

Universidad de las Ciencias Informáticas

Facultad 4



**Herramienta para la evaluación de
videojuegos.**

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

Autor(a):

Patricia Ponce de León Atiés

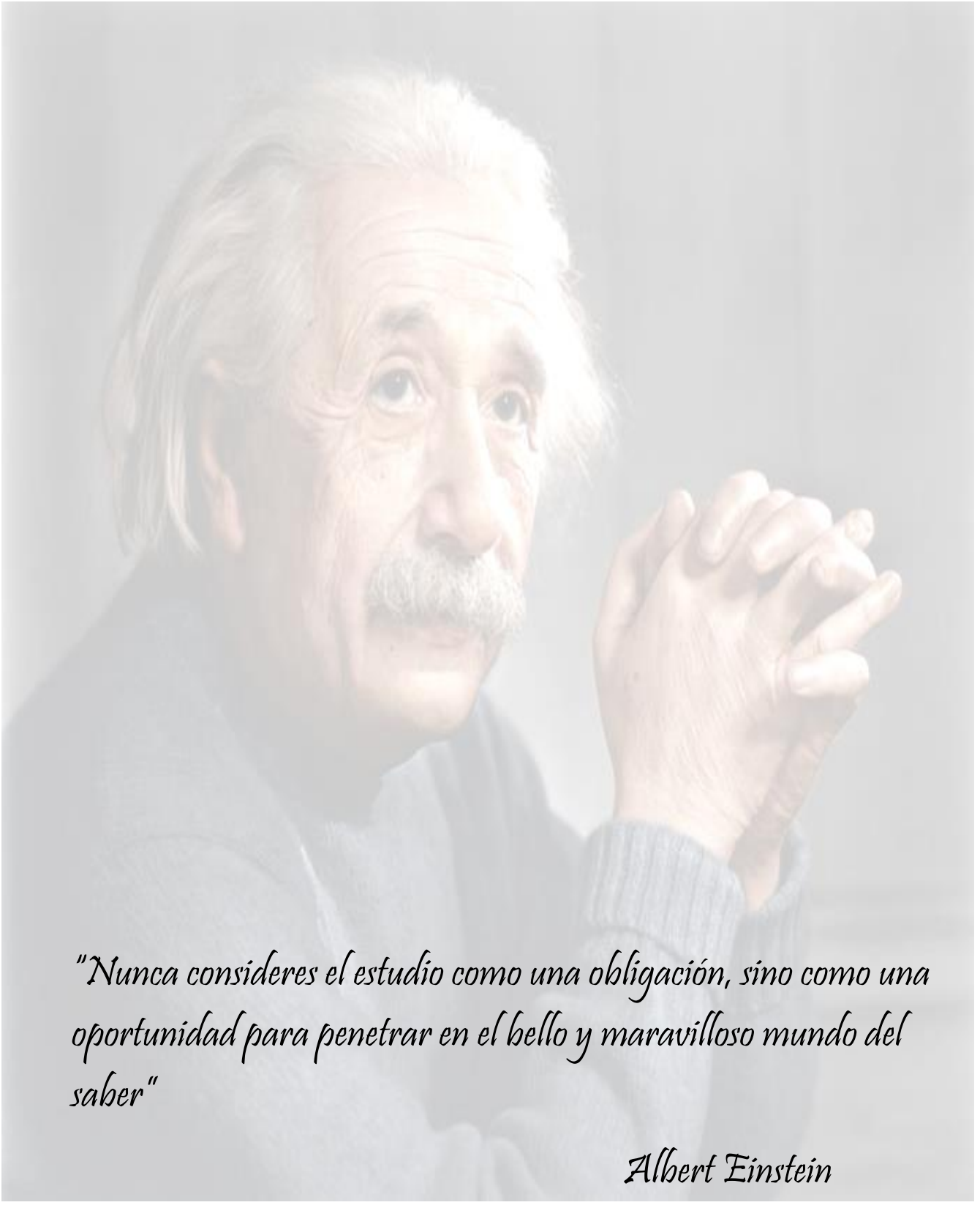
Tutor(es):

MSc. Marvyn Amado Márquez Rodríguez,

Ing. Diannys Batista Rodríguez

La Habana, Junio 2017

“Año 59 de la Revolución”

A portrait of Albert Einstein, showing him from the chest up. He has his characteristic wild, white hair and a mustache. He is wearing a dark, textured sweater. His hands are clasped together in front of him, resting on a surface. The background is a soft, out-of-focus grey.

"Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber"

Albert Einstein

Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 4 días del mes de julio del año 2017.

Autor:

Patricia Ponce de León Atiés

Tutores:

MSc. Marvyn Amado Márquez Rodríguez

Ing. Diannys Batista Rodríguez

Dedicatoria

A mi mamá por todo el amor y dedicación que me brindó, por apoyarme siempre en los momentos difíciles, por ser una madre y mujer ejemplar.

A mi novio por todos los años que hemos pasado juntos, por estar en los buenos y malos momentos.

A mi papá, mi tía Mercedes y mi hermano por brindarme su apoyo para poder realizar este sueño, por motivarme a superarme cada día.

A mi hermanita Elayne por estar siempre a mi lado apoyándome.

A todos de manera general, quiero dedicarles este momento y honrarlos por tanto amor y dedicación.

Agradecimientos

A esa persona tan especial que ya no está a mi lado, mi mamá, a ella le debo todo lo que soy, por ser mi guía, un ejemplo digno de seguir, por darme la vida, por darme su amor incondicional, por siempre estar presente cuando la necesite, por todos los momentos felices que pase junto a ella y por siempre complacerme, aunque mis deseos fueran imposibles, por sus consejos y por ser la persona que impulso mi carrera como profesional por todo eso y mucho más a la mejor madre que pude tener.

A mi papá por los consejos que me brindó, por apoyarme en todo momento, aunque mis decisiones no fueran las mejores, porque a pesar de la distancia siempre estuvo cerca de mí ayudándome y apoyándome para que me convirtiera en ingeniera. A mi hermano, por enseñarme y quererme.

A mi tía Mercedes por su apoyo incondicional, por estar siempre pendiente de mí, y brindarme su amor como si fuera una hija.

A mi novio Jorge Luis, por ser mi soporte cuando para mí todo estaba perdido, por quererme a pesar de mis locuras, por todo el amor que me ha dado. A su familia por brindarme su apoyo incondicionalmente. A mi hermanita Elayne gracias por quererme tanto, animarme y ayudarme. A mi prima Danays por ser mi segunda madre, cuando más la necesité.

A mis tutores en especial a Marvyn por confiar en mí cuando nadie lo hizo y por ayudarme en todo lo que necesite, por ser un excelente tutor, gracias por todo.

A mis amigas Lainet y Leyanet por brindarme su amistad y confianza. A Jose por ser mi mejor amigo y ayudarme cada vez que lo necesite. A Randy y Damian por apoyarme y ayudarme. A Sheila y Yasser gracias por brindarme su amistad.

A todos mis compañeros de aula 4504 y a los que no pudieron terminar quiero agradecerles por poder contar con su amistad. A todos los profesores que a lo largo de la carrera me enseñaron, especialmente a Yadira, Andy y Juan por la ayuda que me brindaron en la confección de esta tesis.

Resumen

El Centro Nacional de Calidad de Software (Calisoft) es una organización que tiene la misión de evaluar la calidad de la producción nacional de software y de los procesos que se llevan a cabo para desarrollarlos. Actualmente Calisoft no cuenta con una estrategia para evaluar videojuegos; lo que imposibilita por parte del equipo de prueba poder comparar resultados, preparar las precondiciones y confeccionar informes centrados en los elementos esenciales a valorar en este tipo de software. El presente trabajo tiene como objetivo crear una herramienta que permita evaluar los videojuegos de producción nacional. Para el desarrollo de la investigación se realizó un estudio perteneciente al estado del arte, además del análisis del proceso de calidad de software. Se utilizó como metodología de desarrollo de software el Proceso Unificado Ágil (AUP) en su versión UCI. Como resultado de la investigación se obtuvo una aplicación web que permite registrar las evaluaciones emitidas del equipo de prueba de Calisoft, teniendo en cuenta los indicadores fundamentales para medir la calidad de un videojuego.

Índice

Declaración de Autoría	II
Dedicatoria	III
Agradecimientos.....	IV
Resumen.....	V
Introducción.....	1
Capítulo 1: Fundamentación Teórica	4
1.1 Calidad.....	4
1.2 Calidad de software	5
1.3 Evaluación de software.....	6
1.4 Normas de calidad de software	7
1.4.1 ISO/IEC 25010	7
1.5 Proceso en Calisoft.....	8
1.6 Videojuegos	10
1.6.1 Géneros de Videojuegos.....	10
1.6.2 Elementos básicos de un videojuego.....	11
1.6.3 Evaluación a videojuegos.....	14
1.6.4 Escalas de Evaluación de Videojuegos.....	15
1.7 Indicadores para evaluar un videojuego	16
1.7.1 Jugabilidad:	17
1.7.2 Sonido:	17
1.7.3 Gráfico:.....	18
1.7.4 Narrativa:.....	18
1.7.5 Inteligencia Artificial:.....	19
1.7.6 Innovación.....	20

1.8 Herramientas y tecnologías a utilizar en la propuesta de solución	20
1.8.1 Metodología	20
1.8.2 Herramienta Case	21
1.8.3 Lenguajes de desarrollo	22
1.8.4 Framework	24
1.9 Consideraciones Parciales	25
Capítulo 2: Descripción de la Propuesta de Solución.....	26
2.1 Descripción del sistema propuesto	26
2.2 Modelo de Dominio.....	26
2.2.1 Definición de las clases del Modelo de dominio:.....	27
2.2.2 Diagrama de dominio:	27
2.3 Descripción de la Aplicación:.....	28
2.3.1 Requisitos Funcionales	28
2.3.2 Requisitos no funcionales	35
2.4 Identificación de los actores y casos de uso del sistema.....	38
2.4.1 Casos de uso del sistema	38
2.4.2 Diagrama de casos de uso del sistema.....	38
2.4.3 Descripciones de los CU	38
2.5 Patrones arquitectónicos	49
2.5.1 Modelo Vista Controlador.....	49
2.6 Modelo del diseño.....	50
2.6.1 Patrones de diseño utilizados	50
2.6.2 Diagramas de clases del diseño	53
2.7 Modelo de datos	54
2.8 Consideraciones parciales.....	55

Capítulo 3: Implementación y pruebas	56
3.1 Estándar de codificación.....	56
3.2 Modelo de implementación	56
3.2.1 Diagrama de componentes	57
3.2.2 Modelo de despliegue	57
3.3 Pruebas	58
3.3.1 Pruebas de caja negra	58
3.3.2 Pruebas de regresión	62
3.4 Resultados de las pruebas	63
3.5 Consideraciones parciales.....	64
Conclusiones Generales	65
Recomendaciones.....	66
Referencias Bibliográficas	67

Índice de Figura

Figura 1. Características de calidad de ISO/ICE25010 (6).....	8
Figura 2. Escala de evaluación de MeriStation. (12).....	16
Figura 3. Modelo de Dominio.	27
Figura 4. Diagrama de CU del sistema	38
Figura 5. Patrón arquitectónico MVC para Framework Yii.	50
Figura 6. Ejemplo de utilización del patrón Creador.....	51
Figura 7. Ejemplo de utilización del patrón Alta Cohesión	52
Figura 8.DCD del CU: Evaluar videojuego	53
Figura 9. Modelo de datos.....	55
Figura 10. Diagrama de Componentes	57
Figura 11.Diagrama de Despliegue.....	58
Figura 12. Gráfico de no conformidades.	63

Índice de Tablas

Tabla 1. Tabla de evaluación de 3DJuegos.	15
Tabla 2. Descripción de los requisitos funcionales.....	35
Tabla 3. Descripción de los actores del sistema	38
Tabla 4. CU6: Evaluar videojuego.....	49
Tabla 5. DCP del CU: Evaluar videojuego, Sección 1: Evaluar narrativa	62
Tabla 6. Descripción de las variables del CU: Evaluar videojuego, Sección 1: Evaluar narrativa.	62
Tabla 7. Cantidad de no conformidades por iteración	63

Introducción

En este momento los videojuegos son la puerta de entrada de niños y jóvenes en las Tecnologías de información y la comunicación (TIC). Mediante el videojuego se adquieren capacidades y desarrollan habilidades diversas, las más importantes son la familiarización con las nuevas tecnologías, su aprecio y su dominio. Por este motivo el videojuego es en estos momentos un elemento determinante para socializarse en el mundo de las nuevas tecnologías.

Actualmente la competencia en el mercado de los videojuegos es elevada. Las grandes empresas de videojuegos para garantizar la calidad entregan el producto a expertos antes de la fecha de lanzamiento. Los expertos prueban el videojuego y realizan un análisis profundo para emitir una evaluación, que influye en el éxito del juego.

Cuba no es la excepción y también ha decidido competir en este mercado a pesar de las limitaciones que presenta. En la Universidad de las Ciencias Informáticas (UCI) se ha creado una infraestructura productiva donde se han realizado productos de exportación. Además, cuenta con la tarea de formar personal altamente capacitado; con la finalidad de producir software de calidad para poder competir en el mercado internacional.

En el centro VERTEX de la Facultad 4 se está desarrollando diversos videojuegos con fines educativos. Para garantizar que estos videojuegos salgan al mercado con una garantía que avale la calidad para su liberación, deben ser revisados por el Centro Nacional de Calidad de Software (Calisoft). Esta organización tiene la misión de evaluar y certificar la calidad de la producción nacional de software y de los procesos que se llevan a cabo para desarrollarlos. Además, brinda asesoría y preparación a los especialistas en el país en los temas de calidad de software.

Actualmente Calisoft no cuenta con una estrategia para evaluar videojuegos. Los probadores solo juegan el videojuego y emiten su criterio, dando como resultado un análisis poco profundo debido a que no evalúan las características esenciales de un videojuego. La inexistencia de una estrategia para evaluar hace imposible por parte

del equipo de prueba poder comparar resultados, preparar las precondiciones y confeccionar los informes que le permitan al equipo de desarrollo mejorar el desarrollo de los videojuegos. Con estos fines se realiza la investigación cuyo **problema científico** es:

¿Cómo mejorar el proceso de evaluación de videojuegos en Calisoft?

El **Objeto de Estudio** de la investigación son las evaluaciones a videojuegos y su **Campo de Acción** son los indicadores para evaluación de los videojuegos. Para dar solución al problema se trazó como **objetivo general**:

Crear una herramienta que permita la evaluación de videojuegos.

Con lo planteado anteriormente se puede obtener como **posibles resultados**:

- Indicadores para medir la calidad a videojuegos.
- Herramienta para la evaluación de videojuegos.

Para dar cumplimiento al objetivo general se plantean las siguientes **tareas de investigación**:

1. Análisis y síntesis de la teoría sobre la calidad de software.
2. Análisis y síntesis de la teoría sobre el desarrollo de Videojuegos.
3. Análisis y Diseño de indicadores para evaluar videojuegos.
4. Diseño y Desarrollo de una herramienta informática que apoye a la evaluación de videojuegos.
5. Validación de la herramienta informática.

Métodos de Investigación:

Para el desarrollo de la investigación, se utilizaron los siguientes métodos teóricos y empíricos:

Métodos teóricos:

Analítico-sintético: se consultará la bibliografía necesaria para dar cumplimiento a las tareas de la investigación y se realizará un resumen de los principales aspectos de cada una de ellas.

Análisis histórico-lógico: se analizará el proceso de pruebas en Calisoft, para conocer las diferentes actividades que se realizan y cómo se lleva a cabo su desarrollo.

Métodos empíricos:

Observación: permitirá analizar todas las actividades que se realizan para evaluar videojuegos.

Entrevista: se realizarán a especialistas de Calisoft para conocer como llevan a cabo el proceso de evaluación.

Resultados esperados:

Con la elaboración de este trabajo de diploma se esperan los siguientes resultados: la evaluación a videojuegos en Calisoft será más precisa y se ajustará a los elementos específicos de un videojuego, de esta manera se elevará la calidad de las evaluaciones a videojuegos.

Capítulo 1: Fundamentación Teórica

El marco teórico de la investigación está dirigido principalmente, a destacar los aspectos más significativos sobre los principales conceptos, términos y estrategias relacionadas con las evaluaciones a videojuegos, con el objetivo de lograr una mejor evaluación en los videojuegos que se desarrolla en el centro de VERTEX. Para profundizar en los aspectos investigativos se utilizaron diferentes fuentes bibliográficas, gracias a estos se pudo conocer algunos criterios y valoraciones que abordan diferentes autores sobre una temática específica.

1.1 Calidad

La calidad es una herramienta básica e importante para una propiedad inherente de cualquier cosa que permite que la misma sea comparada con cualquier otra de su misma especie. La palabra calidad tiene múltiples significados. De forma básica, se refiere al conjunto de propiedades inherentes a un objeto que le confieren capacidad para satisfacer necesidades implícitas o explícitas. Por otro lado, la calidad de un producto o servicio es la percepción que el cliente tiene del mismo, es una fijación mental del consumidor que asume conformidad con dicho producto o servicio y la capacidad del mismo para satisfacer sus necesidades. Por tanto, debe definirse en el contexto que se esté considerando, por ejemplo, la calidad del servicio postal, del servicio dental, del producto, de vida, etc.

Según la NC/ISO9000:2005: Es el grado en el que un conjunto de características inherentes cumple con los requisitos.

Crosby la define como el cumplimiento de normas y requerimientos precisos, la calidad es ajustarse a las especificaciones. Su lema es "hacerlo bien, a la primera vez y conseguir cero defectos", confirmando que la calidad está basada en cuatro principios absolutos: cumplimiento de requisitos, sistema de prevención, su estándar de realización es cero defectos y su medida es el precio del incumplimiento (1).

Para Juran, calidad es el conjunto de características que satisfacen las necesidades de los clientes, además calidad consiste en no tener deficiencias. La calidad es "la adecuación para el uso, satisfaciendo las necesidades del cliente" (2).

Para Deming, calidad es el grado predecible de uniformidad y fiabilidad a un bajo costo y que se ajuste a las necesidades del mercado. La calidad no es otra cosa más que "una serie de cuestionamiento hacia una mejora continua" (3).

Todos estos conceptos se refieren a elementos comunes y fundamentales muy relacionados a la mejora de la eficacia. La diversidad de los mismos se explica por el sobredimensionamiento del significado de la categoría calidad. Se puede llegar a la conclusión que se obtiene un producto con calidad cuando:

1. Cumple el propósito con el que fue creado.
2. Satisface las necesidades del cliente.

1.2 Calidad de software

La rápida evolución de los productos y la aparición de sistemas han favorecido la existencia de un escenario de alto grado de intensidad competitiva entre las empresas productoras de software. Actualmente, los usuarios reclaman productos y servicios informáticos con mayor calidad, sustentados en requerimientos desafiantes y mayores necesidades.

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad (4).

La Organización Internacional de Estándares (ISO) define la calidad del software como: "la totalidad de características de un producto de software que le confiere la capacidad de satisfacer necesidades explícitas e implícitas" (5).

La calidad del software es medible y varía de un sistema a otro o de un programa a otro. Un software elaborado para el control de naves espaciales debe ser confiable al nivel de "cero fallas", al igual que un software donde se impliquen vidas humanas,

mientras que un software hecho para ejecutarse una sola vez, no tiene un impacto muy fuerte o no requiere el mismo nivel de calidad.

La calidad del software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software.

1.3 Evaluación de software

El control de la calidad del software es un paso importante en el desarrollo del software, una manera de realizar este control es mediante la evaluación del software. Controlar y corregir los fallos en un software afecta de manera positiva algunos atributos de la calidad.

En términos generales, se pueden distinguir dos tipos de evaluaciones durante el proceso de desarrollo: Verificaciones y Validaciones. Según el IEEE Std 729-1983 éstas se definen como:

Verificación: Proceso de determinar si los productos de una cierta fase del desarrollo de software cumplen o no los requisitos establecidos durante la fase anterior.

Validación: Proceso de evaluación del software al final del proceso de desarrollo para asegurar el cumplimiento de las necesidades del cliente.

En otras palabras, la verificación comprueba si se ha construido el producto correctamente, que se vincula especialmente con errores de los requisitos de diseño, del diseño al código, etc. Mientras que la validación ayuda a comprobar si se ha construido el producto correcto, que se relaciona con errores al malinterpretar las especificaciones del cliente. El cliente es la única persona que puede validar el software, debido a que es el que puede detectar si se interpretó sus necesidades adecuadamente.

En esta investigación se toma el proceso de validación especificando que el cliente no es el encargado validar el software, sino la institución encargada de evaluar el videojuego.

1.4 Normas de calidad de software

Para evaluar la calidad de software de manera estandarizada surgen las normas de calidad de software que comúnmente se interpretan como un conjunto de reglas, parámetros o principios que definen un comportamiento, o regulan un proceso.

Una norma de calidad es una regla o directriz para las actividades, diseñadas con el fin de conseguir un grado óptimo de orden en el contexto de la calidad. Es un documento, establecido por consenso y probado por un organismo reconocido (nacional o internacional), que proporciona, para un uso común y repetido, reglas, directrices o características para las actividades de calidad o sus resultados, con el fin de conseguir un grado óptimo de orden en el contexto de la calidad.

La estandarización es toda actividad documentada que norma el comportamiento de un grupo de personas. Los estándares nos dan los medios para que todos los procesos se realicen siempre de la misma forma. Son nuestra guía para la productividad y la calidad.

Existen varias organizaciones de estandarización internacional, algunas son regionales mientras que otras son globales. Como por ejemplo la *International Telecommunication Union* (ITU). La *International Electrotechnical Commission* (IEC) que fue fundada en el año 1906 para definir estándares en eléctrica y electrónica, mientras que la *International Organization for Standardization* (ISO) fue creada en 1947 para promover el uso de estándares propietarios, industriales y comerciales a nivel mundial. Ambas tienen por objetivo facilitar el intercambio de bienes y servicios a nivel internacional.

1.4.1 ISO/IEC 25010

El modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software determinado.

La calidad del producto software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor.

Son precisamente estos requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y subcaracterísticas.

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad que se muestran en la siguiente figura:



Figura 1. Características de calidad de ISO/ICE25010 (6).

1.5 Proceso en Calisoft

En Cuba la institución encargada de evaluar el software es Calisoft. Dicha institución tiene como principales objetivos ofrecer servicios de evaluación de la calidad del proceso de desarrollo de software, los procesos de prestación de servicios de tecnología de la información y los productos informáticos. Además, tiene la tarea de establecer estándares técnicos y procedimientos que normalicen la industria informática, dirigiéndola hacia niveles de calidad superiores. El estándar de calidad que utilizan es la ISO25010 descrita anteriormente.

Cuenta con profesionales capacitados y con experiencia en consultorías de mejora de procesos, diagnósticos, auditorías, revisiones, evaluación de productos, pruebas de aceptación y pruebas de adquisiciones.

El proceso de prueba empleado en Calisoft consiste en 4 etapas: Planificación, Diseño, Ejecución y Cierre.

Planificar evaluación del producto de software:

Se elabora un cronograma o Plan de evaluación el cual definirá el tiempo aproximado de la evaluación del producto. Además, en esta etapa se definen los ingenieros de prueba y probadores que participarán en la evaluación del producto.

Diseñar evaluación del producto de software:

Se realiza la reunión de inicio en la cual se aprueba el Plan de la evaluación del producto. Si se cumple con los criterios de inicio de la evaluación, los Ingenieros de pruebas diseñan los casos de pruebas.

Ejecutar evaluación del producto de software:

En esta etapa se realizan las pruebas exploratorias y teniendo en cuenta los resultados obtenidos se determina si se continúa con el proceso de prueba o debe de ser detenido o abortado. En caso de que se detenga el proceso se reinicia la evaluación una vez concluido el plazo de detención del mismo, en caso de abortado se concluye con el proceso y en caso de que el mismo se continúe los probadores ejecutan las pruebas correspondientes a cada una de las subcaracterísticas: Usabilidad, Seguridad, Eficiencia, Funcionalidad, Confiabilidad y las pruebas de integración se realizan en caso necesario.

Cerrar evaluación del producto de software:

En la presente etapa se prepara el cierre de la evaluación, para ello como principal actividad a desarrollar es la realización de la reunión de cierre o de pruebas abortadas; según sea el caso, como fin del proceso de evaluación, además de actualizar y cerrar el expediente de la evaluación. El cliente debe de entregar la Encuesta de satisfacción, donde refleja el grado de satisfacción con el proceso de evaluación.

Teniendo en cuenta el proceso de desarrollo de software de Calisoft, la etapa que se enmarca esta investigación es en Ejecutar evaluación a productos de software. Proponiéndole un paso adicional donde se evidencie la evaluación final del videojuego.

1.6 Videojuegos

En la presente investigación el software que se evalúa son los videojuegos, para ello que se debe conocer su definición. Según la Real Academia Española un videojuego es un dispositivo electrónico que permite, mediante mandos apropiados, simular juegos en las pantallas de un televisor o de una computadora.

Actualmente el videojuego es considerado todo tipo de cosas, desde ergódico hasta lúdico. Se lo considera narración, simulación, performance, re-mediación y arte; una herramienta potencial para la educación o un objeto de estudio para la psicología del comportamiento; un medio para la interacción social, un juguete y un medio de distracción. Asimismo, el campo emergente de la teoría del videojuego también constituye el punto de convergencia de una gran variedad de enfoques, que incluyen la teoría del cine y la televisión, la semiótica, la teoría de la performance, los estudios del juego, la teoría literaria, la informática, las teorías del hipertexto, el cibertexto, la interactividad y la identidad, el posmodernismo, la ludología, la teoría de los medios de comunicación, la narratología, la estética y la teoría del arte, la psicología, las teorías de los simulacros, entre otros (7).

En la actualidad podemos ejecutar un videojuego en diversas plataformas como un dispositivo portátil, una consola de sobremesa, una maquina arcade o un ordenador. Estas plataformas han ido evolucionando a través de los tiempos, las cuales se han dividido por generaciones. La generación actual es la octava, lo destacable de esta generación es el uso de internet como eje central de la funcionalidad de las consolas.

1.6.1 Géneros de Videojuegos

Hoy en día es difícil clasificar un videojuego por su género, ya que algunos no cuentan con un género específico y a veces son combinaciones de varios géneros.

Según Oliver en su investigación define 4 criterios de clasificación de géneros de videojuegos, estos son según: la dialéctica asimilación vs. acomodación, la estructura del juego, la finalidad del jugador implícito y la dominante de mecánicas de juego (8).

En esta investigación se tomará la clasificación de géneros de videojuego según la finalidad del jugador implícito que consta de seis macro-géneros fundamentales de videojuegos estos son:

Videojuego de acción: videojuegos orientados al reto competitivo de victoria/derrota con tendencia a jugabilidad rígida (8).

Videojuego de estrategia: videojuegos orientados al reto competitivo de victoria/derrota con jugabilidad abierta (8).

Videojuego de aventura: videojuegos orientados al descubrimiento de una trama narrativa con tendencia a jugabilidad rígida (8).

Videojuego de rol: videojuegos orientados a la (re)construcción de una narrativa con jugabilidad abierta. El género del juego de rol se caracteriza por la adopción del jugador de una máscara ficcional, el rol de su personaje, y por la búsqueda de puntos de experiencia, que se obtienen por diversas acciones meritorias en el juego y que sirven para mejorar progresivamente las habilidades del personaje (8).

Videojuego de simulador: videojuegos orientados a la comprensión sobre el funcionamiento de un sistema a través de la experimentación, con tendencia relativa a una jugabilidad rígida (en relación con la rigidez en la definición sobre cómo manejar correctamente la máquina) (8).

Videojuego de simulación (fundamentalmente, simulación social): videojuegos orientados a la comprensión sobre el funcionamiento de un sistema a través de la experimentación, con jugabilidad abierta. . El género más popular en este ámbito hasta el momento son las simulaciones sociales (8).

1.6.2 Elementos básicos de un videojuego

Una vez definido los diferentes géneros se hace necesario identificar cuáles serían los elementos básicos de un videojuego. Los mismos van hacer utilizado como variables de entrada para la aplicación que acompaña a esta investigación.

Por tanto, se definen como aspectos fundamentales de un videojuego: la jugabilidad, la narrativa, el grafico y el sonido.

Jugabilidad

El término jugabilidad ha sido definido de distintas maneras por diversos autores:

Según González Sánchez, es el conjunto de propiedades que describen la experiencia del jugador ante un sistema de juego determinado, cuyo principal objetivo es divertir y entretener de forma satisfactoria y creíble ya sea solo o en compañía de otros jugadores (9).

Según Björk, son las estructuras de interacción del jugador con el sistema del juego y con otros jugadores en el juego (10).

Por lo que se asume que jugabilidad sería aquello que hace el jugador.

Narrativa

La narrativa recoge una serie de hechos presentados o explicados por un narrador, que suceden a uno o más personajes que son los que realizan las acciones. Posee numerosos subgéneros, entre los que destacan especialmente la epopeya, la novela y el cuento o relato corto. Existen dos tipos de narrativas: la interactiva y la no interactiva. La no interactiva es la que se encuentra en libro o espectáculos directos. La interactiva es la que se puede ver en los videojuegos.

Gracias a la narrativa el videojuego Pong no son dos simples líneas y un punto, sino una especie de partido de tenis; y cuando se juega está la historia de cómo vas aprendiendo a dominar los controles, cómo te enfrentas a un amigo, cómo salvas un punto casi imposible, cómo remontas en el último suspiro, cómo te pide la revancha. Aquello que surge de tu experiencia como jugador también es una historia.

Gráficos

Para muchos lo primero que llama la atención, es los gráficos. Recientemente la industria se ha dividido de forma notable. Por un lado, se han visto muchos lanzamientos que atesoran una calidad visual abrumantes y que luego han podido ser buenos o no en términos de jugabilidad. Por otra se han visto muchos títulos de estética mucho más modesta que han ofrecido resultados iguales o incluso en

ocasiones superiores. Los gráficos desde el punto de vista de jugabilidad añade aspectos a la inmersión.

Los grandes éxitos de ventas de las dos últimas generaciones han contado con gráficos increíbles, a veces incluso con calidades más que discutibles en cuanto a la diversión. También hay muchos éxitos que han contado con cifras millonarias y unos gráficos muy modestos, sin embargo, casi siempre se circunscriben a móviles, mientras que la escena *indie* logra triunfar comercialmente con mucho encanto en la puesta en escena y propuestas de jugabilidad de gran atractivo.

Por tanto, se puede concluir que los gráficos son tan importantes como la jugabilidad.

Sonido

Hace unos años cuando se crearon las primeras consolas, apenas podíamos escuchar unos cuantos *beeps*. Con el paso de los años, los bits fueron aumentando y los sistemas y motores de programación de videojuegos fueron igual evolucionando hasta llegar a lo que tenemos ahora. A parte de acompañar la imagen, el sonido tiene todo tipo de labores, principalmente en los campos de la narrativa y el contenido emocional del juego, manipulando de diversas formas lo que el jugador entiende, siente y como se desarrolla dentro del juego.

En la parte narrativa, utilizando los diálogos, el jugador entiende y desarrolla su propia historia. Igualmente, con la mezcla interactiva y la ubicación de sonidos en el entorno, se le otorga al jugador la noción de tiempo y espacio, ubicándolo así dentro de una historia o un contexto. En la parte emocional, utilizando todos los aspectos del sonido, se pueden manipular las emociones de muchas maneras, como hacer que el jugador se enfurezca, llore de felicidad o se emocione al llegar a cierto lugar o lograr alguna misión.

Por tanto, los sonidos incluidos en los videojuegos tienen un papel fundamental: hacer que el jugador se adentre en el desarrollo de la trama, el ambiente e incluso en la personalidad de los protagonistas, lo que da como resultado una experiencia que provoca que se sienta en carne propia alegría, tranquilidad, miedo, emoción.

1.6.3 Evaluación a videojuegos

En la actualidad los videojuegos son difíciles de evaluar debido a todos los elementos que lo componen. Las evaluaciones las realizan expertos en el tema, y estas evaluaciones también se les conoce como análisis.

Existen ferias de videojuegos en las que se evalúan y se otorgan premios, La *Electronic Entertainment Expo*, también conocida como E3, es la convención de videojuegos más importante de la industria, se celebró en 1995 por primera vez. Cuando culmina E3 se otorgan un conjunto de premios conocidos como *The Game Critics Awards*, estos se realizan desde 1998. Los premios se otorgan a los productos mostrados en el E3 con el título *Best of E3* de su categoría. Los nominados y los ganadores de los premios son elegidos por jueces individuales que representan a 35 (a partir de 2011) de los principales medios de comunicación de América del Norte. Los premios se otorgan independientemente de los organizadores de la Expo. No se debe confundir *The Game Critics Awards* con *The Game Awards*.

The Game Awards (en español, los premios a los videojuegos) son una gala de videojuegos realizada por primera vez el 5 de diciembre de 2014 en el teatro de AXIS de Las Vegas. Estos premios cuentan con el apoyo de grandes empresas de la industria, tales como *Activision*, *Electronic Arts*, *Kojima Productions*, *Konami*, *Xbox*, *Nintendo*, *Rockstar*, *Sony Computer Entertainment*, *Valve Corporation*, *Ubisoft* y *Warner Brothers*. Además de premiar a los mejores juegos del año en diversas categorías, en el 2015 incluyeron premios a los jugadores, en particular, de los *youtubers* y jugadores de *eSports*. Los premios para los videojuegos son elegidos por un jurado, mientras que los premios a los jugadores se hacen vía online a través de su página oficial.

También existen varias comunidades de videojuegos que se dedican a esto, entre ellas se encuentra 3DJuegos, MeriStation, Vandal, HobbyConsolas, entre otras. Estas evalúan principalmente en los videojuegos los gráficos, sonidos y jugabilidad. Otros aspectos que también evalúan es la innovación/originalidad, duración, la historia y detalles que se definan dependiendo del género del videojuego.

1.6.4 Escalas de Evaluación de Videojuegos

Los críticos de videojuegos se basan para evaluar juegos mayormente en una serie de indicadores a los que le asignan valores que van desde el 1 al 10. Después de evaluar cada indicador por separado, evalúan el juego como conjunto dependiendo de los indicadores más importantes para el tipo de videojuego. Dándole mayor puntuación a los mejores juegos.

Por ejemplo, 3DJuegos es una comunidad de críticos donde realizan evaluaciones a videojuegos. 3DJuegos usa una escala numérica de puntuación cuyo rango oscila del 1 al 10, con intervalos de medio punto. Dentro de este esquema un 1 representa la peor puntuación posible, mientras que un 10 hace lo propio con la mejor. Adicionalmente a la propia puntuación del juego, el sistema de valoración incluye varios indicadores que puedes usar como referencia, por ejemplo, sus puntos más o menos destacados (+ / -), la duración estimada o las posibilidades multijugador que presenta. Asimismo, factores importantes como la jugabilidad, gráficos, sonido e innovación que presenta cada videojuego son representados por estrellas. A mayor número de estrellas, más destacable es el juego en dicho apartado, con unos valores que no tienen por qué afectar a la nota final. (11)

3dJuegos define las evaluaciones de la siguiente manera:

Rango de evaluación	Significado
(1,0 a 4,5)	No te lo recomendamos.
(5,0 a 6,5)	Te puede interesar.
(7,0 a 7,5)	Recomendable.
(8,0 a 8,5)	Muy recomendable.
(9,0 a 9,5)	Imprescindible.
(10)	Para el recuerdo.

Tabla 1. Tabla de evaluación de 3DJuegos.

Meristation es otra comunidad de críticos en la que se basan en realizar un análisis completo de los aspectos positivos y negativos del videojuego. A diferencia de 3DJuegos mide tres factores importantes que son sonido, gráficos y jugabilidad con una escala de 1 al 10 y la puntuación global del juego en la misma escala.

Meristation define su sistema de evaluación de la siguiente manera:

0 Atroz	6 Correcto
1 Pésimo	7 Bueno
2 Muy Malo	8 Muy Bueno
3 Malo	9 Excelente
4 Mediocre	10 Obra Maestra
5 Mejorable	

Figura 2. Escala de evaluación de MeriStation. (12)

Para esta investigación se tomará como escala para evaluación un formato numérico tomando valores desde 1 hasta 10, siendo 10 la puntuación máxima que puede alcanzar un videojuego. La escala queda de la siguiente manera:

1 - 3 Malo

4 - 5 Regular

6 - 7 Bueno

8 - 9 Muy Bueno

10 Excelente

1.7 Indicadores para evaluar un videojuego

Para evaluar un videojuego este debe hacerse en base a los elementos básicos que lo componen. 3DJuegos evalúa los juegos en cuanto a jugabilidad, sonido, gráficos e innovación como indicadores principales, pero también realizan un análisis donde tienen en cuenta la narrativa, la historia, los personajes. Meristation evalúa como indicadores principales, los gráficos, la jugabilidad y el sonido.

Por tanto, en esta investigación se definen como indicadores principales: la jugabilidad, el sonido, el gráfico y la innovación. Además, también se medirá la inteligencia artificial sobre todo en géneros como la estrategia y la simulación, debido a que este es un indicador importante de estos géneros. Otro indicador a tomar en cuenta será la narrativa, ya que este es imprescindible en los juegos de rol y aventura.

1.7.1 Jugabilidad:

En el indicador de jugabilidad se medirán los siguientes aspectos:

- **Control:** es un aspecto importante ya que los controles permiten al jugador tener un contexto físico que le sirve de ventana al mundo virtual, el usar las manos activando diferentes comandos en varias posiciones es una de las premisas o fundamentos que sostienen la relación entre el jugador y el juego. Se evalúan los controles del juego en cuanto a: si son intuitivos, la complejidad de los mismo a la hora de realizar una acción, si son cómodos y ofrecen la posibilidad de personalizarlos.
- **Cámara:** la cámara es importante a la hora de evaluar un videojuego ya que la perspectiva que toma a veces perjudica la jugabilidad. Se evalúa en cuanto a: la perspectiva, la cual debe ser ideal en todo momento. Dependiendo de la perspectiva se evalúa el movimiento de dicha cámara, puede tener total libertad, estar restringida hasta un determinado grado o sencillamente ser fija.
- **Mecánica:** la mecánica es el núcleo de un juego y este varía dependiendo del género del videojuego. Una mecánica es una regla de juego con una entrada y una salida que produce cambios en el sistema. Un sistema de juego está compuesto por una o más mecánicas. Por tanto, la mecánica se evalúa en cuanto a que sean divertidas y atractivas para los usuarios.
- **Curva de aprendizaje:** la curva de aprendizaje en un videojuego no debe ser muy fácil pues el jugador puede aburrirse, pero tampoco debe ser difícil pues puede frustrarse. Se evalúa la facilidad para comprender el sistema y mecánica del videojuego. Además, se tiene en cuenta el tiempo que la persona logra dominar el sistema.

1.7.2 Sonido:

En el indicador de sonido se medirán los siguientes aspectos:

- **Efectos:** los efectos de sonido tienen gran importancia ya que contribuye a la inmersión en el juego, mientras más reales sean será mejor. Se evalúa la

calidad, si se repiten demasiado, si están acorde a la acción que se está realizando.

- **Voces:** las voces pueden estar o no presentes, pero en caso de que estén presentes éstas deben ser claras y entendibles por el jugador. Se evalúa si hay un buen doblaje, si las voces están acordes con las animaciones.
- **Banda sonora:** la música ayuda a mejorar la experiencia de juego. La música tiene la gran tarea de ensalzar las emociones que el juego genera en los jugadores. Se evalúa que esté acorde a la situación, de qué manera mejora el videojuego.

1.7.3 Gráfico:

En el indicador de gráfico se medirán los siguientes aspectos:

- **Escenarios:** se evalúa el diseño de los diferentes niveles, la variedad si cuenta con muchos niveles y si es mundo abierto se evalúa el diseño del mundo. Aquí entran los diferentes errores de gráficos que se pueden encontrar en los niveles o el mundo.
- **Animaciones:** se evalúan las animaciones de manera general, la fluidez con la que se realizan las animaciones de acciones, objetos, carácter, ambientación.
- **Textura:** se evalúan la calidad de las texturas de manera general, las texturas de los personajes, objetos, niveles o mundo.
- **Efecto de luces:** se evalúan los efectos de luces, que no sean muy excesivos, pero tampoco carentes, aquí también entran las sombras de los objetos, personajes.

1.7.4 Narrativa:

En el indicador de narrativa se medirán los siguientes aspectos:

- **Historia:** cualquier videojuego de aventura o rol se basa en una historia más o menos trabajada que sirve como coartada para las acciones que el jugador llevará a cabo durante sus partidas. Este aspecto se evalúa en cuanto al argumento, los personajes, los antagonistas, el conflicto y los sucesos.

- **Misiones:** las misiones son un elemento que ayudan mucho a la narrativa del juego, uno de los puntos débiles de la narrativa de los videojuegos son sus misiones secundarias. Esta debilidad estriba en la ruptura con la continuidad de la historia, en la reiteración y a veces excesiva repetición de misiones secundarias, e incluso en la incoherencia que hay entre ellas y las misiones de la historia principal. Este criterio se evaluará dependiendo de cómo se relacionan las misiones principales con la historia, si las misiones secundarias tienen continuidad de las principales, si se repiten mucho, y de manera general como las misiones ayudan a la narrativa.
- **Diálogos:** los diálogos son una parte muy importante de un videojuego, es su elemento más narrativo, deben ser escritos con buen estilo y encontrar su función. Algunos servirán para dar información, otros para conseguir un objeto y otros, simplemente, para despistar al jugador. Este aspecto se evaluará en cuanto a la función del diálogo, su duración y su redacción.

1.7.5 Inteligencia Artificial:

En el indicador de inteligencia artificial se medirán los siguientes aspectos:

- **Comportamiento:** en la inteligencia artificial el comportamiento es el que determina las acciones que realizan los NPC (*non-player character*), la mayoría de las veces son patrones establecidos. Este aspecto se evaluará en cuanto a la variedad de acciones que puede realizar un NPC, a los patrones si son variados y en cuanto se asemeja a un ser humano.
- **Aprendizaje:** actualmente el aprendizaje de inteligencia artificial se ha utilizado poco en videojuegos, pero sin duda es un elemento importante ya que mediante el aprendizaje los NPC varían su comportamiento haciéndolo más real, añade más variedad al videojuego y nuevos desafíos. Este aspecto se evalúa en cuanto a la capacidad que tiene la inteligencia artificial de aprender.
- **Dificultad:** este es un aspecto importante ya que si la dificultad es muy alta puede hacer que el jugador se sienta frustrado y si la dificultad es muy baja

puede terminar aburriéndolo. Este aspecto se evaluará en cuanto a la dificultad de la inteligencia artificial, no debe ser ni muy fácil, ni muy difícil.

- **Adaptabilidad:** este aspecto está estrechamente relacionado con el comportamiento y aprendizaje, y enriquece el aspecto de dificultad, ya que, si la inteligencia artificial es capaz de establecer un margen de dificultad dependiendo de las acciones del jugador, entonces se adaptaría a cualquier persona que juegue ya sea principiante o avanzado. Este aspecto se evalúa en cuanto a la adaptabilidad que tiene según las acciones del jugador.

1.7.6 Innovación

- **Mecánicas nuevas:** Actualmente las mecánicas de los juegos vienen enmarcadas por su género, por tanto, para analizar este aspecto es necesario identificar el género y las mecánicas que lo definen. Se evalúa si el juego aporta mecánicas nuevas o mezcla mecánicas de otros géneros, logrando una buena combinación.
- **Mezcla de géneros:** Como se definía anteriormente existen numerosos géneros de videojuegos. La mezcla de estos puede dar como resultado un nuevo género y una experiencia de jugador increíble. Por tanto, este aspecto se mide en cuanto a los aspectos que mezcla de otros géneros y si la combinación es buena.

1.8 Herramientas y tecnologías a utilizar en la propuesta de solución

En el siguiente epígrafe se describen las herramientas necesarias para el desarrollo de la aplicación.

1.8.1 Metodología

AUP-UCI es una variación de AUP, de forma tal que se adapta al ciclo de vida definido para la actividad productiva de la UCI. AUP-UCI cuenta con 3 fases: inicio, ejecución y cierre. Define 8 disciplinas, pero a un nivel más atómico que el definido en AUP, estas son: Modelado de negocio, Requisitos, Análisis, Implementación, Pruebas Internas, de Liberación y Aceptación y en la última fase se cubren con las áreas de procesos que define CMMI-DEV v1.3 para el nivel 2, serían CM (Gestión de la configuración), PP (Planeación de proyecto) y PMC (Monitoreo y control de

proyecto). AUP-UCI consta de 4 escenarios. Para la realización de este trabajo se escogió el escenario 1 que se aplica a los proyectos que hayan evaluado el negocio a informatizar, y como resultado se obtenga que puedan modelar una serie de interacciones entre los trabajadores del negocio/actores del sistema (usuario), donde la atención se centra en cómo el usuario va a utilizar el sistema. Este escenario se modela con los casos de usos de sistema (CUS).

1.8.2 Herramienta Case

Una Herramienta CASE es aquella que permite la creación de los artefactos necesarios, siguiendo una metodología en la construcción de un software. En la utilización de la metodología AUP, se hace necesario el uso de la misma en las fases de Análisis, Diseño e Implementación, donde se generan la mayor cantidad de artefactos, a los cuales, es muy fácil realizarles los cambios que ocurren en el diseño de un software, precisamente por ser modelados con estas herramientas. Permite mayor calidad y rapidez del software.

Visual Paradigm 8.0 para UML

Visual Paradigm es una Herramienta CASE que soporta todo el ciclo de vida del desarrollo de un software: Análisis y Diseño, Construcción, Pruebas y Despliegue. Además, permite elaborar todos los diagramas de clases, casos de uso y diagramas de actividades. Genera código y documentación desde los diagramas y posibilita el diseño de prototipos de interfaz de usuario. Proporciona además diferentes tutoriales que sirven para un mejor entendimiento de la herramienta.

Permite el diseño de software con el UML 2.0, en general proporciona un entorno unificado de diseño de software para el analista de sistemas y desarrollador de software para analizar, diseñar y mantener aplicaciones de software en una disciplina (13).

Teniendo presente las ventajas que ofrece Visual Paradigm, se decide utilizar como Herramienta CASE en el presente trabajo.

1.8.3 Lenguajes de desarrollo

Actualmente para desarrollar en la web existen diferentes lenguajes de programación, esto se debe a la necesidad de desarrollar lenguajes de programación dinámicos para la web, que permitieran interactuar con los usuarios y utilizaran sistemas de Bases de Datos. Para la implementación de la aplicación se propone usar el lenguaje PHP.

PHP 7.0

Para el desarrollo del sistema se utiliza como lenguaje de programación por parte del servidor PHP ya que es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Lo que distingue a PHP de algo como JavaScript del lado del cliente, es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El servidor web puede ser incluso configurado para que procese todos los ficheros HTML con PHP. Es extremadamente simple para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales, también es un lenguaje libre e incluye gran cantidad de funciones.

Aunque el desarrollo de PHP está centrado en programación de scripts del lado del servidor, se puede utilizar para otras funcionalidades, gracias a esta característica se pueden recopilar datos de formularios, generar páginas con contenidos dinámicos o enviar y recibir cookies. Aunque PHP puede hacer mucho más (14).

Entorno de Desarrollo Integrado (IDE)

Un entorno de desarrollo integrado consiste básicamente en un software que previamente ha sido instalado en la máquina del cliente y cuyo principal objetivo es el desarrollo de otro software. Permite mantener proyectos informáticos, los cuales se pueden implementar en diferentes lenguajes de programación, así como realizar una serie de operaciones básicas sobre ellos.

PHPStorm

Es un entorno de desarrollo basado en la plataforma JetBrains. PhpStorm posee características avanzadas tales como auto-completado de código, análisis de la

calidad del código, solución de problemas y la depuración de código, muy importante para aumentar la productividad y la eficiencia de los desarrolladores PHP. PhpStorm es compatible con los principales marcos de trabajo como: Joomla, Drupal, WordPress, Symfony, Zend Framework, Magento, CakePHP, laravel, Yii, entre otros. El editor de código ofrece finalización de código, refactorización, la prevención de errores en la marcha, y mucho más. Se incluye además tecnologías front-end de vanguardia, como HTML5, CSS, Sass, Stylus, Less, CoffeeScript, TypeScript, Emmet, y JavaScript con refactorizaciones, depuración y pruebas de unidades disponibles. Puede ver todos los cambios en tiempo real en el navegador gracias a LiveEdit (15).

Gestor de base de datos

Los Sistemas Gestores de Bases de Datos (SGBD, por sus siglas en inglés), también conocidos como sistemas manejadores de bases de datos o DBMS (*DataBase Management System*), son un conjunto de programas que manejan todo acceso a la base de datos, con el objetivo de servir de interfaz entre esta, el usuario y las aplicaciones utilizadas.

Gracias a este sistema de software específico el usuario puede gestionar la base de datos (almacenar, modificar y acceder a la información contenida en esta) mediante el uso de distintas herramientas para su análisis, con las que puede realizar consultas y generar informes (16).

MySQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por *Oracle Corporation* y está considerada como la base de datos de código abierto más popular del mundo. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Es un sistema gestor de bases de datos muy rápido, fiable y fácil de usar (17).

Se destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más empleados como PHP, Perl y Java, y su adaptación a distintos sistemas operativos.

Existen aspectos que lo diferencian de otros productos como:

1. Posibilidad de crear y configurar usuarios, asignando a cada uno de ellos permisos diferentes.
2. Facilidad de exportación e importación de datos, incluso de la base de datos completa.
3. Posibilidad de ejecutar conjuntos de instrucciones guardadas en ficheros externos a la base de datos.
4. Permite escoger entre múltiples motores de almacenamiento para cada tabla.
5. Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo (18).

Servidor web Apache

El servidor Apache es un servidor web HTTP de código abierto. Puede ser usado en varios sistemas operativos, lo que lo hace prácticamente universal. Presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido. Es usado para tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. Es posible elegir qué características van a ser incluidas en el servidor seleccionando y qué módulos se van a cargar, ya sea al compilar o al ejecutar el servidor (19).

Se estará haciendo uso del Servidor Web Apache 2.4.18.

1.8.4 Framework

Un Framework puede definirse como una solución completa que contemplan herramientas de apoyo para el desarrollo y/o implementación de una aplicación. Además, representa una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación a la que se le puede añadir elementos para construir una aplicación correcta (20).

Yii Framework

Yii es un framework PHP basado en componentes de alta performance para desarrollar aplicaciones Web de gran escala. El mismo permite la máxima reutilización en la programación web y puede acelerar el proceso de desarrollo. Para

correr una aplicación Web Yii, solo se necesita tener un servidor Web con soporte PHP 5.1.0 o superior. Yii es un framework totalmente basado en Programación Orientada a Objetos (por sus siglas en inglés OOP) (21).

Como la mayoría de los frameworks PHP, Yii es un framework MVC (modelo-vista-controlador). Yii sobresale frente a frameworks PHP en su eficiencia, su gran cantidad de características y su clara documentación. Yii ha sido diseñado cuidadosamente desde el principio para el desarrollo de aplicaciones Web. No es ni un subproducto de un proyecto ni un conglomerado de trabajo de terceros. Es el resultado de la vasta experiencia de los autores en desarrollo de aplicaciones Web y de la investigación y la reflexión de los más populares frameworks de programación Web y aplicaciones (21).

1.9 Consideraciones Parciales

En este capítulo se realizó la investigación teórica que fundamenta toda la base del desarrollo de vida de la aplicación.

Se realizó un estudio sobre los videojuegos para conocer las características principales y poder definir los indicadores.

Se decide aplicar la metodología AUP debido a que es una unión de las metodologías ágiles y las clásicas, ya que toma de cada una de ellas los mejores puntos.

Se decide utilizar PHP v7.0, Yii 2 como framework de desarrollo y el entorno de desarrollo integrado PHPStorm en el desarrollo de la aplicación.

Capítulo 2: Descripción de la Propuesta de Solución

Las características de un sistema, son los elementos claves para el desarrollo de los artefactos ingenieriles que permiten obtener una vista estática del sistema, el modelado de las mismas, permite ofrecer posteriormente una entrada coherente al análisis y diseño y que finalmente será transformada en código fuente. De esta forma al plantear las características de la propuesta de solución, se están sentando las primeras bases de la arquitectura de la futura solución.

2.1 Descripción del sistema propuesto

La aplicación para la evaluación de videojuegos *GamEva* estará dividida en tres paquetes, uno para la gestión de videojuegos, otro para la gestión de usuarios y el último para la gestión de evaluaciones. El sistema contará con varios usuarios los cuales pueden estar asignados a dos roles principales: administrador y probador.

El administrador será en el encargado de gestionar los paquetes de videojuegos y usuarios. Tiene acceso al paquete de evaluaciones, pero solo para eliminar y visualizar. El administrador no estará autorizado a crear nuevas evaluaciones ni modificarlas. El probador será el encargado de gestionar las evaluaciones, podrá acceder al paquete de videojuegos, pero solo listar los videojuegos disponibles y evaluarlos.

El paquete de usuarios permitirá realizar una total gestión de los usuarios, así como la gestión roles y permisos y la asignación de estos a los usuarios. El paquete de videojuegos permitirá realizar una total gestión de los videojuegos, géneros de videojuegos y plataformas. El paquete de evaluación permitirá al probador evaluar los videojuegos disponibles en el sistema de acuerdo a los indicadores definidos anteriormente en el Capítulo 1 epígrafe 1.7. El probador deberá asignar una puntuación entre 1 y 10 a los aspectos de cada indicador. Una vez evaluado los indicadores el sistema será capaz de emitir una evaluación final del videojuego.

2.2 Modelo de Dominio

El modelo de dominio es un modelo conceptual de todos los temas relacionados con un problema específico. En él se describen las distintas entidades, sus atributos,

papeles y relaciones, además de las restricciones que rigen el dominio del problema.

2.2.1 Definición de las clases del Modelo de dominio:

- **Indicadores:** Son los indicadores que se miden para realizar una evaluación a un videojuego.
- **Jugabilidad:** Es el indicador referente a la jugabilidad del videojuego.
- **Sonido:** Es el indicador referente al sonido del videojuego.
- **Gráfico:** Es el indicador referente al gráfico del videojuego.
- **Narrativa:** Es el indicador referente a la narrativa del videojuego.
- **Innovación:** Es el indicador referente a la innovación del videojuego.
- **Inteligencia Artificial:** Es el indicador referente a la inteligencia artificial del videojuego.
- **Probadores:** Son las personas encargadas de probar el videojuego y emitir su criterio.
- **Evaluación:** Son las evaluaciones que emiten los probadores del videojuego que probaron.
- **Videojuego:** Es el videojuego que se va a evaluar.

2.2.2 Diagrama de dominio:

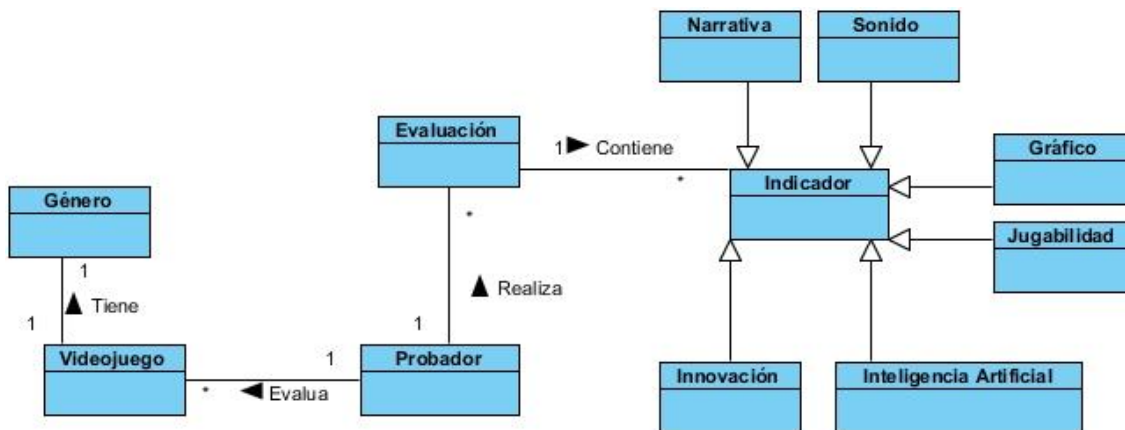


Figura 3. Modelo de Dominio.

Los probadores se encargan de evaluar varios videojuegos de acuerdo a su género, para realizar la evaluación la dividen en indicadores como la narrativa, el sonido, el

gráfico, la jugabilidad, la inteligencia artificial y la innovación; una vez que evalúan todos los indicadores emiten una evaluación final.

2.3 Descripción de la Aplicación:

A continuación, se describirá la aplicación mediante la definición de los requisitos funcionales y los requisitos no funcionales que debe presentar.

2.3.1 Requisitos Funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones específicas (22). A continuación, se presentan la tabla de los requisitos funcionales de la aplicación. Para la elaboración de las misma se usó como referencia la Especificación de requisitos de software que se encuentra en el Expediente de proyectos de desarrollo V4.0.

No.	Requisito	Descripción	Prioridad	Complejidad
RF1	Insertar usuario	El sistema debe permitir al administrador insertar un nuevo usuario, solicitando los datos siguientes: (*) Estado. (*) Nombre. (*) Apellidos. (*) Usuario. (*) Contraseña. (*) Repetir Contraseña. () Anclar IP () E-mail	Alta	Alta
RF2	Eliminar usuario	El sistema debe permitir al administrador eliminar uno o varios usuarios.	Media	Baja
RF3	Modificar usuario	El sistema debe permitir al administrador editar los datos de un usuario.	Alta	Alta

Capítulo 2: Descripción de la Propuesta de Solución

		(*) Estado. (*) Nombre. (*) Apellidos. (*) Usuario. () Anclar IP () E-mail		
RF4	Visualizar usuario	El sistema debe permitir al administrador visualizar los datos de un usuario.	Media	Baja
RF5	Autenticar usuario.	Debe permitir a los usuarios registrados autenticarse en el sistema.	Alta	Media
RF6	Insertar rol	El sistema debe permitir al administrador insertar un nuevo rol, solicitando los datos siguientes: (*) Descripción. (*) Nombre.	Alta	Alta
RF7	Eliminar rol	El sistema debe permitir al administrador eliminar uno o varios roles.	Media	Baja
RF8	Modificar rol	El sistema debe permitir al administrador editar los datos de un rol. (*) Descripción. (*) Nombre.	Alta	Alta
RF9	Visualizar rol	El sistema debe permitir al administrador visualizar los datos de un rol.	Media	Baja
RF10	Asignar rol	El sistema debe permitir al administrador asignar varios roles a un usuario.	Alta	Media
RF11	Insertar permiso.	El sistema debe permitir al administrador insertar un nuevo	Alta	Alta

Capítulo 2: Descripción de la Propuesta de Solución

		permiso, solicitando los datos siguientes: (* Descripción. (* Nombre. (* Grupo.		
RF12	Eliminar permiso	El sistema debe permitir al administrador eliminar uno o varios permisos.	Media	Baja
RF13	Modificar permiso	El sistema debe permitir al administrador editar los datos de un permiso. (* Descripción. (* Nombre. (* Grupo.	Alta	Alta
RF14	Visualizar permiso	El sistema debe permitir al administrador visualizar los datos de un permiso.	Media	Baja
RF15	Asignar permiso	El sistema debe permitir al administrador asignar varios permisos a un rol.	Alta	Media
RF16	Insertar videojuego	El sistema debe permitir al administrador insertar un nuevo videojuego, solicitando los datos siguientes: (* Nombre. (* Género. (* Plataforma. (* Cantidad de jugadores. (* Empresa.	Alta	Alta
RF17	Eliminar videojuego	El sistema debe permitir al administrador eliminar uno o varios videojuegos.	Media	Baja

Capítulo 2: Descripción de la Propuesta de Solución

RF18	Modificar videojuego	El sistema debe permitir al administrador editar los datos de un videojuego. (*) Nombre. (*) Género. (*) Plataforma. (*) Cantidad de jugadores. (*) Empresa.	Alta	Alta
RF19	Visualizar videojuego	El sistema debe permitir al administrador visualizar los datos de un videojuego.	Media	Baja
RF20	Evaluar videojuego	El sistema debe permitir al probador evaluar un videojuego, introduciendo los campos: (*) Resumen. (*) Duración Aproximada.	Alta	Alta
RF21	Evaluar la narrativa de un videojuego	El sistema debe permitir al probador insertar las evaluaciones de los aspectos de narrativa que son los siguientes: (*) Historia. (*) Misiones. (*) Diálogos.	Alta	Media
RF22	Evaluar el gráfico del videojuego	El sistema debe permitir al probador insertar las evaluaciones de los aspectos de gráfico que son los siguientes: (*) Animaciones. (*) Texturas. (*) Efectos. (*) Escenario.	Alta	Media
RF23	Evaluar el sonido del videojuego	El sistema debe permitir al probador insertar las evaluaciones	Alta	Media

Capítulo 2: Descripción de la Propuesta de Solución

		de los aspectos de sonido que son los siguientes: (*) Música. (*) Efectos. () Voces.		
RF24	Evaluar la jugabilidad del videojuego	El sistema debe permitir al probador insertar las evaluaciones de los aspectos de jugabilidad que son los siguientes: (*) Control. (*) Cámara. (*) Mecánicas. (*) Curva de Aprendizaje.	Alta	Media
RF25	Evaluar la inteligencia artificial del videojuego	El sistema debe permitir al probador insertar las evaluaciones de los aspectos de inteligencia artificial que son los siguientes: (*) Comportamiento. (*) Aprendizaje. (*) Dificultad. (*) Adaptabilidad.	Alta	Media
RF26	Evaluar la innovación del videojuego	El sistema debe permitir al probador insertar las evaluaciones de los aspectos de innovación que son los siguientes: (*) Nuevas Mecánicas. (*) Mezcla de géneros.	Alta	Media
RF27	Añadir aspectos positivos y negativos	El sistema debe permitir al probador insertar los aspectos positivos y negativos del videojuego llenando los siguientes datos: (*) Descripción. (*) Tipo.	Baja	Baja

Capítulo 2: Descripción de la Propuesta de Solución

RF28	Visualizar evaluación.	El sistema debe permitir a los usuarios del sistema visualizar los datos de una evaluación.	Media	Media
RF29	Visualizar evaluación de narrativa.	El sistema debe permitir a los usuarios del sistema visualizar los datos de una evaluación del indicador de narrativa.	Media	Baja
RF30	Visualizar evaluación de gráfico	El sistema debe permitir a los usuarios del sistema visualizar los datos de una evaluación del indicador de gráfico.	Media	Baja
RF31	Visualizar evaluación de sonido	El sistema debe permitir a los usuarios del sistema visualizar los datos de una evaluación del indicador de sonido.	Media	Baja
RF32	Visualizar evaluación de jugabilidad	El sistema debe permitir a los usuarios del sistema visualizar los datos de una evaluación del indicador de jugabilidad.	Media	Baja
RF33	Visualizar evaluación de inteligencia artificial	El sistema debe permitir a los usuarios del sistema visualizar los datos de una evaluación del indicador de inteligencia artificial.	Media	Baja
RF34	Visualizar evaluación de innovación	El sistema debe permitir a los usuarios del sistema visualizar los datos de una evaluación del indicador de innovación	Media	Baja
RF35	Eliminar evaluación	El sistema debe permitir al administrador eliminar una o varias evaluaciones.	Media	Baja
RF36	Modificar evaluación	El sistema debe permitir al probador editar los datos de una evaluación.	Alta	Alta

Capítulo 2: Descripción de la Propuesta de Solución

		(*) Resumen. (*) Duración Aproximada		
RF37	Listar videojuegos	El sistema debe permitir a los usuarios del sistema listar todos los videojuegos.	Media	Media
RF38	Listar géneros	El sistema debe permitir a los usuarios del sistema listar todos los géneros.	Media	Media
RF39	Listar plataformas	El sistema debe permitir a los usuarios del sistema listar todas las plataformas.	Media	Media
RF40	Listar usuarios	El sistema debe permitir a los usuarios del sistema listar todos los usuarios.	Media	Media
RF41	Listar roles	El sistema debe permitir a los usuarios del sistema listar todos los roles.	Media	Media
RF42	Listar permisos	El sistema debe permitir a los usuarios del sistema listar todos los permisos.	Media	Media
RF43	Listar evaluaciones	El sistema debe permitir a los usuarios del sistema listar todas las evaluaciones.	Media	Media
RF44	Listar evaluaciones del indicador narrativa	El sistema debe permitir a los usuarios del sistema listar todas las evaluaciones del indicador de narrativa.	Media	Media
RF45	Listar evaluaciones del indicador gráficos	El sistema debe permitir a los usuarios del sistema listar todas las evaluaciones del indicador de gráficos.	Media	Media
RF46	Listar evaluaciones	El sistema debe permitir a los usuarios del sistema listar todas	Media	Media

	del indicador sonido	las evaluaciones del indicador de sonido.		
RF47	Listar evaluaciones del indicador jugabilidad	El sistema debe permitir a los usuarios del sistema listar todas las evaluaciones del indicador de jugabilidad.	Media	Media
RF48	Listar evaluaciones del indicador inteligencia artificial	El sistema debe permitir a los usuarios del sistema listar todas las evaluaciones del indicador de inteligencia artificial.	Media	Media
RF49	Lista evaluaciones del indicador innovación	El sistema debe permitir a los usuarios del sistema listar todas las evaluaciones del indicador de innovación	Media	Media
RF50	Exportar datos	El sistema debe permitir a los usuarios del sistema exportar los datos de las tablas videojuegos y evaluaciones en formato PDF, Excel, CSV (Valores separados por coma), Texto, JSON.	Media	Media

Tabla 2. Descripción de los requisitos funcionales.

2.3.2 Requisitos no funcionales

Los requisitos no funcionales son aquellos que no se refieren a las funcionalidades que proporciona el sistema, sino a las propiedades como restricciones del entorno o de la implementación, rendimiento, dependencias y soportes entre otras.

Requisitos de Software y Hardware

RnF1 Software:

El servidor debe contar con un:

- Servidor de Bases de Datos Relacional MySql

Capítulo 2: Descripción de la Propuesta de Solución

- Servidor de Aplicaciones web

El cliente debe tener instalado un navegador, puede ser:

- Navegador Mozilla Firefox superior a 40.x o superior.
- Internet Explorer 11 o superior.

RnF2 Hardware:

Servidor:

- RAM: 2GB o superior
- Disco Duro: 5 GB o superior.
- Microprocesador: AMD A6-3400 1.40GHz o superior.
- Dispositivo de red: 100 Mbits o superior.

Cliente:

- RAM: 512 MB o superior.
- Microprocesador: AMD A6-3400 1.40GHz o superior.
- Dispositivo de red: 10 Mbits o superior.

Restricciones de diseño e implementación

RnF3: El lenguaje de programación utilizado es PHP 7.

RnF4: El Entorno de Desarrollo Integrado utilizado es JetBrains PhpStorm 2016.3.2.

RnF5: El framework de desarrollo utilizado es Yii 2.

RnF6: El estilo arquitectónico utilizado es Modelo Vista Controlador.

RnF7: La biblioteca JavaScript utilizada es jQuery v1.10.

RnF8: El framework CSS utilizado para el diseño de las interfaces gráficas es Bootstrap v 3.0.

Seguridad y Confiabilidad:

Capítulo 2: Descripción de la Propuesta de Solución

RnF9: Restringir el acceso a usuarios no autorizados para que no se pueda realizar cualquier acción sobre el contenido de la plataforma.

RnF10: Seguridad de acceso y administración de usuarios: otorgamiento de privilegios y roles, asignación de perfiles. Los niveles de acceso están determinados por los diferentes roles válidos dentro de la plataforma.

RnF11: Garantizar que la información sea editada únicamente por quien está autorizado y posea permisos para ello.

RnF12: Verificación sobre acciones irreversibles (mensajes de eliminaciones).

Usabilidad:

RnF13: El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

RnF14: El sistema debe tener acceso al menú general desde cualquiera de sus páginas.

RnF15: Se deben mostrar las rutas de acceso según la navegación que tenga el usuario.

RnF16: Las rutas de acceso deben tener vínculos a las secciones que muestran.

RnF17: Los elementos gráficos como los iconos deberán contar con un mensaje flotante que señalen el tipo de recurso al que se refiere.

Rendimiento:

RnF18: Tiempos de respuestas no excede los 5 segundos para cada petición realizada.

Disponibilidad:

RnF19: Los usuarios del sistema deben tener acceso (según sus permisos) en todo momento a la información solicitada.

2.4 Identificación de los actores y casos de uso del sistema

Con el objetivo de especificar privilegios, nivel de acceso a la información y funciones en la aplicación, se identifican los siguientes actores:

Actores	Descripción
Administrador	Persona registrada en el sistema que cuenta con todos los privilegios.
Probador	Persona registrada en el sistema que posee los privilegios para visualizar los videojuegos y evaluarlos.
Usuarios del sistema	Persona registrada en el sistema que posee los privilegios para visualizar y listar los videojuegos, evaluaciones, indicadores y exportar datos.

Tabla 3. Descripción de los actores del sistema

2.4.1 Casos de uso del sistema

Un caso de uso especifica una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo y que producen un resultado observable de valor para un actor concreto.

2.4.2 Diagrama de casos de uso del sistema

El diagrama de casos de uso del sistema representa gráficamente los procesos y su interacción con los actores del sistema. En la siguiente figura se muestra el diagrama de CU correspondiente a la propuesta de solución.

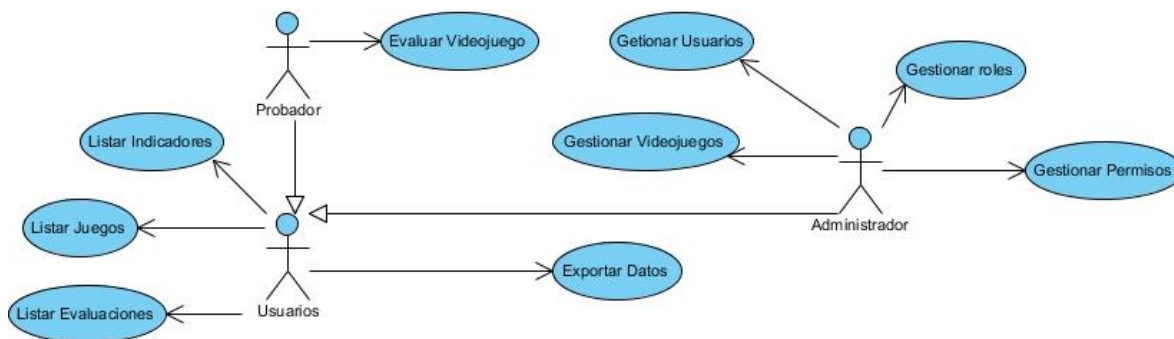


Figura 4. Diagrama de CU del sistema

2.4.3 Descripciones de los CU

En esta sección se presenta la descripción del CU más significativo de la aplicación, las restantes se podrán encontrar en el **Anexo I**.

Capítulo 2: Descripción de la Propuesta de Solución

Objetivo	Evaluar un videojuego seleccionado de acuerdo a los resultados de sus indicadores.	
Actores	Probador(inicia)	
Resumen	El caso de uso inicia cuando el probador selecciona la opción Evaluar juego en lista de juegos y finaliza una vez que se evalúen todos los indicadores	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	El probador debe estar autenticado y debe haber al menos un juego creado.	
Postcondiciones	Las evaluaciones son almacenadas en la base de datos	
Flujo de Eventos		
Flujo básico <Evaluar Videojuego>		
1.	Selecciona de la página principal la opción "Juego" y la funcionalidad "Juegos"	<p>1.1 Muestra una interfaz con la lista de usuarios del sistema que contiene los campos:</p> <ul style="list-style-type: none"> • Juego ID: muestra el ID del videojuego • Nombre: muestra el nombre del videojuego. • Género: muestra el género del videojuego. • Plataforma: muestra la plataforma del videojuego. • Cantidad Jugadores: muestra la cantidad de jugadores que tiene el videojuego. • Evaluación: muestra la evaluación final del videojuego. <p>Además, el siguiente botón:</p> <ul style="list-style-type: none"> • Evaluar: botón que permite al probador comenzar el proceso de evaluación.
2.	Selecciona el juego y la opción Evaluar	<p>2.1 El sistema muestra en un orden secuencial los indicadores a evaluar:</p> <p>Narrativa. Ver Sección 1: "Evaluar Narrativa".</p> <p>Gráficos. Ver Sección 2: "Evaluar Gráficos".</p> <p>Sonido. Ver Sección 3: "Evaluar Sonido".</p> <p>Jugabilidad. Ver Sección 4: "Evaluar Jugabilidad".</p>

Capítulo 2: Descripción de la Propuesta de Solución

		<p>Inteligencia Artificial. Ver Sección 5: “Evaluar Inteligencia Artificial”.</p> <p>Innovación. Ver Sección 6: “Evaluar Innovación”.</p> <p>Aspectos. Ver Sección 7: “Evaluar Aspectos”.</p> <p>Evaluación Final. Ver Sección 8: “Finalizar Evaluación”.</p>
Sección 1: “Evaluar Narrativa”		
Flujo Básico <Evaluar Narrativa >		
	Actor	Sistema
1.		<p>1.1 Verifica que el aspecto de narrativa no esté evaluado.</p> <p>1.2 Muestra una interfaz para evaluar el indicador de narrativa, la misma contiene los siguientes campos:</p> <ul style="list-style-type: none"> • Historia: campo de tipo texto que permite introducir números del 1 al 10. • Misiones: campo de tipo texto que permite introducir números del 1 al 10. • Diálogos: campo de tipo texto que permite introducir números del 1 al 10. <p>Además, los siguientes botones:</p> <ul style="list-style-type: none"> • Siguiente: Botón que almacena las evaluaciones del indicador de narrativa en la base de datos y pasa a la siguiente interfaz. • Omitir: Botón que pasa a la siguiente evaluación. Este botón solo se muestra si el género del videojuego es acción, estrategia o simulación.
2.	Introduce los datos y selecciona la opción Siguiente.	<p>2.1 Verifica que los campos estén llenos.</p> <p>2.2 Verifica si los datos están en el rango establecido.</p> <p>2.3 Verifica si el formato de los datos es válido.</p> <p>2.4 Guarda la evaluación del indicador de narrativa en el sistema.</p> <p>2.5 Va al paso 1.1 de la Sección 2: “Evaluar Gráfico”.</p>
Flujos Alternos		
1.1 <Narrativa evaluada>		
	Actor	Sistema

Capítulo 2: Descripción de la Propuesta de Solución

1.		1.1 Va al paso 1.1 de la Sección 2: “Evaluar Gráfico”.
2 <Botón Omitir>		
	Actor	Sistema
2.	Selecciona el botón Omitir	2.1 Va al paso 1.1 de la Sección 2: “Evaluar Gráfico”.
2.1 <Existen campos vacíos>		
	Actor	Sistema
3.		3.1 Resalta de color rojo los campos vacíos y muestra el mensaje: “No puede estar vacío”.
2.2 <Rango Inválido>		
	Actor	Sistema
4.		4.1 Resalta de color rojo los campos que se pasan del rango establecido y muestra el mensaje: “No debe ser menor a 1” o “No debe ser mayor a 10”.
2.3 <Formato Inválido>		
	Actor	Sistema
5.		5.1 Resalta de color rojo los campos que tienen un formato no válido y muestra el mensaje: “Debe ser un número entero”.
Sección 2: “Evaluar Gráfico”		
Flujo Básico <Evaluar Gráfico >		
1.		<p>1.1 Verifica que el aspecto de gráfico no esté evaluado.</p> <p>1.2 Muestra una interfaz para el indicador de gráfico, el mismo contiene los siguientes campos:</p> <ul style="list-style-type: none"> • Animaciones: campo de tipo texto que permite introducir números del 1 al 10. • Textura: campo de tipo texto que permite introducir números del 1 al 10. • Efectos: campo de tipo texto que permite introducir números del 1 al 10. • Escenarios: campo de tipo texto que permite introducir números del 1 al 10. <p>Además, el siguiente botón:</p>

Capítulo 2: Descripción de la Propuesta de Solución

		<ul style="list-style-type: none"> Siguiente: Botón que almacena las evaluaciones del indicador de gráfico en la base de datos y pasa a la siguiente interfaz.
2.	Introduce los datos y selecciona la opción Siguiente.	2.1 Verifica que los campos estén llenos. 2.2 Verifica si los datos están en el rango establecido. 2.3 Verifica si el formato de los datos es válido. 2.4 Guarda la evaluación del indicador de gráfico en el sistema. 2.5 Va al paso 1.1 de la Sección 3: "Evaluar Sonido".
Flujos Alternos		
1.1 <Gráfico evaluado>		
	Actor	Sistema
1.		1.1 Va al paso 1.1 de la Sección 3: "Evaluar Sonido".
2.1 <Existen campos vacíos>		
	Actor	Sistema
2.		2.1 Resalta de color rojo los campos vacíos y muestra el mensaje: "No puede estar vacío".
2.2 <Rango Inválido>		
	Actor	Sistema
3.		3.1 Resalta de color rojo los campos que se pasan del rango establecido y muestra el mensaje: "No debe ser menor a 1" o "No debe ser mayor a 10".
2.3 <Formato Inválido>		
	Actor	Sistema
4.		4.1 Resalta de color rojo los campos que tienen un formato no válido y muestra el mensaje: "Debe ser un número entero".
Sección 3: "Evaluar Sonido"		
Flujo Básico <Evaluar Sonido >		
1.		1.1 Verifica que el aspecto de sonido no esté evaluado. 1.2 Muestra una interfaz para el indicador de sonido, el mismo contiene los siguientes campos:

Capítulo 2: Descripción de la Propuesta de Solución

		<ul style="list-style-type: none"> • Música: campo de tipo texto que permite introducir números del 1 al 10. • Efectos: campo de tipo texto que permite introducir números del 1 al 10. • Voces: campo de tipo texto que permite introducir números del 1 al 10. <p>Además, el siguiente botón:</p> <ul style="list-style-type: none"> • Siguiente: Botón que almacena las evaluaciones del indicador de sonido en la base de datos y pasa a la siguiente interfaz.
2.	Introduce los datos y selecciona la opción Siguiente.	<p>2.1 Verifica que los campos estén llenos.</p> <p>2.2 Verifica si los datos están en el rango establecido.</p> <p>2.3 Verifica si el formato de los datos es válido.</p> <p>2.4 Guarda la evaluación del indicador de gráfico en el sistema.</p> <p>2.5 Va al paso 1.1 de la Sección 4: "Evaluar Jugabilidad".</p>
Flujos Alternos		
1.1 <Sonido evaluado>		
	Actor	Sistema
1.		1.1 Va al paso 1.1 de la Sección 4: "Evaluar Jugabilidad".
2.1 <Existen campos vacíos>		
	Actor	Sistema
2.		2.1 Resalta de color rojo los campos vacíos y muestra el mensaje: "No puede estar vacío".
2.2 <Rango Inválido>		
	Actor	Sistema
3.		3.1 Resalta de color rojo los campos que se pasan del rango establecido y muestra el mensaje: "No debe ser menor a 1" o "No debe ser mayor a 10".
2.3 <Formato Inválido>		
	Actor	Sistema

Capítulo 2: Descripción de la Propuesta de Solución

4.		4.1 Resalta de color rojo los campos que tienen un formato no válido y muestra el mensaje: “Debe ser un número entero”.
Sección 4: “Evaluar Jugabilidad”		
Flujo Básico <Evaluar Jugabilidad >		
1.		<p>1.1 Verifica que el aspecto de jugabilidad no esté evaluado.</p> <p>1.2 Muestra una interfaz para el indicador de jugabilidad, el mismo contiene los siguientes campos:</p> <ul style="list-style-type: none"> • Control: campo de tipo texto que permite introducir números del 1 al 10. • Cámara: campo de tipo texto que permite introducir números del 1 al 10. • Mecánicas: campo de tipo texto que permite introducir números del 1 al 10. • Curva de aprendizaje: campo de tipo texto que permite introducir números del 1 al 10. <p>Además, el siguiente botón:</p> <ul style="list-style-type: none"> • Siguiente: Botón que almacena las evaluaciones del indicador de jugabilidad en la base de datos y pasa a la siguiente interfaz.
2.	Introduce los datos y selecciona la opción Siguiente.	<p>2.1 Verifica que los campos estén llenos.</p> <p>2.2 Verifica si los datos están en el rango establecido.</p> <p>2.3 Verifica si el formato de los datos es válido.</p> <p>2.4 Guarda la evaluación del indicador de gráfico en el sistema.</p> <p>2.5 Va al paso 1.1 de la Sección 5: “Evaluar Inteligencia Artificial”.</p>
Flujos Alternos		
1.1 <Jugabilidad evaluada>		
	Actor	Sistema
1.		1.1 Va al paso 1.1 de la Sección 5: “Evaluar Inteligencia Artificial”.
2.1 <Existen campos vacíos>		

Capítulo 2: Descripción de la Propuesta de Solución

	Actor	Sistema
2.		2.1 Resalta de color rojo los campos vacíos y muestra el mensaje: “No puede estar vacío”.
2.2 <Rango Inválido>		
	Actor	Sistema
3.		3.1 Resalta de color rojo los campos que se pasan del rango establecido y muestra el mensaje: “No debe ser menor a 1” o “No debe ser mayor a 10”.
2.3 <Formato Inválido>		
	Actor	Sistema
4.		4.1 Resalta de color rojo los campos que tienen un formato no válido y muestra el mensaje: “Debe ser un número entero”,
Sección 5: “Evaluar Inteligencia Artificial”		
Flujo Básico <Evaluar Inteligencia Artificial >		
1.		<p>1.1 Verifica que el aspecto de jugabilidad no esté evaluado.</p> <p>1.2 Muestra una interfaz para el indicador de inteligencia artificial, el mismo contiene los siguientes campos:</p> <ul style="list-style-type: none"> • Comportamiento: campo de tipo texto que permite introducir números del 1 al 10. • Aprendizaje: campo de tipo texto que permite introducir números del 1 al 10. • Dificultad: campo de tipo texto que permite introducir números del 1 al 10. • Adaptabilidad: campo de tipo texto que permite introducir números del 1 al 10. <p>Además, los siguientes botones:</p> <ul style="list-style-type: none"> • Siguiente: Botón que almacena las evaluaciones del indicador de inteligencia artificial en la base de datos y pasa a la siguiente interfaz. • Omitir: Botón que pasa a la siguiente evaluación. Este botón solo se muestra si el género del videojuego es acción, aventura o rol.

Capítulo 2: Descripción de la Propuesta de Solución

2.	Introduce los datos y selecciona la opción Siguiente.	<p>2.1 Verifica que los campos estén llenos.</p> <p>2.2 Verifica si los datos están en el rango establecido.</p> <p>2.3 Verifica si el formato de los datos es válido.</p> <p>2.4 Guarda la evaluación del indicador de gráfico en el sistema.</p> <p>2.5 Va al paso 1.1 de la Sección 6: “Evaluar Innovación”.</p>
Flujos Alternos		
1.1 <Inteligencia Artificial evaluada>		
	Actor	Sistema
1.		1.1 Va al paso 1.1 de la Sección 6: “Evaluar Innovación”.
2 <Botón Omitir>		
	Actor	Sistema
2.	Selecciona el botón Omitir	2.1 Va al paso 1.1 de la Sección 6: “Evaluar Innovación”.
2.1 <Existen campos vacíos>		
	Actor	Sistema
3.		3.1 Resalta de color rojo los campos vacíos y muestra el mensaje: “No puede estar vacío”.
2.2 <Rango Inválido>		
	Actor	Sistema
4.		4.1 Resalta de color rojo los campos que se pasan del rango establecido y muestra el mensaje: “No debe ser menor a 1” o “No debe ser mayor a 10”.
2.3 <Formato Inválido>		
	Actor	Sistema
5.		5.1 Resalta de color rojo los campos que tienen un formato no válido y muestra el mensaje: “Debe ser un número entero”.
Sección 6: “Evaluar Innovación”		
Flujo Básico <Evaluar Innovación >		
1.		1.1 Verifica que el aspecto de innovación no esté evaluado.

Capítulo 2: Descripción de la Propuesta de Solución

		<p>1.2 Muestra una interfaz para el indicador de innovación, el mismo contiene los siguientes campos:</p> <ul style="list-style-type: none"> • Mecánicas nuevas: campo de tipo texto que permite introducir números del 1 al 10. • Mezcla de géneros: campo de tipo texto que permite introducir números del 1 al 10. • Mezcla de roles: campo de tipo texto que permite introducir números del 1 al 10. <p>Además, el siguiente botón:</p> <ul style="list-style-type: none"> • Siguiente: Botón que almacena las evaluaciones del indicador de innovación en la base de datos y pasa a la siguiente interfaz.
2.	Introduce los datos y selecciona la opción Siguiente.	<p>2.1 Verifica que los campos estén llenos.</p> <p>2.2 Verifica si los datos están en el rango establecido.</p> <p>2.3 Verifica si el formato de los datos es válido.</p> <p>2.4 Guarda la evaluación del indicador de gráfico en el sistema.</p> <p>2.5 Va al paso 1.1 de la Sección 7: "Evaluar Aspectos".</p>
Flujos Alternos		
1.1 <Innovación evaluada>		
	Actor	Sistema
1.		1.1 Va al paso 1.1 de la Sección 7: "Evaluar Aspectos".
2.1 <Existen campos vacíos>		
	Actor	Sistema
2.		2.1 Resalta de color rojo los campos vacíos y muestra el mensaje: "No puede estar vacío".
2.2 <Rango Inválido>		
	Actor	Sistema
3.		3.1 Resalta de color rojo los campos que se pasan del rango establecido y muestra el mensaje: "No debe ser menor a 1" o "No debe ser mayor a 10".
2.3 <Formato Inválido>		
	Actor	Sistema

Capítulo 2: Descripción de la Propuesta de Solución

4.		4.1 Resalta de color rojo los campos que tienen un formato no válido y muestra el mensaje: “Debe ser un número entero”.
Sección 7: “Evaluar Aspectos”		
Flujo Básico <Evaluar Aspectos >		
1.		<p>1.1 Muestra una interfaz para los aspectos positivos y negativos, el mismo contiene los siguientes campos:</p> <ul style="list-style-type: none"> • Descripción: es un campo de tipo área de texto para introducir la descripción del aspecto. • Tipo: cuadro de selección que permite elegir si el aspecto es positivo o negativo. <p>Además, los siguientes botones:</p> <ul style="list-style-type: none"> • Agregar: Botón que almacena el aspecto en la base de datos. • Siguiente: Botón que pasa a la siguiente interfaz.
2.	Introduce los datos y selecciona la opción Siguiente.	<p>2.1 Verifica que los campos estén llenos.</p> <p>2.2 Guarda los aspectos positivos y negativos en el sistema.</p> <p>2.3 Va al paso 1.1 de la Sección 8: “Finalizar Evaluación”.</p>
Flujos Alternos		
2.1 <Existen campos vacíos>		
	Actor	Sistema
1.		1.1 Resalta de color rojo los campos vacíos y muestra el mensaje: “No puede estar vacío”.
Sección 8: “Finalizar Evaluación”.		
Flujo Básico < Finalizar Evaluación>		
1.		<p>1.1 Muestra una interfaz para finalizar la evaluación, el mismo contiene los siguientes campos:</p> <ul style="list-style-type: none"> • Resumen: es un campo de tipo área de texto para introducir el resumen de la evaluación del videojuego. • Duración aproximada: campo de tipo texto que permite introducir números enteros. <p>Además, el siguiente botón:</p>

		<ul style="list-style-type: none"> Finalizar: Botón que almacena la evaluación final del videojuego. 	
2.	Introduce los datos y selecciona la opción Finalizar.	2.1 Valida los campos. 2.2 Almacena la evaluación en la base de datos. 2.3 Muestra el CU: Listar Evaluación. Sección 1: Ver Evaluación.	
Flujos Alternos			
2.1 b “Existen campos vacíos”			
	Actor	Sistema	
1.		1.1 Resalta de color rojo los campos vacíos y muestra el mensaje: “No puede estar vacío”.	
Relaciones		CU incluidos	No aplica
		CU extendidos	No aplica
Requisitos no funcionales		No aplica	
Asuntos pendientes.		No aplica	

Tabla 4. CU6: Evaluar videojuego.

2.5 Patrones arquitectónicos

Un patrón arquitectónico es una colección de decisiones de diseño arquitectónicas que tiene un nombre específico y que son aplicables a problemas de diseño recurrentes, y son parametrizadas para tener en cuenta diferentes contextos de desarrollo de software en los cuales el problema aparece (23).

2.5.1 Modelo Vista Controlador

El patrón Modelo Vista Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. EL patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en reducido espacio de tiempo (24).

Este patrón de arquitectura presenta varias ventajas:

- Separación clara entre los componentes de un programa; lo cual permite su implementación por separado.
- Interfaz de Programación de Aplicaciones (API) muy bien definida; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- Conexión entre el Modelo y sus Vistas dinámica; se produce en tiempo de ejecución, no en tiempo de compilación (24).

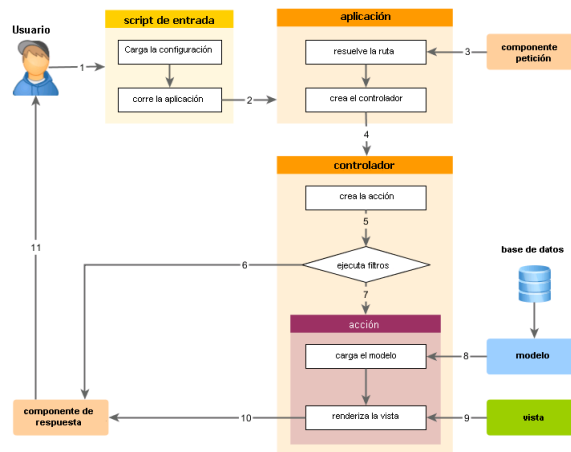


Figura 5. Patrón arquitectónico MVC para Framework Yii.

2.6 Modelo del diseño

El modelo del diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en los requisitos funcionales y no funcionales, junto a otras restricciones relacionadas con el entorno de implementación (20).

2.6.1 Patrones de diseño utilizados

Los patrones de diseño representan soluciones a problemas que surgen cuando se desarrolla un software en un contexto particular. Para realizar el diseño de la aplicación se han utilizado algunos de los patrones existentes, los cuales se mencionan a continuación:

- **Patrones GRASP**

Los Patrones Generales de Software para Asignar Responsabilidades (GRASP, por sus siglas en inglés). Describen los principios fundamentales de la asignación de

responsabilidades a objetos, expresados en forma de patrones. Los patrones GRASP utilizados en el desarrollo de la aplicación son:

Experto: permite asignar una responsabilidad al experto en información, a la clase que cuenta con la información necesaria para cumplir la responsabilidad (25). En Yii toda la información de los datos la manejan las clases modelos, pues son estas las que acceden a la base de datos en respuesta de una petición de las clases controladoras para satisfacer un evento iniciado por el usuario.

Creador: el propósito fundamental de este patrón consiste en guiar la asignación de responsabilidades relacionada con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. Su utilización se evidencia mediante las clases controladoras las cuales crean instancias de las entidades del modelo.

```
public function actionCreate()
{
    $model = new Juego();
    if ($model->load(Yii::$app->request->post()) && $model->save()) {
        return $this->redirect(['view', 'id' => $model->juego_id]);
    } else {
        $generos = Genero::getMapadeGeneros();
        $plataformas = Plataforma::getMapadePlataforma();
        return $this->render('create', [
            'model' => $model,
            'generos' => $generos,
            'plataformas' => $plataformas
        ]);
    }
}
```

Figura 6. Ejemplo de utilización del patrón Creador

Controlador: permite asignar la responsabilidad de manejar los eventos del sistema a una clase. Un evento de entrada al sistema es algún evento desde algún actor externo. La clase controlador es responsable de decidir qué hacer con el evento. El controlador actúa como una fachada entre las interfaces y la aplicación. Yii tiene bien definidas las clases controladoras dentro de su patrón MVC, las cuales se encargan de delegar responsabilidades, en respuesta a alguna acción del usuario, a las clases modelo e interfaces.

Bajo Acoplamiento: este patrón consiste en asignar una responsabilidad para mantener el bajo acoplamiento. El acoplamiento es una medida de la fuerza con que una clase está conectada con otras clases, de cómo se conocen y recurren a ellas (25). El propósito de este patrón es que exista entre las clases la menor dependencia posible, para en caso de producirse una modificación en alguna de

ellas, no afecte las restantes. Se evidencia mediante las clases que implementan la lógica de acceso a datos que se encuentran en el modelo, las cuales no presentan asociaciones con las clases de las vistas o las controladoras.

Alta Cohesión: la solución que propone este patrón consiste en asignar una responsabilidad de modo que la cohesión siga siendo alta. La cohesión es una medida de cuanto están relacionadas y enfocadas las responsabilidades de una clase (25). Se evidencia mediante la implementación de la clase controladora **EvaluacionController.php**, la cual está formada por diferentes funcionalidades en la que se encuentra el método **actionIndex ()**, en donde se realiza búsqueda de datos sobre la entidad **EvaluacionSearch** y el resultado se muestra en la vista **index.html**.

```
public function actionIndex()
{
    $searchModel = new EvaluacionSearch();
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}
```

Figura 7. Ejemplo de utilización del patrón Alta Cohesión

- **Patrones GOF**

Los patrones GOF favorecen la reutilización de código y ayudan a construir un *software* basado en la reutilización (24). A continuación, se describen los utilizados durante el desarrollo de la aplicación para la evaluación de videojuegos.

Singleton (instancia única): garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Restringe la instanciación de una clase o valor de un tipo a un solo objeto (24). Se evidencia en la clase `index.php` cuando se ejecuta la sentencia `Yii::createWebApplication($configFile)->run()`; esta sentencia crea una instancia de la aplicación. Después de creada se puede acceder a ella desde cualquier clase de la aplicación con `Yii::app()`.

- **ActiveRecord**

Para una mejor comprensión de los diagramas, se describe el DCD asociado al CU: Evaluar Videojuego. En el mismo se usan estereotipos para las clases basándose en la estructura del patrón Modelo Vista Controlador. Las clases Vista agrupa las interfaces asociadas al usuario más conocidas como <<*ClientPage*>> (Página Cliente), estas a su vez están compuestas por un formulario que se describen como <<*Form*>>, este es el que permite al usuario entrar datos. Las clases Controladoras se definen como <<*ServerPage*>> (Página Servidora) la cual mediante los métodos implementados accede a las clases del Modelo para dar respuestas a las peticiones del usuario, estas se definen por CE (Clase Entidad). Las clases controladoras heredan de la clase *Controller* y las vistas dependen del CSS para el estilo de las páginas y de *validacion.js* para validar el formato de los datos. La clase *ActiveRecord* es utilizada para la comunicación de las clases del modelo con las entidades de la base de datos.

2.7 Modelo de datos

El modelo de datos constituye la representación de un fenómeno de la realidad objetiva a través de los objetos, sus propiedades y las relaciones que se establecen entre ellos. El mismo está compuesto por tres piezas fundamentales: el objeto de datos, los atributos y las relaciones entre las que se conectan (41).

Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, se describe en el siguiente modelo de datos la representación lógica de los datos del sistema.

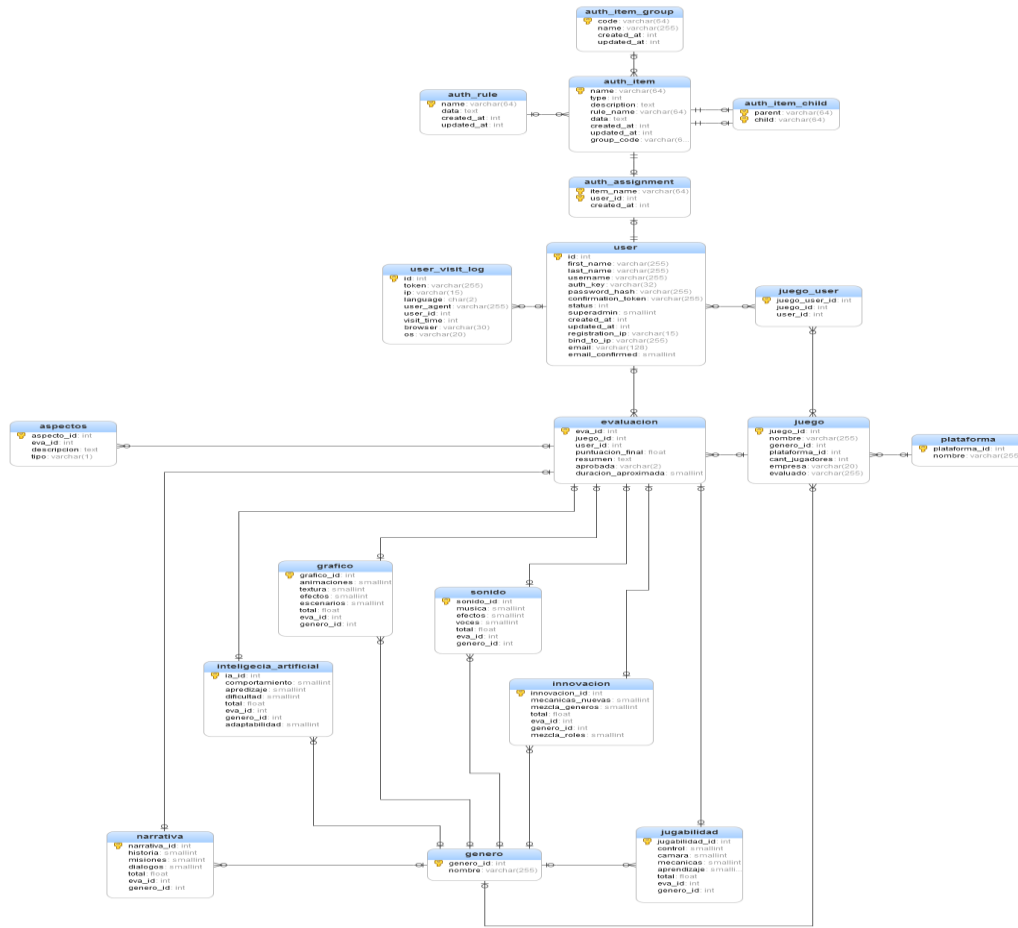


Figura 9. Modelo de datos

2.8 Consideraciones parciales

La identificación de los requisitos tanto funcionales como no funcionales, posibilitó elaborar un modelo del sistema que sirvió de acuerdo entre los desarrolladores y el cliente, constituyendo además la entrada fundamental para las fases posteriores del desarrollo. Las descripciones de los CU posibilitaron conocer el funcionamiento general del sistema, estableciendo para ellos un conjunto de pasos lógicos, flujos de información y respuestas del sistema al usuario, logrando obtener una comprensión general del negocio. El modelo del diseño obtenido permitió describir la realización física y dinámica de los casos de uso y crear de esta manera, una entrada apropiada para las actividades de implementación.

Capítulo 3: Implementación y pruebas

Una vez realizado el flujo de trabajo correspondiente al análisis y diseño, se tiene una visión más general de cómo se implementarán las funcionalidades del sistema. El propósito de este flujo permitirá describir de manera general la implementación de todas las clases y objetos asociados al sistema, teniendo como resultado un módulo ejecutable. Luego de concluida la etapa anterior se decide llevar a cabo las pruebas del sistema con el objetivo de verificar todo el proceso de desarrollo. En la etapa de implementación y pruebas se tienen en cuenta todos los elementos arquitectónicos definidos con anterioridad y permite concluir con éxito el proceso de desarrollo de la propuesta de solución.

3.1 Estándar de codificación

Los estándares de codificación son las reglas que comprenden todo lo relacionado al código fuente de cualquier software, por lo que resulta importante establecer su utilización para asegurar que durante el desarrollo del mismo cualquier programador pueda entender el código y trabajar de forma coordinada. En la implementación del sistema se utilizaron los siguientes estándares:

- Estilo *camelCase*, para nombrar funciones y métodos. Los nombres de las funciones y métodos son palabras compuestas, por lo que se comienzan a escribir con letra minúscula la primera palabra y a partir de la segunda palabra con letra mayúscula.
- Se utiliza el guión bajo (“_”), como separador de los nombres compuestos de las entidades en la base de datos.

3.2 Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo del diseño, como las clases, se implementan en términos de componentes, ficheros de código fuente y ejecutables. Además, define una jerarquía de subsistemas de implementación que contiene componentes e interfaces (27).

3.2.1 Diagrama de componentes

Los diagramas de componentes son usados para mostrar los componentes de software y las relaciones lógicas entre ellos en un sistema. Los estereotipos utilizados para representar dichos componentes son:

`<<file>>`: ficheros de configuración o código fuente.

El siguiente diagrama es la representación del sistema GameEva, dividido según propone el patrón arquitectónico utilizado MVC. El paquete *Views*, agrupa los subpaquetes que contienen los componentes asociados a las vistas HTML, las cuales importan los componentes de los paquetes CSS y JS. En el paquete *Controllers* se encuentran los componentes relacionados a las clases controladoras. Por último, en el paquete *Models* se encuentran las clases del modelo, que accede a la base de datos por el puerto 3306.

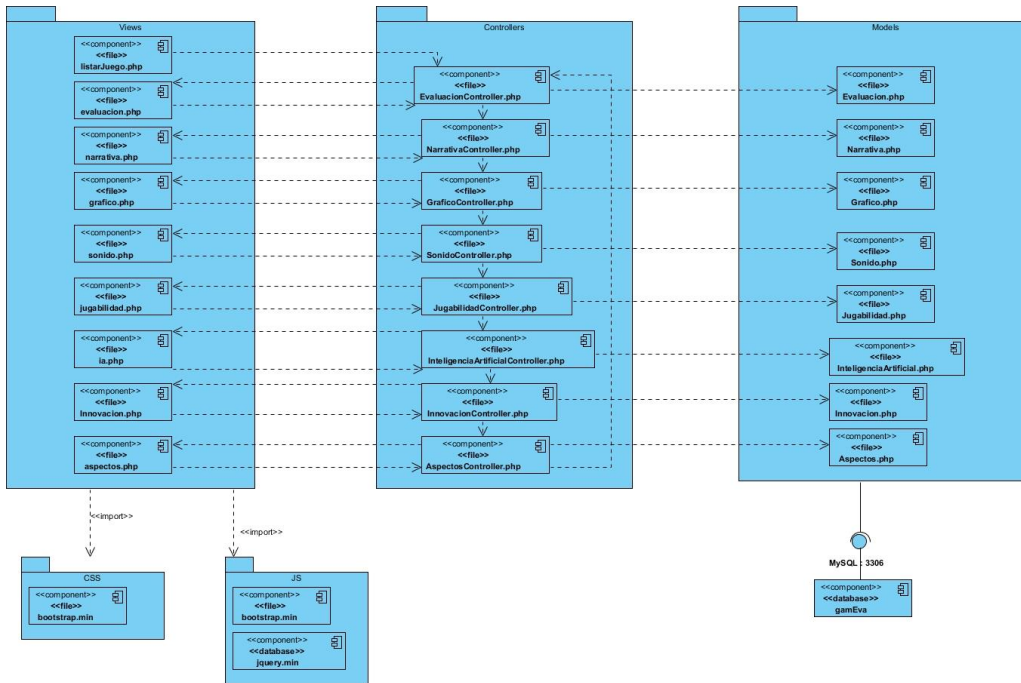


Figura 10. Diagrama de Componentes

3.2.2 Modelo de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. En el mismo se puede observar lo siguiente (27):

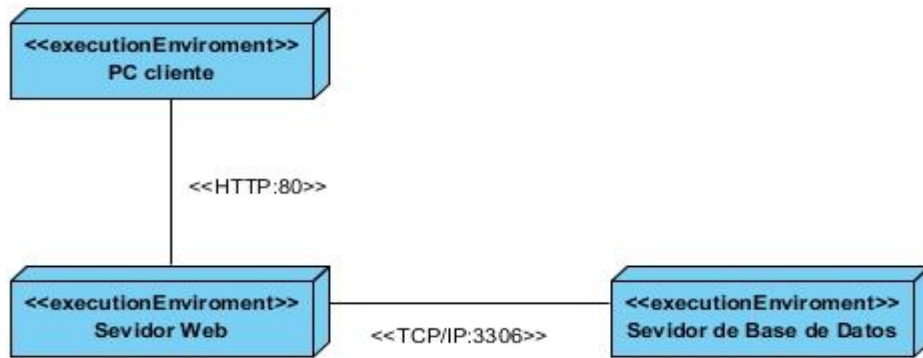


Figura 11. Diagrama de Despliegue

Descripción del diagrama de despliegue

- PC cliente: nodo que representa de la interacción de los usuarios con la aplicación a través de un navegador web conectado por el puerto 80.
- Servidor web: nodo que representa el servidor donde se encuentra la aplicación web.
- Servidor de base de datos: nodo que representa el servidor donde estará la base de datos de la aplicación a la que se accede por el puerto 3306.

3.3 Pruebas

Con el objetivo de controlar que los productos cumplan con los requisitos funcionales y determinar la calidad de los mismos, resulta importante realizar pruebas. Estas constituyen una actividad en la cual un sistema o unos de sus componentes se ejecutan en circunstancias previamente especificadas, donde los resultados se observan, se registran y se realiza una evaluación de algún aspecto del sistema. Dentro de los procedimientos para la realización de pruebas al software, se encuentran las pruebas de caja blanca y las pruebas de caja negra.

3.3.1 Pruebas de caja negra

Las pruebas de caja negra, también conocidas como pruebas de comportamiento, se concentran en los requisitos funcionales del software. Una prueba de este tipo examina algún aspecto funcional del sistema que tiene poca relación con la estructura lógica interna del software. Se encargan de encontrar errores en: las funcionalidades (incorrectas o faltantes), la interfaz, las estructuras de datos o el

Capítulo 3: Implementación y pruebas

acceso a base de datos externas, errores de comportamiento y errores de inicialización (28).

Existen varios métodos para realizar las pruebas de caja negra, entre los que se encuentran, según Pressman:

- Método de partición equivalente: divide el dominio de entradas de un programa en clases de datos, a partir de las cuales pueden derivarse casos de prueba.
- Método del análisis de valores límites: se encarga del análisis de los valores límites en los campos de entrada.
- Método de prueba de la tabla ortogonal: se aplica en problemas donde el dominio de entrada es relativamente pequeño, el método resulta útil para encontrar fallas de región (categoría de error asociado con los defectos de la lógica en un componente de software).

Para la verificación del correcto funcionamiento del sistema de evaluación de videojuegos se realizaron pruebas de caja negra, aplicando el método de partición equivalente. Partiendo de las descripciones de los 9 CU, se generaron los diseños de casos de pruebas (DCP) correspondientes, se confeccionó cada caso de prueba con la identificación de los principales escenarios en dependencia de cada acción del actor. A continuación, se muestra el DCP asociado al CU: Evaluar videojuego, la Sección 1 Evaluar narrativa. Los restantes se encuentran en los Anexos III.

Escenario	Descripción	Datos de Entrada			Respuesta del Sistema	Flujo central
		Historia	Misiones	Diálogos		
EC1.1 Selección a la opción Evaluar	Permite evaluar un videojuego.	N/A	N/A	N/A	1.1 Verifica que el aspecto de narrativa no esté evaluado. 1.2 Muestra una interfaz para evaluar el indicador de	Selecciona en la página principal de Juegos la opción

Capítulo 3: Implementación y pruebas

					<p>narrativa, la misma contiene los siguientes campos:</p> <ul style="list-style-type: none"> -Historia: campo de tipo texto que permite introducir números del 1 al 10. -Misiones: campo de tipo texto que permite introducir números del 1 al 10. -Diálogos: campo de tipo texto que permite introducir números del 1 al 10. <p>Además, los siguientes botones:</p> <ul style="list-style-type: none"> -Siguiente: Botón que almacena las evaluaciones del indicador de narrativa en la base de datos y pasa a la siguiente interfaz. -Omitir: Botón que pasa a la siguiente evaluación. Este botón solo se muestra si el género del videojuego es acción, estrategia o simulación. 	<p>Evaluar y redirecciona a la página de evaluar narrativa</p>
EC1.2	Si la narrativa ya está evaluada.	N/A	N/A	N/A	Va a la interfaz del siguiente indicador.	Redirecciona al usuario a la página de evaluar gráfico
EC1.3	Insertar los datos correspondientes a los campos:	10	9	8	<ul style="list-style-type: none"> -Verifica que los campos estén llenos. -Verifica si los datos están en el rango establecido. 	Se mantiene en la página de evaluar narrativa

Capítulo 3: Implementación y pruebas

	historia, misiones y diálogos.				-Verifica si el formato de los datos es válido. -Guarda la evaluación del indicador de narrativa en el sistema.	
EC1.4	No se inserta ningún dato en los campos obligatorios.	9	9	5	Resalta de color rojo los campos vacíos y muestra el mensaje: "El campo no puede estar vacío".	Se mantiene en la página de evaluar narrativa
Faltan datos obligatorios.						
		5	7			
			5	8		
		9		5		
EC1.5	Los datos insertados no están en el rango establecido	11	12	0	Resalta de color rojo los campos que se pasan del rango establecido y muestra el mensaje: "El campo no debe ser menor a 1" o "El campo no debe ser mayor a 10"	Se mantiene en la página de evaluar narrativa
Datos fuera del rango establecido		5	52	-69		
		1	3	1		
		1	1	2		
		1	7	1		
		1	8	9		
		6	-1	7		
EC1.6	Los datos insertados son inválidos.	Diez	Uno	Dos	Resalta de color rojo los campos que tienen un formato inválido y muestra el mensaje: "El campo debe ser un número entero"	Se mantiene en la página de evaluar narrativa
Datos inválidos		3	10.5	10,8		
		Nueve	4	ocho		
		@	-*+	8		
		6	5	-16.52		
		()	7	9		
		1	Ninguno	10		
EC1.7	Permite omitir la evaluación del	N/A	N/A	N/A	Va a la interfaz del siguiente indicador	Redirecciona al usuario a la página de evaluar gráfico
Opción Omitir						

	indicador narrativa					
--	------------------------	--	--	--	--	--

Tabla 5. DCP del CU: Evaluar videojuego, Sección 1: Evaluar narrativa

Nota: Las celdas de la tabla contienen N/A, lo que significa no aplica.

En la siguiente tabla se muestra la descripción de las variables que intervienen en el DCP, así como los valores válidos que las mismas pueden tomar.

Descripción de las variables

No	Nombre del campo	Clasificación	Valor Nulo	Descripción
1.	Historia	Campo de texto	No	Formato: número entero Rango: 1-10
2.	Misiones	Campo de texto	No	Formato: número entero Rango: 1-10
3.	Diálogos	Campo de texto	No	Formato: número entero Rango: 1-10

Tabla 6. Descripción de las variables del CU: Evaluar videojuego, Sección 1: Evaluar narrativa.

3.3.2 Pruebas de regresión

Las pruebas de regresión consisten en la repetición selectiva de pruebas para detectar fallos introducidos durante la modificación de un sistema o componente de un sistema. Se efectúan para comprobar que los cambios no han originado efectos adversos no intencionados o que se siguen cumpliendo los requisitos especificados (29).

Como parte de la estrategia de prueba sobre la solución desarrollada se efectuaron al inicio de las iteraciones 2 y 3 las pruebas de regresión, que estuvieron dirigidas a detectar no conformidades (NC) que quedaran pendientes de la iteración anterior a estas. De esta manera se garantiza que se puedan iniciar de manera satisfactoria las iteraciones mencionadas teniendo en cuenta que fueron corregidas las NC pendientes de las iteraciones anteriores.

3.4 Resultados de las pruebas

Como resultado de las pruebas realizadas una vez concluida la implementación del módulo, se detectaron un conjunto de no conformidades (NC), las cuales fueron clasificadas en: significativas, no significativas y recomendaciones. En la siguiente tabla se muestran los resultados obtenidos luego de aplicadas las pruebas al módulo desarrollado:

Clasificación de las NC	1ra Iteración	2da Iteración	3ra Iteración
Significativas	18	9	0
No significativas	14	4	0
Recomendaciones	12	6	3

Tabla 7. Cantidad de no conformidades por iteración

Para una mejor comprensión de los resultados, se muestra en el siguiente gráfico la cantidad de no conformidades encontradas en cada una de las iteraciones realizadas.

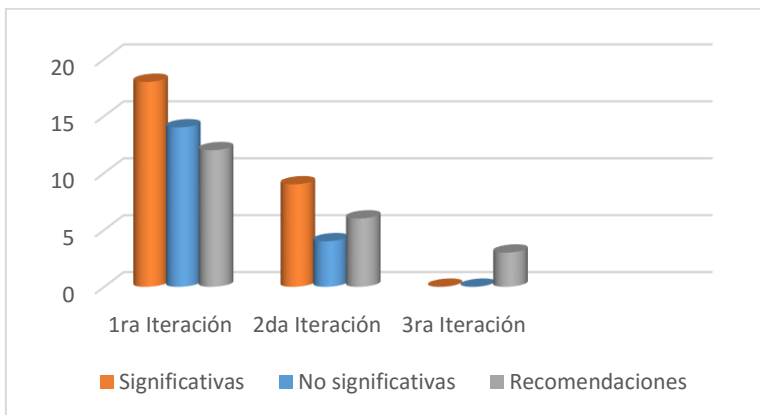


Figura 12. Gráfico de no conformidades.

Las no conformidades detectadas en cada iteración fueron analizadas y resueltas por el equipo de desarrollo. Las pruebas se realizaron de forma iterativa e incremental, logrando que para cada iteración fueran corregidos los errores detectados en las iteraciones anteriores, lo que contribuyó a mejorar la calidad y funcionalidad del módulo implementado. Las principales NC no significativas encontradas fueron errores ortográficos, tanto omisiones de tildes como cambios de mayúsculas por minúsculas y cambios de idioma. Las principales NC significativas

encontradas fueron errores al ejecutar acciones como eliminar y mostrar tanto en las alertas como en los archivos modificados, además de la validación de los datos en cuanto a tipo y rango. Además, la acción exportar no se realizaba correctamente en una de sus opciones.

3.5 Consideraciones parciales

El modelo de implementación obtenido permitió describir cómo los elementos del modelo de diseño se implementan en términos de componentes. El método de prueba seleccionado permitió detectar los errores existentes al concluir la implementación del módulo, los cuales fueron revisados y corregidos para garantizar la calidad del mismo.

Conclusiones Generales

La investigación realizada permite arribar a las siguientes conclusiones:

Se propone la base inicial para el proceso de evaluación de videojuegos en Calisoft, basado en la adecuación de los indicadores definidos como fundamentales en los videojuegos, que orienta a los probadores a enfocarse en los elementos básicos de este tipo de software.

Se propone una herramienta informática que sirve de complemento dentro del proceso de evaluación de videojuegos basada en el criterio de todos los probadores para emitir una evaluación final del producto.

Recomendaciones

A partir del trabajo realizado se recomienda:

- Agregar más géneros de videojuego para la evaluación.
- Agregar más indicadores para evaluar los videojuegos.

Referencias Bibliográficas

1. Crosby, Philip B. *The Absolutes of Leadership*. s.l. : Pfeiffer & Co, 1996. ISBN : 9780893842765.
2. Juran, Joseph M. *Manual de Control De Calidad*. s.l. : S.A. MCGRAW-HILL / INTERAMERICANA DE ESPAÑA, 1993. p. 1984. ISBN: 9788448100551.
3. Deming, W. Edwards. *Calidad, Productividad y Competitividad. La salida de la crisis*. Madrid : Díaz de Santos, S.A, 1989. ISBN:8487189229.
4. García León, D., & Beltrán Benavides, A. *Un enfoque actual sobre la calidad del software*. La Habana : ACIMED, 1995. Vol. 3. ISSN:1024-9435.
5. Cueva Lovelle, Juan Manuel. *Calidad del Software*. España : s.n., 2009.
6. ISO25000 Calidad del Producto Software. *ISO25000 Calidad del Producto Software*. [Online] 2017. [Cited: enero 2017, 5.]
<http://iso25000.com/index.php/normas-iso-25000/iso-25010?limit=3&limitstart=0>.
7. Mark J. P. Wolf, Bernard Perron. INTRODUCCIÓN A LA TEORÍA DEL VIDEOJUEGO. 2005. 4, p. 27. ISSN: 2385-3697.
8. Latorre, Óliver Pérez. Géneros de juegos y videojuegos. Una aproximación desde diversas perspectivas teóricas. España : Revista de Investigación y de Análisis, 2011. Vol. 28, 1. ISSN: 2014-0304.
9. Jose Luis González Sánchez Natalia Padilla Zea Francisco L. Gutiérrez. *From Usability to Playability: Introduction to Player-Centred Video Game Development Process*. Berlin : Springer, 2009.
10. Staffan Bjork; Jussi Holopainen. *Patterns in Game Design*. Hingham : Charles River Media, 2005. ISBN: 1584503548..
11. 3DJuegos. *3DJuegos*. [Online] 2017. [Cited: enero 2017, 17.]
<http://www.3djuegos.com/index.php?zona=3djuegos&subzona=sistema-valoracion>.

12. Meristation. *Miristation*. [Online] 2017. [Cited: enero 2017, 18.]
<http://www.meristation.com/nota>.
13. Company Headquarters. Visual Paradigm. [Online] 2009. [Cited: 3 18, 2014.]
<http://www.visual-paradigm.com>.
14. PHP.net. PHP. [Online] 2001. [Cited: 3 5, 2014.]
<http://www.php.net/manual/es/intro-what-is.php>.
15. Jetbrains. *Jetbrains*. [Online] 2017. [Cited: Mayo 20, 2017.]
<https://www.jetbrains.com/phpstorm/features/>.
16. Gestor de Base de Datos. [Online] [Cited: mayo 2017, 8.]
<http://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bid/406547/tipos-y-funciones-de-los-gestores-de-bases-de-datos>.
17. Oracle. *Oracle*. [Online] 2017. [Cited: Mayo 2017, 30.]
<https://www.oracle.com/mysql/index.html>.
18. Sistemas Manejadores de Bases de datos. *Sistemas Manejadores de Bases de datos*. [Online] Febrero 2011. [Cited: Mayo 2017, 30.]
<http://sistemamanejadordebasededatosmbd.blogspot.com/2011/02/diferentes>.
19. The Apache Software Foundation. *The Apache Software Foundation*. [Online] [Cited: Junio 2017, 1.] http://httpd.apache.org/ABOUT_APACHE.html.
20. Pachucho Hernández, Betty Marlene. Scribd. [Online] 5 2009. [Cited: 2 3, 2014.] <http://es.scribd.com/doc/55136485/47/Ventajas-de-utilizar-un>.
21. Yiiframework. *Yiiframework*. [Online] 2017. [Cited: Junio 2017, 2.]
<http://www.yiiframework.com/doc/guide/1.1/es/quickstart.what-is-yii>.
22. Sommerville, Ian. *Ingeniería del software Séptima edición*. Madrid : s.n., 2005.
23. Richard N. Taylor, Nenad Medvidovic, Eric Dashofy. *Software Architecture- Foundations, Theory and Practice*. 2009. ISBN: 0470167742.
24. E. Gamma, R. Helm, R., Johnson, and J., Vlissides. Design Patterns. [Online] 1995. [Cited: Mayo 23, 2017.] <http://siul02.si.ehu.es/~alfredo/iso/06Patrones.pdf>.

25. Larman, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Mexico : s.n., 1999. ISBN: 9701702611.
26. Fowler, Martin. *Patterns of enterprise application architecture*. s.l. : Addison-Wesley, 2003. ISBN: 0321127420.
27. JACOBSON, Ivar, BOOCH, Grady and RUMBAUGH, James. *El Proceso Unificado de Desarrollo de Software*. Madrid : s.n., 2000. ISBN: 8478290362.
28. Pressman, Roger S. *Ingeniería de Software. Un enfoque práctico*. 2005. ISBN: 0072853182.
29. Técnicas de Evaluación Dinámica. [Online] [Cited: Junio 6, 2017.] <http://www.lsi.us.es/docencia/get.php?id=361>.