

# Validación de Documentos

José Ignacio Botero Osorio

**Instructor**

Lyam Acosta Forero

**Aprendiz**

SENA – Servicio Nacional de Aprendizaje  
Tecnología en análisis y Desarrollo de Software

3118526

Bogotá D.C.

# Contenido

Validación de Documentos .....	3
¿Qué es un artefacto? .....	3
¿Tipos de artefactos? .....	4
¿Qué es la evaluación de artefactos? .....	5
¿Cómo se realizan? .....	5
¿Qué instrumentos se utilizan? .....	6
¿Qué resultados se obtienen? .....	7

# Validación de Documentos

En esta sección revisaremos la validación de los documentos de nuestros videojuegos, mejorando la validación anterior de requisitos o documentos que se presentaron para ellos revisaremos los artefactos de nuestro proyecto, los tipos de artefactos para nuestro videojuego y la evaluación para nuestro videojuego, entre muchos otros, todos estos instrumentos nos permiten verificar el estado de desarrollo de nuestro videojuego y tomar acciones a tiempos de manera más detallada.

## ¿Qué es un artefacto?

“Un **artefacto** es cualquier producto tangible o intangible generado durante el proceso de desarrollo de software. Puede incluir documentos, código fuente, modelos, plantillas, diagramas o incluso bases de datos. En términos generales, representa un elemento que captura y comunica información dentro del ciclo de vida del software.

Ejemplos: especificaciones de requisitos, diagramas UML, manuales de usuario, reportes de pruebas o scripts de instalación.”<sup>1</sup>

“En el desarrollo de software, un artefacto de software es un elemento que se produce durante el proceso de desarrollo. Puede ser un modelo de datos, un prototipo, un diagrama de flujo de trabajo, un documento de diseño o un script de configuración. De hecho, existen artefactos específicos que se requieren durante un ciclo de desarrollo y que deben almacenarse de forma accesible.”<sup>2</sup>

**En nuestro proyecto** todo producto generado durante un ciclo de desarrollo ya sea técnico, artístico o de diseño, incluyen tanto documentos conceptuales como elementos digitales del propio juego los cuales son:

1. Documento de diseño (GDD: Game Desing Document)
1. Modelos 3D y texturas
2. Scripts de programación (Mecánicas y lógica de programación)
3. Archivos de audio, música y voces
4. Niveles, escenarios y prototipos jugables

---

<sup>1</sup> Texto Generado por IA - Perplexity

<sup>2</sup> [https://www-leanix-net.translate.goog/en/wiki/trm/software-artifacts?\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=es&\\_x\\_tr\\_hl=es&\\_x\\_tr\\_pto=tc](https://www-leanix-net.translate.goog/en/wiki/trm/software-artifacts?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)

## ¿Tipos de artefactos?

1. **“Código fuente:** El código real escrito por desarrolladores en lenguajes de programación como Java, Python o JavaScript. Constituye la base de cualquier aplicación de software.
2. **Binarios compilados:** archivos ejecutables generados al compilar el código fuente. Son las versiones ejecutables de la aplicación.
3. **Bibliotecas:** módulos de código reutilizables que proporcionan funcionalidades específicas. Las bibliotecas pueden compartirse entre diferentes proyectos para evitar redundancias.
4. **Archivos de configuración:** archivos que definen la configuración y los parámetros de la aplicación. Garantizan su correcto funcionamiento en diferentes entornos.
5. **Imágenes de contenedor:** Aplicaciones empaquetadas y sus dependencias en una única unidad portátil. Las imágenes de Docker son un ejemplo común, utilizadas para una implementación consistente en diferentes entornos.
6. **Documentación:** documentos, especificaciones y comentarios de código que proporcionan contexto y orientación para el desarrollo y uso del software.
7. **Scripts de compilación:** scripts que compilan el código fuente en binarios ejecutables para garantizar la coherencia en el proceso de compilación.
8. **Scripts de implementación:** scripts automatizados que manejan la instalación y configuración de la aplicación en entornos de producción.
9. **Casos de prueba:** procedimientos utilizados para verificar que el software se comporta según lo previsto. Son cruciales para el control de calidad.
10. **Registros:** registros del comportamiento y el rendimiento de la aplicación para la resolución de problemas y la optimización.”<sup>3</sup>

### En nuestro proyecto necesitaremos los siguientes artefactos:

1. Artefactos de diseño: La cual nos da descripciones de mecánicas, niveles, narrativas, reglas, interfaces y progresión del juego.
2. Artefactos artísticos: Modelos 3D, Sprites, Efectos visuales, animaciones y arte conceptual.
3. Artefactos de Programación: Scripts en motores como Unity o Unreal, control de físicas, Inteligencia Artificial o las interacciones.
4. Artefactos de sonido: Música de ambientación, también los efectos de sonido, voces y mezclas de audio.
5. Artefactos de Prueba: Reporte de testeo, lista de errores y evaluaciones de rendimiento.

---

<sup>3</sup> [https://jfrog-com.translate.goog/learn/devops/software-artifact/?\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=es&\\_x\\_tr\\_hl=es&\\_x\\_tr\\_pto=tc](https://jfrog-com.translate.goog/learn/devops/software-artifact/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)

## ¿Qué es la evaluación de artefactos?

“La evaluación de artefactos es un proceso mediante el cual se **evalúa** el impacto de un **cambio** en un **software** existente. Se realiza mediante el **análisis de los artefactos** generados durante el **desarrollo del software**.

Los resultados de la evaluación de artefactos permiten **determinar el impacto** de un cambio en el **software** y el **grado de madurez** del mismo.”<sup>4</sup>

En nuestro videojuego tenemos que revisar, probar y validar los recursos generados para asegurar que funcionan correctamente, sean coherentes con la visión del juego y proporcionen buena experiencia al jugador lo que significa:

1. Jugabilidad y mecánicas
2. Consistencia visual y artística
3. Calidad de audio y sincronización
4. Interfaz y respuesta al usuario
5. Rendimiento del motor de juego

## ¿Cómo se realizan?

El proceso de evaluación suele incluir los siguientes pasos:

1. **Revisión del artefacto:** análisis detallado del contenido, estructura y cumplimiento de criterios.
2. **Identificación de no conformidades:** detección de errores, omisiones o inconsistencias.
3. **Registro de observaciones:** documentación de comentarios y sugerencias de mejora.
4. **Corrección y seguimiento:** ajuste del artefacto y validación de las correcciones.<sup>5</sup>

En nuestro proyecto el proceso de validación sería el siguiente:

- Pruebas Funcionales y de jugabilidad: Debido a que necesitamos revisar las mecánicas respondan correctamente y sin errores.
- Revisión visual y artística: Se asegura que el arte y las animaciones mantenga coherencia de estilo.

---

<sup>4</sup> <https://brainly.lat/tarea/65663185>

<sup>5</sup> Texto Generado por IA - Perplexity

- Pruebas de Rendimiento: Se evalúan los Frames por segundo (FPS), temperatura de la GPU y estabilidad general.
- Prueba de usuario (Playtesting): Se analiza la satisfacción del jugador y su interacción con la interfaz.
- Revisión entre equipos: Artistas, Programadores y diseñadores validan los artefactos propios y cruzados.

## ¿Qué instrumentos se utilizan?

“Los instrumentos o herramientas más comunes para la validación de artefactos son:

- **Listas de verificación:** guías con criterios o preguntas específicas que ayudan a controlar la calidad del artefacto.
- **Plantillas:** estructuras estandarizadas que garantizan la uniformidad de la documentación.
- **Guías o manuales de evaluación:** documentos que ofrecen procedimientos o criterios calificados.
- **Herramientas de análisis estático:** detectan errores de código sin necesidad de ejecutarlo (por ejemplo, SonarQube o PMD).
- **Herramientas de pruebas dinámicas:** validan el comportamiento del software frente a los requisitos mediante pruebas funcionales o de integración.”<sup>6</sup>

En nuestro videojuego usaremos:

- Lista de chequeo (Lista QA): Que definen criterios técnicos y artísticos
- Herramientas de prueba automatizada: Por ejemplo, Unity Test Runner o Unreal Automation Tool.
- Benchmark Visuales: 3DMark o Unigine Superposition para medir el rendimiento gráfico.
- Tracker de errores: Como Jira, Trello o Bugzilla
- Sistema de control de Versiones: Como Git, Perforce o Plastic SCM para asegurar la integridad y trazabilidad
- Plataforma de testeo masivo: Como game Tester para que nos dé una buena retroalimentación de jugadores.

---

<sup>6</sup> Texto Generado por IA - Perplexity

## **¿Qué resultados se obtienen?**

Para nuestro videojuego se producen resultados medibles que garanticen la calidad de nuestro juego con los siguientes resultados:

1. Confirmación funcional: Las mecánicas y sistemas operas según lo que se planteó en el documento y en las reuniones
2. Uniformidad visual y de sonido: Coherencia entre el arte, animación y audio
3. Lista de incidencias: Registro de Bugs y su estado de Corrección
4. Informes de rendimiento y estabilidad: FPS, carga de CPU y GPU, uso de memoria
5. Feedback del jugador: Datos reales sobre la experiencia de juego y la diversión.