



**Universidad de las Ciencias Informáticas  
Centro de Entornos Interactivos 3D (VERTEX)  
Facultad 5**

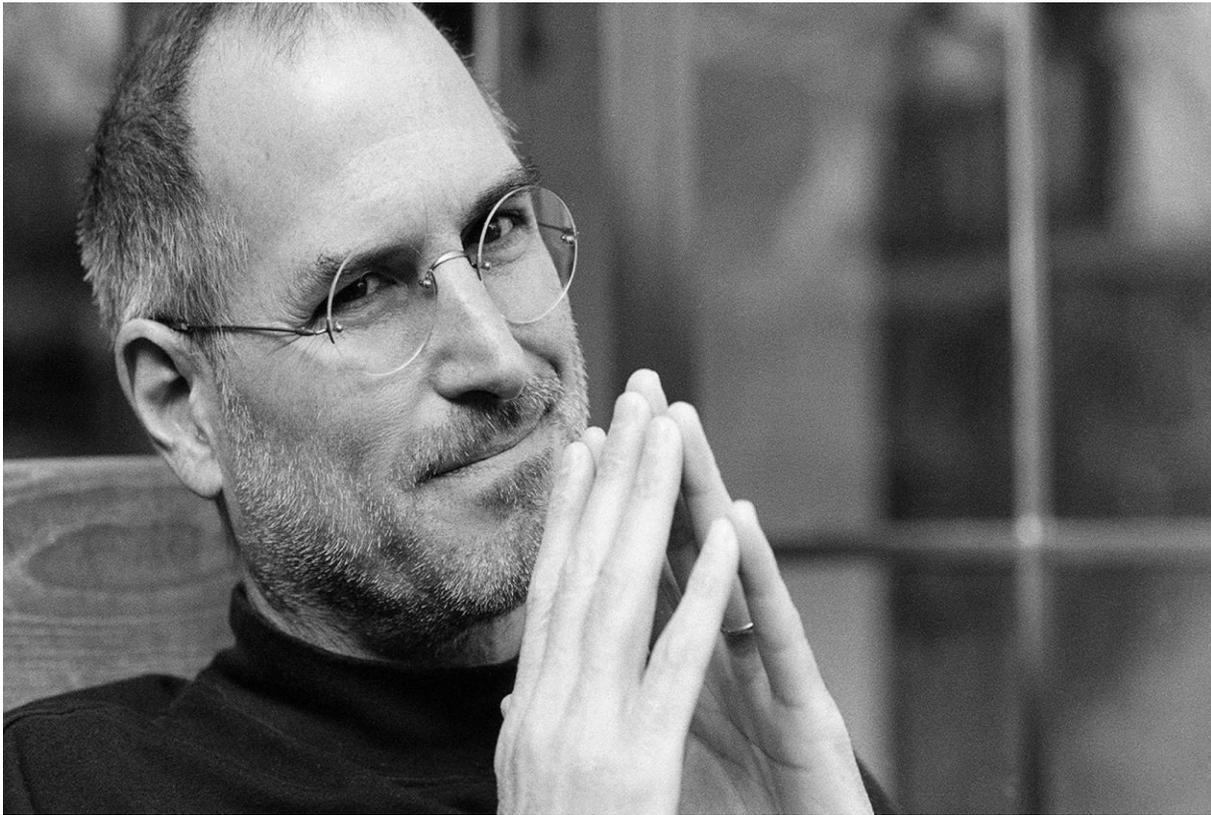
**Plataforma web para la gestión de  
videojuegos serios de navegador con fines  
terapéuticos (Medicando)**

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

**Autor:** Ernesto Gutierrez Ramos

**Tutor:** MSc. Omar Correa Madrigal

“Año 57 de la Revolución”  
La Habana, Cuba  
Junio 2015



*“Tienes que encontrar lo que amas. Y esto es tan verdad para tu trabajo como lo es para tus seres queridos. Tu trabajo va a llenar una gran parte de tu vida y la única manera de estar verdaderamente satisfecho es hacer lo que creas que es un gran trabajo. Y la única manera de hacer un gran trabajo es amar lo que haces. Si no lo has encontrado aún, sigue buscando. No pares.”*

*Steve Jobs*

## Declaración de autoría

Declaro por este medio que yo Ernesto Gutierrez Ramos, con carné de identidad 91100341880, soy el autor principal del trabajo final de tesis “Plataforma *web* para la gestión de videojuegos serios de navegador con fines terapéuticos, Medicando” y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

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

---

Ernesto Gutierrez Ramos

**Autor**

---

MSc. Omar Correa Madrigal

**Tutor**

## Datos de contacto

**Tutor:** MSc. Omar Correa Madrigal.

**Institución:** Universidad de las Ciencias Informáticas (UCI).

**Título:** Máster en Informática Aplicada.

**Síntesis del Tutor:**

- Director del Centro de Entornos Interactivos 3D VERTEX.
- 10 años de experiencia.

**E-mail:** [ocorrea@uci.cu](mailto:ocorrea@uci.cu).

## Dedicatoria

*Dedico esta tesis a mi sobrinito Alain Daniel y a mi papá que son una fuente de inspiración para continuar trabajando en pos de ayudar a las personas con deficiencias visuales.*

*A mi mamá y mi hermano.*

## Agradecimientos

*Si hoy logré convertirme en todo un profesional, es gracias a muchas personas que han estado conmigo y me han apoyado siempre, dándome fuerzas y convenciéndome de que si se podía, y hoy es el día para agradecerle a todos y cada uno de ellos por haber creído en mí. Quiero agradecerle a:*

*Mis padres por ser mi razón de ser, mi inspiración, mi más importante motivo para vivir y por quien hoy me graduó de ingeniero, este título es por ustedes y para ustedes, por ser los mejores padres del mundo, por estar siempre en cada momento, por cada sacrificio que han tenido que hacer por mí, por todo lo que me han querido y todo lo que me han enseñado, la vida entera no me alcanzaría para agradecerles. Los quiero mucho.*

*A mi hermano, por toda su ayuda incondicional y porque mejor no se puede pedir.*

*A mi amiga linda Grethel, porque desde que llegamos a conocernos se convirtió en una amiga muy especial para mí, pude contar con ella, y sé que puedo seguir haciéndolo porque me lo ha demostrado, siempre la seguiré queriendo, gracias mi moshy.*

*A Xiomara y Ángel, que gracias a su ayuda solidaria y desinteresada me prestaron su laptop, quedo agradecido para toda la vida con ustedes por ese gesto.*

*A la familia de Grethel por su cariño incondicional hacia mi persona, se les quiere cantidad.*

*A mi tutor Omar Correa, por sus consejos oportunos y su ayuda.*

*A los miembros del tribunal por sus críticas constructivas en aras de mejorar la calidad del trabajo.*

*A los amigos que encontré aquí en la universidad que han hecho que estos cursos se hayan convertido en buenos tiempos de mi vida. A los profes que me ayudaron y se preocuparon por mí, a los chicos y chicas de mis grupos de estudio, al piquete del medalla hasta las 4am. Un saludo para todos.*

## Resumen

El Centro de Entornos Interactivos 3D (VERTEX) de la Universidad de las Ciencias Informáticas ha incursionado recientemente en la producción de videojuegos serios de navegador que sirven de complemento en el proceso de rehabilitación de pacientes. Estos videojuegos almacenan variables de referencia para que los especialistas puedan llevar el control y seguimiento de los pacientes. Actualmente, no se cuenta con una herramienta que permita a los doctores gestionar videojuegos terapéuticos ni generar estadísticas gráficas de la evolución de los pacientes a partir de la información almacenada. A partir de lo anterior, el presente trabajo se plantea como objetivo, desarrollar una plataforma *web* para la gestión y el control de estadísticas de videojuegos serios de navegador con fines terapéuticos, que brinde solución a la problemática detectada. Para llevar a cabo la construcción del sistema fue seleccionado el *framework* Yii, una herramienta potente para la creación de aplicaciones *web*. Además, en el presente trabajo se detallan los procesos de análisis, diseño, implementación y pruebas de la plataforma en cuestión, guiándose por la metodología de desarrollo *Extreme Programming* (XP). Al concluir el trabajo se confeccionó una aplicación capaz de gestionar videojuegos de navegador con fines terapéuticos y generar estadísticas de pacientes asociados a estos videojuegos.

**Palabras Clave:** gestión, plataforma *web*, videojuegos.

## Índice de contenidos:

INTRODUCCIÓN.....	1
CAPÍTULO 1: Fundamentación teórica .....	- 4 -
1.1    Introducción del capítulo .....	- 4 -
1.2    Términos relacionados con la investigación.....	- 4 -
Plataforma virtual .....	- 4 -
Telemedicina.....	- 4 -
Gestión .....	- 5 -
1.3    Análisis de plataformas virtuales para la gestión de videojuegos .....	- 6 -
1.4    Marcos de trabajo ( <i>Frameworks</i> ).....	- 7 -
Estudio sobre <i>frameworks</i> PHP .....	- 8 -
Lenguajes soportados por el <i>framework</i> Yii.....	- 10 -
1.5    Análisis de Gestores de Base de Datos.....	- 11 -
1.6    Análisis de Entornos de Desarrollo Integrado .....	- 12 -
1.7    Lenguaje Unificado de Modelado .....	- 14 -
1.8    Herramienta de ingeniería de <i>software</i> asistida por computadoras .....	- 14 -
1.9    Metodología de desarrollo de <i>software</i> .....	- 15 -
1.10   Conclusiones del capítulo .....	- 17 -
CAPÍTULO 2: Características del sistema.....	- 19 -
2.1    Introducción del capítulo .....	- 19 -
2.2    Propuesta de sistema .....	- 19 -
Principales funcionalidades .....	- 19 -
Roles de usuarios .....	- 20 -
2.3    Procesos de negocio de la plataforma .....	- 21 -
Gestión de videojuegos en la plataforma .....	- 21 -
Estadísticas asociadas a los videojuegos en la plataforma.....	- 24 -
2.4    Proceso de desarrollo de la plataforma <i>web</i> empleando XP.....	- 24 -

<b>Fase I: Exploración .....</b>	<b>- 24 -</b>
• <b>Historias de Usuario .....</b>	<b>- 25 -</b>
• <b>Características no funcionales .....</b>	<b>- 32 -</b>
<b>Fase II: Planificación.....</b>	<b>- 33 -</b>
• <b>Estimación de esfuerzo por historia de usuario .....</b>	<b>- 34 -</b>
<b>Fase III: Iteraciones.....</b>	<b>- 34 -</b>
• <b>Plan de Duración de las iteraciones .....</b>	<b>- 34 -</b>
• <b>Plan de Entregas.....</b>	<b>- 35 -</b>
• <b>Historias de usuario divididas en tareas.....</b>	<b>- 36 -</b>
<b>2.5 Diseño del sistema .....</b>	<b>- 36 -</b>
<b>2.6 Modelo de datos del sistema.....</b>	<b>- 38 -</b>
<b>2.7 Patrón arquitectónico del sistema .....</b>	<b>- 41 -</b>
<b>2.8 Patrones de diseño .....</b>	<b>- 42 -</b>
<b>2.9 Conclusiones del capítulo .....</b>	<b>- 44 -</b>
<b>CAPÍTULO 3: Implementación y prueba del sistema .....</b>	<b>- 45 -</b>
<b>3.1 Introducción del capítulo .....</b>	<b>- 45 -</b>
<b>3.2 Implementación .....</b>	<b>- 45 -</b>
<b>3.3 Estándar de programación .....</b>	<b>- 51 -</b>
<b>3.4 Diagrama de despliegue .....</b>	<b>- 52 -</b>
<b>3.5 Pruebas .....</b>	<b>- 53 -</b>
<b>3.6 Conclusiones del capítulo .....</b>	<b>- 61 -</b>
<b>Conclusiones generales .....</b>	<b>- 62 -</b>
<b>Recomendaciones.....</b>	<b>- 63 -</b>
<b>Referencias Bibliográficas.....</b>	<b>- 64 -</b>
<b>Anexos .....</b>	<b>- 68 -</b>

## Índice de Ilustraciones

Ilustración 1: Muestra de fichero JSON.....	22 -
Ilustración 2: Variante 1 de flujo de trabajo de la plataforma.....	23 -
Ilustración 3: Variante 2 de flujo de trabajo de la plataforma.....	23 -
Ilustración 4: Muestra del diagrama Entidad-Relación de la BD. ....	39 -
Ilustración 5: Funcionamiento del Patrón Modelo Vista Controlador de Yii. ....	41 -
Ilustración 6: Aplicación del patrón arquitectónico MVC en el módulo Juego .....	42 -
Ilustración 7: Ejemplo de definición de una clase.....	51 -
Ilustración 8: Ejemplo de definición de métodos .....	51 -
Ilustración 9: Ejemplo de llamadas a funciones y asignación de variables .....	52 -
Ilustración 10: Ejemplo de estructura de control indentada .....	52 -
Ilustración 11: Diagrama de despliegue .....	53 -
Ilustración 12: Resultados de las pruebas de aceptación.....	61 -
Ilustración 13: Vista de la gráfica de tiempo de duración .....	68 -
Ilustración 14: Vista de la gráfica de tiempo de reacción.....	68 -
Ilustración 15: Vista de la tabla de rendimiento individual.....	69 -
Ilustración 16: Vista de la gráfica para la matriz de campo visual .....	69 -
Ilustración 17: Vista de la tabla de posiciones por puntuación. ....	70 -
Ilustración 18: Vista de la tabla de posiciones por tiempo de duración .....	70 -
Ilustración 19: Vista de la tabla de posiciones por tiempo de reacción .....	71 -
Ilustración 20: Prototipo de interfaz de usuario Crear Usuario. ....	72 -
Ilustración 21: Prototipo de interfaz de usuario Administrar Usuarios .....	72 -
Ilustración 22: Prototipo de interfaz de usuario Crear Juego .....	73 -
Ilustración 23: Prototipo de interfaz de usuario Gestionar Juego.....	73 -
Ilustración 24: Prototipo de interfaz de usuario Modificar Tratamiento .....	74 -
Ilustración 25: Prototipo de interfaz de usuario Detalles del Tratamiento .....	74 -
Ilustración 26: Diagrama Entidad-Relación del sistema. ....	78 -

## Índice de tablas

Tabla 1: Comparativa de frameworks PHP.....	- 9 -
Tabla 2: Roles del sistema .....	- 20 -
Tabla 3: Historia de usuario - Autenticar Usuario .....	- 25 -
Tabla 4: Historia de Usuario - Gestionar usuario .....	- 26 -
Tabla 5: Historia de Usuario - Gestionar videojuego .....	- 27 -
Tabla 6: Historia de Usuario - Sincronizar datos de videojuego .....	- 28 -
Tabla 7: Historia de Usuario - Gestionar tratamiento.....	- 28 -
Tabla 8: Historia de Usuario – Generar tablas de posiciones grupales .....	- 29 -
Tabla 9: Historia de Usuario - Graficar matriz de campo visual .....	- 30 -
Tabla 10: Historia de Usuario - Graficar variables de tiempo individual.....	- 31 -
Tabla 11: Historia de Usuario - Generar tabla de rendimiento individual.....	- 31 -
Tabla 12 : Estimación de esfuerzo por Historia de Usuario .....	- 34 -
Tabla 13: Plan de Duración de las iteraciones.....	- 35 -
Tabla 14: Plan de entrega de versiones.....	- 35 -
Tabla 15: Historias de Usuario divididas en tareas .....	- 36 -
Tabla 16: Prototipo de tarjeta CRC.....	- 37 -
Tabla 17: Tarjeta CRC - Juego .....	- 37 -
Tabla 18: Tarjeta CRC - Sincronizadora.....	- 37 -
Tabla 19: Tarjeta CRC - Estadística.....	- 38 -
Tabla 20: Tarjeta CRC - Tratamiento .....	- 38 -
Tabla 21: Tarjeta CRC – Usuario .....	- 38 -
Tabla 22: Tarea de ingeniería 1 para HU 1 .....	- 45 -
Tabla 23: Tarea de ingeniería 1 para HU 2.....	- 46 -
Tabla 24: Tarea de ingeniería 1 para HU 3.....	- 47 -
Tabla 25: Tarea de ingeniería 1 para HU 4.....	- 48 -
Tabla 26: Tarea de ingeniería 1 para HU 5.....	- 48 -
Tabla 27: Tarea de ingeniería 1 para HU 6.....	- 49 -

Tabla 28: Tarea de ingeniería 1 para HU 7..... - 49 -

Tabla 29: Tarea de ingeniería 1 para HU 8..... - 50 -

Tabla 30: Tarea de ingeniería 1 para HU 9..... - 50 -

Tabla 31: Caso de prueba HU1\_P1..... - 54 -

Tabla 32: Caso de prueba HU2\_P1..... - 54 -

Tabla 33: Caso de prueba HU2\_P2..... - 55 -

Tabla 34: Caso de prueba HU2\_P3..... - 55 -

Tabla 35: Caso de prueba HU3\_P1..... - 56 -

Tabla 36: Caso de prueba HU4\_P1..... - 56 -

Tabla 37: Caso de prueba HU5\_P1..... - 57 -

Tabla 38: Caso de prueba HU6\_P1..... - 57 -

Tabla 39: Caso de prueba HU6\_P2..... - 58 -

Tabla 40: Caso de prueba HU6\_P3..... - 58 -

Tabla 41: Caso de prueba HU7\_P1..... - 59 -

Tabla 42: Caso de prueba HU8\_P1..... - 59 -

Tabla 43: Caso de prueba HU8\_P2..... - 60 -

Tabla 44: Caso de prueba HU9\_P1..... - 60 -

Tabla 45: Tarea de ingeniería 2 para HU 2..... - 75 -

Tabla 46: Tarea de ingeniería 3 para HU 2..... - 75 -

Tabla 47: Tarea de ingeniería 2 para HU 3..... - 75 -

Tabla 48: Tarea de ingeniería 3 para HU 3..... - 76 -

Tabla 49: Tarea de ingeniería 4 para HU 3..... - 76 -

Tabla 50: Tarea de ingeniería 2 para HU 5..... - 77 -

Tabla 51: Tarea de ingeniería 3 para HU 5..... - 77 -

## INTRODUCCIÓN

El desarrollo y difusión de la informática ha alcanzado fuerza a nivel mundial en los últimos años, llegando a estar presente en cada rama de la sociedad. Esto ha llevado al ser humano a un nivel superior, en lo que a estilo y comodidades de vida respecta. También ha beneficiado las actividades en el área de la salud, facilitando además, la realización de muchos procesos que contribuyen a producir una mejoría en la atención de los pacientes. La unión entre la medicina y las Tecnologías de la Informática y las Comunicaciones (TIC) es asociada al término Telemedicina.

La Universidad de las Ciencias Informáticas (UCI) como parte del proceso de informatización de la sociedad cubana, ha creado un conjunto de proyectos de los cuales varios están encomendados al sector de la salud. Además ha incursionado recientemente en el ámbito de la producción de sistemas de realidad virtual como los videojuegos terapéuticos, siendo centro esencial de este desarrollo el Centro de Entornos Interactivos 3D (VERTEX) de la Facultad 5. Ejemplo de ello son los videojuegos Montaña Rusa Dinámica, Aventura Anatómica, Danzo-Terapia y Meteorix, este último utilizado en la escuela especial Abel Santamaría y en el Instituto Cubano Oftalmológico Ramón Pando Ferrer para la tele-rehabilitación<sup>1</sup> de menores con problemas de agudeza visual, logrando excelentes resultados.

Aunque este tipo de aplicaciones cumplen sus objetivos, no son capaces de integrarse en un único sistema, ya que fueron creadas utilizando diferentes bases tecnológicas y su ejecución es poco eficiente en plataformas de bajas prestaciones. Además, los médicos cubanos no cuentan con una herramienta que les permita gestionar videojuegos terapéuticos, de manera que brinde un nivel de accesibilidad suficiente para que los pacientes realicen los tratamientos de tele-rehabilitación sin tener que trasladarse hasta los centros médicos. Tampoco cuentan con un sistema que ayude en la graficación de estadísticas de la evolución de un paciente a partir del cumplimiento de un tratamiento con videojuegos terapéuticos, recurriendo al modo tradicional de consulta. Estas consultas implican recibir en las salas de espera cuantiosas cifras de personas, provenientes de todas partes del país, llegando a provocar sobrecarga en la cantidad de pacientes que debe atender un especialista, lo que incide de manera negativa en la calidad de la atención a la población.

Atendiendo a lo anterior, se han desarrollado soluciones informáticas con bajos niveles de integración, de accesibilidad y de flexibilidad. En aras de solventar estas limitantes, actualmente se incursiona en el ámbito de la producción de videojuegos serios<sup>2</sup> de navegador con fines terapéuticos, que ofrecen la oportunidad de ver al paciente más allá de los modos clásicos de abordar una problemática de rehabilitación, permitiéndoles llevar el tratamiento al hogar y con una perspectiva recreativa. Esto implica

que los pacientes no necesitan contar con una computadora de altas prestaciones, ni desplazarse hasta los centros médicos para rehabilitarse, solo necesitan contar con una conexión a la *web* que les permita acceder a estos videojuegos.

Teniendo en cuenta dicha situación, se presenta el siguiente **problema de la investigación**: ¿Cómo contribuir a la gestión de videojuegos serios de navegador con fines terapéuticos y al control de sus estadísticas?

Se define como **objeto de estudio** las plataformas *web* para la gestión de videojuegos, y como **campo de acción** las plataformas *web* para la gestión de videojuegos serios de navegador con fines terapéuticos.

Para dar solución al problema científico de la presente investigación se plantea el siguiente **objetivo**: Desarrollar una plataforma *web* que permita la gestión y el control de estadísticas de videojuegos serios de navegador con fines terapéuticos.

Las **tareas a desarrollar** para asegurar el cumplimiento del objetivo trazado son:

- Elaboración de un marco teórico de la investigación a través de un estudio del estado del arte de las plataformas *web* de gestión de videojuegos, para lograr una mejor comprensión del tema.
- Caracterización de las plataformas virtuales para la gestión de videojuegos existentes tanto a nivel nacional como internacional, identificando los principales elementos que las componen y seleccionando aquellas funcionalidades que puedan incluirse en la solución propuesta.
- Caracterización de herramientas y tecnologías para el desarrollo de aplicaciones *web*, seleccionando las que se ajusten a las necesidades de la plataforma propuesta.
- Realización del levantamiento de requisitos funcionales y no funcionales de la plataforma.
- Análisis e implementación de la plataforma *web* Medicando, que brinda la solución al problema planteado.
- Realización de pruebas para validar el cumplimiento de los requerimientos de la solución desarrollada.

Para el correcto desarrollo de la investigación, se emplearán los siguientes **métodos científicos**:

### **Métodos teóricos:**

- **Análisis histórico-lógico:** para el estudio de trabajos similares e investigaciones que abordan el tema de la gestión de videojuegos a nivel nacional e internacional, así como sistemas relacionados con este proceso, que sirvan como punto de partida para el desarrollo de la solución en su conjunto.
- **Método analítico-sintético:** para analizar desde diferentes aristas los conceptos asociados a las plataformas *web* y sintetizar la información recopilada, permitiendo describir las características generales y las relaciones esenciales entre estas.
- **Método de la modelación:** para crear abstracciones con el objetivo de explicar la realidad, se utilizará para la modelación de los diversos diagramas necesarios en cada uno de los flujos de trabajo según la metodología seleccionada.

### **Métodos empíricos:**

- El método **observación científica** se empleará con el objetivo de observar el funcionamiento de algunas plataformas *web* para la gestión de videojuegos existentes, para obtener un registro visual de las características comunes en estas y que pueden formar parte de la solución.

## CAPÍTULO 1: Fundamentación teórica

### 1.1 Introducción del capítulo

Para formalizar una propuesta de solución es imprescindible acercarse al entorno actual en que se encuentra enmarcada la investigación. Así como, comprender los conceptos más importantes y características de las entidades, herramientas y procesos en los cuales pretende influir la investigación. El presente capítulo incluye los principales conceptos asociados al dominio del problema, un estudio del estado del arte de las plataformas virtuales para la gestión de videojuegos, los principales *frameworks* de desarrollo para PHP, los sistemas gestores de base de datos, las metodologías de desarrollo de *software*, entre otros.

### 1.2 Términos relacionados con la investigación

Para entender mejor los temas que serán abordados en la investigación, se hace necesario relacionar a continuación un conjunto de términos y conceptos asociados al dominio del problema.

#### Plataforma virtual

Una plataforma virtual es un sistema que permite la ejecución de diversas aplicaciones bajo un mismo entorno, donde los usuarios acceden a ella a través de un navegador *web*. Con la utilización de estas, el usuario no debe estar en un espacio físico determinado, sino que sólo necesita contar con una conexión a la *web*, que le permita ingresar a la plataforma en cuestión y acceder a sus servicios (DEFINICION.DE, 2014).

La utilización de las plataformas virtuales sirve como base para el desarrollo de aplicaciones que facilitan las labores de trabajo de las comunidades con un campo de investigación determinado. Actualmente este término se ha extendido a la tele-formación, la gestión de videojuegos y la telemedicina, entre otros.

#### Telemedicina

- **Definición de la Organización Mundial de la Salud (OMS):** “ Es el suministro de servicios de atención sanitaria, en los que la distancia constituye un factor crítico, por profesionales que apelan a las TIC con el objetivo de intercambiar datos para hacer diagnósticos, preconizar tratamientos, prevenir enfermedades y accidentes, así como para la formación permanente de los profesionales de atención de salud, con el fin de mejorar la salud de las personas y de las comunidades en que viven” (WHO, 2010).

- **Definición de la American Telemedicine Association:** “Es el intercambio de información médica de un lugar a otro, usando las vías de comunicación electrónicas, para la salud y educación del paciente o el proveedor de los servicios sanitarios, con el objetivo de mejorar la asistencia del paciente” (ASSOCIATION, 2012).
- **Definición del INSALUD:** “ Es la utilización de las TIC como un medio de proveer servicios médicos, independientemente de la localización tanto de los que ofrecen el servicio, los pacientes que lo reciben y la información necesaria para la actividad asistencial” (INSALUD, 2000).

En resumen, el término Telemedicina es la nueva forma de hacer y organizar la provisión de servicios sanitarios, uniendo la medicina y las TIC. Este término resalta la importancia de la tecnología y sus adelantos para ponerla a disposición de la sociedad, logrando un equilibrio equitativo en los servicios que le competen al área de la salud. Los avances logrados en el campo de la Telemedicina ofrecen mayor accesibilidad a los servicios de salud para los pacientes y mayor comodidad para el trabajo de los especialistas.

### **Gestión**

Otro de los términos de importancia para la investigación es la gestión. En informática, generalmente se asocia este término a la implementación de un modelo CRUD (*Create, Read, Update and Delete*, por sus siglas en inglés), o sea, las funciones básicas de insertar, leer, actualizar y eliminar información de bases de datos, pero la gestión no se limita solamente a esto.

La gestión de información en particular es un proceso en donde existe una entrada, almacenamiento, procesamiento y salida de información agregada. El sistema toma los datos que requiere para procesarlos, puede ser alimentado manualmente, ya sea de manera directa por el usuario o automáticamente, donde la información proviene de otros sistemas o módulos (GOÑI CAMEJO, 2000).

La gestión de videojuegos con plataformas virtuales se asocia a la implementación de procesos CRUD<sup>3</sup>, que permiten realizar tareas de promoción, la comercialización de estos productos, entre otros. En casos específicos, como son los videojuegos serios para la salud, la gestión se puede extender a controlar los datos almacenados por estos videojuegos y a generar una serie de estadísticas que ayuden a la toma de decisiones por parte de los médicos.

### 1.3 Análisis de plataformas virtuales para la gestión de videojuegos

Actualmente es posible encontrar varias plataformas virtuales para la gestión de videojuegos situadas en Internet. En la presente investigación se consultó un grupo de estas plataformas para analizar sus características, estas fueron:

- [Gaikai](#) : es un servicio de juego basado en la nube<sup>4</sup> que permite a los usuarios jugar demos de los juegos de PC y aplicaciones de forma instantánea desde una página *web* o dispositivo conectado a internet. El servicio de *streaming*<sup>5</sup> de Gaikai puede ser embebido en cualquier sitio *web* relacionado con juegos, sitios como redes sociales o productos específicos (como dispositivos móviles y televisión digital), según lo determinado por el distribuidor (GAIKAI, 2008).
- [OnLive](#): sistema de distribución de videojuegos bajo demanda alquiler (esto significa que se puede alquilar el juego durante unos días hasta 3 años o siempre). El servicio es un equivalente en la industria de los videojuegos a la computación en la nube, con el juego siendo computado y almacenado en línea (TERDIMAN, 2009).
- [VentureBeat](#): es un sitio de noticias centrado en la tecnología y los negocios. Publica noticias, análisis, entrevistas y vídeos. Abarca una serie de tendencias de empresas, de tecnologías de la salud, nube, juegos, entre otros. Dedicada a la promoción y comercialización de videojuegos (VENTUREBEAT, 2006).
- [Konami](#): *Konami Digital Entertainment, Inc. (Kdei)*, es una subsidiaria de propiedad total de *Konami Corporation*, es un líder editor y fabricante de propiedades de entretenimiento electrónico, que se especializa en el mercado de los videojuegos. Ofrece información y demos de videojuegos, permitiendo comprarlos en línea (KONAMI, 2013).
- [VirtualRehab](#): plataforma para la rehabilitación de pacientes con algún grado de discapacidad física, que combina modernas técnicas de captura de movimiento con tecnología de videojuegos. Requiere un dispositivo Kinect para Windows. Utiliza tecnología de captura de movimiento para convertir al usuario en el actor principal de la rehabilitación. Cada sesión queda registrada en una base de datos de manera que el fisioterapeuta tiene acceso a la información y seguimiento de cada paciente. Contiene licencia privativa (VIRTUALREHAB, 2013).
- [NeuroAtHome](#): plataforma multidisciplinar de neurorrehabilitación<sup>6</sup>, domiciliaria. Mediante la aplicación del dispositivo Kinect y otras tecnologías de Microsoft, acerca la

neurorrehabilitación que generalmente reciben los pacientes en centros especializados, a sus domicilios. Le permite al equipo clínico, desde el centro de rehabilitación, la posibilidad de que diseñe, monitorice y gestione los planes de tratamiento personalizados para sus pacientes (NEUROATHOME, 2012).

### **Características comunes de las plataformas virtuales analizadas**

- **Compatibilidad tecnológica y posibilidades de integración:** Facilidad de adaptación con los distintos sistemas operativos, bases de datos, navegadores *web*, servidores *web*, entre otros.
- **Integración de herramientas de administración y gestión:** Cuentan con herramientas propias de administración y gestión de todos los recursos que en dicho sistema se integran. Entre las herramientas y funcionalidades esenciales se encuentran los procesos de asignación de usuarios, perfiles de acceso y la gestión de videojuegos.
- **Gestión, comercialización y servicios asociados a los videojuegos:** Las plataformas analizadas basan su estilo de negocio en brindar información de los videojuegos y ofrecer facilidades para su comercialización. En el caso de las plataformas orientadas a la salud, se cobra por los servicios asociados a los tratamientos con videojuegos.

### **1.4 Marcos de trabajo (*Frameworks*)**

El desarrollo de aplicaciones *web* está actualmente condicionado por la utilización de los marcos de trabajo; si se concibe un producto medianamente grande es muy recomendable la utilización de un *framework*. Un *framework*, es una estructura de soporte definido, mediante la cual otro proyecto de *software* puede ser organizado y desarrollado. “Es una estructura conceptual y tecnológica compuesta por bibliotecas, componentes y clases que facilitan el desarrollo ágil, seguro y escalable de una aplicación” (AGUIRRE y ESQUIVEL, 2012). Simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Ofrecen una infraestructura que permite tener un código ordenado, limpio y fácil de actualizar, un código seguro, robusto y eficiente (RIEHLE y GROSS, 1998).

La aplicación que se propone desarrollar debe garantizar compatibilidad con diferentes sistemas operativos, gestión de información en bases de datos, trabajo con ficheros compactados y que los usuarios accedan de manera remota a los servicios. Para poder garantizar lo anterior, es necesario implementar una aplicación *web* que utilice PHP 5 como lenguaje de programación, lo que conlleva a una investigación sobre los *frameworks* PHP existentes.

## Estudio sobre *frameworks* PHP

Atendiendo a las necesidades de la aplicación a desarrollar, se considera la información de los cinco mejores *frameworks* para PHP. Se tienen en cuenta los siguientes parámetros para la realización de la comparación que se aprecia en la Tabla 1: Comparativa de *frameworks* PHP (PHPFRAMEWORK, 2014):

- **PHP5:** Indica si el *framework* viene con soporte incorporado para la versión 5 de PHP. Esta versión garantiza manejar los archivos compactados (.ZIP<sup>7</sup>) e incorpora nuevas funciones para manipular arreglos de datos, obtenidos a partir de consultas SQL<sup>8</sup>.
- **MVC:** Indica si el *framework* viene con soporte incorporado para una configuración del patrón Modelo-Vista-Controlador. Este patrón de diseño se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar las tareas de desarrollo de aplicaciones y su posterior mantenimiento.
- **ORM:** Indica si el *framework* es compatible con un asignador de objeto de registro, por lo general una implementación de *ActiveRecord*. Esta funcionalidad permite generar un CRUD a una tabla de la base de datos (BD) de manera rápida.
- **DB Objects:** Indica si el *framework* incluye otros objetos de base de datos, como un *TableGateway*. Esta función es necesaria para acceder a los datos almacenados en la BD.
- **Templates:** Indica si el *framework* tiene un motor de plantillas incorporado que sea flexible en cuanto a la personalización del diseño de interfaz de usuario.
- **Caching:** indica si el *framework* incluye una caché de objetos o de alguna manera otra forma de almacenamiento en caché.
- **Validation:** Indica si el *framework* tiene una validación o filtrado de componente incorporado.
- **Ajax:** Indica si el *framework* viene con soporte incorporado para Ajax que permita establecer comunicación asincrónica en el sistema.
- **Auth Module:** Indica si el *framework* tiene un módulo incorporado para el manejo de la autenticación de usuario y el control de acceso basado en roles.
- **Modules:** Indica si el *framework* es extensible o tiene otros módulos que permitan, entre otras cosas, generar reportes estadísticos.

Tabla 1: Comparativa de frameworks PHP

	Yii	CodeIgniter	CakePhp	Zend	Symfony
<b>PHP5</b>	SI	SI	SI	SI	SI
<b>MVC</b>	SI	SI	SI	SI	SI
<b>ORM</b>	SI	NO	SI	SI	SI
<b>DB Objects</b>	SI	SI	SI	SI	SI
<b>Templates</b>	SI	SI	NO	SI	NO
<b>Caching</b>	SI	SI	SI	SI	SI
<b>Validation</b>	SI	SI	SI	SI	SI
<b>Ajax</b>	SI	NO	SI	SI	SI
<b>Auth module</b>	SI	NO	SI	SI	SI
<b>Modules</b>	SI	NO	SI	SI	SI

Atendiendo a los parámetros medidos en la comparación anterior, resaltan los *frameworks* Yii y Zend como los candidatos más fuertes, debido a que presentan todas las características evaluadas. Para escoger definitivamente, se analiza la facilidad de acceso a documentación actualizada y soporte de ambos *frameworks* desde la UCI, quedando definitivamente Yii como el de mayor accesibilidad. Yii ofrece una amplia gama de video tutoriales y documentación en varios idiomas. Además, cuenta con una gran comunidad de desarrolladores que crece cada día y que cuenta con un fórum excelente ante dudas específicas.

“Yii es un *framework* basado en componentes para el desarrollo de aplicaciones *web* que ayuda a minimizar la complejidad de su construcción a gran escala. Cuenta con un equipo de desarrollo creciente y continúa ganando popularidad entre los desarrolladores de PHP” (WINESETT, 2010). “Todas las aplicaciones Yii se construyen utilizando la arquitectura Modelo-Vista-Controlador (MVC), esto reduce al mínimo la duplicación, ayuda a promover la reutilización de código y facilidad de mantenimiento. Yii ha sido cuidadosamente diseñado para permitir que casi todas las piezas de su código puedan extenderse y personalizarse para satisfacer casi cualquier necesidad o requerimiento” (XUE y ZHUO, 2013).

Como se escoge a Yii como el *framework* para el desarrollo de la plataforma *web* propuesta, a continuación se ofrecen detalles de los lenguajes soportados por el *framework* y que son necesarios utilizar para implementar las funcionalidades del sistema.

## **Lenguajes soportados por el *framework* Yii**

### **PHP 5**

Acrónimo de *Hypertext Preprocessor*, PHP es un lenguaje de código abierto interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Puede ser utilizado en cualquiera de los principales sistemas operativos del mercado actual, incluyendo Linux, UNIX, Microsoft Windows, Mac OS X, RISC OS. Soporta la mayoría de servidores *web*, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet. Tiene la posibilidad de soporte para una gran cantidad de bases de datos, incluye InterBase, FrontBase, mSQL, Direct MS-SQL, MySQL, Oracle (OCI7 and OCI8), Ovrimos, PostgreSQL (ACHOUR *et al.*, 2014).

Las cuatro características más importantes de PHP son simplicidad, seguridad, estabilidad y velocidad. En cuanto a la velocidad, se integra muy bien a otro *software* y requiere pocos recursos de sistema. Garantiza la estabilidad pues utiliza su propio sistema de administración de recursos y posee un sofisticado método para manejar variables, lo que lo hace un sistema robusto. Posee protección contra ataques, proveyendo diferentes niveles de seguridad. La simplicidad es la posibilidad brindada al desarrollador para generar código rápidamente, para esto, el lenguaje incluye gran cantidad de funciones predefinidas (MARIÑO, 2008).

### **HTML 5**

“HTML siglas de *Hypertext Markup Language* (lenguaje de marcado de hipertexto), es un lenguaje diseñado para crear páginas *web* y otros documentos que sean posibles visualizar en el navegador *web*, siendo esencial para la creación de páginas y mostrar el contenido en ellas. El HTML se encarga de desarrollar una descripción sobre los contenidos que aparecen como textos y sobre su estructura, complementando dicho texto con diversos objetos (como imágenes, animaciones, videos)” (LAWSON y SHARP, 2011).

### **JavaScript**

JavaScript es un lenguaje de *script* multiplataforma, diseñado para una fácil incrustación en otros productos y aplicaciones, tales como los navegadores *web*. JavaScript puede ser conectado a los objetos de su entorno para proveer un control programable sobre estos (ZELDMAN, 2010). El empleo de JavaScript para complementar el diseño de interfaz de usuario de la plataforma ofrecerá efectos

de animación que mejoran la visualización de los elementos que la componen, por ejemplo, la graficación de estadísticas.

Como medio de optimización para Javascript se puede utilizar JQuery, que realiza funciones de script frecuentes y utiliza menos líneas de código. Además, para la generación de gráficas se puede utilizar JQChart, que es una de las variantes de JQuery.

### **CSS3**

*Cascading Style Sheets* u Hojas de Estilo en Cascada: es un lenguaje que permite a los desarrolladores describir la presentación de las páginas *web*. Son incluidos los colores, el diseño, y las fuentes, permitiendo adaptar la presentación a los diferentes tipos de dispositivos, tales como diferentes resoluciones de pantallas o impresoras (PÉREZ, 2007). La separación del código HTML de CSS hace que sea fácil mantener sitios, compartir hojas de estilos a través de páginas y adaptarlas a entornos distintos. Esto se conoce como la separación de la estructura (o contenido) de la presentación ((W3C), 2012). Para optimizar CSS3, se puede utilizar Bootstrap, que permite que la plataforma se adapte a las distintas resoluciones de pantalla, aumente el atractivo visual, incorpore las nuevas tendencias del diseño *web* y garantice una fácil actualización del diseño visual.

## **1.5 Análisis de Gestores de Base de Datos**

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la BD, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipularlos, garantizando su seguridad e integridad (ALVAREZ, 2007).

Con el *framework* Yii es posible realizar conexiones a varias BD al mismo tiempo, además que cuenta con la clase *CActiveRecord* que es la que convierte cada tabla de la BD en un modelo mediante el cual se puede trabajar con la información almacenada, esto permite acceder a los datos desde PHP como si fueran variables de tipo arreglo. Entre los SGBD soportados por Yii y de mayor relevancia en la actualidad están Oracle, PostgreSQL y MySQL (CADENA, 2014). A continuación se analizan las características de estos SGBD:

- **Oracle:** Es un sistema de gestión de base de datos relacional. Se considera como uno de los sistemas de bases de datos más completos, destacando su soporte de transacciones, estabilidad; lo que significa que su nivel de fallo disminuye en dependencia de la estabilidad que se requiera y escalabilidad; es decir, su posibilidad de estar preparado para hacerse más

grande sin perder calidad en los servicios ofrecidos, siendo capaz de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes (ORACLE, 2014). Además, es multiplataforma, pero el precio de su licencia es de varios miles de euros, unido a la dificultad que presenta para poder instalarse. Aunque su dominio en el mercado de servidores empresariales ha sido casi total.

- **PostgreSQL:** Es un sistema de gestión de bases de datos objeto-relacional. Está liberado bajo licencia BSD, además es extensible, multiplataforma y presenta modelos de negocios rentables con instalaciones a gran escala (POSTGRESQL, 2012). Tiene ahorros considerables en costos de operación. El *software* ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales, conservando todas las características de rendimiento.
- **MySQL:** Es un SGBD relacional multiplataforma (Linux, Windows, Mac OS). Posee un sistema de privilegios y contraseñas que es muy flexible y seguro, apoyado por los algoritmos de encriptación que se ejecutan cuando se conecta al servidor. Soporta grandes BD. Utiliza un índice que puede usar prefijos de una columna para los tipos de columna CHAR, VARCHAR, BLOB o TEXT. MySQL tiene soporte para comandos SQL para chequear, optimizar y reparar tablas. Es muy rápido, seguro y fácil de usar, esto significa que es un servidor bastante apropiado para acceder a bases de datos en Internet (GROUP, 2014). Presenta características como son: conectividad segura, transacciones y claves foráneas, replicación, búsqueda e indexación de campos de texto y disponibilidad en gran cantidad de plataformas y sistemas (PECOS, 2005).

A partir de realizar un estudio de estos SGBD, se llega a la conclusión de que MySQL se ajusta para ser utilizado en la gestión de datos de la solución propuesta. Esta herramienta posee licencia libre y se integra con el lenguaje de programación seleccionado. Además la integración con el *framework* Yii es excelente y sencilla. Es válido señalar que es el SGBD que viene integrado por defecto en el paquete de programas XAMPP, que se utiliza para poner en funcionamiento la aplicación, por lo que no se necesita realizar instalaciones extras para poner en pleno funcionamiento el sistema.

## **1.6 Análisis de Entornos de Desarrollo Integrado**

Un Entorno de Desarrollo Integrado (IDE, por sus siglas del inglés *Integrated Development Environment*) es un entorno de programación que ha sido empaquetado como un programa de

aplicación. Consiste en un editor de código, un compilador, un depurador y en ocasiones un constructor de interfaz gráfica de usuario. Los IDE pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes (ROUSE, 2009). Una decisión importante a la hora de desarrollar con PHP es la selección del IDE, ya que el entorno de desarrollo que se elija puede suponer una verdadera diferencia en el tiempo de trabajo invertido. Debido a lo anterior se realiza un estudio sobre algunos de los IDE para PHP:

- **Zend Studio:** es el soporte en desarrollos y pruebas de PHP con el set más completo de herramientas para la creación de aplicaciones altamente fiables como lo requiera una empresa. Asegura el desarrollo de *software* mediante la combinación del IDE con un entorno de prueba que agiliza la seguridad de la calidad, integración y las etapas de los procesos. Brinda todo lo que necesita para construir, probar y entregar aplicaciones de alto rendimiento. Contiene distintas funcionalidades como generación de código, formateo de código configurable, pruebas unitarias y desarrollo remoto. Cuesta 299€ y es multiplataforma (Linux, Windows y Mac OS X) (STUDIO, 2014).
- **NetBeans:** es multiplataforma (Windows, Linux, Mac OS X y Solaris), gratuito, de código abierto (con licencia *CDDL*) y se puede utilizar para programas en otros lenguajes además de PHP. A parte de las funciones básicas con las que debería contar cualquier IDE, como resaltado de sintaxis, autocompletado, formateo de código o depurador (xDebug), también cuenta con otras funcionalidades menos comunes como la integración con PHPUnit para las pruebas unitarias y con CVS, Subversion y Mercurial para el control de versiones (MICROSYSTEMS, 2014).
- **PhpStorm:** JetBrains PhpStorm es un IDE multiplataforma comercial para PHP desarrollado sobre la plataforma JetBrains IntelliJ IDEA. Un IDE que tiene un número de funcionalidades similar al Netbeans pero la licencia personal cuesta 88€ y la comercial 176€. Proporciona un editor para PHP, HTML y JavaScript con el análisis de código sobre la marcha, la prevención de errores y refactorizaciones automáticas para PHP y JavaScript. Ofrece soporte para PHP 5, incluyendo generadores, lista en foreach, espacios de nombres, los cierres, los rasgos y la sintaxis de matrices corto. Incluye un editor de pleno derecho de SQL con resultados de la consulta editables (JETBRAINS, 2014).

Atendiendo al estudio anterior, el IDE para la implementación recomendado es NetBeans debido a que está disponible para Windows, Mac, Linux y Solaris, es de código abierto, gratuito y sin restricciones de uso. Existe además un número importante de módulos para extenderlo y que

permiten a los desarrolladores crear rápidamente aplicaciones *web* empresariales, de escritorio y aplicaciones móviles, utilizando PHP, así como Java, JavaScript, Ajax, Groovy, Grails, C / C ++, XHTML y soporte para CSS 3. Puede soportar *frameworks* y está respaldado por una fuerte comunidad de desarrolladores. Ofrece una variada selección de *plugins* de terceros, incluyendo uno para trabajar con el *framework* Yii (MICROSYSTEMS, 2014).

### **1.7 Lenguaje Unificado de Modelado**

El Lenguaje Unificado de Modelado (en lo adelante UML, por sus siglas en inglés, *Unified Modeling Language*) se utiliza para especificar, visualizar, construir y documentar artefactos de un sistema de *software*. Se emplea para diseñar, hojear, configurar, mantener y controlar la información sobre sistemas. Está pensado para emplearse con todos los métodos de desarrollo, etapas del ciclo de vida y dominios de aplicación. UML incluye conceptos semánticos, notación y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes (JACOBSON *et al.*, 1999).

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de *software*, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de *software*, pero no especifica en sí mismo qué metodología o proceso utilizar.

### **1.8 Herramienta de ingeniería de *software* asistida por computadoras**

Las herramientas de ingeniería de *software* asistida por computadoras (en lo adelante CASE, por sus siglas del inglés, *Computer Aided Software Engineering*) brindan ayuda a los analistas, ingenieros de *software* y desarrolladores. Por ejemplo, permiten el modelado de los sistemas mediante diferentes diagramas, generación de código a partir de estos y viceversa. Las herramientas CASE de modelado con UML permiten aplicar la metodología de análisis y diseño orientados a objetos. Además permiten abstraerse del código fuente, en un nivel donde la arquitectura y el diseño se tornan más fáciles de entender (THAMES, 2011).

La herramienta recomendada a utilizar para el modelado de la aplicación es Visual Paradigm, debido fundamentalmente a que es una herramienta multiplataforma que ayuda a una rápida construcción de aplicaciones de calidad. Además, la UCI posee licencia para su utilización. A continuación se brindan detalles de esta herramienta.

## Visual Paradigm

Visual Paradigm para UML es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso y generar documentación. También proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Brinda la posibilidad de generar código a partir de los diagramas para lenguajes como Java y PHP, así como obtener diagramas a partir de código (LTD, 2011). Esta es precisamente una gran ventaja puesto que el sistema será desarrollado en PHP.

### 1.9 Metodología de desarrollo de *software*

Una metodología de desarrollo de *software* es un conjunto de pasos y procedimientos que deben seguirse para desarrollar *software*. Define cómo dividir un proyecto en etapas, qué tareas se llevan a cabo en cada etapa, qué restricciones deben aplicarse, qué técnicas y herramientas se emplean y cómo se controla y gestiona un proyecto. Constituye un medio de estandarización que cubre por completo el proceso de desarrollo de *software*. Posibilita verificar trabajos realizados, corrección de los errores detectados y además provee un lenguaje común entre los analistas, programadores, clientes y usuarios (HERNANDO, 2009). Además, especifica completamente el desarrollo de un sistema informático, pero es necesario tener en cuenta que no es lo mismo utilizar un modelo para la creación de un sitio *web* que para el desarrollo de una aplicación de escritorio, por lo que existe una gran diversidad de metodologías de desarrollo. Debido a lo anterior, es necesario hacer una buena elección, en dependencia de las características del sistema a desarrollar.

#### Metodología Programación Extrema

La Programación Extrema, o *Extreme Programming* (XP según siglas en inglés), surge como posible solución a los problemas derivados del cambio en los requerimientos; esta metodología ofrece la posibilidad de cambiar los requisitos en cualquier momento de la vida de un proyecto, ya que es adaptable a estos cambios. Se centra en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de *software*, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. El ciclo de vida ideal de XP consta de seis fases: la exploración, planificación de la entrega (*release*), iteraciones, producción, mantenimiento y muerte del proyecto (BECK, 2000).

XP es una de las metodologías ágiles más utilizadas en el mundo por sus características, pues se basa en una programación rápida y extrema. Tiene la particularidad de tener como parte del equipo,

al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Las características fundamentales son:

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Programación por parejas: se recomienda que las tareas de ingeniería se lleven a cabo por dos personas en un mismo puesto.
- Frecuente interacción del equipo de programación con el cliente o usuario: Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad: Hacer entregas frecuentes.

### **Metodología Scrum**

“Scrum es un proceso de desarrollo de *software* iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de *software*. Aunque Scrum estaba enfocado a la gestión de procesos de desarrollo de *software*, puede ser utilizado en equipos de mantenimiento de *software* o en una aproximación de gestión de programas: Scrum de Scrums” (SCHWABER, 1997).

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (llamados *requirements churn*) y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes (SUTHERLAND *et al.*, 2007).

Después de realizarse un estudio y análisis de estas metodologías de desarrollo, se tiene a XP como la principal candidata a utilizar, debido a que es adaptada al *software* a desarrollar, así como a las condiciones de trabajo, de forma general. Propone que la comunicación y satisfacción del cliente es lo principal. Es importante destacar de esta metodología, que dada la poca experiencia que tiene el cliente en cuanto a plataformas *web*, los requerimientos actuales pueden estar sujetos a futuros cambios, este es un punto donde la metodología es flexible, ya que permite administrar estos cambios de forma óptima. Además, uno de los objetivos de importancia que aplica esta metodología, es potenciar al máximo el trabajo en grupo, donde los jefes de proyecto, los clientes y desarrolladores son parte del equipo y están involucrados en el desarrollo del *software*.

## 1.10 Conclusiones del capítulo

Después de haber realizado un estudio del estado del arte de las plataformas virtuales para la gestión de videojuegos, los principales *frameworks* para PHP, los SGBD, los IDE, las herramientas CASE y las metodologías de desarrollo de *software*, se concluye que:

- El análisis de los principales conceptos asociados al dominio del problema permitió tener una mejor comprensión del sistema que se va a desarrollar. A partir de esto se resume que las plataformas virtuales permiten implementar servicios ofrecidos por la Telemedicina, permitiendo vincular la gestión de videojuegos con la medicina, principalmente para la rehabilitación, beneficiándose con ello los pacientes y los profesionales de la salud.
- El análisis de las soluciones existentes en el ámbito nacional e internacional permitió estudiar sus características y observar la forma en que funcionan las plataformas *web* para la gestión de videojuegos destinados a la salud. Este estudio indica que en Cuba no se dispone de ninguna aplicación que gestione videojuegos terapéuticos, sin embargo, algunos de los sistemas analizados a nivel internacional se acercan al objetivo que persigue la plataforma Medicando. Las plataformas [NeuroAtHome](#) y [VirtualRehab](#) sirven como puntos de referencias, pero debido a su carácter comercial y privativo, es imposible su reutilización o posibles aplicaciones con fines sociales.
- La caracterización de las tecnologías y herramientas propuestas para la elaboración de la aplicación permitió conocer las ventajas que ofrecen para el desarrollo de plataformas *web*. Teniendo en cuenta lo anterior se escoge el *framework* Yii en su versión 1.1.14 porque permite una construcción estructurada, escalable y extensible de la aplicación, optimizando el empleo de lenguajes como HTML5, CSS3, JavaScript y PHP 5, implementándose sobre el IDE Netbeans 8.0 para minimizar el tiempo de escritura y depuración de códigos. Además, como complemento para garantizar el correcto funcionamiento del *framework*, se selecciona el paquete de programas XAMPP, que incluye el *Apache Server* como servidor *web*, PHP como lenguaje de programación y MySQL como SGBD.
- Para llevar a cabo una correcta ingeniería de *software*, se selecciona XP como metodología de desarrollo, ya que garantiza la generación de artefactos que son necesarios para una mayor comprensión de la aplicación a desarrollar. Complementando el proceso de ingeniería con la utilización del lenguaje de modelado UML y la herramienta CASE Visual Paradigm 8.

De esta manera se está en condiciones de desarrollar la plataforma *web* para la gestión de

videojuegos serios de navegador con fines terapéuticos, Medicando, utilizando las herramientas y tecnologías seleccionadas.

## CAPÍTULO 2: Características del sistema.

### 2.1 Introducción del capítulo

En el presente capítulo se muestran las principales características de la plataforma *web* Medicando, definidas a partir del estudio del estado del arte realizado en el Capítulo 1. Contiene a su vez, la propuesta de solución del sistema a desarrollar, los roles de usuarios definidos para el acceso a la información y la seguridad en la aplicación, las historias de usuario, la planificación de entrega de versiones del producto y el diseño del sistema. Además se describe el funcionamiento del *framework* Yii en su versión 1.1.14 y se describe la estructura a utilizar del patrón arquitectónico Modelo Vista Controlador (MVC) que define dicha herramienta.

### 2.2 Propuesta de sistema

Se propone desarrollar una plataforma *web* que permita gestionar videojuegos de navegador con fines terapéuticos y generar estadísticas a partir de la información almacenada por estos videojuegos. Para lograr lo mencionado anteriormente la plataforma debe garantizar un grupo de funcionalidades que son imprescindibles para su correcto funcionamiento. A continuación se brindan algunas de las particularidades de la plataforma Medicando.

#### Principales funcionalidades

La plataforma propuesta, de manera general, debe garantizar un grupo de funcionalidades para su correcto funcionamiento, estas son:

- **Gestionar videojuegos:** Encargada de la integración, actualización y eliminación de videojuegos en la plataforma. Además, se encarga de los procesos de sincronización de los datos del videojuego tanto los que se ejecutan en línea, como los que son ejecutados en el hogar.
- **Generar estadísticas de los videojuegos:** Encargada de la generación de las estadísticas grupales e individuales de la aplicación. Para las estadísticas individuales se utiliza la generación de la matriz de campo visual (Ilustración 16: Vista de la gráfica para la matriz de campo visual), la generación de la tabla de rendimiento individual (Ilustración 15: Vista de la tabla de rendimiento individual), además de la graficación del tiempo de duración (Ilustración 13: Vista de la gráfica de tiempo de duración) y el tiempo de reacción (Ilustración 14: Vista de la gráfica de tiempo de reacción) por sesiones de juego. Para las estadísticas grupales se utilizan las tablas de posiciones grupales para la puntuación (Ilustración 17: Vista de la tabla de posiciones por

puntuación.), el tiempo de juego (Ilustración 18: Vista de la tabla de posiciones por tiempo de duración) y el tiempo de reacción (Ilustración 19: Vista de la tabla de posiciones por tiempo de reacción).

- **Gestionar tratamientos:** Encargada de que los doctores puedan crear, actualizar y eliminar tratamientos en el sistema. Además permite asignar a los pacientes asignados a un doctor, los tratamientos que este estime conveniente, ya sea los creados por él, como los creados por otros doctores.
- **Gestionar usuarios:** Encargada de la creación, actualización y eliminación de usuarios en el sistema. Además, se encarga del proceso de asignación de roles y permisos en la aplicación.

### **Roles de usuarios**

Parte de la información que brindará la plataforma estará visible para cualquier tipo de usuario que tenga interés en interactuar con el sistema, aunque otro por ciento de la información podrá manipularse por usuarios autenticados. La plataforma tendrá una autenticación por roles, la cual brindará servicios en dependencia del tipo de usuario autenticado, esto permite mantener la seguridad de los datos que se manejan. El administrador es el que tiene todos los privilegios sobre el sistema. A continuación se muestra un listado con los roles y permisos que tendrá cada usuario en la plataforma.

*Tabla 2: Roles del sistema*

Roles definidos	Descripción
<b>Invitado</b>	Usuario que navega por la plataforma sin autenticarse. El nivel de visualización de la información a la que puede acceder es definido por el Administrador.
<b>Paciente</b>	Asignado a los usuarios registrados en el sistema que recibirán tratamientos con videojuegos.
<b>Asistente</b>	Responsable de servir como ayudante de los doctores. Puede registrar y actualizar datos de pacientes, sincronizar datos de videojuegos y consultar estadísticas.
<b>Doctor</b>	Responsable de las tareas de control y seguimiento de los pacientes registrados en el

	sistema. Puede crear y asignar tratamientos a pacientes, consultar estadísticas, gestionar videojuegos, registrar y actualizar datos de pacientes.
<b>Administrador</b>	Responsable de mantener el funcionamiento correcto del sistema. Posee permisos para ejecutar todas las funciones administrativas de la plataforma y es el encargado de gestionar la información. Establece los permisos pertinentes para los diferentes roles de usuarios.

### 2.3 Procesos de negocio de la plataforma

Para una mayor comprensión del funcionamiento de la plataforma, se hace necesario especificar los procesos de negocio implementados en la aplicación. A continuación se brindan detalles sobre estos procesos.

#### Gestión de videojuegos en la plataforma

La funcionalidad principal de la plataforma es la gestión de videojuegos. Para llevar a cabo este proceso de manera organizada y correcta, se establecen una serie de requisitos o restricciones. Los videojuegos que se necesiten integrar a la plataforma deben estar compactados en formato ZIP. Dentro del compactado debe aparecer una carpeta raíz con el nombre del videojuego; en su interior deben aparecer, de manera obligatoria, un fichero JSON<sup>9</sup> donde se almacene la información de salida del videojuego (*data.json*) [Ver Ilustración 1] y un fichero que sirva de ejecutable principal (*index.html*). En el interior de la carpeta raíz también deben estar los directorios y ficheros asociados al funcionamiento del videojuego.

El fichero *data.json* debe tener una estructura como la que se muestra en la Ilustración 1. A partir de las variables almacenadas, la plataforma es capaz de generar una serie de estadísticas de la evolución del paciente. Las variables “tiempoReaccion” y “tiempoDuracion” se utilizan en gráficos lineales y de barras respectivamente; además se utilizan para generar tablas de posiciones grupales. En el caso de la variable “matriz”, se utiliza para generar un gráfico de dispersión que simula el campo visual de los pacientes, ubicando en una matriz, los objetos de la escena del videojuego que el usuario puede ver o no. La variable “score” se utiliza para generar una tabla de posiciones grupal y la tabla de rendimiento

individual. Las variables “tareas” y “hits” se emplean como parámetros para la tabla de rendimiento individual. A continuación se muestra un ejemplo de fichero JSON:

```
[
  {
    "usuario": "alain",
    "sesion": [
      {
        "fecha": "2015-05-22",
        "horaInicio": "11:00",
        "nivel": [
          {
            "level": "1",
            "tareass": "40",
            "hits": "18",
            "score": "450",
            "tiempoReaccion": "4.792969",
            "tiempoDuracion": "2820",
            "matriz": [
              {"x": "492", "y": "485", "b": "true"}, {"x": "44", "y": "555", "b": "true"}
            ]
          }
        ]
      }
    ]
  }
]
```

*Ilustración 1: Muestra de fichero JSON*

Integrar un videojuego implica validar que se cumplan las restricciones, asociarle un nombre, una imagen, una descripción, el área de aplicación del videojuego y permitir o denegar su descarga. Luego de ser almacenado en el servidor, el videojuego será visible para los especialistas y para los pacientes a los que se les asigne como parte de un tratamiento de tele-rehabilitación. A partir de la asignación de un videojuego a un paciente y la información almacenada en el fichero *data.json*, la plataforma sincroniza<sup>10</sup> los datos del JSON con la base de datos del sistema y brindará a los doctores un grupo de estadísticas de la evolución de los pacientes. Lo mencionado anteriormente se puede observar en la *Ilustración 2*.



Ilustración 2: Variante 1 de flujo de trabajo de la plataforma

La plataforma ofrecerá la oportunidad a los pacientes para que descarguen los videojuegos y lleven el tratamiento al hogar. De esta forma se pueden mejorar los resultados obtenidos mediante la rehabilitación, mejorando a su vez la comodidad del paciente. Una vez terminado con el tratamiento, el paciente deberá acceder al sistema y subir el fichero JSON del videojuego (puede subir el videojuego completo que contiene dicho fichero con los datos actualizados del paciente). Para lograr integrar este archivo debe estar compactado en formato ZIP. En la *Ilustración 3* se puede apreciar lo mencionado con anterioridad.

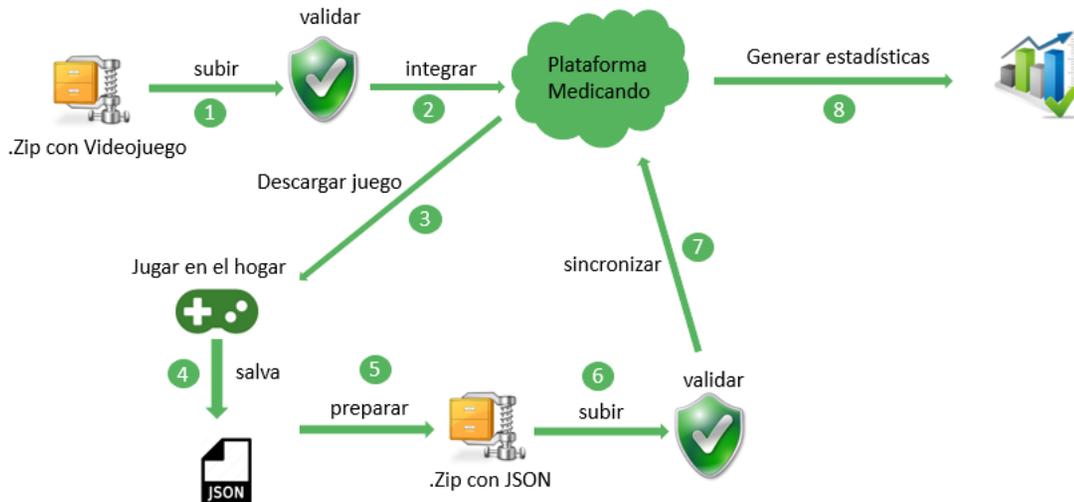


Ilustración 3: Variante 2 de flujo de trabajo de la plataforma

## Estadísticas asociadas a los videojuegos en la plataforma

Uno de los principales servicios que brindará la plataforma es la generación de estadísticas asociadas a los videojuegos integrados en el sistema. Estas estadísticas servirán como fuente para que los doctores conozcan el desempeño en los diferentes videojuegos, sin importar la ubicación del paciente. Con esta información, los especialistas pueden adaptar los tratamientos pautados a sus pacientes de acuerdo con su evolución. Además almacenará un registro histórico de esas estadísticas para que puedan ser utilizados como elementos de comparación entre los diferentes tratamientos. Medicando divide las estadísticas en dos clasificaciones principales, grupales e individuales.

Las estadísticas individuales son las que mostrarán la información de mayor relevancia e interés para los especialistas. Este tipo de estadísticas se enmarca en un paciente específico y su desempeño en un videojuego. En esta clasificación entra la tabla de rendimiento y las gráficas de tiempo de duración y tiempo de reacción por sesiones de juego. En el caso específico de videojuegos para rehabilitación del campo visual como KeylaxyEyes<sup>11</sup>, se grafica la matriz del campo visual del paciente. Por otra parte, en el caso de las estadísticas grupales, el sistema será capaz de generar tablas de *ranking* de usuarios. Estas tablas estarán asociadas a la puntuación del jugador (*score*) por niveles, al tiempo de duración para vencer un nivel y al tiempo de reacción por niveles. Sirven para que los especialistas tengan una visión de los pacientes con mayor o menor rendimiento en el videojuego orientado.

### 2.4 Proceso de desarrollo de la plataforma web empleando XP

El ciclo de vida de un proyecto que utilice la metodología XP incluye, entender lo que el cliente necesita, estimar el esfuerzo, crear la solución y entregar el producto final al cliente. Sin embargo, XP propone un ciclo de vida dinámico, donde se admite expresamente que, en muchos casos, los clientes no son capaces de especificar sus requerimientos al comienzo de un proyecto. Por esto, se trata de realizar ciclos de desarrollo cortos (llamados iteraciones), con entregables funcionales al finalizar cada ciclo. En cada iteración se realiza un ciclo completo de análisis, diseño, desarrollo y pruebas, pero utilizando un conjunto de reglas y prácticas que caracterizan a la metodología XP (JOSKOWICZ, 2008).

#### Fase I: Exploración

Es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente define lo que necesita mediante la redacción de historias de usuario (HU). Los programadores estiman los tiempos de desarrollo en base a esta información. Debe quedar claro que las estimaciones realizadas en esta fase son primarias (ya que estarán basadas en datos de muy alto nivel) y podrían variar cuando se

analicen en cada iteración. Esta fase dura típicamente un par de semanas, el resultado es una visión general del sistema y un plazo total estimado (JOSKOWICZ, 2008).

- **Historias de Usuario**

Las historias de usuario son la técnica utilizada por XP para especificar los requisitos del *software*. Se emplean para hacer estimaciones de tiempo y para el plan de lanzamientos de iteraciones del producto. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. Las historias de usuario deben ser programadas en un tiempo entre una y tres semanas. Si la estimación es superior a las tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra historia de usuario. Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores, utilizando como medida el punto. Un punto, equivale a una semana ideal de programación (5 días laborables). Las historias generalmente valen de 1 a 3 puntos. (ACOSTA y PIS, 2013). A continuación se describen las historias de usuario definidas.

*Tabla 3: Historia de usuario - Autenticar Usuario*

Historia de Usuario	
<b>Número:</b> 1	<b>Nombre:</b> Autenticar usuario
<b>Usuario:</b> Usuarios registrados	<b>Iteración asignada:</b> 1
<b>Prioridad en el Negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<b>Descripción:</b> Los usuarios podrán autenticarse para acceder a los servicios del sistema.  Datos para autenticarse: <ul style="list-style-type: none"> <li>• Nombre de usuario (Obligatorio).</li> <li>• Contraseña (Obligatorio).</li> </ul>	
<b>Observaciones:</b> Para que un usuario pueda autenticarse debe estar registrado en el sistema.	
<b>Prototipo de interfaz de usuario:</b>	

Recordarme

Tabla 4: Historia de Usuario - Gestionar usuario

Historia de Usuario	
<b>Número:</b> 2	<b>Nombre:</b> Gestionar usuario
<b>Usuario:</b> Administrador	<b>Iteración asignada:</b> 1
<b>Prioridad en el Negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<p><b>Descripción:</b></p> <p>El administrador tendrá permisos para crear, editar y eliminar usuarios en el sistema, para esto debe llenar los siguientes campos:</p> <ul style="list-style-type: none"> <li>Nombre(s) (Obligatorio).</li> <li>Apellidos (Obligatorio).</li> <li>Nombre de usuario (Obligatorio).</li> <li>Clave o contraseña (Obligatorio).</li> <li>Correo (Opcional).</li> <li>Foto de perfil (Opcional).</li> </ul>	
<p><b>Observaciones:</b></p> <p>Posterior a la creación de un usuario, el administrador le puede asignar roles con permisos específicos para los servicios de la aplicación.</p>	
<p><b>Prototipo de interfaz de usuario:</b></p>	

- Ilustración 20: Prototipo de interfaz de usuario Crear Usuario.
- Ilustración 21: Prototipo de interfaz de usuario Administrar Usuarios.

Tabla 5: Historia de Usuario - Gestionar videojuego

Historia de Usuario	
<b>Número:</b> 3	<b>Nombre:</b> Gestionar videojuego
<b>Usuario:</b> Administrador o Doctor	<b>Iteración asignada:</b> 1
<b>Prioridad en el Negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<p><b>Descripción:</b></p> <p>El administrador y los doctores se encargarán de la gestión de los videojuegos en la aplicación. Podrán realizar las acciones de adición, modificación, eliminación y visualización de los videojuegos.</p> <p>Datos de un videojuego:</p> <ul style="list-style-type: none"> <li>• Nombre (Obligatorio).</li> <li>• Descripción (Opcional).</li> <li>• Imagen (Opcional).</li> <li>• Fichero (Obligatorio). Este fichero es el compactado con el videojuego.</li> </ul>	
<p><b>Observaciones:</b></p> <p>Para poder integrar un videojuego, este debe cumplir las restricciones de la plataforma.</p>	
<p><b>Prototipo de interfaz de usuario:</b></p> <ul style="list-style-type: none"> <li>• Ilustración 22: Prototipo de interfaz de usuario Crear Juego.</li> <li>• Ilustración 23: Prototipo de interfaz de usuario Gestionar Juego.</li> </ul>	

Tabla 6: Historia de Usuario - Sincronizar datos de videojuego

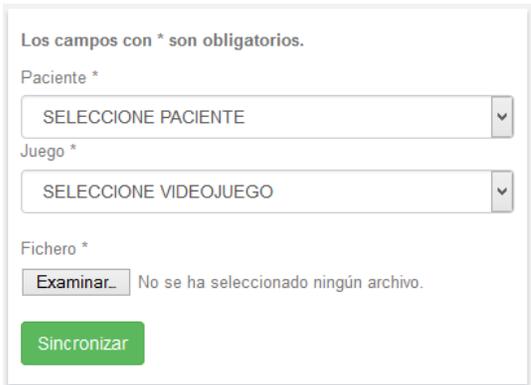
Historia de Usuario	
<b>Número:</b> 4	<b>Nombre:</b> Sincronizar datos de videojuego
<b>Usuario:</b> Usuarios registrados	<b>Iteración asignada:</b> 2
<b>Prioridad en el Negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<b>Descripción:</b>	
<ul style="list-style-type: none"> <li>Los usuarios registrados pueden subir el fichero <i>data.json</i> con la salva del videojuego. La plataforma se encarga de sincronizar estos datos con la base de datos del sistema.</li> </ul>	
<b>Observaciones:</b>	
Para poder sincronizar datos de un videojuego, se debe subir un fichero compactado en formato Zip con el fichero <i>data.json</i> en su interior.	
<b>Prototipo de interfaz de usuario:</b>	
	

Tabla 7: Historia de Usuario - Gestionar tratamiento

Historia de Usuario	
<b>Número:</b> 5	<b>Nombre:</b> Gestionar tratamiento
<b>Usuario:</b> Doctor	<b>Iteración asignada:</b> 2
<b>Prioridad en el Negocio:</b> Alta	<b>Puntos estimados:</b> 1

<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<p><b>Descripción:</b></p> <p>El doctor podrá adicionar, modificar, eliminar y visualizar un tratamiento.</p> <p>Datos de un tratamiento:</p> <ul style="list-style-type: none"> <li>• Nombre (Obligatorio).</li> <li>• Videojuego o videojuegos asignados (Obligatorio).</li> <li>• Descripción (Opcional).</li> <li>• Método (Opcional): Equivale al método médico, o sea, los pasos a seguir por el paciente para cumplir correctamente con el tratamiento.</li> <li>• Evaluación (Opcional).</li> </ul>	
<p><b>Observaciones:</b></p> <p>Para poder crear un tratamiento, el doctor debe haber asignado al menos un videojuego a dicho tratamiento. Una vez cumplido un tratamiento el especialista puede emitir la evaluación final.</p>	
<p><b>Prototipo de interfaz de usuario:</b></p> <ul style="list-style-type: none"> <li>• Ilustración 25: Prototipo de interfaz de usuario Detalles del Tratamiento.</li> <li>• Ilustración 24: Prototipo de interfaz de usuario Modificar Tratamiento.</li> </ul>	

Tabla 8: Historia de Usuario – Generar tablas de posiciones grupales

Historia de Usuario	
<b>Número:</b> 6	<b>Nombre:</b> Generar tablas de posiciones grupales
<b>Usuario:</b> Doctor, Asistente o Administrador	<b>Iteración asignada:</b> 2
<b>Prioridad en el Negocio:</b> Medio	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>• Se podrán consultar las tablas de posiciones asociadas a la puntuación, el tiempo de reacción</li> </ul>	

y el tiempo de duración de los jugadores en un videojuego.
<b>Observaciones:</b> Para generar las tablas deben existir pacientes con sesiones de juego completadas.
<b>Prototipo de interfaz de usuario:</b> <ul style="list-style-type: none"> <li>• Ilustración 17: Vista de la tabla de posiciones por puntuación.</li> <li>• Ilustración 18: Vista de la tabla de posiciones por tiempo de duración.</li> <li>• Ilustración 19: Vista de la tabla de posiciones por tiempo de reacción.</li> </ul>

Tabla 9: Historia de Usuario - Graficar matriz de campo visual

Historia de Usuario	
<b>Número:</b> 7	<b>Nombre:</b> Graficar matriz de campo visual
<b>Usuario:</b> Doctor, Asistente o Administrador	<b>Iteración asignada:</b> 3
<b>Prioridad en el Negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<b>Descripción:</b> <ul style="list-style-type: none"> <li>• Se podrá consultar la gráfica de la matriz de campo visual de los pacientes que realicen rehabilitación con videojuegos y que almacenen esa variable. Se visualiza en forma de gráfica de dispersión.</li> </ul>	
<b>Observaciones:</b> Para generar esta gráfica deben existir pacientes con sesiones de juego completadas.	
<b>Prototipo de interfaz de usuario:</b> <ul style="list-style-type: none"> <li>• Ilustración 16: Vista de la gráfica para la matriz de campo visual.</li> </ul>	

Tabla 10: Historia de Usuario - Graficar variables de tiempo individual

Historia de Usuario	
<b>Número:</b> 8	<b>Nombre:</b> Graficar variables de tiempo individual
<b>Usuario:</b> Doctor, Asistente o Administrador	<b>Iteración asignada:</b> 3
<b>Prioridad en el Negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<b>Descripción:</b>	
<ul style="list-style-type: none"> <li>Se podrán consultar las gráficas de tiempo individual de pacientes que realicen rehabilitación con videojuegos. Para la variable “tiempo de duración” se utiliza una gráfica de barras y para la variable “tiempo de reacción” una gráfica lineal.</li> </ul>	
<b>Observaciones:</b>	
Para generar estas gráficas deben existir pacientes con sesiones de juego completadas.	
<b>Prototipo de interfaz de usuario:</b>	
<ul style="list-style-type: none"> <li>Ilustración 13: Vista de la gráfica de tiempo de duración.</li> <li>Ilustración 14: Vista de la gráfica de tiempo de reacción.</li> </ul>	

Tabla 11: Historia de Usuario - Generar tabla de rendimiento individual

Historia de Usuario	
<b>Número:</b> 9	<b>Nombre:</b> Generar tabla de rendimiento individual
<b>Usuario:</b> Doctor, Asistente o Administrador	<b>Iteración asignada:</b> 3
<b>Prioridad en el Negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<b>Descripción:</b>	
<ul style="list-style-type: none"> <li>Se podrá consultar la tabla de rendimiento individual de pacientes que realicen rehabilitación con videojuegos.</li> </ul>	

**Observaciones:**

Para generar esta gráfica deben existir pacientes con sesiones de juego completadas.

**Prototipo de interfaz de usuario:**

- Ilustración 15: Vista de la tabla de rendimiento individual.

• **Características no funcionales**

“Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Estos requisitos a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema” (SOMMERVILLE *et al.*, 2005). Estos requisitos se refieren a las características o cualidades que debe tener el producto para hacerlo atractivo, usable, rápido y confiable. A continuación se detallan los requisitos no funcionales definidos para el desarrollo de la plataforma Medicando.

**1. Requerimientos de Software**

Se necesitan como requerimientos mínimos:

- **Para el cliente:** Sistema Operativo Windows XP o superior, Linux, navegador *web* estándar con capacidad de interpretación de JavaScript y CSS compatible con Internet Explorer 7 o superior, Netscape, Firefox 3.0 o superior.
- **Para el servidor:** Sistema Operativo Windows XP o superior, Linux (cualquier distribución), servidor *web* Apache 2, lenguaje de programación PHP 5.3 o superior y gestor de BD MySQL 4.0 o superior. Se recomienda la utilización del paquete de programas XAMPP, ya que es multiplataforma e incluye Apache, PHP y MySQL por defecto.

**2. Requerimientos de Hardware**

Se necesitan como requerimientos mínimos:

- **Para el cliente:** Procesador Pentium III, 256 MB de RAM, 100 MB de disco duro.
- **Para el servidor:** Procesador Pentium IV a 2.8 GHz, 1 GB de RAM, 10 GB de disco duro.

### 3. Requerimientos de apariencia

La aplicación contará con una interfaz que no contenga mucha información en sus páginas. Además, incluye animaciones en las gráficas, que generan dinamismo. El diseño podrá verse en distintas resoluciones de pantalla y para diferentes dispositivos como los dispositivos móviles modernos (*tablet, iphone, smartphone*) y las computadoras (*laptop, notebook, PC de escritorio*). Tendrá diferentes tonalidades de color verde en correspondencia con los colores representativos de la estrategia marcaria de los productos de la UCI destinados a la salud, XAVIA. El logotipo estará visible en todo momento y permitirá regresar a la página inicial.

### 4. Requerimientos de Seguridad

- **Confidencialidad:** Existencia de distintos roles que establecen que la información sólo sea vista por aquellos usuarios que posean los privilegios suficientes. Restricción de la ejecución de acciones a usuarios sin credenciales y verificación de que el usuario esté autenticado antes de realizar alguna acción. Además se protegen los videojuegos evitando su descarga no autorizada.
- **Integridad:** Validación de los datos en el servidor para evitar estados inconsistentes. La información manejada por el sistema estará protegida del acceso y divulgación no autorizada. Se debe realizar la confirmación sobre acciones irreversibles como eliminaciones.
- **Disponibilidad:** Los mecanismos utilizados para lograr la seguridad no obstruyen el acceso a la información. Se realizarán salvallas periódicas en otros dispositivos para evitar pérdidas de información en caso de que haya fallos en el sistema.

### Fase II: Planificación

En esta fase, el cliente establece la prioridad de cada historia de usuario. Correspondientemente, los programadores realizan una estimación del esfuerzo necesario para cada una de ellas. Además, se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses (LETELIER y PENADÉS, 2006).

- **Estimación de esfuerzo por historia de usuario**

Se realiza la estimación de esfuerzo que arroja cada historia de usuario con el objetivo de obtener un correcto desarrollo del sistema. A continuación se muestra la estimación realizada:

*Tabla 12 : Estimación de esfuerzo por Historia de Usuario*

Historia de Usuario	Puntos de Estimación(semanas)
<b>Autenticar usuario</b>	1
<b>Gestionar usuario</b>	1
<b>Gestionar videojuegos</b>	1
<b>Sincronizar datos</b>	1
<b>Gestionar tratamiento</b>	1
<b>Generar tablas de posiciones grupales</b>	1
<b>Graficar matriz del campo visual</b>	1
<b>Graficar variables de tiempo individual</b>	1
<b>Generar tabla de rendimiento individual</b>	1

### **Fase III: Iteraciones**

Después de haber identificado las historias de usuario y realizado una previa estimación de esfuerzos, se procede a la planificación de la etapa de implementación del sistema. En esta fase se crea el plan de iteraciones, donde se especifica la prioridad con que se implementarán las historias de usuario organizadas por iteraciones. Teniendo en cuenta el esfuerzo asociado a las historias de usuario y a las prioridades del cliente, se define una versión que sea de valor para este.

- **Plan de Duración de las iteraciones**

A continuación se presenta el plan de duración de las iteraciones. Este plan tiene como finalidad mostrar la duración de cada iteración, así como el orden en que serán implementadas las historias de usuario en cada iteración como se muestra en la tabla siguiente:

Tabla 13: Plan de Duración de las iteraciones

Iteración	Descripción	Historias de Usuario asignadas	Duración total en semanas
1	Se implementan las historias de usuario iniciales, dando al sistema las funcionalidades básicas necesarias y estableciendo una primera versión entregable al cliente.	Autenticar usuario. Gestionar usuario. Gestionar videojuegos.	3
2	Se implementan un grupo de funcionalidades necesarias para lograr la completitud de la gestión de tratamientos con videojuegos.	Sincronizar datos. Gestionar tratamiento. Generar tablas de posiciones grupales.	3
3	Con esta iteración se completan las funcionalidades necesarias para la plataforma propuesta.	Graficar matriz de campo visual. Graficar variables de tiempo individual. Generar tabla de rendimiento individual.	3

- **Plan de Entregas**

En el plan de entrega que se plantea a continuación [Ver Tabla 14: Plan de entrega de versiones] se hace una propuesta de las versiones (*releases*) del sistema. Cada versión se hará al finalizar una iteración.

Tabla 14: Plan de entrega de versiones

Entregable	iteración 1	iteración 2	iteración 3
Plataforma Medicando	Versión 0.3	Versión 0.6	Versión 1.0

- **Historias de usuario divididas en tareas.**

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. Cada una de las historias de usuario se transformará en tareas que serán desarrolladas por los programadores [Ver Tabla 15: Historias de Usuario divididas en tareas] y que corresponden a un período de uno a tres días de desarrollo.

*Tabla 15: Historias de Usuario divididas en tareas*

Historia de Usuario	Tareas a desarrollar
<b>Autenticar usuario</b>	Configurar el módulo de autenticación.
<b>Gestionar usuario</b>	Configurar el módulo de gestión de usuarios.
<b>Gestionar videojuegos</b>	Almacenar videojuegos.
<b>Sincronizar datos</b>	Almacenar información de videojuego.
<b>Gestionar tratamiento</b>	Recopilar información de los tratamientos.
<b>Generar tablas de posiciones grupales</b>	Generar tabla de posiciones para pacientes.
<b>Graficar matriz del campo visual</b>	Generar gráfica de dispersión de puntos.
<b>Graficar variables de tiempo individual</b>	Generar gráficas de tiempo para pacientes.
<b>Generar tabla de rendimiento individual</b>	Generar tabla de datos individual.

## 2.5 Diseño del sistema

Esta es la etapa en donde son evaluadas las historias de usuario para dividir las en tareas, cada tarea representa una característica distinta del sistema y se puede diseñar una prueba de unidad que verifique cada tarea, estas tareas se representan por medio de las tarjetas CRC (Clase-Responsabilidad-Colaborador) [Ver Tabla 16: Prototipo de tarjeta CRC]. Las tarjetas CRC identifican y organizan las clases bajo el paradigma orientado a objetos (lo que incluye asignación de responsabilidades), cada tarjeta contiene el nombre de la clase (que representa una o más historias de usuario), una descripción de las responsabilidades o métodos asociados con la clase, así como la

lista de las clases con que se relaciona o que colaboran con ella. Las tarjetas CRC son el único trabajo de diseño que se genera como parte del proceso de XP (GÓMEZ *et al.*, 2014).

Tabla 16: Prototipo de tarjeta CRC

Nombre de la clase	
Responsabilidades	Colaboradores

**Clase:** Se refiere a las clases persistentes del sistema a desarrollar.

**Responsabilidades:** Función que realiza la clase dentro del sistema.

**Colaboradores:** Relación que tiene la clase con otras clases persistentes del sistema.

A continuación se muestran las tarjetas CRC pertenecientes a la plataforma Medicando:

Tabla 17: Tarjeta CRC - Juego

Tarjeta CRC	
<b>Clase:</b> Juego	
<b>Responsabilidades:</b> Gestionar videojuegos	<b>Colaboradores:</b> Estadística Tratamiento Sincronizadora

Tabla 18: Tarjeta CRC - Sincronizadora

Tarjeta CRC	
<b>Clase:</b> Sincronizadora	
<b>Responsabilidades:</b> Sincronizar datos	<b>Colaboradores:</b> Estadística Juego

Tabla 19: Tarjeta CRC - Estadística

Tarjeta CRC	
<b>Clase:</b> Estadística	
<b>Responsabilidades:</b>	<b>Colaboradores:</b>
Graficar matriz de campo visual	Juego
Graficar variables de tiempo individual	Usuario
Generar tabla de rendimiento individual	
Generar tablas de posiciones grupales	

Tabla 20: Tarjeta CRC - Tratamiento

Tarjeta CRC	
<b>Clase:</b> Tratamiento	
<b>Responsabilidades:</b>	<b>Colaboradores:</b>
Gestionar Tratamiento	Juego
	Usuario

Tabla 21: Tarjeta CRC – Usuario

Tarjeta CRC	
<b>Clase:</b> Usuario	
<b>Responsabilidades:</b>	<b>Colaboradores:</b>
Autenticar usuario	
Gestionar usuario	

## 2.6 Modelo de datos del sistema

El diseño de la base de datos es de gran importancia para el almacenamiento de los datos y para permitir a los usuarios recuperar y actualizar la información en base a sus peticiones posteriormente.

Una característica fundamental del enfoque de BD es que proporciona cierto nivel de abstracción de los datos, al ocultar detalles de almacenamiento que la mayoría de los usuarios no necesitan conocer. Los modelos de datos son el principal instrumento para ofrecer esta abstracción. Además, describen la representación lógica y física de los datos persistentes en el sistema (ALCALÁ y PELTA, 2010).

El modelo de datos de la aplicación tiene alta complejidad, debido a que se utiliza la extensión CRUGE<sup>12</sup> de Yii, que genera un número considerable de entidades manejadas por el sistema, por lo que no se contemplarán todas en el modelo. En su lugar solo se visualizarán aquellas entidades principales que conforman el sistema base [Ver Ilustración 4: Muestra del diagrama Entidad-Relación de la BD.]. El diagrama Entidad-Relación del sistema puede ser consultado en la Ilustración 26: Diagrama Entidad-Relación del sistema.

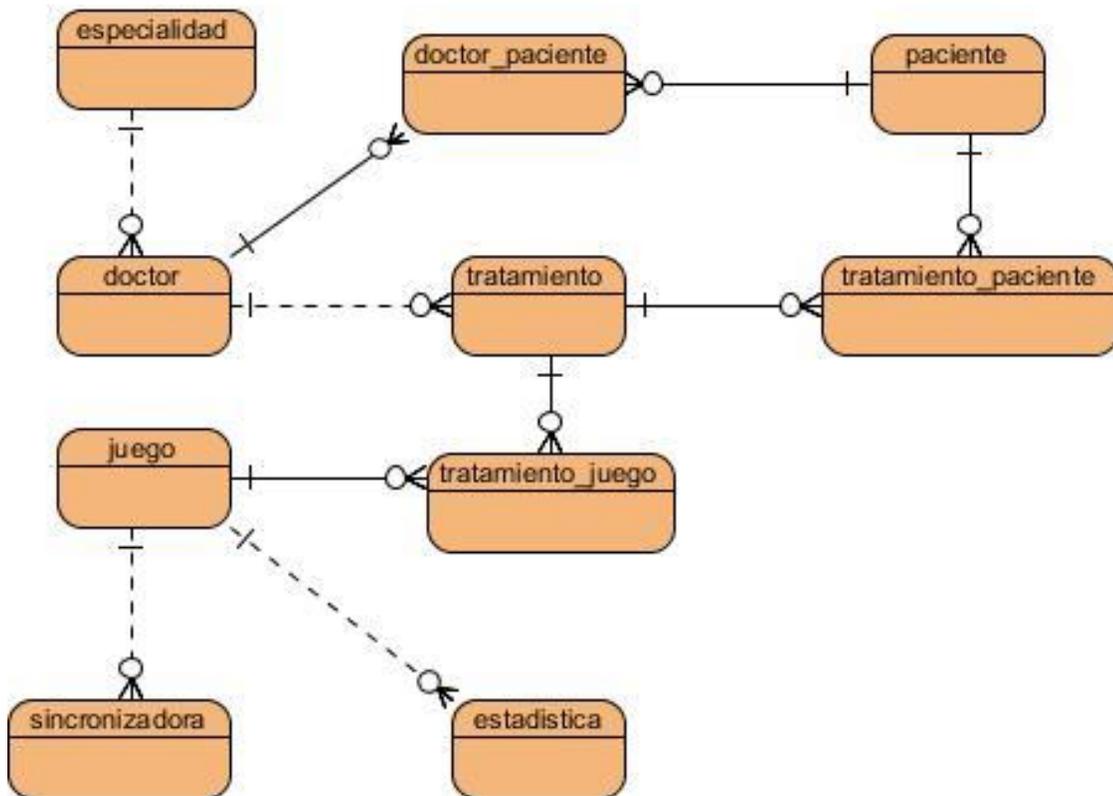


Ilustración 4: Muestra del diagrama Entidad-Relación de la BD.

Para lograr una mayor comprensión del diagrama anterior, a continuación se procede a describir las entidades principales dentro del sistema:

Entidad	Descripción
<b>Juego</b>	<p>En esta entidad se almacenan los datos referentes a los videojuegos y a continuación se detallan sus principales atributos.</p> <ul style="list-style-type: none"> <li>• <b>nombre:</b> se refiere al nombre del videojuego.</li> <li>• <b>imagen:</b> almacena el nombre de la imagen asociada al videojuego.</li> <li>• <b>descrip:</b> se refiere a la descripción del videojuego.</li> <li>• <b>area:</b> almacena el área de aplicación del videojuego (funciones visuales, motoras o cognitivas).</li> <li>• <b>descargable:</b> almacena el valor correspondiente a si el videojuego puede ser descargado o no.</li> </ul>
<b>Tratamiento</b>	<p>En esta entidad se almacenan los datos referentes a los tratamientos y a continuación se detallan sus principales atributos.</p> <ul style="list-style-type: none"> <li>• <b>nombre:</b> se refiere al nombre del tratamiento.</li> <li>• <b>modo:</b> se refiere al modo en que debe ser ejecutado el tratamiento, es el equivalente al método médico.</li> <li>• <b>descripción:</b> almacena una breve descripción del tratamiento.</li> </ul>
<b>Estadística</b>	<p>En esta entidad se almacenan los datos referentes a las estadísticas y a continuación se detallan sus principales atributos.</p> <ul style="list-style-type: none"> <li>• <b>tiempoReaccion:</b> almacena el valor correspondiente al tiempo de reacción.</li> <li>• <b>tiempoDuracion:</b> almacena el valor correspondiente al tiempo de duración de una sesión de juego.</li> <li>• <b>score:</b> almacena el valor correspondiente a la puntuación obtenida en una</li> </ul>

	<p>sesión de juego.</p> <ul style="list-style-type: none"> <li>• <b>matriz:</b> almacena una cadena de texto que es el contenido de la variable asociada a la matriz del campo visual.</li> </ul>
--	---

## 2.7 Patrón arquitectónico del sistema

Yii define una arquitectura que implementa el diseño del patrón Modelo-Vista-Controlador (*model-view-controller*, MVC) [Ver Ilustración 5: Funcionamiento del Patrón Modelo Vista Controlador de Yii.] el cual es adoptado ampliamente en la programación *web*. Este patrón tiene por objetivo separar la lógica del negocio de las consideraciones de la interfaz de usuario; permitiendo que los desarrolladores puedan modificar cada parte fácilmente sin afectar a la otra. En el patrón MVC el modelo representa la información (los datos) y las reglas del negocio; la vista contiene elementos de la interfaz de usuario como textos, formularios de entrada; y el controlador administra la comunicación entre la vista y el modelo (SALAZAR, 2008).

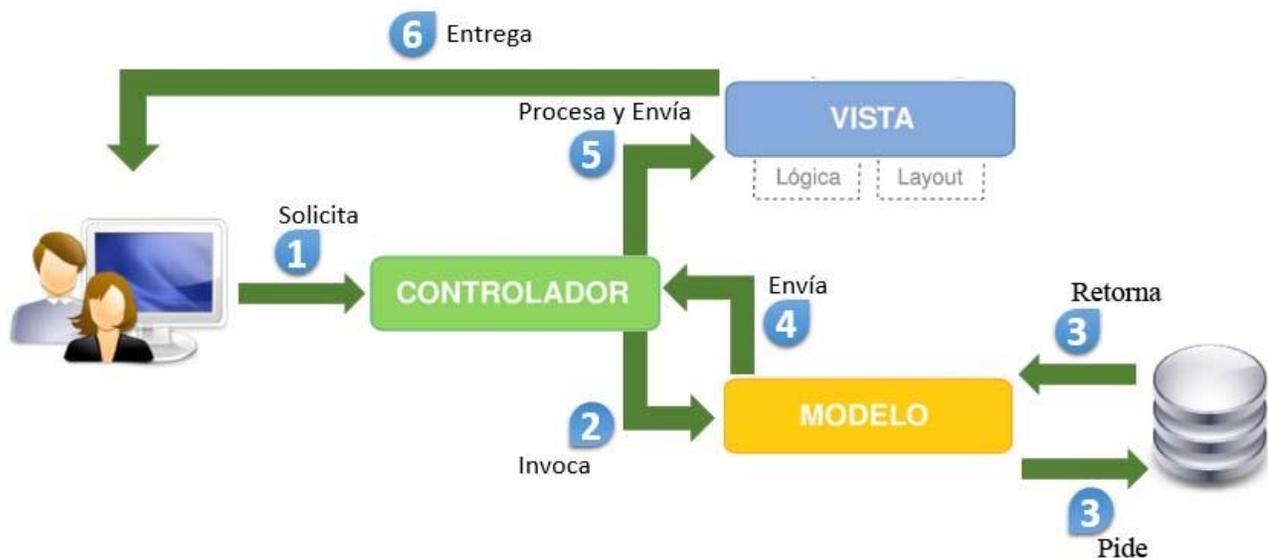


Ilustración 5: Funcionamiento del Patrón Modelo Vista Controlador de Yii.

En la Ilustración 6: Aplicación del patrón arquitectónico MVC se presenta una muestra de la aplicación del patrón arquitectónico Modelo-Vista-Controlador en la plataforma Medicando. A continuación se describen los elementos que conforman el patrón empleado:

- **Modelo:** es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones,

implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). En los *frameworks* actuales normalmente representa una entidad del diagrama entidad-relación.

- **Vista:** presenta el modelo (información y lógica de negocio) en un formato adecuado para que un usuario pueda interactuar (interfaz de usuario).
- **Controlador:** es el intermediario entre la vista y el modelo, su función consiste en controlar el flujo de datos, responder a eventos (generalmente provocados por los usuarios) e invocar peticiones al modelo.

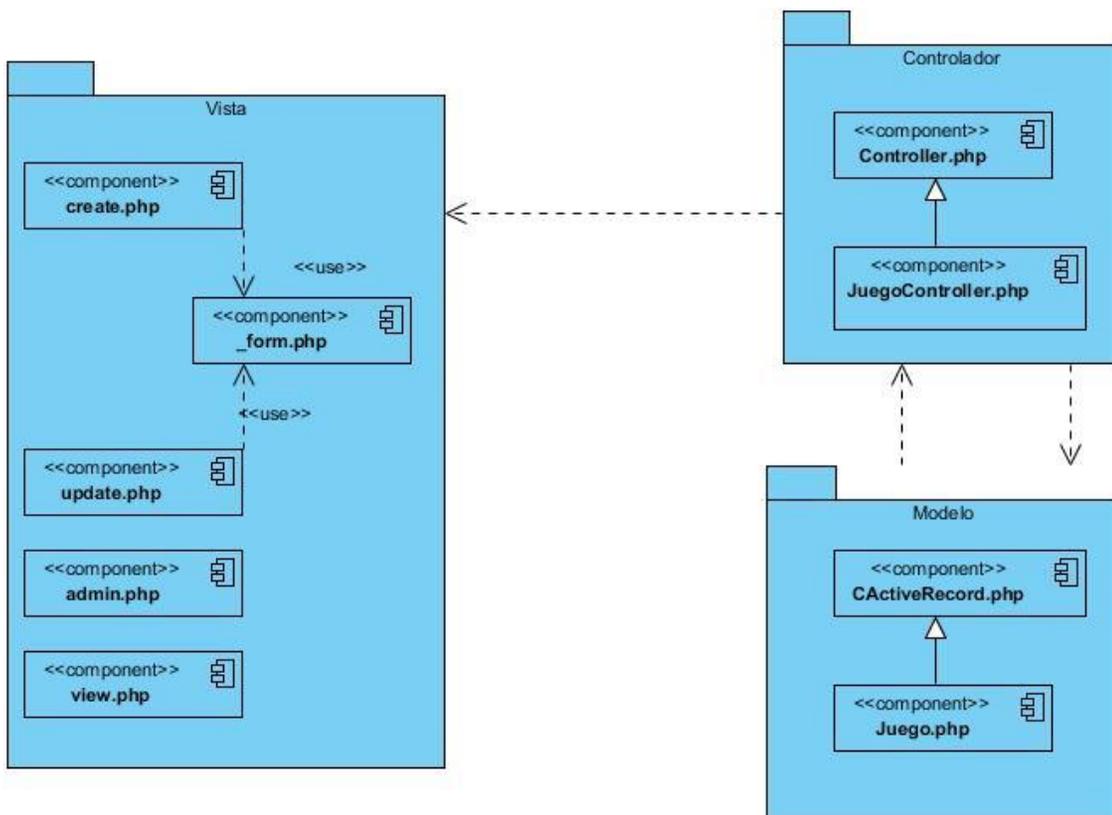


Ilustración 6: Aplicación del patrón arquitectónico MVC en el módulo Juego

## 2.8 Patrones de diseño

Los patrones de diseño tratan los problemas que se repiten y que se presentan en situaciones particulares del diseño, con el fin de proponer soluciones a ellas. Por lo tanto, los patrones de diseño son soluciones exitosas a problemas comunes en el desarrollo de *software* y otros ámbitos referentes al diseño de interacción o interfaces. Los patrones dan nombre y forma a heurísticas abstractas, reglas y buenas prácticas de técnicas orientadas a objetos (LARMAN, 1999).

Los patrones de *software* para la asignación general de responsabilidades (GRASP, por las siglas del inglés *General Responsibility Assignment Software Patterns*) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. A continuación se describen aquellos patrones utilizados de acuerdo a las soluciones que brindan dentro del contexto del sistema:

- **Experto:** es uno de los más utilizados, debido a que se debe tener en cuenta que solo se asignarán responsabilidades a las clases que contengan la información para llevarlas a cabo. Por ejemplo, en el módulo de estadísticas, la clase encargada de crear una estadística es “Estadística”, debido a que es la que posee todos los atributos para su creación. Igual sucede con la clase “Juego” que es la encargada de crear un nuevo juego en el sistema.
- **Creador:** se evidencia en la clase controladora “JuegoController”. Esta clase se encarga de crear instancias del objeto que maneja, o sea, la clase “Juego”. Este patrón brinda soporte a un **bajo acoplamiento**, lo cual supone menos dependencias respecto al mantenimiento, mejores oportunidades de reutilización y propicia una **alta cohesión**.
- **Controlador:** se encarga de que una clase actúe como intermediaria para el manejo de eventos. Este patrón se evidencia en las clases controladoras definidas. En el caso de la clase “JuegoController”, se encarga de manejar todos los eventos asociados a la gestión de videojuegos.

A continuación se detallan algunos de los patrones de la "pandilla de los cuatro" (*Gang - of-Four*) que se utilizaron en el diseño:

- **Singleton:** Este patrón propone una sola instancia de la clase controladora para acceder a los datos y modificarlos, al igual que el *framework* Yii, que usa una sola instancia para manejar las sesiones y los *layout*.
- **Decorator:** Permite añadir funcionalidades dinámicamente a una clase con la creación de otra clase, sin necesidad de la herencia. En el *framework* Yii, los *layouts* decoran a las vistas, pues en un momento determinado les confieren propiedades dinámicas que no poseían.
- **Observer:** define una dependencia de manera que cuando uno de los objetos cambia su estado, notifica este cambio a todos los dependientes. Se trata de un patrón de comportamiento, es decir, está relacionado con algoritmos de funcionamiento y asignación de

responsabilidades a clases y objetos. Se evidencia en la implementación del patrón arquitectónico MVC definido en la aplicación.

## **2.9 Conclusiones del capítulo**

En este capítulo se han abordado los aspectos referentes a la concepción del producto a construir y sus características funcionales y no funcionales. Se detallaron las principales particularidades del funcionamiento de la plataforma Medicando, haciendo especial énfasis en la gestión de videojuegos y la generación de estadísticas. Se expusieron los artefactos generados en las fases que establece la metodología XP. Con la utilización de esta metodología aparecen factores primordiales para el éxito del producto, como el diálogo entre clientes y desarrolladores, argumentando entre lo deseable y lo posible. Ya establecida una solución propuesta para el sistema, y definidos aspectos como la arquitectura y el diseño a utilizar, queda establecida la vía para llevar a cabo la implementación y prueba del sistema.

## CAPÍTULO 3: Implementación y prueba del sistema

### 3.1 Introducción del capítulo

La metodología de desarrollo de *software* XP plantea que la implementación de un *software* debe realizarse de forma iterativa, obteniendo, al culminar cada iteración, un producto funcional que debe ser probado y mostrado al cliente para incrementar la visión de los desarrolladores. En el presente capítulo se detallan las iteraciones realizadas durante la etapa de construcción de la plataforma, además de exponer las tareas de ingeniería generadas para cada historia de usuario que se desarrolló, así como las pruebas de aceptación planificadas para el sistema.

### 3.2 Implementación

Esta fase en XP plantea que las historias de usuario seleccionadas para ser implementadas en cada iteración se van realizando durante el transcurso de la iteración a la cual pertenecen. Por estas razones, se lleva a cabo una revisión del plan de iteraciones y se modifican en caso de ser necesario. Como parte de este plan se descomponen las HU en tareas de ingeniería. Estas pueden estar descritas por un lenguaje técnico y no ser necesariamente entendible por el cliente. Tienen como objetivo definir cada una de las actividades que dan cumplimiento a las HU, de forma tal que se entienda lo que el sistema tiene que hacer y facilite su construcción. A continuación se describen algunas de las tareas de ingeniería correspondientes a las HU del sistema, el resto pueden ser consultadas en los anexos.

- **HU Autenticar Usuario**

*Tabla 22: Tarea de ingeniería 1 para HU 1*

Tarea de ingeniería	
<b>Número de Tarea:</b> 1	<b>Número de HU:</b> 1
<b>Nombre de la Tarea:</b> Configuración del módulo de autenticación	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha de Inicio:</b> 16/3/2015	<b>Fecha de Fin:</b> 20/3/2015
<b>Programador responsable:</b> Ernesto Gutierrez Ramos	
<b>Descripción:</b> Se configura el módulo CRUGE de Yii para la verificación de los datos que son introducidos por el usuario; si son válidos, la aplicación permite el acceso a los servicios. Si no son válidos mostrará un mensaje de error y pedirá nuevamente los	

datos de autenticación. Los datos necesarios para autenticar un usuario son:

- Nombre de usuario (Obligatorio).
- Contraseña (Obligatorio).

- **HU Gestionar Usuario**

*Tabla 23: Tarea de ingeniería 1 para HU 2*

Tarea de ingeniería	
<b>Número de Tarea:</b> 1	<b>Número de HU:</b> 2
<b>Nombre de la Tarea:</b> Crear nuevo usuario	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha de Inicio:</b> 23/3/2015	<b>Fecha de Fin:</b> 27/3/2015
<b>Programador responsable:</b> Ernesto Gutierrez Ramos	
<p><b>Descripción:</b> Se configura el módulo CRUGE para garantizar la creación de usuarios. Los datos necesarios para crear un usuario son:</p> <ul style="list-style-type: none"> <li>• Nombre(s) (Obligatorio).</li> <li>• Apellidos (Obligatorio).</li> <li>• Nombre de usuario (Obligatorio).</li> <li>• Clave o contraseña (Obligatorio).</li> <li>• Correo (Opcional).</li> <li>• Foto de perfil (Opcional).</li> <li>• Rol (Obligatorio).</li> </ul> <p>El proceso debe dividirse en dos pasos, la captura de los datos necesarios para la creación de un usuario y la asignación de roles.</p>	

- HU Gestionar Videojuego

Tabla 24: Tarea de ingeniería 1 para HU 3

Tarea de ingeniería	
<b>Número de Tarea:</b> 1	<b>Número de HU:</b> 3
<b>Nombre de la Tarea:</b> Cargar videojuego	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha de Inicio:</b> 30/3/2015	<b>Fecha de Fin:</b> 3/4/2015
<b>Programador responsable:</b> Ernesto Gutierrez Ramos	
<p><b>Descripción:</b> Se de permitir cargar un nuevo videojuego para la plataforma, así como configurar la opción de permitir o denegar la descarga. Para lograr cargar un videojuego se deben tener en cuenta los siguientes datos:</p> <ul style="list-style-type: none"> <li>• Nombre (Obligatorio).</li> <li>• Descripción (Opcional).</li> <li>• Imagen (Opcional).</li> <li>• Fichero (Obligatorio). Este fichero es el compactado con el videojuego.</li> </ul> <p>Para el caso del fichero, este debe ser de tipo ZIP y cumplir con las restricciones definidas con anterioridad (Ver - 21 - para mayor información).</p>	

- **HU Sincronizar datos**

Tabla 25: Tarea de ingeniería 1 para HU 4

Tarea de ingeniería	
<b>Número de Tarea:</b> 1	<b>Número de HU:</b> 4
<b>Nombre de la Tarea:</b> Sincronización	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha de Inicio:</b> 6/4/2015	<b>Fecha de Fin:</b> 10/4/2015
<b>Programador responsable:</b> Ernesto Gutierrez Ramos	
<p><b>Descripción:</b> Se debe garantizar la sincronización de los datos almacenados en el fichero JSON de los videojuegos con los datos de la tabla estadística de la BD del sistema. Para lograr lo anterior, se tienen que controlar, el paciente, el videojuego y el fichero JSON.</p>	

- **HU Gestionar tratamiento**

Tabla 26: Tarea de ingeniería 1 para HU 5

Tarea de ingeniería	
<b>Número de Tarea:</b> 1	<b>Número de HU:</b> 5
<b>Nombre de la Tarea:</b> Crear nuevo tratamiento	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha de Inicio:</b> 13/4/2015	<b>Fecha de Fin:</b> 17/4/2015
<b>Programador responsable:</b> Ernesto Gutierrez Ramos	
<p><b>Descripción:</b> Se debe garantizar a los doctores la creación de un tratamiento en el sistema. Los datos necesarios para crear un tratamiento son:</p> <ul style="list-style-type: none"> <li>• Nombre (Obligatorio).</li> <li>• Videojuegos asignados (Obligatorio).</li> <li>• Descripción (Opcional).</li> <li>• Método (Opcional): Equivale al método médico, o sea, los pasos a seguir por</li> </ul>	

el paciente para cumplir correctamente con el tratamiento.

- Evaluación (Opcional).

- **HU Generar tablas de posiciones grupales**

*Tabla 27: Tarea de ingeniería 1 para HU 6*

Tarea de ingeniería	
<b>Número de Tarea:</b> 1	<b>Número de HU:</b> 6
<b>Nombre de la Tarea:</b> Generar tablas de posiciones	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha de Inicio:</b> 20/4/2015	<b>Fecha de Fin:</b> 24/4/2015
<b>Programador responsable:</b> Ernesto Gutierrez Ramos	
<b>Descripción:</b> A partir de los datos almacenados en la tabla “estadística” de la BD del sistema, se deben generar tablas de posiciones asociadas a la puntuación, el tiempo de reacción y el tiempo de duración de los pacientes.	

- **HU Graficar matriz de campo visual**

*Tabla 28: Tarea de ingeniería 1 para HU 7*

Tarea de ingeniería	
<b>Número de Tarea:</b> 1	<b>Número de HU:</b> 7
<b>Nombre de la Tarea:</b> Graficar matriz del campo visual de pacientes	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha de Inicio:</b> 27/4/2015	<b>Fecha de Fin:</b> 1/5/2015
<b>Programador responsable:</b> Ernesto Gutierrez Ramos	
<b>Descripción:</b> A partir de los datos almacenados en la tabla “estadística” de la BD del sistema, se debe generar una gráfica de dispersión en la que se muestren los puntos que el paciente logra ver o no. Esta gráfica es específica para el videojuego KeylaxyEyes.	

- **HU Graficar variables de tiempo individual**

*Tabla 29: Tarea de ingeniería 1 para HU 8*

Tarea de ingeniería	
<b>Número de Tarea:</b> 1	<b>Número de HU:</b> 8
<b>Nombre de la Tarea:</b> Generación de gráficas de tiempo individual	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha de Inicio:</b> 4/5/2015	<b>Fecha de Fin:</b> 8/5/2015
<b>Programador responsable:</b> Ernesto Gutierrez Ramos	
<b>Descripción:</b> A partir de los datos almacenados en la tabla “estadística” de la BD del sistema se debe generar una gráfica de barras con intervalos personalizados para el tiempo de duración y una gráfica lineal para mostrar la evolución del tiempo de reacción de un paciente.	

- **HU Generar tabla de rendimiento individual**

*Tabla 30: Tarea de ingeniería 1 para HU 9*

Tarea de ingeniería	
<b>Número de Tarea:</b> 1	<b>Número de HU:</b> 9
<b>Nombre de la Tarea:</b> Generación de la tabla de rendimiento por paciente	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha de Inicio:</b> 11/5/2015	<b>Fecha de Fin:</b> 15/5/2015
<b>Programador responsable:</b> Ernesto Gutierrez Ramos.	
<b>Descripción:</b> A partir de los datos almacenados en la tabla “estadística” de la BD del sistema, se debe generar una tabla en la que se muestre la evolución de un paciente. Esta tabla debe contener datos asociados a la puntuación alcanzada por un jugador y los objetivos vencidos en los diferentes niveles.	

### 3.3 Estándar de programación

Los estándares de programación se enfocan en definir la estructura y apariencia física del código a programar, lo que facilita su entendimiento y lectura. En la implementación de la aplicación se utilizaron diferentes estándares que se describen a continuación:

#### Definición de clases

Las declaraciones de clases tienen su llave de apertura una línea más abajo de la declaración y el nombre de la clase comienza con mayúscula. A continuación un ejemplo de la definición de una clase.

```
class Juego extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'juego';
    }
}
```

*Ilustración 7: Ejemplo de definición de una clase*

#### Definiciones de métodos

El nombre de los métodos comienza con una letra minúscula, en caso de ser un nombre compuesto por dos o más palabras, estas comenzarán con letra mayúscula. A continuación un ejemplo:

```
public function tableName()
{
    return 'juego';
}
```

*Ilustración 8: Ejemplo de definición de métodos*

#### Llamadas a funciones y asignación de variables

Las llamadas a las funciones de una clase se realizarán sin espacios entre el nombre de la función y los paréntesis. En el caso de utilizar llamadas a funciones de una misma clase, se utilizará (*\$this->*) antes del nombre de la función. Las asignaciones se realizarán mediante el signo de igualdad (=). A continuación se muestra un ejemplo:

```

$dataProvider= Paciente::model()->findAll();
$paciente=$this->MisPacientes();

$idusuarioactivo=Yii::app()->user->id;
    $doc= Doctor::model()->find("iduser=$idusuarioactivo");
    $elIdDoctor= $doc->iddoctor;
    
```

Ilustración 9: Ejemplo de llamadas a funciones y asignación de variables

### Estructuras de control

Con el objetivo de mejorar la legibilidad del código fuente se define que las estructuras de control deben estar indentadas. Las estructuras de control incluyen *if*, *for*, *while*, entre otras como se aprecia a continuación:

```

if(isset($_POST['Tratamiento']))
{
    $model->attributes=$_POST['Tratamiento'];
    if($model->save())
        $this->redirect(array('view','id'=>$model->idtratamiento));
}
    
```

Ilustración 10: Ejemplo de estructura de control indentada

### 3.4 Diagrama de despliegue

El diagrama de despliegue permite evaluar de forma visual cómo se encuentran relacionados físicamente los componentes de la aplicación. En este caso la aplicación se encuentra alojada en un servidor *web* Apache y se comunica con un SGBD MySQL [Ver Ilustración 11: Diagrama de despliegue]. A continuación se detallan los elementos que componen el diagrama de despliegue:

- **PC Cliente:** Es la computadora desde donde los usuarios podrán acceder a la plataforma Medicando y se conecta a través del protocolo HTTP con el servidor *web*,
- **Servidor Web:** Es el nodo intermediario entre las PC clientes que realizan sus peticiones y el nodo donde se encuentra el servidor de BD con las informaciones. Este servidor *web* toma los datos, realiza sus funciones y presenta la información a las PC clientes. El servidor *web* se comunica con el nodo donde se encuentra el servidor de BD a través del protocolo TCP/IP.

- **Servidor de BD:** En el nodo donde se encuentra el servidor de base de datos (BD) con toda la información de la plataforma.

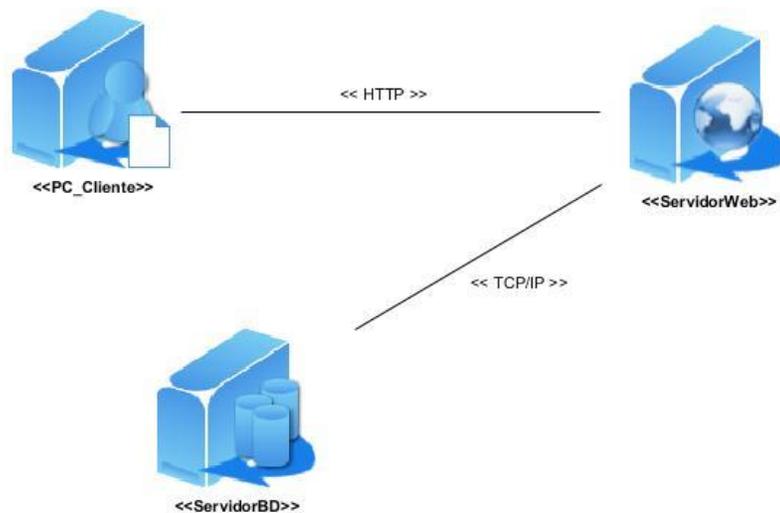


Ilustración 11: Diagrama de despliegue

### 3.5 Pruebas

Uno de los pilares fundamentales de XP es el proceso de prueba, lo cual constituye el último bastión desde el que se puede evaluar la calidad de forma pragmática y descubrir los errores. Las pruebas son un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo sistemáticamente. Por esta razón se debe definir en el proceso de la ingeniería del *software*. Todo esto contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones (PRESSMAN, 2005).

La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente (BECK, 2000).

Se decidió realizar las pruebas de aceptación a la plataforma Medicando, debido a que su objetivo es verificar que el sistema cumpla con los requisitos establecidos por el usuario, por lo que significa la satisfacción del cliente con el producto desarrollado.

### Pruebas de Aceptación

Las pruebas de aceptación, también llamadas pruebas del cliente, son especificadas por el cliente, se centran en las características y funcionalidades generales del sistema que son visibles. Estas pruebas derivan de las HU que se han implementado como parte de la liberación del *software*. Una HU puede tener varias pruebas de aceptación para asegurar su funcionamiento (PRESSMAN, 2005).

Una prueba de aceptación es como una caja negra. Cada una de ellas representa una salida esperada del sistema. Es responsabilidad del cliente verificar la corrección y toma de decisiones acerca de estas pruebas (CORBEA y PÉREZ, 2007). A continuación se muestran las pruebas de aceptación a realizarse en cada iteración.

#### Iteración 1:

*Tabla 31: Caso de prueba HU1\_P1*

Caso de prueba de aceptación	
<b>Código:</b> HU1_P1	<b>HU:</b> 1
<b>Nombre:</b> Autenticar usuario en el sistema.	
<b>Descripción:</b> Prueba para la funcionalidad autenticar usuario.	
<b>Condiciones de ejecución:</b> El usuario debe estar previamente registrado. El usuario y contraseña deben ser válidos.	
<b>Pasos de ejecución:</b> Se intenta autenticar un usuario en el sistema con los datos válidos.	
<b>Resultados esperados:</b> El usuario se autentica correctamente en el sistema.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla 32: Caso de prueba HU2\_P1*

Caso de prueba de aceptación	
<b>Código:</b> HU2_P1	<b>HU:</b> 2
<b>Nombre:</b> Crear nuevo usuario en el sistema.	
<b>Descripción:</b> Prueba para la funcionalidad crear usuario.	

<b>Condiciones de ejecución:</b> El administrador debe introducir correctamente el nombre, los apellidos, el nombre de usuario y la clave o contraseña del usuario a crear. En el caso de la clave, esta debe contener al menos 6 caracteres, que pueden ser letras, dígitos o los caracteres (#@\$%).
<b>Pasos de ejecución:</b> Se intenta crear un usuario en el sistema con los datos válidos.
<b>Resultados esperados:</b> El usuario se crea correctamente en el sistema.
<b>Evaluación de la prueba:</b> Satisfactoria.

Tabla 33: Caso de prueba HU2\_P2

Caso de prueba de aceptación	
<b>Código:</b> HU2_P2	<b>HU:</b> 2
<b>Nombre:</b> Modificar usuario en el sistema.	
<b>Descripción:</b> Prueba para la funcionalidad modificar usuario.	
<b>Condiciones de ejecución:</b> El usuario debe estar previamente registrado. Se utilizará un usuario con sus datos válidos. Deben existir usuarios en el sistema.	
<b>Pasos de ejecución:</b> El administrador intenta modificar los datos de un usuario registrado en el sistema.	
<b>Resultados esperados:</b> El usuario se modifica correctamente en el sistema.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

Tabla 34: Caso de prueba HU2\_P3

Caso de prueba de aceptación	
<b>Código:</b> HU2_P3	<b>HU:</b> 2
<b>Nombre:</b> Eliminar usuario en el sistema.	
<b>Descripción:</b> Prueba para la funcionalidad eliminar usuario.	
<b>Condiciones de ejecución:</b> El usuario debe estar previamente registrado en el sistema.	
<b>Pasos de ejecución:</b> El administrador intenta eliminar un usuario del sistema.	

<b>Resultados esperados:</b> El usuario se elimina correctamente del sistema.
<b>Evaluación de la prueba:</b> Satisfactoria.

Tabla 35: Caso de prueba HU3\_P1

Caso de prueba de aceptación	
<b>Código:</b> HU3_P1	<b>HU:</b> 3
<b>Nombre:</b> Crear nuevo juego en el sistema.	
<b>Descripción:</b> Prueba para la funcionalidad crear nuevo videojuego.	
<b>Condiciones de ejecución:</b> El usuario debe llenar correctamente los campos para la creación de un videojuego. Estos campos son el nombre y el fichero a subir. En el caso del fichero, debe ser un fichero compactado en formato ZIP que en su interior contenga el videojuego.	
<b>Pasos de ejecución:</b> Se intenta crear un nuevo videojuego con los datos correctos.	
<b>Resultados esperados:</b> Se crea un nuevo videojuego correctamente en el sistema.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Iteración 2:**

Tabla 36: Caso de prueba HU4\_P1

Caso de prueba de aceptación	
<b>Código:</b> HU4_P1	<b>HU:</b> 4
<b>Nombre:</b> Sincronizar fichero JSON con el sistema.	
<b>Descripción:</b> Prueba para la funcionalidad sincronizar datos de videojuego.	
<b>Condiciones de ejecución:</b> El usuario debe llenar correctamente los campos requeridos para la sincronización del fichero. Estos campos son, el paciente, el videojuego y el fichero JSON. En el caso del fichero, debe ser un ZIP que en su interior contenga el archivo data.json. El fichero JSON está estructurado correctamente.	
<b>Pasos de ejecución:</b> Se intenta sincronizar un fichero JSON con la BD del sistema.	

<p><b>Resultados esperados:</b> Se sincronizan correctamente los datos del fichero JSON con la base de datos del sistema. Los datos a sincronizar son: la matriz de campo visual, el tiempo de reacción, el tiempo de duración de juego, la fecha y hora de la sesión de juego, el nivel, la puntuación obtenida y la cantidad de tareas cumplidas.</p>
<p><b>Evaluación de la prueba:</b> Satisfactoria.</p>

Tabla 37: Caso de prueba HU5\_P1

Caso de prueba de aceptación	
<b>Código:</b> HU5_P1	<b>HU:</b> 5
<b>Nombre:</b> Crear nuevo tratamiento en el sistema.	
<b>Descripción:</b> Prueba para la funcionalidad crear tratamiento.	
<b>Condiciones de ejecución:</b> El usuario debe llenar correctamente los campos requeridos para la creación de un tratamiento. Estos datos son el nombre del tratamiento y los videojuegos asignados.	
<b>Pasos de ejecución:</b> Se intenta crear un tratamiento por un usuario que cumple con el rol de doctor.	
<b>Resultados esperados:</b> El tratamiento se crea correctamente en el sistema.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

Tabla 38: Caso de prueba HU6\_P1

Caso de prueba de aceptación	
<b>Código:</b> HU6_P1	<b>HU:</b> 6
<b>Nombre:</b> Consultar tabla de posiciones de puntos acumulados.	
<b>Descripción:</b> Prueba para la funcionalidad generar tabla de puntuación grupal.	
<b>Condiciones de ejecución:</b> El usuario debe estar previamente registrado. Deben existir registros de estadísticas de pacientes en el sistema.	
<b>Pasos de ejecución:</b> Se intenta consultar la tabla de posiciones asociadas a la puntuación lograda por los pacientes de un videojuego.	

<b>Resultados esperados:</b> Se consulta correctamente la tabla de posiciones.
<b>Evaluación de la prueba:</b> Satisfactoria.

Tabla 39: Caso de prueba HU6\_P2

Caso de prueba de aceptación	
<b>Código:</b> HU6_P2	<b>HU:</b> 6
<b>Nombre:</b> Consultar tabla de posiciones de tiempo de duración.	
<b>Descripción:</b> Prueba para la funcionalidad generar tabla de tiempo de duración grupal.	
<b>Condiciones de ejecución:</b> El usuario debe estar previamente registrado. Deben existir registros de estadísticas de pacientes en el sistema.	
<b>Pasos de ejecución:</b> Se intenta consultar la tabla de posiciones de tiempo de duración de un videojuego.	
<b>Resultados esperados:</b> Se consulta correctamente la tabla de posiciones de tiempo de duración.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

Tabla 40: Caso de prueba HU6\_P3

Caso de prueba de aceptación	
<b>Código:</b> HU6_P3	<b>HU:</b> 6
<b>Nombre:</b> Consultar tabla de posiciones de tiempo de reacción grupal.	
<b>Descripción:</b> Prueba para la funcionalidad generar tabla de tiempo de reacción grupal.	
<b>Condiciones de ejecución:</b> El usuario debe estar previamente registrado. Deben existir registros de estadísticas de pacientes en el sistema.	
<b>Pasos de ejecución:</b> Se intenta consultar la tabla de posiciones de tiempo de reacción de un videojuego.	
<b>Resultados esperados:</b> Se consulta correctamente la tabla de posiciones.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Iteración 3:**

*Tabla 41: Caso de prueba HU7\_P1*

Caso de prueba de aceptación	
<b>Código:</b> HU7_P1	<b>HU:</b> 7
<b>Nombre:</b> Consultar gráfica de matriz del campo visual.	
<b>Descripción:</b> Prueba para la funcionalidad generar gráfica de matriz.	
<b>Condiciones de ejecución:</b> El usuario debe estar previamente registrado. Deben existir registros de estadísticas de pacientes en el sistema.	
<b>Pasos de ejecución:</b> Se intenta consultar la gráfica de la matriz del campo visual de un paciente.	
<b>Resultados esperados:</b> Se consulta correctamente la gráfica de matriz del campo visual en el sistema.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla 42: Caso de prueba HU8\_P1*

Caso de prueba de aceptación	
<b>Código:</b> HU8_P1	<b>HU:</b> 8
<b>Nombre:</b> Consultar gráfica de tiempo de reacción individual.	
<b>Descripción:</b> Prueba para la funcionalidad generar gráfica de tiempo de reacción.	
<b>Condiciones de ejecución:</b> El usuario debe estar previamente registrado. Deben existir registros de estadísticas de pacientes en el sistema.	
<b>Pasos de ejecución:</b> Se intenta consultar la gráfica de tiempo de reacción de un paciente.	
<b>Resultados esperados:</b> Se consulta correctamente la gráfica de tiempo de reacción individual en el sistema.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

Tabla 43: Caso de prueba HU8\_P2

Caso de prueba de aceptación	
<b>Código:</b> HU8_P2	<b>HU:</b> 8
<b>Nombre:</b> Consultar gráfica de tiempo de duración individual.	
<b>Descripción:</b> Prueba para la funcionalidad generar gráfica de tiempo de duración.	
<b>Condiciones de ejecución:</b> El usuario debe estar previamente registrado. Deben existir registros de estadísticas de pacientes en el sistema.	
<b>Pasos de ejecución:</b> Se intenta consultar la gráfica de tiempo de duración de un paciente.	
<b>Resultados esperados:</b> Se consulta correctamente la gráfica de tiempo de duración individual en el sistema.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

Tabla 44: Caso de prueba HU9\_P1

Caso de prueba de aceptación	
<b>Código:</b> HU9_P1	<b>HU:</b> 9
<b>Nombre:</b> Consultar tabla de rendimiento individual.	
<b>Descripción:</b> Prueba para la funcionalidad generar tabla de rendimiento.	
<b>Condiciones de ejecución:</b> El usuario debe estar previamente registrado. Deben existir registros de estadísticas de pacientes en el sistema.	
<b>Pasos de ejecución:</b> Se intenta consultar la tabla de rendimiento de un paciente.	
<b>Resultados esperados:</b> Se consulta correctamente la tabla de rendimiento individual en el sistema.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

### Resultados de las pruebas de aceptación

Durante el proceso de pruebas, se realizaron un total de 14 casos de pruebas de aceptación (caja negra), de ellos todos resultaron satisfactorios, lo cual representa el 100% de satisfacción [Ver

Ilustración 12: Resultados de las pruebas de aceptación]. La aplicación de estas pruebas permitió comprobar la correcta implementación de las historias de usuario definidas con anterioridad.



*Ilustración 12: Resultados de las pruebas de aceptación*

### 3.6 Conclusiones del capítulo

En este capítulo se especificó el proceso de implementación del sistema. Se detallaron las tareas de ingeniería asociadas a las HU definidas. Además, se definió el estándar de programación a utilizar y se detallaron las pruebas de aceptación que brindarán al cliente conformidad y seguridad ante las funcionalidades del sistema.

## Conclusiones generales

En el presente trabajo de diploma se presentó la investigación y el posterior proceso de desarrollo de la plataforma Medicando y se arriba a las siguientes conclusiones:

- Con la implementación de una aplicación capaz de gestionar videojuegos de navegador con fines terapéuticos y generar estadísticas de pacientes, se dio cumplimiento al objetivo de la investigación presentada.
- El proceso de desarrollo guiado por la metodología XP e implementado con el framework Yii de conjunto con MySQL, facilitaron la creación de la plataforma en un periodo corto de tiempo. Además, se evidenció que Yii es una tecnología flexible a la hora de integrarse con bibliotecas de terceros como Bootstrap y jqChart.

## Recomendaciones

- Continuar con el estudio de las plataformas para gestión de tratamientos de rehabilitación con videojuegos, con el objetivo de encontrar mejoras e incluirlas en futuras versiones.
- Perfeccionar el módulo de sincronización de datos para flexibilizar la estructura del fichero JSON soportado.
- Perfeccionar el módulo de generación de estadísticas para visualizar las estadísticas de manera dinámica y personalizada.

## Referencias Bibliográficas

- (W3C), T. W. W. W. C. *HTML & CSS* [Consultado el: 22 de Octubre de 2014]. Disponible en: <http://www.w3.org/standards/webdesign/htmlcss#whatcss>.
- ACOSTA, R. V. y PIS, V. M. A. *Portal web para la gestión de información de los procesos sustantivos de la Facultad 5 de la Universidad de las Ciencias Informáticas*. Tesis para optar por el título de Ingeniero en Ciencias Informáticas, Facultad 5. Universidad de las Ciencias Informáticas, 2013.
- ACHOUR, M.; BETZ, F., *et al. Manual de PHP* [Consultado el: 17 de Octubre de 2014]. Disponible en: <http://docs.php.net/manual/es>.
- AGUIRRE, C. B. E. y ESQUIVEL, P. *Comparativa de Frameworks para el desarrollo de aplicaciones con php* [Consultado el: 24 de junio de 2015]. Disponible en: <http://dSPACE.uazuay.edu.ec/bitstream/datos/2125/1/08920.pdf>.
- ALCALÁ, J. y PELTA, D. *Fundamentos de Informática, modelado de Bases de Datos* [Consultado el: 24 de Junio de 2015]. Disponible en: <http://www.ugr.es/~jalcala/teaching/informatica/teoria/Tema3.pdf>.
- ALVAREZ, S. *Sistemas gestores de bases de datos* Última actualización: 31 de Julio de 2007. [Consultado el: 17 de Octubre de 2014]. Disponible en: <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
- ASSOCIATION, A. T. *What is Telemedicine?* [Consultado el: 19 de Junio de 2015]. Disponible en: <http://www.americantelemed.org/about-telemedicine/what-is-telemedicine>.
- BECK, K. *Extreme programming explained: embrace change*. Addison-Wesley Professional, 2000. ISBN 0201616416.
- CADENA, J. P. *Implementación de Yii Framework con MVC y SCRUM en el ciclo de producción de la empresa eDesarrollos*. Tesis para obtener el título de Ingeniería en Telemática y Sistemas, División de Ingeniería y Tecnologías. Universidad de la Sierra, 2014.
- CORBEA, M. R. y PÉREZ, M. O. *LA METODOLOGÍA XP APLICABLE AL DESARROLLO DEL SOFTWARE EDUCATIVO EN CUBA*. TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS, Universidad de las Ciencias Informáticas, 2007.
- DEFINICION.DE. *Definición de plataforma virtual* [Consultado el: 2 de Octubre de 2014]. Disponible en: <http://definicion.de/plataforma-virtual/>.
- GAIKAI. *What is Gaikai?* [Consultado el: 19 de Junio de 2015]. Disponible en: <https://www.gaikai.com/#!/qa>.

- GÓMEZ, A. R.; DUARTE, A. Q., *et al.* Desarrollo ágil de software aplicando programación extrema. *Revista Ingenio*, 2014, vol. 5, nº 1, p. 24-29. ISSN 2389-864X.
- GOÑI CAMEJO, I. Algunas reflexiones sobre el concepto de información y sus implicaciones para el desarrollo de las ciencias de la información. *Acimed*, 2000, vol. 8, nº p. 201-207. ISSN 1024-9435.
- GROUP, T. P. *API MySQL original* [Consultado el: 17 de Octubre de 2014]. Disponible en: <http://php.net/manual/es/book.mysql.php>.
- HERNANDO, R. *Metologías de desarrollo de software* [Consultado el: 29 de Octubre de 2014]. Disponible en: [http://www.rhernando.net/modules/tutorials/doc/ing/met\\_soft.html](http://www.rhernando.net/modules/tutorials/doc/ing/met_soft.html).
- INSALUD. *Plan de Telemedicina del INSALUD*. Dirección General de Organización y Planificación Sanitaria del INSALUD. Subdirección General de Informática del INSALUD. Subdirección General de Sistemas y Tecnologías de la Información del Ministerio de Sanidad y Consumo. Madrid, [Consultado el: 20 de Enero de 2015].
- JACOBSON, I.; BOOCH, G., *et al.* *El Lenguaje Unificado de Modelado (UML)* [Consultado el: 29 de Octubre de 2014]. Disponible en: <http://ingenieriasoftware2011.files.wordpress.com/2011/07/el-lenguaje-unificado-de-modelado-manual-de-referencia.pdf>.
- JETBRAINS. *PHP IDE :: JetBrains PhpStorm* [Consultado el: 26 de Octubre de 2014]. Disponible en: <https://www.jetbrains.com/phpstorm/>.
- JOSKOWICZ, J. Reglas y prácticas en eXtreme Programming. *Universidad de Vigo*, 2008, nº p. 22.
- KONAMI. *About* [Consultado el: 19 de junio de 2015]. Disponible en: <https://www.konami.com/about/>.
- LARMAN, C. *UML y Patrones*. 1999, 520 p. Disponible en: [http://sunshine.prod.uci.edu/gridfs/sunshine/books/Craig\\_Larman\\_-\\_UML\\_y\\_Patrones.pdf](http://sunshine.prod.uci.edu/gridfs/sunshine/books/Craig_Larman_-_UML_y_Patrones.pdf).
- LAWSON, B. y SHARP, R. *Introducing HTML5*. 2011, 240 p. Disponible en: <http://sunshine.prod.uci.edu/gridfs/sunshine/books/Introducing-HTML5---New-Riders.pdf>. ISBN 978-0-321-68729-6.
- LETELIER, P. y PENADÉS, M. C. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)* [Consultado el: 10 de Febrero de 2015]. Disponible en: [http://www.cyta.com.ar/ta0502/b\\_v5n2a1.htm](http://www.cyta.com.ar/ta0502/b_v5n2a1.htm).
- LTD, V. P. I. *Visual Paradigm for UML* [Consultado el: 30 de Octubre de 2014]. Disponible en: <http://www.visual-paradigm.com>.
- MARIÑO, C. V. *Programación en PHP5. Nivel Básico*. 2008, 180 p. Disponible en: <https://n-1.cc/file/download/745373>.

- MICROSYSTEMS, S. *NetBeans IDE 8.0* [Consultado el: 17 de Octubre de 2014]. Disponible en: <https://netbeans.org/community/releases/80>.
- NEUROATHOME. *¿Para que sirve NeuroAtHome?* [Consultado el: 19 de Junio de 2015]. Disponible en: <http://www.neuroathome.org/p/home.html>.
- ORACLE. *Productos y servicios Oracle* [Consultado el: 25 de Octubre de 2014]. Disponible en: <http://www.oracle.com/es/products/index.html>.
- PECOS, D. *PostgreSQL vs. MySQL* [Consultado el: 24 de Junio de 2015]. Disponible en: <http://danielpecos.com/documents/postgresql-vs-mysql/>.
- PÉREZ, J. E. *Introducción a CSS* [Consultado el: 17 de Octubre de 2014]. Disponible en: <http://www.librosweb.es/css>.
- PHPFRAMEWORK. *Top 10 php frameworks* [Consultado el: 12 de Noviembre de 2014]. Disponible en: <http://www.phpframeworks.com/top-10-php-frameworks/>.
- POSTGRESQL. *PostgreSQL Global Development Group* [Consultado el: 25 de Octubre de 2014]. Disponible en: <http://www.postgresql.org/about/>.
- PRESSMAN, R. S. *Software engineering: a practitioner's approach*. Palgrave Macmillan, 2005. ISBN 007301933X.
- RIEHLE, D. y GROSS, T. Role model based framework design and integration. En *ACM SIGPLAN Notices*. 1998. p. 117-133.
- ROUSE, M. *integrated development environment (IDE) definition* [Consultado el: 24 de Junio de 2015]. Disponible en: <http://searchsoftwarequality.techtarget.com/definition/integrated-development-environment>.
- SALAZAR, C. *Modelo-Vista-Controlador (Model-View-Controller MVC)* [Consultado el: 20 de Marzo de 2015]. Disponible en: <http://www.yiiframework.com/doc/guide/1.1/es/basics.mvc#sec-2>.
- SCHWABER, K. Scrum development process. En *Business Object Design and Implementation*. Springer, 1997, p. 117-134.
- SOMMERVILLE, J. A.; WOODWARD, A. L., *et al.* Action experience alters 3-month-old infants' perception of others' actions. *Cognition*, 2005, vol. 96, nº 1, p. B1-B11. ISSN 0010-0277.
- STUDIO, Z. *The PHP IDE for Smarter Development | Zend Studio* [Consultado el: 26 de Octubre de 2014]. Disponible en: <http://www.zend.com/en/products/studio>.

- SUTHERLAND, J.; VIKTOROV, A., *et al.* Distributed scrum: Agile project management with outsourced development teams. En *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on. 2007.* p. 274a-274a.
- TERDIMAN, D. *OnLive could threaten Xbox, PS3, and Wii* [Consultado el: 19 de Junio de 2015]. Disponible en: <http://www.cnet.com/news/onlive-could-threaten-xbox-ps3-and-wii/>.
- THAMES, J. P. B. *Ingeniería del software asistida por computadora (case)* [Consultado el: 19 de Junio de 2015]. Disponible en: <http://es.slideshare.net/jpbthames/ingeniera-del-software-asistida-por-computadora-case>.
- VENTUREBEAT. *About VentureBeat* [Consultado el: 19 de Junio de 2015]. Disponible en: [http://venturebeat.com/about/?utm\\_source=vbevents&utm\\_medium=clicked-menu&utm\\_campaign=About](http://venturebeat.com/about/?utm_source=vbevents&utm_medium=clicked-menu&utm_campaign=About).
- VIRTUALREHAB. *Acerca de VirtualRehab* [Consultado el: 19 de Junio de 2015]. Disponible en: <http://www.virtualrehab.info/es/>.
- WHO. *Telemedicine. Opportunities and developments in member states. Report on the second global survey on eHealth.* World Health Organization., [Consultado el: 20 de Enero de 2015]. Disponible en: [http://www.who.int/goe/publications/goe\\_telemedicine\\_2010.pdf](http://www.who.int/goe/publications/goe_telemedicine_2010.pdf). ISBN 978 92 4 156414 4
- WINESETT, J. *Agile Web Application Development with Yii 1.1 and PHP5.* 2010, 368 p. Disponible en: <http://fatah.afraid.org/files/books/Agile%20Web%20Application%20Development%20with%20Yii%201.1%20and%20PHP5.pdf>. ISBN 978-1-847199-58-4.
- XUE, Q. y ZHUO, X. W. *The Definitive Guide to Yii 1.1.* 2013, 258 p. Disponible en: <http://www.yiiframework.com/files/yii-guide-1.1.3.pdf>.
- ZELDMAN, J. *Diseño con estándares web* [Consultado el: 20 de Octubre de 2014]. Disponible en: [http://books.google.com.ar/books/about/Diseño\\_con\\_estándares\\_web.html?id=tMJ\\_AAAACAAJ](http://books.google.com.ar/books/about/Diseño_con_estándares_web.html?id=tMJ_AAAACAAJ).

Anexos

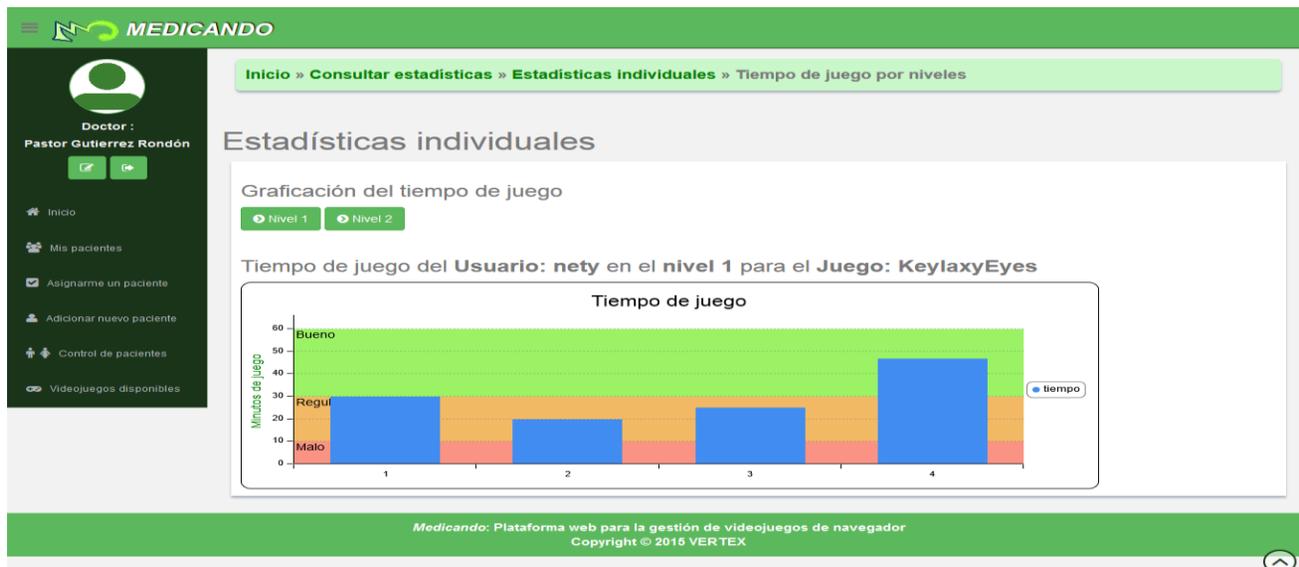


Ilustración 13: Vista de la gráfica de tiempo de duración

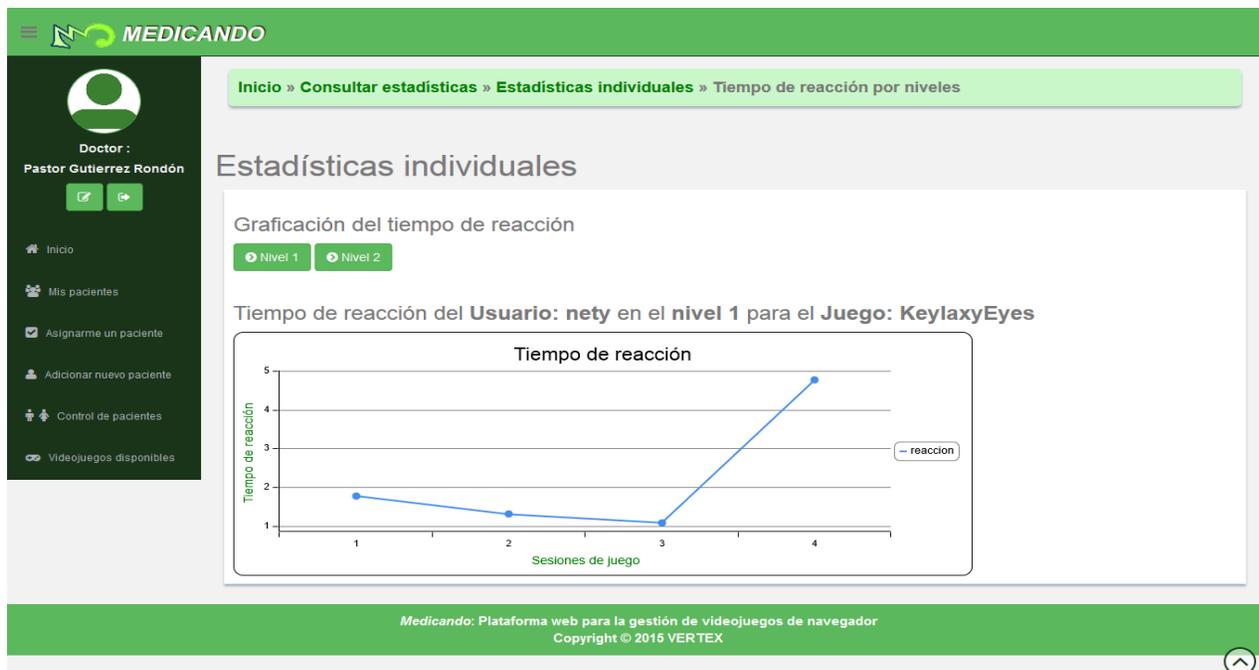


Ilustración 14: Vista de la gráfica de tiempo de reacción

**MEDICANDO**

Inicio » Consultar estadísticas » Estadísticas individuales » Tabla de rendimiento

## Estadísticas Individuales

Estadísticas del rendimiento del **Usuario: nety** en el nivel 1 del Juego: **KeylaxyEyes**

Nivel 1 Nivel 2

No. Sesión	Fecha	Hora	Tareas	Tareas vencidas	Puntuación
1	2015-04-24	10:30:00	40	20	500
2	2015-04-25	12:30:00	40	30	900
3	2015-04-30	13:00:00	40	30	1852
4	2015-05-22	11:00:00	40	18	450

Medicando: Plataforma web para la gestión de videojuegos de navegador  
Copyright © 2015 VERTEX

Ilustración 15: Vista de la tabla de rendimiento individual

**MEDICANDO**

Inicio » Consultar estadísticas » Estadísticas individuales » Matriz del Campo Visual

## Estadísticas individuales

Graficación de la matriz del campo visual

Matriz # 1 - 2015-04-24 Matriz # 2 - 2015-04-25 Matriz # 3 - 2015-04-30 Matriz # 4 - 2015-05-22

Matriz del campo visual del **Usuario: nety** en la fecha **2015-04-24** para el Juego: **KeylaxyEyes**

Objetos de la escena

Encontrados  
No encontrados

Medicando: Plataforma web para la gestión de videojuegos de navegador  
Copyright © 2015 VERTEX

Ilustración 16: Vista de la gráfica para la matriz de campo visual

**MEDICANDO**

Inicio » Consultar estadísticas » Seleccionar estadísticas grupales » Estadísticas Grupales de Scores

### Estadísticas Grupales

Juego KeylaxyEyes

Nivel 1 Nivel 2

Ranking de puntuación para el nivel 1

Pos.	Usuario	Puntuación	Fecha
1	antolin	1900	2015-04-24
2	nety	1852	2015-04-30
3	antolin	1852	2015-04-25
4	nety	900	2015-04-25
5	lety	800	2015-04-25
6	nety	500	2015-04-24
7	nety	450	2015-05-22
8	antolin	450	2015-05-22
9	alain	300	2015-04-26
10	antolin	300	2015-04-23
11	alain	200	2015-04-25

Medicando: Plataforma web para la gestión de videojuegos de navegador  
Copyright © 2015 VERTEX

Ilustración 17: Vista de la tabla de posiciones por puntuación.

**MEDICANDO**

Inicio » Consultar estadísticas » Seleccionar estadísticas grupales » Estadísticas Grupales para el Tiempo de duración

### Estadísticas Grupales

Juego KeylaxyEyes

Nivel 1 Nivel 2

Ranking de tiempo de duración para el nivel 1

Pos.	Usuario	Tiempo de duración(en segundos)	Fecha
1	antolin	1110	2015-04-25
2	nety	1200	2015-04-25
3	alain	1200	2015-04-26
4	antolin	1200	2015-04-24
5	nety	1500	2015-04-30
6	nety	1800	2015-04-24
7	lety	1800	2015-04-25
8	antolin	2000	2015-04-23
9	nety	2820	2015-05-22
10	antolin	2820	2015-05-22
11	alain	3800	2015-04-25

Medicando: Plataforma web para la gestión de videojuegos de navegador  
Copyright © 2015 VERTEX

Ilustración 18: Vista de la tabla de posiciones por tiempo de duración

**MEDICANDO**

Inicio » Consultar estadísticas » Seleccionar estadísticas grupales » Estadísticas Grupales para el Tiempo de Reacción

## Estadísticas Grupales

Juego KeylaxyEyes

Nivel 1 Nivel 2

Ranking de tiempo de reacción para el nivel 2

Pos.	Usuario	Tiempo de reacción	Fecha
1	nety	1.0852	2015-04-25
2	nety	1.0852	2015-04-30
3	antolin	1.0852	2015-04-24
4	antolin	1.0852	2015-04-25
5	nety	1.3333	2015-04-24
6	lety	1.3333	2015-04-25
7	nety	2.1	2015-05-22
8	antolin	2.1	2015-05-22
9	alain	2.3333	2015-04-25
10	antolin	2.3333	2015-04-23

Medicando: Plataforma web para la gestión de videojuegos de navegador  
Copyright © 2015 VERTEX

Ilustración 19: Vista de la tabla de posiciones por tiempo de reacción

## Crear Nuevo Usuario

### Paso 1 de 2

Los campos con (\*) son obligatorios.

Datos de la cuenta

Seleccionar Foto

Nombre(s) \*

Apellidos \*

Nombre de usuario \*

Correo

Clave \*

Generar clave

Siguiente

Ilustración 20: Prototipo de interfaz de usuario Crear Usuario.

Iduser ▼	Nombre de usuario	Correo	Estado de la cuenta	último acceso	
47	reinaluisa	reinaluisa@noemail.local	Cuenta Activada	12 Jun, 2015 10:06:39 pm	 
46	alain	alain@noemail.local	Cuenta Activada		 
45	lety	lety@noemail.local	Cuenta Activada		 
44	nety	nety@estudiantes.uci.cu	Cuenta Activada	12 Jun, 2015 10:06:27 pm	 
43	antolin	antolin@noemail.local	Cuenta Activada		 

Ilustración 21: Prototipo de interfaz de usuario Administrar Usuarios

Los campos con \* son obligatorios.

Nombre \*

Fichero \*

 No se ha seleccionado ningún archivo.
 

Área de aplicación \*

Opción se descarga \*

Descripción

Ilustración 22: Prototipo de interfaz de usuario Crear Juego

Nombre	Descripción	Área de aplicación	.....
KeylaxyEyes	Rehabilitar el campo visual	FUNCIONES VISUALES	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Ilustración 23: Prototipo de interfaz de usuario Gestionar Juego

Datos del tratamiento	Videojuegos disponibles				
<p>Los campos con * son obligatorios.</p> <p>Nombre *</p> <input type="text" value="Trata1"/> <p>Método</p> <input type="text" value="jugar el videojuego dos veces a la semana"/> <p>Descripción</p> <input type="text" value="un tratamiento piloto"/> <p>Evaluación</p> <input type="text"/>	<table border="1"> <thead> <tr> <th>Videojuego</th> <th>Estado</th> </tr> </thead> <tbody> <tr> <td>  <p>o Detalles del juego</p> </td> <td> <p>Asignado</p> </td> </tr> </tbody> </table>	Videojuego	Estado	 <p>o Detalles del juego</p>	<p>Asignado</p>
Videojuego	Estado				
 <p>o Detalles del juego</p>	<p>Asignado</p>				
<p>Guardar</p>					

Ilustración 24: Prototipo de interfaz de usuario Modificar Tratamiento

<b>Nombre</b>	Trata1
<b>Método</b>	jugar el videojuego dos veces a la semana
<b>Descripción</b>	un tratamiento piloto
<b>Evaluación</b>	

**Videojuego(s) asociado(s) al tratamiento:**



o Ver juego

Ilustración 25: Prototipo de interfaz de usuario Detalles del Tratamiento

Tabla 45: Tarea de ingeniería 2 para HU 2

Tarea de ingeniería	
<b>Número de Tarea:</b> 2	<b>Número de HU:</b> 2
<b>Nombre de la Tarea:</b> Modificar usuario	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha de Inicio:</b> 23/3/2015	<b>Fecha de Fin:</b> 27/3/2015
<b>Programador responsable:</b> Ernesto Gutierrez Ramos	
<b>Descripción:</b> Se debe garantizar que la aplicación pueda modificar usuarios registrados en el sistema. Se pueden modificar todos los datos asociados a un usuario.	

Tabla 46: Tarea de ingeniería 3 para HU 2

Tarea de ingeniería	
<b>Número de Tarea:</b> 3	<b>Número de HU:</b> 2
<b>Nombre de la Tarea:</b> Eliminar usuario	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha de Inicio:</b> 23/3/2015	<b>Fecha de Fin:</b> 27/3/2015
<b>Programador responsable:</b> Ernesto Gutierrez Ramos	
<b>Descripción:</b> Se debe poder eliminar usuarios registrados en el sistema a partir de una acción sobre un botón. Antes de eliminar, se debe lanzar un mensaje de confirmación brindando la opción de cancelar o aceptar.	

Tabla 47: Tarea de ingeniería 2 para HU 3

Tarea de ingeniería	
<b>Número de Tarea:</b> 2	<b>Número de HU:</b> 3
<b>Nombre de la Tarea:</b> Modificar videojuego	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1

<b>Fecha de Inicio:</b> 30/3/2015	<b>Fecha de Fin:</b> 3/4/2015
<b>Programador responsable:</b> Ernesto Gutierrez Ramos	
<b>Descripción:</b> Se debe permitir la modificación de los datos un videojuego que se haya cargado en la plataforma. Todos los datos asociados a un videojuego pueden ser actualizados.	

*Tabla 48: Tarea de ingeniería 3 para HU 3*

Tarea de ingeniería	
<b>Número de Tarea:</b> 3	<b>Número de HU:</b> 3
<b>Nombre de la Tarea:</b> Eliminar videojuego	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha de Inicio:</b> 30/3/2015	<b>Fecha de Fin:</b> 3/4/2015
<b>Programador responsable:</b> Ernesto Gutierrez Ramos	
<b>Descripción:</b> Se debe poder eliminar videojuegos registrados en el sistema a partir de una acción sobre un botón. Antes de eliminar se debe lanzar un mensaje de confirmación brindando la opción de cancelar o aceptar.	

*Tabla 49: Tarea de ingeniería 4 para HU 3*

Tarea de ingeniería	
<b>Número de Tarea:</b> 4	<b>Número de HU:</b> 3
<b>Nombre de la Tarea:</b> Ver detalles de videojuego	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha de Inicio:</b> 30/3/2015	<b>Fecha de Fin:</b> 3/4/2015
<b>Programador responsable:</b> Ernesto Gutierrez Ramos	
<b>Descripción:</b> Se debe brindar una vista de los detalles del videojuego que incluya, nombre, descripción, imagen, un botón de jugar (para iniciar el juego <i>online</i> ), un botón para descargar (en caso de que esté autorizada la descarga) y el área de aplicación.	

Tabla 50: Tarea de ingeniería 2 para HU 5

Tarea de ingeniería	
<b>Número de Tarea:</b> 2	<b>Número de HU:</b> 5
<b>Nombre de la Tarea:</b> Modificar tratamiento	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha de Inicio:</b> 13/4/2015	<b>Fecha de Fin:</b> 17/4/2015
<b>Programador responsable:</b> Ernesto Gutierrez Ramos	
<b>Descripción:</b> Se debe permitir la modificación de los datos un tratamiento almacenado en el sistema. Todos los datos asociados a un tratamiento pueden ser actualizados por los doctores.	

Tabla 51: Tarea de ingeniería 3 para HU 5

Tarea de ingeniería	
<b>Número de Tarea:</b> 3	<b>Número de HU:</b> 5
<b>Nombre de la Tarea:</b> Eliminar tratamiento	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha de Inicio:</b> 13/4/2015	<b>Fecha de Fin:</b> 17/4/2015
<b>Programador responsable:</b> Ernesto Gutierrez Ramos	
<b>Descripción:</b> Los doctores puedan eliminar tratamientos registrados en el sistema a partir de una acción sobre un botón. Antes de eliminar se debe lanzar un mensaje de confirmación brindando la opción de cancelar o aceptar.	

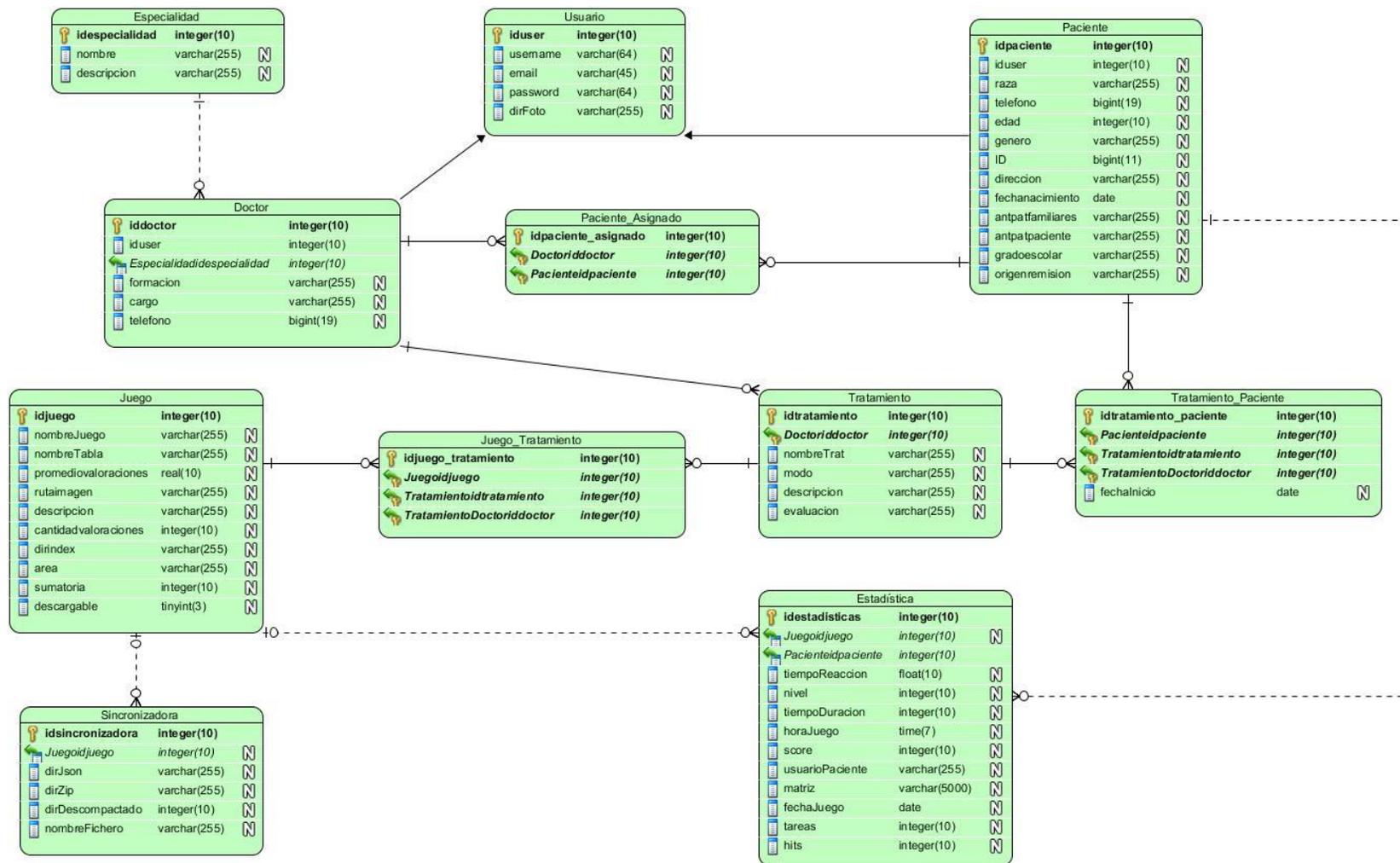


Ilustración 26: Diagrama Entidad-Relación del sistema.

## ***Glosario de términos***

---

<sup>1</sup> La Tele-rehabilitación es el uso y práctica de servicios para la rehabilitación a través de las tecnologías de la información y las comunicaciones.

<sup>2</sup> Los juegos serios (del inglés "*serious game*"), son juegos diseñados para un propósito principal distinto del de la pura diversión. Normalmente, el adjetivo "serio" pretende referirse a productos utilizados por industrias como la de defensa, educación, exploración científica, sanitaria, urgencias, planificación cívica, ingeniería, religión y política.

<sup>3</sup> En informática, modelo CRUD (*Create, Read, Update and Delete*, por sus siglas en inglés), se refiere las funciones básicas de Insertar, Leer, Actualizar y Eliminar información de bases de datos.

<sup>4</sup> Juego en la nube, también llamado juego bajo demanda, es un tipo de juego en línea que permite la transmisión directa y bajo demanda de juegos en un ordenador mediante el uso de un cliente ligero, donde se almacena el juego real en el servidor de la compañía del juego y es transmitido directamente a las computadoras con acceso al servidor a través del cliente.

<sup>5</sup> El *streaming* es la distribución digital de multimedia a través de una red de computadoras, de manera que el usuario consume el producto (generalmente archivo de video o audio) en paralelo mientras se descarga.

<sup>6</sup> La neurorrehabilitación es una disciplina médica integrada por un equipo interdisciplinario (médicos neurólogos, fisiatras, terapistas físicos, del lenguaje, psicopedagogos, entre otros) con una visión de tratamiento ecléctico de acuerdo a los diferentes métodos clásicos y al ejercicio físico terapéutico, siendo este último el medio fundamental en el proceso de neurorrehabilitación como el mayor estimulador del sistema nervioso.

<sup>7</sup> En informática, ZIP o zip es un formato de compresión sin pérdida, muy utilizado para la compresión de datos como documentos, imágenes o programas. Para este tipo de archivos se utiliza generalmente la extensión ".zip".

<sup>8</sup> El lenguaje de consulta estructurado o SQL (por sus siglas en inglés *Structured Query Language*) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como hacer cambios en ellas.

<sup>9</sup> JSON, acrónimo de *JavaScript Object Notation*, es un formato ligero para el intercambio de datos. Es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

<sup>10</sup> La sincronización es el proceso por el cual se transfiere cambios de datos de un equipo a un servidor.

<sup>11</sup> KeylaxyEyes es un videojuego para la rehabilitación de pacientes jóvenes con trastornos del campo visual, desarrollado en el Centro VERTEX como tesis de pregrado.

<sup>12</sup> CRUGE es una extensión para gestión de usuarios y control de acceso basado en roles para el *framework* Yii. Permite administrar y controlar de manera segura los usuarios y los roles que ellos deban tener en una aplicación *web*.