



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SOFTWARE

Integrante: Carlos Morales

Materia: Desarrollo de Juegos Interactivos

Curso: GR3SW

Proyecto.

**Análisis y Descomposición de Requerimientos del
Game Design Document (GDD)**

Tema: *El GDD: De Requerimientos a un Proyecto Gestionable.*

1. Fase I: Arqueología de Diseño (Investigación del GDD)

En el desarrollo de software, uno de los principales factores de fracaso de los proyectos no es la calidad del código, sino una deficiente definición y gestión del alcance (scope). En el desarrollo de videojuegos, esta problemática se aborda mediante el uso del *Game Design Document* (GDD), el cual actúa como la fuente principal de requerimientos del sistema.

El presente análisis se enfoca en estudiar un GDD real y comprender cómo una visión creativa puede traducirse en un proyecto de software gestionable desde una perspectiva de ingeniería.

1.1. Juego y Fuente

Juego: DOOM (1993).

Fuente: <https://www.doomworld.com/10years/doomdesign.pdf>

1.2. Visión Central del Juego

El concepto central de DOOM es un videojuego de disparos en primera persona enfocado en la acción rápida, la exploración de niveles laberínticos y el combate continuo contra múltiples enemigos, priorizando la fluidez del movimiento y la respuesta inmediata del jugador por encima de elementos narrativos complejos.

1.3. Resumen Estructural del GDD

El GDD de DOOM se encuentra organizado en varias secciones clave que definen el alcance del sistema desde diferentes perspectivas:

- **Mecánicas de Juego:** Describe el movimiento del jugador, el sistema de disparo, el uso de armas y la interacción con el entorno.



- **Diseño de Niveles:** Explica la estructura de los mapas, el uso de habitaciones, pasillos, puertas y llaves para guiar la progresión del jugador.
- **Enemigos e Inteligencia Artificial:** Define los distintos tipos de enemigos, sus patrones de comportamiento y su rol dentro del desafío del juego.
- **Sistema de Combate:** Detalla el funcionamiento de las armas, el daño, la munición y el balance del enfrentamiento.
- **Interfaz de Usuario (HUD):** Describe los elementos visuales que informan al jugador sobre su estado, como salud, munición y armas disponibles.
- **Aspectos Técnicos:** Incluye consideraciones sobre rendimiento, motor gráfico y limitaciones tecnológicas de la época.

1.4. Análisis desde la Perspectiva de Ingeniería

Desde un punto de vista de ingeniería de software, la sección más clara y útil del GDD es la relacionada con las mecánicas de juego y el diseño de niveles, ya que estas describen comportamientos concretos que pueden traducirse fácilmente en funcionalidades técnicas.

Sin embargo, se identifica una carencia en la definición explícita de requerimientos no funcionales, tales como métricas de rendimiento, consumo de recursos o escalabilidad del sistema. Estos aspectos son críticos en proyectos modernos y su ausencia puede generar ambigüedad durante la implementación.

2. Fase II: Traducción a Historias de Usuario (Desglose de Requerimientos)

Para transformar el GDD en un proyecto gestionable, los requerimientos se descomponen en Épicas, utilizando el formato estándar de Historias de Usuario. Estas épicas representan funcionalidades de alto nivel que, en conjunto, describen el alcance total del sistema.

2.1. Lista de Épicas (Entregable 2)

1. **Control** **del** **Jugador**
Como Jugador, quiero moverme libremente en primera persona, para explorar los niveles y enfrentar enemigos.
2. **Sistema** **de** **Disparo**
Como Jugador, quiero disparar armas utilizando diferentes tipos de munición, para eliminar enemigos.
3. **Cambio** **de** **Armas**
Como Jugador, quiero cambiar entre distintas armas, para adaptarme a diferentes situaciones de combate.
4. **Enemigos** **Básicos**
Como Motor del Juego, quiero que los enemigos se muevan y ataquen de forma autónoma, para generar desafío.
5. **Daño** **y** **Salud**
Como Jugador, quiero recibir daño al ser atacado, para que exista una consecuencia al fallar.



6. **Recolección** **de** **Ítems**
Como Jugador, quiero recoger munición, botiquines y armaduras, para mantenerme con vida durante el nivel.
7. **Diseño** **de** **Niveles**
Como Motor del Juego, quiero cargar niveles con múltiples habitaciones y pasillos, para estructurar la experiencia de exploración.
8. **Llaves** **y** **Puertas**
Como Jugador, quiero encontrar llaves que desbloqueen puertas, para avanzar en el nivel.
9. **HUD** **del** **Jugador**
Como Jugador, quiero ver mi salud, munición y arma actual en pantalla, para conocer mi estado en todo momento.
10. **Sistema** **de** **Muerte**
Como Jugador, quiero morir al perder toda mi salud, para que el juego tenga un estado de falla claro.
11. **Progresión** **de** **Niveles**
Como Jugador, quiero completar un nivel y avanzar al siguiente, para progresar en el juego.
12. **Audio** **y** **Efectos** **Sonoros**
Como Jugador, quiero escuchar disparos y sonidos ambientales, para aumentar la inmersión.
13. **Ambientación** **Visual**
Como Jugador, quiero gráficos coherentes con una atmósfera oscura y violenta, para reforzar la temática del juego.

3. Fase III: Mapa de Proyecto (Estimación y Priorización)

En metodologías ágiles, la planificación inicial se centra en definir un Producto Mínimo Viable (MVP). Para ello, las épicas se agrupan en fases de entrega que permiten comprender la magnitud y el flujo del proyecto.

3.1.Fase 1: Prototipo Jugable (MVP)

Las siguientes épicas son absolutamente esenciales para que el juego sea funcional:

- Control del Jugador
- Sistema de Disparo
- Enemigos Básicos
- Daño y Salud
- HUD del Jugador

Sin estas funcionalidades, el juego no puede considerarse jugable.

3.2.Fase 2: Juego Completo (Features Centrales)

Estas épicas construyen la experiencia completa del juego:

- Cambio de Armas
- Recolección de Ítems
- Diseño de Niveles



- Llaves y Puertas
- Sistema de Muerte
- Progresión de Niveles

3.3.Fase 3: Pulido y Contenido Adicional

Las siguientes épicas mejoran la experiencia, pero no son críticas para el lanzamiento inicial:

- Audio y Efectos Sonoros
- Ambientación Visual

4. Conclusión

El análisis del GDD de DOOM demuestra cómo una visión de diseño puede transformarse en un conjunto estructurado de requerimientos técnicos mediante el uso de Historias de Usuario. Este proceso permite gestionar el alcance del proyecto, identificar riesgos tempranos y establecer una planificación realista basada en entregas incrementales, alineando el diseño creativo con prácticas formales de ingeniería de software.