



## DESARROLLO DE JUEGOS INTERACTIVOS

**NOMBRE ESTUDIANTE:** Marlow Armijo  
**FECHA:** 18-11-2025

**TEMA:** Era de la Revolución 3D en consolas (aprox. 1994–2000) –  
PlayStation / Nintendo 64

---

### Juego Seleccionado

#### Crash Bandicoot

- **Año de lanzamiento:** 1996
- **Plataforma original:** Sony PlayStation (PS1)
- **Desarrollador:** Naughty Dog
- **Distribuidor:** Sony Computer Entertainment / Universal Interactive



#### 1 Hito Tecnológico Clave

##### Tecnología dominante:

- Consola PlayStation 1 con CD-ROM y capacidades 3D por hardware.
- PS1 ofrecía 2 MB de RAM principal y 1 MB de VRAM, muy poco para manejar mundos 3D grandes, pero con una unidad de CD-ROM que permitía almacenar decenas de megabytes de datos (texturas, modelos, música).

##### Por qué es clave para este juego:

- **CD-ROM:** permitía niveles con más geometría, cinemáticas y música comprimida que lo que cabía en un cartucho típico de 16 bits.



## DESARROLLO DE JUEGOS INTERACTIVOS

- **Hardware 3D dedicado:** la PS1 podía dibujar polígonos en tiempo real, lo que permitió que Crash fuera una plataforma 3D “into the screen” (avanzas hacia el fondo) en vez de sólo plataformas 2D.
- Pero todo esto estaba limitado por muy poca RAM, lo que obligó a Naughty Dog a ser extremadamente creativos: ellos mismos cuentan que tenían niveles con más de 10 MB de datos corriendo en una máquina con sólo 2 MB de RAM, usando técnicas de paginación agresivas.

## 2 Análisis de Diseño (MDA)

### Mecánicas (M)

Acciones y sistemas centrales del jugador:

#### 1. Plataformeo 3D guiado por corredores

- Correr, saltar, hacer spin y rebotar sobre enemigos mientras avanzas por corredores estrechos 3D (hacia el fondo de la pantalla, lateralmente o hacia la cámara).
- La cámara está muy controlada: casi siempre fija o sobre raíles, no libre.

#### 2. Gestión de cajas (crates) y Wumpas

- Romper cajas con diferentes funciones: normales, explosivas (TNT), de checkpoint, de vida extra, etc.
- Coleccionar frutas Wumpa y vidas, sistema de riesgo/recompensa (te arriesgas a romper todas las cajas del nivel para el 100%).

#### 3. Ritmo de ensayo y error

- Niveles relativamente cortos, pero muy exigentes: pequeños errores te matan.
- Checkpoints y vidas limitadas generan un bucle de intentar → fallar → aprender patrones → dominar.

### Estéticas (A)

¿Qué siente el jugador?

- **Desafío (Challenge / Superación):**  
Dominar el timing de saltos, esquivar obstáculos y memorizar patrones de enemigos. La dificultad va escalando, aunque la estructura del juego sea simple.
- **Sensación de Flujo / Dominio:**  
Cuando conoces un nivel, puedes encadenar saltos y spins casi de memoria, logrando una sensación de fluidez muy satisfactoria.
- **Fantasía / Personaje icónico:**  
Eres una especie de “mascota loca” de PlayStation, Crash, con animaciones



## DESARROLLO DE JUEGOS INTERACTIVOS

exageradas y humor slapstick. La estética caricaturesca suaviza la frustración de morir muchas veces.

- **Tensión ligera:**

En secciones donde corres hacia la cámara o te persigue una roca gigante, se genera una tensión divertida, pero no de terror, sino de adrenalina tipo dibujos animados.

### 3 Innovación Clave (El “Salto”)

¿Qué hizo Crash que sus predecesores no podían (o no habían hecho así)?

1. Plataforma 3D “pasillo” super pulido en consola doméstica

- En lugar de un mundo completamente abierto (como Mario 64), Crash propone una estructura de niveles lineales en 3D, que mezcla la claridad de una plataforma 2D con la espectacularidad 3D.
- Esta decisión de diseño está ligada directamente a la capacidad limitada de la PS1 (ver sección 4) y se convirtió en un subgénero propio: “corridor 3D platformer”.

2. Calidad gráfica altísima para la época en un hardware muy limitado

- Uso agresivo de texturas, animaciones y escenarios relativamente detallados para PS1, al punto de convertir a Crash en mascota no oficial de PlayStation.
- Naughty Dog desarrolló herramientas propias de compresión y streaming para meter esa calidad en sólo 2 MB de RAM.

3. Construcción de marca/mascota para una consola nueva

- Crash ayudó a definir la identidad de PlayStation (juvenil, irreverente, “cool”), compitiendo con Mario y Sonic.

### 4 La “Restricción Ingeniosa” (El Desafío de Ingeniería)

La Restricción: ¿Cuál era el problema de hardware?

1. Muy poca memoria RAM para un juego 3D “grande”

- La PS1 tenía 2 MB de RAM principal + 1 MB de VRAM.
- Los niveles de Crash, según los propios desarrolladores, podían tener más de 10 MB de datos, mucho más de lo que cabía en memoria de golpe.

2. Unidad de CD-ROM relativamente lenta

- Leer datos desde el CD en medio del juego podía causar “hitches” o tirones (bajones de FPS), lo cual arruinaría un plataformero que exige precisión.

3. Limitaciones del hardware 3D

- El PS1 tenía capacidad de polígonos limitada y sin z-buffer moderno, por lo que demasiada geometría visible a la vez generaba caída de rendimiento y problemas de orden de dibujo (polígonos que se “atraviesan”).



## DESARROLLO DE JUEGOS INTERACTIVOS

### 4. Objetivo de mantener 30 FPS estables

- Naughty Dog quería que el juego se sintiera fluido y responsivo, por lo que no podían permitirse picos de carga ni escenas con muchos polígonos simultáneos.

#### La Solución

Aquí viene la parte jugosa de ingeniería. Naughty Dog responde a esas restricciones con una combinación de diseño de niveles + “truco” de ingeniería de sistemas:

##### a) Streaming y paginación agresiva desde el CD

- Andy Gavin (cofundador de Naughty Dog) explica que diseñaron un sistema de paginación de datos que cargaba y descargaba páginas de 64 KB de datos de nivel mientras Crash avanzaba, de modo que nunca estaba todo el nivel completo en RAM, sólo la parte necesaria alrededor del jugador.
- Esto significa que el juego streamea constantemente del CD durante el gameplay, en lugar de cargar todo al principio como hacían muchos juegos más simples.
- El truco fue hacerlo de forma invisible para el jugador: sin pantallas de carga en medio del nivel, sin pausas notorias.

#### Impacto en el diseño:

- Los niveles están construidos como “tubos” o caminos estrechos que permiten predecir qué datos se necesitarán a continuación, haciendo más fácil decidir qué páginas cargar/descargar.

##### b) Corredores y cámara sobre raíles para controlar el costo 3D

- Naughty Dog “hackeó” la forma de usar el 3D:
  - En lugar de un mundo abierto donde las cámaras pueden mirar a cualquier lado (y por tanto ver MUCHOS polígonos), limitaron el juego a corredores con cámara fija o predefinida.
  - Así podían garantizar un límite máximo de polígonos visibles, manteniendo el rendimiento dentro de lo que la PS1 podía manejar.
- Andy Gavin comenta que tuvieron que decidir restringir una dimensión para evitar el problema de “espacios vacíos” y mantener el control técnico del nivel: el juego es esencialmente una plataforma 2D “curvado” en 3D.

#### Impacto en la experiencia del jugador (Estética):

- Esa decisión técnica se convierte en parte de la identidad del juego:
  - Secciones de correr hacia el fondo (3D tradicional).
  - Secciones “side-scroller” (más 2D).



## DESARROLLO DE JUEGOS INTERACTIVOS

- Secciones de correr hacia la cámara (tensión extra porque ves menos lo que viene).

### c) Reutilización creativa de assets: las cajas

- El diseño de Crash está descrito por el equipo como “51% necesidad técnica y visual, 49% inspiración”: muchas decisiones visuales nacen de la necesidad de simplificar y reutilizar.
- Las cajas (crates) son un elemento perfecto para esto:
  - Se pueden repetir constantemente con variaciones mínimas (textura y comportamiento: normal, TNT, salto, etc.), lo que ahorra memoria.
  - Sirven como feedback visual claro en escenarios con detalle limitado.
  - Funcionan como herramienta de diseño de niveles (marcan ritmo, peligros, recompensas) sin requerir nuevos modelos complejos.

### d) Hacking de las librerías de Sony

- Según entrevistas y el mini-documental “War Stories: How Crash Bandicoot hacked the original PlayStation”, Naughty Dog llegó a reemplazar partes de las librerías oficiales de Sony, optimizando directamente cómo se usaba la memoria y el hardware para ganar velocidad y espacio.
- Esto es literalmente “ingeniería de sistemas”: reescribir capas de bajo nivel para exprimir cada ciclo de CPU.

### Resumen de la “Restricción Ingeniosa”

- **Restricción:**
  - Consola con muy poca RAM (2 MB) y hardware 3D limitado.
  - CD-ROM lento para cargar grandes niveles sin pausas.
- **Hack:**
  - Sistema de streaming y paginación de 64 KB que está leyendo datos del CD continuamente para simular niveles mucho más grandes de lo que la memoria permitiría.
  - Diseño de niveles en forma de corredor con cámara sobre raíles para limitar polígonos en pantalla y predecir qué datos se necesitan.
  - Reutilización masiva de assets simples pero muy expresivos (cajas, props modulares) y optimización extrema de las librerías de hardware.



## DESARROLLO DE JUEGOS INTERACTIVOS

### Cómo conecta con el ejemplo de Space Invaders del enunciado:

- Igual que en Space Invaders, donde el hardware lento hacía que el juego se acelerara cuando quedaban menos aliens (bug → feature), en Crash la limitación de memoria/CD obligó a inventar un tipo de juego 3D “en tubo”.
- El resultado no es solo un “apaño técnico”, sino una identidad de juego: niveles muy dirigidos, ritmo alto y una estética visual que sigue siendo reconocible décadas después.

### 5 Resultados

Del análisis del caso de estudio Crash Bandicoot (Naughty Dog, 1996) en la era de la PlayStation 1 (consolas 3D con CD-ROM) se obtienen los siguientes resultados:

#### 1. Limitaciones de hardware bien definidas

- La PlayStation 1 disponía de solo 2 MB de RAM principal y 1 MB de VRAM, junto a una unidad de CD-ROM de ~300 KB/s de lectura máxima; esto es muy poco para manejar mundos 3D continuos con gráficos detallados.
- A pesar de esto, los niveles de Crash Bandicoot podían ocupar del orden de 10 MB de datos o más, es decir, varias veces la memoria disponible de la consola.

#### 2. Sistema de streaming y paginación como solución central

- Naughty Dog implementó un sistema de paginación/streaming que dividía los datos del nivel en bloques de aproximadamente 64 KB, cargados dinámicamente desde el CD conforme el jugador avanzaba.
- El diseño del disco (la organización física de los datos) fue optimizado para que esos bloques estuvieran en posiciones contiguas, reduciendo los tiempos de búsqueda del CD y evitando pausas visibles durante el juego.

#### 3. Diseño de niveles tipo “corredor” y cámara sobre raíles

- Para mantener el número de polígonos visibles dentro del límite que la PS1 podía manejar sin caídas de rendimiento, se diseñaron niveles como “túneles” o corredores 3D, en los que la cámara se desplaza sobre un “rail” predefinido en lugar de ser libre.
- Esta elección permitió precalcular visibilidad y orden de dibujo de polígonos en estaciones SGI externas, compensando que la PS1 no disponía de z-buffer moderno y tenía un límite estricto de cálculo geométrico.

#### 4. Optimización extrema del código y de las librerías

- Los desarrolladores llegaron a reescribir y optimizar partes de las librerías oficiales de Sony (“hacking” del SDK) para exprimir el rendimiento del hardware, tanto en memoria como en capacidad de cálculo.
- Se emplearon estructuras de datos muy compactas, compresión agresiva de recursos y un uso cuidadoso de la CPU MIPS de 33 MHz, ajustando incluso detalles de DMA y acceso a memoria.



## DESARROLLO DE JUEGOS INTERACTIVOS

### 5. Reutilización creativa de assets visuales y de diseño

- El personaje Crash está construido con alrededor de 512 polígonos y con un uso muy limitado de texturas para ahorrar memoria; el resto del “estilo” viene de la animación y la silueta del modelo.
- Elementos como las cajas (crates) se reutilizan masivamente, cambiando comportamiento y variando mínimamente su apariencia (TNT, vida, rebote, etc.), lo que reduce costos de memoria a la vez que aporta claridad y ritmo al gameplay.

### 6 Conclusiones

#### ▪ Las restricciones de hardware fueron el motor del diseño.

La combinación de poca RAM, CD lento y GPU limitada obligó a Naughty Dog a inventar soluciones como el streaming por páginas de 64 KB y el diseño de niveles tipo “corredor”. Estas decisiones técnicas terminaron definiendo la forma de jugar Crash Bandicoot.

#### ▪ El diseño de juego y la ingeniería de sistemas son inseparables.

La elección de una cámara sobre raíles, niveles lineales y la reutilización de assets no fue solo artística: fue una respuesta directa a los límites de la PlayStation 1. El resultado es un juego que se percibe fluido, vistoso y con una personalidad propia, pero construido sobre una fuerte disciplina de ingeniería.

#### ▪ Un “hack” técnico puede convertirse en una estética de juego.

Igual que en Space Invaders el aumento de velocidad no fue un diseño planeado sino un efecto de hardware que se aprovechó como mecánica, en Crash Bandicoot la necesidad de controlar el presupuesto de polígonos y la memoria se transformó en una estética de plataformeo 3D guiado, con secciones icónicas de correr hacia la cámara y un ritmo muy particular.

#### ▪ Lección general para la ingeniería de software/juegos.

Este caso demuestra que las restricciones no solo limitan, sino que canalizan la creatividad. Aprender a trabajar con presupuestos de CPU, memoria y ancho de banda no es un detalle de implementación, sino una parte central del diseño de sistemas interactivos.

### 7 Referencias

[1] A. Gavin, “Making Crash Bandicoot – Part 3,” *All Things Andy Gavin*, 04-Feb-2011. [En línea]. Disponible en: <https://all-things-andy-gavin.com/2011/02/04/making-crash-bandicoot-part-3/> (Consultado: 18-nov-2025).

[2] K. Orland, “War Stories: How Crash Bandicoot hacked the original PlayStation,” *Ars Technica*, 2021. [En línea]. Disponible en: <https://arstechnica.com/gaming/2021/09/war-stories-how-crash-bandicoot-hacked-the-original-playstation/> (Consultado: 18-nov-2025).



## DESARROLLO DE JUEGOS INTERACTIVOS

[3] “How Naughty Dog Fit Crash Bandicoot into 2MB of RAM on the PS1,” *Movie and TV Buff*, 25-jun-2015. [En línea]. Disponible en:

<https://movieandtvbuff.com/2015/06/25/how-naughty-dog-fit-crash-bandicoot-into-2mb-of-ram-on-the-ps1/> (Consultado: 18-nov-2025).

[4] “PlayStation technical specifications,” *Wikipedia, The Free Encyclopedia*. [En línea]. Disponible en:

[https://en.wikipedia.org/wiki/PlayStation\\_technical\\_specifications](https://en.wikipedia.org/wiki/PlayStation_technical_specifications) (Consultado: 18-nov-2025).

[5] “Sony PlayStation | RAM,” *Retro Arcade Memories*. [En línea]. Disponible en:

<https://retroarcadememories.wordpress.com/playstation/> (Consultado: 18-nov-2025).

[6] “Memory Map – PlayStation Specifications – PSX-SPX,” *ConsoleDev*. [En línea].

Disponible en: <https://psx-spx.consoledev.net/memorymap/> (Consultado: 18-nov-2025).

[7] B. Morton, “The Secrets of Space Invaders,” *IEEE Spectrum*, 2021. [En línea].

Disponible en: <https://spectrum.ieee.org/space-invaders> (Consultado: 18-nov-2025).