



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
VERTEX, ENTORNOS INTERACTIVOS 3D, FACULTAD 4

MÓDULOS GRAFICACIÓN Y REPORTES PARA LA PLATAFORMA WEB MEDICANDO

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

Autora: Mirdolys Nieves Llopiz Santovenia

Tutores: Ing. Reinaldo Garcia Maturell

Ing. Juan Gabriel Valdés Díaz

Ing. Julio César Espronceda Pérez

La Habana, 2018

Nunca pares, nunca te conformes, hasta que lo bueno sea mejor y lo mejor sea excelente.

Dedico mi tesis a mis padres, Jesús y Maura, por ser las personas más importantes de mi vida, por todas las enseñanzas, por ser mis guías, por cuidarme tanto, por quererme y abrazarme siempre, por darme una sonrisa aunque todo este mal y levantarme el ánimo, por esto y mucho más, los amo, y siempre los amaré. Hoy pueden decir que su niña es toda una ingeniera, la primera de la familia y que siempre estará muy orgullosa de ustedes.

Agradecimientos

Antes de todo, quisiera agradecerles a las personas más importantes de mi vida, que siempre han estado ahí para brindarme todo su apoyo y amor. A mi papito por ser todo para mí, porque a pesar de los problemas y las situaciones que hemos pasado, la vida me ha dado la felicidad de tenerlo hoy caminando, fuerte y feliz. A mimi, por ser mi guía, mi apoyo, por preocuparse y no dejar que me preocupe, por no entender y entenderlo todo, porque sin ella no estaría aquí, por ser todo para mí.

También quiero agradecerle a la Day por todas las cosas que pasamos juntas en la universidad y por soportarme todo este tiempo, que creo, no fue nada comparado con lo que yo he soportado de ella, por ser como una hermana para mí. A la Raque por ser más que una amiga, ser parte de mi familia, ser mi chiquitica, mi prima. A mi primo por ser una de las personas que más amo y odio en este mundo. A mi hermano porque a pesar de todo, sigue siendo muy importante para mí. Además agradezco el apoyo brindado por la familia de Ronald, a mi querida Lisbette y a mi Cosi por todo el apoyo, entrega, comprensión y amor.

Agradecer a mis tutores por el tiempo, la guía y la ayuda que me brindaron, en especial a Juan, por ser sencillamente especial. Al tribunal por todas las críticas constructivas para mejorar la calidad del trabajo. A todos los profes que me formaron durante estos años de universidad, a la profe Zaida y a los profes del Judo. A todos los muchachos del aula por todos los momentos vividos, a mi querido Sergito, se te quiere un mundo, a Andro, Alvaro y todos los Malos Tipos, a los amigos que hice estando en el secretariado de la FEU y a todas las amistades de estos 5 años de universidad. En general a todos los que me ayudaron, malo, bueno o regular, Muchas Gracias.

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales sobre esta, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Mirdolys Nieves Llopiz Santovenia
Autora

Ing. Reinaldo Garcia Maturell
Tutor

Ing. Juan Gabriel Valdés Díaz
Tutor

Ing. Julio César Espronceda Pérez
Tutor

El Centro de Entornos Interactivos 3D (VERTEX) de la Universidad de las Ciencias Informáticas (UCI), ha incursionado en la producción de videojuegos serios de navegador para la rehabilitación, los cuales almacenan informaciones que pueden ser utilizadas en el área de la medicina, para que especialistas lleven un control y seguimiento de la evolución de pacientes. Para ello, el centro ha desarrollado la plataforma web Medicando que permite gestionar videojuegos de navegador para facilitar la rehabilitación de pacientes y permite la retroalimentación al médico del progreso de este. Cuenta con una arquitectura modular que permite incorporar nuevas funcionalidades sin afectar el funcionamiento base. Actualmente, el sistema no permite visualizar gráficas ni reportes de la información generada y almacenada por los videojuegos. A partir de esto, el presente trabajo tiene como objetivo, desarrollar módulos que permitan la visualización de los datos generados y almacenados por los videojuegos que ayude en la toma de decisiones de los doctores sobre la evolución de sus pacientes. Para llevar a cabo la construcción de los módulos fue seleccionado Yii 2.0 como marco de trabajo, NetBeans como IDE y como Sistema de Gestor de Bases de Datos, MySQL. Además, se detallan los procesos de planificación, diseño, implementación y pruebas de los módulos, guiados por la metodología de desarrollo Extreme Programming (XP). Al concluir el trabajo se confeccionaron módulos Gráficos y Reportes, que ofrecen una vía para interpretar y analizar los datos generados por los videojuegos utilizados, en los tratamientos asociados a cada paciente en la plataforma Medicando.

Palabras clave: datos, gráficas, Medicando, módulos, reportes.

Introducción	1
1 Fundamentación teórica	4
1.1 Introducción al capítulo	4
1.2 Principales conceptos	4
1.2.1 Videojuegos	4
1.2.2 Graficación	5
1.2.3 Reporte	5
1.3 Tipos de gráficas	5
1.3.1 Bibliotecas <i>JavaScript</i> para renderizar gráficas en la web	8
1.4 Soluciones existentes	10
1.4.1 Medicando	10
1.4.2 Generadores de reporte a nivel mundial	11
1.4.3 Generadores de reporte a nivel nacional	13
1.4.4 Análisis de los sistemas existentes	14
1.5 Herramientas y tecnologías	14
1.5.1 Marco de trabajo	14
1.5.2 Entorno de desarrollo	15
1.5.3 Sistema Gestor de Base de Datos	15
1.6 Metodología de desarrollo de <i>software</i>	16
1.6.1 SCRUM	16
1.6.2 AUP-UCI	16
1.6.3 XP	17
1.6.4 Justificación de la metodología seleccionada	17
1.7 Conclusiones del capítulo	18
2 Características y diseño del sistema	19
2.1 Introducción al capítulo	19
2.2 Propuesta de Solución	19
2.2.1 Principales funcionalidades	19

2.2.2	Definición de estructuras de configuración y salva de los videojuegos	20
2.3	Proceso de desarrollo empleando la metodología Extreme Programming (XP)	23
2.4	Fase I: Planificación	23
2.4.1	Historias de Usuario	23
2.4.2	Características no funcionales	28
2.4.3	Estimación de esfuerzo por Historia de Usuario	29
2.4.4	Plan de iteraciones	30
2.5	Fase II: Diseño	31
2.5.1	Diseño del sistema	31
2.5.2	Patrones de diseño	32
2.5.3	Tarjetas CRC	34
2.6	Conclusiones del capítulo	37
3	Implementación y pruebas	38
3.1	Introducción al capítulo	38
3.2	Fase III: Implementación	38
3.2.1	Tareas de ingeniería	38
3.3	Fase IV: Pruebas	44
3.3.1	Pruebas unitarias	45
3.3.2	Pruebas de integración	45
3.3.3	Pruebas de aceptación	46
3.3.4	Resultados de las pruebas	54
3.4	Conclusiones del capítulo	55
	Conclusiones	56
	Recomendaciones	57
	Glosario	58
	Acrónimos	59
	Referencias bibliográficas	60
	Apéndices	63
A	Anexos	64
A.1	Ejemplo de las estructuras de configuración y salva en JSON	64
A.2	Pruebas unitarias	65
A.3	Pruebas de aceptación	66

A.4 Métodos de prueba 67

Índice de figuras

1.1	Ejemplo de Gráfico de Barras	6
1.2	Ejemplo de Gráfico de Líneas	7
1.3	Ejemplo de Gráfico de Pastel	7
1.4	Ejemplo de Gráfico de Dispersión	8
2.1	Estructura de atributo json_config del modelo Videojuego.	21
2.2	Estructura JSON para la salva de los videojuegos.	22
2.3	Modelo-Vista-Controlador	31
2.4	Diseño de la arquitectura del sistema	32
2.5	Prototipo de tarjeta CRC	34
3.1	Prueba de integración	46
3.2	Resultados de las pruebas de aceptación	54
A.1	Ejemplo de estructura de atributo json_config	64
A.2	Ejemplo de estructura de atributo json_salva	65
A.3	Ejemplo de algoritmo	66
A.4	Grafo de flujo asociado al algoritmo para renderizar los reportes de los pacientes de un doctor	66
A.5	Métodos de prueba	68

Índice de tablas

1.1	Comparación de bibliotecas <i>JavaScript</i>	9
2.1	Historia de usuario # 1	23
2.2	Historia de usuario # 2	24
2.3	Historia de usuario # 3	25
2.4	Historia de usuario # 4	26
2.5	Historia de usuario # 5	27
2.6	Historia de usuario # 6	28
2.7	Estimación de esfuerzo	30
2.8	Plan de iteraciones	30
2.9	Plan de entrega	30
2.10	Tarjeta CRC # 1	34
2.11	Tarjeta CRC # 2	35
2.12	Tarjeta CRC # 3	36
3.1	Tarea de ingeniería # 1	39
3.2	Tarea de ingeniería # 2	39
3.3	Tarea de ingeniería # 3	39
3.4	Tarea de ingeniería # 4	40
3.5	Tarea de ingeniería # 5	40
3.6	Tarea de ingeniería # 6	40
3.7	Tarea de ingeniería # 7	41
3.8	Tarea de ingeniería # 8	41
3.9	Tarea de ingeniería # 9	41
3.10	Tarea de ingeniería # 10	42
3.11	Tarea de ingeniería # 11	42
3.12	Tarea de ingeniería # 12	43
3.13	Tarea de ingeniería # 13	43
3.14	Tarea de ingeniería # 14	43
3.15	Tarea de ingeniería # 15	43
3.16	Tarea de ingeniería # 16	44

3.17 Tarea de ingeniería # 17	44
3.18 Prueba de aceptación # 1	47
3.19 Prueba de aceptación # 2	47
3.20 Prueba de aceptación # 3	48
3.21 Prueba de aceptación # 4	49
3.22 Prueba de aceptación # 5	49
3.23 Prueba de aceptación # 6	50
3.24 Prueba de aceptación # 7	50
3.25 Prueba de aceptación # 8	51
3.26 Prueba de aceptación # 9	51
3.27 Prueba de aceptación # 10	52
3.28 Prueba de aceptación # 11	52
3.29 Prueba de aceptación # 12	53
3.30 Prueba de aceptación # 13	53
3.31 Prueba de aceptación # 14	53
A.1 Prueba de aceptación # 15	66
A.2 Prueba de aceptación # 16	67

En el mundo de la medicina, el empleo de los videojuegos serios como parte de las nuevas tecnologías, ha ayudado en la comunicación entre los profesionales de la salud y el paciente. Estos videojuegos, recomendados como terapia, sirven para obtener información de la evolución o regresión de pacientes que sufren de algún tipo de dolencia (Agudo, 2013). Esta información se extrapola a un ambiente de interpretación que permite apoyar la toma de decisiones a partir de gráficos e imágenes, debido a que se pueden obtener puntos de vista más claros y resumidos (Tomás-Sábado, 2010).

En Cuba se trabaja en el desarrollo de software para informatizar la sociedad aplicando las [Tecnologías de la Información y las Comunicaciones \(TIC\)](#), así surge la [Universidad de las Ciencias Informáticas \(UCI\)](#), la cual cuenta con 14 centros de desarrollo dentro de los que se encuentra el Centro [Centro de Entornos Interactivos 3D \(Vertex\)](#). Este centro tiene como objetivo desarrollar productos y servicios informáticos asociados a los entornos virtuales interactivos 3D. Una de las directrices del centro es el desarrollo de videojuegos serios con fines terapéuticos enmarcados en la [Telemedicina](#), para la rehabilitación física y cognitiva de las personas que utilizan estas herramientas ([La UCI de un vistazo 2018](#)).

Para dar el control y evolución del uso de los videojuegos aplicados en los pacientes, surge en el año 2015 la plataforma web Medicando. Esta permite que sean asignados videojuegos de rehabilitación a través de tratamientos, que son orientados por los especialistas dependiendo de su especialidad. También controlaba la posibilidad de descargar los videojuegos y ejecutarlos directamente en el navegador y la sincronización de los datos, cuando se jugaba en línea y de forma offline mediante un fichero JSON. Además, permitía generar estadísticas, que estaban sujetas a las variables del unico videojuego registrado, de manera que al ser incorporado uno nuevo, este no podía generar sus estadísticas.

Debido a que la plataforma no contaba con una arquitectura definida y la versión 1.1.14 del *framework* Yii deja de tener soporte al ser liberada la versión 2.0, el Centro decide migrar el sistema a la nueva tecnología y utilizar una arquitectura modular. Medicando en su nueva versión, permite incorporar nuevas funcionalidades sin afectar el funcionamiento base del mismo, contando con la administración de usuarios y el establecimiento de roles. Al realizarse un estudio sobre ella, se detectaron varias deficiencias que deben ser atendidas.

- No cuenta aún con un servicio de graficación para las variables que manejan los videojuegos registrados, debilitando el proceso de personalización a la hora de visualizar dinámicamente los datos de los videojuegos que se incorporen.

- No presenta un mecanismo que genere informes sobre la información manejada en la plataforma para ser utilizada en análisis posteriores por los doctores, debilitando la posibilidad de brindar la mejor asistencia médica al paciente.

Teniendo en cuenta lo antes mencionado se define como **problema de la investigación**: ¿Cómo contribuir a la visualización de la información proporcionada por los videojuegos gestionados en la plataforma web Medicando?

Para ello se tiene como **objeto de estudio** el proceso de desarrollo de módulos de graficación y reporte en plataformas web, comprendido en el **campo de acción**, módulos de graficación y reporte para plataformas web de rehabilitación a distancia.

Para dar solución al problema planteado se define como **objetivo**: desarrollar módulos que permitan la visualización de los datos generados y almacenados por los videojuegos en la plataforma web Medicando, que ayuden en la toma de decisiones de los doctores sobre la evolución de sus pacientes.

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

- Elaboración del marco teórico de la investigación asociado a la generación de gráficas y reportes.
- Selección de las herramientas y tecnologías necesarias para la confección de los módulos propuestos.
- Realización del análisis y diseño de la propuesta de solución.
- Implementación de los módulos de graficación y reportes propuestos.
- Realización de pruebas a los módulos durante la implementación.

Para el desarrollo del trabajo se utilizaron **métodos de investigación científica** dentro de los que se incluyen los métodos teóricos y empíricos que se muestran a continuación:

Métodos Teóricos:

- **Método analítico-sintético**: se utiliza 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.

Métodos Empíricos:

- **Observación científica**: se emplea con el objetivo de observar el funcionamiento de algunas plataformas web orientadas a la generación de estadísticas y reportes. 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 documento tendrá la siguiente **estructura de tesis**:

Capítulo 1-Fundamentación teórica: se realiza una descripción de los conceptos asociados a los mecanismos de reportes y gráficas que utilizan las plataformas virtuales que se encuentran en la web. Se

seleccionan las tecnologías que se utilizarán en el desarrollo de los módulos propuestos con el objetivo de incorporar nuevas funcionalidades asociadas a la generación de gráficas y reportes para la plataforma Medicando.

Capítulo 2-Características y diseño del sistema: se realiza la definición de los objetivos y los servicios que brindarán los módulos de graficación y reportes a desarrollar. Comprende las etapas de planificación y diseño de la solución, donde se definen las principales funcionalidades, para darle cumplimiento a lo requerido por el cliente a través de las historias de usuario, así como la generación de artefactos correspondientes que propone la metodología XP.

Capítulo 3-Implementación y pruebas: se analizan temas referentes a la implementación de la solución, donde se definen las tareas a desarrollar por cada funcionalidad requerida. Se detallan las pruebas unitarias, de aceptación e integración realizadas que propician al cliente conformidad, seguridad y comprobar el correcto desarrollo de las funcionalidades implementadas.

1.1. Introducción al capítulo

En el presente capítulo se refleja la búsqueda y análisis de la información relacionada con los conceptos fundamentales asociados al dominio de la investigación, el estudio del estado del arte de la nueva versión de la plataforma para comprender mejor su entorno y arquitectura. También se describen y analizan las principales herramientas y tecnologías que permitirán llevar a cabo la propuesta de solución.

1.2. Principales conceptos

Para entender mejor los temas que serán abordados en la investigación, se hace necesario relacionar a continuación un conjunto de términos y conceptos asociados al dominio del problema. Al ser la gestión de videojuegos funcionalidad clave para la plataforma, se comienza definiendo este término.

1.2.1. Videojuegos

Los videojuegos son aquellos programas informáticos, normalmente asociados a un hardware específico, que recrea un ejercicio sometido a reglas, donde se deben lograr uno o varios objetivos. Es la interacción de los jugadores mediante la toma de decisiones la que permite que se logren o no los objetivos antes mencionados (Perez y Gardey, 2013).

Los videojuegos se pueden clasificar de dos tipos, los lúdicos, que no representan una enseñanza para el usuario, se enfocan en la diversión y entretenimiento del mismo logrando una mayor dependencia de estos tipos de juegos. Por otra parte se encuentran los juegos serios, los cuales hacen uso de la diversión como modo de formación (López Raventós, 2016).

Un juego serio es un videojuego diseñado que, junto al propio entretenimiento, transmite mensajes o ideas relevantes sobre diversos aspectos no relacionados con la industria de los videojuegos y que entran dentro del ámbito de la ética (Aragon, 2011).

Cada videojuego almacena una serie de variables que sirven para los diferentes datos que definen de cierta forma los objetivos de estos juegos. Estos datos pueden ser utilizados en la representación estadística, para analizar valores que contribuyen a la toma de decisiones.

1.2.2. 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. El primero consiste en el ordenamiento de los datos seleccionados, en filas y columnas, denominado cuadro estadístico (Estrella, 2014). Mientras que 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 (Solano Carrasco y Villalva Zhindón, 2012).

Por otro lado, los datos obtenidos por los videojuegos también pueden ser consultados para el análisis a través de los reportes.

1.2.3. Reporte

Un reporte es una noticia o informe, que puede ser impreso, digital o audiovisual y pretende transmitir información de distintos tipos. Existen diversos tipos de reportes como los divulgativos, persuasivos, informativos y de otros tipos (*Real Academia Española s.f.*).

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 darles un formato determinado a los datos para mostrarlo a través de un diseño más atractivo y a su vez, fácil de interpretar por los usuarios, con la ayuda de gráficos y diagramas debidamente confeccionados (*Definición de Reporte s.f.*).

Antes de realizar un análisis de los sistemas que presentan características similares a la que se desea implementar, se pudo observar que para la representación gráfica de los datos, existen varios tipos de gráficas. Estas permiten obtener varios puntos de vista que aumentan el espectro de análisis, posibilitando una mejor interpretación de la información, por lo que se estudian los más utilizados mundialmente, para ser luego empleados en la solución.

1.3. Tipos de gráficas

Las gráficas o gráficos son las denominaciones de la representación de datos, generalmente numéricos, mediante recursos gráficos, para que se manifieste visualmente la relación matemática o correlación estadística que guardan entre sí. Llamado también **diagrama**, es una representación visual de datos estadísticos por medio de puntos, líneas, barras, polígonos o figuras asociadas a escalas de medición, que permite una fácil comprensión de la información en su conjunto (ABAD y HUAPAYA, 2009).

Los gráficos transmiten de forma inmediata una idea general sobre los principales aspectos de la información, por lo que debe construirse de manera sencilla, de fácil interpretación y suministrar valores aproximados (Tomás-Sábado, 2010). A continuación, se dará una breve explicación de los principales gráficos utilizados.

Gráfico de Barras

Un gráfico de barras es una representación gráfica en un eje cartesiano de las frecuencias de una variable cualitativa o discreta. Se utiliza para representar valores utilizando trazos verticales u horizontales. Este tipo de gráfico permite realizar comparaciones individuales entre dos o más elementos independientemente de la clasificación que exista, gracias a su facilidad de interpretación y construcción (Guerra Cardet y Rufino Nápoles, 2015).

Aunque no existen reglas estrictas, sí se siguen algunas normas generales para este tipo de representación como lo son el ancho de la barra el cual debe ser uniforme para todas las barras del diagrama, la longitud que debe ser proporcional a la cantidad que representa, el espacio de separación entre barras por cada concepto que debe ser constante y por último las barras en estos gráficos pueden disponerse vertical u horizontalmente (ver figura 1.1) (Valdés, Garcia y Ramírez, 2016).

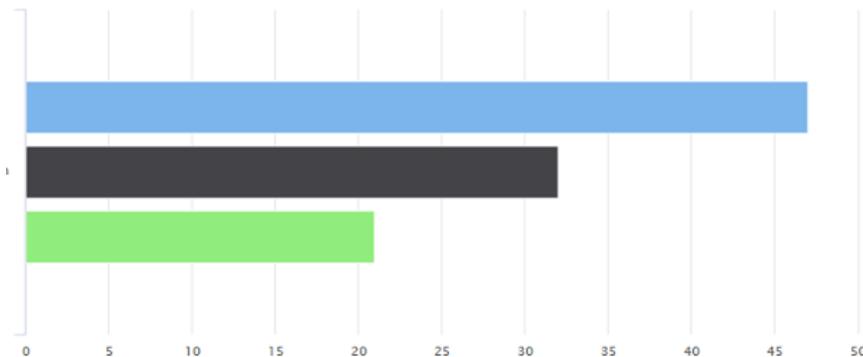


Figura 1.1. Ejemplo de Gráfico de Barras.

Gráfico de Líneas

Es una representación en un eje cartesiano de la relación que existe entre dos variables reflejando con claridad los cambios producidos. Este tipo de gráfico se utiliza para mostrar tendencias en el tiempo, frecuentemente para resaltar la manera en que cambian los datos de situaciones que ocurren en períodos sucesivos. Los valores se representan en dos ejes cartesianos (ver figura 1.2) (Espinosa Robert, Barroso Rodríguez y Gámez Guedes, 2015).

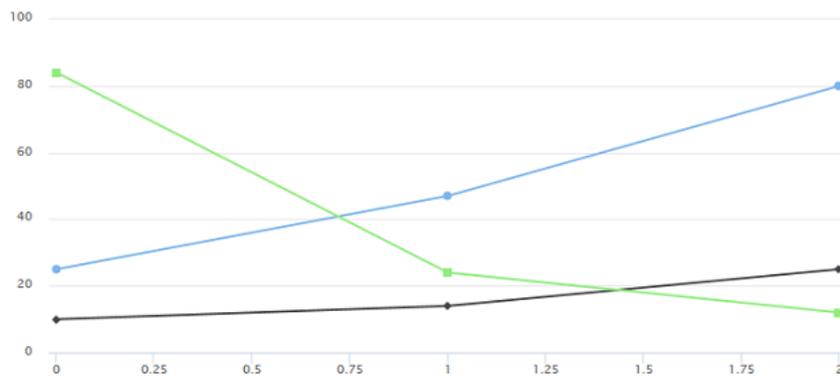


Figura 1.2. Ejemplo de Gráfico de Líneas.

Gráfico de Pastel

Se conoce también como gráfico de torta o circular y constituyen recursos estadísticos que se emplean para representar porcentajes y proporciones de datos cualitativos. Este tipo de gráfico refleja claramente la cantidad que representa cada valor en un hecho (*Gráfica de Pastel s.f.*).

Los gráficos circulares muestran el tamaño de los elementos de una serie de datos, en proporción a la suma de los elementos. Son excelentes para mostrar las relaciones entre el todo y sus partes. Destacan la magnitud del cambio en el tiempo y se pueden utilizar para llamar la atención hacia el valor total en una tendencia (ver figura 1.3) (Hernández Hermosillo y Sánchez Ángeles, 2013).

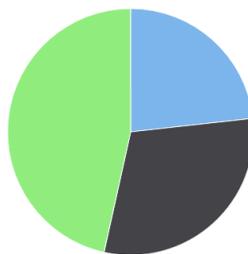


Figura 1.3. Ejemplo de Gráfico de Pastel.

Gráfico de Dispersión

Los gráficos XY o de dispersión, también llamados diagramas de burbujas, muestran las series como un conjunto de puntos. Los valores se representan mediante la posición de los puntos en el espacio del gráfico (Kanashiro Medina et al., 2013). Las categorías, por su parte, mediante diferentes puntos del gráfico, en este tipo, suelen utilizarse para comparar valores distintos de las categorías y evidencian la relación entre los valores numéricos de varios datos (Guerra Cardet y Rufino Nápoles, 2015).

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, además, puede ser utilizado para determinar el grado de

correlación entre las dos variables. Este es muy útil para visualizar en el plano, simultáneamente, hasta tres variables o incluso cuatro, si mediante el color pudiera representarse una cuarta variable cualitativa (ver figura 1.4) (Gea et al., 2014).

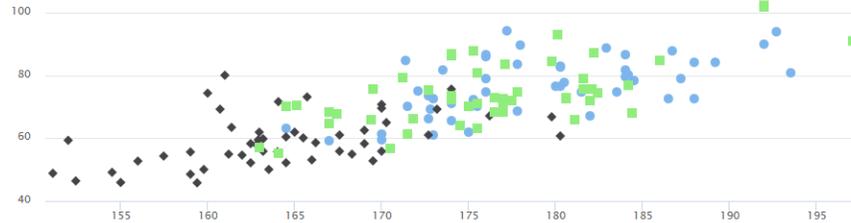


Figura 1.4. Ejemplo de Gráfico de Dispersión.

Análisis de los gráficos

Se pueden encontrar otros tipos de gráficos, siendo estos los más comunes y usados para representar valores científicos, donde lo principal es la información que brinda. Los gráficos son un elemento importante en la comunicación de resultados, ayudando a su interpretación y cumpliendo con el buen lenguaje. Son una función estética que facilita atraer la atención del lector y convencerle del mensaje que se trasmite. Por lo que estos gráficos estudiados, serán los empleados en la implementación de la solución para ayudar a la comprensión y comunicación de la evidencia aportada por los datos (Molinero, 2003). Para ello se hace necesario apoyarse en alguna extensión o biblioteca *JavaScript* que facilite la visualización de los gráficos en la web, lo que permitirá fortalecer la comprensión de variables y el proceso de toma de decisiones por parte de los especialistas.

1.3.1. Bibliotecas *JavaScript* para renderizar gráficas en la web

Las bibliotecas o librerías gráficas son un *software* que permite generar imágenes a partir de modelos matemáticos y patrones de iluminación y textura. Las bibliotecas *JavaScript* facilitan el manejo de las funcionalidades provistas por los navegadores y ayudan a cerrar la brecha de incompatibilidad existente entre los mismos.

En la actualidad, el uso de estas bibliotecas ha ganado gran aceptación, no solo por desarrolladores independientes, sino también por las grandes compañías que las utilizan para añadir dinamismo a sus sitios y desarrollar sus aplicaciones. Básicamente, una biblioteca *JavaScript* es un archivo o conjunto de archivos que al ser incluidos en una página web, proporcionan, por medio de funciones y utilidades implementadas en su interior, facilidades de desarrollo al programador (Guzmán et al., 2013).

A continuación, se realiza un estudio breve de las bibliotecas más comunes que permiten la generación de gráficos, para una mejor visualización de las ventajas que estas proveen y la posterior selección de la más

adecuada según el objetivo de la investigación (ver tabla 1.1). Su análisis se elabora según los parámetros que se describen a continuación.

- **Opciones de graficación:** se describen las distintas opciones que tiene la biblioteca para graficar, los distintos tipos de visualización de la información que posee.
- **Facilidad de uso:** se expone el nivel de complejidad tecnológica que presentan a la hora de trabajar con la biblioteca. Los parámetros descritos anteriormente se rigen bajo una clasificación de baja, media y alta.
- **Compatibilidad con Yii2:** describe la compatibilidad de la tecnología con el *framework* a utilizar. Define la existencia de alguna extensión que facilite el uso de la biblioteca.
- **Licencia:** se manifiesta el tipo de licencia por la cual se rige la tecnología.

Tabla 1.1. Comparación de las bibliotecas *JavaScript* según los parámetros definidos

Bibliotecas	Opciones de graficación	Facilidad de uso	Compatibilidad	Licencia
MorrisJS	Media	Baja	No	BSD
Google Charts	Alta	Media	Si	BSD
Chartist	Alta	Alta	Si	Código Abierto
D3	Alta	Media	No	BSD
HighChart	Alta	Alta	Si	BSD

Justificación de la biblioteca seleccionada

Luego de haber realizado un estudio de las bibliotecas de graficación para apoyar en la visualización de los datos obtenidos de los videojuegos, se concluye que **MorrisJS** no ofrece gran cantidad de característica como gráficos de dispersión, no es compatible con el *framework* a utilizar y su comprensión es mínima; **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; **D3.js** queda descartada ya que no permite desplegar la aplicación que la utilice sin pagar por ella ya que presenta una licencia **BSD** modificada. Por otro lado, **Chartist** no cuenta con una documentación amplia, además, al ser probada en la plataforma no tuvo un resultado satisfactorio; mientras que la biblioteca **HighChart** demostró una gran sincronización con el *framework* Yii 2, soporta más de 24 tipos de gráficos, dentro de los que se encuentran los seleccionados para utilizar en la solución. También permite exportar los gráficos a diversos formatos como PNG, JPG, SVG y PDF. Cuenta con una licencia *Creative Commons Attribution-NonCommercial* 3.0 que permite su utilización en proyectos con fines educativos y cabe destacar la gran comunidad de desarrollo y documentación que presenta esta herramienta.

1.4. Soluciones existentes

En la actualidad, existen varios sistemas informáticos empleados para la generación de reportes. Estas son herramientas complementarias de los sistemas de información, que utilizan una especie de lenguaje transparente para el usuario, donde al realizar consultas a la base de datos, obtiene la información en forma de reporte. A continuación se detallan algunas características de las herramientas para la generación dinámica de reportes y gráficos, donde se comienza analizando la plataforma en cuestión.

1.4.1. Medicando

Es una plataforma web desarrollada por el centro [Vertex](#) para facilitar la rehabilitación de pacientes en los tratamientos que les son asignados en el sistema, donde cada tratamiento debe estar relacionado con al menos un videojuego de rehabilitación. Además, permite la retroalimentación al médico de cómo va evolucionando o involucionando el paciente.

El sistema cuenta con una nueva versión que utilizó para su desarrollo el *framework* Yii en su versión 2.0, la cual condiciona que utilice varios lenguajes de programación como *Hypertext Markup Language (HTML5)*, *Cascade Style Sheet (CSS3)*, *JavaScript* y *PHP*; mientras que la gestión de los datos es administrada utilizando el *Sistema de Gestión de Bases de Datos (SGBD)*, (MySQL) y *Apache Server* como servidor. En esta plataforma los usuarios son identificados según el rol que desempeñan en correspondencia con la interacción que tienen con el sistema, donde estos roles son Paciente, Doctor y Administrador, estableciendo así un control de acceso sobre las funcionalidades disponibles en la plataforma (García Díaz et al., 2017).

Análisis de la situación actual

A partir de un análisis funcional a Medicando 1.0 (Ramos Gutiérrez y O. Correa Madrigal, 2015), se pudo detectar que los videojuegos gestionados eran capaces de almacenar un grupo de variables que luego se podían manipular para facilitar su análisis por parte de los especialistas. Entre las variables más comunes que se manejaban se encuentran las siguientes:

- **Puntuación:** variable que mide el cumplimiento de objetivos del juego, donde se estimula la habilidad de los jugadores en su neuro-cinética, reflejos visuales y enfoque de múltiples puntos de visión, es una variable de tipo entera y acumuladora.
- **Tiempo de respuesta o juego:** con la variable de tiempo se puede medir la agilidad del paciente al cumplir con los objetivos en tiempo o en uno menor. Son datos de tipo real.
- **Niveles:** con esta variable pueden saber la complejidad que va superando el usuario empezando desde la menos compleja hasta la mayor. Utilizan variables estáticas para la representación de cada nivel de juego, generalmente es una variable de tipo entera.
- **Matrices:** para la representación de información, como la matriz de campo visual o atencional. Se representa a través de un arreglo de enteros.

Estas son las variables utilizadas que condicionaban la generación de estadísticas, que permitían a un especialista poder medir la capacidad del jugador al cumplir con los objetivos del juego, luego de realizar un análisis, de forma que represente un tratamiento para el paciente en su rehabilitación.

Una vez obtenidos los datos que generan los videojuegos, se mostraban algunos de ellos a partir de un tipo específico de gráfica, lo que apoyaba en alguna medida el análisis de los doctores. Sin embargo las representaciones visuales eran predeterminadas para las mismas variables del único videojuego registrado. Esto dificultaba que al incorporarse nuevos videojuegos, estos no puedan aportar los datos que generaban, para generar nuevas estadísticas y gráficas. Esto se evidenció en la primera versión de la plataforma, pero al ser rediseñada dejó de presentar tal funcionalidad, por lo que se decide incluirla nuevamente por la importancia que presenta.

Por otro parte, resulta necesario que la plataforma cuente con algún mecanismo que permita consultar la información manejada, para ello se hace uso de los reportes.

Importancia de un sistema de reportes

A través de los reportes se refleja el comportamiento de los diferentes componentes que integran cualquier sistema de información de una empresa o institución, permitiendo el reconocimiento, control y monitoreo de las diversas problemáticas existentes en la misma, para de esa forma apoyar en la correcta toma de decisiones (González E. y Alfonso P., 2013).

En la plataforma Medicando se evidencia la importancia de tener informados a los doctores y pacientes de forma digital, ya que a través de este tipo de visualización de la información, se puede brindar a los médicos la oportunidad de dar seguimiento a la evolución de sus pacientes.

El sistema debe realizar reportes capaces de gestionar gran cantidad de información con gran rapidez y confiabilidad en los datos. Por tanto es necesario contar con reportes personalizados, que se obtengan partiendo de las necesidades y exigencias de los usuarios o sea, que estén constituidos por los aspectos que sean de interés a los doctores. Para obtener un producto de esta calidad es necesario analizar las diferentes tecnologías que intervienen y las características de los generadores de reportes a nivel mundial y nacional.

1.4.2. Generadores de reporte a nivel mundial

Para comprender las salidas de informes especializados en plataformas para la [Telemedicina](#), se observan varias aplicaciones que implementan algún mecanismo de reportes a nivel mundial (Marder, 2016).

Business Intelligence and Reporting Tools

BIRT por sus siglas en inglés, es un proyecto de *software* de código abierto que proporciona la plataforma tecnológica BIRT. Este crea visualizaciones de datos e informes que se pueden incrustar en aplicaciones del cliente web, especialmente los basados en *Java*. Se desarrolla dentro de la Fundación *Eclipse*, un consorcio independiente sin ánimo de lucro y formado por los proveedores de la industria de *software* y una

comunidad de código abierto.

El proyecto está patrocinado por *Innovent Solutions*, Accionar y junto con contribuciones de IBM por sus siglas en inglés *International Business Machines*. Además, tiene el apoyo del Centro de desarrolladores de BIRT. Una de sus ventajas es que está muy respaldado por una gran comunidad de usuarios, lo que permite tener una especie de soporte comunitario, detectando rápidamente incidencias (Peh, Hague y Tatchell, 2011) y (Golfarelli, 2009).

Pentaho

Pentaho ofrece una versión gratuita de su *suite* que incluye análisis de negocios, así como la integración de datos, un diseñador de informes y un mercado de *plugins*, desarrollados por la comunidad para ampliar las capacidades de la plataforma *Pentaho*. La empresa promueve una versión de prueba y la evaluación antes de actualizar el producto comercial al completo, que incluye el análisis visual, el modelado predictivo, cuadros de mando y un elemento móvil (E. N. Hernández y Mosquera, 2012).

La unidad de reportes de *Pentaho* (*Pentaho Reporting*) permite a las organizaciones acceder, dar formato y distribuir fácilmente la información a empleados, clientes y asociados. *Pentaho* provee acceso a fuentes de datos relacionales, procesamiento analítico en línea, en inglés **On-Line Analytical Processing (OLAP)** o basadas en **Extensible Markup Language (XML)**, además de ofrecer varios formatos de salida como PDF, **HTML5**, Excel o hasta texto plano. También permite llevar esta información a los usuarios finales vía web, correo electrónico, portales corporativos o aplicaciones propias. Tiene características como proporcionar funcionalidades crítica para usuarios finales como acceso vía web, informes parametrizados, y otros. Proporciona ventajas a especialistas en informes como acceso a fuentes de datos heterogéneos, capacidad de integración en aplicaciones o portales, definición modular de informes, diseño de informes flexible debido a su entorno de diseño gráfico, capacidad de uso de plantillas, acceso a datos relacionales, **OLAP** y **XML**, además de ser multiplataforma (Patil y Thia, 2013).

Jaspersoft

Jaspersoft es un proveedor de *software* comercial de código abierto centrado en Inteligencia de Negocio. Este incluye la visualización de datos, informes y análisis, este fue adquirido por *Tibco Software*, lo que significa que es un *software* privativo. Su *suite* de *software* de **Business Intelligence (BI)**, que ahora se llama *Tibco Jaspersoft* incluye un diseñador de informes, un motor de informes, un depósito del servidor, la programación de informes y **BI** móvil, junto con los foros de la comunidad y una documentación para el apoyo. También cuenta con ediciones más completas, Profesionales y Empresariales, que incluye características como la visualización interactiva y el informe de diseño interactivo (Bernardino y Tereso, 2013) y (Olszak, 2014).

1.4.3. Generadores de reporte a nivel nacional

Los generadores de reportes que se encuentran a nivel nacional, tienen el objetivo principal de resolver problemas específicos presentados en una entidad que necesite obtener reportes informáticos sobre sus datos, destacándose principalmente el desarrollo de herramientas en la UCI. Esto trae consigo que las herramientas que se desarrollen no sean de carácter general, sino que resuelvan solamente los problemas detectados internamente por los especialistas del equipo de desarrollo, a continuación se enuncian los principales sistemas generadores de reportes y gráficas.

Generador Dinámico de Reportes

El **GDR** desarrollado en el [Centro de Tecnologías de Gestión de Datos \(DATEC\)](#) de la facultad de [Ciencias y Tecnologías Computacionales \(CITEC\)](#) tiene entre sus principales características que es una aplicación para la web, la cual necesita de la instalación de un servidor de aplicaciones web *Apache*, un **SGBD** *PostgreSQL* y un servidor de gráficos preferentemente el *Rich Chap Server*. Su código fuente es el de la herramienta *PHP Reports* y el trabajo con grandes cantidades de volúmenes de datos es deficiente al no tener implementado una solución de paginado para el trabajo con los mismos. Es un sistema multiplataforma y muy empleado por algunos de los servicios internos de la institución. El lenguaje de programación que se emplea para trabajar es PHP, y se cuenta con un grupo de desarrollo, trabajando en generar plantillas en formatos jrxml para lograr una integración con la biblioteca *JasperReports* y poder trabajar además con *Java* (Acosta Ortiz y Sánchez Delgado, 2012).

Olympia

Aplicación desarrollada sobre el *framework* *Symfony* y escrita en *PHP*. Es multiplataforma y soporta imágenes, gráficas y sub-reportes, así como varios orígenes de datos. Proporciona a los usuarios, entre otras opciones, agilizar la toma de decisiones, generar reportes en varios formatos y con gran variedad de opciones en su diseño, marcando una diferencia entre los reportes tradicionales y los reportes dinámicos, objetos de este producto.

Está compuesto por varias aplicaciones entre las que se encuentran el Visor de Reportes, Diseñador de Modelos y el Diseñador de Reporte. Aunque permite abstraerse en parte de los conocimientos relacionados con los gestores de bases de datos, el usuario aún debe poseer conocimientos básicos de los mismos. Además, su entorno de trabajo está estructurado de forma que es difícil guiarse en la creación y generación de reportes. Sólo soporta consultas estándares de SQL, teniendo como lenguaje de programación a *PHP*, no presenta buen rendimiento dentro de las bases de datos *Oracle* y necesita de [Apache Server](#) para poder entrar en funcionamiento (Y. A. M. Hernández y Olivares, 2008).

1.4.4. Análisis de los sistemas existentes

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 sistemas 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 y tienen como ventaja que todos los analizados son de código abierto. Las diferencias radican en que cada uno responde a ciertos requerimientos del sistema de información para el cual se implementó.

Las herramientas estudiadas permiten extraer datos de varias fuentes, que sirven de guía para generar y exportar los informes en varios formatos, donde el más utilizado y común, resulta ser el formato PDF. Teniendo en cuenta el análisis anterior, se decide realizar un mecanismo propio para elaborar los reportes personalizados y atendiendo a las necesidades de los especialistas registrados en Medicando. Destaca la necesidad de utilizar tecnologías libres para el desarrollo de los reportes y se tiene en cuenta la extensión *yii-mpdf* que incorpora el propio marco de trabajo, para facilitar la generación de documentos en formato PDF.

1.5. Herramientas y tecnologías

Una vez realizado un estudio previo en la presente investigación, se asumen y adoptan las herramientas y tecnologías utilizadas por el equipo de desarrollo de la versión 2.0 de la plataforma web Medicando y se consideran algunos cambios realizados por la actual versión (García Díaz et al., 2017). Las principales herramientas a utilizar para llevar a cabo el proceso de desarrollo de la solución se describen a continuación.

1.5.1. Marco de trabajo

Se utiliza Yii en su versión 2.0, este es un *framework* genérico de programación web de alto rendimiento, lo que significa que se puede utilizar para desarrollar todo tipo de aplicaciones web modernas en PHP en poco tiempo. Debido a su arquitectura basada en componentes y a su sofisticada compatibilidad de caché, es especialmente apropiado para el desarrollo de aplicaciones de gran envergadura, como proyectos de comercio electrónico, servicios web compatibles con la arquitectura [Transferencia de Estado Representacional \(REST\)](#) y muchas más ([Yii2 Framework s.f.](#)).

Tiene varias características que lo resaltan con respecto a otros marcos de trabajo:

- Como la mayoría de los marcos de trabajo de [PHP](#), Yii implementa el patrón de diseño [Modelo-Vista-Controlador \(MVC\)](#) y promueve la organización de código basada en este patrón.
- La filosofía de Yii consiste en escribir el código de manera simple y elegante, sin sobre diseñar nunca por el hecho de seguir un patrón de diseño determinado.

- Es extremadamente extensible, puedes personalizar o reemplazar prácticamente cualquier pieza de código de base, como se puede también aprovechar su sólida arquitectura de extensiones para utilizar o desarrollar extensiones distribuibles.

1.5.2. Entorno de desarrollo

Se utiliza *NetBeans* en su versión 8.2, este es un Entorno Integrado de Desarrollo, del inglés *Integrated Development Enviroment* (IDE) disponible para los sistemas operativos *GNU/Linux* y *Windows*. Es un producto libre, gratuito y sin restricciones de uso, que posibilita el desarrollo de aplicaciones haciendo uso de varios lenguajes de programación incluido *JavaScript* y *PHP*.

NetBeans tiene una plataforma de aplicaciones que permite a los desarrolladores crear rápidamente aplicaciones web, de escritorio y móviles. La programación mediante *NetBeans* se realiza a través de componentes de *software* modulares, los cuales están a disposición del usuario, en su página oficial, para conseguir mejoras en las aplicaciones (*Oracle 2013*).

Posee una arquitectura consistente, robusta y un mejor rendimiento en cuanto a tiempo de ejecución y optimización de recursos, lo que permitirá a los desarrolladores la creación de un producto completo. Entre sus características se encuentran la identificación y completamiento de código, resaltado de sintaxis y corrección de errores (*NetBeans s.f.*).

1.5.3. Sistema Gestor de Base de Datos

Se utiliza *MySQL* como sistema de gestión de bases de datos relacional, multihilo y multiusuario que fue desarrollado bajo licencia dual GPL/Licencia comercial por la Corporación *Oracle*. Tiene un gran rendimiento, confiabilidad y facilidad de uso como base de datos para aplicaciones basadas en la web, como *phpBB*, en plataformas (*Linux, Windows, Apache, PHP, Perl*) y por herramientas de seguimiento de errores como *Bugzilla*. Es importante destacar que en la actualidad en el mercado está más valorado en el segmento de desarrollo web (*MySQL s.f.*).

Presenta múltiples ventajas como (*Fossati, 2014*):

- Por ser cliente servidor puedes crear respaldos sin tener que cerrar todos los objetos bloqueados por los usuarios.
- Soporta tablas de hasta 1 Terabyte.
- Su configuración es simple.
- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación de multihilo, lo cual lo hace una base de datos muy veloz.
- Soporta gran cantidad de tipo de datos para las columnas.
- Dispone de API's para gran cantidad de lenguajes (*PHP, C++, Perl, Java*)
- Ofrece gran compatibilidad entre sistemas.

- Proporciona gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad de los datos.

1.6. Metodología de desarrollo de *software*

Para guiar el proceso ingenieril se sigue un enfoque ágil, ya que es flexible, sus proyectos son subdivididos en proyectos más pequeños, incluye comunicación constante con el cliente, es altamente colaborativo y se adapta mejor a los cambios. Con el propósito de seleccionar la metodología ágil en la que se enfocará esta investigación se realiza un estudio a continuación.

1.6.1. SCRUM

La metodología SCRUM para el desarrollo ágil de *software* es un marco de trabajo diseñado para lograr la colaboración eficaz de equipos en proyectos, que emplea un conjunto de reglas y artefactos y define roles generando la estructura necesaria para su correcto funcionamiento.

SCRUM utiliza un enfoque incremental que tiene como fundamento la teoría de control empírico de procesos. Esta teoría se fundamenta en transparencia, inspección y adaptación; la transparencia, que garantiza la visibilidad en el proceso de las cosas que pueden afectar el resultado; la inspección, que ayuda a detectar variaciones indeseables en el proceso; y la adaptación, que realiza los ajustes pertinentes para minimizar el impacto de las mismas.

En SCRUM se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales (Cadavid, Martínez y Vélez, 2013).

1.6.2. AUP-UCI

Agile Unified Process (AUP por sus siglas en inglés) es una metodología de Proceso Unificado que aplica enfoques iterativos e incrementales dentro de un ciclo de vida estructurado. Utiliza una filosofía ágil que se enfoca en la naturaleza de colaboración para el desarrollo de *software*. La UCI toma la iniciativa de hacer una variación de esta metodología, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la entidad.

Esta metodología es una versión simplificada de RUP. Variación AUP-UCI describe de una manera simple y fácil, la forma de desarrollar aplicaciones de *software* de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. Permite adaptarse a los cambios de los requisitos, así como detectar y gestionar los riesgos durante todo el ciclo de vida del proyecto. La metodología Variación AUP-UCI propone tres fases y ocho disciplinas. Estas disciplinas guían el *software* desde su concepción hasta su entrega al usuario final. (Gómez Martínez y Durán Bolaño, 2015).

1.6.3. XP

Es una metodología ágil para el desarrollo de *software* y consiste básicamente en ajustarse estrictamente a una serie de reglas que se centran en las necesidades del cliente para lograr un producto de buena calidad en poco tiempo, centrada en potenciar las relaciones interpersonales como clave para el éxito.

En comparación con las metodologías tradicionales es más rápida, ya que conlleva menos protocolo, lo que evita que existan jerarquías dentro del grupo, la misma se enfoca en resultados a corto plazo es decir los resultados que se van obteniendo a lo largo de la modulación serán verificados al instante y de existir alguna anomalía o falta se harán las correcciones correspondientes. De esta forma rápida es posible obtener los resultados esperados a corto plazo y de manera eficiente el sistema tomará robustez en la utilización de los módulos de manera instantánea (Clemente Menéndez y Rodríguez Cotorruelo, 2013).

XP es la más destacada de los procesos ágiles de desarrollo de *software*. 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. Esta metodología 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 (Marín y Lanier, 2016).

1.6.4. Justificación de la metodología seleccionada

Una vez estudiadas las diferentes metodologías mencionadas, fueron seguidos los pasos que define la metodología XP para guiar el proceso de desarrollo de *software*, adoptando solamente cuatro fases (Planificación, Diseño, Implementación y Prueba), debido a las características que ofrece para soluciones de este tipo.

- **Tiempo de desarrollo:** el método está basado en las mejoras continuas, a base de iteraciones y del desarrollo incremental al estilo espiral permitiendo al equipo de desarrollo dividir el trabajo a realizar en cada una de las iteraciones.
- **Tamaño del equipo:** el objetivo es que el enfoque en parejas sea mayor, en este caso el equipo de desarrollo es conformado por un desarrollador, pero cumple con ser un equipo de desarrollo pequeño, algo que señala esta característica.
- **Corrección de errores:** esta característica es aplicada cada vez que se detectan errores en el desarrollo del producto final, estos errores pueden ser detectados por el propio desarrollador o por el cliente en cada una de las reuniones que se desarrollen.
- **Participación del cliente:** es fundamental que el cliente intervenga en el desarrollo, pero no estará en la sala de desarrollo, se debe asignar a una persona que sea la encargada de tener las reuniones con el cliente de forma constante. Este será quien comunique al equipo los cambios o el seguimiento del proyecto. En el desarrollo de la propuesta de solución, desde el comienzo se realizan reuniones periódicas con el cliente, donde se expone cada avance en el desarrollo y donde se obtienen cada uno

de los cambios sugeridos por el mismo.

- **Reestructuración del código:** una re facturación del código siempre se debe realizar, con esto se simplifica el código, pero no las funciones. Pues regularmente cuando se desarrolla, se agregan cosas que pueden ser innecesarias y que no afectan en el funcionamiento del sistema, estas son precisamente las que hay que re facturar. Para el desarrollo del producto final está es una característica muy aplicada, debido a que solo cuenta con un desarrollador y el principal objetivo es simplificar el código.

1.7. Conclusiones del capítulo

A partir del estudio realizado fue posible arribar a las siguientes conclusiones:

- Las principales formas de representación visual de dato de forma tradicional estudiadas, permitieron seleccionar las formas: gráficos de barra, de línea, de pastel y de dispersión, para visualizar los posibles datos generados por los videojuegos en la plataforma Medicando.
- El análisis de los sistemas existentes a nivel internacional y nacional, permitió concluir que se va a definir un mecanismo propio con tecnologías libres para realizar los reportes de acuerdo a las necesidades de los especialistas de la salud, asociados al sistema web, utilizando la extension *yii-mpdf*.
- El estudio de las principales herramientas y tecnologías permitió definir como marco de trabajo Yii en su versión 2.0, el IDE *NetBeans* 8.2 y *MySQL* como **SGBD**, para el desarrollo de la solución.
- El estudio de las características de las diferentes metodologías, para llevar a cabo una correcta ingeniería de *software*, permitió acoger **XP** como metodología de desarrollo, ya que garantiza la generación de artefactos que son necesarios para la solución a desarrollar.

2.1. Introducción al capítulo

El presente capítulo comprende las etapas de planificación y diseño de los módulos. Se realiza la propuesta de solución en la que se definen las principales funcionalidades a desarrollar, para darle cumplimiento a lo requerido por el cliente a través de las [Historias de Usuario \(HU\)](#), así como la generación de artefactos correspondientes a la etapa de diseño que propone la metodología [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 de la investigación.

Módulo de Graficación: este módulo permitirá visualizar la información almacenada en la plataforma, de manera tal, que los especialistas puedan seleccionar los datos y el tipo de representación mediante un formulario. La forma de representación estará sujeta a cuatro tipos de gráficos: barras, líneas, pastel y dispersión, siempre que la forma seleccionada sea posible.

Módulo de Reporte: este módulo facilitará a los doctores la obtención de informes especializados en correspondencia con la información que maneja la plataforma. Brinda una mejor interpretación de la información para el análisis de los datos generados a partir de la aplicación de tratamientos a pacientes.

2.2.1. Principales funcionalidades

El **Módulo de Graficación** permitirá graficar las variables de los videojuegos según las sesiones registradas. Por otra parte el usuario podrá generar todo tipo de gráficas de las variables que desea de forma general sin estar sujetas a un tratamiento o paciente, y tendrá como principal funcionalidad la siguiente:

- **Generación de gráficas de los videojuegos:** se visualiza la información almacenada por las variables de los videojuegos registrados en la plataforma, dando la posibilidad de seleccionar los datos y el tipo

de representación gráfica si aplica al tipo de información a graficar. Es decir, el usuario identificado con el rol de Doctor podrá acceder al formulario de graficación donde deberá llenar todos los campos solicitados. Una vez seleccionado el tratamiento, en el campo de paciente y videojuego aparecen solo los correspondientes a ese tratamiento, luego a partir del videojuego seleccionado se escoge el atributo a representar y por último el tipo de gráfica a utilizar en el proceso de visualización.

Por otra parte, en el **Módulo de Reportes**, su principal funcionalidad será facilitar a los doctores la obtención de informes especializados, con la opción de poder exportar la información en formato PDF, describiéndose a continuación:

- **Generar reportes de Tratamientos:** el usuario con el rol de Doctor puede ver la información de los pacientes asociados a ese tratamiento, así como el cumplimiento y terminación de este, donde el cumplimiento del presente reporte estará reflejado en por ciento.
- **Generar reportes de pacientes asociados a un doctor:** este reporte muestra la lista de los pacientes asignados a un determinado doctor mostrando en una tabla el nombre del paciente, los tratamientos asignados y el estado de su cumplimiento.
- **Generar reportes de Historia Clínica:** al seleccionar un paciente se genera un reporte personal que muestra los datos específicos del paciente y adicionalmente, se muestra una tabla con los tratamientos que ha realizado y la fecha en que comenzó estos tratamientos. También contendrá el progreso independiente del paciente en cada videojuego asociado al tratamiento analizado. Por último, muestra el estado de cumplimiento general del tratamiento en cuestión.
- **Generación de reportes de Videojuegos:** estos serán para identificar los más usados por los doctores para un tratamiento, así podrán asignar los que cumplan una mayor y eficaz función dependiendo del tipo de rehabilitación del paciente. En este reporte se muestran los videojuegos registrados en la plataforma, la cantidad de tratamientos que hacen referencia al mismo, la cantidad de pacientes que utilizan el videojuego en sus tratamientos y el porcentaje de eficiencia del videojuego. Este último parámetro está dado por la relación de pacientes que han culminado sus sesiones con el videojuego y la cantidad de pacientes que lo utilizan.

2.2.2. Definición de estructuras de configuración y salva de los videojuegos

Para una mayor comprensión y adecuado funcionamiento de los módulos Graficación y Reportes se deben especificar los procesos de negocio implementados, a continuación se brindan detalles de estos.

Estructura para registrar videojuegos en la plataforma Medicando

Con el fin de ofrecer y facilitar la visualización gráfica de los datos que generan los videojuegos, se hace necesario especificar una estructura que permita sincronizar estos datos y salvarlos en el sistema. Al registrarse un videojuego y estos puedan ser asignados a los tratamientos, se generará una tabla en la base de datos que recopilará todos los datos del videojuego. Esto facilitará realizar operaciones de consulta y análisis,

capaces de proporcionar información útil para que los doctores realicen un seguimiento de la evolución de los pacientes. A continuación se describe la estructura definida asociadas a los videojuegos que se registren en Medicando (ver figura 2.1).

```
{
  "table_config": [
    {
      "column_name": "Nombre del atributo",
      "column_type": "Tipo de dato",
      "column_label": "Etiqueta para atributo",
      "charts": ["Lista de tipos de gráficas"]
    }, (...)
  ]
}
```

Figura 2.1. Estructura de atributo **json_config** del modelo Videojuego.

Este atributo deberá estar en formato [JavaScript Object Notation \(JSON\)](#), el cual es un formato ligero para el intercambio de datos. Es una cadena de notación literal de objetos de *JavaScript* que no requiere el uso de [XML](#). Este puede representar dos tipos de estructuras, un conjunto de pares (clave, valor) encerrado entre llaves {}, separando la clave del valor por los dos puntos (:), y separando cada par del siguiente por la coma (,). También un conjunto ordenado de valores encerrado entre corchete ([]), y separando cada valor del siguiente con la coma (Nolan y Lang, 2014). A continuación se describe el formato propuesto:

- **table_config**: índice por el cual se carga la configuración para generar la tabla de estadísticas asociada al juego en cuestión. Carácter obligatorio.
- **column_name**: índice por el cual se define el nombre de la columna en la tabla de estadísticas. Carácter obligatorio.
- **column_type**: índice por el cual se define el tipo de dato columna en la tabla de estadísticas. Carácter obligatorio. Los tipos de datos pueden ser [*int | integer*, *boolean | bool*, *date*, *json*, *string*].
- **column_label**: índice que permite utilizar un nombre específico para representar los datos del atributo *column_name* en la vista de estadísticas. Carácter obligatorio.
- **charts**: índice que se utiliza para ofrecer los tipos de gráficos a utilizar para visualizar los datos almacenados en la tabla de estadísticas. Carácter Opcional. Los tipos de gráfica disponibles son [*bar*, *line*, *pie*, *scatter*].
- **(...)**: se repite la estructura para cada atributo que se desee registrar del fichero salva del videojuego.

Una vez terminado con el tratamiento, o la sesión de juego, el paciente deberá acceder al sistema y subir la salva del videojuego, este fichero deberá tener una estructura compatible con la definida en la plataforma, para poder registrar los datos y procesarlos para análisis posteriores.

Estadísticas asociadas a los videojuegos

Uno de los principales servicios que brinda el Módulo de Graficación es la generación de estadísticas asociadas a los videojuegos integrados en el sistema. Estas estadísticas servirán como fuente para que los doctores conozcan el desempeño de los pacientes en sus tratamientos y adaptar los tratamientos pautados a sus pacientes de acuerdo con su evolución.

Correspondiente a la estructura de registro, los videojuegos deben seguir un estándar para salvar sus datos, estos deben estar sincronizados con los atributos definidos en la estructura antes mencionada. A continuación se describe la estructura de los ficheros salva que deben seguir los videojuegos en la figura 2.2.

```
{
  "lista_jugadores": [
    {
      "session_id": "id",
      "nombre_jugador": "Nombre",
      "atributo1": "valor1",
      "atributoX": "valorX",
    }, (...)
  ]
}
```

Figura 2.2. Estructura JSON para la salva de los videojuegos.

A continuación se evidencian los detalles de la estructura definida.

- **lista_jugadores:** índice que permite cargar la lista de las sesiones. Carácter obligatorio.
- **session_id:** índice que especifica el identificador secuencial para las sesiones jugadas por cada jugador. Carácter obligatorio.
- **nombre_jugador:** índice que especifica el nombre del jugador, este tiene que ser el mismo nombre del paciente registrado en el sistema para poder sincronizar el cumplimiento de los tratamientos. Carácter obligatorio.
- **atributo1:** índice del primer atributo a sincronizar con las estadísticas del videojuego. Carácter opcional.
- **atributoX:** indica que pueden ser varios atributos por cada sesión de juego.

Es muy importante destacar que los nombres de los atributos de las sesiones de juego deben ser exactamente iguales al definido en la propiedad *column_name* especificada al registrar la estructura del videojuego correspondiente.

En el anexo A.1 se muestra un ejemplo de estructura para un videojuego que va a registrar los atributos *score*, *time*, *skills* y *visual_fields*, y la estructura de la salva asociada a ese videojuego y sus variables.

2.3. Proceso de desarrollo empleando la metodología XP

El ciclo de vida de un proyecto que utilice la metodología XP incluye, entender lo que el cliente necesita, estimar el esfuerzo, crear la solución y entregar el producto final al cliente. Sin embargo, XP propone un ciclo de vida dinámico, donde se admite expresamente que, en muchos casos, los clientes no son capaces de especificar sus requerimientos al comienzo de un proyecto. Por esto, se trata de realizar ciclos de desarrollo cortos (llamados iteraciones), con entregables funcionales al finalizar cada ciclo. En cada iteración se realiza un ciclo completo de planificación, diseño, desarrollo y pruebas, pero utilizando un conjunto de reglas y prácticas que caracterizan a la metodología XP (Ramos Gutiérrez y O. Correa Madrigal, 2015).

2.4. Fase I: Planificación

Es la fase en la que se define el alcance general del proyecto. En esta fase, se define lo que necesita mediante la redacción de sencillas historias de usuarios y se estiman los tiempos de desarrollo en base a esta información. Las estimaciones realizadas en esta fase son primarias, ya que estarán basadas en datos de muy alto nivel, y podrían variar cuando se analicen más en detalle en cada iteración (Torres Mirabales, M. O. Correa Madrigal y Garcia Maturell, 2015).

2.4.1. Historias de Usuario

Uno de los artefactos más importantes que genera la metodología XP son las HU. Estas tienen el mismo propósito que los casos de uso y son confeccionadas por el cliente. Las mismas son la técnica para especificar los requisitos del *software*, expresan su punto de vista en cuanto a las necesidades del sistema, son descripciones cortas y escritas en el lenguaje del usuario sin terminología técnica. Cada HU es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. Otra de sus características es que solamente proporcionan los detalles sobre la estimación del riesgo y cuánto tiempo tomará su implementación (Bustamante y Rodríguez, 2014), a continuación se presentan las principales HU que serán implementadas en el sistema.

Tabla 2.1. Historia de usuario # 1

Historia de usuario	
Número: 1	Nombre: Generar gráficas de los videojuegos
Usuario: Doctor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Mirdolys Nieves Llopiz Santovenia	

Continúa en la próxima página

Tabla 2.1. Continuación de la página anterior

<p>Descripción: Permite 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, tratamiento, videojuego 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 elige la deseada. Dentro de las gráficas que se manejan se encuentran:</p> <ul style="list-style-type: none"> • Gráfica de barras, • Gráfica de líneas, • Gráfica de pastel, • Gráfica de dispersión.
<p>Observaciones: El usuario con el rol de doctor será el único autorizado para acceder a este servicio.</p>
<p>Interfaz:</p> 

Tabla 2.2. Historia de usuario # 2

Historia de usuario	
Número: 2	Nombre: Generar reporte de Tratamientos
Usuario: Doctor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
<p>Descripción: Permite generar un reporte de los tratamientos asignados en el sistema donde se tienen en cuenta la siguiente información:</p> <ul style="list-style-type: none"> • Nombre (nombre del tratamiento). • Pacientes asociados. • Sesiones cumplidas por videojuego (cantidad de sesiones cumplidas con respecto al total). • Progreso (relación entre la cantidad de sesiones totales y las cumplidas). • Estado del tratamiento. <p>La información tendrá la opción de poder ser exportada en formato PDF.</p>	
Observaciones: El usuario con el rol de doctor será el único autorizado para acceder a esta funcionalidad.	

Continúa en la próxima página

Tabla 2.2. Continuación de la página anterior

Interfaz:

Pacientes Asociados	Fecha de Comienzo	Sesiones por Videojuego	Progreso	Estado del Tratamiento
Paciente 1	2018-1-2	(Videojuego 1 - 0 de 5)	0%	Pendiente
Paciente 2	2018-1-5	(Videojuego 1 - 1 de 5)	20%	Pendiente

Tabla 2.3. Historia de usuario # 3

Historia de usuario	
Número: 3	Nombre: Generar reporte de Historia Clínica
Usuario: Doctor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
<p>Descripción: Permite generar un reporte con la información general del paciente seleccionado y además se tienen en cuenta los siguientes datos:</p> <ul style="list-style-type: none"> • Tratamientos asignados (nombre del tratamiento) • Fecha de comienzo • Videojuego • Sesiones definidas • Sesiones cumplidas • Progreso (relación entre la cantidad de sesiones totales y las cumplidas por videojuego) • Estado (estado general del tratamiento) • Progreso General (relación entre la cantidad de sesiones totales y las cumplidas) <p>La información tendrá la opción de poder ser exportada en formato PDF.</p>	
<p>Observaciones: Para visualizar tal reporte es necesario haber seleccionado el paciente. El usuario con el rol de doctor será el único autorizado para acceder a esta funcionalidad.</p>	

Continúa en la próxima página

Tabla 2.3. Continuación de la página anterior

Interfaz:

Nombre del Paciente Exportar

Nombre Correo Dirección Teléfono Móvil Teléfono Fijo Fecha de Nacimiento Género	Mirdolys mnlopiz@uci.cu ave 51, apto10, edf 8, Marianao No definido No definido 1995-4-25 Femenino
---	--

Nombre	Fecha de Comienzo	Videojuego	Sesiones Definidas	Sesiones Cumplidas	Progreso	Progreso General	Estado
Tratamiento 1	2018-2-1	Videojuego 1	10	2	20%	20%	Pendiente
		Videojuego 2	5	1	20%		

Tabla 2.4. Historia de usuario # 4

Historia de usuario	
Número: 4	Nombre: Generar reporte de Pacientes
Usuario: Doctor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
<p>Descripción: Permite generar un reporte de los pacientes que le son asignados al doctor autenticado, donde se tienen en cuenta la siguiente información:</p> <ul style="list-style-type: none"> Nombre (nombre del paciente). Tratamiento (tratamientos por paciente). Fecha de comienzo (fecha de comienzo del tratamiento). Progreso (por ciento del cumplimiento del paciente, dado por la cantidad de sesiones definidas y las realizadas) <p>La información tendrá la opción de poder ser exportada en formato PDF.</p>	
<p>Observaciones: Para que se realice tal reporte el doctor debe de tener asignado al menos un paciente y este tener un tratamiento iniciado, además de estar autenticado. El usuario con el rol de doctor será el único autorizado para acceder a esta funcionalidad.</p>	

Continúa en la próxima página

Tabla 2.4. Continuación de la página anterior

Interfaz:



Nombre	Tratamientos	Fecha de Comienzo	Progreso
Paciente 1	Tratamiento 1	2018-3-4	0%
	Tratamiento 2	2018-1-2	0%
Paciente 2	Tratamiento 2	2018-1-5	20%

Tabla 2.5. Historia de usuario # 5

Historia de usuario	
Número: 5	Nombre: Generar reporte de Videojuegos
Usuario: Doctor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
<p>Descripción: Permite generar un reporte de los videojuegos que son asignados a pacientes como parte del tratamiento donde se tienen en cuenta la siguiente información:</p> <ul style="list-style-type: none"> • Nombre (nombre del videojuego) • Tratamientos (cantidad de tratamientos a los que fue asignado) • Pacientes (cantidad de pacientes que les fue asignado) • Eficiencia (donde se muestra un por ciento de calidad de forma general al ser utilizado en tratamientos) <p>La información tendrá la opción de poder ser exportada en formato PDF.</p>	
Observaciones:	

Continúa en la próxima página

Tabla 2.5. Continuación de la página anterior

Interfaz:



Videojuego	Tratamientos	Pacientes	Eficiencia
Videojuego 1	2	5	99
Videojuego 2	1	3	95

Tabla 2.6. Historia de usuario # 6

Historia de usuario	
Número: 6	Nombre: Generar documento PDF
Usuario: Doctor	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
Descripción: Permite generar un documento en formato PDF a partir de la información dada.	
Observaciones: La información que contendrá el documento PDF será la generada por los reportes.	
Interfaz:	

2.4.2. Características no funcionales

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Estos requisitos a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema.

En el caso de los módulos de Graficación y Reportes, se asumen y adoptan las herramientas y tecnologías utilizadas por el equipo de desarrollo de la versión 2.0 (García Díaz et al., 2017). Por tanto, para que la aplicación sea más atractiva, confiable y usable, se describen a continuación los requerimientos mínimos con los que debe contar el sistema.

1. Requerimientos de *Software*

- **Cliente:** Sistema Operativo *Windows 7* o superior, *Linux*, navegador web estándar con capacidad de interpretación de *JavaScript* y *CSS3* compatible con *Internet Explorer 7* o superior, *Chrome*, *Firefox 51.0.1* o superior.
- **Servidor:** Sistema Operativo *Windows 10* o superior, *Linux* (cualquier distribución), servidor web *Apache 2.4.9*, lenguaje de programación *PHP 5.4.0* o superior y *SGBD MySQL 5.6.16* o superior. Se recomienda la utilización del paquete de programas *XAMPP 1.8.3*, ya que es multiplataforma e incluye *Apache*, *PHP* y *MySQL* por defecto.

2. Requerimientos de *Hardware*

- **Cliente:** Procesador Pentium III, 256 MB de RAM, 100 MB de disco duro. En caso de los móviles con sistema Androide y 1GB de RAM.
- **Servidor:** Procesador Pentium IV a 2.8 GHz, 1 GB de RAM, 10 GB de disco duro.

3. Requerimientos de *Apariencia*

La aplicación incluirá animaciones en las gráficas, que generan dinamismo, se podrá ver en distintas resoluciones de pantalla y para diferentes dispositivos como los móviles modernos (*Tablet*, *iPhone*, *Smartphone*) y las computadoras (*laptop*, *notebook*, PC de escritorio). Tendrá diferentes tonalidades de color guiados por el verde en correspondencia con los colores representativos de la estrategia marcaria de los productos de la *UCI* destinados a la salud, *XAVIA*.

4. Requerimientos de *Diseño e Implementación*

Se hace uso de *Yii* en su versión 2.0 como marco de trabajo, el IDE *NetBeans 8.2* y *MySQL* como *SGBD*.

5. Requerimientos de *Seguridad*

- **Confidencialidad:** Existencia de distintos roles que establecen que la información sólo sea vista por aquellos usuarios que posean los privilegios, suficiente restricción de la ejecución de acciones a usuarios sin credenciales y verificación de que el usuario esté autenticado antes de realizar alguna acción.
- **Integridad:** Validación de los datos en el servidor para evitar estados inconsistentes. La información manejada por el sistema estará protegida del acceso y divulgación no autorizada.

6. Requerimientos de *Rendimiento*

La aplicación podrá general los reportes en PDF y poder visualizar las gráficas en un tiempo menor a 10 segundos.

2.4.3. Estimación de esfuerzo por Historia de Usuario

Con el objetivo de obtener un correcto desarrollo del sistema, se realiza la estimación de esfuerzo que arroja cada *HU* la cual se muestra a continuación en la tabla 2.7.

Tabla 2.7. Estimación de esfuerzo por HU

No.	Historia de Usuario	Puntos Estimados (semanas)
1	Generar gráficas de los videojuegos	4
2	Generar reporte de Tratamientos	2
3	Generar reporte de Historia Clínica	3
4	Generar reporte de Pacientes	2
5	Generar reporte de Videojuegos	2
6	Generar Documento PDF	1

2.4.4. Plan de iteraciones

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

Plan de duración de las iteraciones

El siguiente 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 a continuación en la tabla 2.8.

Tabla 2.8. Plan de duración de las iteraciones

Iteraciones	No.	Historias de Usuario	Duración
1	1	Generar gráficas de los videojuegos	4
2	2	Generar reporte de Tratamientos	2
	3	Generar reporte de Historias Clínicas	3
3	4	Generar reporte de Pacientes	2
	5	Generar reporte de Videojuegos	2
	6	Generar Documento PDF	1

Plan de entrega

En el plan de entrega que se plantea a continuación se hace una propuesta de la fecha aproximada en que se harán versiones al sistema al finalizar cada iteración en la fase de implementación (ver tabla 2.9).

Tabla 2.9. Plan de entrega

Entregable	1ra Iteración	2da Iteración	3ra Iteración
Módulo	versión 0.3	versión 0.7	versión 1.0
Fecha	(25-01-2018)	(15-03-2018)	(24-05-2018)

2.5. Fase II: Diseño

Para organizar estructuralmente los elementos se utilizan los patrones arquitectónicos. A continuación, se describe el patrón usado en la arquitectura propuesta, que fue definido por el equipo de desarrollo de la actual versión.

Como patrón arquitectónico se propone el **MVC** (ver figura 2.3). Este patrón es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. A partir del uso de marcos de trabajo basados en el patrón **MVC** se puede lograr una mejor organización del trabajo y mayor especialización de los desarrolladores y diseñadores (Javier J Gutiérrez, 2014).

Elementos del patrón:

- **Modelo:** datos y reglas de negocio.
- **Vista:** muestra la información del modelo al usuario.
- **Controlador:** gestiona las entradas del usuario.

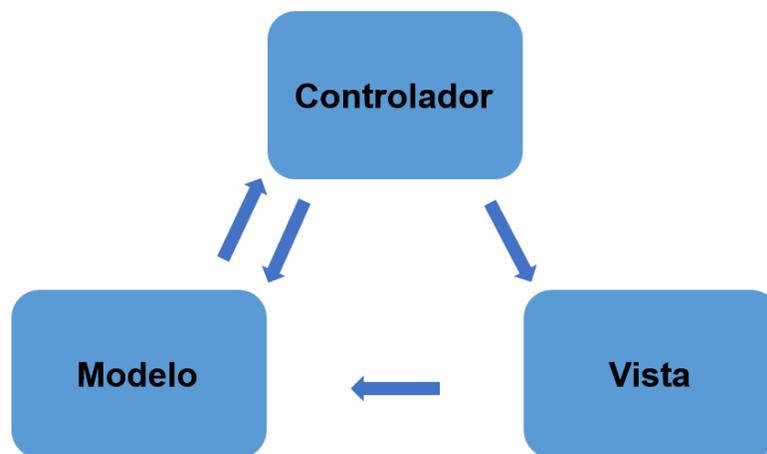


Figura 2.3. Modelo-Vista-Controlador

2.5.1. Diseño del sistema

La propuesta de solución estará basada de igual forma en el patrón arquitectónico mencionado. La figura 2.4 muestra la representación arquitectónica de los módulos de Graficación y Reportes integrados con la plataforma.

Como se muestra en la figura anterior, los módulos de Graficación y Reportes se ubican dentro de *modules*, que es el paquete donde se registran los módulos del sistema. El controlador en ambos casos son los encargados de renderizar las vistas y ejecutar todas las funcionalidades para visualizar los datos, en el caso del módulo de Graficación, no es necesario una clase modelo. La extensión para renderizar gráficas es

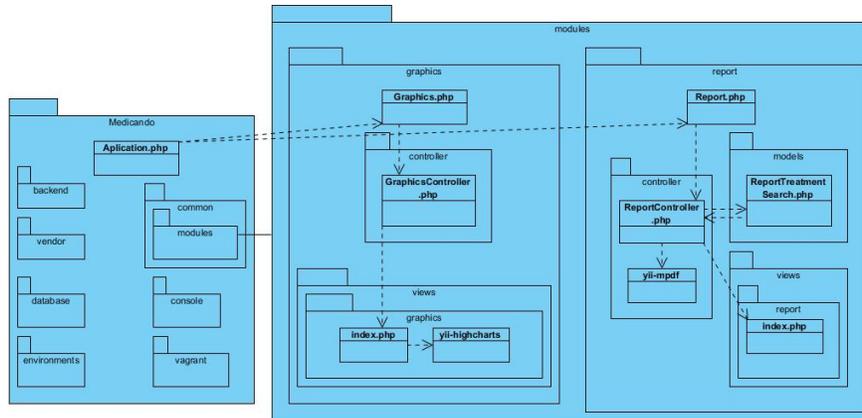


Figura 2.4. Representación de la arquitectura del sistema.

utilizada por la vista *index* de las gráficas, mientras que el controlador *ReportController* utiliza la extensión *yii-mpdf* para generar los documentos en formato PDF atendiendo a los contenidos especificados para cada reporte.

A lo largo del proceso de diseño y desarrollo, los atributos de calidad juegan un papel importante, pues en base a estos se generan las decisiones de diseño y argumentos que los justifican. Dado que la arquitectura de *software* inhibe o facilita los atributos de calidad, resulta de particular interés analizar la influencia de ciertos elementos de diseño utilizados para la definición de la misma, determinando sus características. Estos elementos de diseño son los patrones de diseño.

2.5.2. Patrones de diseño

Los patrones de diseño tratan los problemas que se repiten y que se presentan en situaciones particulares del diseño, con el fin de proponer soluciones a ellas. Por lo tanto, los patrones de diseño son soluciones exitosas a problemas comunes en el desarrollo de *software* y otros ámbitos referentes al diseño de interacción o interfaces. Los patrones dan nombre y forma a heurísticas abstractas, reglas y buenas prácticas de técnicas orientadas a objetos (LARMAN, 2002).

Los patrones [General Responsibility Assignment Software Patterns \(GRASP\)](#) del español Patrones Generales de *Software* para Asignar Responsabilidades, representan los principios básicos de la asignación de responsabilidades a objetos, expresados en forma de patrones de diseño (Larman, 2003). A continuación se describen aquellos patrones utilizados de acuerdo a las soluciones que brindan dentro del contexto del sistema:

- **Experto:** asigna una responsabilidad al experto, es la clase que cuenta con la información necesaria para cumplir la responsabilidad. El comportamiento se distribuye entre las clases que cuentan con la información requerida, alentando con ello definiciones de clases sencillas y más cohesivas que son más fáciles de comprender y mantener. Un ejemplo de esto se ve en la clase *Graphics* que es

la encargada de proporcionar toda la información relacionada con las gráficas que se generan en el sistema, debido a que representa todos los atributos para la creación de este objeto.

- **Creador:** define la clase que debe tener la responsabilidad de crear una nueva instancia de otra clase. El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Se evidencia en la clase controladora *GraphicsController*. Esta clase se encarga de modificar la configuración de las gráficas, 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:** se encarga de que una clase actúe como intermediaria para el manejo de eventos. Este patrón se evidencia en las clases controladoras definidas. En el caso de la clase *GraphicsController*, se encarga de manejar todos los eventos asociados a la generación de gráficas.
- **Alta Cohesión:** asigna una responsabilidad de forma tal que la cohesión siga siendo alta. En este caso se puede apreciar en las clases *GraphicsController* y *ReportController* encargadas de modificar la configuración de las gráficas y los reportes.
- **Bajo Acoplamiento:** Este patrón es el encargado de asignar una responsabilidad para conservar bajo acoplamiento. Este patrón se evidencia en las clases controladoras *GraphicsController* y *ReportController*, las cuales propician menos dependencias funcionales y ofrece mejores oportunidades de reutilización.

También fueron utilizados patrones [Pattern Gang of Four \(GOF\)](#), en español, Pandilla de los Cuatro, como son:

- **Singleton (Creación Única) :** garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella (ejemplo, a la hora de hacer una consulta a la base de datos a través del ORM *CActiveRecord* que tiene el framework Yii se hace una sola instancia a esa conexión).
- **Decorator (Estructural) :** permite añadir funcionalidades dinámicamente a una clase con la creación de otra clase, sin necesidad de la herencia. En el framework Yii, los *layouts* decoran a las vistas, pues en un momento determinado les confieren propiedades dinámicas que no poseían. En esta aplicación se implementan dos *layouts* (*login_layout* y *main_layout*, el primero enriquece la vista de la página de autenticación y el segundo al resto de las páginas del sistema).

Para el diseño de la aplicación, la metodología [XP](#) no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar se usan otras técnicas como las tarjetas [Contenido, Responsabilidad y Colaboración \(CRC\)](#). El uso de estos diagramas puede aplicarse siempre y cuando intervengan en el mejoramiento de la comunicación, tratando que no sean un peso en su mantenimiento, no sean extensos y que enfoquen la información de mayor importancia. Una vez definidos los patrones de diseño, se describen las clases utilizadas en la solución, se analizan las [HU](#) y se descomponen en tareas independientes.

2.5.3. Tarjetas CRC

Las tarjetas **CRC** permitirán desprenderse del método de trabajo basado en procedimientos y trabajar con una metodología asentada en objetos, permiten también que el equipo completo contribuya en la tarea del diseño. Estas tarjetas identifican y organizan las clases bajo el paradigma orientado a objetos (lo que incluye asignación de responsabilidades), cada tarjeta contiene el nombre de la clase (que representa una o más **HU**), una descripción de las responsabilidades o métodos asociados con la clase, así como la lista de las clases con que se relaciona o que colaboran con ella. Las tarjetas **CRC** son el único trabajo de diseño que se genera como parte del proceso de **XP** (R. y Q., 2014).

Las principales características de las tarjetas CRC son:

- Identificación de clases y asociaciones que participan del diseño del sistema.
- Obtención de las responsabilidades que debe cumplir cada clase.
- Establecimiento de cómo una clase colabora con otras clases para cumplir con sus responsabilidades.

Las tarjetas **CRC** permiten que el equipo completo contribuya en la tarea del diseño. El nombre de la clase se encarga de identificar la tarjeta, las responsabilidades se colocan a la izquierda, y las clases que se implican en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente, tal y como muestra la figura 2.5.

Nombre de la clase	
Responsabilidades	Colaboradores

Figura 2.5. Prototipo de tarjeta **CRC**

El campo **nombre de la clase** se refiere a las clases persistentes del sistema a desarrollar, **responsabilidad** a la función que realiza la clase dentro del sistema y **colaboradores** a la relación que tiene la clase con otras clases, a continuación se muestran las tarjetas confeccionadas.

Tabla 2.10. Tarjeta CRC # 1

Tarjeta CRC	
Clase: <i>GraphicsController</i>	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • <i>LoadForm()</i>: encargada de cargar las configuraciones para el formulario. • <i>RenderCharts()</i>: encargada de renderizar el gráfico generado con los datos enviados. 	<ul style="list-style-type: none"> • <i>TreatmentPatient</i> • <i>TreatmentVideogame</i> • <i>Treatment</i> • <i>Patient</i> • <i>Videogame</i>

Tabla 2.11. Tarjeta CRC # 2

Tarjeta CRC	
Clase: <i>ReportTreatmentSearch</i>	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Representa la clase Modelo encargada de realizar la búsqueda en la base de datos de los tratamientos. • <i>search(\$ params)</i>: retorna el identificador guardado para acceder al dato de la fila de la columna especificada. 	<ul style="list-style-type: none"> • <i>Treatment</i>

Tabla 2.12. Tarjeta CRC # 3

Tarjeta CRC	
Clase: <i>ReportController</i>	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • <i>Treatments()</i>: encargada de renderizar la vista donde se muestra el reporte generado con los datos de los tratamientos. • <i>ExportTreatments()</i>: encargada de reorganizar la información del reporte de tratamientos para mostrarlo en PDF. • <i>ClinicHistory(\$ patient_id = null)</i>: encargada de renderizar la vista donde se muestra el reporte generado con los datos del paciente seleccionado. • <i>LoadClinicHistory()</i>: encargada de cargar el formulario para seleccionar el paciente que se desee consultar su historia clínica. • <i>ExportHistory()</i>: encargada de reorganizar la información del reporte de la historia clínica del paciente seleccionado para mostrarlo en PDF. • <i>MyPatients()</i>: encargada de renderizar la vista donde se muestra el reporte generado con los datos de los pacientes y sus tratamientos. • <i>ExportMyPatients()</i>: encargada de reorganizar la información del reporte de los pacientes del doctor para llevarlo al PDF. • <i>Videogames()</i>: encargada de renderizar la vista donde se muestra el reporte generado con los datos de los videojuegos. • <i>ExportVideogames()</i>: encargada de reorganizar la información del reporte de videojuegos para cargarlo en PDF. 	<ul style="list-style-type: none"> • <i>TreatmentSearch</i> • <i>VideogameSearch</i> • <i>Doctor</i> • <i>Patient</i>

2.6. Conclusiones del capítulo

Una vez realizadas las etapas de planificación y diseño de los módulos de Graficación y Reportes, se concluye lo siguiente:

- La propuesta de solución planteada en la que se definieron las principales funcionalidades a desarrollar, dieron cumplimiento a lo requerido por el cliente a través de las HU.
- La definición de la estructura para la configuración a la hora de registrar un videojuego (Json Config) y de la salva de este, permitió una representación dinámica de las variables que almacenan los mismos.
- La generación de artefactos correspondientes a la etapa de diseño que propone la metodología, así como la generación de tarjetas CRC, permitieron trabajar con una metodología asentada en objetos.

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.

3.1. Introducción al capítulo

El presente capítulo comprende las etapas de implementación y pruebas de los módulos correspondientes. Se detalla el proceso de implementación que se realiza de forma iterativa e incremental, donde se exponen las tareas generadas por cada HU, para obtener al final de cada iteración, un producto que debe ser probado y mostrado al cliente. Se evidencian las diferentes pruebas realizadas: pruebas unitarias, de integración y aceptación permitiendo de esta forma, lograr una constante retroalimentación y un producto final que satisfaga al cliente.

3.2. Fase III: Implementación

En esta fase, las historias de usuarios seleccionadas para ser implementadas en cada iteración, se van realizando durante el transcurso de la iteración a la cual pertenecen; por lo que 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 historias de usuario en tareas de ingeniería, asignando a un grupo de desarrollo o a una persona responsable de la implementación.

3.2.1. Tareas de ingeniería

Las tareas de ingeniería 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. Pueden estar descritas por un lenguaje técnico y no ser necesariamente entendibles por el cliente Bustamante y Rodríguez, 2014. A continuación se describen las tareas de ingeniería correspondientes a las HU del sistema. Para una mayor organización, se definen en correspondencia con las iteraciones definidas como se manifiesta a continuación.

Iteración I

Esta iteración tendrá como objetivo darle cumplimiento a las **HU** correspondientes, separadas en varias tareas de ingeniería, donde se expondrán las más importantes.

- **HU Generar gráficas de los videojuegos**

Tabla 3.1. Tarea de ingeniería # 1

Tarea	
Número de tarea: 1	Número de Historia de usuario: 1
Nombre de la tarea: Cargar datos de cadena JSON del fichero de configuración del videojuego	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 6 de noviembre de 2017	Fecha de fin: 23 de noviembre de 2017
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
Descripción: Se implementa una funcionalidad dentro del modelo de videojuegos para crear la tabla en la base de datos con las variables del juego (json_config) dentro del formulario de creación.	

Tabla 3.2. Tarea de ingeniería # 2

Tarea	
Número de tarea: 2	Número de Historia de usuario: 1
Nombre de la tarea: Seleccionar la información a graficar	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 24 de noviembre de 2017	Fecha de fin: 4 de diciembre de 2017
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
Descripción: Se implementa una vista con el formulario que permitirá seleccionar el paciente, tratamiento, videojuego y el campo o variable a representar visualmente. En correspondencia con la variable seleccionada, se ofrece la forma de representación posible. Las vías disponibles (gráfica de barras, líneas, pastel y dispersión).	

Tabla 3.3. Tarea de ingeniería # 3

Tarea	
Número de tarea: 3	Número de Historia de usuario: 1
Nombre de la tarea: Generar gráfica a partir de los datos seleccionados	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 5 de diciembre de 2017	Fecha de fin: 24 de enero de 2018
Programador responsable: Mirdolys Nieves Llopiz Santovenia	

Continúa en la próxima página

Tabla 3.3. Continuación de la página anterior

Descripción: Se desarrolla una vista, para generar una gráfica con los datos dados y la forma de visualización elegida.
--

Iteración II

En la siguiente iteración se describen algunas de las tareas realizadas a las HU correspondientes al módulo de Reportes.

- **HU Generar reporte de Tratamientos**

Tabla 3.4. Tarea de ingeniería # 4

Tarea	
Número de tarea: 4	Número de Historia de usuario: 2
Nombre de la tarea: Mostrar la información sobre todos los tratamientos registrados en el sistema	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 29 de enero de 2018	Fecha de fin: 9 de febrero de 2018
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
Descripción: Se implementan varias funcionalidades, que en conjunto convergen en una función principal, capaz de organizar en una tabla, la información sobre los tratamientos existentes en el sistema, para el reporte.	

Tabla 3.5. Tarea de ingeniería # 5

Tarea	
Número de tarea: 5	Número de Historia de usuario: 2
Nombre de la tarea: Calcular el progreso del paciente en el tratamiento	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 9 de febrero de 2018	Fecha de fin: 14 de febrero de 2018
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
Descripción: Se desarrolla una función para calcular el porcentaje de cumplimiento del tratamiento de los pacientes asociados y los que ya hayan finalizado, siendo esta información visualizada mediante un gráfico de pastel.	

- **HU Generar reportes de Historia Clínica**

Tabla 3.6. Tarea de ingeniería # 6

Tarea	
Número de tarea: 6	Número de Historia de usuario: 3

Continúa en la próxima página

Tabla 3.6. Continuación de la página anterior

Nombre de la tarea: Seleccionar el paciente asociado al doctor	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 15 de febrero de 2018	Fecha de fin: 19 de febrero de 2018
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
Descripción: Se implementa una vista con el formulario que permitirá seleccionar el paciente deseado del doctor en cuestión.	

Tabla 3.7. Tarea de ingeniería # 7

Tarea	
Número de tarea: 7	Número de Historia de usuario: 3
Nombre de la tarea: Mostrar la información general del paciente seleccionado	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 20 de febrero de 2018	Fecha de fin: 27 de febrero de 2018
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
Descripción: Se implementan varias funcionalidades, que en conjunto convergen en una función principal, capaz de organizar la información general del paciente seleccionado.	

Tabla 3.8. Tarea de ingeniería # 8

Tarea	
Número de tarea: 8	Número de Historia de usuario: 3
Nombre de la tarea: Mostrar la información de los tratamientos asignados al paciente	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 28 de febrero de 2018	Fecha de fin: 2 de marzo de 2018
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
Descripción: Se implementan un conjunto de funcionalidades para organizar y mostrar la información de los tratamientos asignados al paciente.	

Tabla 3.9. Tarea de ingeniería # 9

Tarea	
Número de tarea: 9	Número de Historia de usuario: 3
Nombre de la tarea: Calcular el progreso del paciente en el tratamiento y en cada uno de los videojuegos.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 3 de marzo de 2018	Fecha de fin: 8 de marzo de 2018
Programador responsable: Mirdolys Nieves Llopiz Santovenia	

Continúa en la próxima página

Tabla 3.9. Continuación de la página anterior

Descripción: Se desarrollan funciones para calcular el progreso del tratamiento de los pacientes asociados y los que ya hayan finalizado, así como de los diferentes videojuegos asociados.

Iteración III

En esta última iteración se dará cumplimiento a las HU correspondientes, separadas en varias tareas de ingeniería.

- **HU Generar reportes de Pacientes**

Tabla 3.10. Tarea de ingeniería # 10

Tarea	
Número de tarea: 10	Número de Historia de usuario: 4
Nombre de la tarea: Mostrar la información de los paciente asociados a el doctor identificado	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha de inicio: 9 de marzo de 2018	Fecha de fin: 26 de marzo de 2018
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
Descripción: Se implementan varias funcionalidades, que en conjunto convergen en una función principal, capaz de organizar en un arreglo, la información sobre los pacientes asociados, atendiendo al doctor que realiza la petición del reporte.	

Tabla 3.11. Tarea de ingeniería # 11

Tarea	
Número de tarea: 11	Número de Historia de usuario: 4
Nombre de la tarea: Seleccionar doctor para el reporte de sus pacientes	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha de inicio: 27 de marzo de 2018	Fecha de fin: 30 de marzo de 2018
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
Descripción: Se implementa una vista con el formulario que permitirá seleccionar el doctor deseado para realizar el reporte de sus pacientes. Esto se evidencia para el usuario con el rol de administrador.	

- **HU Generar reportes de videojuegos**

Tabla 3.12. Tarea de ingeniería # 12

Tarea	
Número de tarea: 12	Número de Historia de usuario: 5
Nombre de la tarea: Mostrar la información de los videojuegos que existen en la plataforma	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 2 de abril de 2018	Fecha de fin: 13 de abril de 2018
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
Descripción: Se implementan varias funcionalidades para organizar la información de los videojuegos existentes en el sistema.	

Tabla 3.13. Tarea de ingeniería # 13

Tarea	
Número de tarea: 13	Número de Historia de usuario: 5
Nombre de la tarea: Calcular la eficiencia de cada uno de los videojuegos	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 26 de abril de 2018	Fecha de fin: 30 de abril de 2018
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
Descripción: Se implementa una función para calcular la eficiencia del videojuego, dado por la cantidad de pacientes que hayan terminado una sesión en los diferentes tratamientos que este el videojuego.	

- **HU Generar Documento PDF**

Tabla 3.14. Tarea de ingeniería # 14

Tarea	
Número de tarea: 14	Número de Historia de usuario: 6
Nombre de la tarea: Exportar información del reporte de tratamiento a documento PDF	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 1 de mayo de 2018	Fecha de fin: 5 de mayo de 2018
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
Descripción: Se implementa una funcionalidad para visualizar en formato PDF los datos adquiridos del reporte de tratamientos.	

Tabla 3.15. Tarea de ingeniería # 15

Tarea	
Número de tarea: 15	Número de Historia de usuario: 6

Continúa en la próxima página

Tabla 3.15. Continuación de la página anterior

Nombre de la tarea: Exportar información del reporte de la historia clínica del paciente a documento PDF	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 7 de mayo de 2018	Fecha de fin: 12 de mayo de 2018
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
Descripción: Se implementa una funcionalidad para visualizar en formato PDF los datos adquiridos del paciente en su historia clínica.	

Tabla 3.16. Tarea de ingeniería # 16

Tarea	
Número de tarea: 16	Número de Historia de usuario: 6
Nombre de la tarea: Exportar información del reporte de los pacientes del doctor autenticado a documento PDF	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 14 de mayo de 2018	Fecha de fin: 19 de mayo de 2018
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
Descripción: Se implementa una funcionalidad para visualizar en formato PDF los datos de los pacientes del doctor autenticado.	

Tabla 3.17. Tarea de ingeniería # 17

Tarea	
Número de tarea: 17	Número de Historia de usuario: 6
Nombre de la tarea: Exportar información del reporte de videojuegos a documento PDF	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 21 de mayo de 2018	Fecha de fin: 24 de mayo de 2018
Programador responsable: Mirdolys Nieves Llopiz Santovenia	
Descripción: Se implementa una funcionalidad para visualizar en formato PDF los datos de los videojuegos en el reporte.	

Una vez definidas las tareas de ingeniería, se hace necesario comprobar la calidad de la solución implementada a través de pruebas. Estos casos de prueba se analizan y se ejecutan, lo que permite medir si los requisitos implementados satisfacen las necesidades del cliente o no.

3.3. Fase IV: Pruebas

El proceso de prueba es una de las fases fundamentales de **XP**. Las pruebas son un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo sistemáticamente. Por esta razón se debe

definir en el proceso de la ingeniería del *software*. Todo esto contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones. La metodología *XP* divide las pruebas en dos grupos (J. Gutiérrez et al., 2006):

- **Pruebas unitarias:** desarrolladas por los programadores, encargadas de verificar el código de forma automática.
- **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.

Con el objetivo de comprobar que los sistemas desarrollados funcionan de acuerdo a las especificaciones descritas por el cliente, se realizaron diferentes pruebas teniendo en cuenta las características de los módulos. Las pruebas unitarias se realizaron para buscar errores en el funcionamiento interno de los módulos, errores en el código. Las pruebas de aceptación, para comprobar que los requisitos implementados satisfacen las necesidades del cliente. Pero además se realizaron pruebas de integración para comprobar que el sistema y los módulos funcionan correctamente como un todo.

3.3.1. Pruebas unitarias

Las pruebas unitarias son una forma de probar el buen funcionamiento de un módulo o una parte del sistema, con el fin de asegurar el correcto funcionamiento de todos los módulos por separados. Las pruebas unitarias aseguran que un único componente de la aplicación produce una salida correcta para una determinada entrada. Este tipo de pruebas comprueban la forma en que las funciones y los métodos trabajan en cada caso particular (Sommerville, 2010).

En el caso de los módulos, se realizaron este tipo de pruebas en conjunto con las de aceptación, debido a que en la realización de las mismas, los casos de pruebas son diseñados para cada una de las tareas de ingeniería definidas. Estas tareas constituyen la unidad atómica de las *HU*, es decir corresponden a la unidad más pequeña de las *HU*. Por tanto la realización de todos los casos de pruebas definidos para cada tarea, constituye la realización de la prueba a esa unidad, verificando así, el correcto funcionamiento.

En el Anexo A.2 se puede ver un ejemplo de la aplicación de estas pruebas, donde se decide utilizar el método de caja blanca (ver Anexo A.4), para garantizar que se ejerciten por lo menos una vez todos los caminos independientes de los módulos implementados. Para realizar la prueba fue necesario realizar primeramente el análisis de la complejidad del algoritmo sobre el que se va a desarrollar la prueba, con el fin de calcular los valores de la complejidad ciclomática y poder obtener el número de pruebas que se deben realizar.

3.3.2. Pruebas de integración

Las pruebas de integración son una técnica sistemática para construir la arquitectura del *software* mientras se llevan a cabo pruebas para descubrir errores asociados con la interfaz. El objetivo es tomar los

componentes probados de manera individual y construir una estructura de programa que se haya dictado por diseño (Pressman, 2010).

Durante la realización de las pruebas de integración en los módulos de Graficación y Reportes, al incorporarlos en la plataforma Medicando no se detectaron no conformidades. En total se realizaron 3 iteraciones de prueba, donde en la segunda y tercera no se evidenciaron ninguna no conformidad. En la imagen 3.1 se muestra la prueba realizada aplicando como estrategia la integración descendente.

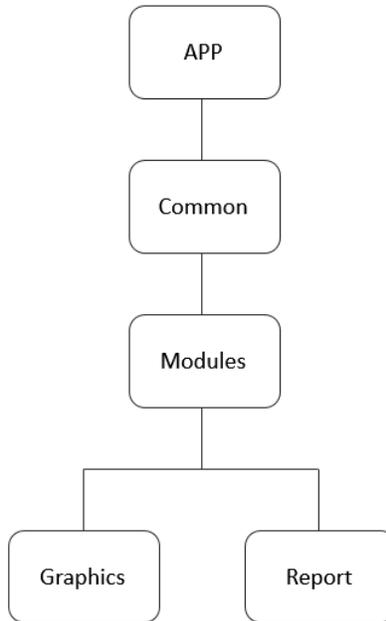


Figura 3.1. Prueba de integración

La prueba de integración descendente es un enfoque incremental a la construcción de la arquitectura de *software*. Los módulos se integran al moverse hacia abajo a través de la jerarquía de control, comenzando con el módulo de control principal. Los módulos subordinados al módulo de control principal se incorporan en la estructura en una forma de primero en anchura.

La integración primero en anchura incorpora todos los componentes directamente subordinados en cada nivel, y se mueve horizontalmente a través de la estructura. En este caso *Graphics* y *Report* se integran a *Modules*, luego este se integra a *Common* que finalmente se vincula con *APP*.

3.3.3. Pruebas de aceptación

Las pruebas de aceptación se realizan sobre el producto terminado. Están concebidas para que sea un usuario final quien detecte los posibles errores. En esta prueba se evalúa el grado de calidad del *software*

con relación a todos los aspectos relevantes para que el uso del producto se justifique. Este tipo de pruebas son comúnmente realizadas por el usuario final, quien debe informar de todas las deficiencias o errores que encuentre antes de dar por aprobado el sistema definitivamente (J. Gutiérrez et al., 2006).

Iteración I

En la primera iteración se realizaron pruebas de aceptación, enfocadas en evaluar la implementación del módulo de Graficación, para comprobar que luego de ejecutada cada una de las funcionalidades asociadas a los gráficos del módulo, se muestre correctamente el gráfico correspondiente. A continuación, se muestran algunas de ellas, el resto puede ser consultado en el Anexo A.3.

Tabla 3.18. Prueba de aceptación # 1

Caso de prueba de aceptación	
Código: HU1_P1	Historia de usuario: 1
Nombre: Generar gráfica de videojuegos (barras)	
Descripción: Prueba de la funcionalidad de generación de gráficas de los videojuegos (Barras)	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario deberá estar autenticado y tener el rol de Doctor. • Deberá tener un paciente el cual siga un tratamiento. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El Doctor desde la vista de graficación seleccionará un tratamiento. 2. Se selecciona el paciente donde solo aparecerán los asociados al tratamiento seleccionado. 3. Se selecciona el videojuego (Ejercita tu mente). 4. Se selecciona la variable que desea graficar, donde solo aparecerán las relacionadas con el videojuego (puntuacionAcierto). 5. Se selecciona el tipo de gráfico que estará en correspondencia a la variable seleccionada (Gráfico de Barras). 6. Como respuesta del sistema se genera una gráfica (de barras) con toda la información del paciente y el videojuego. 	
Resultados esperados: Satisfactorio	

Tabla 3.19. Prueba de aceptación # 2

Caso de prueba de aceptación	
Código: HU1_P2	Historia de usuario: 1
Nombre: Generar gráfica de videojuegos (líneas)	
Descripción: Prueba de la funcionalidad de generación de gráficas de los videojuegos (Líneas)	

Continúa en la próxima página

Tabla 3.19. Continuación de la página anterior

<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> • El usuario deberá estar autenticado y tener el rol de Doctor. • Deberá tener un paciente el cual siga un tratamiento.
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El Doctor desde la vista de graficación seleccionará un tratamiento. 2. Se selecciona el paciente donde solo aparecerán los asociados al tratamiento seleccionado. 3. Se selecciona el videojuego (Ejercita tu mente). 4. Se selecciona la variable que desea graficar, donde solo aparecerán las relacionadas con el videojuego (intentosFallidosAcierto). 5. Se selecciona el tipo de gráfico que estará en correspondencia a la variable seleccionada (Gráfico de Líneas). 6. Como respuesta del sistema se genera una gráfica (de líneas) con toda la información del paciente y el videojuego.
<p>Resultados esperados: Satisfactorio</p>

Tabla 3.20. Prueba de aceptación # 3

Caso de prueba de aceptación	
Código: HU1_P3	Historia de usuario: 1
Nombre: Generar gráfica de videojuegos (pastel)	
Descripción: Prueba de la funcionalidad de generación de gráficas de los videojuegos (Pastel)	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> • El usuario deberá estar autenticado y tener el rol de Doctor. • Deberá tener un paciente el cual siga un tratamiento. 	
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El Doctor desde la vista de graficación seleccionará un tratamiento. 2. Se selecciona el paciente donde solo aparecerán los asociados al tratamiento seleccionado. 3. Se selecciona el videojuego (Ejercita tu mente). 4. Se selecciona la variable que desea graficar, donde solo aparecerán las relacionadas con el videojuego (intentosFallidosAcierto). 5. Se selecciona el tipo de gráfico que estará en correspondencia a la variable seleccionada (Gráfico de Pastel). 6. Como respuesta del sistema se genera una gráfica (de pastel) con toda la información del paciente y el videojuego. 	
<p>Resultados esperados: Satisfactorio</p>	

Tabla 3.21. Prueba de aceptación # 4

Caso de prueba de aceptación	
Código: HU1_P5	Historia de usuario: 1
Nombre: Generar gráfica de videojuegos (dispersión)	
Descripción: Prueba de la funcionalidad de generación de gráficas de los videojuegos (Dispersión)	
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario deberá estar autenticado y tener el rol de Doctor. • Deberá tener un paciente el cual siga un tratamiento. 	
Pasos de ejecución: <ol style="list-style-type: none"> 1. El Doctor desde la vista de graficación seleccionará un tratamiento. 2. Se selecciona el paciente donde solo aparecerán los asociados al tratamiento seleccionado. 3. Se selecciona el videojuego (Ejercita tu mente). 4. Se selecciona la variable que desea graficar, donde solo aparecerán las relacionadas con el videojuego (intentosFallidosAcierto). 5. Se selecciona el tipo de gráfico que estará en correspondencia a la variable seleccionada (Gráfico de Dispersión). 6. Como respuesta del sistema se genera una gráfica (de dispersión) con toda la información del paciente y el videojuego. 	
Resultados esperados: Satisfactorio	

Iteración II

En la próxima iteración se realizaron pruebas enfocadas a evaluar la implementación del módulo de Reportes, para comprobar que luego de ejecutada cada una de las funcionalidades asociadas a los reportes de los tratamientos y videojuegos, se muestre correctamente la información.

Tabla 3.22. Prueba de aceptación # 5

Caso de prueba de aceptación	
Código: HU2_P1	Historia de usuario: 2
Nombre: Generar reporte de Tratamientos	
Descripción: Prueba de la funcionalidad de generación de reporte de Tratamientos	
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario deberá estar autenticado y tener el rol de Doctor. • Deberá tener un paciente el cual siga un tratamiento. 	

Continúa en la próxima página

Tabla 3.22. Continuación de la página anterior

<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El Doctor desde la vista principal selecciona el tipo de reporte (Tratamientos) o desde la vista de tratamientos selecciona 'Realizar Reporte'. 2. Como respuesta del sistema, se genera una tabla con la información del reporte. 3. Se muestra además la opción de exportar el reporte como documento PDF.
<p>Resultados esperados: Satisfactorio</p>

Tabla 3.23. Prueba de aceptación # 6

Caso de prueba de aceptación	
Código: HU3_P2	Historia de usuario: 3
Nombre: Generar reporte de Historia Clínica	
Descripción: Prueba de la funcionalidad de generación de reportes de la historia clínica de un paciente	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> • El usuario deberá estar autenticado y tener el rol de Doctor. • Deberá tener un paciente el cual siga un tratamiento. 	
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El Doctor desde la vista principal selecciona el tipo de reporte 'Historia Clínica'. 2. Como respuesta del sistema, se direcciona hacia la vista donde se muestra un campo para seleccionar el paciente (solo los del doctor). 3. Se selecciona el paciente. 4. Se genera una tabla con la información del paciente. 5. Se muestra además la opción de exportar el reporte como documento PDF. 	
Resultados esperados: Satisfactorio	

Tabla 3.24. Prueba de aceptación # 7

Caso de prueba de aceptación	
Código: HU3_P2	Historia de usuario: 3
Nombre: Generar reporte de Historia Clínica	
Descripción: Prueba de la funcionalidad de generación de reportes de la historia clínica de un paciente	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> • El usuario deberá estar autenticado y tener el rol de Doctor. • Deberá tener un paciente el cual siga un tratamiento. 	

Continúa en la próxima página

Tabla 3.24. Continuación de la página anterior

<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El Doctor desde la vista de pacientes selecciona 'ver Historia clínica'. 2. Como respuesta del sistema, se direcciona hacia la vista donde se muestra la información general del paciente y los tratamientos asignados. 3. Se muestra además la opción de exportar el reporte como documento PDF.
<p>Resultados esperados: Satisfactorio</p>

Iteración III

En la última iteración se mantuvo el enfoque de evaluar la implementación del módulo de Reportes, finalizando así con las últimas funcionalidades del módulo, asociadas a los reportes de los pacientes, los videojuegos y la generación del PDF.

Tabla 3.25. Prueba de aceptación # 8

Caso de prueba de aceptación	
Código: HU4_P1	Historia de usuario: 4
Nombre: Generar reporte de Pacientes	
Descripción: Prueba de la funcionalidad de generación de reporte de los pacientes del doctor	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> • El usuario deberá estar autenticado y tener el rol de Doctor. • Deberá tener un paciente el cual siga un tratamiento. 	
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El Doctor desde la vista principal selecciona el tipo de reporte 'Mis Pacientes'. 2. Como respuesta del sistema, se genera una tabla con la información de los pacientes. 3. Se muestra además la opción de exportar el reporte como documento PDF. 	
Resultados esperados: Satisfactorio	

Tabla 3.26. Prueba de aceptación # 9

Caso de prueba de aceptación	
Código: HU4_P2	Historia de usuario: 4
Nombre: Generar reporte de Pacientes	
Descripción: Prueba de la funcionalidad de generación de reporte de los pacientes del doctor	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> • El usuario deberá estar autenticado y tener el rol de Administrador. 	

Continúa en la próxima página

Tabla 3.26. Continuación de la página anterior

<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El Administrador desde la vista principal selecciona el tipo de reporte 'Mis Pacientes'. 2. Como respuesta del sistema, se direcciona hacia la vista donde se muestra un campo para seleccionar el doctor. 3. Se selecciona el doctor. 4. Como respuesta del sistema, se genera una tabla con la información de los pacientes. 5. Se muestra además la opción de exportar el reporte como documento PDF.
<p>Resultados esperados: Satisfactorio</p>

Tabla 3.27. Prueba de aceptación # 10

Caso de prueba de aceptación	
Código: HU6_P1	Historia de usuario: 5
Nombre: Generar reporte de Videojuegos	
Descripción: Prueba de la funcionalidad de generación de reporte de Videojuegos	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> • El usuario deberá estar autenticado y tener el rol de Doctor. 	
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El Doctor desde la vista principal selecciona el tipo de reporte 'Videojuegos' o desde la vista de videojuegos selecciona 'Realizar Reporte'. 2. Como respuesta del sistema, se genera una tabla con la información del reporte. 3. Se muestra además la opción de exportar el reporte como documento PDF. 	
<p>Resultados esperados: Satisfactorio</p>	

Tabla 3.28. Prueba de aceptación # 11

Caso de prueba de aceptación	
Código: HU6_P1	Historia de usuario: 6
Nombre: Generar documento PDF (Tratamientos)	
Descripción: Prueba de la funcionalidad de exportar a PDF (Tratamientos)	
<p>Condiciones de ejecución:</p>	
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Se da clic desde la vista del reporte de tratamientos 'Exportar'. 2. Como respuesta del sistema, se genera un documento PDF con la información del reporte de tratamientos. 	
<p>Resultados esperados: Satisfactorio</p>	

Tabla 3.29. Prueba de aceptación # 12

Caso de prueba de aceptación	
Código: HU6_P2	Historia de usuario: 6
Nombre: Generar documento PDF (Historia Clínica)	
Descripción: Prueba de la funcionalidad de exportar a PDF (Historia Clínica)	
Condiciones de ejecución:	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Se da clic desde la vista del reporte de historia clínica 'Exportar'. 2. Como respuesta del sistema, se genera un documento PDF con la información del reporte de la historia clínica del paciente. 	
Resultados esperados: Satisfactorio	

Tabla 3.30. Prueba de aceptación # 13

Caso de prueba de aceptación	
Código: HU6_P3	Historia de usuario: 6
Nombre: Generar documento PDF (Mis Pacientes)	
Descripción: Prueba de la funcionalidad de exportar a PDF (Mis Pacientes)	
Condiciones de ejecución:	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Se da clic desde la vista del reporte de pacientes 'Exportar'. 2. Como respuesta del sistema, se genera un documento PDF con la información del reporte de pacientes del doctor autenticado. 	
Resultados esperados: Satisfactorio	

Tabla 3.31. Prueba de aceptación # 14

Caso de prueba de aceptación	
Código: HU6_P4	Historia de usuario: 6
Nombre: Generar documento PDF (Videojuegos)	
Descripción: Prueba de la funcionalidad de exportar a PDF (videojuegos)	
Condiciones de ejecución:	

Continúa en la próxima página

Tabla 3.31. Continuación de la página anterior

<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Se da clic desde la vista del reporte de videojuegos 'Exportar'. 2. Como respuesta del sistema, se genera un documento PDF con la información del reporte de videojuegos.
<p>Resultados esperados: Satisfactorio</p>

3.3.4. Resultados de las pruebas

Durante las pruebas de integración se obtuvieron resultados satisfactorios al integrar ambos módulos con la plataforma, donde se comprobó el buen funcionamiento de los mismos.

En el proceso de pruebas de aceptación a partir de los 16 casos de pruebas definidos por el cliente durante las 3 iteraciones de implementación planificadas, arrojó un total de 3 resultados no satisfactorios. En la primera y segunda iteración se detectaron resultados no satisfactorios, de carácter no significativos, los cuales fueron solucionados, mientras que en la tercera todas fueron positivas.

A las HU correspondientes a cada iteración se le realizaron varias pruebas unitarias, donde las dificultades fueron solucionadas en un corto plazo, antes de pasar a la iteración posterior, cumpliéndose correctamente todos los requisitos solicitados por el cliente. En la siguiente ilustración se muestran los resultados de las pruebas, donde por cada iteración se muestra la cantidad de pruebas que fueron satisfactorias o no.

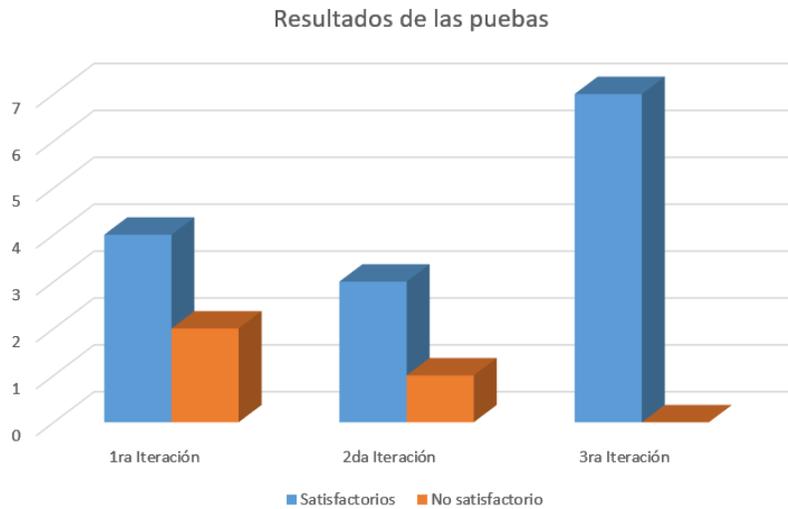


Figura 3.2. Resultados de las pruebas de aceptación

3.4. Conclusiones del capítulo

En el presente capítulo se mostraron los resultados de la implementación y pruebas realizadas a los módulos de Graficación y Reportes por lo que se concluye:

- Al definirse las tareas de ingeniería correspondientes a cada HU se obtuvo un nivel más detallado de la implementación de la aplicación.
- Se realizaron pruebas de integración que permitieron comprobar la correcta vinculación de los módulos con la plataforma, evidenciando el correcto funcionamiento del sistema tanto con estos módulos activos, como inactivos.
- Se realizaron pruebas unitarias en conjunto con las de aceptación que permitieron detectar errores durante las iteraciones. Luego de ser corregidas en su totalidad, validaron su correcto funcionamiento en relación a los requisitos definidos por el cliente.

Una vez finalizada la fundamentación teórica que sustentó la presente investigación, definidas las características de la propuesta de solución y efectuado su desarrollo y validación, se obtuvieron resultados que permiten arribar a las siguientes conclusiones:

- La definición de un atributo **json_config** de configuración y el estándar para salvar los datos de los videojuegos, permitió que estos datos fueran representados dinámicamente por los gráficos.
- El desarrollo de los módulos de Graficación y Reportes, ofrecen una vía para interpretar y analizar los datos generados por los videojuegos utilizados, en los tratamientos asociados a cada paciente en la plataforma Medicando.
- El uso de gráficas y reportes permite a los especialistas, perfeccionar el seguimiento de la evolución de los pacientes en sus tratamientos con videojuegos.

Recomendaciones

A partir de los resultados obtenidos con el desarrollo de los módulos propuestos, se recomienda:

- Incorporar nuevos reportes basados en las necesidades de los especialistas.

Apache Server Servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual. [10](#), [13](#)

BSD Licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution), un tipo del sistema operativo Unix-like. Es una licencia de software libre permisiva como la licencia de OpenSSL o la MIT License. [9](#)

JavaScript Lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. [8](#), [10](#), [15](#), [21](#), [29](#)

PHP Lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. [10](#), [14](#), [15](#), [29](#)

Telemedicina Distribución de servicios de salud, en el que la distancia es un factor crítico, donde los profesionales de la salud usan información y tecnología de comunicaciones para el intercambio de información válida para el diagnóstico, tratamiento y prevención de enfermedades o daños, investigación y evaluación; y para la educación continuada de los proveedores de salud pública, todo ello en interés del desarrollo de la salud del individuo y su comunidad. [1](#)

XAVIA Marca registrada por la Universidad de las Ciencias Informáticas para las soluciones informáticas en el sector de Salud, donde son implementadas en su mayoría sobre plataformas de software libre o código abierto. [29](#)

- BI** Business Intelligence. [12](#)
- CITEC** Ciencias y Tecnologías Computacionales. [13](#)
- CRC** Contenido, Responsabilidad y Colaboración. [33](#), [34](#), [37](#)
- CSS3** Cascade Style Sheet. [10](#), [29](#)
- DATEC** Centro de Tecnologías de Gestión de Datos. [13](#)
- GOF** Pattern Gang of Four. [33](#)
- GRASP** General Responsibility Assignment Software Patterns. [32](#)
- HTML5** Hypertext Markup Language. [10](#), [12](#)
- HU** Historias de Usuario. [19](#), [23](#), [29](#), [30](#), [33](#), [34](#), [37–40](#), [42](#), [45](#), [54](#), [55](#)
- JSON** JavaScript Object Notation. [21](#), [39](#)
- MVC** Modelo-Vista-Controlador. [14](#), [31](#)
- OLAP** On-Line Analytical Processing. [12](#)
- REST** Transferencia de Estado Representacional. [14](#)
- SGBD** Sistema de Gestión de Bases de Datos. [10](#), [13](#), [18](#), [29](#)
- TIC** Tecnologías de la Información y las Comunicaciones. [1](#)
- UCI** Universidad de las Ciencias Informáticas. [1](#), [12](#), [16](#), [29](#)
- Vertex** Centro de Entornos Interactivos 3D. [1](#), [10](#)
- XML** Extensible Markup Language. [12](#), [21](#)
- XP** Extreme Programming. [17–19](#), [23](#), [33](#), [34](#), [44](#), [45](#)

Referencias bibliográficas

- ABAD, P y E HUAPAYA (2009). «Guía para la presentación de Gráficos estadísticos». En: *Instituto Nacional de Estadística e informática, Lima* (vid. pág. 5).
- Acosta Ortíz, Luis Ernesto y Daylenis Sánchez Delgado (2012). «Sistema informático para la gestión de los planes de ingreso a la Educación Media y Superior en Cuba». En: *Revista Cubana de Ciencias Informáticas* 6.4 (vid. pág. 13).
- Agudo, Alejandra (2013). «Para su enfermedad, tome este videojuego». En: *El Pais* (vid. pág. 1).
- Aragon, Rafa (2011). «Videojuegos terapeuticos: Serious Games». En: *Psique Viva* (vid. pág. 4).
- Bernardino, Jorge y Marco Tereso (2013). «Business intelligence tools». En: *Computational Intelligence and Decision Making*. Springer, págs. 267-276 (vid. pág. 12).
- Bustamante, Dayana y Jean C. Rodríguez (2014). «Metodología Actual. Metodología XP». En: *UNIVERSIDAD NACIONAL EXPERIMENTAL DE LOS LLANOS OCCIDENTALES EZEQUIEL ZAMORA* (vid. págs. 23, 38).
- Cadavid, Andrés Navarro, Juand Daniel Fernández Martínez y Jonathan Moralez Vélez (2013). «Revisión de metodologías ágiles para el desarrollo de software». En: *Prospectiva* 11.2, págs. 30-39 (vid. pág. 16).
- Clemente Menéndez, Fco. Javier y Enrique Rodríguez Cotorruelo (2013). «Metodologías Ágiles .Extreme Programming"(XP)». En: *SlideShare* (vid. pág. 17).
- Definición de Reporte* (s.f.). URL: <http://definicion.de/reportes/#ixzz2ENIiAwvK> (vid. pág. 5).
- Espinosa Robert, Ana del Carmen, Yadira Barroso Rodríguez y Anyer Gámez Guedes (2015). «Sistema Auditoría de Datos: Módulo de Gráficos v2.0». Tesis doct. Universidad de las Ciencias Informáticas (vid. pág. 6).
- Estrella, Soledad (2014). «Tabular format: a review of literature». En: *Actualidades Investigativas en Educación* 14.2, págs. 449-478 (vid. pág. 5).
- Fossati, Matias (2014). *Todo sobre MySQL: Libro ideal para ingresar en el mundo de la base de datos MySQL*. Natsys (vid. pág. 15).
- García Díaz, Gabriel Lázaro et al., (2017). «Arquitectura para la plataforma de gestion de videojuegos serios Medicando». Tesis doct. Universidad de las Ciencias Informaticas (vid. págs. 10, 14, 28).
- Gea, M Magdalena et al., (2014). «La distribución de datos bidimensionales en los libros de textos de matemáticas de Bachillerato». En: *Cuadrante* 23.2, págs. 147-172 (vid. pág. 8).

- Golfarelli, Matteo (2009). «Open source BI platforms: A functional and architectural comparison». En: *International Conference on Data Warehousing and Knowledge Discovery*. Springer, págs. 287-297 (vid. pág. 12).
- Gómez Martínez, Greter y Anchel Durán Bolaño (2015). «Biblioteca de componentes de interfaz de usuario para aplicaciones de escritorio desarrolladas en Java.» Tesis doct. Universidad de las Ciencias Informáticas (vid. pág. 16).
- González E., Geidy y Yoandi Alfonso P. (2013). «Sistema de Reportes y apoyo a la toma de decisiones para la Dirección de Alimentación de la Universidad de las Ciencias Informáticas». Tesis doct. Universidad de las Ciencias Informáticas (vid. pág. 11).
- Gráfica de Pastel* (s.f.). Universidad Nacional Autónoma de México, Facultad de Estudios Superiores Cuautitlán. URL: <http://www.cuautitlan.unam.mx> (vid. pág. 7).
- Guerra Cardet, Maiyara y Frank Rufino Nápoles (2015). «Desarrollo del componente gráfico de barras para el HMI del SCADA SAINUX». Tesis doct. Universidad de las Ciencias Informáticas (vid. págs. 6, 7).
- Gutiérrez, Javier J (2014). «¿ Qué es un framework web». En: *Available in: http://www.lsi.us.es/~javier-j/investigacion_ficheros/Framework.pdf Accessed May 12* (vid. pág. 31).
- Gutiérrez, JJ et al., (2006). «Pruebas del Sistema en Programación Extrema». En: *Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla* (vid. págs. 45, 47).
- Guzmán, José Roberto et al., (2013). «Biblioteca JavaScript para el desarrollo de interfaces gráficas de usuario de RIA». Tesis doct. Universidad de las Ciencias Informáticas (vid. pág. 8).
- Hernández Hermosillo, Silvia Mireya y Angélica Sánchez Ángeles (2013). «Excel básico 2010-gráficos». En: *Sistema de Universidad Virtual* (vid. pág. 7).
- Hernández, Esther Naranjo y C Ing Inty Sáez Mosquera (2012). «Pentaho: software líder de Inteligencia de Negocio de código abierto.» En: *Revista Telem@tica* 10.2 (vid. pág. 12).
- Hernández, Yanoski Agneri Martínez y José Rolando Lafaurie. Olivares (2008). «Sistema para la generación de reportes en la plataforma alasGRATO». En: *Universidad de las Ciencias Informáticas* (vid. pág. 13).
- Kanashiro Medina, Antonio et al., (2013). «Uso de Microsoft Excel para análisis y elaboración de informes de resultados de evaluación de aprendizajes». En: *Universidad Peruana de Ciencias Aplicadas* (vid. pág. 7).
- La UCI de un vistazo* (2018). URL: www.uci.cu (vid. pág. 1).
- LARMAN, C (2002). «UML y Patrones. 2da Edición». En: 2002 (vid. pág. 32).
- Larman, Craig (2003). *UML y Patrones*. Ed. por David Fayerman Aragón. Pearson Educación ^ eMadrid Madrid (vid. pág. 32).
- López Raventós, Cristian (2016). «El videojuego como herramienta educativa. Posibilidades y problemáticas acerca de los serious games.» En: *Guadalajara Apertura Jal.* 8.1, págs. 0–0 (vid. pág. 4).
- Marder, Andrew (2016). «5 Awesome Free and Open Source Reporting Software Packages». En: *Capterra Finance Software Blog* (vid. pág. 11).

- Marín, Ambuludi y José Lanier (2016). «Implementación de una aplicación web para la gestión de procesos administrativos de la cooperativa de transporte panamá internacional». En: *Buscar* (vid. pág. 17).
- Molinero, Luis M. (2003). «Gráficos de datos estadísticos en medicina.» En: *Asociación de la Sociedad Española de Hipertensión*. (Vid. pág. 8).
- MySQL (s.f.). URL: <https://www.mysql.com/> (vid. pág. 15).
- NetBeans (s.f.). URL: <https://www.netbeans.org/> (vid. pág. 15).
- Nolan, Deborah y Duncan Temple Lang (2014). «Javascript object notation». En: *XML and Web Technologies for Data Sciences with R*. Springer, págs. 227-253 (vid. pág. 21).
- Olszak, Celina M (2014). «Business Intelligence in cloud». En: *Polish Journal of Management Studies* 10 (vid. pág. 12).
- Oracle (2013). GlassFish - World's first Java EE 7 Application Server. URL: <https://glassfish.java.net/> (vid. pág. 15).
- Patil, Manoj R y Feris Thia (2013). *Pentaho for big data analytics*. Packt Publishing Ltd (vid. pág. 12).
- Peh, Diana, Nola Hague y Jane Tatchell (2011). *BIRT: a field guide*. Addison-Wesley Professional (vid. pág. 12).
- Perez, Julian y Ana Gardey (2013). «Videojuego». En: *Definicion.de* (vid. pág. 4).
- Pressman, Roger S. Ph.D. (2010). *INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO Séptima edición*. Ed. por Pablo Roig Vázquez. MCGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V. (vid. pág. 46).
- R., Gomez A. y Duarte A. Q. (2014). «Desarrollo ágil de software aplicando programación extrema.» En: *Revista Ingenio* (vid. pág. 34).
- Ramos Gutiérrez, Ernesto y Omar Correa Madrigal (2015). «Plataforma web para la gestión de videojuegos serios de navegador Medicando». En: *RCCI* (vid. págs. 10, 23).
- Real Academia Española (s.f.). Real Academia Española. URL: www.rae.es (vid. pág. 5).
- Solano Carrasco, Esteban Fabián y Geovanny Edmundo Villalva Zhindón (2012). «Diseño de un sistema de graficación de datos». B.S. thesis. Universidad del Azuay (vid. pág. 5).
- Sommerville, Ian (2010). *Software engineering*. New York: Addison-Wesley (vid. pág. 45).
- Tomás-Sábado, Joaquín (2010). *Fundamentos de bioestadística y análisis de datos para enfermería*. Vol. 2. Univ. Autónoma de Barcelona (vid. págs. 1, 6).
- Torres Mirabales, Marco A., MSc. Omar Correa Madrigal e Ing. Reinaldo Garcia Maturell (2015). «Portal web del Centro de Entornos Interactivos 3D». Tesis doct. Universidad de las Ciencias Informáticas (vid. pág. 23).
- Valdés, Juan Gabriel, Reinaldo Garcia y Yadira Ramírez (2016). «Módulos de visualización de la información paciente-especialista para la plataforma de gestión de videojuegos Medicando». En: *UCIENCIA* (vid. pág. 6).
- Yii2 Framework (s.f.). URL: <https://yii2-framework.readthedocs.io/en/stable/> (vid. pág. 14).

Apéndices

A.1. Ejemplo de las estructuras de configuración y salva en JSON

```
{
  "table_config": [
    {
      "column_name": "score",
      "column_type": "int",
      "column_label": "Puntuación",
      "charts": ["bar", "line"]
    }, {
      "column_name": "time",
      "column_type": "date",
      "column_label": "Tiempo"
    }, {
      "column_name": "skills",
      "column_type": "boolean",
      "column_label": "Habilidad"
    }, {
      "column_name": "visual_field",
      "column_type": "json",
      "column_label": "Matriz",
      "charts": ["scatter"]
    }
  ]
}
```

Figura A.1. Ejemplo de estructura de atributo **json_config**

```

{
  "lista_jugadores":[
    {
      "session_id":1,
      "nombre_jugador":"Álvaro",
      "score":1000,
      "time":"10-4-2018",
      "skills":58,
      "visual_field":[[3,4],[6,7]]
    }, {
      "session_id":1,
      "nombre_jugador":"Nieves",
      "score":2050,
      "time":"10-5-2018",
      "skills":84,
      "visual_field":[[10,5],[7,9]]
    }, {
      "session_id":2,
      "nombre_jugador":"Nieves",
      "score":200,
      "time":"10-5-2018",
      "skills":90,
      "visual_field":[[10,5],[7,9]]
    }
  ]
}

```

Figura A.2. Ejemplo de estructura de atributo `json_salva`

A.2. Pruebas unitarias

La técnica empleada en las pruebas unitarias es camino básico, donde para realizarla se analizó la complejidad algorítmica, con el fin de calcular los valores de la complejidad ciclomática, la cual para el siguiente ejemplo quedo de la siguiente forma:

- **V(G)= # de regiones del grafo:** $V(G)=2$
- **V(G)= Aristas - Nodos + 2:** $V(G)= 5-5+2= 2$
- **V(G)= Nodos Predicados + 1:** $V(G)= 1+1= 2$

Teniendo la complejidad ciclomática se obtiene el número de caminos independientes que dan un valor límite para el número de pruebas que se deben realizar.

```

public function actionMyPatients()
{
    ① $doctor_id = Yii::$app->request->post('doctor_id', null);

    ② if (isset($doctor_id)) {
        $doctor = Doctor::findOne($doctor_id);
    } else {
    ③ $doctor = Doctor::findOne(['user_id' => Yii::$app->user->id]);
    }

    ④ return $this->render('my-patients/myPatients', [
        'doctor' => $doctor,
    ]);
}

```

Figura A.3. Ejemplo de algoritmo.

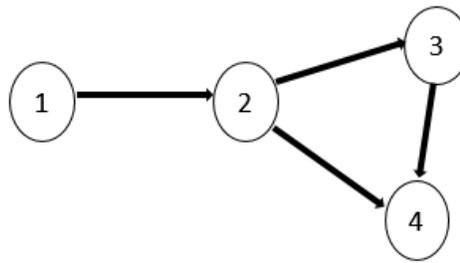


Figura A.4. Grafo de flujo asociado al algoritmo para renderizar los reportes de los pacientes de un doctor.

A.3. Pruebas de aceptación

Pruebas de aceptación para la Iteración I

Tabla A.1. Prueba de aceptación # 15

Caso de prueba de aceptación	
Código: HU1_P4	Historia de usuario: 1
Nombre: Generar gráfica de videojuegos (pastel)	
Descripción: Prueba de la funcionalidad de generación de gráficas de los videojuegos (Pastel)	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario deberá estar autenticado y tener el rol de Doctor. • Deberá tener un paciente el cual siga un tratamiento. 	

Continúa en la próxima página

Tabla A.1. Continuación de la página anterior

<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El Doctor desde la vista de graficación seleccionará un tratamiento. 2. Se selecciona el paciente donde solo aparecerán los asociados al tratamiento seleccionado. 3. Se selecciona el videojuego (Ejercita tu mente). 4. Se selecciona la variable que desea graficar, donde solo aparecerán las relacionadas con el videojuego (intentosFallidosAcierto). 5. Se selecciona el tipo de gráfico que estará en correspondencia a la variable seleccionada (Gráfico de Pastel). 6. Como respuesta del sistema no se muestra nada.
<p>Resultados esperados: No Satisfactorio</p>

Pruebas de aceptación para la Iteración II

Tabla A.2. Prueba de aceptación # 16

Caso de prueba de aceptación	
Código: HU3_P1	Historia de usuario: 3
Nombre: Generar reporte de Historia Clínica	
Descripción: Prueba de la funcionalidad de generación de reportes de la historia clínica de un paciente	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> • El usuario deberá estar autenticado y tener el rol de Doctor. • Deberá tener un paciente el cual siga un tratamiento. 	
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El Doctor desde la vista principal selecciona el tipo de reporte 'Historia Clínica'. 2. Como respuesta del sistema, se direcciona hacia la vista donde se muestra un campo para seleccionar el paciente (todos los paciente registrados del sistema). 	
Resultados esperados: No Satisfactorio	

A.4. Métodos de prueba

La prueba de caja blanca, en ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba. Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez, revisen todas las decisiones lógicas en sus lados verdadero y falso, ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas y revisen estructuras de datos internas para garantizar su validez.

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimien-

tos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable que descubra una clase de errores diferente que los métodos de caja blanca.

Las pruebas de caja negra intentan encontrar errores en funciones incorrectas o faltantes, errores de interfaz, errores en las estructuras de datos o en el acceso a bases de datos externas, errores de comportamiento o rendimiento y errores de inicialización y terminación.

A diferencia de las pruebas de caja blanca, que se realizan tempranamente en el proceso de pruebas, la prueba de caja negra tiende a aplicarse durante las últimas etapas de la prueba. Puesto que, a propósito, la prueba de caja negra no considera la estructura de control, la atención se enfoca en el dominio de la información.

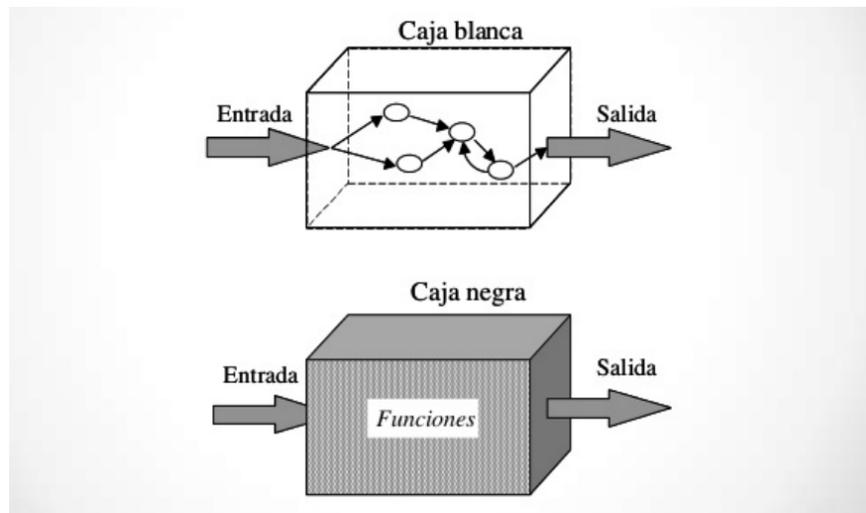


Figura A.5. Métodos de prueba