



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
ENTORNOS INTERACTIVOS 3D VERTEX, FACULTAD 5

# MÓDULOS DE VISUALIZACIÓN DE LA INFORMACIÓN PACIENTE-ESPECIALISTA PARA LA PLATAFORMA DE GESTIÓN DE VIDEOJUEGOS MEDICANDO

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

Autor: Juan Gabriel Valdés Díaz

Tutores: Ing. Reinaldo Garcia Maturell

Ing. Yadira Ramírez Rodríguez

**La Habana, 2016**

*« Tu tiempo es limitado, de modo que no lo malgastes viviendo la vida de alguien distinto. No quedes atrapado en el dogma, que es vivir como otros piensan que deberías vivir. No dejes que los ruidos de las opiniones de los demás acallen tu propia voz interior. Y, lo que es más importante, ten el coraje para hacer lo que te dice tu corazón y tu intuición »*

*Steve Jobs*

*« Siempre me siento feliz. ¿Sabes por qué? Porque no espero nada de nadie; esperar siempre duele. Los problemas no son eternos, siempre tienen solución. Lo único que no se resuelve es la muerte. La vida es corta, por eso ámala, se feliz y siempre sonríe, solo vive intensamente. Antes de hablar, escucha. Antes de escribir, piensa. Antes de herir, siente. Antes de rendirte, intenta. Antes de morir, vive »*

*William Shakespeare*

---

## Dedicatoria

---

*El esfuerzo y compromiso que he dedicado al transitar por esta casa de altos estudios, lo dedico a mi familia por tener tanta confianza en mi persona y en especial a mi madre Marelys, quien es mi lucero guía en este viaje por el sendero de los vivos.*

---

## Agradecimientos

---

*Gracias amigos míos por todo su apoyo y sus críticas, no voy a mencionar nombres porque alguno se quedará y surgirán celos. Existe un grupo que siempre estará en mi corazón, saben que son especiales y tengan claro que toda idea que se hagan de cuánto los aprecio, les quedará corta, para que lo sepan.*

---

## Declaración de autoría

---

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

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

---

Juan Gabriel Valdés Díaz  
Autor

---

Ing. Reinaldo Garcia Maturell  
Tutor

---

Ing. Yadira Ramírez Rodríguez  
Tutora

El [Centro de Entornos Interactivos 3D \(VERTEX\)](#) de la [Universidad de las Ciencias Informáticas \(UCI\)](#) ha incursionado de forma reciente 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. Breve tiempo atrás, este Centro desarrolló la plataforma Medicando; una aplicación web que permite a los doctores gestionar videojuegos terapéuticos y generar estadísticas gráficas para analizar la evolución de los pacientes a partir de la información almacenada en el sistema, por cada sesión de juego. Aunque cuenta con un grupo de funcionalidades útiles para el fin con que fue concebida, aún existen deficiencias que necesitan ser atendidas, así como dar los primeros pasos en la modularidad para la herramienta. En el presente trabajo se plantea como objetivo, desarrollar módulos de visualización de la información paciente-especialista, para mejorar la retroalimentación en la plataforma. Para llevar a cabo la construcción de los módulos Notificaciones, Gráficas y Reportes fue seleccionado como marco de trabajo el *framework* Yii y como [Sistema Gestor de Base de Datos \(SGBD\)](#), MySQL. Para guiar el desarrollo desde el análisis hasta las pruebas, se siguen los pasos que propone la metodología de desarrollo ágil [Extreme Programing \(XP\)](#).

**Palabras clave:** gestión de tratamientos, plataforma web, videojuegos terapéuticos, visualización gráfica.

<b>Introducción</b>	<b>1</b>
<b>1 Fundamentación Teórica</b>	<b>4</b>
1.1 Introducción al capítulo . . . . .	4
1.2 Plataforma virtual . . . . .	4
1.3 Medicando . . . . .	5
1.4 Análisis de plataformas virtuales para la gestión de videojuegos . . . . .	5
1.5 Notificación . . . . .	6
1.5.1 Observación del funcionamiento de mecanismos de notificaciones en aplicaciones web . . . . .	7
1.5.2 Selección de tecnologías para el envío de notificaciones . . . . .	8
1.6 Graficación . . . . .	10
1.6.1 Tipos de gráficas y su utilización . . . . .	10
1.6.2 Selección de tecnología para representar información . . . . .	13
1.7 Reportes . . . . .	14
1.7.1 Observación del comportamiento de mecanismos de reportes en aplicaciones web . . . . .	15
1.8 Herramientas, tecnología y metodología . . . . .	16
1.9 Conclusiones del capítulo . . . . .	18
<b>2 Características y diseño del sistema</b>	<b>19</b>
2.1 Introducción al capítulo . . . . .	19
2.2 Propuesta de solución . . . . .	19
2.3 Fase I: Planificación . . . . .	20
2.3.1 Historias de Usuarios . . . . .	20
2.3.2 Estimación de esfuerzo por Historia de Usuario . . . . .	26
2.3.3 Desarrollo del plan de iteraciones . . . . .	26
2.3.4 Plan de duración de las iteraciones . . . . .	27
2.3.5 Plan de entregas . . . . .	27
2.4 Fase II: Diseño del sistema . . . . .	28
2.5 Patrones de diseño . . . . .	29

2.6	Tarjetas CRC	30
2.7	Conclusiones del capítulo	36
<b>3</b>	<b>Implementación y prueba del sistema</b>	<b>37</b>
3.1	Introducción al capítulo	37
3.2	Fase III: Desarrollo	37
3.2.1	Tareas de ingeniería	37
3.3	Fase IV: Pruebas	44
3.3.1	Pruebas de aceptación	44
3.3.2	Pruebas de aceptación para la Iteración I	44
3.3.3	Pruebas de aceptación para la Iteración II	46
3.3.4	Pruebas de aceptación para la Iteración III	48
3.3.5	Análisis de las pruebas de aceptación	49
3.4	Conclusiones del capítulo	51
	<b>Conclusiones</b>	<b>52</b>
	<b>Recomendaciones</b>	<b>53</b>
	<b>Referencias bibliográficas</b>	<b>54</b>
	<b>Anexos</b>	<b>57</b>
.1	Diagnóstico de la plataforma Medicando	58
.1.1	Principales funcionalidades y comportamiento actual	58
.1.2	Análisis constructivo de funcionalidades	58
.1.3	Resultados del análisis de las funcionalidades	59
.2	Tareas de ingeniería asociadas a Historias de Usuarios	60
.2.1	Tareas de ingeniería para la Iteración I	60
.2.2	Tareas de ingeniería para la Iteración II	61
.3	Pruebas de aceptación	62
.3.1	Pruebas de aceptación para la Iteración I	62
.3.2	Pruebas de aceptación para la Iteración II	66
.3.3	Pruebas de aceptación para la Iteración III	68

---

## Índice de figuras

---

1.1	Gráficos de barras vertical y horizontal . . . . .	11
1.2	Gráfico de líneas . . . . .	11
1.3	Gráfico de sectores . . . . .	12
1.4	Gráfico de dispersión . . . . .	12
2.1	Representación arquitectónica del sistema . . . . .	29
2.2	Prototipo de tarjeta CRC . . . . .	30
3.1	Resultados de las pruebas de aceptación . . . . .	50

---

## Índice de tablas

---

1.1	Roles del sistema	5
1.2	Comparativa de tecnologías para el envío de notificaciones en el sistema. Fuente: Elaboración propia	9
1.3	Comparativa de tecnologías para la representación visual de la información. Fuente: Elaboración propia	14
2.1	Historia de Usuario # 1	20
2.2	Historia de Usuario # 2	21
2.3	Historia de Usuario # 3	22
2.4	Historia de Usuario # 4	22
2.5	Historia de Usuario # 5	23
2.6	Historia de Usuario # 6	24
2.7	Historia de Usuario # 7	24
2.8	Historia de Usuario # 8	25
2.9	Historia de Usuario # 9	25
2.10	Estimación de esfuerzo por Historia de Usuario	27
2.11	Plan de duración de las iteraciones	27
2.12	Plan de entrega de versiones	28
2.13	Tarjeta CRC # 1	30
2.14	Tarjeta CRC # 2	31
2.15	Tarjeta CRC # 3	31
2.16	Tarjeta CRC # 4	31
2.17	Tarjeta CRC # 5	32
2.18	Tarjeta CRC # 6	32
2.19	Tarjeta CRC # 7	32
2.20	Tarjeta CRC # 8	34
2.21	Tarjeta CRC # 9	35
3.1	Tarea de ingeniería # 1	38
3.2	Tarea de ingeniería # 2	38
3.3	Tarea de ingeniería # 3	38

3.4	Tarea de ingeniería # 4	39
3.5	Tarea de ingeniería # 5	39
3.6	Tarea de ingeniería # 6	40
3.7	Tarea de ingeniería # 7	40
3.8	Tarea de ingeniería # 8	40
3.9	Tarea de ingeniería # 9	41
3.10	Tarea de ingeniería # 10	41
3.11	Tarea de ingeniería # 11	42
3.12	Tarea de ingeniería # 12	42
3.13	Tarea de ingeniería # 13	43
3.14	Tarea de ingeniería # 14	43
3.15	Prueba de aceptación # 1	44
3.16	Prueba de aceptación # 2	45
3.17	Prueba de aceptación # 3	45
3.18	Prueba de aceptación # 4	46
3.19	Prueba de aceptación # 5	47
3.20	Prueba de aceptación # 6	48
3.21	Prueba de aceptación # 7	48
3.22	Prueba de aceptación # 8	49
23	Tarea de ingeniería # 15	60
24	Tarea de ingeniería # 16	60
25	Tarea de ingeniería # 17	61
26	Tarea de ingeniería # 18	61
27	Tarea de ingeniería # 19	62
28	Tarea de ingeniería # 20	62
29	Prueba de aceptación # 9	62
30	Prueba de aceptación # 10	63
31	Prueba de aceptación # 11	63
32	Prueba de aceptación # 12	64
33	Prueba de aceptación # 13	64
34	Prueba de aceptación # 14	65
35	Prueba de aceptación # 15	65
36	Prueba de aceptación # 16	66
37	Prueba de aceptación # 17	67
38	Prueba de aceptación # 18	68
39	Prueba de aceptación # 19	68
40	Prueba de aceptación # 20	69

Cuba en los últimos años ha estado inmersa en el desarrollo de *software* para informatizar la sociedad, en especial el área de la medicina. Desde 1997 se concibe una primera estrategia de informatización como respuesta del sector de la salud a los lineamientos estratégicos para la informatización de la sociedad cubana, con la finalidad de coordinar esfuerzos para el desarrollo de este proceso en el Sistema Nacional de Salud (RAMOS y LEDO, 2006). El eje fundamental y centro del proceso de informatización del sector lo constituye el paciente, quién será el principal beneficiado al garantizar las aplicaciones, la calidad, oportunidad y consistencia de la información.

La utilización de videojuegos como herramientas de trabajo para los doctores ha tomado un creciente protagonismo enmarcado en la Telemedicina, pues brinda la posibilidad de mejorar la calidad de atención al cliente, para ello aprovecha la utilización de las [Tecnologías de la Información y las Comunicaciones \(TIC\)](#). En aras de apoyar este proyecto el Centro de Entornos Interactivos 3D (VERTEX), perteneciente a la UCI, ha desarrollado un grupo de videojuegos terapéuticos para favorecer los procesos de rehabilitación.

VERTEX desarrolló hace poco tiempo la plataforma Medicando, este producto cuenta con un grupo de funcionalidades como son: la administración de usuarios y el establecimiento de roles a estos, lo que permite definir control de acceso y niveles de seguridad. Además ofrece facilidades para la gestión de videojuegos, doctores, pacientes y tratamientos, así como la sincronización de los datos almacenados por los videojuegos y el control de las estadísticas. Esta herramienta permite proporcionar al personal de salud, un medio para llevar un control de la evolución de los pacientes en sus tratamientos con videojuegos. Para ello la plataforma genera una serie de estadísticas gráficas y tabulares a partir de la información almacenada por los videojuegos durante las sesiones de rehabilitación. Desde un diagnóstico inicial basado en las funcionalidades que se realizó a Medicando, se detectan ciertas deficiencias que necesitan ser atendidas:

- El servicio de graficación se encuentra condicionado para las variables que manejan los videojuegos que se gestionan en el sistema, por lo que no se ofrece personalización para la representación dinámica de nuevas variables. Además, los datos a representar corresponden a campos predeterminados según el tipo de gráfico que se seleccione.
- Los especialistas no tienen la posibilidad de ser informados cuando un paciente suyo ha iniciado o terminado un tratamiento, estos tienen que chequear de manera constante si existen actualizaciones en este aspecto. Además, no se cuenta con un mecanismo que informe a los pacientes que se les ha asignado un nuevo tratamiento o de recomendaciones que les pueda dar un doctor con respecto al tratamiento que realiza.

- Los datos que consultan los especialistas son accesibles solo en la plataforma, atentando así contra la disponibilidad de la información. Además no se cuenta con un mecanismo que genere informes especializados sobre la información que se maneja en Medicando de manera que pueda ser utilizada en análisis posteriores.

A partir de la situación antes descrita se plantea como **problema de la investigación**: ¿Cómo contribuir a la retroalimentación entre el paciente-especialista en la plataforma Medicando?

Teniendo como **objeto de estudio**: la visualización de la información paciente-especialista en plataformas web para la Telemedicina<sup>1</sup>.

Se define como **objetivo general**: desarrollar módulos de visualización de la información paciente-especialista para aumentar la retroalimentación en la plataforma de gestión de videojuegos Medicando; enmarcado en el **campo de acción**: la visualización de la información paciente-especialista en la plataforma Medicando.

Para dar cumplimiento al objetivo planteado se trazan las siguientes tareas de investigación:

- Realización de diagnóstico a la plataforma Medicando para comprender el entorno en que se enmarca la investigación.
- Elaboración del marco teórico de la investigación a través de un estudio del estado del arte de las plataformas web atendiendo a la gestión de gráficas, reportes y notificaciones.
- Selección de las tecnologías y herramientas que se ajusten a las necesidades de la plataforma.
- Realización del levantamiento de las funcionalidades relacionados con los módulos de graficación, reportes y notificaciones de la plataforma Medicando.
- Implementación de la solución propuesta.
- Realización de pruebas de aceptación para comprobar el cumplimiento de los requerimientos definidos.

Para el desarrollo de la investigación se emplean los siguientes **métodos científicos**:

### Métodos teóricos:

- **Método lógico de soporte-sistémico**: se utiliza para realizar un estudio de la plataforma en su totalidad, lo que permite obtener una mejor comprensión de la complejidad y dinámica funcional propia.
- **Método analítico-sintético**: para analizar desde diferentes aristas los conceptos asociados a la visualización de la información en plataformas virtuales y sintetizar los datos recopilados. Este procedimiento permite describir las características generales y establecer las principales formas de retroalimentación entre usuarios.

---

<sup>1</sup>Suministro de servicios de atención sanitaria empleando las TIC.

**Métodos empíricos:**

- El método **observación científica** se emplea con el objetivo de observar el funcionamiento de algunas plataformas web orientadas a la generación de estadísticas y reportes; así como mecanismos de notificaciones que se utilizan en estas. Este método proporciona la vía para realizar un registro visual de las características comunes en este tipo de sistemas e identificar las que puedan formar parte de la solución.

El presente trabajo de diploma, está compuesto por tres capítulos, que incluyen los procedimientos desarrollados en relación con el trabajo investigativo, así como la propuesta de solución y validación de la investigación.

- **Capítulo 1 - Fundamentación Teórica:** en esta primera etapa se aborda el estado del arte del tema que se investiga. Se destaca un estudio bibliográfico detallado sobre los principales conceptos asociados a los mecanismos de notificaciones, reportes y gráficas que utilizan las plataformas virtuales que se encuentran en la red de redes. En el transcurso de esta etapa, mientras se realiza el análisis del objeto de investigación, se seleccionan las tecnologías de desarrollo acordes para incorporar estas prestaciones en la plataforma Medicando.
- **Capítulo 2 - Características y diseño del sistema:** se describe el procedimiento seguido en las etapas de planificación y diseño que propone la metodología **XP**. Se especifican las **Historias de Usuarios (HU)**, el plan de iteraciones y el plan de entregas. También se describe la propuesta de solución y se mencionan los patrones de diseño utilizados en la implementación de los módulos.
- **Capítulo 3 - Implementación y prueba del sistema:** en esta etapa se definen las tareas de ingenierías correspondientes a cada **HU**. También se aplican las pruebas de aceptación que permiten comprobar el correcto desarrollo de las funcionalidades implementadas.

### 1.1. Introducción al capítulo

En este capítulo se sintetiza la búsqueda y análisis de la información relacionada con el dominio del problema. Son descritos los conceptos fundamentales relacionados con la investigación y analizadas algunas soluciones existentes referentes al mismo entorno. Aborda además, la selección y descripción de las principales tecnologías y herramientas que serán utilizadas para llevar a cabo la implementación de la solución a desarrollar.

### 1.2. Plataforma virtual

Una plataforma para el desarrollo de aplicaciones electrónicas, es un sistema informático base, sobre el cual se ejecutan o realizan un conjunto de servicios que este sistema puede ofrecer (ROJAS y MONTIEL, 2012). También se conoce como plataforma, a un sistema que sirve como base para hacer funcionar determinados módulos de *hardware* o de *software* con los que es compatible. Dicho sistema, está definido por un estándar alrededor del cual se determina una arquitectura de *hardware* y una plataforma de *software* (incluyendo entornos de aplicaciones). Al definir plataformas se establecen los tipos de arquitectura, sistema operativo, lenguajes de programación o interfaz de usuario compatibles (APRENDIZAJE, 2015).

Se puede entender plataforma virtual, teniendo en cuenta las definiciones anteriores, como un sistema que permite la ejecución de varias aplicaciones bajo un mismo entorno, acorde a las restricciones tecnológicas que se encuentren definidas para la compatibilidad. Los usuarios solo necesitan de una conexión a la web para acceder a este tipo de plataforma y así poder utilizar los servicios provistos, sin la necesidad de encontrarse en un espacio físico determinado. En correspondencia al entorno en que se enmarca la presente investigación, se toma como referencia la plataforma Medicando.

### 1.3. Medicando

Es una plataforma web desarrollada con el objetivo de facilitar el seguimiento de pacientes en los tratamientos que les son asignados en el sistema, donde cada tratamiento debe estar relacionado con un videojuego de rehabilitación. Esta herramienta fue desarrollada con tecnología [HyperText Preprocessor \(PHP\)](#), para ello se utilizó como marco de trabajo el *framework* Yii; mientras que la gestión de los datos es administrada utilizando el [SGBD MySQL](#) y como servidor *Apache Server*.

Los usuarios son identificados por la plataforma en correspondencia con el rol que desempeñen, estos se asignan de acuerdo a la interacción que tendrá cada usuario con el sistema. Los roles que se definen son Paciente, Doctor y Administrador; donde cada rol puede ejecutar un grupo de operaciones específicas (GUTIERREZ, 2015). La siguiente tabla describe las principales funcionalidades que son accesibles por los diferentes roles.

Tabla 1.1. Roles del sistema

Roles	Interacción en Medicando
<b>Paciente</b>	Asignado a los usuarios registrados en la aplicación que recibirán tratamientos con videojuegos. Estos pueden consultar los tratamientos que tienen asignados y acceder a la vez al videojuego asociado. También tienen la posibilidad de subir al sistema la salva de las sesiones de rehabilitación realizadas sin conexión.
<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 de la herramienta. 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.

Para detectar las nuevas funcionalidades a desarrollar, se llevó a cabo un diagnóstico a la plataforma que permitió identificar las potencialidades y deficiencias con que cuenta el sistema. Los resultados de este análisis se pueden consultar en el anexo 1. Diagnóstico de la plataforma Medicando; estos permitieron detectar las necesidades a cubrir en la presente investigación que se pueden agrupar en cuanto a las notificaciones del sistema, generación de reportes y visualización de estadísticas almacenadas por las sesiones de rehabilitación. Con el objetivo de observar el comportamiento de módulos que atienden las insuficiencias detectadas se observan diferentes plataformas orientadas a la Telemedicina.

### 1.4. Análisis de plataformas virtuales para la gestión de videojuegos

En la actualidad se pueden encontrar diferentes plataformas para la gestión de videojuegos terapéuticos. En la investigación se consultaron un grupo de ellas para analizar sus características en correspondencia con

las notificaciones que se generen, las formas de visualización de las estadísticas y la generación de reportes:

- **VirtualRehab**: es un producto clínicamente validado para la rehabilitación de pacientes con algún grado de discapacidad física, que combina modernas técnicas de captura de movimientos con tecnología de videojuegos. Es el primer producto de rehabilitación virtual que utiliza dispositivos como el Kinect de Microsoft. Permite al terapeuta, llevar un seguimiento individualizado por cada paciente a partir de la evaluación y análisis gráfico de los progresos. También cuenta con un sistema de alertas y notificaciones. (VIRTUALREHAB.INFO, 2015)
- **NeuroAtHome**: plataforma multidisciplinar de neurorrehabilitación<sup>1</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, monitoree y gestione los planes de tratamiento personalizados para sus pacientes (NEUROATHOME.ORG, 2015).

Después de haber analizado algunas plataformas virtuales que basan su negocio en la gestión de videojuegos terapéuticos, se obtuvieron evidencias del uso de las funcionalidades que se pretendían estudiar. Sin embargo, no es posible acceder a estos sistemas por tener un carácter propietario, por lo que no se puede comprobar de forma empírica cómo se gestionan las características deseadas. De acuerdo a esta situación, se decide realizar un estudio individualizado basado en los grupos funcionales previamente mencionados en la sección 1.3, para lo que se comenzará con el uso de notificaciones en diferentes sistemas.

## 1.5. Notificación

La comunicación e interacción son elementos fundamentales para la socialización de las comunidades, este es un aspecto que cobra protagonismo en la web. Ambas hacen posible la transmisión de conocimientos e información, de forma sencilla y fácil de entender. Una notificación se genera cuando el estado compartido de una aplicación cambia, es decir, si un usuario pone en conocimiento a los demás usuarios de un evento específico o tema en común (ESCOBAR; URANGO; CASTRO et al., 2015).

La Real Academia Española define:

- **Notificar**: dar extrajudicialmente, con propósito cierto, noticia de algo. Comunicar formalmente a su destinatario una resolución administrativa o judicial.
- **Notificación**: Acción y efecto de notificar. Documento en que consta la resolución comunicada.

Teniendo en cuenta los autores anteriores, se puede definir como notificación electrónica a la acción de comunicar a un usuario, o grupo de usuarios, sobre un cambio, acción o evento que ha ocurrido en el sistema. Donde el envío del mensaje se realiza a partir de la utilización de las TIC. También se puede inferir como sistema de notificaciones a la combinación de servicios e infraestructura web que provee un mecanismo para

---

<sup>1</sup>Disciplina médica orientada a estimular el sistema nervioso.

la entrega de un mensaje de notificación, a uno o varios receptores. Existen varios medios para comunicar el mensaje, entre los más utilizados se encuentra la vía correo electrónico, SMS y alertas en la propia aplicación web.

### 1.5.1. Observación del funcionamiento de mecanismos de notificaciones en aplicaciones web

Con el objetivo de identificar las principales formas de notificación en plataformas orientadas a la Telemedicina, se realiza un estudio de diferentes sistemas informáticos que utilizan mecanismos de notificación. Para ello se observaron algunas funcionalidades que presentan determinados Sistemas de Información Hospitalaria que se utilizan en varios países del mundo.

- **Michigan:** es un Sistema de Información Hospitalaria desarrollado por la empresa Michigan Ingeniería Informática S.A. En algunos centros de salud de Argentina se cuenta con esta herramienta para la gestión hospitalaria a partir de un grupo de módulos. Para el control de las llegadas tarde y las ausencias del personal médico se utiliza el módulo Control de Accesos y Horarios, este permite la generación y envío de correos electrónicos automáticos a los destinatarios necesarios con la información pertinente. Además posibilita la remisión de correos electrónicos a modo de notificaciones a trabajadores y pacientes (MICHIGANSA, 2016).
- **Sistema de Información Hospitalaria alas HIS:** es un sistema desarrollado por el Centro de Informática Médica (CESIM) de la UCI, con el objetivo de gestionar los datos de los diferentes procesos de una institución hospitalaria. Este sistema cuenta con varios módulos que interconectan en la aplicación las distintas áreas de un centro de salud tales como: Banco de Sangre, Epidemiología, Anatomía Patológica, Bloque Quirúrgico, Emergencia, Hospitalización, Consulta Externa, entre otros (HERNÁNDEZ y DARIAS, 2013). Además cuenta con un componente de notificaciones que utiliza el envío de correos electrónicos, mensajería beeper y SMS. Este componente permite extender los espacios de intercambio entre los usuarios del sistema, así como mantener una activa retroalimentación entre los médicos y los pacientes.
- **Sistema de Recordatorio SMS:** es un *software* que se utiliza en los centros hospitalarios del Instituto Catalán de la Salud (ICS), en España. Esta aplicación fue desarrollada por Esendex, empresa líder en el mercado europeo de mensajes SMS. Los usuarios reciben un mensaje de texto al móvil con información sobre la fecha, hora y lugar de su consulta; así como un número de teléfono de contacto para cualquier duda o cambio de cita. Todo ello, a partir del servicio que se ofrece mediante recordatorios de forma automatizada desde una plataforma compartida por los hospitales del ICS. Por otra parte, los usuarios del sistema de salud son informados de cuándo pueden ir a recoger los resultados de distintas pruebas sanitarias. Se potencia además la notificación vía correo electrónico en la red de atención primaria de estos centros. Aunque se encuentra establecido un sistema de precios para los envíos de SMS, las aplicaciones de Esendex son de uso gratuito, por lo que no se tiene que pagar por derechos de licencia para su empleo (BIRD y HUCKER, 2012).
- **Sistema de Información Clínico-Hospitalaria del Hospital Padua, Italia:** este es un sistema que

gestiona todas las peticiones realizadas por los médicos del hospital Pauda y reúne la información procedente de todos los servicios de diagnóstico, incluyendo el laboratorio clínico y los departamentos de imagen. Se notifica a los especialistas en tiempo real, de forma automática, con información sobre los resultados del laboratorio mediante un mensaje de correo electrónico. Para el envío de estos mensajes, posee un componente computarizado de notificaciones a partir del envío de alertas y mensajes SMS, que fue desarrollado por la Sociedad de Procesamiento de Datos Médicos (GMD, por sus siglas en alemán). Además de los resultados, de forma independiente se generan dos mensajes: un SMS hacia el teléfono móvil del médico de turno y otro a nivel de departamento, que llega al monitor del clínico que ordenó el análisis. Ambos tipos de mensaje incluyen los códigos apropiados para la identificación del paciente y especifican el número de teléfono móvil del médico de guardia en el laboratorio. El éxito o el fracaso de cada proceso de notificación se lleva en constancia por el sistema de información de la institución (PIVA; SCIACOVELLI; ZANINOTTO; LAPOSATA y PLEBANI, 2009).

Con el estudio realizado se observa que la mayoría de los sistemas analizados no brindan información suficiente sobre las herramientas con las que fueron desarrollados, por lo que resulta difícil decidir si se ajustan o no, a las necesidades reales de la plataforma Medicando. Algunos son desarrollados con una combinación de tecnologías libres y privativas, lo que dificulta obtener versiones funcionales de los mismos para su estudio. Muchos de estos sistemas notifican sólo por la vía del teléfono celular mientras que, la forma más utilizada es el envío de correos electrónicos. Teniendo en cuenta las características de Medicando, la forma de notificación a emplear será mediante el envío de correos electrónicos y mensajes de alerta a través del propio sistema.

### 1.5.2. Selección de tecnologías para el envío de notificaciones

Atendiendo a la necesidad de establecer un sistema de notificaciones en la aplicación Medicando, se consideran dos tipos fundamentales de notificaciones según la vía que se utilice para la entrega del mensaje: notificaciones a nivel de correo y notificaciones a nivel de sistema. A continuación se realiza un estudio de las tecnologías en correspondencia con el tipo de notificación:

#### A nivel de correo

La plataforma cuenta con un sistema de autenticación basado en roles que es controlado con la utilización del módulo CRUGE (GUTIERREZ, 2015). Este módulo permite el uso de la extensión *Crugemailer*, la cual proporciona las facilidades necesarias para el envío de correos electrónicos basados en esquema de vistas y patrones (BELISARIO, 2012). Teniendo en cuenta estas características se hará uso de la misma para el aviso de las acciones que se realizan en la plataforma.

## A nivel de sistema

Se considera la información de bibliotecas *javascript* de las que se toman como referencia cinco seleccionadas según (JQUERYHOUSE.COM, 2015). Se tienen en cuenta los siguientes parámetros para establecer la comparación que se aprecia en la Tabla 1.2. Los criterios son ponderados de 1 a 10 según el nivel de correspondencia con el parámetro, estos se midieron a partir de la utilización de cada una de las bibliotecas en un caso de prueba o demo desarrollado:

- **Facilidad de uso:** indica el nivel de complejidad que requiere su utilización.
- **Opciones de posicionamiento:** indica el nivel de opciones de configuración para establecer el posicionamiento del mensaje en la web.
- **Efectos visuales:** indica el nivel de opciones de configuración de efectos visuales para mostrar el mensaje en la web.
- **Documentación:** indica el volumen de documentación existente para la utilización de la tecnología.

Tabla 1.2. Comparativa de tecnologías para el envío de notificaciones en el sistema. Fuente: Elaboración propia

Criterio/Tecnología	Notify.js	Noty.js	Toastr.js	PNotify.js	Amaran.js
Facilidad de uso	7	4	7	10	9
Opciones de posicionamiento	7	9	8	10	8
Efectos visuales	6	8	9	9	8
Documentación	6	7	9	10	10
<b>Total</b>	26	28	33	39	35

Atendiendo los parámetros medidos en la comparación de las bibliotecas anteriores resalta PNotify<sup>2</sup> como la biblioteca *javascript* más completa de acuerdo a las opciones de configuración que se necesitan para la implementación del envío de notificaciones a nivel de sitio en la solución a desarrollar. PNotify es un plugin de notificación *javascript*, desarrollado por *SciActive*<sup>3</sup>. Está diseñado para proporcionar un nivel de flexibilidad sin precedentes, sin dejar de ser fácil de implementar y utilizar. PNotify ofrece notificaciones de escritorio basados en web. Puede proporcionar avisos no bloqueantes, o sea, permite que el usuario haga clic en elementos detrás de la notificación, sin tener que descartarlo. PNotify utiliza Bootstrap o jQuery UI para el estilo visual, lo que significa que es fácil de adaptar. Funciona en los principales navegadores y proporciona una interfaz consistente (WEBAPPERS.COM, 2014). Como valor agregado cuenta con una extensión que facilita su uso en el *framework* Yii. Por las características antes mencionadas, se selecciona PNotify en el desarrollo del módulo de notificaciones. También surge como necesidad la visualización flexible de la información almacenada sobre las sesiones de juego registradas para cada paciente, por lo que a continuación se profundizará en las formas de representación más utilizadas en la web.

<sup>2</sup>Disponible en [Repositorio GitHub](#)

<sup>3</sup>Sitio Oficial: <http://www.sciactive.net/>

## 1.6. Graficación

La presentación de información estadística, es una representación visual de una serie de datos estadísticos que tiene dos opciones de formato: el tabular y el gráfico (ABAD y HUAPAYA, 2009). El primero consiste en el ordenamiento de los datos seleccionados, en filas y columnas, denominado cuadro estadístico. El formato gráfico consiste en la utilización de puntos, líneas y figuras que sirven para mostrar magnitudes asociadas a una escala de medición. De esta manera es fácil realizar la comparación e interpretación de los datos estadísticos sin la necesidad de incluir los valores numéricos.

### 1.6.1. Tipos de gráficas y su utilización

La mayor parte de gráficos son de ejes coordenados, que usan líneas rectas ortogonales e interceptadas. Uno de los ejes se selecciona para indicar la escala de valores de los datos estadísticos (eje de valores) y el otro para indicar los conceptos a los que se refieren los datos estadísticos como objeto de representación (eje de conceptos) (DÍAZ-LEVICOY, 2014). A continuación, es ofrecida una breve explicación de los principales gráficos utilizados a nivel mundial, mencionando en cada uno la tendencia que se sigue para el uso correspondiente.

#### Gráfico de barras

Los gráficos de barra se utilizan para representar datos cualitativos o cuantitativos discretos. En este gráfico se suelen disponer los datos en el primer cuadrante de unos ejes de coordenadas cartesianas, levantando sobre el eje de abscisas un bloque o barra para cada modalidad de la variable observada. En uno de los ejes son representadas las categorías de la variable, mientras que para cada valor de esta, la frecuencia de aparición se representa mediante una barra (ver Figura 1.1) (ABAD y HUAPAYA, 2009). También es posible construir con porcentajes o frecuencias relativas, en lugar de absolutas. Si la variable es numérica, los valores de la variable se presentan en orden numérico y las barras se reducen a una línea situada sobre el valor de la variable. Aunque no existen reglas estrictas, sí se siguen algunas normas generales para este tipo de representación:

1. El ancho de la barra debe ser uniforme para todas las barras del diagrama.
2. La longitud de la barra debe ser proporcional a la cantidad que representa.
3. El espacio de separación entre barras por cada concepto debe ser constante.
4. Las barras en estos gráficos pueden disponerse vertical u horizontalmente.

Ejemplo de gráficas de barras:

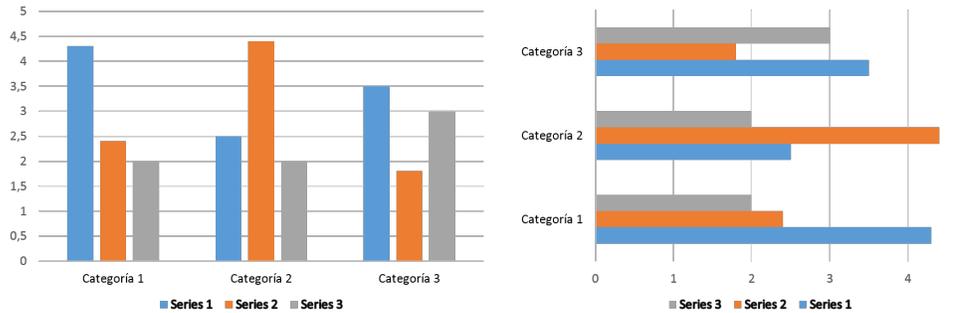


Figura 1.1. Gráficos de barras vertical y horizontal

### Gráfico de líneas

Es un gráfico estadístico que se utiliza para representar frecuencias de una variable cualitativa o bien valores numéricos de una serie de datos, que permite comparar la información de forma aislada. En este tipo de gráfico se utilizan puntos cuya altura indica la frecuencia y el valor de la variable. De forma tradicional, los puntos van unidos con líneas para evidenciar el cambio de una variable a medida que pasa el tiempo o cambian los valores numéricos de la serie de datos (DÍAZ-LEVICOY, 2014) (ver Figura 1.2). Para mejor comprensión, se muestra un ejemplo de gráfico de líneas.

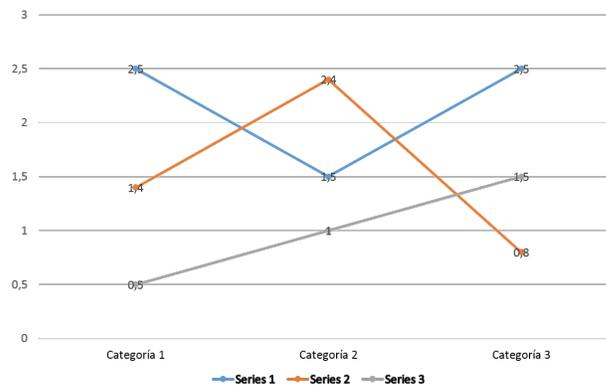


Figura 1.2. Gráfico de líneas

### Gráfico circular o de sectores

También conocido como gráfico de sectores, es una representación circular de las frecuencias relativas de una variable cualitativa o discreta que permite ver la distribución interna de los datos que representan un hecho en forma de porcentajes sobre un total (ABAD y HUAPAYA, 2009). Consiste en dividir un círculo en secciones cuyas superficies sean proporcionales a las cantidades correspondientes a cada categoría

## 1.6. GRAFICACIÓN

(ver Figura 1.3). Es recomendable utilizarlo cuando la cantidad de categorías a representar es pequeña. A continuación, se muestra un ejemplo de gráficas de pastel o torta.

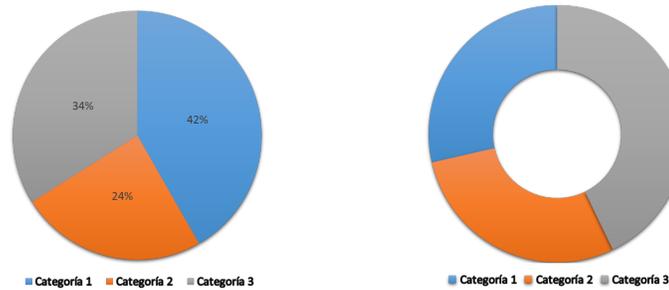


Figura 1.3. Gráfico de sectores

### Diagrama de dispersión o XY

Este tipo de gráfico es de mucha utilidad en los estudios que tienen muchas observaciones, debido a que muestra la asociación o relación que existe entre dos variables unidas a un mismo fenómeno. En este gráfico cada observación debe aparecer situada en el plano como un simple punto, cuya posición es fijada en relación a cada una de las variables (ver Figura 1.4). Las escalas van precisadas en cada uno de los ejes rectangulares del dibujo, una para cada variable; asentándose mediante el mismo principio todas las demás observaciones que constituyen los datos del problema. Además puede ser utilizado para determinar el grado de correlación entre las dos variables (DÍAZ-LEVICOY, 2014). Un ejemplo de un gráfico de dispersión, es representado por el siguiente ejemplo.

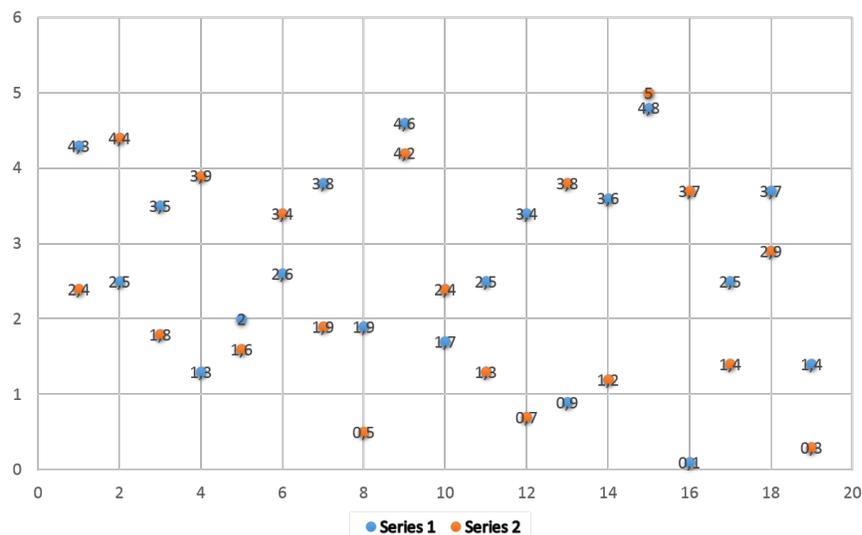


Figura 1.4. Gráfico de dispersión

Interminable es la lista de aplicaciones que utilizan la generación de gráficas en sus servicios. Para esta tarea se han desarrollado una gran variedad de bibliotecas *javascript* que ofrecen un amplio conjunto de opciones para visualizar información de una forma más intuitiva. Sus resultados son aprovechados por los profesionales para realizar un mejor análisis estadístico y reforzar la base de la toma de sus decisiones.

### 1.6.2. Selección de tecnología para representar información

Para implementar el mecanismo de graficación se hace necesario seleccionar la tecnología o herramienta que mejor se adapte a las especificaciones de la plataforma. Se han revisado diversos análisis por parte de varios especialistas (DORMEHL, 2014; LEARNINGJQUERY.COM, 2015; RAHMAN, 2015), y las bibliotecas *javascript* que más resaltan son: D3.js, Chartist.js, FusionCharts, Chart.js, Google Charts y HighCharts. Se han descartado las bibliotecas FusionCharts, Google Charts y HighCharts por particularidades especiales que son explicadas a continuación:

- **FusionCharts:** aunque cuenta con una alternativa liberada bajo licencia *Free*, para poder utilizarla, la aplicación debe estar montada en un servidor pagado (WEBGIS, 2015).
- **Google Charts:** carga las funcionalidades directamente desde servidores de Google, por lo que es indispensable que la aplicación cuente con una conexión ininterrumpida a Internet (DEVELOPERS, 2015).
- **HighCharts:** ofrece infinidad de opciones de visualización, incluso con habilidades para exportar los gráficos a diversos formatos, sin embargo no permite desplegar la aplicación que la utilice sin pagar por ello (HIGHCHARTSJS, 2015).

Quedan para el análisis las restantes bibliotecas, para las cuales se realiza una comparativa a partir de la evaluación de cada una en función de su aplicación en ejemplos de prueba. A continuación se describen los parámetros evaluados en la comparativa:

- **Documentación y comunidad:** permite evaluar el volumen de documentación que existe referente al uso de la biblioteca en conjunto con la extensión y actividad que manifieste la comunidad, en caso de que exista.
- **Facilidad de uso:** mide el nivel de complejidad que presenta la tecnología en función de explotar sus funcionalidades.
- **Opciones de graficación:** permite medir la biblioteca en correspondencia con la cantidad de opciones de visualización de la información que ofrezca. Para ello se relacionan con las etiquetas de la siguiente manera: Bajo={de 1 a 3}, Medio={de 4 a 6}, Alto={más de 6}.
- **Formato de entrada de datos:** describe los formatos de entrada de los datos para construir el gráfico, donde JS API se refiere a simples arreglos *javascript*.
- **Licencia:** identifica el tipo de licencia bajo la que se distribuye la biblioteca.

Tabla 1.3. Comparativa de tecnologías para la representación visual de la información. Fuente: Elaboración propia

Parámetro/Biblioteca	D3.js	Chartist.js	Chart.js
Documentación y comunidad	Alto	Alto	Alto
Facilidad de uso	Bajo	Alto	Medio
Opciones de graficación	Alto	Alto	Medio
Formato de entrada	JSON, XML	JSON, JS API	JSON, JS API
Licencia	Open Source	Open Source	Open Source

A partir del análisis realizado con las diferentes bibliotecas, teniendo en cuenta los parámetros comparados y otras facilidades detectadas, resalta Chartist.js<sup>4</sup> como la que mejor se adapta a las necesidades y condiciones de la plataforma. Chartist.js es una biblioteca desarrollada para graficar de forma simple y *responsive*, resulta fácil de usar y configurar. Utiliza imágenes en formato [Gráfico de Vectores Escalables \(SVG, por sus siglas en inglés\)](#) para el renderizado y permite aplicar diversos estilos [Cascade Style Sheet \(CSS3\)](#), incluyendo atractivas animaciones. Cuenta con una vasta comunidad y clara documentación que facilita su utilización (GIONKUNZ, 2015). Se escoge Chartist.js para la generación de las gráficas en la plataforma por las razones antes mencionadas. A pesar de que esta forma de visualizar la información permiten obtener una centralización de la misma, resulta muy provechoso contar con opciones de reportes que agilicen el análisis de esta.

## 1.7. Reportes

Al manejar gran volumen de información se hace necesario filtrar la misma, esta acción requiere tener en cuenta los aspectos más relevantes con el fin de agrupar estos datos para un mejor provecho. Los reportes de aplicaciones web proporcionan de forma organizada gran volumen de información, lo que permite a los usuarios receptores realizar un mejor análisis de la misma.

Un reporte es un informe o documento que pretende transmitir una información, aunque puede tener diversos objetivos. Estos documentos combinan varios tipos de información como son el contenido, la estructura y el diseño de cómo serán presentados. En el ámbito de la informática, los reportes son informes que organizan y exhiben la información contenida en una base de datos. Su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios (QUESADA, 2012).

A través de los reportes se refleja el comportamiento de los diferentes componentes que integran cualquier sistema de información de una empresa o institución. Esta técnica permite el reconocimiento, control y monitoreo de las diversas problemáticas existentes en la misma, para apoyar así en la correcta toma de decisiones (GONZÁLEZ y ALFONSO, 2013).

Teniendo en cuenta las definiciones anteriores se puede entender reporte, en el contexto web, como

---

<sup>4</sup>Disponible en [Repositorio GitHub](#)

un documento generado por la aplicación web que presenta de una forma estructurada, datos relevantes almacenados en una base de datos o generados por la misma aplicación. Estos reportes de forma general agrupan los datos de acuerdo a un interés específico y representan todos los documentos a los que tiene acceso el sistema. Son generados de forma dinámica, o sea, siempre que se invoque un reporte este se actualiza incluyendo la información más reciente. Permiten al usuario realizar de forma transparente consultas a las bases de datos y obtener información de ellas en un determinado formato. Este procedimiento es ejecutado con rapidez y con un mayor nivel de detalle y flexibilidad.

### 1.7.1. Observación del comportamiento de mecanismos de reportes en aplicaciones web

Para comprender las salidas de informes especializados en plataformas para la Telemedicina, se observan varias aplicaciones que implementan algún mecanismo de reportes. Para ello se analizan los siguientes sistemas:

- **SALUS**: es un *software* para la gestión integral de clínicas, centros médicos y hospitales, desarrollado por la empresa española QSOFT, que permite gestionar el conjunto de áreas de gestión de un centro de salud. Es utilizada para facilitar la organización y comunicación entre profesionales del mismo centro. En su versión para hospitales, toda la información económica y de gestión es explotable mediante un conjunto de funcionalidades de *reporting* y *business intelligence*. Estas permiten un control total, interactivo y en tiempo real del conjunto de indicadores de actividad del hospital. Los indicadores pueden ser tanto económicos como sanitarios, siendo posible consultar sus valores estáticos, así como su evolución a lo largo del tiempo. Mediante una herramienta de diseño de listados personalizados disponible para los clientes que se abastecen del servicio de soporte técnico, las consultas a la base de datos pueden ser definidas y personalizadas por el propio hospital. Estos listados permiten su exportación hacia aplicaciones de cálculo como Excel u otros (QSOFT, 2015).
- **National Rehabilitation Reporting System (NRS)**: es un sistema de análisis de informes de los Sistemas Hospitalarios de Canadá, desarrollado para la organización CIHI<sup>5</sup>. El conjunto de datos mínimos de NRS contiene datos clínicos sobre el estado funcional del paciente basado en elementos cognitivos. También se recoge información de la salud, características socio-demográfico y administrativas para cada cliente en rehabilitación. Instalaciones recogen datos de los clientes en materia de admisión y alta del programa de rehabilitación para pacientes hospitalizados y envían los datos directamente a CIHI trimestralmente para ser incluidos en los informes comparativos. NRS libera a través de la línea segura de *eReporting*<sup>6</sup> los datos trimestrales para las instalaciones participantes y otros usuarios autorizados. Los reportes están disponibles ocho semanas después del final del trimestre fiscal. Estos informes de rehabilitación e indicadores proporcionan facilidades con información para evaluar los cambios en las instituciones; así como el mantenimiento de los resultados de los clientes. Permiten además examinar

---

<sup>5</sup>Organización que proporciona información sobre el sistema de salud de Canadá.

<sup>6</sup>Línea de ayuda que atiende solicitudes de reportes con información predefinida.

el acceso a la rehabilitación de pacientes, evaluar los programas de salud y analizar los servicios de las instituciones (CIHI, 2016).

El estudio de algunos de los diferentes sistemas informáticos existentes, permitió observar y analizar los diferentes formatos para la presentación de las salidas de información. Se pudo observar que estos módulos de reportes tienen características y funcionalidades similares, ya que fueron creados con el mismo propósito de ayudar a los usuarios a obtener reportes de información almacenada en una base de datos. Además la mayoría ofrecen la posibilidad de poder exportarlos a formatos para ser impresos. Las diferencias radican en que cada uno responde a ciertos requerimientos del sistema de información para el cual se implementó. Estas aplicaciones son de carácter propietario y su obtención para el estudio resulta difícil, por lo que se decide realizar un mecanismo propio con tecnologías libres, que garantice la independencia de terceros.

Para generar los reportes se definen procedimientos desarrollados en correspondencia con los datos que se almacenan sobre las sesiones de juego almacenadas en el sistema. Mientras que para exportar esta información se seleccionan las facilidades que proporciona la clase **mPDF.php**.

Con las tecnologías individuales, identificadas para la implementación de las funcionalidades que responderán a las necesidades identificadas, solo resta definir las herramientas más generales utilizadas para desarrollar la solución propuesta. A continuación se define el marco de trabajo, lenguajes de programación, **SGBD**, entorno de desarrollo y la metodología que se utilizarán para lograr los resultados esperados.

### 1.8. Herramientas, tecnología y metodología

Considerando el estudio realizado previo a la presente investigación, se asumen y adoptan las herramientas y tecnologías utilizadas por el equipo de desarrollo de la versión 1.0 de la plataforma Medicando (GUTIERREZ, 2015). Las principales herramientas a utilizar para llevar a cabo el proceso de desarrollo de la solución se describen a continuación.

- **Marco de trabajo:** se utilizó el *framework* Yii en su versión 1.1.14. Yii es un *framework* basado en componentes escrito en **PHP**, para el desarrollo rápido a gran escala de aplicaciones web. Permite la máxima reutilización en la programación web y acelera de manera significativa el proceso de desarrollo de aplicaciones web (WINESETT, 2010). Todas las aplicaciones se construyen bajo el patrón arquitectónico **Modelo-Vista-Controlador (MVC)**. Yii, ha sido diseñado para permitir que casi todas las piezas de su código puedan extenderse y personalizarse para satisfacer casi cualquier necesidad o requerimiento. Sobresale en comparación con otros *frameworks* **PHP** en ser eficiente, rico en características y en estar bien documentado (XUE y ZHUO, 2013). Esta herramienta soporta como lenguajes de programación **HyperText Markup Language (HTML5)**, **CSS3**, **PHP** v5 y JavaScript.
- **Entorno de desarrollo:** fue utilizado NetBeans 8.0, esta herramienta permite el desarrollo rápido y fácil de aplicaciones Java de escritorio, móviles y aplicaciones web, así como aplicaciones **HTML5** con JavaScript y **CSS3**. El **Integrated Development Environment (IDE)** también proporciona un gran conjunto de herramientas para desarrolladores de **PHP** y C/C ++. Es gratuito y de código abierto,

además es objeto de colaboración de una gran comunidad de usuarios y desarrolladores de todo el mundo (NETBEANS.ORG, 2015).

- **Sistema Gestor de Base de Datos:** se utilizó MySQL, este es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Es uno de los gestores más usado en el mundo del *software* libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación, es debida a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración (PECOS, 2002). Las características fundamentales que refleja para su elección por encima de otros gestores de base de datos son: el coste gratuito, la velocidad operacional, facilidad de uso y de integración con la mayor parte de los entornos de programación, la existencia de una nutrida y activa comunidad (GILFILLAN, 2008).
- **Metodología de desarrollo de *software*:** Se siguieron los pasos que define la metodología *XP* para guiar el proceso de desarrollo de *software*, atendiendo al esquema que se proponen en (ESCRIBANO, 2002), adoptando sus 4 fases (Planificación, Diseño, Desarrollo y Prueba). *XP* es la más destacada de los procesos ágiles de desarrollo de *software* formulada por Knet Beck<sup>7</sup>. Tiene éxito porque hace hincapié en la satisfacción del cliente. Faculta a sus desarrolladores para responder con seguridad a las necesidades cambiantes de los clientes, incluso en etapas tardías del ciclo de vida del producto. Enfatiza el trabajo en equipo. Los jefes de proyecto, clientes y desarrolladores son socios iguales en un equipo de colaboración que se auto-organiza en torno al problema a resolver de la forma más eficiente posible. *XP* mejora un proyecto de *software* en cinco aspectos esenciales; la comunicación, la sencillez, la retroalimentación, el respeto y el valor. Los programadores extremos se comunican constantemente con sus clientes y colegas de trabajo, mantienen su diseño simple y limpio. Reciben retroalimentación probando su *software* a partir del primer día y entregan el sistema a los clientes tan pronto como sea posible e implementan cambios como se sugiere. Cada pequeño éxito profundiza su respeto por las contribuciones únicas de cada uno y cada miembro del equipo. Con esta base los programadores extremos son capaces de responder con valentía a las cambiantes necesidades y la tecnología (JOSKOWICZ, 2008).

---

<sup>7</sup>Padre fundador de la metodología *XP*

## 1.9. Conclusiones del capítulo

A partir del estudio de aplicaciones web orientadas a la Telemedicina y la observación del comportamiento de los módulos de notificaciones, reportes y visualización de la información, fue posible arribar a las siguientes conclusiones:

- Se identificó como las principales vías para el envío de notificaciones, la generación de mensajes de alerta por el propio sistema y el despacho de correos electrónicos.
- Para solucionar las insuficiencias detectadas se selecciona PNotify.js como biblioteca *javascript* que permitirá mostrar los mensajes en la plataforma, mientras que la extensión CrugeMailer facilitará el envío de correos electrónicos. Chartist.js es escogida para la reproducción de las gráficas que representarán las estadísticas almacenadas por las sesiones de rehabilitación y los reportes que se generen serán exportados utilizando la clase mPDF.php.
- Se adopta como marco de trabajo el *framework* Yii en su versión 1.1.14 porque permite una construcción estructurada, escalable y extensible de aplicaciones web, optimizando el empleo de lenguajes como [HTML5](#), [CSS3](#), JavaScript y [PHP](#) v5, 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 acoge XP como metodología de desarrollo, ya que garantiza la generación de artefactos que son necesarios para una mayor comprensión de la solución a desarrollar.

De esta manera se está en condiciones de desarrollar los módulos Notificaciones, Gráficas y Reportes para la plataforma Medicando, utilizando las herramientas y tecnologías seleccionadas.

#### 2.1. Introducción al capítulo

En el presente capítulo se presenta una propuesta de solución para los problemas detectados en el estudio del estado del arte realizado en el Capítulo 1. Se mencionan los roles que intervienen en la solución. También se definen las [HU](#), la planificación de entrega de versiones del producto y el diseño del sistema de acuerdo a las fases que propone [XP](#).

#### 2.2. Propuesta de solución

Luego de haber analizado las necesidades del sistema y seleccionado las herramientas para la implementación, se definen los módulos a desarrollar para dar solución al problema planteado.

- **Módulo Notificaciones:** responderá por la necesidad a informar a los usuarios del sistema, ante la ejecución de acciones en la aplicación, que afecte información sensible relacionada con estos.
- **Módulo Gráficas:** ofrecerá un mecanismo que permite visualizar la información almacenada por las sesiones de juegos, de forma que los especialistas puedan seleccionar el campo y la forma de representar los datos, siempre que la forma seleccionada sea posible.
- **Módulo Reportes:** permitirá a los doctores obtener informes especializados en correspondencia con la información que maneja la plataforma. Pretende brindar una mejor interpretación de la información para el análisis de los datos generados a partir de la aplicación de tratamientos a pacientes.

Para el desarrollo de los módulos propuestos se siguen los pasos que establece la metodología ágil [XP](#), donde se respetará el esquema que explica (ESCRIBANO, 2002).

## 2.3. Fase I: Planificación

Es la fase en la que se define el alcance general del proyecto. En esta el cliente define lo que necesita mediante la redacción de **HU** y establece la prioridad de cada una. Luego, los programadores estiman los tiempos de desarrollo en base a esta información. Las estimaciones realizadas en esta fase son primarias, debido a que estarán basadas en datos de muy alto nivel y podrían variar cuando se analicen en cada iteración. 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 (ESCRIBANO, 2002; JOSKOWICZ, 2008).

### 2.3.1. Historias de Usuarios

Las **HU** son la técnica que utiliza **XP** para especificar los requisitos de *software*, estas 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 **HU**. 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. (ESCRIBANO, 2002). A continuación se describen las **HU** definidas para llevar a cabo el desarrollo de los módulos.

Tabla 2.1. Historia de Usuario # 1

Historia de Usuario	
<b>Número:</b> 1	<b>Nombre:</b> Administrar notificaciones
<b>Usuario:</b> Administrador, Usuario	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 1.4	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<p><b>Descripción:</b> Ofrece al administrador una vista para la eliminación de las notificaciones ya leídas que aún se encuentran almacenadas en la base de datos. También permite listar y eliminar las notificaciones realizadas vía correo electrónico. Además, se ofrecen funcionalidades para que cada usuario pueda configurar en su perfil, las notificaciones que desea recibir y por la vía que se realizará. Estas vías son: en el sistema y por correo electrónico. Las notificaciones generales que se encuentran activas para los distintos usuarios son:</p> <ul style="list-style-type: none"> <li>• Asignación y liberación de doctor/paciente</li> <li>• Asignación y liberación de tratamiento</li> <li>• Tratamiento iniciado</li> <li>• Tratamiento terminado</li> <li>• Creación de cuenta</li> <li>• Sincronización de salva</li> </ul>	

Continúa en la próxima página

Tabla 2.1. Continuación de la página anterior

**Observaciones:** Cada usuario cuenta con una configuración por defecto según el rol que desempeñe en la aplicación. Las notificaciones que elija serán las que recibirá; en correspondencia además con la vía seleccionada.

**Prototipo de interfaz:**

Vista para administrar notificaciones

## Administración de notificaciones

Configurar notificaciones por rol
Eliminar notificaciones leídas
Borrar seleccionados

Viendo 1-2 de 2 resultados.

<input type="checkbox"/>	Destinatario	Cuerpo	Fecha	Enviado	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
<input type="checkbox"/>	jgvaldes@uci.cu	A usted se le ha creado una cuenta con rol Doctor, en la plataforma Medicando.	2016-06-06	Enviado	
<input type="checkbox"/>	pepito@uci.cu	A usted se le ha creado una cuenta con rol Doctor, en la plataforma Medicando.	2016-06-06	Enviado	

Tabla 2.2. Historia de Usuario # 2

<b>Historia de Usuario</b>	
<b>Número:</b> 2	<b>Nombre:</b> Enviar notificación al usuario
<b>Usuario:</b> Doctor, Paciente	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 1.8	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<p><b>Descripción:</b> De forma dinámica, envía una notificación a los usuarios cuando se realiza alguna acción en el sistema que le afecte o involucre directamente. Para ello, se confirma según su configuración de notificaciones en el sistema. Las notificaciones se hacen llegar a través del sitio y por correo electrónico, ellas son:</p> <ul style="list-style-type: none"> <li>Asignación y liberación de paciente/doctor</li> <li>Asignación y liberación de tratamiento</li> <li>Tratamiento iniciado por un paciente</li> <li>Tratamiento terminado por un paciente</li> <li>Enviar notificación al paciente</li> <li>Sincronización de salva</li> <li>Creación de cuenta personal</li> </ul>	
<p><b>Observaciones:</b> Se enviarán las notificaciones a cada usuario del sistema en caso de que este desee recibirla y por la vía correspondiente de acuerdo a su configuración personal para recibir los mensajes.</p>	

### 2.3. FASE I: PLANIFICACIÓN

Tabla 2.3. Historia de Usuario # 3

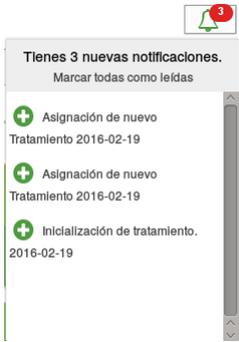
Historia de Usuario	
<b>Número:</b> 3	<b>Nombre:</b> Mostrar notificaciones
<b>Usuario:</b> Usuario	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Bajo
<b>Puntos estimados:</b> 0.4	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Permite a cada usuario observar cada notificación pendiente a partir de una lista desplegable que contiene el título del mensaje. Además, ofrece la posibilidad de marcar todas las notificaciones como leídas.	
<b>Observaciones:</b> Una vez seleccionada una notificación, esta se muestra y se marca automáticamente como leída, por lo que no será listada nuevamente.	
<b>Prototipo de interfaz:</b>	
	

Tabla 2.4. Historia de Usuario # 4

Historia de Usuario	
<b>Número:</b> 4	<b>Nombre:</b> Enviar notificación al paciente
<b>Usuario:</b> Doctor	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Bajo
<b>Puntos estimados:</b> 0.4	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Permite al doctor enviar una notificación directamente a un paciente seleccionado. Esta contará con los siguientes campos: <ul style="list-style-type: none"> <li>• Asunto (Obligatorio)</li> <li>• Mensaje (Obligatorio)</li> </ul>	
<b>Observaciones:</b> La notificación le llegará al paciente en caso de que este tenga activada la opción de recibir este tipo de mensaje en su configuración personal.	

Continúa en la próxima página

Tabla 2.4. Continuación de la página anterior

**Prototipo de interfaz:**

Tabla 2.5. Historia de Usuario # 5

Historia de Usuario	
<b>Número:</b> 5	<b>Nombre:</b> Generar gráfica de sesiones de juego
<b>Usuario:</b> Doctor	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<p><b>Descripción:</b> Permite a los doctores generar una gráfica con los valores de las sesiones de juego almacenadas en la base de datos. Para ello se hace necesario seleccionar el paciente y el campo o variable a representar visualmente. Luego en correspondencia con la variable seleccionada, se ofrece la forma de representación posible y el usuario elije la deseada. Dentro de las gráficas que se manejan se encuentran:</p> <ul style="list-style-type: none"> <li>• Gráfica de barras</li> <li>• Gráfica de líneas</li> <li>• Gráfica de pastel o circular</li> <li>• Gráfica de dispersión</li> </ul>	
<p><b>Observaciones:</b> El doctor selecciona un videojuego y se muestra una vista para que seleccione el paciente que desea analizar. Además, este selecciona el campo a graficar y se muestran las opciones de visualización.</p>	
<p><b>Prototipo de interfaz:</b></p>	

Tabla 2.6. Historia de Usuario # 6

Historia de Usuario	
<b>Número:</b> 6	<b>Nombre:</b> Generar documento PDF
<b>Usuario:</b> Doctor	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Permite a los doctores, generar un documento en formato PDF a partir de la información visualizada o seleccionada en los reportes.	
<b>Observaciones:</b>	

Tabla 2.7. Historia de Usuario # 7

Historia de Usuario																						
<b>Número:</b> 7	<b>Nombre:</b> Generar reporte: Mis Pacientes																					
<b>Usuario:</b> Doctor																						
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Media																					
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 3																					
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz																						
<p><b>Descripción:</b> Permite al doctor, generar un reporte sobre los pacientes que tiene asignado, mostrando de forma organizada los siguientes datos por cada paciente:</p> <ul style="list-style-type: none"> <li>• Nombre del paciente</li> <li>• Tratamientos asignados</li> <li>• Áreas (Áreas de los videojuegos asociados al tratamiento)</li> <li>• Porcentaje de cumplimiento de su tratamiento</li> </ul> <p>Además, se ofrece la posibilidad de exportar la información a documento de formato PDF.</p>																						
<p><b>Observaciones:</b> El doctor debe tener al menos un paciente asignado y además, que este tenga al menos un tratamiento iniciado, de lo contrario, no existirá información que reportar.</p>																						
<p><b>Prototipo de interfaz:</b></p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>Tratamientos de mis pacientes</b></p> <table border="1"> <thead> <tr> <th>Paciente</th> <th>Tratamientos Asignados</th> <th>Áreas</th> <th>Porcentaje de cumplimiento</th> </tr> </thead> <tbody> <tr> <td rowspan="2">Eliuvis Matos Matos</td> <td>Tratamiento con V2</td> <td>FUNCIONES VISUALES</td> <td>0%</td> </tr> <tr> <td>Tratamiento Remember</td> <td>No hay videojuegos asociados aún.</td> <td>0%</td> </tr> <tr> <td rowspan="3">Danier Lorenzo Pérez</td> <td>Tratamiento con V2</td> <td>FUNCIONES VISUALES</td> <td>25%</td> </tr> <tr> <td>Tratamiento de varios juegos</td> <td>FUNCIONES VISUALES</td> <td>0%</td> </tr> <tr> <td>Tratamiento Remember</td> <td>No hay videojuegos asociados aún.</td> <td>0%</td> </tr> </tbody> </table> <p> </p> </div>		Paciente	Tratamientos Asignados	Áreas	Porcentaje de cumplimiento	Eliuvis Matos Matos	Tratamiento con V2	FUNCIONES VISUALES	0%	Tratamiento Remember	No hay videojuegos asociados aún.	0%	Danier Lorenzo Pérez	Tratamiento con V2	FUNCIONES VISUALES	25%	Tratamiento de varios juegos	FUNCIONES VISUALES	0%	Tratamiento Remember	No hay videojuegos asociados aún.	0%
Paciente	Tratamientos Asignados	Áreas	Porcentaje de cumplimiento																			
Eliuvis Matos Matos	Tratamiento con V2	FUNCIONES VISUALES	0%																			
	Tratamiento Remember	No hay videojuegos asociados aún.	0%																			
Danier Lorenzo Pérez	Tratamiento con V2	FUNCIONES VISUALES	25%																			
	Tratamiento de varios juegos	FUNCIONES VISUALES	0%																			
	Tratamiento Remember	No hay videojuegos asociados aún.	0%																			

Tabla 2.8. Historia de Usuario # 8

Historia de Usuario																			
<b>Número:</b> 8	<b>Nombre:</b> Generar reporte: Historia Clínica																		
<b>Usuario:</b> Doctor																			
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Media																		
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 3																		
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz																			
<p><b>Descripción:</b> Permite al doctores, realizar un reporte sobre un paciente en específico, donde se muestre la siguiente información:</p> <ul style="list-style-type: none"> <li>• Tratamientos asignados</li> <li>• Áreas (Áreas de los videojuegos asociados al tratamiento)</li> <li>• Porcentaje de cumplimiento vencido</li> <li>• Tiempo bajo tratamiento (tiempo transcurrido en días desde que inició el tratamiento hasta la fecha actual)</li> <li>• Observaciones (observaciones o valoraciones que opina el doctor sobre la ejecución del tratamiento)</li> </ul> <p>Además, se ofrece la posibilidad de exportar la información a documento de formato PDF.</p>																			
<b>Observaciones:</b> El doctor debe seleccionar el paciente que desea visualizar en el reporte, de lo contrario no se muestra información alguna.																			
<p><b>Prototipo de interfaz:</b></p>  <table border="1"> <thead> <tr> <th>Tratamientos Asignados</th> <th>Áreas</th> <th>Porcentaje de cumplimiento</th> <th>Tiempo bajo tratamiento</th> <th>Observaciones</th> <th></th> </tr> </thead> <tbody> <tr> <td>Tratamiento con V2</td> <td>FUNCIONES VISUALES</td> <td>0%</td> <td>1 días</td> <td>No existen observaciones aún.</td> <td>🟢</td> </tr> <tr> <td>Tratamiento Remember</td> <td>No hay videojuegos asociados aún.</td> <td>0%</td> <td>0 días</td> <td>No existen observaciones aún.</td> <td>---</td> </tr> </tbody> </table>		Tratamientos Asignados	Áreas	Porcentaje de cumplimiento	Tiempo bajo tratamiento	Observaciones		Tratamiento con V2	FUNCIONES VISUALES	0%	1 días	No existen observaciones aún.	🟢	Tratamiento Remember	No hay videojuegos asociados aún.	0%	0 días	No existen observaciones aún.	---
Tratamientos Asignados	Áreas	Porcentaje de cumplimiento	Tiempo bajo tratamiento	Observaciones															
Tratamiento con V2	FUNCIONES VISUALES	0%	1 días	No existen observaciones aún.	🟢														
Tratamiento Remember	No hay videojuegos asociados aún.	0%	0 días	No existen observaciones aún.	---														

Tabla 2.9. Historia de Usuario # 9

Historia de Usuario	
<b>Número:</b> 9	<b>Nombre:</b> Generar reporte: Cumplimiento de Tratamientos
<b>Usuario:</b> Doctor	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Baja
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	

Continúa en la próxima página

Tabla 2.9. Continuación de la página anterior

**Descripción:** Permite al doctor, realizar un reporte donde se obtenga y muestre la información de los tratamientos, organizada de forma tal, que se cuente con los siguientes campos:

- Nombre del tratamiento
- Áreas (Áreas de los videojuegos asociados al tratamiento)
- Doctor que crea el tratamiento
- Cantidad de pacientes asociados
- Cantidad de veces terminado (Número de pacientes que han culminado el tratamiento)
- Porcentaje de cumplimiento (relación entre pacientes asociados y pacientes que terminan el tratamiento)

Además, se ofrece la posibilidad de exportar la información a documento de formato PDF.

**Observaciones:**

**Prototipo de interfaz:**

**Tratamientos registrados**

Nombre	Áreas	Creado por	Pacientes asociados	Cant. de veces terminado	Porcentaje de cumplimiento
Tratamiento con V2	FUNCIONES VISUALES	Pastor Segundo Gutiérrez Rondón	2	0	0%
Tratamiento de varios juegos	FUNCIONES VISUALES	Pastor Segundo Gutiérrez Rondón	2	0	0%
Tratamiento Remember	No hay videojuegos asociados aún.	Pastor Segundo Gutiérrez Rondón	2	0	0%

### 2.3.2. Estimación de esfuerzo por Historia de Usuario

Se realiza la estimación de esfuerzo que arroja cada HU, con el objetivo de obtener un correcto desarrollo del sistema. Para una mayor organización se decide además, asignar a cada iteración el conjunto de historias agrupadas en correspondencia con el módulo al que representen. A continuación se muestra la estimación realizada:

### 2.3.3. Desarrollo del plan de iteraciones

Una vez definidas las HU y realizada una previa estimación de esfuerzos, se procede a la planificación de la etapa de implementación del sistema. En este espacio, se crea el plan de iteraciones, donde se especifica la prioridad con que se implementarán las HU organizadas por iteraciones. Teniendo en cuenta el esfuerzo asociado a las HU y a las prioridades del cliente, se define una versión que sea de valor para este.

Tabla 2.10. Estimación de esfuerzo por Historia de Usuario

Historias de Usuario		Puntos estimados(semanas)
1	Administrar notificaciones	1.4
2	Enviar notificación al usuario	1.8
3	Mostrar notificaciones	0.4
4	Enviar notificación al paciente	0.4
5	Generar gráfica de sesiones de juego	3
6	Generar documento PDF	1
7	Generar reporte: Mis Pacientes	2
8	Generar reporte: Historia Clínica	2
9	Generar reporte: Cumplimiento de Tratamientos	1

### 2.3.4. 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 HU en cada iteración como se muestra en la tabla siguiente:

Tabla 2.11. Plan de duración de las iteraciones

Iteración	Historias de Usuario		Duración (semanas)
1	1	Administrar notificaciones	4.0
	2	Enviar notificación al usuario	
	3	Mostrar notificaciones	
	4	Enviar notificación al paciente	
2	1	Generar gráfica de sesiones de juego	4.0
	2	Generar documento PDF	
3	1	Generar reporte: Mis Pacientes	5.0
	2	Generar reporte: Historia Clínica	
	3	Generar reporte: Cumplimiento de Tratamientos	
<b>Total</b>			<b>13.0</b>

### 2.3.5. Plan de entregas

En el plan de entrega que se plantea a continuación, se hace una propuesta de las versiones (*releases*) del sistema. Cada versión se conformará al finalizar una iteración.

Tabla 2.12. Plan de entrega de versiones

Entregable	Iteración 1	Iteración 2	Iteración 3
Módulos	Versión 0.1	Versión 0.2	Versión 0.3
Fecha	4/12/2015	24/2/2016	6/5/2016

## 2.4. Fase II: Diseño del sistema

La plataforma se implementó siguiendo los principios del patrón arquitectónico MVC (GUTIERREZ, 2015), la propuesta de solución estará basada de igual forma en dicho patrón. Medicando cuenta con los paquetes funcionales orientados a la gestión de Doctores, Pacientes, Tratamientos, Videojuegos, así como la sincronización de salvos y la visualización de estadísticas grupales (referentes a los videojuegos) e individuales (orientadas a los pacientes).

La Figura 2.1 muestra la representación arquitectónica de los módulos Notificaciones, Gráficas y Reportes integrados con la plataforma. Se puede observar que el módulo Notificaciones trabaja en su totalidad con los tres componentes del patrón MVC. Cada usuario puede interactuar directamente con la vista para la configuración del perfil de notificaciones, mientras que los administradores son los encargados de monitorear las opciones individuales que ofrece este módulo. Las solicitudes de los usuarios son procesadas por los controladores *NotifyconfigController.php* y *NotifyemailController.php*, para ello hacen uso de los modelos correspondientes. Medicando también es potenciado por las funcionalidades que ofrece la interfaz *NotifySender.php* para la gestión y envío de las notificaciones generadas por el sistema; mientras que para la remisión de correos electrónicos hace uso de la interfaz *NotifyCrugemailer.php*.

Por otra parte, se hace necesario exportar los reportes que se generan en los paquetes funcionales que atienden la administración de doctores y tratamientos, para ello se utiliza el componente *ReportComponent.php* que se encuentra contenido en el módulo Reportes. Este utiliza la extensión *EYiPdf.php*, que permite confeccionar un documento en formato PDF a partir de un volumen de información especificada. En tanto la representación flexible de las estadísticas que gestiona el sistema, es favorecida por las opciones que proporciona el componente *GraphicComponent.php* contenido en el módulo Gráficas.

Como valor agregado, para poder escalar el sistema y permitir que sea modular, se hace necesario desarrollar un paquete funcional que se especialice en la administración y activación de los posibles módulos a incorporar en Medicando. Como se detalla en la representación anterior, de estas funcionalidades se encargará un paquete denominado "Module".

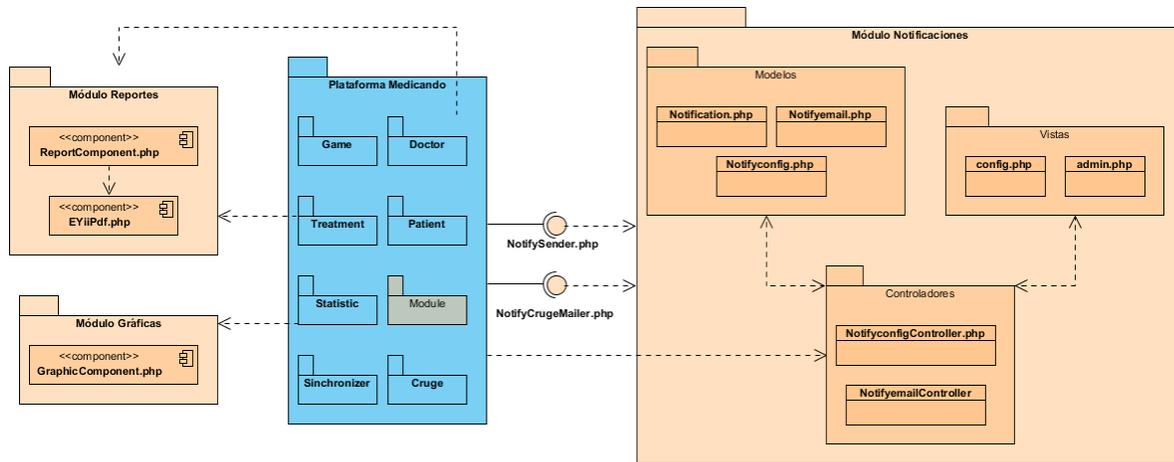


Figura 2.1. Representación arquitectónica del sistema

Para obtener una correcta implementación se hace necesario tener en cuenta las mejores soluciones conocidas, con el objetivo de diseñar las clases y las funcionalidades a codificar. Para ello se analizan y proponen algunos patrones de diseño que responden ante las necesidades de desarrollo.

## 2.5. 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. Se encargan de identificar clases, instancias, roles, colaboraciones entre estas, así como la distribución de responsabilidades (LARMAN, 2002). En resumen, es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular.

Para la asignación general de responsabilidades en el *software* existen patrones denominados **Patrones Generales de Asignación de Responsabilidades (GRASP, por sus siglas en inglés)**, que describen los principios fundamentales de asignación de responsabilidades a objetos, expresadas como patrones. Seguidamente se exponen algunos patrones utilizados dentro del contexto de los módulos a implementar:

- **Experto:** permite mantener el encapsulamiento de la información, puesto que los objetos utilizan su propia información para llevar a cabo las tareas asociadas a este. Por ejemplo, la clase *Notifyemail* es la encargada de proporcionar toda la información relacionada con las notificaciones que se generan en el sistema y se envían por correo electrónico, debido a que es la que representa todos los atributos para la creación de este objeto.
- **Creador:** se evidencia en la clase controladora *NotifyconfigController*. Esta clase se encarga de modificar la configuración de las notificaciones de cada usuario, se ofrece así una alta cohesión. Permite, de esta forma, establecer un bajo acoplamiento, lo que propicia menos dependencias funcionales y ofrece mejores oportunidades de reutilización.

- **Controlador:** este patrón se evidencia en las clases que actúan como intermediarias para el manejo de eventos. En el sistema se puede identificar en la clase *NotifyemailController*, esta se encarga de listar y eliminar notificaciones enviadas por correo electrónico a partir de la plataforma.

Por su parte, el *framework* Yii, implementa el patrón de persistencia *Active Record*, que facilita el control de la información almacenada en la base de datos. Para ello utiliza el mapeo *Object-Relational Mapping* (ORM, por sus siglas en inglés). Se trata de una clase que se encarga de implementar todas las operaciones de consulta y modificación de una tabla concreta de la base de datos. De esta forma, la aplicación delega el trabajo con SQL, a la capa de componentes *ActiveRecord* que maneja Yii. Las funcionalidades que se ofrecen, están orientadas principalmente a las propiedades de las columnas de la tabla representada.

Una vez definidos los patrones de diseño, se describen las clases utilizadas en la solución. Siguiendo la metodología XP, se analizan las HU y se descomponen en tareas independientes.

## 2.6. Tarjetas CRC

A continuación, las HU son evaluadas para dividir las 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 *Clase-Responsabilidad-Colaborador (CRC)*.

Nombre de la clase	
Responsabilidades	Colaboradores

Figura 2.2. Prototipo de tarjeta CRC

Como se puede observar en la Figura 2.2, cada tarjeta contiene el nombre de la clase, 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. A continuación se describen las tarjetas definidas para la implementación de la solución.

Tabla 2.13. Tarjeta CRC # 1

Tarjeta CRC	
Clase: Notification	
Responsabilidad	Colaboración
Representar los datos persistentes en la base de datos correspondientes a las notificaciones del sistema. Siendo una clase Modelo o también conocida como Entidad.	NotifySender Doctor Patient TreatmentHasPatient Synchronizer

Tabla 2.14. Tarjeta CRC # 2

Tarjeta CRC	
<b>Clase:</b> Notifyconfig	
Responsabilidad	Colaboración
Representar los datos persistentes en la base de datos correspondientes a las notificaciones del sistema. Siendo una clase Modelo o también conocida como Entidad.	NotifySender

Tabla 2.15. Tarjeta CRC # 3

Tarjeta CRC	
<b>Clase:</b> Notifyemail	
Responsabilidad	Colaboración
Representar los datos persistentes en la base de datos correspondientes a las notificaciones de correo del sistema. Siendo una clase Modelo o también conocida como Entidad. getStatus(key): retorna la palabra <i>Enviado</i> en caso de que la notificación identificada haya sido enviado. Muestra <i>No enviado</i> en caso contrario.	NotifyCrugeMailer NotifyemailController

Tabla 2.16. Tarjeta CRC # 4

Tarjeta CRC	
<b>Clase:</b> NotifyconfigController	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• updateUserConfig(idUser): renderiza la vista para realizar las configuraciones orientadas a recibir las notificaciones del usuario identificado.</li> <li>• updateConfig(): recibe por vía <i>AJAX</i> el identificador del usuario que solicita actualizar su configuración y los campos seleccionados en la vista para actualizar los datos en la base de datos.</li> </ul>	NotifySender

Tabla 2.17. Tarjeta CRC # 5

Tarjeta CRC	
<b>Clase:</b> NotificationController	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• ShowNotification(idNoty): muestra un mensaje con la información almacenada en la notificación identificada.</li> <li>• cleanUserNotifications(idUser): elimina de la base de datos todas las notificaciones almacenadas que involucren al usuario identificado.</li> <li>• cleanReadedNotifications(): elimina de la base de datos todas las notificaciones que tengan el estado de <i>leídas</i>.</li> </ul>	NotifySender

Tabla 2.18. Tarjeta CRC # 6

Tarjeta CRC	
<b>Clase:</b> NotifyCrugeMailer	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• notifyUser(model, service): se encarga de chequear si el usuario al que va dirigida la notificación tiene activo el tipo de notificación solicitado según la variable <i>service</i> y envía el mensaje utilizando el correo electrónico.</li> </ul>	Notifyemail NotifySender Doctor Patient TreatmentHasPatient Synchronizer

Tabla 2.19. Tarjeta CRC # 7

Tarjeta CRC	
<b>Clase:</b> ReportComponent	
Responsabilidad	Colaboración

Continúa en la próxima página

Tabla 2.19. Continuación de la página anterior

<p>buildReport(title, body, type): se encarga de crear un documento en formato PDF con el contenido que se instancia en el parámetro <i>body</i>, este documento toma como título el nombre instanciado como <i>title</i> y se ofrece la visualización o descarga del mismo, en dependencia del valor que tome la variable <i>type</i>.</p>	<p>DoctorController TreatmentController</p>
---	---

Tabla 2.20. Tarjeta CRC # 8

Tarjeta CRC	
Clase: NotifySender	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>● getAllNotifications(idUser): obtiene todas las notificaciones pendientes del usuario indicado.</li> <li>● cleanReadedNotifications(): elimina de la base de datos todas las notificaciones con estado <i>Leída</i>.</li> <li>● buildDefaultConfigurations(): se encarga de construir una configuración por defecto (para recibir notificaciones) por cada usuario que no cuente con ella, este se realiza en función del rol que desempeñe el usuario en cuestión.</li> <li>● getUserConfigurations(idUser): obtiene la configuración para recibir notificaciones del usuario indicado.</li> <li>● checkUserService(idUser, service): chequea si el usuario indicado tiene activado el tipo de mensaje solicitado para emitir una notificación.</li> <li>● getNotification(idNotification): obtiene una notificación de la base de datos según el identificador proporcionado.</li> <li>● sendNotification(model): se encarga de almacenar la notificación indicada en la base de datos, por defecto se almacena con estado <i>No leída</i>.</li> <li>● removeAllConfigurations(): se encarga de eliminar todas las configuraciones de la base de datos. Es utilizada cuando se desactiva el módulo.</li> <li>● removeAllNotificationsFromDB(): se encarga de eliminar todas las notificaciones de la base de datos. Es utilizada cuando se desactiva el módulo.</li> </ul>	Notification Doctor Patient TreatmentHasPatient Synchronizer

Tabla 2.21. Tarjeta CRC # 9

Tarjeta CRC	
Clase: GraphicComponent	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• buildChart(type, data, options): Se encarga de registrar un <i>script</i> que genera la gráfica del tipo especificado por el parámetro <i>type</i>, este puede ser: barras, línea, pastel, dispersión. Utiliza para ello los datos referenciados por el parámetro <i>data</i> y aplica las opciones <a href="#">CSS3</a> parametrizadas por <i>options</i>. En caso de que este último parámetro sea nulo, entonces se utilizan opciones por defecto según el tipo de gráfica. Este <i>script</i> se registra en la vista mediante la cual se realiza la petición.</li> <li>• getDefaultOptions(type): retorna en una cadena <i>string</i> las opciones <a href="#">CSS3</a> de acuerdo al tipo de gráfica especificado por el parámetro <i>type</i>.</li> <li>• buildBarOptions(): retorna en una cadena <i>string</i>, las opciones <a href="#">CSS3</a> para generar una gráfica de barras.</li> <li>• buildLineOptions(): retorna en una cadena <i>string</i>, las opciones <a href="#">CSS3</a> para generar una gráfica de líneas.</li> <li>• buildPieOptions(): retorna en una cadena <i>string</i>, las opciones <a href="#">CSS3</a> para generar una gráfica de pastel o circular.</li> <li>• buildDispOptions(): retorna en una cadena <i>string</i>, las opciones <a href="#">CSS3</a> para generar una gráfica de dispersión.</li> </ul>	StatisticController

## 2.7. Conclusiones del capítulo

En este capítulo se han abordado los aspectos referentes a la concepción del producto a desarrollar y sus características funcionales. Esto permitió arribar a las siguientes conclusiones:

- A partir de la definición de las [HU](#) se pudieron identificar las principales funcionalidades a desarrollar en correspondencia con los módulos Notificaciones, Gráficas y Reportes propuestos.
- La estimación del tiempo para la implementación de las [HU](#) definidas, permitió calcular una entrega final del producto en tres meses y una semana.
- Fue posible identificar los patrones GRASP para el desarrollo de la solución, así como el patrón de persistencia *Active Record* que implementa el *framework* Yii.
- Se exponen los artefactos generados en las fases de Planificación y Diseño que establece la metodología [XP](#).

Con los aspectos arquitectónicos y de diseño a utilizar definidos, queda establecida la vía para llevar a cabo la implementación y prueba de los módulos. Además se adoptan los estándares de programación definidos en (GUTIERREZ, 2015) por el equipo original de desarrollo de la plataforma Medicando.

### **3.1. Introducción al capítulo**

En el presente capítulo se detallan las iteraciones realizadas durante la etapa de construcción de los módulos propuestos, además se exponen las tareas de ingeniería generadas para cada HU que que fueron definidas, así como las pruebas de aceptación planificadas para el sistema. De esta forma es obtenido un producto funcional probado y listo para entregar al cliente al final de cada iteración como propone XP.

### **3.2. Fase III: Desarrollo**

En esta fase, XP plantea que las HU seleccionadas para ser implementadas se realizan durante el transcurso de la iteración a la que 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 (JOSKOWICZ, 2008).

#### **3.2.1. Tareas de ingeniería**

Las tareas de ingeniería pueden estar descritas por un lenguaje técnico y no ser necesariamente entendibles 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. Se describen algunas de las tareas de ingeniería correspondientes a las HU del sistema, el resto pueden ser consultadas en los anexos. Para una mayor organización, se definen en correspondencia con las iteraciones definidas como se manifiesta a continuación.

### Tareas de ingeniería para la Iteración I

Para la primera iteración, se definieron un total de 7 tareas de ingeniería. Todas, desglosadas a partir de las HU correspondientes al módulo Notificaciones. Se describen algunas de las tareas realizadas en esta iteración.

- **HU Administrar notificaciones**

Tabla 3.1. Tarea de ingeniería # 1

Tarea	
<b>Número de tarea:</b> 1	<b>Número de Historia de Usuario:</b> 1
<b>Nombre de la tarea:</b> Implementar la actualización de la configuración personal del usuario	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.6
<b>Fecha de inicio:</b> 2 de noviembre de 2015	<b>Fecha de fin:</b> 5 de noviembre de 2015
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Se implementa una funcionalidad que permite obtener la configuración de notificaciones, a partir del identificador del usuario que solicita la acción de administrar su configuración para recibir notificaciones. Luego este modifica los campos deseados y se actualiza su configuración personal.	

Tabla 3.2. Tarea de ingeniería # 2

Tarea	
<b>Número de tarea:</b> 2	<b>Número de Historia de Usuario:</b> 1
<b>Nombre de la tarea:</b> Implementar la eliminación de correos electrónicos de forma simultánea	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Fecha de inicio:</b> 5 de noviembre de 2015	<b>Fecha de fin:</b> 7 de noviembre de 2015
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Se implementa una funcionalidad, que permite eliminar un conjunto de correos electrónicos, hayan sido enviados o no, de la base de datos. Estos correos son seleccionados por el administrador del sistema.	

- **HU Enviar notificación al usuario**

Tabla 3.3. Tarea de ingeniería # 3

Tarea	
<b>Número de tarea:</b> 3	<b>Número de Historia de Usuario:</b> 2
<b>Nombre de la tarea:</b> Implementar cuerpo de notificación a enviar	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1.2

Continúa en la próxima página

Tabla 3.3. Continuación de la página anterior

<b>Fecha de inicio:</b> 9 de noviembre de 2015	<b>Fecha de fin:</b> 15 de noviembre de 2015
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Se diseña un procedimiento lógico para elaborar el cuerpo de las notificaciones. Para ello, se tiene en cuenta el usuario que realiza la acción, los usuarios implicados y la acción realizada. Luego, este mecanismo se utiliza para implementar el desarrollo del cuerpo de la notificación, o sea, desarrollar el modelo o entidad.	

Tabla 3.4. Tarea de ingeniería # 4

Tarea	
<b>Número de tarea:</b> 4	<b>Número de Historia de Usuario:</b> 2
<b>Nombre de la tarea:</b> Implementar la comprobación de destinatarios y enviar notificación	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.6
<b>Fecha de inicio:</b> 16 de noviembre de 2015	<b>Fecha de fin:</b> 20 de noviembre de 2015
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Se implementa una funcionalidad que permita chequear si cada destinatario de la notificación generada, tiene configurado el tipo de notificación en cuestión. En caso positivo, se envía la notificación por las vías que se encuentren definidas por el destinatario. En caso contrario, la notificación no se envía, ni se almacena en la base de datos.	

- **Mostrar notificaciones**

Tabla 3.5. Tarea de ingeniería # 5

Tarea	
<b>Número de tarea:</b> 5	<b>Número de Historia de Usuario:</b> 3
<b>Nombre de la tarea:</b> Implementar el mecanismo para mostrar las notificaciones pendientes de un usuario	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Fecha de inicio:</b> 23 de noviembre de 2015	<b>Fecha de fin:</b> 24 de noviembre de 2015
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Al usuario autenticado se le muestra, en la parte superior, a la izquierda de la zona de identificación, una campana donde se lista en un menú desplegable las notificaciones de sistema que tenga pendiente. Se implementa una funcionalidad para que, al seleccionar la notificación del menú, pueda ver los detalles de la notificación seleccionada. Además, se implementa una funcionalidad que ofrezca la posibilidad de marcar todos los mensajes, como leídos.	

### Tareas de ingeniería para la Iteración II

Para una segunda iteración, se elaboraron un total de 7 tareas de ingeniería. La mayoría, desglosadas a partir de las HU correspondientes al módulo Gráficas. Se describen algunas de las tareas realizadas, el resto pueden ser consultadas en los anexos de la investigación.

- **HU Generar gráfica de sesiones de juego**

Tabla 3.6. Tarea de ingeniería # 6

Tarea	
<b>Número de tarea:</b> 6	<b>Número de Historia de Usuario:</b> 5
<b>Nombre de la tarea:</b> Desarrollar vista para la selección de variables a graficar	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha de inicio:</b> 1 de febrero de 2016	<b>Fecha de fin:</b> 2 de febrero de 2016
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Se implementa una vista que permita seleccionar el paciente a analizar, así como la variable o campo a graficar. En dependencia del campo seleccionado, se ofrecen las vías o formas de visualización disponibles para los datos de la variable. Las vías disponibles son: gráfica de barras, líneas, pastel y de dispersión.	

Tabla 3.7. Tarea de ingeniería # 7

Tarea	
<b>Número de tarea:</b> 7	<b>Número de Historia de Usuario:</b> 5
<b>Nombre de la tarea:</b> Implementar la conversión de cadena en formato JSON para generar gráfica de dispersión	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1.2
<b>Fecha de inicio:</b> 3 de febrero de 2016	<b>Fecha de fin:</b> 10 de febrero de 2016
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Se implementa una funcionalidad que transforme los datos a representar en una cadena en formato JSON. De forma tal, que se pueda utilizar para generar una gráfica de dispersión, utilizando la biblioteca <i>javascript</i> Chartist.js	

Tabla 3.8. Tarea de ingeniería # 8

Tarea	
<b>Número de tarea:</b> 8	<b>Número de Historia de Usuario:</b> 5
<b>Nombre de la tarea:</b> Desarrollar vista para generar gráfica a partir de los datos en formato JSON	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2

Continúa en la próxima página

Tabla 3.8. Continuación de la página anterior

<b>Fecha de inicio:</b> 19 de enero de 2016	<b>Fecha de fin:</b> 20 de enero de 2016
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Se desarrolla una vista, para generar una gráfica con los datos dados y la forma de visualización elegida.	

### Tareas de ingeniería para la Iteración III

Para la tercera y última iteración, se definieron un total de 6 tareas de ingeniería. Todas, desglosadas a partir de las **HU** correspondientes al módulo Reportes. Se describen algunas de las tareas realizadas, el resto pueden ser consultadas en los anexos de la investigación.

- **HU Generar reporte: Mis Pacientes**

Tabla 3.9. Tarea de ingeniería # 9

Tarea	
<b>Número de tarea:</b> 9	<b>Número de Historia de Usuario:</b> 7
<b>Nombre de la tarea:</b> Generar reporte con la información de los paciente asociados al doctor X	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1.2
<b>Fecha de inicio:</b> 7 de marzo de 2016	<b>Fecha de fin:</b> 14 de marzo de 2016
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Se implementan varias funcionalidades, que en conjunto convergen en una función principal capaz de organizar en una arreglo la siguiente información sobre los pacientes, atendiendo al doctor que realiza la petición del reporte: <ul style="list-style-type: none"> <li>○ Nombre del paciente</li> <li>○ Tratamientos asignados</li> <li>○ Áreas (Áreas de los videojuegos asociados al tratamiento)</li> <li>○ Porcentaje de cumplimiento de su tratamiento</li> </ul>	

Tabla 3.10. Tarea de ingeniería # 10

Tarea	
<b>Número de tarea:</b> 10	<b>Número de Historia de Usuario:</b> 7
<b>Nombre de la tarea:</b> Desarrollar vista para mostrar la información del reporte: Mis Pacientes	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.8
<b>Fecha de inicio:</b> 14 de marzo de 2016	<b>Fecha de fin:</b> 18 de marzo de 2016
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	

Continúa en la próxima página

Tabla 3.10. Continuación de la página anterior

<b>Descripción:</b> Se desarrolla una vista para mostrar la información recopilada sobre los pacientes del doctor en cuestión. Además, se ofrece la posibilidad de exportar la información a documento de formato PDF.
--

- **HU Generar reporte: Historia Clínica**

Tabla 3.11. Tarea de ingeniería # 11

Tarea	
<b>Número de tarea:</b> 11	<b>Número de Historia de Usuario:</b> 8
<b>Nombre de la tarea:</b> Generar reporte con la información del paciente X para generar documento histórico de sus tratamientos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1.2
<b>Fecha de inicio:</b> 28 de marzo de 2016	<b>Fecha de fin:</b> 4 de abril de 2016
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<p><b>Descripción:</b> Se implementan varias funcionalidades, que en conjunto convergen en una función principal, capaz de organizar en un arreglo, la siguiente información sobre el paciente seleccionado:</p> <ul style="list-style-type: none"> <li>○ Tratamientos asignados</li> <li>○ Áreas (Áreas de los videojuegos asociados al tratamiento)</li> <li>○ Porcentaje de cumplimiento vencido</li> <li>○ Tiempo bajo tratamiento (tiempo transcurrido en días desde que inició el tratamiento hasta la fecha actual)</li> <li>○ Observaciones (observaciones o valoraciones que opina el doctor sobre la ejecución del tratamiento)</li> </ul>	

Tabla 3.12. Tarea de ingeniería # 12

Tarea	
<b>Número de tarea:</b> 12	<b>Número de Historia de Usuario:</b> 8
<b>Nombre de la tarea:</b> Desarrollar vista para mostrar la información del reporte: Historia Clínica	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.8
<b>Fecha de inicio:</b> 4 de abril de 2016	<b>Fecha de fin:</b> 8 de abril de 2016
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<p><b>Descripción:</b> Se desarrolla una vista para mostrar la información recopilada sobre el paciente a analizar por el reporte Historia Clínica, al ser seleccionado este, por el doctor. Además, se ofrece la posibilidad de exportar la información a documento de formato PDF.</p>	

• HU Generar reporte: Cumplimiento de Tratamientos

Tabla 3.13. Tarea de ingeniería # 13

Tarea	
<b>Número de tarea:</b> 13	<b>Número de Historia de Usuario:</b> 9
<b>Nombre de la tarea:</b> Generar reporte con la información sobre todos los tratamientos registrados en el sistema	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.6
<b>Fecha de inicio:</b> 25 de abril de 2016	<b>Fecha de fin:</b> 28 de abril de 2016
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<p><b>Descripción:</b> Se implementan varias funcionalidades, que en conjunto convergen en una función principal, capaz de organizar en un arreglo, la siguiente información sobre los tratamientos existentes en el sistema:</p> <ul style="list-style-type: none"> <li>○ Nombre del tratamiento</li> <li>○ Áreas (Áreas de los videojuegos asociados al tratamiento)</li> <li>○ Doctor que crea el tratamiento</li> <li>○ Cantidad de pacientes asociados</li> <li>○ Cantidad de veces terminado (Número de pacientes que han culminado el tratamiento)</li> <li>○ Porcentaje de cumplimiento (relación entre pacientes asociados y pacientes que terminan el tratamiento)</li> </ul>	

Tabla 3.14. Tarea de ingeniería # 14

Tarea	
<b>Número de tarea:</b> 14	<b>Número de Historia de Usuario:</b> 9
<b>Nombre de la tarea:</b> Desarrollar vista para mostrar la información del reporte: Cumplimiento de Tratamientos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Fecha de inicio:</b> 29 de abril de 2016	<b>Fecha de fin:</b> 30 de abril de 2016
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<p><b>Descripción:</b> Se desarrolla una vista para mostrar la información recopilada sobre los tratamientos a analizar por el reporte: Cumplimiento de Tratamientos. Además, se ofrece la posibilidad de exportar la información a documento de formato PDF.</p>	

Con las tareas de ingeniería definidas, se hace necesario establecer un conjunto de pruebas para comprobar la calidad de la solución implementada. Luego, se analizan estos casos de prueba y se ejecutan, lo que permite medir el nivel de cumplimiento con los objetivos de implementación trazados y el nivel de satisfacción del cliente. Así lo propone [XP](#).

### 3.3. Fase IV: Pruebas

Las pruebas son un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo sistemáticamente. Por esta razón, se deben 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.

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 (ESCRIBANO, 2002).

Se decide realizar las pruebas de aceptación a los módulos implementados, debido a que el objetivo de estas, es verificar que el sistema cumpla con los requisitos establecidos por el usuario. De esta forma se puede obtener el grado de satisfacción del cliente.

#### 3.3.1. Pruebas de aceptación

Las pruebas de aceptación 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 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. A continuación, se muestra una representación de las pruebas de aceptación a realizarse en cada iteración.

#### 3.3.2. Pruebas de aceptación para la Iteración I

Para la primera iteración, se definieron un total de 10 casos de pruebas de aceptación. Todas enfocadas a evaluar la implementación del módulo de Notificaciones. Se describen algunas de las pruebas realizadas, el resto pueden ser consultadas en los anexos de la investigación.

Tabla 3.15. Prueba de aceptación # 1

Caso de prueba de aceptación	
<b>Código:</b> HU1_P1	<b>Historia de Usuario:</b> 1
<b>Nombre:</b> Actualizar configuración de notificaciones de un usuario	
<b>Descripción:</b> Prueba para la funcionalidad: actualizar la configuración de notificaciones de un usuario.	
<b>Condiciones de ejecución:</b>	
<ul style="list-style-type: none"> <li>• El usuario debe estar previamente autenticado.</li> <li>• El usuario debe contar con una configuración para recibir las notificaciones.</li> </ul>	

Continúa en la próxima página

Tabla 3.15. Continuación de la página anterior

<p><b>Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. El usuario entra a su configuración de perfil.</li> <li>2. El usuario selecciona la configuración de notificaciones.</li> <li>3. Marca todas las opciones de notificación vía correo electrónica y ninguna a través del sistema.</li> <li>4. El usuario selecciona la opción de <i>Actualizar</i>.</li> <li>5. Se muestra una alerta indicando que se actualizó correctamente la configuración del usuario.</li> </ol>
<p><b>Resultado:</b> Satisfactorio</p>

Tabla 3.16. Prueba de aceptación # 2

Caso de prueba de aceptación	
<b>Código:</b> HU1_P2	<b>Historia de Usuario:</b> 1
<b>Nombre:</b> Eliminar correos seleccionados	
<b>Descripción:</b> Prueba para la funcionalidad: administrar notificaciones (eliminar correos seleccionados).	
<p><b>Condiciones de ejecución:</b></p> <ul style="list-style-type: none"> <li>• El usuario debe estar previamente autenticado y contar con rol Administrador.</li> </ul>	
<p><b>Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. El administrador entra a la vista para la configuración del módulo Notificaciones.</li> <li>2. Se seleccionan todos los correos correspondientes a la primera página de correos listados.</li> <li>3. Se marca la opción <i>Borrar seleccionados</i>.</li> <li>4. Se muestra una alerta indicando que se eliminaron correctamente los correos seleccionados.</li> </ol>	
<b>Resultado:</b> Satisfactorio	

Tabla 3.17. Prueba de aceptación # 3

Caso de prueba de aceptación	
<b>Código:</b> HU2_P1	<b>Historia de Usuario:</b> 2
<b>Nombre:</b> Asignar paciente a doctor	
<b>Descripción:</b> Prueba para la funcionalidad: enviar notificación al usuario, activada al asignar un paciente a un doctor determinado.	

Continúa en la próxima página

Tabla 3.17. Continuación de la página anterior

<p><b>Condiciones de ejecución:</b></p> <ul style="list-style-type: none"> <li>• El usuario debe estar previamente autenticado y contar con rol de Doctor o el de Paciente.</li> <li>• El doctor a notificar debe existir en el sistema.</li> <li>• El paciente a asignar debe existir en el sistema.</li> <li>• Los usuarios implicados, deben tener configurado, recibir una notificación para la acción: <i>Asignar paciente</i>.</li> </ul>
<p><b>Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. El doctor autenticado, lista los usuarios registrados en el sistema con rol Paciente.</li> <li>2. Se selecciona el primer paciente listado.</li> <li>3. Se marca la opción <i>No asignado</i> para cambiar este estado.</li> <li>4. Se asigna el paciente seleccionado, al doctor en cuestión. En correspondencia se envía automáticamente un correo electrónico y un mensaje del sistema a los dos usuarios implicados, dichos mensajes contienen los detalles de la acción realizada.</li> </ol>
<p><b>Resultado:</b> Satisfactorio</p>

### 3.3.3. Pruebas de aceptación para la Iteración II

Para una segunda iteración, se diseñaron un total de 4 casos de pruebas de aceptación. Todas enfocadas a evaluar la implementación del módulo de Gráficas. Se describen algunas de las pruebas realizadas, el resto pueden ser consultadas en los anexos de la investigación.

Tabla 3.18. Prueba de aceptación # 4

Caso de prueba de aceptación	
<b>Código:</b> HU5_P1	<b>Historia de Usuario:</b> 5
<b>Nombre:</b> Generar gráfica de barras para el campo <i>Tiempo de duración</i>	
<b>Descripción:</b> Prueba para la funcionalidad: generar gráfica de sesiones de juego, específicamente, generar una gráfica de barras.	
<p><b>Condiciones de ejecución:</b></p> <ul style="list-style-type: none"> <li>• El usuario debe estar previamente autenticado y contar con rol de Doctor.</li> <li>• Debe existir al menos un paciente del doctor autenticado, con al menos una sesión de juego registrada, para el videojuego seleccionado.</li> </ul>	

Continúa en la próxima página

Tabla 3.18. Continuación de la página anterior

<p><b>Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. El doctor autenticado, selecciona la opción <i>Estadísticas individuales</i> para el videojuego <i>KeylaxyEyes</i>.</li> <li>2. Como respuesta del sistema, se direcciona a una vista donde selecciona el primer paciente listado.</li> <li>3. Se selecciona el campo <i>Tiempo de duración</i>.</li> <li>4. Se selecciona la opción de generar un gráfico de barras.</li> <li>5. Se genera una gráfica de barras, con los datos de correspondientes a todas las sesiones de juego del paciente, para el videojuego <i>KeylaxyEyes</i>.</li> </ol>
<p><b>Resultado:</b> Satisfactorio</p>

Tabla 3.19. Prueba de aceptación # 5

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU5_P2	<b>Historia de Usuario:</b> 5
<b>Nombre:</b> Generar mapa del campo visual	
<b>Descripción:</b> Prueba para la funcionalidad: generar gráfica de sesiones de juego, específicamente, generar una gráfica de dispersión.	
<p><b>Condiciones de ejecución:</b></p> <ul style="list-style-type: none"> <li>• El usuario debe estar previamente autenticado y contar con rol de Doctor.</li> <li>• Debe existir al menos un paciente del doctor autenticado, con al menos una sesión de juego registrada, para el videojuego seleccionado.</li> </ul>	
<p><b>Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. El doctor autenticado, selecciona la opción <i>Estadísticas individuales</i> para el videojuego <i>KeylaxyEyes</i>.</li> <li>2. Como respuesta del sistema, se direcciona a una vista donde selecciona el primer paciente listado.</li> <li>3. Se selecciona el campo <i>Mapa del campo visual</i>.</li> <li>4. Se selecciona la opción de generar un gráfico de dispersión.</li> <li>5. Se selecciona la primera sesión de juego listada.</li> <li>6. Se genera una gráfica de dispersión, con los datos de correspondientes a la sesión de juego seleccionada, para el videojuego <i>KeylaxyEyes</i>.</li> </ol>	
<p><b>Resultado:</b> Satisfactorio</p>	

### 3.3.4. Pruebas de aceptación para la Iteración III

Para una tercera y última iteración, se confeccionaron un total de 6 casos de pruebas de aceptación. Estas enfocadas a evaluar la implementación del módulo de Reportes. Se describen algunas de las pruebas realizadas, el resto pueden ser consultadas en los anexos de la investigación.

Tabla 3.20. Prueba de aceptación # 6

Caso de prueba de aceptación	
<b>Código:</b> HU7_P1	<b>Historia de Usuario:</b> 7
<b>Nombre:</b> Generar reporte: Mis Pacientes para un doctor determinado	
<b>Descripción:</b> Prueba para la funcionalidad: generar reporte Mis Pacientes, correspondiente a los pacientes del doctor X.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"> <li>• El usuario debe estar autenticado en el sistema y contar con el rol de Doctor.</li> <li>• El doctor en cuestión, debe tener al menos un paciente asignado para poder mostrar información.</li> </ul>	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. El doctor autenticado, selecciona la opción de reporte <i>Mis Pacientes</i>.</li> <li>2. Se genera una tabla con la información reunida por el reporte, sobre los pacientes del doctor en cuestión.</li> <li>3. Se muestra además la opción de visualizar el reporte en formato PDF y exportar el mismo como documento PDF.</li> </ol>	
<b>Resultado:</b> Se muestra una tabla que representa la información reunida por el reporte, sobre los pacientes del doctor en cuestión.	

Tabla 3.21. Prueba de aceptación # 7

Caso de prueba de aceptación	
<b>Código:</b> HU8_P1	<b>Historia de Usuario:</b> 8
<b>Nombre:</b> Generar reporte: Historia Clínica para el primer paciente del doctor autenticado	
<b>Descripción:</b> Prueba para la funcionalidad: generar reporte Historia Clínica.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"> <li>• El usuario debe estar autenticado en el sistema y contar con el rol de Doctor.</li> <li>• El doctor a solicitar el reporte, debe tener al menos un paciente asignado.</li> </ul>	

Continúa en la próxima página

Tabla 3.21. Continuación de la página anterior

<p><b>Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. El doctor autenticado, selecciona la opción de reporte <i>Historia Clínica</i>.</li> <li>2. Como respuesta del sistema, se direcciona hacia una vista donde selecciona el paciente a analizar.</li> <li>3. Se genera una tabla con la información reunida por el reporte, sobre el paciente seleccionado.</li> <li>4. Se muestra además la opción de visualizar el reporte en formato PDF y exportar el mismo como documento PDF.</li> </ol>
<p><b>Resultado:</b> Satisfactorio</p>

Tabla 3.22. Prueba de aceptación # 8

Caso de prueba de aceptación	
Código: HU9_P1	Historia de Usuario: 9
Nombre: Generar reporte: Cumplimiento de Tratamientos para el doctor autenticado	
Descripción: Prueba para la funcionalidad: generar reporte Cumplimiento de Tratamientos.	
<p><b>Condiciones de ejecución:</b></p> <ul style="list-style-type: none"> <li>• El usuario debe estar autenticado en el sistema y contar con el rol de Doctor.</li> <li>• En el sistema debe existir al menos un tratamiento registrado.</li> </ul>	
<p><b>Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. El doctor autenticado, selecciona la opción de reporte <i>Mis Pacientes</i>.</li> <li>2. Se genera una tabla con la información reunida por el reporte, sobre todos los tratamientos que se encuentran registrados en el sistema.</li> <li>3. Se muestra además la opción de visualizar el reporte en formato PDF y exportar el mismo como documento PDF.</li> </ol>	
<p><b>Resultado:</b> Satisfactorio</p>	

### 3.3.5. Análisis de las pruebas de aceptación

Hasta el momento se han desarrollado un total de 20 casos de pruebas de aceptación. Estas pruebas fueron realizada de forma organizada, por cada iteración definida. A continuación, se muestra en gráficas, los porcentos de satisfacción alcanzados en cada iteración.



Figura 3.1. Resultados de las pruebas de aceptación

Como se puede observar en la representación, en la primera iteración se realizaron un total de 10 pruebas, de ellas 9 alcanzaron el nivel de satisfacción esperado alcanzando un 90 %, mientras que una de las pruebas, detectó que se notificaba al paciente de forma directa, sin validar que los campos estuviesen vacíos. Lo que permitió corregir la falla, e incorporar la seguridad de esta operación en el sistema. En tanto, en las iteraciones II y III, se realizaron 4 y 6 pruebas respectivamente, donde todas son evaluadas de resultado satisfactorio. Con estas comprobaciones, se obtiene un 100 % de satisfacción en la iteración final del producto, comprobando el correcto funcionamiento de las funcionalidades implementadas.

### 3.4. Conclusiones del capítulo

En este capítulo se especificó el proceso de implementación del sistema a partir del desglose de las HU en tareas de ingeniería, lo que permitió especificar los procedimientos necesarios para dar cumplimiento a cada HU. Además se definieron y aplicaron las pruebas de aceptación a las funcionalidades de los módulos desarrollados. Estas pruebas permitieron detectar una falla en el sistema y corregirla, lo que posibilitó mejorar la operabilidad del mismo.

---

## Conclusiones

---

Con el desarrollo de la presente investigación se arriba a la siguiente conclusión:

- La implementación de los módulos propuestos incorpora funcionalidades asociadas con notificaciones, visualización gráfica y generación de reportes, permitiendo aumentar la retroalimentación entre el paciente-especialista en la plataforma Medicando.

---

## Recomendaciones

---

A partir de los resultados obtenidos se recomienda:

- Utilizar bibliotecas *javascript* que permitan el uso de *socket* para realizar el envío de las notificaciones en tiempo real.
- Incorporar nuevos reportes basados en las necesidades de los especialistas y teniendo en cuenta la gestión de nuevos videojuegos.

---

## Referencias bibliográficas

---

- ABAD, P. y HUAPAYA, E. 2009. *Guía para la presentación de datos estadísticos*. 2009.
- APRENDIZAJE. 2015. *Plataforma Informática*. 2015. Dirección: <<http://www.reddeaprendizaje.com/inicio/item/47-plataforma-informatica>>.
- BELISARIO, C. 2012. *yiiframeworkenespanol/crugemailer*. 2012. Dirección: <<https://github.com/yiiframeworkenespanol/crugemailer/>>.
- BIRD, A. y HUCKER, J. 2012. *Los SMS reducen un 20 por ciento el absentismo en los hospitales*. 2012. Dirección: <<http://www.esendex.es/blog/post/los-sms-reducen-un-20-el-absentismo-en-los-hospitales/>>.
- CIHI. 2016. *Background on the National Rehabilitation Reporting System (NRS) | CIHI*. 2016. Dirección: <<https://www.cihi.ca/en/types-of-care/hospital-care/rehabilitation/background-on-the-national-rehabilitation-reporting>>.
- DEVELOPERS. 2015. *Charts - Google Developers*. 2015. Dirección: <<https://developers.google.com/chart/>>.
- DÍAZ-LEVICOY, D. 2014. *Un estudio empírico de los gráficos estadísticos en libros de texto de Educación Primaria española*. 2014. Url: <<http://funes.uniandes.edu.co/6385/1/TFMDanilo.pdf>>.
- DORMEHL, L. 2014. *The Five Best Libraries For Building Data Visualizations*. 2014. Dirección: <<http://www.fastcompany.com/3029760/the-five-best-libraries-for-building-data-visualizations>>.
- ESCOBAR, J.; URANGO, A.; CASTRO, L. et al., 2015. Gestión de discusiones y notificaciones en el espacio virtual como base para el trabajo con comunidades de práctica en el contexto educativo. 2015.
- ESCRIBANO, G. 2002. Introducción a Extreme Programming. *Introducción a Extreme Programming*. 2002.
- GILFILLAN, I. 2008. *La biblia de MySQL*. 2008.
- GIONKUNZ. 2015. *Chartist - Simple responsive charts*. 2015. Dirección: <<https://gionkunz.github.io/chartist-js/index.html>>.
- GONZÁLEZ, G. y ALFONSO, Y. 2013. *Sistema de Reportes y apoyo a la toma de decisiones para la Dirección de Alimentación de la Universidad de las Ciencias Informáticas*. 2013. Url: <[http://bibliodoc.uci.cu/RDigitales/2013/septiembre/29/TD\\_06601\\_13.pdf](http://bibliodoc.uci.cu/RDigitales/2013/septiembre/29/TD_06601_13.pdf)>.

- GUTIERREZ, E. 2015. *Plataforma web para la gestión de videojuegos serios de navegador con fines terapéuticos (Medicando)*. 2015. Url: <http://publicaciones.uci.cu/index.php/SC/article/view/1742>).
- HERNÁNDEZ, L. y DARIAS, A. 2013. Desarrollo del componente de notificaciones del Sistema de Información Hospitalaria alas HIS. 2013. Url: [http://bibliodoc.uci.cu/RDigitales/2013/octubre/1/TD\\_06388\\_13.pdf](http://bibliodoc.uci.cu/RDigitales/2013/octubre/1/TD_06388_13.pdf)).
- HIGHCHARTSJS. 2015. *Highcharts JS*. 2015. Dirección: <http://shop.highsoft.com/highcharts.html>).
- JOSKOWICZ, J. 2008. Reglas y prácticas en eXtreme Programming. *Universidad de Vigo*. 2008, págs. 22.
- JQUERYHOUSE.COM. 2015. *25 Best jQuery Notification plugins | jQueryHouse*. 2015. Dirección: <http://jqueryhouse.com/best-jquery-notification-plugins/>).
- LARMAN, C. 2002. *UML y Patrones*. 2da Edición. 2002.
- LEARNINGJQUERY.COM. 2015. *8 Useful JavaScript Libraries to Build Interactive Charts | Learning jQuery*. 2015. Dirección: <http://www.learningjquery.com/2015/04/8-useful-javascript-libraries-to-build-interactive-charts>).
- MICHIGANSA. 2016. *Michigan Ingenieria Informatica S.A*. 2016. Dirección: <http://www.michigan.com.ar/>).
- NETBEANS.ORG. 2015. *NetBeans IDE - Overview*. 2015. Dirección: <https://netbeans.org/features/index.html>).
- NEUROATHOME.ORG. 2015. *NeuroAtHome | Rehabilitación Virtual | Neurorehabilitación Telerehabilitación Kinect*. 2015. Dirección: <http://www.neuroathome.org/p/home.html>).
- PECOS, D. 2002. *PostGreSQL vs. MySQL*. 2002. Dirección: <http://danielpecos.com/documents/postgresql-vs-mysql/#AEN11>).
- PIVA, E.; SCIACOVELLI, L.; ZANINOTTO, M.; LAPOSATA, M. y PLEBANI, M. 2009. Evaluation of effectiveness of a computerized notification system for reporting critical values. *American journal of clinical pathology*. 2009, vol. 131, n.º 3, págs. 432-441.
- QSOFT. 2015. *SALUS - Software para Hospitales, Clínicas y Centros Médicos*. 2015. Dirección: [http://www.softwaresalus.com/descripcion\\_software\\_salus.aspx](http://www.softwaresalus.com/descripcion_software_salus.aspx)).
- QUESADA, A. B. 2012. *Módulos de Estadísticas y Reportes del Sistema de Almacenamiento, Clasificación y Análisis de Noticias*. 2012. Url: [http://bibliodoc.uci.cu/RDigitales/2013/febrero/8/TD\\_05809\\_12.pdf](http://bibliodoc.uci.cu/RDigitales/2013/febrero/8/TD_05809_12.pdf)).
- RAHMAN, S. 2015. *The 15 Best JavaScript Charting Libraries*. 2015. Dirección: <http://www.sitepoint.com/15-best-javascript-charting-libraries/>).

- RAMOS, A. y LEDO, M. 2006. Estrategias de Informatización del Sistema Nacional de Salud. *Scielo Cuba*. 2006.
- ROJAS, M. y MONTIEL, A. 2012. Plataforma para el desarrollo de Sistemas de Información Geográfica para dispositivos móviles en la Universidad de las Ciencias Informáticas. 2012. Url: ([http://bibliodoc.uci.cu/RDigitales/2013/febrero/18/TD\\_05855\\_12.pdf](http://bibliodoc.uci.cu/RDigitales/2013/febrero/18/TD_05855_12.pdf)).
- VIRTUALREHAB.INFO. 2015. *Virtualrehab | Virtual Rehabilitation System*. 2015. Dirección: (<http://www.virtualrehab.info/es>).
- WEBAPPERS.COM. 2014. *PNotify - JavaScript Notifications for Bootstrap or jQuery UI*. 2014. PNotify. Dirección: (<http://www.webappers.com/2014/03/20/pnotify-javascript-notifications-for-bootstrap-or-jquery-ui/>).
- WEBGIS. 2015. *FusionCharts License & Pricing Information*. 2015. Dirección: (<http://www.webgis.csi.it/sandbox/rucapp/lib/FusionCharts/Contents/assets/ui/license.html>).
- WINESETT, J. 2010. *Agile Web Application Development with Yii 1.1 and PHP5*. 2010.
- XUE, Q. y ZHUO, X. 2013. *The Definitive Guide to Yii 1.1*. 2013.

# **Anexos**

## .1. Diagnóstico de la plataforma Medicando

**Medicando** es un producto que fue elaborado bajo la guía de la metodología de desarrollo **XP**, utilizando para el modelado de los artefactos el lenguaje **Lenguaje de Modelado Unificado (UML, por sus siglas en inglés)**. Para implementar la aplicación se empleó el *framework* **Yii** en su versión 1.1.14, el cual permitió el empleo de lenguajes como **HTML5 v5**, **CSS3 v3**, **JavaScript** y **PHP v5**. Para el almacenamiento de los datos se utilizó el **SGBD MySQL** y como servidor web *Apache Server*. A continuación se realiza un análisis completo de la herramienta, donde se evalúan sus características principales, ya que enmarcan el objeto de estudio de la presente investigación.

### .1.1. Principales funcionalidades y comportamiento actual

A continuación, se describe brevemente el comportamiento básico de las principales funcionalidades. Estas se agrupan por paquetes, para un mejor entendimiento de acuerdo al enfoque lógico y al propósito que cumplen dentro del sistema.

- **Gestión de videojuegos:** se encarga de la integración, actualización y eliminación de videojuegos en la plataforma, así como la sincronización de los datos y garantiza el **CRUD (Create, Read, Update and Delete)** de los videojuegos. Controla la posibilidad de descargar o no los videojuegos y permite ejecutarlos directamente en el navegador. Posibilita cargar al servidor videojuegos que estén compactados en formato **Zip**. En cuanto al proceso individual de sincronización, logra almacenar los datos de la salva de las sesiones de rehabilitación con los videojuegos. Cuando se ejecuta el videojuego *online*, sincroniza los datos cuando el doctor entra a la sección *Consultar Estadísticas*. Además, brinda la opción de subir un fichero compactado que contenga el **JSON** (o salva) y ejecutar la sincronización.
- **Generar estadísticas de los videojuegos:** encargada de la generación de las estadísticas grupales e individuales de la aplicación. A partir de los datos almacenados en el sistema, genera un grupo de estadísticas grupales e individuales asociadas a los videojuegos y a los pacientes asociados a estos.
- **Gestionar tratamientos:** encargada de que los doctores puedan crear y actualizar tratamientos en el sistema. Además, permite asignar uno o varios tratamientos a los pacientes asignados a un doctor.
- **Gestionar usuarios:** encargada de la creación, actualización y eliminación de usuarios en el sistema. Además, es responsable del proceso de asignación de roles y permisos en la aplicación. Se hace uso del módulo **CRUGE** para facilitar el control de los usuarios y sus perfiles.

### .1.2. Análisis constructivo de funcionalidades

En lo adelante, se realiza un análisis constructivo de las principales funcionalidades. Se detectan debilidades generales del sistema y se identifican deficiencias específicas en ciertas funcionalidades. Para una mejor organización y entendimiento, se describen los problemas enmarcados en el paquete de funciones correspondiente.

---

- **Debilidades generales**

1. Hasta el momento no es posible realizar operaciones como eliminar y visualizar a un conjunto de elementos de manera simultánea, aumentando el esfuerzo y tiempo de trabajo.
2. La base de datos del sistema no está utilizando las llaves foráneas y el tratamiento desde la aplicación se hace complejo debido a que para eliminar un usuario, hay que realizar el proceso de eliminación en cascada de manera manual. No se utiliza el sistema ofrecido por Yii para la interrelación de tablas de la base de datos.
3. No se implementa un mecanismo de filtrado para encontrar un elemento específico.
4. Los especialistas no cuentan con los permisos para eliminar cualquier información.

- **Gestionar tratamientos:** Los especialistas no tienen la oportunidad de crear un historial de valoraciones a los tratamientos realizados por los pacientes, sirviendo a estos últimos de guía o para autocorrección. Además, no se evidencia una forma sencilla de conocer qué tratamiento está asignado a qué paciente, así como medir el nivel de cumplimiento de estas relaciones. No se encuentra definido un mecanismo para asignar un mismo tratamiento a un grupo de pacientes de manera simultánea.

- **Generar estadísticas de los videojuegos:** Los especialistas se apoyan en la plataforma, con el principal objetivo de dar seguimiento al tratamiento de sus pacientes y personalizar cada uno, enfocado a la necesidad o el avance que se muestre. Para ello, necesitan consultar estadísticamente las variables medidas en el tratamiento. Las estadísticas se encuentran asociadas a los videojuegos integrados en el sistema y se dividen en dos categorías: individuales y grupales.

- **Estadísticas grupales:** el sistema es capaz de generar tablas de *ranking* de usuarios en distintas categorías, que sirven para que los especialistas tengan una visión de los pacientes con mayor o menor rendimiento en el videojuego orientado.
- **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.

Hasta el momento esta funcionalidad se encuentra condicionada para las variables que manejan los videojuegos que se gestionan en el sistema, dando lugar a que no se ofrezca personalización para la graficación dinámica de nuevas variables. Además, los datos a representar corresponden a campos predeterminados según el tipo de gráfico que se seleccione. También, con el incremento de las sesiones de rehabilitación almacenadas, será necesario el filtrado de las estadísticas según una fecha, un período o por paginación, facilitando su selección.

### .1.3. Resultados del análisis de las funcionalidades

1. Los especialistas no tienen la posibilidad de ser informados cuando un paciente ha iniciado o terminado un tratamiento, por lo que tienen que chequear constantemente si existen actualizaciones en este aspecto. Además, no se implementa un mecanismo que informe a los pacientes que se les ha asignado

un nuevo tratamiento o de recomendaciones y valoraciones que les pueda dar un doctor, con respecto al tratamiento que realiza.

2. Los datos que consultan los especialistas se encuentran disponibles solo en la plataforma. Esta organización, no permite que los interesados puedan seleccionar la información que necesitan y exportarla o extraerla para análisis posteriores.
3. No se cuenta con un mecanismo definido de reportes personalizados que permita a los especialistas seleccionar un grupo de variables y que estas sean agrupadas bajo un determinado criterio de forma que facilite el análisis y la toma de decisiones.
4. Se necesita centralizar la información de las acciones realizadas por parte de los pacientes, dígame realización de tratamientos y sesiones de juegos. También se hace necesaria la personalización de los datos a ser consultados, con el objetivo de facilitar el seguimiento de la evolución de los evaluados.

Teniendo en cuenta que actualmente se trabaja en la corrección de deficiencias detectadas por el equipo de desarrollo de la plataforma, la presente investigación se basa en cubrir las necesidades relacionadas con las notificaciones del sistema, reportes y visualización de estadísticas.

## .2. Tareas de ingeniería asociadas a Historias de Usuarios

### .2.1. Tareas de ingeniería para la Iteración I

- **HU Administrar notificaciones**

Tabla 23. Tarea de ingeniería # 15

Tarea	
<b>Número de tarea:</b> 15	<b>Número de Historia de Usuario:</b> 1
<b>Nombre de la tarea:</b> Implementar la eliminación de notificaciones en estado <i>leídas</i>	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Fecha de inicio:</b> 7 de noviembre de 2015	<b>Fecha de fin:</b> 9 de noviembre de 2015
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Se implementa una funcionalidad que permite eliminar todas las notificaciones que se encuentren almacenadas en la base de datos del sistema. Para ello se tiene en cuenta que se encuentren en estado de <i>leídas</i> .	

- **Enviar notificación al paciente**

Tabla 24. Tarea de ingeniería # 16

Tarea	
<b>Número de tarea:</b> 16	<b>Número de Historia de Usuario:</b> 4
<b>Nombre de la tarea:</b> Implementar el envío de mensaje de notificación al paciente	

Continúa en la próxima página

Tabla 24. Continuación de la página anterior

<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Fecha de inicio:</b> 25 de noviembre de 2015	<b>Fecha de fin:</b> 26 de noviembre de 2015
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Se implementa una funcionalidad que permite al doctor seleccionar un paciente en específico. Luego utilizando una vista, permite llenar los siguientes campos y enviar el mensaje al paciente especificado previamente: <ul style="list-style-type: none"> <li>○ Asunto (Obligatorio)</li> <li>○ Mensaje (Obligatorio)</li> </ul>	

## .2.2. Tareas de ingeniería para la Iteración II

- HU Generar gráfica de sesiones de juego

Tabla 25. Tarea de ingeniería # 17

Tarea	
<b>Número de tarea:</b> 17	<b>Número de Historia de Usuario:</b> 5
<b>Nombre de la tarea:</b> Implementar la conversión de cadena en formato JSON para generar gráfica de barras	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Fecha de inicio:</b> 11 de enero de 2016	<b>Fecha de fin:</b> 12 de enero de 2016
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Se implementa una funcionalidad que transforme los datos a representar en una cadena en formato JSON. De forma tal que se pueda utilizar para generar una gráfica de barras utilizando la biblioteca <i>javascript</i> Chartist.js.	

Tabla 26. Tarea de ingeniería # 18

Tarea	
<b>Número de tarea:</b> 18	<b>Número de Historia de Usuario:</b> 5
<b>Nombre de la tarea:</b> Implementar la conversión de cadena en formato JSON para generar gráfica de líneas	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Fecha de inicio:</b> 13 de enero de 2016	<b>Fecha de fin:</b> 14 de enero de 2016
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Se implementa una funcionalidad que transforme los datos a representar en una cadena en formato JSON. De forma tal que se pueda utilizar para generar una gráfica de líneas utilizando la biblioteca <i>javascript</i> Chartist.js.	

Tabla 27. Tarea de ingeniería # 19

Tarea	
<b>Número de tarea:</b> 19	<b>Número de Historia de Usuario:</b> 5
<b>Nombre de la tarea:</b> Implementar la conversión de cadena en formato JSON para generar gráfica de pastel	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.6
<b>Fecha de inicio:</b> 15 de enero de 2016	<b>Fecha de fin:</b> 18 de enero de 2016
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Se implementa una funcionalidad que transforme los datos a representar en una cadena en formato JSON. De forma tal que se pueda utilizar para generar una gráfica de pastel utilizando la biblioteca <i>javascript</i> Chartist.js.	

- **HU Generar documento PDF**

Tabla 28. Tarea de ingeniería # 20

Tarea	
<b>Número de tarea:</b> 20	<b>Número de Historia de Usuario:</b> 6
<b>Nombre de la tarea:</b> Implementar el mecanismo para exportar información a documento PDF	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 15 de febrero de 2016	<b>Fecha de fin:</b> 19 de febrero de 2016
<b>Programador responsable:</b> Juan Gabriel Valdés Díaz	
<b>Descripción:</b> Se implementa una funcionalidad que dado el nombre del documento, contenido del documento y forma de exportar; permita construir o visualizar un documento en formato PDF con los datos adquiridos.	

### .3. Pruebas de aceptación

#### .3.1. Pruebas de aceptación para la Iteración I

Tabla 29. Prueba de aceptación # 9

Caso de prueba de aceptación	
<b>Código:</b> HU1_P3	<b>Historia de Usuario:</b> 1
<b>Nombre:</b> Eliminar correos sin ser seleccionados	
<b>Descripción:</b> Prueba para la funcionalidad: administrar notificaciones (eliminar correos seleccionados). En este caso no se selecciona ningún elemento a eliminar.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"><li>• El usuario debe estar previamente autenticado y contar con rol Administrador.</li></ul>	

Continúa en la próxima página

Tabla 29. Continuación de la página anterior

<p><b>Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. El administrador entra a la vista para la configuración del módulo Notificaciones.</li> <li>2. Se marca la opción <i>Borrar seleccionados</i>, sin seleccionar ningún correo para realizar la acción.</li> <li>3. Se muestra una alerta de error indicando que no se ha seleccionado ningún elemento para la operación solicitada.</li> </ol>
<p><b>Resultado:</b> Satisfactorio</p>

Tabla 30. Prueba de aceptación # 10

Caso de prueba de aceptación	
<b>Código:</b> HU1_P4	<b>Historia de Usuario:</b> 1
<b>Nombre:</b> Eliminar notificaciones con estado <i>leídas</i>	
<b>Descripción:</b> Prueba para la funcionalidad: administrar notificaciones (eliminar notificaciones leídas).	
<b>Condiciones de ejecución:</b>	
<ul style="list-style-type: none"> <li>• El usuario debe estar previamente autenticado y contar con rol Administrador.</li> </ul>	
<b>Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. El administrador entra a la vista para la configuración del módulo Notificaciones.</li> <li>2. Se marca la opción <i>Eliminar notificaciones leídas</i>.</li> <li>3. Se eliminan de la base de datos las notificaciones a nivel de sistema con estado <i>leídas</i>. Luego se muestra un mensaje indicando que se han eliminado correctamente las notificaciones.</li> </ol>	
<b>Resultado:</b> Satisfactorio	

Tabla 31. Prueba de aceptación # 11

Caso de prueba de aceptación	
<b>Código:</b> HU2_P2	<b>Historia de Usuario:</b> 2
<b>Nombre:</b> Iniciar tratamiento por un paciente	
<b>Descripción:</b> Prueba para la funcionalidad: enviar notificación al usuario, activada al iniciar un tratamiento por parte de un paciente determinado.	
<b>Condiciones de ejecución:</b>	
<ul style="list-style-type: none"> <li>• El usuario debe estar previamente autenticado y contar con rol de Paciente.</li> <li>• El doctor a notificar debe existir en el sistema.</li> <li>• El paciente debe tener al menos un tratamiento asignado para poder iniciarlo.</li> <li>• Los usuarios implicados deben tener configurado recibir una notificación para la acción <i>Iniciar tratamiento</i>.</li> </ul>	

Continúa en la próxima página

Tabla 31. Continuación de la página anterior

<b>Pasos de ejecución:</b> <ol style="list-style-type: none"><li>1. El paciente autenticado, lista los tratamientos asignados.</li><li>2. Selecciona la opción <i>Ver</i> del primer tratamiento que no haya observado.</li><li>3. El paciente entra a la vista de detalles del tratamiento seleccionado.</li><li>4. En correspondencia con la acción se envía automáticamente un correo electrónico y un mensaje del sistema a los dos usuarios implicados (el paciente y el doctor que lo atiende), dichos mensajes contienen los detalles de la acción realizada.</li></ol>
<b>Resultado:</b> Satisfactorio

Tabla 32. Prueba de aceptación # 12

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU3_P3	<b>Historia de Usuario:</b> 3
<b>Nombre:</b> Mostrar notificación	
<b>Descripción:</b> Prueba para la funcionalidad: mostrar notificación seleccionada.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"><li>• El usuario debe estar previamente autenticado.</li><li>• El usuario autenticado debe tener al menos una notificación pendiente que visualizar.</li></ul>	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"><li>1. El usuario marca el menú de notificaciones pendientes.</li><li>2. Se despliegan las notificaciones pendientes correspondientes al usuario autenticado.</li><li>3. Se selecciona la primera notificación listada para ver los detalles del mensaje.</li><li>4. Se muestra la información correspondiente en una alerta especial del sistema y se elimina la notificación de la lista pendiente.</li></ol>	
<b>Resultado:</b> Satisfactorio	

Tabla 33. Prueba de aceptación # 13

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU3_P4	<b>Historia de Usuario:</b> 3
<b>Nombre:</b> Marcar todas las notificaciones como <i>leídas</i>	
<b>Descripción:</b> Prueba para la funcionalidad: marcar todas las notificación como <i>leídas</i> .	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"><li>• El usuario debe estar previamente autenticado.</li><li>• El usuario autenticado debe tener al menos una notificación pendiente que visualizar.</li></ul>	

Continúa en la próxima página

Tabla 33. Continuación de la página anterior

<p><b>Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. El usuario marca el menú de notificaciones pendientes.</li> <li>2. Se despliegan las notificaciones pendientes correspondientes al usuario autenticado.</li> <li>3. Se selecciona la opción <i>Marcar todas como leídas</i>.</li> <li>4. Se eliminan todas las notificaciones de la lista pendiente.</li> </ol>
<p><b>Resultado:</b> Satisfactorio</p>

Tabla 34. Prueba de aceptación # 14

Caso de prueba de aceptación	
<b>Código:</b> HU4_P1	<b>Historia de Usuario:</b> 4
<b>Nombre:</b> Enviar notificación a paciente (caso positivo)	
<b>Descripción:</b> Prueba para la funcionalidad: enviar notificación al paciente por parte de un doctor.	
<p><b>Condiciones de ejecución:</b></p> <ul style="list-style-type: none"> <li>• El usuario debe estar previamente autenticado y contar con rol de Doctor.</li> <li>• El doctor autenticado debe tener al menos un paciente asignado.</li> <li>• El paciente a notificar debe tener habilitado la opción de recibir este tipo de notificación en su configuración personal.</li> </ul>	
<p><b>Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. El doctor autenticado, lista los pacientes que tiene asignados.</li> <li>2. Se selecciona el primer paciente listado.</li> <li>3. Se marca la opción <i>Notificar</i> y como respuesta del sistema, se muestra un formulario con los campos a llenar.</li> <li>4. Se llenan los campos: <i>Asunto</i> y <i>Mensaje</i>.</li> <li>5. Se selecciona la opción <i>Enviar</i>.</li> <li>6. Se retorna a la vista de los pacientes del doctor y se muestra un mensaje indicando que se envió la notificación correctamente.</li> </ol>	
<b>Resultado:</b> Satisfactorio	

Tabla 35. Prueba de aceptación # 15

Caso de prueba de aceptación	
<b>Código:</b> HU4_P2	<b>Historia de Usuario:</b> 4
<b>Nombre:</b> Enviar notificación a paciente (caso negativo)	
<b>Descripción:</b> Prueba para la funcionalidad: enviar notificación al paciente por parte de un doctor. En esta ocasión, con valores no válidos.	

Continúa en la próxima página

Tabla 35. Continuación de la página anterior

<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"><li>• El usuario debe estar previamente autenticado y contar con rol de Doctor.</li><li>• El doctor autenticado debe tener al menos un paciente asignado.</li><li>• El paciente a notificar debe tener habilitado la opción de recibir este tipo de notificación en su configuración personal.</li></ul>
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"><li>1. El doctor autenticado, lista los pacientes que tiene asignados.</li><li>2. Se selecciona el primer paciente listado.</li><li>3. Se marca la opción <i>Notificar</i> y como respuesta del sistema, se muestra un formulario con los campos a llenar.</li><li>4. Se llena solo el campo: <i>Asunto</i>.</li><li>5. Se selecciona la opción <i>Enviar</i>.</li><li>6. Se muestra un mensaje de error indicando que existen campos vacíos, por lo que no se envía el mensaje.</li></ol>
<b>Resultado:</b> No satisfactorio

### .3.2. Pruebas de aceptación para la Iteración II

Tabla 36. Prueba de aceptación # 16

Caso de prueba de aceptación	
<b>Código:</b> HU5_P3	<b>Historia de Usuario:</b> 5
<b>Nombre:</b> Generar gráfica de líneas para el campo <i>Puntuación</i>	
<b>Descripción:</b> Prueba para la funcionalidad: generar gráfica de sesiones de juego, específicamente, generar una gráfica de líneas.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"><li>• El usuario debe estar previamente autenticado y contar con rol de Doctor.</li><li>• Debe existir al menos un paciente del doctor autenticado con al menos una sesión de juego registrada para el videojuego seleccionado.</li></ul>	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"><li>1. El doctor autenticado, selecciona la opción <i>Estadísticas individuales</i> para el videojuego <i>KeylaxyEyes</i>.</li><li>2. Como respuesta del sistema, se direcciona a una vista donde selecciona el primer paciente listado.</li><li>3. Se selecciona el campo <i>Puntuación</i>.</li><li>4. Se selecciona la opción de generar un gráfico de líneas.</li><li>5. Se genera una gráfica de líneas con los datos de todas las sesiones de juego del paciente para el videojuego <i>KeylaxyEyes</i>.</li></ol>	
<b>Resultado:</b> Satisfactorio	

Tabla 37. Prueba de aceptación # 17

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU5_P4	<b>Historia de Usuario:</b> 5
<b>Nombre:</b> Generar gráfica de pastel para el campo <i>Efectividad</i>	
<b>Descripción:</b> Prueba para la funcionalidad: generar gráfica de sesiones de juego, específicamente, generar una gráfica de pastel.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"> <li>• El usuario debe estar previamente autenticado y contar con rol de Doctor.</li> <li>• Debe existir al menos un paciente del doctor autenticado con al menos una sesión de juego registrada para el videojuego seleccionado.</li> </ul>	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. El doctor autenticado, selecciona la opción <i>Estadísticas individuales</i> para el videojuego <i>KeylaxyEyes</i>.</li> <li>2. Como respuesta del sistema, se direcciona a una vista donde selecciona el primer paciente listado.</li> <li>3. Se selecciona el campo <i>Efectividad</i>.</li> <li>4. Se selecciona la opción de generar un gráfico de sectores.</li> <li>5. Se selecciona la primera sesión de juego listada.</li> <li>6. Se genera una gráfica de pastel con los datos de correspondientes a la sesión de juego seleccionada para el videojuego <i>KeylaxyEyes</i>.</li> </ol>	
<b>Resultado:</b> Satisfactorio	

### .3.3. Pruebas de aceptación para la Iteración III

Tabla 38. Prueba de aceptación # 18

Caso de prueba de aceptación	
<b>Código:</b> HU6_P1	<b>Historia de Usuario:</b> 6
<b>Nombre:</b> Exportar reporte Mis Pacientes	
<b>Descripción:</b> Prueba para la funcionalidad: generar documento PDF para el reporte Mis Pacientes.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"><li>• El usuario debe estar previamente autenticado y contar con el rol de Doctor.</li><li>• El reporte <i>Mis Pacientes</i> debe contener alguna información que exportar.</li></ul>	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"><li>1. El doctor autenticado, selecciona la opción de reporte <i>Mis Pacientes</i>.</li><li>2. Se genera una tabla con la información reunida por el reporte sobre los pacientes del doctor en cuestión.</li><li>3. Se selecciona la opción de exportar el reporte en formato PDF.</li><li>4. Se genera y descarga el documento con la información, en formato PDF.</li></ol>	
<b>Resultado:</b> Satisfactorio	

Tabla 39. Prueba de aceptación # 19

Caso de prueba de aceptación	
<b>Código:</b> HU6_P2	<b>Historia de Usuario:</b> 6
<b>Nombre:</b> Exportar reporte Historia Clínica	
<b>Descripción:</b> Prueba para la funcionalidad: generar documento PDF para el reporte Historia Clínica.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"><li>• El usuario debe estar previamente autenticado y contar con el rol de Doctor.</li><li>• Debe existir algún paciente asignado al doctor en cuestión.</li><li>• El reporte Historia Clínica debe contener alguna información que exportar.</li></ul>	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"><li>1. El doctor autenticado, selecciona la opción de reporte <i>Historia Clínica</i>.</li><li>2. Como respuesta del sistema, se direcciona hacia una vista donde selecciona el paciente a analizar.</li><li>3. Se genera una tabla con la información reunida por el reporte sobre el paciente seleccionado.</li><li>4. Se selecciona la opción de exportar el reporte en formato PDF.</li><li>5. Se genera y descarga el documento con la información en formato PDF.</li></ol>	
<b>Resultado:</b> Satisfactorio	

Tabla 40. Prueba de aceptación # 20

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU6_P3	<b>Historia de Usuario:</b> 6
<b>Nombre:</b> Exportar reporte Cumplimiento de Tratamientos	
<b>Descripción:</b> Prueba para la funcionalidad: generar documento PDF para el reporte Cumplimiento de Tratamientos.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"> <li>• El usuario debe estar previamente autenticado y contar con el rol de Doctor.</li> <li>• El reporte Cumplimiento de Tratamientos debe contener alguna información que exportar.</li> </ul>	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. El doctor autenticado, selecciona la opción de reporte <i>Cumplimiento de Tratamientos</i>.</li> <li>2. Se genera una tabla con la información reunida por el reporte sobre todos los tratamientos que se encuentran registrados en el sistema.</li> <li>3. Se selecciona la opción de exportar el reporte en formato PDF.</li> <li>4. Se genera y descarga el documento con la información en formato PDF.</li> </ol>	
<b>Resultado:</b> Satisfactorio	