la seguridad de los ficheros que manipulan datos de diferentes tipos: alto, medio y básico. El nivel básico engloba toda la información referente a datos de salud, que se refieran exclusivamente al grado o condición de discapacidad o la simple declaración de invalidez, con motivo del cumplimiento de deberes públicos [19].

Debido al carácter terapéutico de Meteorix los datos a tratar en el presente trabajo se encuentran clasificados en el nivel básico de seguridad, es decir que no es necesario extremar las medidas a tomar para garantizar su seguridad. Se decide entonces utilizar el algoritmo de cifrado César ya que los algoritmos anteriormente expuestos poseen alta complejidad desde el punto de vista de implementación y tienen además un alto costo temporal en materia de funcionamiento. En cambio César posee una complejidad temporal de  $O(n)$ , lo que lo hace idóneo para sistemas de escasos recursos.

### 1.10.3 Algoritmo de cifrado Cesar

En criptografía, el cifrado César, también conocido como cifrado por desplazamiento, código de César o desplazamiento de César, es una de las técnicas de codificación más simples y más usadas. Es un tipo de cifrado por sustitución en el que una letra en el texto original es reemplazada por otra letra que se encuentra un número fijo de posiciones más adelante en el alfabeto. Por ejemplo, con un desplazamiento

de 3, la A sería sustituida por la D (situada 3 lugares a la derecha de la A ), la B sería reemplazada por la E, etc. El cifrado César muchas veces puede formar parte de sistemas más complejos de codificación, como el cifrado Vigenère [20].

### 1.11 Trabajos relacionados en la Universidad de las Ciencias Informáticas

Durante la investigación se analizaron trabajos realizados anteriormente relacionados con la gestión de los perfiles de usuario. Se pudo conocer sobre la existencia de dos tesis realizadas en la Universidad de las Ciencias Informáticas, una de ellas, aunque no está exactamente relacionada con los videojuegos, acumula algunas experiencias que pueden tomarse como punto de partida.

Se trata del “Módulo Registro de Evaluación para el Simulador Quirúrgico” realizada en el 2008, en la cual se realizó la investigación e implementación de un módulo para el registro de la evaluación del simulador, que permite brindar un reporte detallado de los resultados obtenidos por el médico cirujano en la realización de un examen. Las pruebas realizadas por los cirujanos en el simulador se guardan en una base de datos con toda la información correspondiente al ejercicio para mantenerla organizada. El simulador brinda además reportes de los resultados obtenidos por los médicos examinados.

Para el desarrollo de ese trabajo se empleó como medio de almacenamiento de la información el gestor de Base de Datos Firebird, encargado de todo el trabajo con el volumen de información registrado por los médicos que pasaron el examen. El módulo contiene datos personales de los médicos, brinda la posibilidad de llevar un control de los valores de todas las variables de cada prueba, permitiendo la gestión de la información de los resultados alcanzados por cada usuario. Para lograr la correcta automatización de todo el proceso de almacenamiento de datos fue necesario procesar los siguientes datos personales de los médicos cirujanos que realizan el ejercicio: nombre, apellidos, usuario, contraseña, dirección particular, teléfono, centro de trabajo, años de experiencia. Estos parámetros permiten a los profesores o especialistas encargados de la evaluación acceder a los reportes de toda la actividad relacionada con el ejercicio, facilitando el trabajo y la rapidez en la evaluación [11].

La otra tesis consultada sobre el tema es “Gestión de los perfiles de usuario en el videojuego de conducción de automóviles “Rápido y Curioso”. En ella se lleva a cabo una investigación sobre el tema y



se propone un método para gestionar dichos perfiles en el videojuego de conducción de automóviles “Rápido y Curioso”, que se desarrolló en la Facultad 5 de la Universidad de las Ciencias Informáticas. Se analizan diferentes estrategias empleadas para almacenar la información relativa a los jugadores y al juego en general. En la investigación se describe además la estructura del modelo de persistencia propuesto para realizar un módulo capaz de administrar el perfil de cada jugador del videojuego en cuestión.

El análisis de ambos trabajos permitió conocer algunas de las técnicas utilizadas para la gestión de los perfiles de usuario. Se valoraron las más convenientes a utilizar y los aspectos a tener en cuenta. También se profundizó en el aporte de la base de datos para el registro de la información, y su utilización en el almacenamiento de los datos históricos.

## 1.12 Metodología y Herramientas

La selección de herramientas tecnológicas y metodología que se muestra a continuación surge a partir del estudio y análisis de las mismas de acuerdo con sus características, de forma que satisfagan los objetivos y las necesidades en el desarrollo del módulo para la gestión de datos históricos del perfil de usuario del videojuego Meteorix.

### 1.12.1 TinyXML

Los programas encargados de leer e interpretar la entrada se conocen como analizadores sintácticos (o, en inglés, parser). Estos programas permiten navegar por el contenido del fichero en forma de árbol. Normalmente el **parser** se llama dentro de los constructores de los objetos, los cuales reciben como parámetro el fichero donde están contenidos los datos del objeto que queremos crear.

TinyXML es un pequeño parser XML fácilmente integrable en otros programas. TinyXML parsea un documento XML y construye un árbol DOM (Document Object Model) que puede ser recorrido, modificado y guardado en disco. Se puede también construir un árbol con objetos C++ o desde un string y escribirlo a disco como documento XML. TinyXML está diseñado para ser rápido y fácil de aprender. Soporta completamente UTF-8 (8-bit Unicode Transformation Format). Está desarrollado bajo licencia Zlib, que permite su uso en códigos open source y comercial [21].

## 1.12.2 Metodología Programación Extrema (Extreme Programming, XP)

La Programación Extrema (XP) es una metodología ligera utilizada para proyectos de corto plazo y equipos pequeños. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Se define como especialmente adecuada para proyectos con funcionalidades imprecisas y muy cambiantes. Mientras que la metodología RUP intenta reducir la complejidad del software por medio de estructura y la preparación de las tareas pendientes en función de los objetivos de la fase y actividad actual, XP, como toda metodología ágil, lo intenta por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción [22].

La base para el desarrollo del software que usa esta metodología son las historias de usuarios, documentos escritos por el cliente en las que describe los escenarios sobre el funcionamiento del sistema y que no sólo están limitados a la interfaz de usuario, sino que también pueden describir modelos o dominios. Estas historias de usuarios junto a la arquitectura que se persigue, sirve de base para crear un plan de entregas de software entre el equipo de desarrollo y el cliente, para cada una de las cuales se definen los objetivos y las iteraciones (generalmente cortas) necesarias para cumplirlo [22]. Las historias de usuarios y los casos de pruebas son la base sobre la que se asienta el trabajo del desarrollador [23].

XP impone un alto nivel de disciplina entre los programadores. Permite mantener un mínimo nivel de documentación, lo cual a su vez se traduce en una gran velocidad en el desarrollo (Hernán 2004)

A continuación se presenta las características esenciales de XP organizadas en los tres apartados siguientes: historias de usuario, roles y proceso.

### **Las Historias de Usuario**

Las historias de usuario son la técnica utilizada en XP para especificar las necesidades del software. Se trata de tarjetas en las cuales el cliente describe brevemente las características que el sistema debe poseer o sea los requisitos funcionales.

Respecto a la información contenida en la historia de usuario, existen varias plantillas sugeridas pero no existe un consenso al respecto. En muchos casos sólo se propone utilizar un nombre y una descripción. Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración [24].

## Roles de XP

Para una mejor organización de la ejecución de las actividades para el desarrollo de proyectos guiados por esta metodología se establecen los siguientes roles: Programador, Cliente, Encargado de pruebas, Encargado de seguimiento, Entrenador, Consultor, Gestor [24].

## Proceso XP

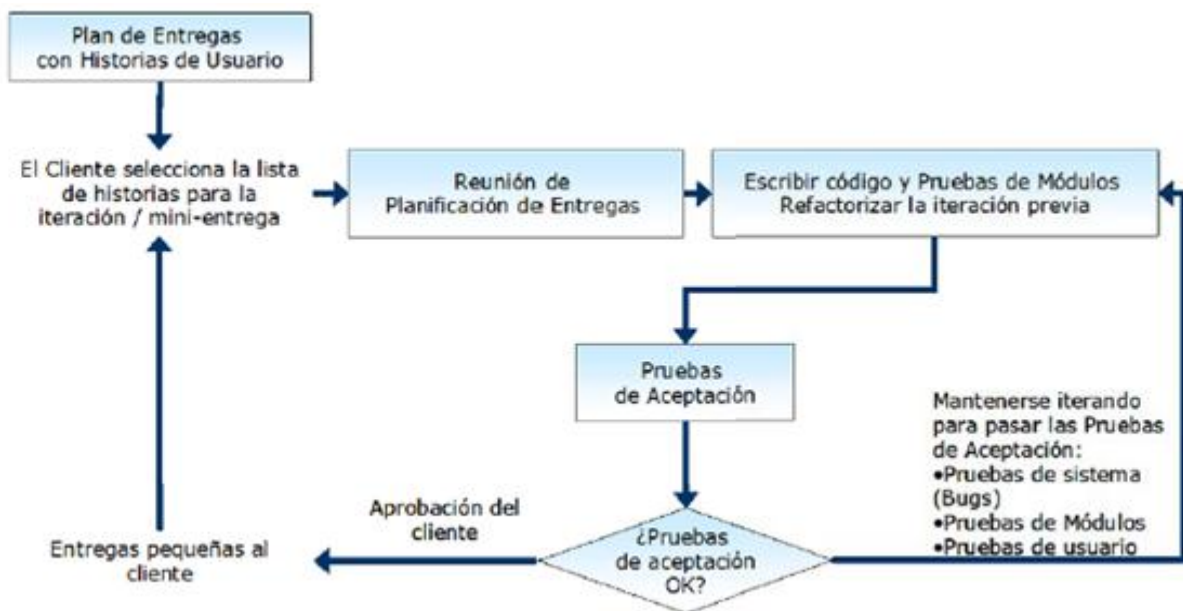


Figura 5: Proceso de XP

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1 [24].

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. Se debe presionar al programador a realizar más trabajo que el estimado, ya que, de no ser así, perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de 6 fases: Exploración, Planificación de entrega, Iteraciones, Producción, Mantenimiento, Muerte del Proyecto [24].

### **1.12.3 Lenguaje de modelado (UML)**

El Lenguaje de Modelado Unificado (UML - Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un producto de software que responde a un enfoque orientado a objetos. Este lenguaje fue creado por un grupo de estudiosos de la Ingeniería de Software formado por: Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1995. Desde entonces, se ha convertido en el estándar internacional para definir, organizar y visualizar los elementos que configuran la arquitectura de una aplicación orientada a objetos [25].

### **1.12.4 Herramienta CASE (Visual Paradigm)**

Como herramienta CASE (Computer Aided Software Engineering) se emplea Visual Paradigm para el trabajo con UML (VP-UML). La herramienta está diseñada para una amplia gama de usuarios que incluye a: ingenieros de software, analistas de sistemas, analistas de negocios y arquitectos de sistemas,

interesados en la creación de grandes sistemas de software de manera confiable, a través del paradigma orientado a objetos. VP-UML soporta los últimos estándares de anotaciones de JAVA y UML y provee soporte para la generación de código y la ingeniería inversa para Java. Las transiciones del análisis al diseño, y de éste a la implementación, están adecuadamente integradas dentro de la herramienta CASE, de manera que reduce significativamente los esfuerzos de todas las etapas del ciclo de desarrollo de software.

## 1.12.5 Librería libpq

La librería libpq es la interfaz del programador de la aplicación de C para PostgreSQL. Libpq es un conjunto de funciones de biblioteca que permiten a los programas cliente realizar consultas al servidor PostgreSQL y recibir los resultados de estas. Libpq es también el motor base de varias otras interfaces de aplicaciones de PostgreSQL, incluyendo las escritas en C + +, Perl y Python. Los programas cliente que utilicen libpq deben incluir el archivo de cabeceras libpq-fe.h y vincular con la biblioteca libpq [26].

## 1.12.6 Gestor de base de datos PostgreSQL

Luego de analizados los principales sistemas gestores de base de datos que se utilizan en la actualidad se selecciona PostgreSQL para darle solución al problema planteado. Ya que PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multiversión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario).

PostgreSQL se diseñó como una base de datos orientada a objetos. Esto significa, que las tablas no son tablas, sino objetos, y las tuplas son instancias de ese objeto. Mediante un sistema denominado MVCC (Acceso concurrente multiversión) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Esto evita que la transacción vea datos inconsistentes que pueden ser causados por la actualización de otra transacción concurrente en la misma fila de datos, proporcionando aislamiento transaccional para cada sesión de la base de datos.

PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles. Los bloqueos de tabla han sido sustituidos por el control de

conurrencia multiversión, el cual permite a los accesos de solo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite copias de seguridad en caliente desde pg\_dump mientras la base de datos permanece disponible para consultas.

Es apropiada para aplicaciones más complejas, en las cuáles se utilizan las normas ANSI-SQL, vistas (views), métodos (stored procedures) y disparadores (triggers) [27].

### **Características del PostgreSQL:**

- PostgreSQL es el servidor de bases de datos de código abierto más potente que existe y es por tanto la alternativa a MySQL cuando se necesitan características avanzadas como transacciones, procedimientos almacenados, triggers, vistas, etc.
- Es el servidor de bases de datos más utilizado por los programadores de servlets de Java y, en general, por todos aquellos que realizan aplicaciones cliente servidor complejas o críticas en el mundo Linux/Unix.
- Para aplicaciones Windows, PostgreSQL es una alternativa económica a SQL Server, pues su coste es menor y tiene similares prestaciones. Esta diferencia económica es especialmente sustancial si se necesita un Servidor Dedicado de bases de datos.
- Buena escalabilidad. Debido a su diseño, conforme se aumenta el número de procesadores aumenta el rendimiento.
- Soporta un subconjunto del estándar SQL mayor que el de MySQL y tiene propiedades de orientación a objetos.
- Incorpora una estructura de datos array (arreglos).
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

- En principio, tiene un sistema de bases de datos de mayor nivel a MySQL, a la altura de Oracle, Sybase.

### Conclusiones del Capítulo

En este capítulo se le dio cumplimiento a las dos primeras tareas investigativas trazadas en el presente trabajo, las cuales sirven de soporte teórico para la investigación. Se abordaron conceptos como: **perfil de usuario, registro y persistencia de los datos**, para lograr un mayor entendimiento del tema. Se analizaron estudios anteriores, a nivel mundial y en la Universidad sobre la gestión de los perfiles de usuario. Además se mencionaron las principales tendencias en la gestión del perfil de usuario en los videojuegos y fueron seleccionadas las herramientas y metodología con que se desarrollará la solución del problema.

## Introducción

En este capítulo, se define la forma en que se le da solución al problema. Se describen brevemente las principales funciones y el flujo actual de los procesos involucrados. Se muestran las historias de usuarios que fueron escritas por el cliente y se definen tres iteraciones por los programadores, elaborándose el Plan de Entrega para cada versión. Son establecidas las tareas generadas a partir del desarrollo de las historias de usuarios durante las tres iteraciones planificadas y se puntualizan los artefactos que dirigen la descripción del sistema.

### 2.1 Situación actual

Hoy en día el videojuego Meteorix no posee un módulo que se encargue de crear un registro de los datos históricos-clínicos de cada usuario-paciente durante el desarrollo del tratamiento. Estos resultados son requeridos posteriormente por el especialista encargado para evaluar la evolución del paciente. Tampoco existe la forma de organizar y relacionar los resultados (indicadores) obtenidos con el paciente correspondiente, de manera que resulta un poco engorroso el proceso de clasificación de la información por parte del personal médico, así como tampoco provee un indicador por el cual medir la eficacia de este método basado en las nuevas tecnologías.

En consecuencia es imprescindible la creación de un módulo que automatice el proceso de gestión de los datos históricos-clínicos obtenidos por el usuario-paciente. De esta forma al final de la sesión de terapia con Meteorix, quedaran registrados los valores necesarios para cuantificar del tratamiento.

### 2.2 Propuesta del sistema

El objetivo es desarrollar un módulo que se encargue de garantizar la persistencia y la seguridad de los datos históricos obtenidos durante la terapia con Meteorix. Este módulo estará constituido por dos submódulos: uno de ellos basado en ficheros xml, que permitirá gestionar el perfil de usuario de forma



local; y otro que gestionará el perfil de usuario de forma distribuida mediante el uso de bases de datos. En el caso de la variante distribuida la información se salvará en un servidor de base datos, lo que permitirá a los especialistas mantener el control centralizado de dicha información y además de garantizar su seguridad. Es decir que el perfil de usuario en el modo distribuido estará definido por la relación de tuplas de la base de datos. Por otra parte la variante local salvará los datos de la terapia en ficheros guardados localmente en un directorio designado para ello. Cada fichero constituye el perfil de usuario de un paciente. Este último modo de persistencia está incluido en la variante distribuida y surge como alternativa a los posibles problemas de conectividad, además para brindarle al sistema mayor extensibilidad.

### 2.3 Selección de los datos persistentes

Teniendo en cuenta que el *perfil* es una manera de identificación del usuario y que en este se guardan todas las características referentes a la interacción entre el usuario y el videojuego, se decide entonces crear un perfil que recoja toda la información resultante de la terapia. Dicho perfil tiene características diferentes a los analizados anteriormente en el capítulo 1 debido al tipo de información que contendrá. Normalmente esta información se compone por preferencias de configuración del sistema, datos referentes al progreso del tratamiento con el videojuego y características particulares del usuario. En el videojuego Meteorix, debido a su carácter terapéutico, no es objetivo realizar acción ninguna sobre ella. En su lugar es preciso almacenar los valores de las variables que permiten cuantificar la terapia. Según la lógica de Meteorix estas variables son fundamentales para que los especialistas realicen el diagnóstico de la efectividad del tratamiento con respecto a la evolución del paciente.

A continuación se exponen todas las variables que se tienen en cuenta en el presente trabajo:

- **Tiempo de software:** Es el tiempo total de la terapia, o sea desde el momento en que el usuario-paciente empieza a jugar hasta que termina.
- **Tareas:** Representa la cantidad total de tareas u objetivos (meteoritos o caracteres) que enfrenta en un nivel determinado.
- **Aciertos:** Es el número total de objetivos alcanzados satisfactoriamente (destruidos).

- **Promedio de errores por tarea:** Se refiere al promedio de errores (fallos) por cada tarea enfrentada.
- **Tiempo de reacción:** Tiempo que demora en alcanzar(destruir) una tarea.
- **Promedio de profundidad:** Se refiere al promedio de las profundidades de colisión de las tareas.

Todos estos indicadores son críticos en el proceso de evaluación que realizarán los especialistas luego de la terapia.

### 2.4 Submódulo de persistencia local

El submódulo de persistencia local salvará la información en formato xml. Para esto se utilizará TinyXml, el cual brinda muchas facilidades para el trabajo con ficheros xml. Para conformar y modificar la estructura del perfil se utilizarán las clases: TiXmlDocument, TiXmlElement, TiXmlHandler, TiXmlText y TiXmlDeclaration; así como algunas de sus funcionalidades, tales como: LoadFile, FirstChildElement, LinkEndChild, NextSiblingElement, SetAttribute, entre otras. Se decide utilizar dicho formato xml debido a las principales ventajas que posee:

- Es sencillo entender su estructura y procesarlo.
- Posee gran compatibilidad entre las aplicaciones.
- Es formal y conciso desde el punto de vista de los datos y la manera de almacenarlos.

#### 2.4.1 Especificación de la estructura interna del fichero xml

La estructura de un fichero que contiene la información del perfil de un usuario será la siguiente: cada etiqueta (tag) posee un nombre descriptivo del valor que almacena y dentro de ella dicho valor guardado. Los tipos de datos encontrados en el fichero son datos de tipo entero (int), datos de tipo cadena de caracteres (char \*) y datos de tipo real (float). El fichero en formato xml se encuentra formado por las siguientes etiquetas:

**Tabla 1: Etiquetas**

Etiquetas:	Descripción:
UserProfile	Dentro tiene toda la información del perfil
Name	Contiene los datos del paciente
Sessions	Contiene la cantidad de sesiones de terapia
Session	Contiene los datos de cada sesión
Level	Contiene los datos del nivel
SoftTime	Contiene el valor del tiempo de software utilizado en dicho nivel
Tasks	Contiene la cantidad de tareas enfrentadas durante el nivel
Hits	Contiene la cantidad de objetivo destruidos en el nivel
ErrorAverage	Contiene el promedio de errores en el nivel
ReactionTime	Contiene el tiempo de reacción en el nivel
DepthAverage	Contiene el promedio de profundidad en el nivel

En la Figura 2.1 se muestra la vista del fichero en formato xml, ya que después de encriptado no es entendible.

```
- <UserProfile>
  <Name id="80080926185" age="23" visiongrade="">Osmay Lojo Carnero</Name>
  - <Sessions total="3">
    - <Session number="1" day="24" month="5" year="2012" id="93645245201280080926185">
      - <Level type="1" id="93755245201280080926185">
        <SoftTime>27</SoftTime>
        <Tasks>10</Tasks>
        <Hits>10</Hits>
        <ErrorAverage>2.412342</ErrorAverage>
        <ReactionTime>2.188745</ReactionTime>
        <DepthAverage>4216.479492</DepthAverage>
      </Level>
      - <Level type="2" id="9388245201280080926185">
        <SoftTime>3</SoftTime>
        <Tasks>1</Tasks>
        <Hits>1</Hits>
        <ErrorAverage>0.000000</ErrorAverage>
        <ReactionTime>2.124023</ReactionTime>
        <DepthAverage>3900.000000</DepthAverage>
      </Level>
    </Session>
    - <Session number="2" day="24" month="5" year="2012" id="93954245201280080926185">
      - <Level type="1" id="94030245201280080926185">
        <SoftTime>23</SoftTime>
        <Tasks>7</Tasks>
        <Hits>6</Hits>
        <ErrorAverage>3.511295</ErrorAverage>
        <ReactionTime>1.555350</ReactionTime>
        <DepthAverage>3983.457031</DepthAverage>
      </Level>
    </Session>
  </Sessions>
</UserProfile>
```

Figura 6: Apariencia del fichero xml

### 2.4.2 Fichero “.mtx”

Luego de creado, cargado o modificado el perfil de usuario en formato xml será encriptado utilizando el algoritmo de cifrado César, seleccionado en el epígrafe 1.10.2. De esta forma se garantiza la seguridad de la información almacenada localmente. El fichero encriptado tendrá extensión “.mtx”.

### 2.5 Submódulo de persistencia distribuido

Este submódulo se encargará de gestionar los datos históricos en el servidor PostgreSQL, que registrará la información referente a los perfiles de usuario. De esta forma se garantiza el almacenamiento centralizado de la información y su seguridad. En el diseño de la base de datos se definen las siguientes tablas:

**Tabla 2: Patients**

Nombre:	Patients	
Descripción:	En esta tabla se almacenan los datos correspondientes al paciente	
Atributo	Tipo	Descripción
name	Varchar(255)	Nombre del paciente
dni	Varchar(11)	Carnet de identidad del paciente
age	Int4	Edad del paciente

**Tabla 3: Sessions**

Nombre:	Sessions	
Descripción:	En esta tabla se almacenan los datos correspondientes a la sesión	
Atributo	Tipo	Descripción
sessionid	Varchar(255)	Identificador de la sesión
date	Date	Fecha en que se realiza la sesión
dni	Varchar(11)	Llave foránea de la tabla. Referencia a la tabla Patients

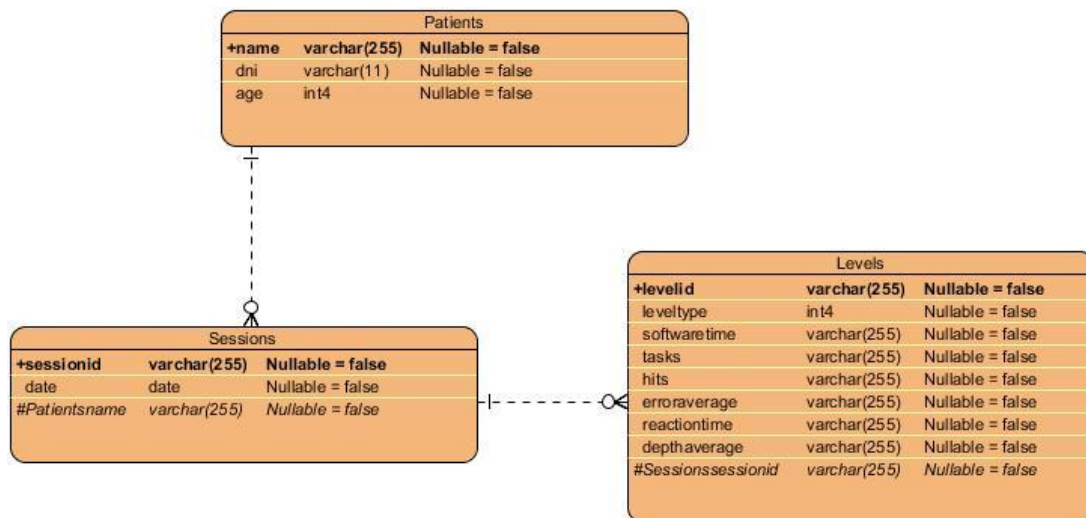
**Tabla 4: Levels**

Nombre:	Levels	
Descripción:	En esta tabla se almacenan los datos asociados a un nivel	
Atributo	Tipo	Descripción
levelid	Varchar(255)	Identificador del nivel
leveltype	Int4	Tipo de nivel
softwaretime	Varchar(11)	Tiempo general del nivel
tasks	Varchar(255)	Cantidad de objetivos

hits	Varchar(255)	Cantidad de aciertos
erroraverage	Varchar(255)	Promedio de errores por tarea
depthaverage	Varchar(255)	Promedio de profundidad
sessionid	Varchar(255)	Llave foránea de la tabla. Referencia a la tabla Sessions

### 2.5.1 Diagrama Entidad-Relación

En la siguiente figura se muestran las entidades, sus atributos y relaciones dentro del contexto de la solución.



**Figura 7: Diagrama Entidad-Relación**

### 2.6 Momentos de salva

Durante la terapia el perfil de usuario será modificado constantemente. Cuando el paciente comienza a jugar se agrega una nueva sesión al perfil con los valores correspondientes a la fecha, id y número de la sesión. Luego de comenzado el juego a medida que este se desarrolla se agregan los valores obtenidos en los niveles. Estos valores se actualizarán en tres momentos principales: cuando ocurre el ajuste de profundidad, cuando todos los objetivos de un nivel sean vencidos, o sea cuando se pase el nivel, y automáticamente cada 3 minutos. Se deciden estos momentos específicos ya que en el primer caso es

fundamental registrar los datos en el momento en que ocurre el ajuste de profundidad pues esto significa el cumplimiento de una de las tareas del nivel. Por otro lado 3 minutos representa el 10% del tiempo total de la terapia y al cabo de este tiempo es necesario efectuar otra salva.

### 2.7 Análisis del dominio

Aunque la metodología XP no define el modelo de dominio, se decide incluirlo para un mayor entendimiento del negocio, ya que su objetivo es detallar y describir el problema de manera entendible a los desarrolladores y ofrecer un vocabulario común para poder entender el contexto en que se trabaja en el sistema. Seguidamente se identificarán todos los conceptos asociados con el problema:

#### Conceptos del modelo de dominio

**Paciente:** Usuario del sistema, es el que juega el videojuego.

**Meteorix:** Videojuego, es el sistema en sí.

**Gestión de Datos Históricos en el perfil de usuario:** Módulo encargado de gestionar los datos históricos.

**Especialista:** Encargado de realizar la evaluación de la terapia.

**Perfil:** Ficheros o tuplas de una base de datos que contienen los valores obtenidos por el paciente en la terapia.

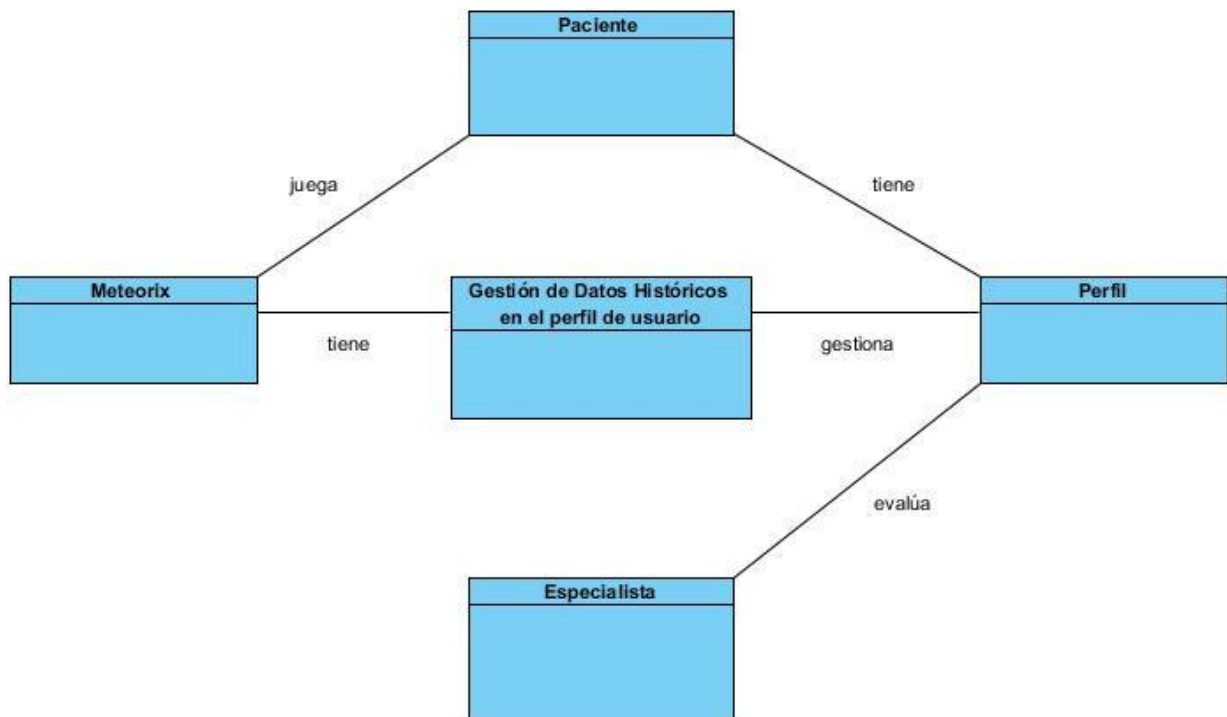


Figura 8: Modelo de dominio

### 2.8 Requerimientos no funcionales

Los requerimientos no funcionales son las propiedades o cualidades que el producto debe tener, imponen restricciones en el diseño o la implementación. Deben pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Por tanto son los criterios que se deben juzgar para determinar el funcionamiento de un sistema, verificando cómo este debería ser [28].

#### Software

*Para el cliente:* Sistema operativo Windows XP o superior.

*Para el servidor:* Sistema operativo Windows o superior. Un servidor de bases de datos PostgreSQL.

#### Hardware

*Para el cliente:* Se requiere un mínimo de 512 MB de RAM y 1.6 GHz de velocidad de procesamiento.

*Para el servidor:* Se requiere un mínimo de 1 GB de RAM y 1.6 GHz de velocidad de procesamiento.



### 2.9 Fase de exploración

Los objetivos de XP son muy simples: la satisfacción del cliente. Esta metodología trata de dar al cliente el software que él necesita y cuando lo necesita. El segundo objetivo es potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software.

Según propone XP, en esta fase los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto [24].

Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La extensión de la fase de exploración se sugiere que sea de 2 ó 3 semanas hasta pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

Para el desarrollo de la herramienta que se desea implementar solo se utilizarán 2 semanas, debido a que el mismo no es un sistema de gran extensión y el equipo de desarrollo tiene dominio del trabajo con las herramientas que se utilizarán.

#### 2.9.1 Actores del sistema

Tabla 5: Actores de sistema

Actores	Descripción
Paciente	Interacciona con el videojuego
Sistema	Es el encargado de gestionar los datos históricos en el perfil de usuario

#### 2.9.2 Historias de usuario

En XP la gestión de requerimientos del sistema es extremadamente simple, el cliente describe y prioriza sus necesidades mediante historias de usuario (HU). Son descripciones cortas y escritas sin terminología técnica.

## CAPÍTULO 2: DESARROLLO DE LA SOLUCIÓN

---

**Tabla 6: HU Crear perfil de usuario**

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Crear perfil de usuario
Actor: Paciente	Iteración Asignada: 1
Prioridad de Negocio: Alta	Puntos Estimados: 1
Nivel de Complejidad: Media	Puntos Reales: 1
Descripción: El paciente introduce su información personal. Los datos asociados son: nombre, apellidos, carnet de identidad y edad y da clic en el botón aceptar	
Observaciones: Se crea el perfil del paciente con la información correspondiente	

**Tabla 7: Cargar perfil de usuario**

Historia de Usuario	
Número: 2	Nombre Historia de Usuario: Cargar perfil de usuario
Actor: Paciente	Iteración Asignada: 1
Prioridad de Negocio: Alta	Puntos Estimados: 1
Nivel de Complejidad: Media	Puntos Reales: 1
Descripción: Se muestra una lista de perfiles existentes y el paciente selecciona el que le corresponde dando clic en su nombre	
Observaciones: Se carga el perfil correspondiente al nombre seleccionado	

**Tabla 8: Salvar sesión**

Historia de Usuario	
Número: 3	Nombre Historia de Usuario: Salvar sesión
Actor: Sistema	Iteración Asignada: 2
Prioridad de Negocio: Alta	Puntos Estimados: 1
Nivel de Complejidad: Media	Puntos Reales: 1

## CAPÍTULO 2: DESARROLLO DE LA SOLUCIÓN

---

Descripción: El paciente selecciona la opción “jugar” y se registra una nueva sesión en su perfil. Los datos asociados son número de sesión, id de sesión y fecha siguiendo la estructura día, mes y año
Observaciones:

**Tabla 9: Salvar nivel**

Historia de Usuario	
Número: 4	Nombre Historia de Usuario: Salvar nivel
Actor: Sistema	Iteración Asignada: 2
Prioridad de Negocio: Alta	Puntos Estimados: 2
Nivel de Complejidad: Alta	Puntos Reales: 2
Descripción: Cuando el paciente termina de jugar un nivel los valores obtenidos automáticamente son salvados en su perfil. Los datos asociados son número del nivel, id de nivel, tiempo de software, cantidad de tareas, promedio de aciertos, promedio de errores por tarea, tiempo de reacción, promedio de profundidad	
Observaciones:	

**Tabla 10: Actualizar base de datos**

Historia de Usuario	
Número: 5	Nombre Historia de Usuario: Actualizar Base de Datos
Actor: Sistema	Iteración Asignada: 3
Prioridad de Negocio: Alta	Puntos Estimados: 1
Nivel de Complejidad: Alta	Puntos Reales: 1
Descripción: El sistema actualiza automáticamente la base de datos con los perfiles que no aparecen registrados	
Observaciones:	

### 2.9.3 Diseño de Casos de Pruebas

A diferencia las metodologías tradicionales, donde la fase de pruebas, incluyendo la definición de las mismas, es realizada al final de cada iteración, o sobre el final del desarrollo de cada módulo; la metodología XP propone un modelo inverso, en el que, lo primero que se escribe son las pruebas que el sistema debe pasar [29].

#### Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las historias de usuarios definidas por el cliente. Las pruebas de aceptación son consideradas como pruebas de **caja negra**. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos [29].

Las pruebas de aceptación son creadas a partir de las HU (Historias de Usuario). Durante una iteración la HU seleccionada en la planificación de iteraciones se convertirá en una prueba de aceptación. El cliente o usuario especifica los aspectos a testear cuando una HU ha sido correctamente implementada. Una HU puede tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento [30]. Una historia de usuario no se puede considerar terminada hasta que no pase correctamente todas las pruebas de aceptación.

Para la realización de las pruebas al módulo se diseñaron 4 casos de pruebas donde se ejecutan paso a paso cada una de las posibles entradas al sistema y se evaluaron el resultado de las mismas.

Las tablas con los diseños de casos de pruebas se encuentran en el epígrafe 3.1 del capítulo 3 donde además se le incluyen los resultados de la ejecución de las mismas.

### 2.10 Planificación y entrega

La planificación es una fase corta en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario y asociadas a éstas, las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación. El resultado de esta fase es un Plan de Entregas [29].

## CAPÍTULO 2: DESARROLLO DE LA SOLUCIÓN

En esta fase el cliente establece la prioridad de cada HU y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debe obtenerse en no más de tres meses.

### 2.10.1 Plan de entrega

El Plan de Entregas se hace con la intención de que los programadores obtengan una estimación de dichas historias en cuanto al nivel de detalle, o sea, para fijar el período de tiempo que se puede tardar en la implementación de cada una.

**Tabla 11: Plan de entrega**

Historias de usuario	Iteración 1 (7/2/2012)	Iteración 2 (21/2/2012)	Iteración 3 (21/3/2012)
Crear Perfil Cargar Perfil	2	Terminado	Terminado
Salvar Sesión Salvar Nivel	No empezado	3	Terminado
Actualizar Base de Datos	No empezado	No empezado	1

**Tabla 12: Estimación del esfuerzo por historia de usuarios**

Historias de usuario	Tiempo estimado (semana ideal de trabajo)	Iteración asignada	Tiempo real
Crear Perfil Cargar Perfil	2	1	2
Salvar Sesión Salvar Nivel	3	2	3
Actualizar Base de Datos	1	3	1

### 2.11 Diseño del sistema

La metodología XP no requiere la descripción del sistema por medio de diagramas de clase utilizando notación UML, sino que se guía por técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración). Esto no implica que no se utilicen los diagramas para obtener una mejor visión y comunicación entre el equipo de trabajo, siempre y cuando su complejidad no sea alta y defina información importante.

#### 2.11.1 Arquitectura del sistema

Como se explicó anteriormente en el epígrafe 1.4 Meteorix está soportado por el motor **CNeuroGameEngine**, este posee una arquitectura en 3 capas:

- *Capa de Presentación*: Está formada por los formularios y los controles que se encuentran en los formularios. Es la capa con la que interactúa el usuario.
- *Capa de Negocio*: Está formada por las entidades, que representan objetos que van a ser manejados o utilizados por toda la aplicación.
- *Capa de Soporte*: Contiene las librerías para la física, los gráficos, el sonido, y el almacenamiento de los datos.



Figura 9: Arquitectura del motor CNeuroGameEngine

### 2.11.2 Tarjetas CRC del sistema

Las tarjetas CRC se utilizan para estructurar las clases y a su vez definir las responsabilidades sobre las mismas, así como la simulación de escenarios en el sistema.

El sistema a desarrollar dispone de diversos objetos, por lo que se definen las siguientes clases:

CUser, CProfile, CProfileXML, CProfileDB.

Tabla 13: Tarjeta CRC para la clase CUser

CUser	
Descripción: Encargada de almacenar los datos personales del paciente	
Atributos:	
Nombre	Descripción
m_acName	Contiene el nombre del paciente
m_acId	Contiene el carnet de identidad del paciente

m_acAge	Contiene la edad del paciente
<b>Responsabilidades:</b>	
Nombre	Colaborador
acName()	
acId()	
acAge()	

**Tabla 14: Tarjeta CRC para la clase CProfile**

<b>CProfile</b>	
Descripción: Clase genérica que posee los atributos y métodos comunes de CProfileXml y CProfileDB	
<b>Responsabilidades:</b>	
Nombre	Colaborador
LoadProfile()	
CreateProfile()	CUser
SaveSession()	CUser
SaveLevel()	CUser
UpdateLevel()	CUser
CesarEncrypt()	
kCesarDecrypt()	
acIdLevel()	CUser

**Tabla 15: Tarjeta CRC para la clase CProfileXml**

<b>CProfileXml</b>	
Descripción: Especialización de la clase CProfile encargada de gestionar el perfil de usuario local	
<b>Atributos:</b>	
Nombre	Descripción
m_pkdocument	Contiene el fichero xml
m_pkTagLevel	Contiene los datos de un nivel



Responsabilidades:	
Nombre	Colaborador
LoadProfile()	
CreateProfile()	CUser
SaveSession ()	CUser
SaveLevel ()	CUser
UpdateLevel()	CUser

**Tabla 16: Tarjeta CRC para la clase CProfileDB**

CProfileDB	
Descripción: Especialización de la clase CProfile encargada de gestionar el perfil local	
Atributos:	
Nombre	Descripción
m_acIdSession	Contiene el identificador de la sesión
m_pkProfileXml	Es un objeto de la clase CProfileXml
m_acTemporalLevel	Contiene el identificador del nivel
Responsabilidades:	
Nombre	Colaborador
LoadProfile()	CProfileXml
CreateProfile()	CUser, CProfileXml
SaveSession()	CUser, CProfileXml
SaveLevel()	CUser, CProfileXml
UpdateLevel()	CUser
AddPatient()	CUser
AddSessionTuple()	CUser
AddLevelTuple()	CUser
pkConnectDB()	
CloseConn()	
UpdateDB()	CUser

TagSessionToTuple()	CUser
TagLevelToTuple()	CUser
blsThisPatientInDB()	
blsThisSessionInDB()	

### 2.11.3 Patrones de diseño

Los Patrones son soluciones comunes a problemas de diseño de software orientado a objetos y que además poseen ciertas características de efectividad para resolver ese problema. Son reusables ya que pueden ser aplicados en otros diseños o problemas [31].

En el presente trabajo se aplican los patrones GRASP (General Responsibility Assignment Software Patterns) que son patrones generales de software para asignación de responsabilidades.

**Experto:** Indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo.

**Alta cohesión:** Establece que la información que almacena una clase debe de ser coherente y debe estar relacionada con la misma.

**Bajo acoplamiento:** Se refiere a tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

Los patrones analizados se aplican en la clase CProfileXml que es la encargada de realizar todas las operaciones con el perfil local y CProfileDB es la responsable de manejar toda la información referente a la conexión a la base de datos y las consultas.

### 2.11.4 Estándares de codificación

El código de la herramienta sigue algunos estándares propuestos por el grupo de desarrollo (respetando los estándares de codificación para C++ (uso de espacios y líneas en blanco, etc.)). Está programado en

inglés. El conocimiento de los estándares seguidos para el desarrollo de la misma permitirá un mayor entendimiento del código, y es una exigencia de los autores de la misma que cualquier módulo que se añada debe estar codificado siguiendo estos estándares.

### **Definiciones generales**

Las definiciones se realizan en inglés de manera descriptiva, evitando las abreviaturas y los nombres cortos. Se utiliza este idioma debido a que las palabras son simples, no se acentúan y está muy difundido en el mundo informático.

### **Clases:**

Las clases comienzan con la letra “C” en mayúscula al inicio de la palabra y en caso de estar conformada por palabras compuestas, la definición debe ser continua y cada palabra debe iniciar con mayúscula siguiendo el estilo determinado.

Ejemplos:

*ej:* class **C**Profile;

### **Declaración de variables:**

Los nombres de las variables comienzan con un identificador del tipo de dato al que correspondan, como se muestra a continuación. En el caso de que sean variables miembros de una clase, se le antepone el identificador “m\_” (en minúscula), si son globales se les antepone el identificador “g\_”, y en caso de ser argumentos de algún método, se les antepone el prefijo “arg\_”

### **Tipos simples:**

*ej:*

bool **b**flag;

float **f**Result;

```
char cName;  
  
char* acName; // arreglo de caracteres  
  
char* pcName; // puntero a un char  
  
char** aacName; // bidimensional  
  
char** apcName; // arreglo de punteros  
  
bool m_bMemberVarName; // variable miembro de una estructura o clase  
  
char g_cGlobalVarName; // variable global  
  
short sName;  
  
void* pvName;
```

### Instancias de tipos creados:

*ej:*

```
CClassName kObjectName; // objeto de una clase  
  
CClassName* pkName; // puntero a objeto  
  
CClassName* akName; // arreglo de objetos  
  
CClassName* m_akName; // variable miembro de clase
```

### Métodos:

En el caso de los métodos, se les antepone el identificador del tipo de dato de devolución, y en caso de no tenerlo (void), no se les antepone nada. Los constructores y destructores, como lo exigen los compiladores, llevan el nombre de la clase.

### Constructor y destructor:

*ej:*

```
CClassName (bool arg_bVarName, float& arg_fVarName);
```

```
~CClassName ();
```

### Funciones:

*ej:*

```
bool bFunction1 (...);
```

```
int* piFunction2 (...);
```

```
CClassName* pkFunction3 (...);
```

### Procedimientos:

*ej:*

```
void Procedure4 (...);
```

### Métodos de acceso a miembros:

Los métodos de acceso a los miembros de las clases no se nombrarán “Gets” ni “Sets”, sino como los demás métodos, pero **con el nombre** de la variable a la que se accede y sin el prefijo “m\_”:

*ej:* Para la siguiente variable, los métodos de acceso serían:

```
int m_iMyVar; //variable
```

```
int iMyVar(); // R
```

```
void MyVar(int arg_iMyVar) // W
```

```
int& iMyVar(); // R/W
```

### 2.12 Desarrollo de Iteraciones

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta etapa, generando al final de cada una, un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo. Las iteraciones son también utilizadas para medir el progreso del proyecto.

En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor del negocio). Al final de la última iteración el sistema estará listo para entrar en producción [30].

#### 2.12.1 Implementación

Las historias de usuario agrupadas en cada iteración se van implementando durante el transcurso de la iteración a la cual pertenecen. Al principio de cada iteración se lleva a cabo una revisión del plan de iteraciones y se modifica en caso de ser necesario. Como parte de este plan se descomponen las HU en tareas de desarrollo, asignando a un grupo o una persona como responsable de su implementación. Estas tareas son para el uso estricto de los programadores por lo que pueden ser escritas en lenguaje técnico y no necesariamente entendible por el cliente.

A continuación quedan detalladas las tareas de desarrollo realizadas en cada una de las iteraciones.

#### Iteración 1

## CAPÍTULO 2: DESARROLLO DE LA SOLUCIÓN

La primera iteración tendrá como objetivo darle cumplimiento a las HU 1,2 que representan la base de la estructura del negocio. Las mismas son críticas para el proyecto, pues las demás funcionalidades dependen totalmente de estas.

**Tabla 17: HU planificadas para la primera iteración**

Historia de usuario	Tiempo estimado (Semanas)	Tiempo real (Semanas)
Crear Perfil	1	1
Cargar Perfil	1	1

### Tareas de las historias de usuarios desarrolladas en la primera iteración.

Luego de relacionar las HU pertenecientes a esta iteración, se procede a la especificación de las principales tareas de desarrollo que se realizaron para cumplir el propósito de la misma.

**Tabla 18: Diseñar la estructura del fichero xml**

Tarea	
Número de la tarea: 1	Número de HU: 1,2
Nombre de la tarea: Diseñar la estructura del fichero xml	
Tipo de tarea: Diseño	Puntos estimados: 0.1
Programador(es) responsable(s): Osmay Rafael Lojo Carnero	
Descripción: Esta tarea está referida al diseño de la estructura interna del fichero xml.	

**Tabla 19: Diseñar la base de datos**

Tarea	
Número de la tarea: 2	Número de HU:
Nombre de la tarea: Diseñar la base de datos	
Tipo de tarea: Diseño	Puntos estimados: 0.1
Programador(es) responsable(s): Osmay Rafael Lojo Carnero	

Descripción: Esta actividad engloba todas las acciones que intervienen en el diseño de la base de datos, crear las tablas y sus relaciones para darle solución al problema

**Tabla 20: Crear perfil de usuario**

Tarea	
Número de la tarea: 3	Número de HU: 2
Nombre de la tarea: Crear perfil de usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.9
Programador(es) responsable(s): Osmay Rafael Lojo Carnero	
Descripción: Con esta tarea se realiza la creación del perfil en los modos de persistencia local y distribuido	

**Tabla 21: Salvar perfil de usuario**

Tarea	
Número de la tarea: 4	Número de HU: 2
Nombre de la tarea: Salvar perfil de usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.9
Programador(es) responsable(s): Osmay Rafael Lojo Carnero	
Descripción: Esta tarea se refiere a la salva del perfil en los modos de persistencia local y distribuido	

### Iteración 2

La segunda iteración está centrada en desarrollar la HU 3, 4 además de que incluyen una mayor complejidad de desarrollo. Estas HU son las responsables de registrar los datos históricos en el perfil de usuario.

**Tabla 22: HU planificadas para la segunda iteración**

Historia de usuario	Tiempo estimado (Semanas)	Tiempo real (Semanas)
Salvar sesión	1	1



Salvar Nivel	2	12
--------------	---	----

### Tareas de las historias de usuarios desarrolladas en la segunda iteración

Luego de relacionar las HU pertenecientes a esta iteración, se procede a la especificación de las principales tareas de desarrollo que se realizaron para cumplir el propósito de la misma.

**Tabla 23: Salvar sesión**

Tarea	
Número de la tarea: 5	Número de HU: 3,4
Nombre de la tarea: Salvar sesión	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Programador(es) responsable(s): Osmay Rafael Lojo Carnero	
Descripción: El desarrollo de esta tarea tiene como objetivo registrar los datos referentes a la sesión en los modos de persistencia definidos	

**Tabla 24: Salvar nivel**

Tarea	
Número de la tarea: 6	Número de HU: 3,4
Nombre de la tarea: Salvar nivel	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Programador(es) responsable(s): Osmay Rafael Lojo Carnero	
Descripción: Esta tarea es desarrollada con el objetivo de registrar los datos referentes al nivel en los modos de persistencia definidos	

### Iteración 3

La tercera iteración está centrada en desarrollar la HU 5 que posee alta prioridad en el negocio.

**Tabla 25: HU planificadas para la tercera iteración**

Historia de usuario	Tiempo estimado (Semanas)	Tiempo real (Semanas)

## CAPÍTULO 2: DESARROLLO DE LA SOLUCIÓN

---

Actualizar la base de datos	1	1
-----------------------------	---	---

### Tareas de las historias de usuarios desarrolladas en la tercera iteración

En la tercera iteración se definen las tareas para darle solución a la HU 5.

**Tabla 26: Actualizar la base de datos**

Tarea	
Número de la tarea: 5	Número de HU: 3,4
Nombre de la tarea: Actualizar la base de datos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Programador(es) responsable(s): Osmay Rafael Lojo Carnero	
Descripción: Esta tarea se refiere la actualización de la base de datos con los perfiles locales	

Las iteraciones de desarrollo sobre el sistema, permitieron que al finalizar se obtuviera un producto con todas las restricciones y características requeridas. Es importante señalar que el desarrollo de la solución se dividió en 3 iteraciones para ordenar la implementación, realizándose pequeñas entregas al final de cada una de ellas y recibir la aceptación del mismo.

### Conclusiones del capítulo

Luego de explicada la situación actual del videojuego Meteorix, se elaboró una propuesta del sistema. Se explicaron las particularidades técnicas de la misma. Se especificaron los usuarios que estarán relacionados con su utilización y funcionamiento. Se realizó una descripción de las HU, estableciendo la prioridad de cada una según el cliente, definiendo así el orden en el que serán implementadas las mismas. Se cuenta con 5 HU que serán implementadas en tres iteraciones. Quedó elaborado el modelo necesario para llevar a cabo la implementación del sistema mediante la descripción de los estándares de codificación y la arquitectura utilizada. Se plantearon y describieron las tareas de desarrollo para dar cumplimiento a las funcionalidades abordadas por las HU y se diseñaron las pruebas de aceptación que

## CAPÍTULO 2: DESARROLLO DE LA SOLUCIÓN

---

determinan la confianza y seguridad de las funcionalidades del sistema para el cliente, la descripción de estas pruebas en conjunto con los resultados se encuentran en el siguiente capítulo.

## Introducción

En el presente capítulo se documentan los resultados de la ejecución de las pruebas previamente diseñadas para probar las funcionalidades descritas. Además se definen las actividades necesarias para poner en funcionamiento la herramienta mediante el despliegue de la misma.

### 3.1 Pruebas de aceptación

Mediante las pruebas de aceptación se determina que el sistema cumple con lo deseado y se obtiene la conformidad del cliente. Los resultados de las mismas se encuentran a continuación.

**Tabla 27: Caso de prueba Crear el perfil de usuario**

Caso de prueba de Aceptación	
<b>Número:</b> 1	<b>Historia de Usuario:</b> 1
<b>Nombre:</b> Crear el perfil de usuario	
<b>Descripción:</b> El caso de prueba permite al paciente la posibilidad de crearse un perfil de usuario nuevo	
<b>Condiciones de ejecución:</b> El perfil no debe existir	
<b>Entradas/Pasos de ejecución:</b> -El paciente escoge la opción "Cambiar Jugador" -Luego el paciente introduce sus datos personales en los campos Nombre, 1er Apellido, 2do Apellido, Carnet y Edad en el formulario. -El paciente da clic en el botón "Aceptar" -Se crea un nuevo perfil y muestra la pantalla del "Menú principal"	
<b>Resultado esperado:</b>	

<p>Modo local: Se crea el fichero xml correspondiente al perfil en el directorio “Profiles”</p> <p>Modo distribuido: Se inserta el nuevo paciente en la tabla “Patients” de la base de datos con los datos correspondientes</p>
<b>Evaluación de la prueba:</b> Satisfactorio

**Tabla 28: Caso de prueba Seleccionar perfil de usuario**

<b>Caso de prueba de Aceptación</b>	
<b>Número:</b> 2	<b>Historia de Usuario:</b> 2
<b>Nombre:</b> Seleccionar perfil de usuario	
<b>Descripción:</b> El caso de prueba permite al paciente seleccionar su perfil	
<b>Condiciones de ejecución:</b> El perfil debe estar previamente creado	
<b>Entradas/Pasos de ejecución:</b> -El paciente escoge la opción “Cambiar Jugador” -Luego el paciente selecciona su perfil de la lista de perfiles seleccionados haciendo clic sobre su nombre -El sistema carga el perfil del paciente y muestra la pantalla del “Menú principal”	
<b>Resultado esperado:</b> Modo local: Se carga el fichero xml perteneciente al perfil seleccionado Modo distribuido: Se verifica que dicho perfil no se encuentre activo al mismo tiempo desde otra fuente, en caso afirmativo no permite cargar el perfil inhabilitando la función de la selección de perfil	
<b>Evaluación de la prueba:</b> Satisfactorio	

**Tabla 29: Caso de prueba Salvar sesión**

<b>Caso de prueba de Aceptación</b>	
<b>Número:</b> 3	<b>Historia de Usuario:</b> 3

<b>Nombre:</b> Salvar sesión
<b>Descripción:</b> El caso de prueba agrega una nueva sesión al perfil de usuario seleccionado
<b>Condiciones de ejecución:</b> El perfil debe haber sido o seleccionado o creado
<b>Entradas/Pasos de ejecución:</b> -El paciente escoge la opción “Jugar” en el menú principal -Se agrega una nueva sesión al perfil del usuario
<b>Resultado esperado:</b> Modo local: Se agrega al fichero xml una nueva etiqueta con los datos de la nueva sesión Modo distribuido: Se inserta la nueva sesión en la tabla “Sessions” de la base de datos con los datos correspondientes
<b>Evaluación de la prueba:</b> Satisfactorio

**Tabla 30: Caso de prueba Salvar nivel**

Caso de prueba de Aceptación	
<b>Número:</b> 4	<b>Historia de Usuario:</b> 4
<b>Nombre:</b> Salvar nivel	
<b>Descripción:</b> El caso de prueba agrega un nivel nuevo al perfil del usuario	
<b>Condiciones de ejecución:</b> El paciente debe terminar el nivel	
<b>Entradas/Pasos de ejecución:</b> -Luego de terminado el nivel el sistema automáticamente agrega un nuevo nivel al perfil de usuario	
<b>Resultado esperado:</b> Modo local: Se agrega al fichero xml una nueva etiqueta con los datos del nivel con los datos correspondientes en la sesión actual	

<p>Modo distribuido: El sistema inserta un nivel en la tabla “Levels” de la base de datos con los datos correspondientes</p>
<p><b>Evaluación de la prueba:</b> Satisfactorio</p>

**Tabla 31: Actualizar base de datos**

<b>Caso de prueba de Aceptación</b>	
<b>Número:</b> 4	<b>Historia de Usuario:</b> 5
<b>Nombre:</b> Actualizar Base de datos	
<b>Descripción:</b> El caso de prueba actualiza la base de datos	
<b>Condiciones de ejecución:</b>	
<b>Entradas/Pasos de ejecución:</b> El paciente ejecuta el juego	
<b>Resultado esperado:</b> Modo distribuido: Se verifica que todos los perfiles locales estén debidamente registrados en la base de datos, si no lo están los registra, además verifica que estén actualizados, sino los actualiza	
<b>Evaluación de la prueba:</b> Satisfactorio	

### **Conclusiones del capítulo**

Se ejecutaron las pruebas de aceptación que permitieron demostrar que el sistema está terminado, que desarrolla puntualmente las necesidades de la organización y que es aceptado por los usuarios finales.

### Conclusiones

Con motivo de dar cumplimiento al objetivo general de este trabajo en correspondencia con las exigencias del proyecto Investigación Juegos CNeuro, se realizó un estudio de diferentes géneros de videojuegos para analizar las técnicas utilizadas en la gestión de los perfiles de usuario y su posible aporte en el desarrollo de la solución.

Como la mayoría de los videojuegos pertenecen a empresas privadas, a los responsables no les reporta ningún beneficio compartir información referente al tema de perfiles de usuario, por lo que la materia existente es escasa, por tal motivo, este trabajo servirá de apoyo a todas aquellas personas interesadas en el tema pues se obtuvo una bibliografía actualizada que será de gran ayuda.

Se analizaron las herramientas y tecnologías utilizadas para llevar a cabo el proceso de desarrollo del módulo de gestión de datos históricos en el perfil de usuario. Se identificaron los elementos de los que se debía garantizar su persistencia y seguridad en el videojuego en cuestión. A partir de ahí se diseñó el modelo de persistencia utilizado para gestionar los datos históricos en el perfil de cada jugador.

Se generaron los artefactos propuestos por la metodología seleccionada en cada una de sus fases y se establecieron los requisitos del sistema mediante las historias de usuario definiendo las tareas a realizar por cada una de ellas. Quedó establecida la arquitectura del sistema y la integración con el motor CNeuroGameEngine. Además se desarrolló la solución propuesta en el trabajo.

Finalmente se probaron las funcionalidades del módulo de gestión de los datos históricos en el videojuego “Meteorix”, obteniéndose resultados satisfactorios.

Con la creación del módulo para la gestión de los datos históricos en los perfiles de usuario en el videojuego “Meteorix”, se le dio cumplimiento al objetivo de esta investigación y se resolvió el problema científico planteado al inicio de la misma.



### **Recomendaciones**

Se recomienda realizar las pruebas de rendimiento del módulo desarrollado en el entorno real donde será utilizado el videojuego.

### Trabajos citados

1. **Pérez Martinto, PC.** *Alternativa metodológica basada en el software clínico psicopedagógico (meteorix) para la estimulación visual de escolares que presentan ambliopía con fijación central.* La Habana : Universidad de Ciencias Pedagógicas “Enrique José Varona”, 2012.
2. **Abt, Clark C.** *Serious Games.* s.l. : University Press of America, 1987.
3. **Duch i Gavaldà, Jordi y Tejedor Navarro, Heliodoro.** *Introducción a los videojuegos.* s.l. : Universidad Abierta de Cataluña, 2005.
4. **Michael, David y Chen, Sande.** *Serious Games: Games that Educate, Train, and Inform.* s.l. : Thomson Course Technology, 2006.
5. **Rivero Cruz, Maité.** *Ingeniería inversa al motor de videojuego CNEURO GameEngine.* 2010.
6. **Kerlinger, Fred N.** *Investigación del comportamiento.* 1982. pág. 105.
7. **Méndez Gonzalez, Ramón.** MERISTATIONXBOX 360. [En línea] 2009. [Citado el: 8 de diciembre de 2011.]  
[http://www.meristation.com/v3/des\\_analisis.php?id=cw498ee13056b24&idj=cw49515f45db1ae&pic=PC](http://www.meristation.com/v3/des_analisis.php?id=cw498ee13056b24&idj=cw49515f45db1ae&pic=PC).
8. Guía del juego. *Sitio Oficial WoW Europe.* [En línea] 2008. [Citado el: 15 de diciembre de 2011.]  
<http://www.wow-europe.com/es/info/basics/>.
9. **Porteiro Rives, Juan.** MERISTATIONXBOX 360. *FIFA 08.* [En línea] 2007. [Citado el: 15 de diciembre de 2011.]  
[http://www.meristation.com/v3/des\\_analisis.php?pic=360&id=cw46fa6a645eb7e&idj=cw4655607c0ad29&idp=&tipo=art&c=1&pos=0](http://www.meristation.com/v3/des_analisis.php?pic=360&id=cw46fa6a645eb7e&idj=cw4655607c0ad29&idp=&tipo=art&c=1&pos=0).
10. **García Sánchez-Colomer, Miguel.** MERISTATIONXBOX 360. *Need For Speed Underground.* [En línea] 2003. [Citado el: 23 de diciembre de 2011.]
11. **Arango Bujardón, Aylin y Quintana Echevarria, Yohandri.** *Módulo Registro de Evaluación para el Simulador Quirúrgico.* s.l. : Universidad de las Ciencias Informáticas, 2008.
12. **Garcia Perez-Schofield, Baltasar.** *Zero: una máquina virtual persistente, orientada a objetos pura y basada en prototipos.* [En línea] 2005. [Citado el: 18 de diciembre de 2011.]  
[http://euitio178.ccu.uniovi.es/infoeuitio/IMG/pdf/transpas\\_zero-2.pdf](http://euitio178.ccu.uniovi.es/infoeuitio/IMG/pdf/transpas_zero-2.pdf).

13. **Álvarez, Sara.** Sistemas gestores de bases de datos. [En línea] 2007. [Citado el: 20 de diciembre de 2011.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
14. **Freijomil Viña, Sebastián.** MeriStationXBOX 360. *Second Life*. [En línea] 2006. [Citado el: 12 de diciembre de 2011.] [http://www.meristation.com/v3/des\\_analisis.php?pic=PC&id=cw44f70f76e7747&idj=6123&idp=&tipo=art&c=1&pos=0](http://www.meristation.com/v3/des_analisis.php?pic=PC&id=cw44f70f76e7747&idj=6123&idp=&tipo=art&c=1&pos=0).
15. **Labra Gayo, José Emilio.** *Lenguaje XML*. s.l. : Universidad de Oviedo, 2002.
16. **Unibertsitatea, Euskal Herriko.** Lectura y escrituras de ficheros. [En línea] 2006. [Citado el: 13 de diciembre de 2011.] <http://usuarios.lycos.es/dhascome/practicas/ProgC/Cap9.pdf>.
17. **Lombera Rodríguez, Hassán, Pacheco, Allende y Alberto, Eliseo.** *Edición de sistemas articulados de cuerpos rígidos para las animaciones en los videojuegos*. s.l. : Universidad de las Ciencias Informáticas, 2008.
18. SPAM LAWS. [En línea] [Citado el: 21 de enero de 2012.] <http://www.spamlaws.com/data-security.html>.
19. Regulacion sobre proteccion de datos. [En línea] [Citado el: 8 de enero de 2012.] [http://www.educastur.princast.es/fp/hola/hola\\_bus/cursos/curso16/documentos/Regulacion\\_proteccion\\_datos%20\\_hola\\_.pdf](http://www.educastur.princast.es/fp/hola/hola_bus/cursos/curso16/documentos/Regulacion_proteccion_datos%20_hola_.pdf).
20. **Tortosa Grau, Leandro.** *Mensajes secretos y códigos con TI-83*. s.l. : Club Universitario, 2006.
21. TinyXml Documentation2.6.0. [En línea] [Citado el: 15 de enero de 2012.] <http://www.grinninglizard.com/tinyxmldocs/index.html>.
22. **Molpeceres, Alberto.** Procesos de desarrollo: RUP, XP y FDD. [En línea] Diciembre de 2002. [Citado el: 17 de febrero de 2012.] <http://www.willydev.net/descargas/Articulos/General/cualxpfdrup.PDF>.
23. **Beck, Kent.** Extreme Programming Explained. [En línea] Septiembre de 1999. [Citado el: 19 de febrero de 2012.] <http://www.mip.sdu.dk/~brianj/Extreme%20Programming%20Explained%20-%20Kent%20Beck;%20Addison-Wesley,%201999.pdf>.
24. **Letelier, Patricio.** Metodologías Ágiles en el Desarrollo de Software. [En línea] 2008. [Citado el: 20 de febrero de 2012.] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.4553&rep=rep1&type=pdf#page=9>.

25. **Hernández Orallo, Enrique.** El Lenguaje Unificado de Modelado (UML). [En línea] 2002. [Citado el: 20 de enero de 2012.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
26. libpq - C Library . [En línea] <http://www.postgresql.org/docs/8.3/static/libpq.html>.
27. **Quiñones, A.** Introducción a PostgreSQL. [En línea] 2 de noviembre de 2007. [Citado el: 12 de enero de 2012.] [http://www.eqsoft.net/presentas/introduccion\\_a\\_postgresql.pdf](http://www.eqsoft.net/presentas/introduccion_a_postgresql.pdf).
28. **Mamani Poma, Orlando Jimmy.** Desarrollo de un Sistema Informático para la Administración de Expedientes de la Universidad José Carlos Mariátegui. [En línea] [Citado el: 18 de enero de 2012.] <http://www.monografias.com/trabajos82/informe-practicas-preprofesionales-ing-sistemas-e-informatica/informe-practicas-preprofesionales-ing-sistemas-e-informatica3.shtml>.
29. **Joskowicz, José.** Reglas y Prácticas en eXtreme Programming. [En línea] 2008. [Citado el: 21 de febrero de 2012.] <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>.
30. **Escribano, Gerardo Fernández.** Introducción a Extreme Programming. [En línea] 2002. [Citado el: 21 de febrero de 2012.] <http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-XP.pdf>.
31. **Olivares Rojas, Mc Juan Carlos.** Patrones de Diseño. [En línea] Mayo de 2007. [Citado el: 15 de enero de 2012.] <http://antares.itmorelia.edu.mx/~jcolivar/courses/dp07b/patrones.pdf>.
32. **Mueller, John.** *Programación Visual C++ .Net.*
33. **Gamma, Erich y Helm, Richard.** *Design Patterns, Elements of Reusable Object-Oriented Software.* 1998.
34. Gestores de Base de Datos. [En línea] 26 de octubre de 2007. [Citado el: 6 de enero de 2012.] <http://www.di.ujaen.es/~barranco/publico/ofimatica/tema7.pdf>.
35. **González Castellanos, Ma. Argenis y Rojas Pabón, Wilson.** Comparación entre sistemas de gestión de bases de datos (SGBD) bajo licenciamiento libre y comercial. [En línea] 2005. [Citado el: 22 de febrero de 2012.] <http://www.ilustrados.com/documentos/sghd.pdf>.

### Glosario

**Realidad Virtual:** Es una ciencia basada en el empleo de ordenadores y otros dispositivos, cuyo fin es producir una apariencia de realidad que permita al usuario tener la sensación de estar presente en ella.

**3D:** En geometría y análisis matemático, un objeto o ente es tridimensional si tiene tres dimensiones. Es decir cada uno de sus puntos puede ser localizado especificando tres números dentro de un cierto rango.

**Estereopsis:** La estereopsis (de stereo que significa sólido, y ophis visión o vista) es el fenómeno dentro de la percepción visual por el cual, a partir de dos imágenes ligeramente diferentes del mundo físico proyectadas en la retina de cada ojo, el cerebro es capaz de recomponer una tridimensional.

**UTF-8:** Es un formato de codificación de caracteres.

**CNeuro:** Proyecto para el desarrollo de aplicaciones informáticas relacionados con las ciencias cognitivas

**Lúdica:** Perteneciente o relativo al juego.

**MySQL Admin:** Es un software de administración de servidores de Bases de Datos de MySQL.

**W3C:** El World Wide Web Consortium, abreviado W3C, es un consorcio internacional que desarrolla estándares para guiar la expansión de la Web.

**Bytes:** Es una secuencia de bits contiguos, cuyo tamaño depende del código de información o código de caracteres en que sea definido.

**Multiparadigma:** Un lenguaje de programación multiparadigma es el cual soporta más de un paradigma de programación como por ejemplo programación estructurada y la programación orientada a objetos.

**C#:** Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft.

**Asp.Net:** Es un framework para aplicaciones web desarrollado y comercializado por Microsoft.

**Perl:** Es un lenguaje de programación diseñado por Larry Wall en 1987.

**Python:** Es un lenguaje de programación de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible.

**ANSI-SQL:** American National Standards Institute standardized Structured Query Language.

**Linux:** *Linux* es un núcleo libre de sistema operativo basado en Unix.

**Unix:** Es un sistema operativo portable, multitarea y multiusuario; desarrollado en 1969 por un grupo de empleados de los laboratorios Bell de AT&T.

**Pruebas de caja negra:** Pruebas que se basan en los requisitos funcionales del software.