

Desarrollo de Videojuegos

Autor: Alejandro Adrián Iglesias

Director: Javier Blanqué



Universidad Nacional de Luján
Int. Ruta 5 y 7
6700 Luján, Buenos Aires
República Argentina
Año 2011

Desarrollo de videojuegos.

Alejandro Adrián Iglesias
Universidad Nacional de Luján
Int. Ruta 5 y 7
6700, Luján, Buenos Aires
República Argentina

Alejandro.adrian.iglesias@gmail.com

Resumen

El presente trabajo presenta los conceptos principales acerca del desarrollo de videojuegos teniendo en cuenta para ello una mirada integral de las características de los mismos, su historia, conceptos de diseño, clasificaciones, sus implicaciones sociales y éticas, las técnicas y procesos utilizadas para su desarrollo, las técnicas y algoritmos utilizados para su implementación y finalmente, las características de la industria mundial junto con una breve caracterización del escenario actual de la industria de videojuegos en Argentina. Este trabajo por lo tanto, tiene por objetivo introducir, desarrollar y brindar las herramientas necesarias para comprender este apasionante mundo que surge tanto de la industria del software, como la del entretenimiento y la del arte mismo.

Índice.

PARTE I: INTRODUCCION Y CONCEPTOS.

1 Introducción.

2 Videojuegos.

2.1 ¿Qué es un juego?

2.2 ¿Qué es un videojuego?

2.3 Reseña Histórica.

2.3.1 Primeros experimentos.

2.3.2 Primeros videojuegos y el Nacimiento de la Industria.

2.3.3 “La época dorada”.

2.3.4 El crash de la de industria.

2.3.5 El resurgimiento de la industria y la era de los 8 bits.

2.3.6 El período 1990-1995 y las consolas de 16 bits.

2.3.7 El periodo 1995-2000 y las consolas de 32 y 64 bits.

2.3.8 El periodo 2000-2005 y la sexta Generación de consolas.

2.3.9 2005-Actualidad: La séptima y octava generación de consolas.

3. Taxonomías.

3.1 Clasificación de los videojuegos.

3.2.1 Juegos de acción y habilidad.

3.2.2 Juegos de estrategia.

3.2.3 Juegos de aventura.

3.2.4 Juegos RPG (Role Playing Games).

3.2.5 Juegos de puzzles.

3.2.6 Simulaciones de vehículos.

3.2.7 Juegos de deportes.

3.2.8 Juegos de construcción y administración.

3.3 Juegos serios.

3.4 Juegos Casuales.

3.5 Juegos sociales.

3.6 Juegos Independientes.

3.7 Caracterización de los jugadores.

3.7.1 Jugadores “Core” y jugadores casuales.

3.7.2 Clasificaciones por contenido y edad.

3.7.2.1 Clasificación ESRB.

3.7.2.2 Clasificación PEGI.

4. Videojuegos, sociedad y cultura.

4.1 Videojuegos y controversias.

4.2 Videojuegos y violencia.

4.3 Los videojuegos y salud.

4.4 Los videojuegos y la cultura.

4.5 Los videojuegos como arte.

PARTE II: DESARROLLO.

5. Desarrollo de videojuegos.

- 5.1 Entidades intervinientes.
- 5.2 Ciclo de vida.
 - 5.2.1 Desarrollo del concepto.
 - 5.2.2 Pre-producción.
 - 5.2.1.1 El documento de diseño del juego.
 - 5.2.2.2 Plan de producción de arte.
 - 5.2.2.3 El documento de diseño técnico.
 - 5.2.2.4 El plan de proyecto.
 - 5.2.2.5 El prototipo.
 - 5.2.3 Producción/desarrollo.
- 5.3 Roles.
 - 5.3.1 División de producción.
 - 5.3.2 Diseño del juego.
 - 5.3.3 La división de programación.
 - 5.3.4 La división de Arte.
 - 5.3.5 La división de Música y sonido, y medios especializados.
 - 5.3.6 La división de control de calidad (QA).
- 5.4. Metodologías de desarrollo.
 - 5.4.1 Cascada.
 - 5.4.2 Prototipo Iterativo.
 - 5.4.3 Metodológica Incremental.
 - 5.4.4 Metodologías ágiles.

PARTE III: TECNOLOGÍA

- 6 Tecnología de videojuegos.
 - 6.1 Hardware.
 - 6.1.1 Consolas.
 - 6.1.2 Consolas portátiles.
 - 6.1.3 PC.
 - 6.1.3 Móviles.
 - 6.1.4 Interfaces.
 - 6.2 Software.
 - 6.2.1 Breve reseña histórica.
 - 6.2.2 Estado del arte.
 - 6.2.3 Estructura principal de un videojuego.
 - 6.2.3.1 Procesamiento de entrada.
 - 6.2.3.2 Procesamiento del estado del mundo simulado.
 - 6.2.3.3 Procesamiento gráfico.
 - 6.2.4 Representación 2D.
 - 6.2.5 Representación 3D.
 - 6.2.5.1 Determinación de la visibilidad.
 - 6.2.5.2 Determinación de la resolución.
 - 6.2.5.3 Transformaciones y luces.
 - 6.2.5.4 Rasterización.
 - 6.2.5.5 *Shaders*.
 - 6.2.6 APIs gráficas.
 - 6.2.6.1 DirectX.
 - 6.2.6.2 OpenGL.
 - 6.2.7 Manejo de audio.

- 6.2.8 Inteligencia Artificial.
 - 6.2.8.1 Maquinas de estados finitos.
 - 6.2.8.2 Lógica difusa.
 - 6.2.8.3 Árboles de decisiones.
 - 6.2.8.4 Pathfinding: A* (A estrella).
 - 6.2.8.5 Técnicas no determinísticas para aprendizaje.
 - 6.2.8.6 Scripting.
- 6.2.9 Juegos multiplayer.

PARTE IV: INDUSTRIA

- 7 Industria de los videojuegos.
 - 7.1 Caracterización general.
 - 7.2 Estructura.
 - 7.3 Cadena de valor.
 - 7.4 El mercado.
 - 7.5 Modelos de negocios.
 - 7.5.1 Retail.
 - 7.5.2 Distribución digital.
 - 7.5.3 Videojuegos en *Smartphones* y Dispositivos móviles.
 - 7.5.4 Suscripción.
 - 7.5.5 Micro-transacciones.
 - 7.5.6 Sponsorships y publicidad.
 - 7.5.7 Advergaming.
 - 7.5.8 Juegos sociales.
- 8. Desarrollo de videojuegos en argentina.
 - 8.1 Características de la industria nacional.
 - 8.2 El empleo y los profesionales.
 - 8.2 Organizaciones y eventos.

PARTE V: CONSIDERACIONES FINALES.

- 9 Consideraciones finales.
- 10 Glosario.
- 11 Abreviaturas.
- 12 Bibliografía.

1. Introducción.

Los videojuegos se encuentran cada vez más presentes en nuestra vida diaria, y en una sociedad orientada al consumo tecnológico y a la transmisión de significados simbólicos a través de Internet esta tendencia parece acrecentarse [5]. Se encuentran presentes en cualquier dispositivo con capacidades multimedia (lectores de libros digitales, celulares, tablets, televisión digital, etc.) [170] y acompañaron el crecimiento de una generación creando una base de consumidores asiduos con años de juego, y con una audiencia diversificada que abarca no solamente a niños, sino también a personas de todas las edades, y hombres y mujeres por igual [46][51].

El desarrollo de videojuegos, por su parte, comprende una actividad multidisciplinaria que está no solamente relacionada con la parte tecnológica sino también con la creación y producción de arte, y de representaciones cargadas de significado. Este carácter multidisciplinario, y la complejidad y los riesgos asociados a la creación de productos culturales hacen particularmente compleja la tarea del desarrollo [93].

La capacidad de los videojuegos de cargar con significados ha producido que protagonice diferentes planteos sobre la violencia y los valores que estos pueden ser capaces de transmitir [11][58]. En respuesta, la industria de videojuegos ha desarrollado mecanismos para proteger a la audiencia de menores de edad basados en sistemas de clasificación que advierte sobre contenido inapropiado [52][53].

Por otro lado, esta misma capacidad de los videojuegos para transmitir ideas también ha comenzado a explorarse para intentar conseguir diversos objetivos sin perder de vista al entretenimiento. Juegos educativos, advergames, y juegos artísticos son algunos de los ejemplos que están tomando importancia en la actualidad [43][170].

En cuanto a los procesos y técnicas de implementación, el desarrollo de videojuegos, toma elementos de distintos sectores del software, la computación gráfica, simulaciones físicas y la Inteligencia Artificial entre otras. Los lenguajes de programación y las plataformas objetivo son variados, pero en general, se busca conseguir el mayor provecho del hardware para ofrecer experiencias audiovisuales interesantes. La reutilización de componentes de software es una de las principales herramientas utilizadas por los desarrolladores para disminuir los tiempos y costos de producción [136][137].

Existe poca bibliografía que intente dar un enfoque general y abarcador acerca del desarrollo de videojuegos, y menos aún en el idioma castellano, de modo que este trabajo intentará cubrir esta falencia e intentará satisfacer las necesidades de información de aquellas personas que tengan interés de comprender o en iniciarse en el desarrollo de videojuegos.

El trabajo se encuentra dividido en cinco partes:

INTRODUCCIÓN Y CONCEPTOS: en esta parte del trabajo se presentan las nociones principales sobre videojuegos. Comprendiendo desde conceptos básicos y clasificaciones de juegos y jugadores, pasando por la historia de los videojuegos, hasta las implicaciones que estos tienen en la sociedad y la cultura.

DESARROLLO: en esta parte se explican las tareas y el proceso de desarrollo de videojuegos. También se explican cuáles son los roles de las entidades y las personas que intervienen así como las metodologías y algunas de las problemáticas propias del desarrollo de videojuegos.

TECNOLOGÍA: esta parte se enfoca, por un lado en caracterizar el hardware y plataformas utilizadas en la actualidad junto con una breve reseña historia; y por otro lado, caracterizar el desarrollo del software de videojuegos, cuales son las técnicas, algoritmos y áreas usualmente aplicadas en videojuegos.

INDUSTRIA: el objetivo de esta parte es caracterizar y describir la industria global de los videojuegos junto a los modelos de negocios más usuales, y además describir brevemente el desarrollo de videojuegos en Argentina.

CONSIDERACIONES FINALES: en esta última parte del trabajo se detallan las consideraciones finales y las ideas que surgen a partir de la escritura de este trabajo acompañado de algunas proyecciones junto con posibles líneas de investigaciones futuras.

2 Videojuegos.

En este segundo capítulo, se dará una descripción de los conceptos principales que rodean el mundo de los videojuegos con el objetivo de establecer un lenguaje común que permita avanzar con el resto del trabajo.

Se darán definiciones de lo que es un juego, que es un videojuego, cuales son las características del mismo, se hará un breve repaso sobre la historia de los videojuegos y su industria y cuales son las disciplinas que el desarrollo de los videojuegos abarca.

2.1 ¿Qué es un juego?

Existen diversas definiciones provenientes de disciplinas distintas acerca del concepto del juego. Se destacarán a continuación las de los autores más importantes para este campo para comenzar el análisis:

Roger Caillois, uno de los pensadores más influyentes de la filosofía del juego, define al juego como una actividad voluntaria e improductiva que es gobernada por reglas que definen a un universo ficticio separado de la realidad del jugador [4].

Para Johan Huizinga, historiador y fisiólogo [8], un juego es una actividad propia de la vida animal, que escapa de la vida ordinaria del jugador y que lo sumerge por propia voluntad a una competencia o una representación de algo sin tener por objetivo de satisfacer una necesidad básica. [5]

Para Jesper Juuls, un teórico influyente del campo del estudio de los videojuegos [7], un juego es un sistema basado en reglas con resultado variable y cuantificable, donde el jugador ejerce esfuerzo por influir en el resultado y se siente emocionalmente ligado al mismo. Esta actividad es opcional y los resultados de la misma pueden ser negociables [6].

Chris Crawford, un diseñador y teórico de los videojuegos describe al juego en cuatro características que complementan la definición de juego [3]. Estas son:

- Representación: La representación se refiere a dos aspectos del juego: uno objetivo, que es el sistema de reglas cerrado por el cual está formado, y otro un aspecto subjetivo, que implica que el juego es una versión simplificada y subjetiva de una realidad emocional.

- Interacción: La actividad de jugar es un proceso activo, y es lo que lo diferencia de otros medios de entretenimiento.
- Conflicto: Surge naturalmente de la interacción del jugador y es una característica intrínseca del juego.
- Seguridad: El jugador está a salvo de las consecuencias del juego y de los conflictos que este presenta. Son una manera segura de experimentar la "realidad".

Los cuatro autores, entonces, coinciden en caracterizar al juego como una actividad:

- Voluntaria: El jugador participa del juego por propia decisión es una actividad fuera de su rutina habitual y completamente opcional.
- Regida por reglas: Las acciones de los jugadores están controladas por un sistema de reglas. Estas reglas pueden ser fijadas anteriormente o surgir espontáneamente del juego (pero siempre deben ser respetadas).
- Sin fines productivos: El juego deja a los jugadores en las mismas condiciones que al inicio del mismo sin haber creado bienes, ni riqueza, ni nuevos elementos de ningún tipo. Esto no quita que la experiencia del juego haya sido enriquecedora a un nivel cognitivo.
- Con un resultado desconocido: Los resultados del juego, si bien están regidos por reglas, no pueden saberse de antemano y dependen de las acciones que tomen los jugadores según las mismas (lo cual podría incluir la interacción con el azar).
- Se desarrolla en un marco ficticio: El juego se desarrolla en un mundo separado de la realidad del jugador. Las reglas del juego no son las mismas que las del mundo real.

2.2 ¿Qué es un videojuego?

Así mismo como en la sección anterior, se presentan los pensamientos de los autores más importantes, esta vez, de una disciplina propia del mundo de los videojuegos, el Game Design.

Según Chris Crawford, los videojuegos representan un subconjunto de la realidad, en donde se coloca al jugador en una situación de conflicto definido por las reglas del juego e interactúa a través de un dispositivo electrónico [3].

Según Andrew Rollings y Ernest Adams, un videojuego es una serie de uno o más desafíos conectados causalmente donde el jugador toma un papel activo dentro de un universo simulado [9].

Según Raph Koster los videojuegos son *puzzles* a resolver que representan patrones abstractos del mundo (reales o no) y se basan principalmente en el conocimiento, el aprendizaje y el análisis de estos patrones por parte del jugador. En consecuencia estos representan herramientas de aprendizaje [1].

Es interesante señalar, en la última definición, la forma en que surge una relación entre los videojuegos y el proceso de aprendizaje. Las definiciones anteriores de juego descritas por Johan, lo sitúan al juego como un proceso natural de la vida animal (no como un invento humano) ya que constituye una forma natural y segura de aprendizaje y por lo tanto es una función vital para cualquier criatura capaz de aprender [3].

Los videojuegos no son la excepción, y esta postura se mantiene en los diversos autores. Más adelante se abordará el tema con más atención.

Un videojuego es, en resumen, un juego utilizado bajo el soporte de un dispositivo electrónico. Este soporte electrónico junto con su sistema operativo, lleva el nombre de plataforma. La lista completa de plataformas existentes hasta ahora alcanza las 117 teniendo en cuenta plataformas antiguas [11]. Sin embargo se considerarán a continuación las más actuales para ilustrar una taxonomía práctica.

- Computadores personales: Son dispositivos de propósito general usualmente con gran poder de cálculo con software y hardware heterogéneo.
 - Microsoft Windows PC. (Por convención se refiere a una PC con sistema operativo Windows en su versión más reciente)
 - Mac.
 - Linux.

- Consolas de sobremesa: son dispositivos diseñados específicamente para juegos, generalmente con hardware de gran capacidad dedicado al procesamiento de gráficos. No tienen incorporados dispositivos de salidas (sin pantalla ni parlantes), y se conectan a un Televisor o Monitor. Obtienen la energía conectándose directamente a un tomacorriente.

- Xbox 360 (Microsoft)
- PS3 (Sony)
- Wii (Nintendo)

- Consolas Portátiles: Son dispositivos electrónicos portátiles con su propia fuente de energía, dispositivos de salida y con una capacidad de procesamiento más reducida que las consolas de sobremesa, son diseñadas específicamente para juegos.

- PSP (Sony)
- 3DS (Nintendo)

- Máquina de Arcade: Consiste en un dispositivo electrónico especializado en la ejecución de juegos. Existen máquinas de arcade específicas para un solo juego. Se caracterizan por poseer un sistema de créditos ya sea por monedas especiales o lectores de tarjetas ya que son utilizadas en lugares públicos con fines comerciales donde se paga por cada jugada.

- Móviles: Teléfonos celulares. Tienen una capacidad de procesamiento y almacenamiento reducida, y sus objetivos primordiales son de comunicación y entretenimiento.

- Iphone (Apple)
- Android (Google)
- J2me(Oracle)

Existen otras plataformas para dispositivos móviles tales como Rim de BlackBerry, Windows Phone de Microsoft y MeeGo (Distribución de Linux para

dispositivos multimedia) pero tienen una extensión y popularidad mucho menor que la mencionadas.

2.3 Reseña Histórica

2.3.1 Primeros experimentos.

Los primeros videojuegos fueron creados en universidades con fines de investigación o por visionarios experimentando con dispositivos electrónicos, y no fue hasta 1971 que llegaron al público con el juego "**Computer Space**". Algunos historiadores de videojuegos no consideran a estos como los primeros videojuegos [12], sin embargo, constituyen un claro antecedente. Entre estos primeros experimentos se encuentran:

- En 1947 con "**Cathode Ray Tube Amusement Device**" desarrollado por Thomas T. Goldsmith Jr. y Estle Ray Mann, donde se controlaba por medio de un circuito analógico el brillo y la posición de un punto con un tubo de rayos catódicos intentando crear un simulador de misiles. Los objetivos debían ser superpuestos dado que no era posible dibujar gráficos todavía.[12]
- En 1952 Alexander S. Douglas creó para su tesis de doctorado en la Universidad de Cambridge una versión del Ta-te-ti programada para el EDSAC (La primer computadora con posibilidad de almacenar programas [13]). Éste fue el primer videojuego en utilizar una pantalla grafica digital y al mismo tiempo el primer programa de juegos del mundo almacenado en una computadora.
- En 1958 William Higinbotham creó "**Tennis for Two**" como un experimento en el laboratorio de Brookhaven. El objetivo del experimento era poder demostrar que era posible el control interactivo en pantalla. El juego fue programado sobre un osciloscopio [11][14].

2.3.2 Primeros videojuegos y el nacimiento de la Industria.

"**Computer Space**" es considerado uno de los primeros videojuegos, inicialmente fue desarrollado por estudiantes del MIT bajo el nombre de "**Space War**" para la computadora DEC PDP-1 del instituto tecnológico de Massachusetts.

La versión arcade de este juego ("**Computer Space**") fue lanzada en 1972 siendo considerada la primer máquina de arcade con ranura para monedas (Siguiendo con la tradición de otros juegos electrónicos basados en el sistema de créditos como por ejemplo los pinballs) y al mismo tiempo el primer videojuego con fines comerciales [11].

El juego "**Table Tennis**" surge como una adaptación de "**Tennis for two**" para la primer consola hogareña la "**Magnavox Odyssey**" en 1972. Esta consola sentó precedentes y era un híbrido entre circuitos analógicos y digitales y usaba un soporte de cartucho para almacenar los juegos vendiendo cerca de 300.000 unidades [15].

En ese mismo año Atari confeccionó una maquina de arcade con el juego "**Pong**", una versión arcade del juego de la consola "**Magnavox Odyssey**". Magnavox denunció legalmente a Atari por copiar la idea de su juego a lo que le siguieron una serie de batallas legales que terminaron en un acuerdo extrajudicial. Atari acordó pagarle 800.000 dólares a Magnavox por la licencia exclusiva del juego y el juego "**Pong**" se convirtió en el primero en alcanzar una popularidad con el público [11].

Este comportamiento acompaña a la industria de los videojuegos desde el inicio hasta la actualidad. Cada nuevo juego generado por la industria era copiado entre las empresas bajo diferentes nombres y diferente hardware.

Varios años más tarde en 1977 se lanzó la Atari 2600, la primer consola que alcanzó un éxito considerable en cuanto a ventas, popularidad y distribución. La Atari 2600 popularizó el concepto de un microprocesador dedicado y el sistema de almacenamiento de juegos por cartuchos y fue la primera consola de la segunda generación.

2.3.3 "La época dorada".

La consola Atari 2600 no representó un éxito inmediato [14], y no fue hasta 1979 que con juegos como "**Space Invaders**", "**Missile Command**" y "**Pac-Man**" logró retribuirle a Atari un negocio de 2.000.000 de dólares. [16]

En 1979 la industria de las máquinas de arcade se había consolidado por completo, y había desplazado a las máquinas de pinballs del negocio del entretenimiento interactivo [11].

Los juegos, tanto de consola como de arcade, comenzaron a mejorar técnicamente. Los juegos a colores propios de la consola Atari también se popularizaron en las máquinas de arcade, al mismo tiempo que resurgieron los juegos con gráficos basados en vectores, como por ejemplo el juego “**Asteroids**” de Atari [11]. Los juegos comenzaron a diversificarse en contenido. Surgieron otros grandes clásicos como “**Breakout**”, “**Centipede**”, “**Gauntlet**” y “**Steel Talons**”.

Atari obtenía los grandes éxitos para la consola, de sus propios juegos de arcade. En ese mismo período las máquinas de Arcade de Midway se hicieron populares. La presión dentro de Atari para los desarrolladores crecía, y el descontento de un grupo de empleados que se separaron, terminó por formar la empresa Activision. Activision se convirtió de esta manera en la primera empresa en hacer juegos para terceros que produjo juegos para consolas. Tras su éxito varias compañías más le sucedieron. [11]

También se crearon otras consolas hogareñas para competir con la Atari. Éstas fueron la Coleco Vision (1982) y la Intellivision (1979) de Mattel. Coleco Vision licenció el juego “**Donkey Kong**” a Nintendo, el cual fue su mayor éxito. Considerando la competencia Atari lanzó una versión mejorada de su consola, la Atari 5200, lanzada en 1982.[14]

Hacia el año 1982 la mayoría de los géneros de videojuegos fueron creados para distintas versiones de consolas y arcades.

En cuanto a los juegos para las computadoras personales de la época como la Comodore 64, Amiga, ZX Spectrum, Apple II , al inicio la distribución de los juegos estuvo a cargo de aficionados que clonaban los juegos más exitosos de los arcades como “**Donkey Kong**”, “**Pac Man**”, “**Frogger**” y “**Space invaders**”. Revistas especializadas comenzaron a formarse y tomaron el papel de distribuir estos códigos fuentes. Con el tiempo surgieron las empresas distribuidoras que introdujeron los primeros juegos comerciales para PC con títulos originales como: “**Zork**”, “**Roberta Williams' Mystery House**” para Apple II y “**King's Quest**” de la empresa Sierra [17].

En cuanto a consolas de videojuegos portátiles existían los denominados “**Handled electronic Games**”. Estos consisten en versiones reducidas de consolas que pueden ser llevadas en el bolsillo. Los “**Handled electronic Games**” de esta época consistían en juegos muy pequeños, cuyas consolas eran diseñadas específicamente para cada juego y con el display generalmente hecho a medida. El más famoso de ellos es la

colección “**Game and Watch**” de Nintendo con juegos como “**Donkey Kong**”, “**Egg**” y “**Ballon Figh**”.

2.3.4 El crash de la de industria.

El plan de Atari era ganar dinero a través de la venta de los cartuchos de juegos, y no a partir de la venta de la consola misma. No obstante la decadencia en calidad de sus juegos frente a la competencia y a las nuevas empresas distribuidoras quienes producían juegos de una calidad superior, terminaron por llevar al fracaso esta estrategia.

El mercado de videojuegos había sido inundado con muchos juegos de poca calidad, adaptaciones y secuelas de otros juegos viejos pero creados con presupuestos reducidos. Los consumidores empezaron a ser decepcionados por los productos.

Se desató un efecto en cadena que afectó a toda la industria, Atari comenzó a perder dinero, Mattel decidió retirarse de la industria de juegos para siempre terminando con su consola Intellivision. En total la industria de los videojuegos paso a tener una ganancia de 2.9 miles de millones de dólares en 1983, es decir un 35 por ciento menos que el año anterior. En total la industria había perdido cerca de 1.5 miles de millones de dólares [11]

2.3.5 El resurgimiento de la industria y la era de los 8 bits.

El crash terminó cuando el éxito de Japón, la Nintendo Entertainment System (NES), llego para revitalizar a la industria. La NES fue una consola con un procesador de 8 bits con sistema de almacenamiento de cartuchos.[19]

Alcanzó un éxito inmediato vendiendo cerca de 60 millones de unidades [18], con juegos populares como “**Super Mario Bross**”, “**The legend of Zelda**” y “**Metroid**”.

Nintendo había aprendido del “crash” que le antecedió, y había desarrollado una política estricta para el licenciamiento de los juegos y de esta manera se aseguraba que la consola no sería invadida con juegos de calidad cuestionable. Los juegos sin licencia de Nintendo no serían tolerados por la consola. La licencia de Nintendo limitaba la cantidad de juegos que una empresa de terceros podía crear por año y

además, los mismos juegos no podían ser hechos para otras consola hasta pasado un periodo de 2 años. [19][11]

En cuanto a los videojuegos para PC se estaba dando a conocer lo que se llama el “Shareware”. El shareware consiste en una licencia de software que entrega una versión reducida del programa en forma gratuita, e invita a los usuarios a comprar la versión completa. [17]

Los juegos online también comenzaron a ganar un poco de terreno gracias a Internet aunque se seguía manteniendo como un terreno poco explorado. Estos juegos eran generalmente juegos basados en texto [17]

2.3.6 El período 1990-1995 y las consolas de 16 bits.

En 1989 se lanzó el Game Boy, un sistema portátil de juego con una pantalla LCD con juegos almacenados en cartuchos [14] Su popularidad fue muy grande, y logró vender un total de 30.3 Millones de unidades. Sus títulos más conocidos son “**Tetris**” (Nintendo negoció la Licencia con la Unión Soviética para poder distribuirlo legalmente con su consola), y “**Pokemon Blue**” y “**Pokemon Red**” dos juegos que impactaron en la cultura Japonesa, llegando a producirse series animadas, películas, juegos de cartas y una infinidad de productos como muñecos y ropa [20].

La primera consola de 16 bits fue de la empresa NEC, llamada TurboGrafx 16 cuya sucesora fue la primera en utilizar CD-ROMs. Sega, con una mejor estrategia comercial liberada en 1989 comenzó a ganar popularidad. La competencia con la NES de Nintendo fue creciendo hasta superar a esta última. Es por eso que en 1991 Nintendo lanzó la Super Nintendo Entertainment System (SNES) con un procesador de 16 bits que seguía utilizando cartuchos.

Sega CD fue lanzada en 1992 (Mega drive en Japón lanzada en 1991) con una gran capacidad de almacenamiento para los juegos, 2.000 veces superior a la de un cartucho gracias al uso de los CD-ROMs.

En esa misma época surgió la máquina de arcade SNK. Una máquina de 21 bits que superaba ampliamente en poder de procesamiento y gráficos a las consolas de la época. Su versión de consola no tardó en salir, y fue llamada Neo Geo. Si bien los

juegos lanzados para esta consola fueron muy populares, el alto precio impidió que se convirtiera en un éxito comercial [1].

Aún así, con la llegada de las consolas de 16 bits la diferencia con la calidad de los juegos con las máquinas de arcade fue disminuyendo y éstas últimas comenzaron a decaer en ventas, generando cada vez menos ganancias. El último gran éxito de los arcade fue el **Street Fighter I** un juego de pelea para dos jugadores.

En PC la capacidad de procesamiento había crecido notablemente, en el mercado se contaba con los procesadores Intel 80386, 80486 y Motorola 68000, 68030. Al mismo tiempo se vislumbró el nacimiento del 3D, y las capacidades multimedia con las tarjetas de sonido y los CD-Roms.

La distribución de los juegos se realizó bajo la licencia de shareware, el cual se volvió usual gracias a que los juegos eran desarrollados por pequeños equipos como Apogee (ahora 3D Realms) Epic Megagames (Epicgames) e ID software. Dado que el tamaño de los juegos creció, se fue convirtiendo impracticable utilizar disquetes para distribuir los juegos y se comenzaron a utilizar los CD-Roms. Las revistas especializadas distribuían CDs llenas de versiones Shareware de los juegos.

Surgieron juegos como "**Wolfenstein 3D**" y "**Doom**" los cuales fueron los primeros juegos del género en primera persona modernos (FPS por sus iniciales en inglés). Los FPS son un género muy popular de videojuegos en la actualidad.

En 1992 se liberó el juego "**Dune II**" que fue el primer juego de estrategia moderno (RTS por las siglas en inglés de Real Time Strategy). Su formato marco tendencia en los RTS que le siguieron como "**Command and Conquer**", "**Warcraft**" de Blizzard y "**Age of Empires**" de Microsoft Games.

En 1990 Maxis, comenzó a publicar su línea exitosa de juegos: "**Sim**". Comenzando con "**SimCity**", "**Sim Earth**" y "**SimCity 2000**".

Muchos juegos fueron lanzados con la opción de juego online, juegos como "**Age of Empires**", "**Starcraft**", y "**Ultima Online**" [17].

2.3.7 El periodo 1995-2000 y las consolas de 32 y 64 bits.

En 1996 se lanzó la 3dfx Voodoo Chipset, la primera tarjeta aceleradora de gráficos para computadoras personales. Este procesador permitía mejores gráficos con resoluciones mayores gracias a un procesador y memorias dedicadas para el procesamiento de gráficos en la tarjeta aceleradora [21]. Juegos como “**Quake**” fueron los primeros en hacer uso de estas capacidades mostrando gráficos totalmente en 3D.

La popularidad de los juegos Online creció, juegos como “**EverQuest**” permitían jugar con un número limitado de jugadores al mismo tiempo bajo el nuevo concepto de mundos persistentes para juegos multiplayer gráficos.

Más adelante el elemento social se hizo notar aún más con los juegos online. Los juegos “**Ultima Online**”, “**Asheron’s Call**”, “**Star Wars Galaxies**”, y “**World of Warcraft**” alcanzaban los millones de jugadores alrededor del mundo.

El desarrollo de los plugin de Flash y de Java para los exploradores web abrió paso a juegos simples ejecutados directamente en el explorador. Estos juegos no requieren de instalación y se pueden jugar sin representar riesgos de seguridad a los jugadores ya que son ejecutados en máquinas virtuales.

Surgió una nueva modalidad en los juegos y esta fue la posibilidad de hacerles modificaciones. El juego más exitoso con esta característica fue “**Half Life**”. Uno de sus Mods (modificaciones) más populares fue el “**Counter Strike**” un juego multiplayer de gran popularidad. [17]

Este período se caracterizó por la popularización de los juegos 3D y el soporte de CD-ROMs para los juegos, y esto se dio tanto en computadores personales, como en consolas y arcades.

Nintendo y Sony comenzaron negociaciones para desarrollar un accesorio para la SNES que le permitiera leer CD-ROMs. Sin embargo, Sony consideró que la SNES era tecnología obsoleta, y diseñó una consola nueva por completo llamada PlayStation. Esta consola tendría una ranura para los cartuchos de SNES y podrían ser jugados en ella los juegos diseñados para SNES CD. Esto le daría control completo a Sony de los títulos que podrían ser desarrollados con los CD-ROMs, al contrario de lo buscaba Nintendo con su consola. Al comprender esto el presidente de Nintendo, Hiroshi Yamauchi, dio fin a las negociaciones.

Nintendo volvió a intentarlo con la empresa Philips pero aconteció una historia similar. Philips desarrolló su propia consola, la CDTV, la cual terminó en un fracaso comercial.

Finalmente, Nintendo desistió de agregar una lectora de CD-ROMs a la consola SNES al ver el poco éxito conseguido por Sega CD y terminó por desarrollar la Nintendo 64, una consola de 64 bits con poco éxito ya que seguía utilizando cartuchos, cuyos precios resultaban muchísimo más caros que los de los almacenados en CD-ROMs. No obstante juegos como "**Super Mario 64**" marcaron un nuevo rumbo en el género de juegos de plataformas en el mundo 3D, así también como otros juegos recordados de la consola [11]

Por su lado la Playstation de Sony lanzada en 1994 tuvo un gran éxito comercial. Mundialmente se distribuyeron más 100 Millones de unidades [22] convirtiéndose en la primera consola en vender tal cantidad. La Playstation contaba con un procesador 32 bits y 2 MB de memoria RAM de propósito general, 1MB de memoria RAM para video, y 512 KB de RAM para sonido. Utilizaba soporte de CD para sus juegos y una memoria especial para poder almacenar las partidas llamada Playstation Memory card.

En PC, se desarrollaron múltiples placas aceleradoras cuyos mayores exponentes fueron las series GeForce de Nvidia, y las placas aceleradoras de la serie Raedon de ATI [25] Cada nuevo juego en PC requería de un poder de procesamiento mayor, lo que empujó a una relación entre fabricantes de hardware y las empresas desarrolladoras de juegos.

Game Boy Color de Nintendo, en el marco de las consolas portátiles, fue lanzada 1998 con 118 millones de unidades vendidas y destacándose nuevamente juegos de la serie Pokemon. Tenía un procesador de 8 bits, y utilizaba soporte de cartuchos para sus juegos.

Neo Geo Pocket fue lanzada en ese mismo año, repitiendo el error de sucesora en consola de sobremesa. Si bien, la Neo Geo Poket, era superior técnicamente a su contraparte de Nintendo, su precio era muy elevado y solo alcanzó a vender 2 millones de unidades.

2.3.8 El periodo 2000-2005 y la sexta Generación de consolas.

En el año 2000, Sony libera su segundo gran éxito, la Playstation 2 (PS2), la cual alcanzó un nuevo récord de ventas llegando a vender 100 millones de unidades solamente a 5 años de su lanzamiento [25] y un total de 150 millones de unidades hasta la fecha [26]. Pese a ser una consola de más de 10 años de antigüedad y de competir con su propia sucesora, se continua fabricando.

La PS2 fue la primer consola en utilizar soporte de DVD para sus juegos, y también era retrocompatible con la Playstation (Se pueden jugar juegos de su predecesora). Los juegos más vendidos para ésta consola son “**God of War**,” “**Gran Turismo**,” “**SingStar**” .“**Ratchet & Clank**” y “**Grand Theft Auto: San Andreas**” [26]. La elección de Sony por el soporte de DVD estuvo fundada por dos razones: los DVD podían almacenar seis veces más información que los CDs, y además le permitiría a la consola reproducir películas, algo que ninguna otra consola podía hacer [11]. La PS2 contaba con un procesador de 128 bits de 294 MHz, un procesador gráfico de 147 MHz, y con una memoria RAM de 32 MB.

Nintendo lanzó su primera consola no basada en cartuchos denominada Nintendo Game Cube. Los juegos de la Game Cube utilizaban un soporte de almacenamiento propietario que podía almacenar 4 o 5 veces la capacidad de un DVD. Sin embargo Nintendo solamente logró alcanzar la venta de 21 millones de unidades.

En el 2001, Microsoft hace su aparición en el mercado de las consolas con la Xbox. La Xbox tenía un procesador Intel Pentium III de 733 MHz, un procesador gráfico de Nvidia llamado NV2 con 250 MHz de procesador y 64 MB de RAM, soporte para conexión a Internet y un disco de 8GB. Su mayor éxito fue “**Halo 2**” que vendió 8 millones de copias, mientras que la consola misma, solamente vendió 28 Millones de unidades [11].

Si bien la consola no fue del todo un éxito comercial, introdujo el concepto de juego online en consolas con el sistema por suscripción de Xbox Live. El Xbox live permitía a los jugadores competir y compartir jugadas con personas a través del mundo. Sony lanzó su propio sistema de juego online, pero a diferencia de Microsoft, el servicio era gratuito una vez comprado el accesorio que permitía a la PS2 la conexión a Internet [11].

En PC los juegos multijugadores online se hicieron aún más populares. En general, la mayoría de los juegos incluía ahora un modo de juego multiplayer para jugar en línea.

El juego “**World of Warcraft**” con millones de jugadores representa el 60% del mercado de suscripciones online de juegos.

En el 2003 el juego “**Second Life**” hace su aparición. Es esencialmente un mundo virtual persistente donde los jugadores pueden crear contenido y son los propietarios de los derechos de propiedad intelectual de estas creaciones. El juego cuenta con un sistema de economía propia basada en la moneda Linden dólar el cual puede ser utilizado para comprar tierras y bienes virtuales.

En el año 2000 Maxis lanza el juego “**The Sims**” y se convierte en un éxito instantáneo y uno de los juegos de computadora mejores vendidos de todos los tiempos [17]. Otros juegos como “**Doom 3**” y “**Half Life 2**” hacen su aparición como secuelas de juegos anteriormente exitosos.

2.3.9 2005-Actualidad: La séptima y octava generación de consolas.

En el 2005, Microsoft lanza la sucesora de la Xbox, la Xbox 360. Ésta es la primera consola de la séptima generación, con un procesador gráfico más potente y más espacio en disco. Dicho espacio en disco fue necesario ya que las funciones de su sistema online no solamente se limita a conectar jugadores y partidas, sino también con el lanzamiento del Xbox Live Arcade (XBLA) le permite la compra de juegos de manera online. También fue la primera consola en poseer salida HDMI para ser utilizada en televisores de alta definición. Esto generó el descontento de algunos usuarios, ya que solamente las mejoras gráficas de la consola podían ser aprovechadas al máximo teniendo un televisor con esta característica. La Xbox 360 alcanza hasta la fecha un total de 53 millones de unidades vendidas desde su aparición.

En el 2006 Sony lanza su tercera consola de sobremesa, la Playstation 3 con características muy similares a la Xbox 360, y alcanzando hasta la fecha un total de 57 millones de unidades vendidas.

En ese mismo año Nintendo lanza su nueva consola, esta vez apuntando a una dirección distinta. El nombre de esta consola es Wii, que en ingles tiene la misma pronunciación que la palabra “We” (nosotros). La característica principal de la Wii no

reside en su capacidad de procesamiento ni en el desempeño gráfico sino en su forma innovadora de control. Sus controles inalámbricos llamados Wii Remote interpretan la aceleración de los movimientos en tres dimensiones, permitiendo utilizar el control de diversas formas. La consola alcanza un total de 83 millones de unidades vendidas, superando a sus competidores.

El Wii Remote no fue el único cambio que introdujo Nintendo con la Wii, sino el hecho de apuntar a un público más casual y no tanto a los jugadores hardcore(Este tema se explicará con mayor detenimiento más adelante)[36]. Esta fue la estrategia que le permitió a la Wii destacarse por sobre sus competidores, quienes no tardaron en imitarlo.

En el 2010 Microsoft lanza Kinectic, un accesorio para la Xbox 360 que incorpora tecnología de visión artificial y el procesamiento de imagen y sonido, utilizando las cámaras incluidas en el accesorio para convertir al jugador en el control del juego.[27]

Sony por su parte, lanza un híbrido entre la tecnología utilizada en el Wii Remote y el Kinectic de Microsoft llamado PS Move y también desarrolla un catálogo de juegos orientado a los jugadores casuales.

Al igual que la Xbox 360, las consolas Wii y PlayStation 3 cuentan con su propio sistema online de ventas de juegos descargables. WiiWare Para Wii, Playstation Network para Playstation 3.

El mercado de consolas portátiles se ve caracterizado por las consolas de Nintendo y Sony. Al igual que con la Wii la consola portátil de Nintendo, la Nintendo DS (NDS), busca la innovación en los controles y se presenta como una consola portátil con doble pantalla, una de las cuales es táctil. Sony por su parte lanza la Playstation Portátil (PSP) superior en cuanto capacidad de procesamiento y almacenamiento de su competidora [11].

En cuanto a PC grandes secuelas vuelven a hacer su aparición. “**Half Life 2: Episodio I**”, “**Half Life 2: Episodio II**” y “**StarCraft II**” bajo las expectativas de millones de jugadores. El concepto de pre-orden comienza a hacerse presente al mismo tiempo que se hace más usual la distribución digital. La pre-orden consiste en la compra anticipada del producto antes de que este sea lanzado obteniendo como beneficio para el comprador descuentos.

La plataforma de distribución Steam toma un papel importante en la distribución digital de los juegos en PC, y se convierte en la plataforma online de juegos más grande del mundo. Este servicio ofrece una gran cantidad de títulos tanto a usuarios de Mac como de PC, y contiene una base de usuarios activos de 30 millones [29].

El mercado de juegos online, protagonizado por los juegos Flash, basan su rentabilidad en la publicidad. Según, Mochi Ads, una red de avisos publicitarios que conecta a desarrolladores independientes con anunciantes, un tercio de los usuarios de Internet visitan los sitios de juegos online (llamados portales) lo que representa una audiencia creciente de 237 millones de personas en el mundo entero [30].

Los juegos casuales, tanto en consolas, como en PC y juegos online, toman importancia en el mercado como resultado de una búsqueda de expandir los videojuegos hacia nuevos horizontes. El público de los videojuegos se amplía, y en el caso de los jugadores casuales son caracterizados en cuanto a edad, un 65% mayores de 35 años, y en cuanto a género un 29% son hombres y el otro 71% son mujeres. Por el contrario, los jugadores en general de videojuegos tienen una edad promedio de 33 años con un 62% de hombres y un 38% de mujeres. [31] Mas adelante se abordará este tema con más detalle.

En febrero de 2011 Nintendo lanzó la consola portátil 3DS (La última consola lanzada hasta la fecha de la construcción de este trabajo). Entre sus características se encuentran: una placa WIFI para permitir conectar la consola a internet, acelerómetros para detectar los movimientos y la inclinación de la consola, y 2 cámaras (para lograr el efecto 3D con fotografías). La novedad de esta consola radica en la pantalla superior, que posee la cualidad de mostrar imágenes con efecto estereoscópico, también conocidas como imágenes 3D, sin la necesidad de utilizar lentes. [31]

3. Taxonomías.

A continuación se presentan las clasificaciones utilizadas por la bibliografía, y la industria para diferencias diferentes tipo de videojuegos y de jugadores.

3.1 Clasificación de los videojuegos.

Lograr una taxonomía es útil para los diferentes aspectos de los videojuegos: en estudios teóricos para facilitar la investigación, la búsqueda de ejemplos y referencias; en materia de *game design* para encontrar técnicas comunes de diseño entre juegos de un mismo tipo; en el desarrollo de herramientas para encontrar los problemas comunes de implementación, y desarrollar herramientas más precisas que faciliten el desarrollo de los juegos y reduzcan los errores y tiempos de producción; y desde un punto de vista comercial, para ayudar a los consumidores a encontrar los juegos que coinciden con sus gustos personales, y a las empresas distribuidoras y desarrolladoras para aprender de los gustos de sus consumidores.

Sin embargo, no existe una taxonomía universalmente aceptada, y resulta difícil conseguirla ya que esta disciplina es todavía muy joven y en continuo crecimiento [3]. A diferencia de lo que sucede con otros medios de entretenimiento como los libros, y películas los cuales se clasifican por los temas que tratan, los videojuegos utilizan la el *gameplay* (la forma en que se juega) para dividir los géneros [39].

Existen otras características por las cuales se pueden clasificar los juegos:

- El tema: Se refiere a la ambientación, la historia, y el contenido cultural del juego. Por ejemplo: un juego de acción en primera persona con tema de “Western” es el juego Gun (Neversoft).
- La audiencia: Se refiere al público al cual está destinado el juego. La clasificación más común son los juegos para jugadores “core” y los juegos para jugadores casuales. Se retomará este tema más adelante.
- El propósito: Se refiere al propósito de los creadores del juego. Este propósito puede ser solamente entretener, obtener dinero, resolver un problema, o transmitir un mensaje entre otros. Los juegos con el propósito de

resolver un problema son llamados juegos serios, e incluyen desde juegos para publicidad hasta juegos educativos.

A continuación, se detallan algunas de las categorías utilizadas por tiendas online, portales y organismos, para clasificar a sus juegos con el objetivo de ilustrar qué categorías existen desde la perspectiva comercial.

Entre los analizados se encuentran: Amazon, una tienda Online de todo tipo de productos como libros, juegos, música y equipos electrónicos; Steam, la plataforma de juegos online más grande del mundo; GameStop, un sitio popular de prensa y *reviews* (reseñas y críticas sobre juegos); Kongregate, un portal de juegos online; y por último ESA la asociación del software de entretenimiento de Estados Unidos que tiene el objetivo de satisfacer las necesidades de la industria y del público, a través de investigaciones y diferentes programas institucionales.

Categorías de juegos en Amazon.	
Casual Games.	Core Games.
Acción.	Acción.
Aventura.	Aventura.
Arcade.	Carreras.
Juegos de Tableros.	RPG.
Juegos de Cartas.	Simulación.
Puzzles.	Deportes.
RPG.	
Buscar y Encontrar.	
Carreras.	
Simulación.	
Estrategia.	
Deportes.	
Trivias.	
Palabras.	

Tabla 3.1.1 Categorías de juegos en Amazon.

Categorías de juegos en Steam.
Aventura.
Estrategia.
Rol.
Independiente.
Multijugador masivo.
Casual.
Juegos familiares.
Simuladores.
Carreras.
Deportes.

Tabla 3.1.2 Categorías de juegos en Steam.

Categorías de juegos en GameStop.
Acción.
Estrategia.
RPG.
Juegos de Manejo.
Aventura.
Simulaciones.
Puzzles.
Juegos para chicos.

Tabla 3.1.3 Categorías de juegos en GameStop.

Categorías de juegos en Kongregate.
Acción.
Multijugador.
Aventuras y RPG.
Shooter.
MMO.
Estrategia y defensa.
Puzzles.
Musicales.

Tabla 3.1.4 Categorías de juegos en GameSpot.

Categorías de juegos según ESA.
Estrategia.
Deportes.
Shooter.
RPG.
Carreras.
Acción.
Arcade.
Juegos para chicos.
Entretenimiento familiar.
Peleas.
Vuelo.

Tabla 3.1.5 Categorías de juegos según ESA.

Existen categorías en común tales como: acción, aventuras, RPG, estrategia, puzzles, simuladores y deportes. Aunque también hacen aparición los juegos casuales (tanto en Amazon como en Steam), y solamente en Steam el concepto de juegos independientes.

En cuanto a la bibliografía, los autores reconocen esta inconsistencia [3] e individualmente ofrecen su propia taxonomía [3] [9] [35] [33][38] No obstante las elecciones para definir los géneros son casi arbitrarias dependiendo del contexto [9]. Aún si se logrará una clasificación universal ésta sería transitoria ya que los juegos constantemente se están redefiniendo y generando nuevas tendencias y géneros [37]. Además, un juego no siempre puede clasificarse dentro de un solo género, y suele representarse como una combinación de 2 o más.

Para este trabajo, se consideran las categorías definidas por los autores Chris Crawford [3], Andrew Rollings y Ernest Adams[9] y Bob Bates [38], presentándolas en forma combinada.

3.2.1 Juegos de Acción y habilidad.

Es uno de los géneros más populares y uno de los más viejos iniciándose con las primeras máquinas de arcade [3][35]. Su principal característica es la de requerir coordinación “mano-ojo” y un buen tiempo de reacción. En este tipo de juegos, se hace énfasis en la parte gráfica y sonora, mientras que durante el transcurso del juego se prioriza la acción sobre todas las cosas (por ejemplo, por sobre la historia) [9].

Los llamados juegos FPS, uno de sus subgéneros, se caracterizan por presentarle al jugador una vista en primera persona del mundo del juego (Ver figura 3.2.1.1). Se centran principalmente en el combate a distancia con armas de fuego, aunque existen algunos juegos con armas de ataque cuerpo a cuerpo. Para algunos autores como Andrew Rollings y Ernest Adams este no es realmente un subgénero de los juegos de acción ya que la característica principal que lo identifica en el género es la representación en primera persona. Sin embargo, el termino FPS es muy utilizado tanto comercial como popularmente, por lo que no se podría haber omitido.

Algunos juegos representativos de este género: **Wolfenstein 3** (ID Software), **Doom** (ID Software), **Half-Life** (Valve), **F.E.A.R** (Monolith Productions), **Crysis** (Crytek) y **Halo** (Bungie, Microsoft).



Figura 3.2.1.1 Half Life 2 (Valve)

Los juegos de pelea constituyen otro subgénero de los juegos de acción, y se caracterizan por centrarse en combates uno contra uno, ya sea con otro jugador o contra una inteligencia artificial diseñada para el juego. La jugabilidad se basa en un sistema de combos (conjunto consecutivos de acciones), y el componente “piedra, papel o tijera” [9] donde para cada acción existe otra que la supera. Si bien existen en representación 2D y 3D, la vista es primordialmente de lado (ver figura 3.2.1.2).

El detalle y el realismo de los personajes, y de las interacciones entre ellos, es donde se hace énfasis en este tipo de juegos. Algunos juegos representativos del subgénero son: **Mortal Kombat** (Midway), **Street Fighter** (Capcom) y **Tekken** (Namco). [9]



Figura 3.2.1.2 Mortal Kombat (Midway).

Los juegos de plataformas, son aquellos en los cuales el protagonista debe correr y saltar entre superficies, esquivando objetos y siendo afectado por la gravedad. Tradicionalmente estos juegos fueron con vistas de lado y en una perspectiva de 2D (ver figura 3.2.1.3), sin embargo con las nuevas tecnologías el género se ha adaptado también a representaciones 3D (ver Figura 3.2.1.3.4).

Algunos juegos notables de este subgénero: **Super Mario Bros** (Nintendo), **Sonic** (Sega), **Príncipe de Persia** (Broderbund, Red Orb, Ubisoft, Pipeworks, Gameloft) y **Super Mario 64** (Nintendo).



Figura 3.2.1.3 Super Mario World (Nintendo).



Figura 3.2.1.4 Super Mario 64 (Nintendo).

Los juegos del subgénero “*Shoot ‘em up*” (Dispárale a todos) son juegos que se basan en destruir múltiples enemigos a través de disparos. La jugabilidad se basa en el reconocimiento de patrones de movimientos, disparos y orden en que aparecen estos enemigos. Algunos juegos utilizan vista de lado, tal como los juegos de plataformas y otros utilizan una vista llamada cenital (Ver figura 3.2.1.5) [9].

Algunos juegos representativos de este subgénero son: **Asteroids** (Atari), **R-Type** (Irem), **Space Invaders** (Taito) y **Geometry Wars** (Bizarre Creations).



3.2.1.5 Asteroids (Atari)

Otro subgénero de los juegos de acción son los juegos de *Maze* o Laberintos, donde la característica particular en estos juegos es que el jugador debe recorrer los diferentes senderos que propone el juego en busca de ítems. Ejemplos de este subgénero son: **Pacman** (Namco) y **Kroz** (Apogee Software). [3]

Por último, los juegos del subgénero “*Paddle Games*” son aquellos en que la jugabilidad se basa en interceptar un proyectil con una pieza controlada por el jugador la cual desvía su trayectoria o la crea (Ver figura 3.2.1.6). Algunos ejemplos de este tipo de juegos son: **Pong** (Atari), **Peegle** (Popcap) y **Arkanoid** (Taito).



Figura 3.2.1.6 Peegle (PopCap).

3.2.2 Juegos de Estrategia.

Los juegos de estrategia se distinguen de los demás por priorizar el pensamiento y la planeación [35]. En general este tipo de juegos requiere de un tiempo mayor para el desarrollo de las partidas y la jugabilidad gira en torno a la administración de recursos,

la producción y la conquista de territorios. Existen dos grandes variantes en los juegos de estrategia: los juegos de estrategia en tiempo real (RTS por sus siglas en ingles) y los juegos de estrategia por turnos.[9]

En los RTS, al ocurrir el juego en tiempo real, la velocidad y el tiempo de reacción toman importancia e influyen sobre las estrategias de juego. Ejemplos de este género son: **Age of Empires** (Microsoft), **Warcraft** (Blizzard), **StarCraft** (Blizzard) y **Command and Conquer** (Electronic Arts).

En los juegos de estrategia por turnos las acciones se encuentran regladas, como el nombre lo indica, por turnos. Cada jugador solo puede actuar en su turno y al realizar la jugada (un conjunto finito de acciones) y debe esperar la jugada de su oponente (que puede ser otro jugador o la IA del juego) para poder continuar. Ejemplos de este tipo juego son: **Civilization** (Sid Meier), **Heroes of Might and Magic** (New World Computing, Nival Interactive, Black Hole Entertainment) y **Age of Wonders** (Epic Games)

Usualmente los juegos de estrategia suelen utilizar una perspectiva que le permite al jugador, ver y manipular múltiples objetos a la vez (Ver figura 3.2.2.1).



Figura 3.2.2.1 Age of Empires III.

3.2.3 Juegos de Aventura

El género de Aventura se caracteriza por priorizar la narración de una historia, donde el jugador toma el papel de uno de sus personajes. En general estos juegos tienden a ser lineales (una sola solución/desenlace) y la jugabilidad se basa en la exploración

del mundo para obtener ítems e herramientas que le posibilitará alcanzar nuevas áreas del juego y a resolver puzzles simples [3].

Inicialmente estos juegos comenzaron como aventuras de texto, pero a medida que la capacidad de procesamiento creció fue posible la introducción de gráficos que cambiaron la jugabilidad haciéndolos más populares. [9]

Dentro del género de aventura se pueden destacar dos subgéneros importantes: las aventuras gráficas o de “*point and click*” (apuntar y clickear), y los juegos de acción aventura.

Los juegos de acción aventura, como su nombre lo indican, mezclan elementos clásicos de los juegos de acción, como la necesidad de una buena coordinación “mano-ojo” con la narración de una historia y más importante aún, la noción de un mundo para explorar. Ejemplos de este subgénero son: ***The Legend of Zelda*** (Nintendo), ***Metroid*** (Nintendo), ***Soul Reaver*** (Eidos) y ***Tomb Raider*** (Eidos). No existe una perspectiva predominante de este género, pero en las representaciones 3D suele utilizarse la denominada vista de tercera persona (ver figura 3.2.3.1).



Figura 3.2.3.1. Legacy of Kain: Soul Reaver.

En el subgénero de las aventuras gráficas, se priorizan la resolución de puzzles, la búsqueda de ítems, los diálogos entre personajes y la historia. Algunos ejemplos de este tipo de juegos son: ***Monkey Island*** (Lucas Arts), ***Myst*** (Cyan), ***Maniac Mansion*** (Lucas Arts), ***Broken Sword*** (Revolution Software) y ***Machinarium*** (Amanita Design).

El juego ***Myst*** por su parte fue uno de los más juegos más populares para PC hasta el año 2002 con 12 millones de copias vendidas.

Estos suelen jugarse utilizando solamente el *mouse* para guiar al personaje, recoger ítems e interactuar con el mundo de juego. También es usual el concepto de inventario donde se almacenan los objetos que el jugador recoge (Ver figura 1.4.3.2).



Figura 3.2.3.2 The Secret of Monkey Island (Lucas Arts)

3.2.4 Juegos RPG (Role Playing Games).

Los juegos RPG, por las siglas en inglés de *Role playing games*, surgen de sus antecesores directos de lápiz y papel [3] Estos juegos constaban de un conjunto de reglas propuestas por el denominado Maestro del calabozo (Dungeon master en inglés) quien controlaba las acciones y conflictos que debían sortear los jugadores sumergidos en un mundo generalmente fantástico medieval.

Los videojuegos este género se caracterizan por ofrecerle al jugador la opción de personalizar a su personaje tanto estética como funcionalmente. Este personaje puede adquirir habilidades con el transcurso del juego basado en la experiencia que obtiene realizando acciones tales como eliminar enemigos y cumplir objetivos propuestos por el juego llamados “*Quest*”. La experiencia ganada suele ser representada como un valor numérico que es dividida en niveles. Los niveles de un personaje, ayudan a determinar las habilidades, ítems y acciones que le son permitidos utilizar.

Otro factor importante en este tipo de juegos, suele ser la historia que envuelve al mundo de juego. Ejemplos de este género son: **Diablo II** (Blizzard), **Final Fantasy** (Square Enix), **Kingdom Hearts** (Square Enix), **Baldurs Gate** (Bioware).

Una variante de este genero son los juegos RPG multijugador masivos online (MMORPG), que se caracterizan por ofrecer la experiencia multijugador con miles de

jugadores conectados a un mundo de fantasía persistente en el cual los jugadores coexisten. Ejemplos de estos juegos son: **World of Warcraft** (Blizard), **Ultima Online** (Origin Systems) y **Linage** (NCSoft).

3.2.5 Juegos de Puzzles.

Si bien el contenido *puzzle* esta presente en todos los géneros de los videojuegos, en los juegos de este tipo, el *puzzle* es el elemento fundamental. En general se basan en un sistema de reglas simples aplicados sobre un área con piezas que el jugador puede manipular para buscar la resolución de cada nivel. Ejemplos de estos juegos son: *Tetris* (Unión Soviética), *Sokoban* (Hiroyuki Mabayashi), *Bejeweled* (PopCap) y *The Incredible Machine* (Sierra).

3.2.6 Simulaciones de Vehículos.

Los juegos de simulaciones de vehículos buscan emular las condiciones de operación de maquinarias complejas actuando en el mundo real. Los vehículos pueden ser reales, tales como aviones, helicópteros y tanques, o imaginarios. [9]

Estos juegos se caracterizan por requerir de un grado mayor de entrenamiento sobre los controles para poder disfrutar del juego, por esta razón los jugadores de este género se encuentran divididos en dos grupos. El primero, que prioriza una representación más realista a favor de los controles y reglas más realistas [38]. El otro grupo, prefiere una representación más simplificada priorizando la acción del juego. Por esta razón, la mayoría de los juegos de este género suelen ofrecer dos esquemas de controles y juego para satisfacer a ambos grupos. Ejemplos de este género son: **Microsoft Fligth Simulator** (Microsoft), **Mech Warrior 4** (Microsoft Game Studios), **Flightgear** (proyecto open source) y **Need for Speed** (EA).

3.2.7 Juegos de deportes.

Este tipo de juegos se desarrollaron cuando los videojuegos no con contaban con identidad propia y no habían ideas originales al respecto para desarrollar los juegos [3]. Sin embargo este género continuó creciendo siendo hoy en día uno de los más populares [51].

Estos juegos intentan simular las condiciones reales de los deportes, sus reglas y sus estrategias. Diferentes juegos del género pueden abarcar diferentes aspectos del deporte, ya sea: el juego en sí, el desarrollo atlético del jugador (entrenar un atleta virtual) o la administración del mismo (como por ejemplo simular ser director técnico de un equipo de fútbol).

- Ejemplos de este género son: **Sydney 2000** (Eidos Interactive), **Pro Evolution Soccer** (Konami).

3.2.8 Juegos de Construcción y administración.

Estos juegos, si bien tienen características propias de los juegos de estrategia, se destacan por no poseer un enemigo explícito en el juego y porque las acciones del jugador no interfieren de manera directa ni inmediata en el juego. Se basan en la administración de recursos, en la construcción y en tratar de mantener un equilibrio dentro del territorio que controla el jugador. [9]

Ejemplos de este género son: **Sim City** (Maxis), **Zoo Tycoon** (Microsoft) y **Trópico** (Poptop Software).

3.3 Juegos serios.

Los juegos serios son aquellos juegos que intentan lograr un fin más allá del entretenimiento. Se definen formalmente como una aplicación de software interactiva que:

- Tiene por objetivo proponer un desafío.
- Es divertido de jugar o ejecutar.
- Incorpora conceptos de puntajes y medición de resultados.
- Le otorga al usuario habilidades, conocimientos o actitudes que pueden ser aplicadas en el mundo real.

La única diferencia que existe entre los juegos serios y el resto de los videojuegos es su propósito utilitario [40]. Este tipo de juegos pueden ser utilizados para diversos fines

tales como educación, entrenamiento, defensa (reclutamiento), salud, arte, marketing, etc. [43]

Si bien este concepto no es nuevo, los juegos serios comienzan a tomar importancia a partir del año 2002 con el juego **American's Army** [43]. Éste juego desarrollado por la armada de Estados Unidos (US Army) tiene el objetivo de persuadir a los jugadores de unirse al ejército. El jugador toma el papel de un soldado que debe cumplir un entrenamiento, y luego librar una serie de batallas en modo multijugador en un ambiente simulado de combate. [40]

Según la *US Army* el juego se convirtió en una herramienta de reclutamiento exitosa, argumentando que el 28% de los jugadores clickean el link hacia el sitio web de reclutamiento [42].

Por otro lado, los juegos conocidos como *advergames* (juegos publicitarios) son caracterizados por tener el objetivo de comunicarle al jugador una marca o un producto. Este tipo de juegos no son nuevos tampoco ya que en el año 1983 se lanzó el juego **Pepsi Invaders** para Atari. Este consistía en un clon del juego **Space Invaders** pero con gráficos representando a la marca Pepsi (Ver figura 3.3.1) Este tipo de juegos tienen mayor atención en la actualidad ya que los costos de distribución de los juegos es muchísimo menor gracias a la distribución digital a través Internet.



Figura 3.3.1 Pepsi Invaders (Atari)

Los videojuegos educativos, por su parte, buscan transmitirle al jugador conocimientos a través del videojuego sin dejar de lado el entretenimiento. Si bien todos los juegos enseñan al jugador algún tipo de habilidad, estos juegos tienen como principal objetivo transmitir contenidos y habilidades que le puedan servir al jugador fuera del juego.

Existen estudios sobre este tema en diferentes instituciones alrededor del mundo. En nuestro país, el grupo de investigación llamado “Tiza, Papel y Byte” ha desarrollado desde el 2008 distintas experiencias con alumnos tanto de escuelas privadas, como públicas [44]. En una de las experiencias documentadas, dejaron que los alumnos jueguen a un juego llamado “Ayti: The cost of life” (propiedad de Unicef) y se guió a partir del juego a reflexiones sobre la educación, el trabajo, y la salud en torno de la pobreza.

Otro ejemplo de este tipo de juegos es el juego **Revolution** desarrollado por el MIT como parte de su programa de educación “*The Educational Arcade*”. Éste consiste en una modificación del juego **Neverwinter** (Bioware) el cual ayuda a los adolescentes a aprender sobre el día de la independencia estadounidense basado en varias historias que plantea el juego [43].

El juego **12 de Septiembre** es un ejemplo de los juegos denominados de activismo. En éste se plantea la posibilidad de disparar misiles para eliminar los terroristas de una ciudad, sin embargo, la explosión del misil termina por generar daño colateral matando civiles en la ciudad. Un conjunto de ciudadanos llora a los civiles muertos, y se transforman en terroristas. Este juego surge como protesta acerca del accionar estadounidense contra el terrorismo.

Existen otros tipos de juegos llamados juegos de salud, en donde el jugador se enfoca en el entrenamiento de sus habilidades mentales o incluso en el cuidado de su salud física. Accesorios como el Wii Fitness (una balanza que permite interpretar los movimientos e inclinación del jugador para consola Nintendo Wii), el Kinectic (para la consola Xbox360), o el PS Move (para la PS3) permiten capturar los movimientos físicos del jugador y guiarlo a través de juegos que requieren actividades físicas. Por otro lado juegos como **Brain Challenge** (GameLoft), desafían al jugador a actividades que ponen a prueba su memoria, su capacidad de cálculo matemático y lógico.

3.4 Juegos Casuales.

Existe otra forma de clasificar los juegos, y esta se basa en la inversión de tiempo y esfuerzo que debe realizar el jugador para que pueda disfrutar de una sesión de juego [47].

Los juegos casuales se diferencian de los juegos “Core” o tradicionales, en que éste tipo de juegos requiere de un menor tiempo para aprender a jugar y para desarrollar una sesión de juego [46], cuentan con un gameplay indulgente (con menor penalización por los errores cometidos) e incluyen un esquema de controles más simplificados. [48]

Si bien la audiencia de ambos tipos de juegos no es mutuamente excluyente, los jugadores casuales presentan una caracterización distinta a los de los jugadores tradicionales. Este tipo de audiencia reúne a personas que no estaban tradicionalmente interesados por videojuegos tal como un abundante público femenino y gente mayor de edad [49]. En la siguiente sección se analizará este tema con mayor detenimiento.

Si bien el mercado de los juegos casuales toma importancia a partir de 2002 donde los ingresos generados por esta industria pasan de no significar nada a ser más de 600 millones de dólares en Estados Unidos, este tipo de juegos guarda mucha similitud con los juegos de los comienzos de la industria.

En los primeros, tal como en los juegos casuales, los juegos eran más simples y apuntaban hacia un público amplio (no existía una audiencia con trayectoria en videojuegos dado que la industria acababa de inventarse). Es por esto que es importante señalar que no son los juegos por si mismos la clave del éxito sino los modelos de negocios que surgieron gracias a la penetración en el mercado que proporciona Internet y la distribución digital [48]. En el 2007 la industria creció aportando una ganancia de \$2.25 miles de millones de dólares con un crecimiento estimado del 20% anual. [50]

Entre las formas más usuales de modelos de negocios se encuentran los juegos descargables con la estrategia de “prueba y compra” donde se ofrece una versión reducida del juego en forma gratuita con la opción de pagar por expandir el juego a la versión completa; juegos patrocinados por propagandas, donde el juego muestra propagandas en momentos predeterminados (donde es más común al inicio del juego o al finalizar niveles); juegos con suscripción (populares en los juegos multiplayers online) donde se paga una cuota fija para poder jugar durante un periodo fijo de tiempo; y por último los juegos con micro-transacciones las cuales constan de acciones pagas para agregar contenido a un juego que se distribuye en forma gratuita.[50]

Algunos ejemplos de juegos casuales son: **Bejeweled** (Pop Cap), **Luxor** (Mumbo Jumbo) y **Diner Dash** (Shockwave).

3.5 Juegos sociales.

Los juegos sociales son aquellos que se desarrollan en el marco de una red social como puede ser Facebook, My Space, Orkut, HI-5, o bien incorporan elementos de las redes sociales [189]. El objetivo de las redes sociales es la interacción entre nodos (individuos u organizaciones) que se encuentran unidos por distintos tipos de interdependencia (valores, ideas, amistad) y que se dan mediante la creación de vínculos y el intercambio de mensajes y contenidos [191]. Los juegos sociales permiten la creación de amistades virtuales, y la interacción social, a partir de compartir espacios de juegos. El objetivo de ambas, redes y juegos sociales, es complementario y es por ello que el crecimiento de las redes sociales y este tipo de juegos se ven acompañados [189][190]. Si bien existen juegos con estas características que no forman parte de redes sociales, los juegos que si utilizan el marco de una red social ocupan el 57% del mercado [189].

Los juegos sociales en cuanto a mecánicas y modos de juegos son variados, pero generalmente se ubican en la categoría de los juegos casuales. La audiencia de este tipo de juegos se concentra mayormente en el público femenino (un 55% de los jugadores son mujeres), y el jugador promedio de estos juegos son mujeres de 43 años de edad [189]. La jugabilidad en los juegos sociales esta mayormente centrada en la interacción social, no obstante se prevé que en los próximos años es espectro y la riqueza de los juegos crezca, junto con el costo y tiempo de desarrollo [190].

Los juegos sociales representan hoy en día un sector con un gran crecimiento económico que tiene una audiencia y características particulares. Este tema será tratado nuevamente en el capítulo 7.

3.6 Juegos Independientes.

Los videojuegos independientes se definen como aquellos que son desarrollados por una persona, o un grupo reducido de personas, que no cuentan con el soporte financiero de un inversor privado o un publicador de juegos (*publisher*) [92].

El desarrollo independiente nace al mismo tiempo que la industria de videojuegos durante los la década del 70's, sin embargo, la distribución de estos juegos estaba controlada por los *publishers* y minoristas dificultando esta actividad. A partir de los años 1990's la conexión a Internet en los hogares se hizo más popular, y los videojuegos independientes tuvieron la oportunidad de distribuirse a través de Internet. La aparición de servicios gratuitos como You tube, y Flickr, a partir del año 2000, facilitaron la promoción de estos juegos, y el fortalecimiento de los portales online de juegos (que distribuyen juegos gratuitos que se pueden jugar en el explorador) hicieron aún más fácil la distribución [11].

Los juegos independientes no tienen una limitación de género ni de tamaño, y el tiempo de desarrollo varia según la complejidad del juego, el tamaño y la experiencia de sus desarrolladores. Algunos ejemplos de estos juegos son: **World of Goo** (2D boy), **Minecraft** (Mojang), **Passage** (Jason Rohrer), **Machinarium** (Amanita Design) y **Braid** (Number None, Inc).

3.7 Caracterización de los jugadores.

El estereotipo común de jugador lo caracteriza como un joven masculino, que pasa la mayor parte de su tiempo jugando videojuegos, sin embargo, las estadísticas recopiladas muestran una caracterización muy distinta [47].

El jugador, según la ESA, tiene edad promedio de 34 años. Siendo un 49% en el rango de los 18-49 años, un 25% debajo de los 18 años y un 26% sobre los 50 años (Ver figura 3.7.1). En cuanto a género, un 60% son hombres y un 40% son mujeres [51].

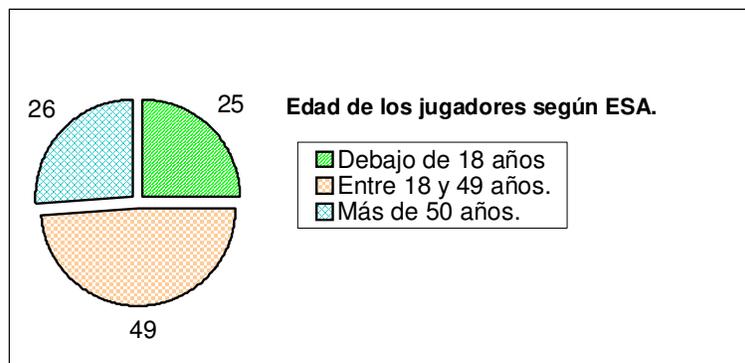


Figura 3.7.1 Edad de los jugadores según ESA.

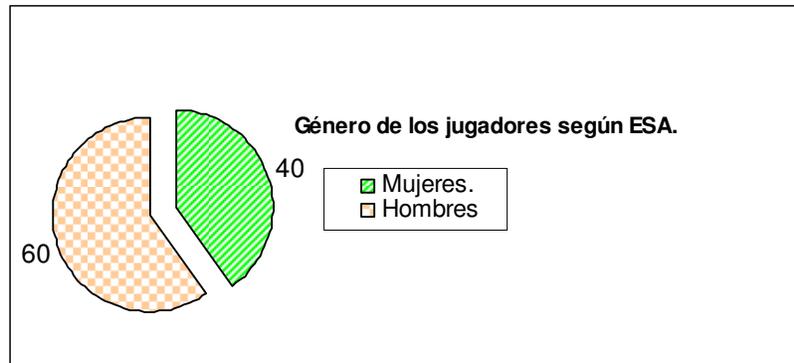


Gráfico 3.7.1 Género de los jugadores según ESA

Sin embargo entre los jugadores debajo de 18 años la cantidad de mujeres (30%) es significativamente mayor a la cantidad de hombres (20%) [51]. El crecimiento y la diversificación del mercado de los juegos se debe en parte al surgimiento de los juegos casuales. Se puede observar que, dentro de los juegos casuales, la caracterización del jugador es diferente ya que la edad promedio es mayor que los 35 años y con un 29% hombres y un 71% mujeres [47].

Esta diferencia demográfica entre ambos grupos se debe a que los juegos casuales tuvieron éxito en diversificar el mercado, y exploraron las diferencias de gustos entre los jugadores de diferentes géneros y edad [47].

Además, los juegos familiares tomaron importancia gracias a la entrada de los juegos casuales al mundo de las consolas de la mano de la Nintendo Wii. Reginald Fils-Aime, de Nintendo, afirma que la idea tras la consola Wii era priorizar la jugabilidad con una nueva forma de jugar que permitiría que cualquiera pueda jugar [49]. Según el ESA, los juegos familiares representan el 15.3% de las ventas dentro de los *súper géneros* (una clasificación más abstracta y menos estructurada de juegos) quedando en tercer lugar, luego de los juegos de deporte (con un 19.6%) y los juegos de acción con un (19.5%) [51].

3.7.1 Jugadores “Core” y jugadores casuales.

Dadas estas características demográficas, la forma más usual de clasificar a los jugadores es en jugadores “Core” y jugadores “Casuales”.

Los jugadores de juegos “Core” prefieren juegos con controles y jugabilidad compleja que requieren sesiones de juegos largas (Ver gráfico 3.7.1.1). Prefieren que el mundo que representa el juego tenga una ficción negativa, es decir, que involucren conflictos con violencia, conceptos de supervivencia y de combates. En cuanto a la dificultad, los jugadores prefieren que el juego les presente retos (Ver gráfico 3.7.1.2) [49].

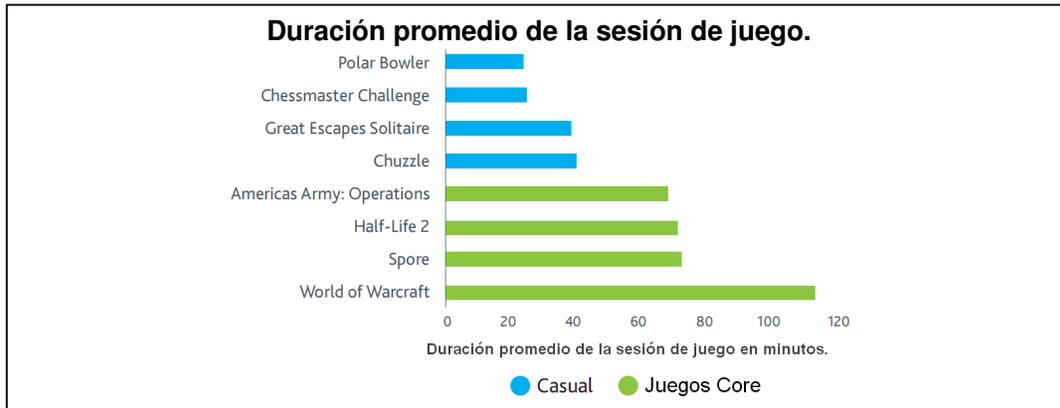


Gráfico 3.7.1.1 Duración de sesión en juegos casuales y juegos *hardcore* [46].

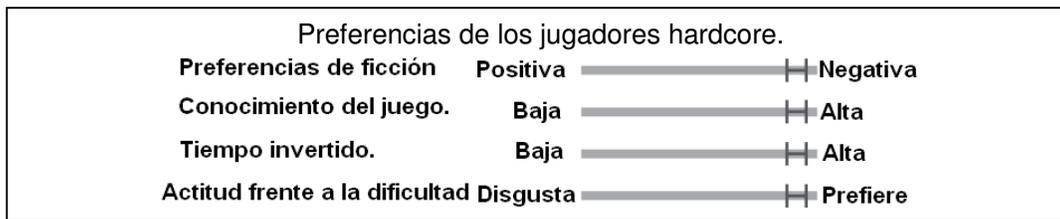


Gráfico 3.7.1.2 Preferencias de los jugadores *hardcore*.

En cuanto a los jugadores casuales, prefieren juegos que requieran sesiones de juego cortas (y que puedan interrumpirlas sin comprometerlas) y una ambientación positiva donde los conflictos presentados minimicen la violencia. En cuanto al conocimiento del juego los jugadores prefieren juegos sencillos de aprender, lo que no significa que no disfruten de juegos con mecánicas más complejas. El tiempo invertido en los juegos varía desde invertir poco a invertir mucho tiempo en jugar. En cuanto la dificultad, prefieren juegos más fáciles aunque no les disgustan los desafíos. (Ver figura 3.6.1.2) [49]

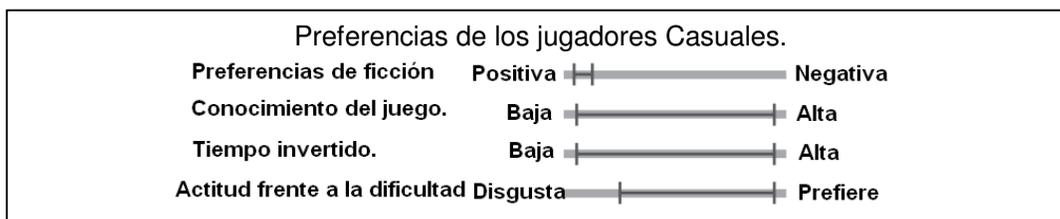


Figura 3.7.1.2 Preferencias de los jugadores casuales.

3.6.2 Clasificaciones por contenido y edad.

La creación de videojuegos con contenidos violentos o de contenidos que pueden ser perturbadores para audiencias de edades menores a la mayoría de edad hizo que surgiera la necesidad de clasificar a los juegos para ayudar a los padres y a los cuidadores de estos menores a saber cual es la edad recomendada para cada juego y que clase de contenidos puede llegar estar presente en cada juego. Distintas organizaciones sin fines de lucro y autorreguladas fueron creadas con el objetivo de satisfacer esta necesidad. Las clasificaciones más comunes y utilizadas son la clasificación PEGI y la clasificación ESRB [52] [53].

Algunos países tienen sus propios sistemas de clasificación, que incluso, pueden depender de algún organismo estatal. Como ejemplos de estas clasificaciones se pueden citar la “*The Office Of Film & Literature Classification*” de Nueva Zelanda [54], la “*Computer Entertainment Rating Organization*” (CERO) en Japón [55] y la USK en Alemania [56].

Ninguno de estos sistemas de clasificación tiene en cuenta el nivel de dificultad ni de calidad del juego ya que el criterio se centra en los contenidos que el jugador podrá experimentar al jugar el juego y no en la calidad del mismo.

3.7.2.1 Clasificación ESRB.

La clasificación ESRB fue establecida por la ESA en 1994, y divide a los juegos en 6 categorías basadas en el contenido y en las diferentes etapas de madurez del jugador.

Estas categorías son:

- eC “*Early Childhood*” (Niñez temprana): El juego tiene contenido apto para chicos con edad de 3 o más años.
- E “*Everyone*” (Para todos): El juego es apto a partir de los 6 años de edad. Los juegos en esta categoría pueden incluir violencia caricaturizada o uso poco frecuente de lenguaje soez.

- E 10+ “*Everyone +10*” (Para todos +10). Tiene contenido apto a partir de los 10 años de edad. Dado que incluye más violencia caricaturizada que la anterior, y puede incluir un uso más frecuente de lenguaje soez, y temas sugestivos.
- T “*Teen*” (Adolescentes): Son aptos a partir de los 13 años de edad. Pueden tener violencia, temas sugestivos, humor crudo, sangre (no en proporciones exageradas), apuestas simuladas y lenguaje adulto.
- M “*Mature*” (Maduros): Es apto para personas mayores de 17 años. Puede contener violencia intensa, sangre (en exageración) y lenguaje adulto.
- AO “*Adults Only*” (Solo adultos): El contenido es apto a partir de los 18 años de edad y puede contener escenas intensas y explícitas de violencia, situaciones sexuales o desnudez.

Cuando un juego todavía no ha sido clasificado (más probable durante las campañas publicitarias) se debe colocar la leyenda “*Rating Pending*”, es decir, calificación pendiente a la publicidad que acompaña el juego.

Además de esta clasificación se suele acompañar al juego con los denominados descriptores de contenidos ESRB que muestran, en manera resumida, qué se puede encontrar en el juego explícitamente. Ejemplos de estos descriptores son: “Sangre”, “Violencia Caricaturizada”, “Uso de drogas”, “Uso de alcohol”, etc. En la figura 3.7.2.1.1 Se puede ver un ejemplo de clasificación ESRB del juego **StarCraft II** de Blizzard. A la izquierda se muestra la clasificación “T” (para Adolescentes) y los descriptores a la derecha (Derramamiento de Sangre, Lenguaje adulto, Temas Insinuantes, Uso de alcohol y tabaco, Violencia).



3.7.2.1.1 Ejemplo de clasificación ESRB del juego StarCraft II.

No es obligatoria la clasificación de los juegos por ningún tipo de ley, pero si forma parte de una responsabilidad social desde las empresas distribuidoras y desarrolladoras. Al día de hoy ESA tiene registrados 22.330 juegos de más de 350 empresas distribuidoras [53].

3.7.2.2 Clasificación PEGI.

El sistema de clasificación PEGI fue estrenado en la primavera del 2003, y surgió como una solución a la heterogeneidad de las clasificaciones existentes en los distintos países de Europa. Sin embargo, hoy en día, es también aplicada en distintos países entre los que se encuentra la Argentina. Este sistema de clasificación divide a los juegos en 5 categorías de edades y 8 descriptores de contenido gráficos.

- PEGI 3. Son juegos aptos para todos los grupos de edades. Se tolera el uso de violencia caricaturizada en un contexto cómico, y de manera que el jugador no asocie la violencia del mundo fantástico con la del mundo real. No debe contener lenguaje soez, ni escenas de violencia ni referencias a la actividad sexual.
- PEGI 7: Son juegos similares a la de la categoría anterior pero que pueden tener contenido que puede asustar al jugador, se permiten escenas de desnudo parcial pero no dentro de un contexto sexual.
- PEGI 12: Se permiten juegos con violencia menos caricaturizada. El lenguaje soez debe ser poco frecuente y no debe contener referencias sexuales.
- PEGI 16: Esta categoría incluye a los juegos con representación más realista de violencia (o actividad sexual). Puede incluir lenguaje soez y referencias al uso de tabaco, y drogas junto con la representación de actividades delictivas.
- PEGI 18: La clasificación de adulto se aplica cuando incluye elementos de violencia brutal, o específicos de violencia de manera exagerada.

Los descriptores gráficos utilizados por este sistema pueden verse en la tabla 3.7.2.1, mientras que un ejemplo de la utilización de este sistema puede verse en la figura 3.7.2.2 del juego **Call Of Juarez** de Ubisoft [52].

Símbolo	Descripción
	Utilización de lenguaje Soez.
	Alusión a la Discriminación
	Alusión al uso de drogas.
	Juego con capacidades online.
	Juego que pueden asustar.
	Juegos que incluyen apuestas.
	Juegos con contenido sexual.
	Juegos con Violencia.

Figura 3.7.2.1 Tabla de descriptores gráficos del sistema PEGI.

16 **Call of Juarez: Bound in Blood**  [www](#)

Ubisoft EMEA

 The content of this game is suitable for persons aged 16 years and over only.
It contains: Realistic looking violence - - Strong language

 This game allows the player to interact with other players ONLINE

 Plataforma: XBox 360

 Género: Action/First Person Shooters

Releasedate: 2009-07-27

Figura 3.6.2.1 Ejemplo de clasificación PEGI

4. Videojuegos, sociedad y cultura.

Tal como sucede con los libros y las películas y otros medios de expresión, los videojuegos dejan marcas en la sociedad que los consume.

Por un lado, los videojuegos como tales representan un *espacio de juego* para los chicos [57] y forman parte importante de su formación dada las prolongadas sesiones de juegos que suelen acontecer. La violencia, el contenido sexual, y cualquier tipo de contenidos que pueden resultar ofensivos para la exposición ante menores ha generado desde el inicio de los videojuegos [11] conflictos éticos y morales entre las empresas desarrolladoras y distribuidoras, diseñadores de juegos, jugadores y padres. La adicción a los videojuegos y las consecuencias para la salud también han sido objeto de discusiones, de numerosos estudios y controversias [60].

Por otro lado los videojuegos también han demostrado haber creado nuevas formas de relaciones sociales a través de juegos multijugadores y comunidades de jugadores que comparten e incluso crean contenidos [59]. También han demostrado la capacidad de transmitir mensajes [61] a través de nuevas formas de narrativas y, por otro lado también han mostrado ser capaces de perseguir otros objetivos, además del entretenimiento, como lo son la educación y el ejercicio físico [43].

4.1 Videojuegos y controversias.

Los videojuegos, tal como las películas y la televisión, son a veces criticados por tener contenidos con demasiada violencia, con sexo, con personajes estereotípicos o falta general de contenido edificante. Pero, a diferencia de estos otros medios que ya atravesaron por su respectivo proceso de críticas, los videojuegos han generado nuevos tipos de interrogantes éticos y morales que captan la atención de psicólogos, sociólogos, y de diversos investigadores. [11]

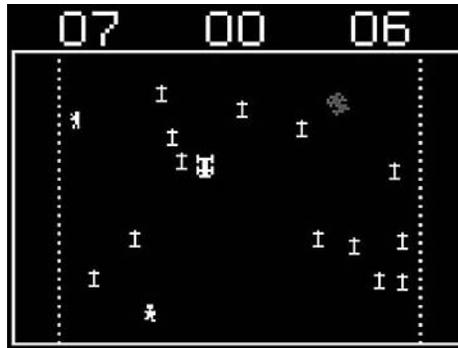


Figura 4.1.1 “**Death Race**” (Exidy 1976).

Si bien al inicio, los juegos eran de una naturaleza más “inocente” y con menos capacidad de representación gráfica y sonora que en la actualidad, uno de los primeros juegos que llevó a la controversia fue creado en esos comienzos durante 1976 y bajo el nombre de “**Death Race**” por la empresa Exidy. El objetivo del juego consistía en conseguir puntos atropellando el mayor número de “*gremlins*” posibles. Sin embargo la representación de los “*gremlins*” con homrecitos de palitos resultó indistinguible de la representación de figuras humanas (Ver figura 4.1.1). Estos personajes al ser atropellados emitían el sonido de un grito y en el lugar en que sean atropellados, se colocaba una tumba con una cruz. Cabe destacar que el nombre con que se referían al juego durante la etapa de desarrollo fue “*Pedestrian*”, peatón en inglés, y se cree que fue cambiado para facilitar la distribución.

La diferencia que plantea este juego con los de su época, es que incitaba a que el jugador atentara contra gente ordinaria y no contra enemigos deshumanizados [11]. Ésta controversia adquirió una gran exposición mediática gracias a una noticia lanzada en el programa “60 minutos” [60]. Este evento le otorgó publicidad gratuita al juego que, sin embargo, no logró hacer que sus ventas aumentarán o disminuyeran significativamente.

En el año 1992, el juego “**Mortal Kombat**” (Midway) fue el primero en abrir un debate acerca de la influencia de los videojuegos violentos sobre la juventud. Si bien no fue uno de los primeros juegos de peleas, lo que lo destacó sobre los demás, fue el hincapié que éste hacía en la violencia física. Algunos de los elementos más sobresalientes del juego son los efectos de sangre y daño exagerados, gráficos foto realistas conseguidos a partir de técnicas similares a la rotoscopía¹, y un elemento

¹ Rotoscopia: Técnica que consiste en realizar animaciones a partir de grabaciones reales.

especial del juego llamado "Fatality". Una "fatality" consiste en una serie de movimientos que terminan por destruir al oponente por medio de diversas animaciones violentas que son activadas por una combinación de botones cuando un oponente es derrotado [11].

En 1993 el senador Joseph Lieberman organizó una audiencia sobre los videojuegos con el argumento de que éstos eran simplemente "la comercialización de la violencia para los niños". Esta audiencia se centraba principalmente en dos juegos, el anteriormente mencionado "**Mortal Kombat**" y el juego para Sega CD llamado "**Night Trap**" creado en 1992. [11] El juego "**Night Trap**" utilizaba escenas grabadas de actores reales en donde el jugador debía evitar que unas jóvenes adolescentes (a veces mostradas con escasa vestimenta) sean asesinadas. El objetivo, sin embargo, fue trasgredido por el senador Josep Lieberman quien afirmaba que en realidad era asesinarlas [60]. Tras la exposición pública que consiguió este juego, y pese a que fue retirado de algunas tiendas, sus ventas se incrementaron notablemente permitiendo que el juego sea trasladado a otras consolas. En respuesta de estos debates surge el sistema de clasificación de juegos ESRB que fue explicada en el capítulo anterior [11].

Otro juego que incitó similares repercusiones en los medios fue **Wolfestein 3D** (id Software). Este es un juego de acción en primera persona donde el protagonista debe escapar de una celda en un castillo y matar a los soldados del ejército Nazi que intenten impedirlo. El juego fue prohibido en Alemania en 1994 por ser considerado una ofensa federal por la utilización de los símbolos y la música del himno Nazi. Años más tarde, Id Software, lanzó otro juego de acción en primera persona llamado **Doom** donde el jugador tomaba el papel de un soldado en una base espacial que debía aniquilar todo tipo de monstruos y demonios. El juego por sí solo no ocasionó controversia hasta que se produjo en 1999 la llamada "La Masacre de Columbine". Dos estudiantes, Dylan Klebold y Eric Harris, de 18 y 17 años respectivamente, entraron armados con armas de fuego y explosivos a una escuela en el pueblo Columbine en Colorado, Estados Unidos. En total ocasionaron la muerte de 12 personas y varios heridos, entre las que se encontraban maestros y alumnos. Los investigadores descubrieron que ambos adolescentes estaban obsesionados con los videojuegos como "**Doom**" (Id software), "**Duke Nukem 3D**" (3D Realms) y "**Mortal Kombat**" (Midway). Por esta razón los familiares de las víctimas demandaron a las compañías culpándolas por los crímenes de los estudiantes. En 2001 las familias habían demandado a 25 compañías incluyendo a Sega, Nintendo, Aol Time Warner y a Id software por un total de 5 miles de millones de dólares en daños. La justicia,

finalmente, determinó que Klebold y Harris fueron los únicos culpables por el crimen [60].

En cuanto a contenido sexual, uno de los primeros juegos en atraer controversia fue "**Custer's Revenge**" creado en 1982 por la empresa Mystique para la consola Atari 2600. En él, el jugador debía tomar el papel de un vaquero desnudo que corre a través de una llanura esquivando cactus y flechas que caen del cielo para llegar al otro lado donde se encuentra una mujer nativa americana atada a un poste. Al llegar allí el vaquero procede a violarla intentando evitar ser asesinado. El juego fue recibido con controversia desde distintos grupos sociales afectando a la reputación de Atari, pese a que esta empresa no estuvo involucrada ni en el desarrollo ni en la distribución del juego.

Un ejemplo más actual, que ha causado distintos tipos controversias en muchos aspectos tales como la violencia, el sexo y consumo de drogas entre otros contenidos poco edificantes, es la serie de juegos **Grand Theft Auto** (GTA) de Rockstar. En estos juegos el jugador toma el papel de un criminal que tiene a su disposición un mundo abierto con el cual puede interactuar. Robos, violencia, guerras de pandillas, alusión al consumo y venta de drogas y el consumo de prostitutas son algunas de las acciones que van guiando al jugador a una vida de criminalidad [11]. El juego sin embargo, logró ganarse el título de "gran éxito de ventas" en distintas consolas tales como Xbox, PS2 y PS3 y alcanzó un total de 124 millones de unidades vendidas [63].

Otros juegos igualmente controversiales son **Bully** (Rockstar) donde el jugador toma el papel de un chico que busca problemas en la escuela dando la posibilidad de golpear a otros chicos y gastar bromas pesadas generalmente causando daño.

El juego "**Rule of Rose**" (Punchline), por su parte, fue cancelado en Inglaterra, y censurado en Australia, dado que en él, el jugador toma el papel de una joven adolescente que es maltratada por otro grupo de chicas, sufriendo diversos tipos de torturas, llegando incluso a ser enterrada viva.

El cuidado de los padres, y los diversos sistemas de calificación son los que protegen a los niños de este tipo de contenidos. Según el ESA el 70% de los padres, en Estados Unidos, creen que el sistema de calificación ESRB resulta útil para este control, y el 64% de ellos cree que los videojuegos constituyen una parte positiva de sus vidas [51].

Según la “Kaiser Family Foundation”, una fundación sin fines de lucros que se ocupa de realizar encuestas e investigaciones que se centran en el cuidado de la salud y el bienestar en EEUU [65] revelan, en una encuesta realizada en el 2007, que solo el 8% de los padres cree que los videojuegos es el medio que más les preocupa que afecte a sus hijos (Ver gráfico 4.1.1), mientras que el 32% cree que es la televisión y sólo 16% piensa que son todos por igual de preocupantes (TV, Internet, Videojuegos y Películas).

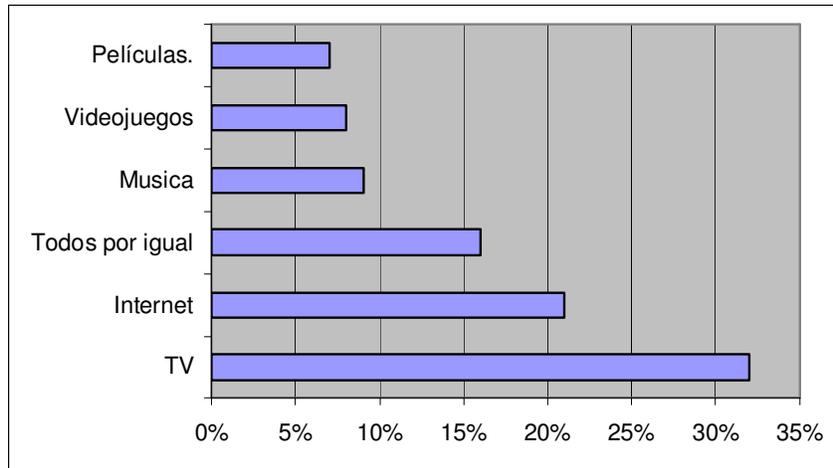


Gráfico 4.1.1 ¿Cuál medio es más preocupante para los padres?

4.2 Videojuegos y violencia.

Desde la aparición de juegos con grandes contenidos de violencia y dado que el 73% de los jugadores varones inclinan sus preferencias hacia este tipo de juegos [67], se ha planteado el interrogante de que si los videojuegos constituyen causantes del comportamiento violento [11][66][60]. Sin bien los estudios que se encuentran a favor de esta hipótesis son numerosos, también los son los estudios que demuestran lo contrario [66].

Sin embargo, la mayoría de las investigaciones que sostienen que existe este efecto negativo suelen presentar falencias metodológicas, ignoran factores tales como la cultura y el estado socioeconómico y no suelen ser concluyentes [62].

Las investigaciones, en general, se centran en encontrar la correlación entre el comportamiento violento y los videojuegos violentos [67], no obstante, no se ocupan en encontrar una relación causal entre estos, de modo que los resultados también pueden ser interpretados como que las personas con comportamientos violentos prefieren también el entretenimiento violento [62].

Incluso, según las estadísticas de delitos federales de EEUU, la tasa de delincuencia juvenil violenta ha alcanzado un nivel mínimo en 30 años, siendo que en ese tiempo la cantidad, la popularidad, y las capacidades de representación de los videojuegos ha estado creciendo significativamente [60][62][36]. De existir esta relación causal, los crímenes violentos no hubieran estado descendiendo desde los años 90 [66].

Si bien existen los casos de violencia con jóvenes que juegan videojuegos de este tipo, hay que considerar que el 90% de los varones y el 40% de las niñas son consumidores de videojuegos, por lo cual, la inmensa mayoría de ellos no cometen actos de violencia [60].

Según el ESA, el 86% de los niños compran los juegos con permiso de sus padres, por lo que la oportunidad de descalificar el contenido gracias a las facilidades que otorgan los sistemas de calificaciones como el ESRB, recae sobre los padres [51].

4.3 Los videojuegos y salud.

Los videojuegos también han sido objeto de estudio en investigaciones que los relacionan con riesgos para la salud. Los jugadores que pasan mucho tiempo jugando tienden a llevar una vida de sedentarismo que suele estar acompañado de problemas como el aislamiento social y la obesidad [60].

Si bien estos riesgos son propios de la actividad sedentaria y no del acto de jugar, la adicción a los videojuegos es un problema que concierne al estado, a las empresas, a niños, adultos, y padres. Según la psicoterapeuta Hilarie Crash y el profesional del cuidado de la salud Kim McDaniel:

“La adicción es cualquier comportamiento placentero que lleva a una persona a que no sea capaz de detenerse una vez que ha comenzado, lo cual por supuesto, tiene consecuencias negativas para esta persona.”[69]

La adicción a los videojuegos tiene las mismas consecuencias que cualquier otra adicción: cambios en el comportamiento social, aislamiento, conflictos con miembros de su entorno familiar y renuencia para trabajar o estudiar [60]. En general, los chicos que pasan demasiado tiempo jugando todos los días muestran un más bajo desempeño escolar [67]. Sin embargo los videojuegos por si solos no constituyen el causante de estos problemas sino que contribuyen con estos [60].

Solo el 46% de los padres limitan las horas que los chicos pasan jugando videojuegos [60] y es frecuente la utilización de los videojuegos como sustituto de “niñeras” ya que mantienen entretenidos a los niños por largos periodos de tiempo [71].

“Jugadores en línea anónimos” es una organización sin fines de lucro que proporciona a personas con problemas de adicción y a familiares involucrados, un espacio para compartir, orientarse e informarse. Esta organización surge en respuesta al conflicto personal de su fundadora Elizabeth Woodley cuyo hijo se suicidó en frente de su computadora en el 2001 y era un jugador adicto al juego Everquest Online. Él había renunciado a su trabajo para poder pasar más tiempo jugando, y se había recluido en su apartamento para lograr largas sesiones de juego in-interrumpidas [72]. Sin embargo a Shawn Woodley, el hijo de Elizabeth, se le había sido diagnosticado TDAH (Trastorno por déficit de atención con hiperactividad) y epilepsia, cuyas patologías controlaba con medicamentos [73].

El TDAH se trata de un trastorno neurológico del comportamiento caracterizado por distracción moderada a severa, períodos de atención breve, inquietud motora, inestabilidad emocional y conductas impulsivas. El cuadro empeora en situaciones que exigen mucha atención y tareas repetitivas [74]. El juego no fue la causa del suicidio, sino un contribuyente, dada la patología de la víctima.

Por otro lado, los videojuegos también han sido el objeto de estudio de investigaciones que los relacionan con beneficios para la salud. La mejora de los reflejos y la coordinación mano-ojo, especialmente en los juegos de acción, ha sido destacada en los “archivos de cirugía”² tras el estudio de un grupo de cirujanos. Aquellos que jugaban al menos tres horas por semana cometieron, en promedio, un 37% menos de errores durante las simulaciones y las completaron un 27% más rápido [75].

² Una colección de artículos e investigaciones relacionadas con la cirugía.

Los juegos tipos puzzles, por su parte, ayudan a mejorar las capacidades de reconocimiento de patrones, la memoria y la capacidad de análisis, mientras que los juegos de estrategia tal como el **Age of Empires** (Microsoft), **Civilization**, y **Sim City** (Maxis) ayudan a mejorar la capacidad de planeamiento y la administración de recursos [70].

Por otro lado los juegos también han sido utilizados para procesos de rehabilitación. Un grupo de investigación de la Universidad del sur de California, ha modificado un juego comercial para utilizarlo como terapia para los veteranos de la guerra de Iraq y ayudar a debilitar el estrés post traumático que sufren [76].

En New Jersey, investigadores del Instituto de Tele-rehabilitación de la Universidad de Rutgers, han desarrollado un programa de rehabilitación que mediante la utilización de videojuegos ayuda a ejercitar a las funciones motoras de la mano y del antebrazo a pacientes adolescentes con parálisis cerebral³. El tratamiento consiste en jugar 30 minutos al día un videojuego hecho a medida y ejecutado sobre un hardware modificado que utiliza un guante con sensores especiales. Los pacientes que ejecutaron dicha rutina 5 días a la semana tuvieron una mejoría en sus quehaceres diarios y pudieron entrenar sin la necesidad de la presencia de los profesionales, ya que eran monitoreados a distancia [76].

Otra práctica desarrollada por la psicóloga Deborah Strokes consiguió en el 70% de los pacientes reducir o hasta incluso eliminar la medicación para la migraña luego de que estos se sometieran a un tratamiento de 14 meses y medio de duración en el cual debían jugar videojuegos de entrenamiento cerebral. Los participantes de este tratamiento también notaron una reducción en los síntomas que acompañan a este desorden los cuales incluyen a la depresión, la ansiedad, los dolores de cabeza, el insomnio y otros síndromes de dolor [77].

A partir del 2006, y gracias a la Nintendo Wii, el concepto de nuevas formas de controles se hizo presente en el mercado de las consolas. Este nuevo tipo de controles requieren de una participación más activa, en el sentido físico, de parte del jugador [60]. Los juegos de ejercicios toman importancia a causa de este fenómeno y Nintendo

³ La parálisis cerebral es un trastorno permanente y no progresivo que afecta a la psicomotricidad del paciente afectado.

a conciencia de ello lanza el accesorio Wii Balance Board. Este accesorio consta de una balanza que es capaz de detectar las secciones que ejercen mayor presión sobre ella, y de esta manera intuir parte de la posición en que se encuentra el usuario. Según Nintendo este accesorio “[...] combina diversión y entrenamiento físico en un solo producto”[78]. Este tipo de juegos ayuda a los usuarios a mejorar su frecuencia cardíaca, su equilibrio y la musculatura [43]. Por su parte Sony y Microsoft lanzan sus respectivos accesorios Ps Move y Kinectic, y desarrollan un catálogo de juegos de ejercicio similares.

4.4 Los videojuegos y la cultura.

Según Chris Crawford la representación es uno de los componentes del concepto de juego, ya que como tal, es un subconjunto de una realidad [3]. Los videojuegos en consecuencia, no se encuentran aislados de la cultura que los rodea ya que son construcciones humanas cargadas de conceptos culturales de quienes los desarrollan. Se define a la cultura como un conjunto de todas las formas, los modelos o los patrones, explícitos o implícitos, a través de los cuales una sociedad se manifiesta [79].

Considerando, por ejemplo, un mazo de cartas popularizada durante el siglo XIV y XV en Europa se puede ver cómo elementos de la cultura en ese punto de la historia influyeron en su diseño. Cada uno de los cuatro palos significan un elemento distinto de la sociedad de la época: el oro representa a los comerciantes, la copa al clérigo, la espada a la realeza y el basto a los campesinos [80]. Otro ejemplo similar, es el caso de las cartas españolas que, reflejando el machismo de la época (siglo XVIII), la carta que representaba a la reina fue eliminada del mazo [81].

En el juego GTA (Rockstar), por ejemplo, se pueden observar mitos urbanos de la cultura criminal y de los mafiosos del cine. En Jet Grind Radio (Smilebit) se ofrece una exposición, acompañada de ciencia ficción, de la cultura de los grafitis y el patinaje callejero. En el juego Sims (Maxis) se pueden observar fenómenos como el consumismo, preconceptos de la vida cotidiana, e idealizaciones del ser humano en cada una de sus etapas de madurez.

El jugador no solamente interactúa con las reglas del juego, sino también con las convenciones de la cultura que ellos reflejan y transforman [80].

El reflejo cultural en los juegos no solo sucede de manera natural o implícita, sino también puede darse de forma deliberada. Will Wright, al momento de diseñar el juego Sims (Maxis) tomó conceptos de trabajos de Christopher Alexander, un arquitecto que ha estudiado cómo los diseños influyen en el comportamiento humano, y los trabajos de un sociólogo llamado John Robinson, que analizó la manera en que las familias distribuían su tiempo [82].

El juego gratuito “**American Dreams**” de Terry Cavanagh, Stephen Lavelle, Jasper Byrne y Tom Morgan Jones, por su parte, deliberadamente trata temas como el consumismo, y la acumulación de riquezas propias de la sociedad actual incitando a la autorreflexión [83].

Los juegos no solamente interactúan con los conceptos culturales de quienes los desarrollan sino también con el contexto cultural en el que son jugados.

Un ejemplo de comportamiento emergente creado a partir de referencias culturales de sus jugadores es un fenómeno que tiene lugar en el juego multijugador masivo Ultima Online. El juego sitúa al jugador en un mundo medieval fantástico con su propio sistema de economía. También da la posibilidad a los jugadores de personalizar la apariencia del personaje que controlan (avatar). Si bien no estuvo contemplado en el diseño del juego ni en las profesiones que el jugador puede elegir, los jugadores recrearon la prostitución en una estructura que denominaron “Jenny y Pimp Daddy”. Un jugador debe actuar como prostituta mientras que el otro se encarga de conseguir a los clientes y de garantizar la seguridad al primer jugador. El sexo no está contemplado como acción en el juego, pero la posibilidad de que el jugador pueda mostrar un conjunto limitado de expresiones corporales, dar y recibir dinero, y chatear, tomaron este sentido gracias al contexto cultural de los jugadores [80].

Algunos juegos ofrecen una serie de herramientas que permiten personalizar e incluso crear nuevos elementos. Los juegos como cultura abierta [80], se refieren a esta posibilidad de que los jugadores sean capaces de introducir nuevos contenidos en estos juegos. Muchas comunidades online crecen entorno a esta posibilidad, creando lugares de diálogo donde se comparten estas creaciones [59].

Los jugadores son siempre libres de crear, fuera del entorno de juego, referencias a ellos en señal de admiración. Los “fan art” son elementos creados por estos usuarios sin el objetivo de obtener ganancias. En general estos se refieren a dibujos digitales

(ver figura 4.4.1), sin embargo, también existen los denominados “*fan films*” los cuales consisten en cortometrajes y los llamados “*fan games*” que son juegos creados por aficionados intentando recrear, continuar o simplemente homenajear al original [59]



Figura 4.4.1 Fan art de Mario Bros.

Un videojuego también es capaz de transformar su entorno cultural, y cuando esto sucede, lleva el nombre de juego transformativo. La acción de jugar puede crear en el jugador nuevos conceptos cognitivos que pueden trascender el denominado “Circulo mágico”. El “Circulo mágico”, por su parte, es un concepto que define que cuando un jugador interactúa con un juego éste ingresa dentro del área que limita, y adopta comportamientos y rituales propios del juego. Una vez que la sesión de juego termina, el “Circulo mágico” se disuelve y el jugador regresa al mundo real [80]. Éste es el objetivo principal que persiguen los videojuegos serios, sin embargo, muchos juegos fueron capaces de dejar marcas culturales a un nivel más sutil.

Como consecuencias los videojuegos han influido en otros medios. Juegos como ***Resident Evil*** (Capcom), ***Toimn Raider*** (Eidos), ***Mortal Kombat*** (Midway) y ***Doom*** (Id software), entre otros, tuvieron su contraparte, con diferentes niveles de éxito, en el cine. El videojuego ***Pokemon*** (Nintendo) por su parte, tuvo gran éxito en Japón y en gran parte del mundo, con revistas, muñecos, series animadas en televisión y varias películas en el cine [11]. ***Pacman*** (Namco), ***Super Mario Bros*** (Nintendo), ***Sonic*** (Sega) y ***Donkey Kong Country*** (Nintendo) también hicieron su aparición en televisión como series animadas.

Películas tales como ***Stay Alive***, ***Gamer***, ***Scott Pilgrim vs The World***, entre otras, fueron fuertemente influenciadas por los conceptos de los videojuegos sin basarse en ninguno en particular.

Otro ejemplo de la influencia de los juegos es que en Zaragoza, capital de la Comunidad autónoma de Aragón y de la provincia homónima de España, se han utilizado a los nombres de videojuegos para nombrar las calles en un nuevo barrio. **Super Mario Bros** (Nintendo), **Sonic: The Edgedog** (Sega), **The Legend of Zelda** (Nintendo), **Space Invaders** (Atari) son algunos de los juegos que nombran las calles [84].

Los videojuegos también han mostrado su influencia desde graffitis callejeros (Ver figura 4.4.2) [85] hasta obras de teatro [86].



Figura 4.4.2 Graffiti callejero, **ExciteBike** (Nintendo)

4.5 Los videojuegos como arte.

Cada vez que una nueva tecnología provee un medio distinto en el arte, éste es considerado una falta de respeto, es recibido con indiferencia e indignación y debe atravesar un proceso de aceptación en el mundo del arte [61]. Los medios que suelen estar asociados a fines utilitarios son los que tienen mayores dificultades en este proceso dado que el arte no persigue este fin. Muchas discusiones y críticas se centraron en cuestionar si los videojuegos pueden o no ser considerados un medio artístico, y recientemente esta discusión tuvo un final formal durante el año 2011.

El Fondo nacional de Artes de Estados Unidos (NEA) publicó, en su programa para subvenciones, que los videojuegos serán considerados para el apoyo económico de dicha institución a partir del año 2012 [87]. Esto significa, según las palabras de NEA, que los desarrolladores que busquen crear juegos para las personas y distribuirlo sin la intención de ganar dinero, contarán con una opción más para poder subsistir

realizando dicha actividad[88], Al mismo tiempo, este reconocimiento, le da a los videojuegos la posibilidad de ser considerados, formalmente, otro medio de expresión artística [87].

Al igual que sucede con el cine, o la literatura, no todos los videojuegos pueden ser considerados una obra de arte. Como tal, los artgames (videojuegos artísticos) deben tener los siguientes elementos:

- Contenido: Expresar un mensaje.
- Estética: Tener un conjunto de criterios que posibilite discriminar lo bello de lo que no lo es.
- Ideas: Tener la capacidad de expresar ideas.
- Evocar sentimientos: Ser capaz de generar sentimientos en su audiencia.
- Sin fórmulas: No debe poseer un conjunto de reglas que determinen la forma en que se puede o no, hacer la obra. El proceso de creación no debe ser determinístico [61].

Algunos ejemplos de juegos artísticos son:

- **12 de septiembre**: una crítica al accionar estadounidense sobre el terrorismo.
- **Passage**: una reflexión acerca de las decisiones que se toman en la vida.
- **Judith**: una reflexión acerca de cuales son las libertades que una persona realmente tiene.

Los juegos artísticos, y los videojuegos en general también cuentan con presentaciones en diferentes exposiciones [91] y museos. En el año 2012 se planea una muestra para en el Museo Smithsonian de Arte Americano, con el motivo de explorar los 40 años de los videojuegos como medios artísticos. En él se expondrán juegos para todo tipo de plataforma y pertenecientes a diferentes períodos de tiempo [90].

En Argentina, Objeto A, una organización sin fines de lucro que organiza distintas muestras artísticas, desarrolló la denominada Game on! La cual consiste en exposiciones relacionadas con el arte de videojuegos, y videojuegos artísticos nacionales e internacionales. Entre los desarrollos nacionales se encuentran dos trabajos de Agustín Pérez Fernández, los juegos **Mantra** y **Vestigios**. También se desarrollo una Jam de videojuegos (Evento de corta duración en el cual se intenta desarrollar un juego desde el inicio) en la cual participaron varios desarrolladores independientes locales: Martin Sebastián Wain, Martin Gonzáles, Daniel Benmergui, Fernando Ramallo y Alejandro Adrián Iglesias [89].

5. Desarrollo de videojuegos.

Al comienzo, el desarrollo de videojuegos no requería de artistas, ni diseñadores e incluso hasta prescindía de programadores. En los años 1970, los primeros videojuegos eran construidos con hardware dedicado [11]. Cuando la tecnología de los microprocesadores alcanzó un nivel de costo y accesibilidad suficiente, los creadores de videojuegos comenzaron a desarrollar juegos más complejos. La necesidad de programadores se hizo presente cuando la cuando la lógica de cada juego se pasó de implementar en el hardware a implementarse mediante software [96].

En varios años consecutivos las siguientes generaciones de procesadores, siguiendo la ley de Moore⁴, crecieron en su capacidad exponencialmente, y con ellos la capacidad de representación de los videojuegos y la necesidad de artistas, músicos y sonidistas. La cantidad de años-hombre que se necesita para concluir un juego AAA⁵ ha estado en aumento desde 1980 (Ver figura 5.1) [96].

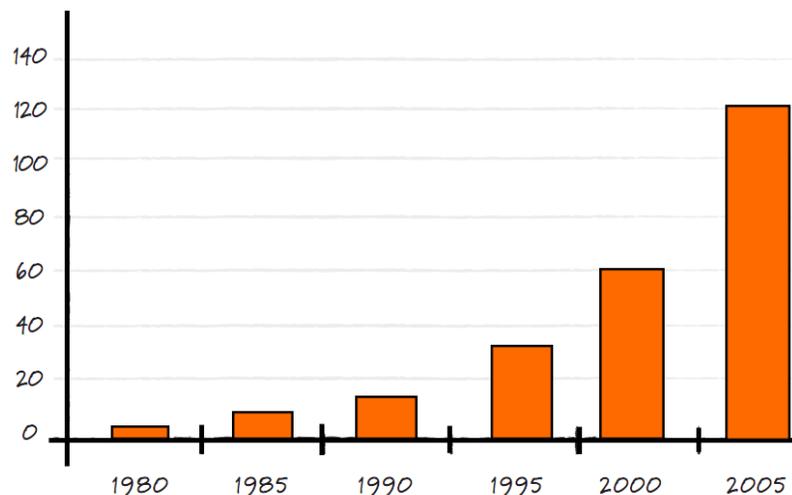


Figura 5.1 años-hombre necesarias para concluir un juego AAA [96].

Cada generación de hardware trajo consigo también nuevas capacidades: gráficos 3D, calidad de sonidos de CD y gráficos alta definición por nombrar algunos. Al mismo tiempo, la memoria y la capacidad de almacenamiento también crecieron rápidamente [11]. Hace 30 años, la Atari 2600 tenía menos de 1.000 bytes de memoria y 4.000 bytes de espacio en los cartuchos. En la actualidad, una Playstation 3 tiene 500.000

⁴ La ley de Moore predice que el número de transistores que un chip podría contener sería el doble cada 2 años.[97]

⁵ Un videojuego AAA es un juego que cuenta con un gran presupuesto, y tiene por objetivo alcanzar grandes ventas en el mercado internacional [100].

veces la memoria y 10.000.000 de veces la capacidad de almacenamiento de ésta [96].

En la actualidad el desarrollo de un videojuego es una actividad multidisciplinaria que requiere del trabajo conjunto y coordinado de productores, diseñadores de juegos, programadores, artistas gráficos, músicos y administradores de proyectos [38].

El tiempo de desarrollo de un videojuego puede variar desde un par de meses, a varios años, y los equipos pueden estar formados desde una sola persona o un equipo reducido, a más de cien personas [94][95]. Sin embargo, existe una serie de pasos, documentación, estrategias y roles que son comunes al proceso de desarrollo de un videojuego independientemente del tamaño del equipo. El objetivo de este capítulo será el de desarrollar esta información. Por otro lado, con el crecimiento de la industria, las entidades que conformaron el proceso de creación y distribución de un juego se diversificaron, y es objetivo de este capítulo también definir cuales son estas entidades y roles que intervienen en el proceso.

5.1 Entidades intervinientes.

La cantidad de entidades que intervienen en la creación de un videojuego varía dependiendo del tipo de videojuego, del tamaño del juego, del tamaño de la empresa que lo desarrolle y de la plataforma en la cual este se ejecutará. En consecuencia, se definen a continuación algunas de las entidades más importantes, sus interacciones y responsabilidades (ver figura 5.1.1) [101].

Desarrolladora: Es la entidad en la cual se transforma una idea a un videojuego. Esta idea puede ser propiedad intelectual del equipo desarrollador, del *publisher*, o de un tercero (más frecuente en el caso de los juegos serios [43]).

Dentro de esta entidad se desarrolla un diseño detallado del juego por parte de los diseñadores de juego y se traduce a líneas de código gracias al trabajo de un equipo de programadores y arquitectos de software. Dentro del equipo de desarrollo también se cuenta con la intervención de artistas y músicos que crean los contenidos artísticos del juego [11]. El proceso completo de creación de un videojuego será descrito en este capítulo con más detenimiento.

El equipo desarrollador, dependiendo de su tamaño o de sus políticas, puede tercerizar algunas de las actividades del desarrollo tales como: el arte del juego, las pruebas del mismo y la distribución [99].

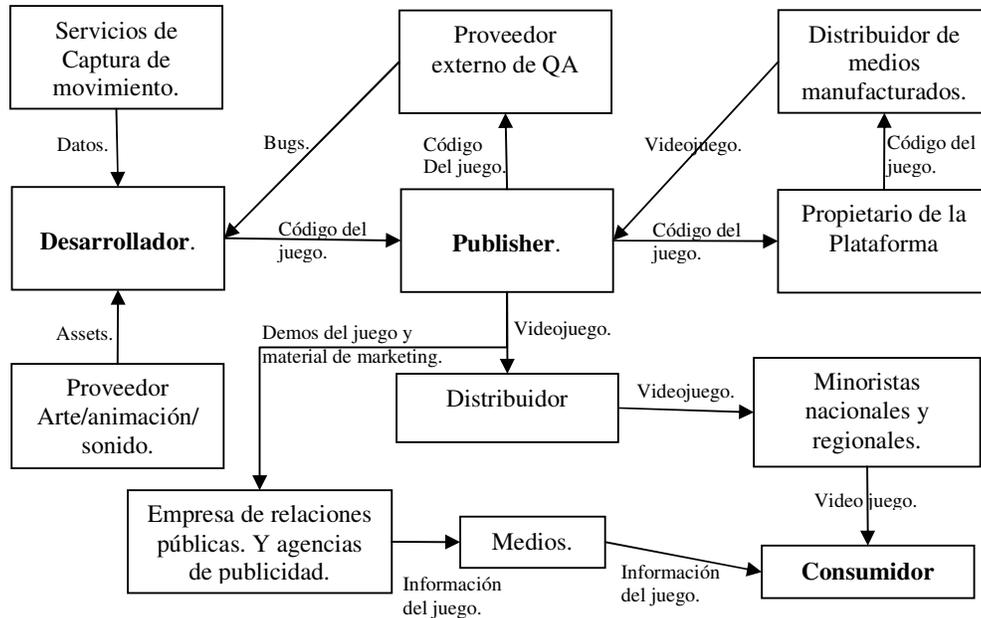


Figura 5.1.1 Diagrama de entidades y sus interacciones en el desarrollo y distribución de un videojuego para consolas [101].

Proveedor externo de QA (Control de Calidad): El juego es sometido a un conjunto de pruebas planificadas por la entidad desarrolladora, y tiene la responsabilidad de ejecutar dichas pruebas repetidas veces para poder entregarle un reporte detallado de los errores encontrados en el juego ya sean técnicos o funcionales.

Tercerización del equipo artístico: Si el estudio de desarrollo es pequeño es usual que la creación de los componentes artísticos del juego sean delegados a particulares (o estudios especializados) tercerizados. En el caso de la música y los sonidos, es más usual este tipo de estrategia ya que la creación de estos componentes no requiere de interacciones frecuentes con el equipo desarrollador.

Servicios de captura de movimiento: en el caso de videojuegos que utilicen gráficos 3D y requieran de la calidad de animación provista por la técnica de captura de

movimientos⁶, es común que la empresa desarrolladora contrate los servicios de una empresa que se dedica a ello.

Publisher: Las responsabilidades del Publisher incluyen acompañar la gestión del desarrollo del juego desde el momento de la idea hasta su puesta en el mercado. Asegurar la calidad del producto para que cumpla con los estándares del propietario de la plataforma en que se ejecutará. Distribuir el juego terminado, negociar con minoristas y publicitar el videojuego. También es el encargado de tratar con las licencias, y todo tipo de asuntos legales.

Empresas de relaciones públicas, marketing, y publicidad: el *publisher* tiene la responsabilidad de contactar con dichas entidades para lograr la promoción adecuada del juego para intentar llegar a la mayor cantidad de consumidores posibles.

Propietarios de las plataformas: son los creadores del hardware (y en algunos casos del software) sobre el cual se ejecuta un videojuego. Estos propietarios pueden incluir a empresas de desarrollos de móviles, fabricantes de dispositivos portátiles y PCs, y principalmente fabricantes de consolas. Los propietarios pueden controlar o influir el software que se ejecuta en su plataforma. También son los que proveen las interfaces de programación (APIs) para acceder al hardware que utilizan los desarrolladores del juego.

Distribuidoras: hacen de intermediarios entre los *publishers* y los minoristas para facilitar la distribución del juego. En la actualidad, sin embargo, si el tamaño del *publisher* es lo suficientemente grande y posee los contactos suficientes, no se requiere de los servicios de esta entidad.

Minoristas: Son quienes venden el juego al pormenor directamente al cliente. Deben negociar con los *publishers* o con las distribuidoras para obtener las copias del juego y conseguir descuentos. Algunos ejemplos de minoristas varían desde tiendas especializadas a tiendas de música, tiendas de películas o de productos tecnológicos hasta supermercados.

5.2 Ciclo de vida.

⁶ La captura de movimientos es un proceso tecnológico que permite registrar los movimientos de actores humanos a través de distintos sensores, almacenando esta información de forma en que pueda ser utilizada en herramientas de animación computadas [101].

El desarrollo de un proyecto de videojuego se puede definir en tres etapas bien diferenciadas (Ver figura 5.1.1) [11][95][101]. La etapa del desarrollo del concepto del juego, las tareas de preproducción, y finalmente el desarrollo del videojuego que transformara la documentación en el videojuego propiamente dicho. La porción del tamaño del equipo que trabaja durante cada una de las etapas varía, comenzando con un equipo muy pequeño, hasta crecer en su máximo tamaño durante la etapa de producción [95].

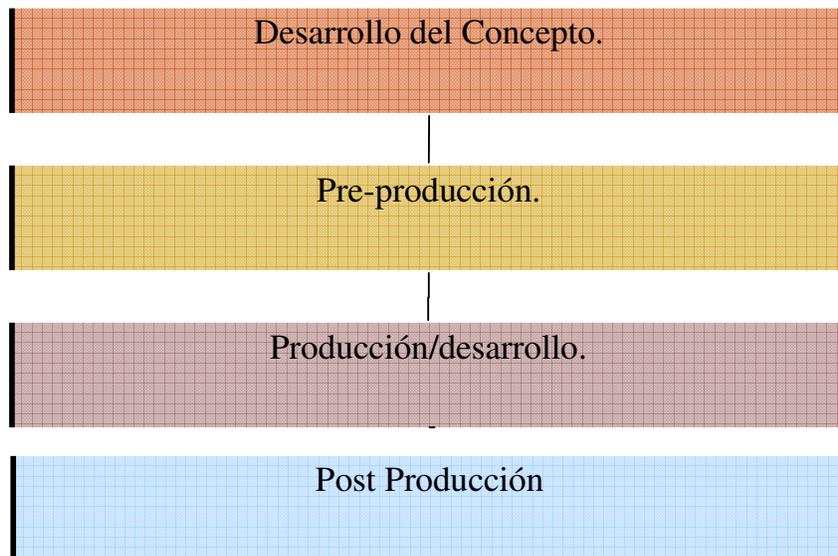


Figura 5.2.1 Etapas del desarrollo de un videojuego.

La incertidumbre en cada etapa de desarrollo decrece significativamente conforme se avanza en el desarrollo de juego. La dinámica del mercado de videojuegos ha cambiado junto con el crecimiento de la industria (tema que se tratará más detalladamente en el capítulo 7). Durante la primera etapa de desarrollo la incertidumbre existente es muy alta (Ver figura 5.1.2), es por ello, que se han implementado distintas metodologías de desarrollo. En la actualidad, el uso de metodologías ágiles tiene mayor importancia y se busca, durante las primeras etapas, reducir la incertidumbre lo más posible (a través de documentación, determinación de límites, análisis del mercado, y creación de prototipos) [96].

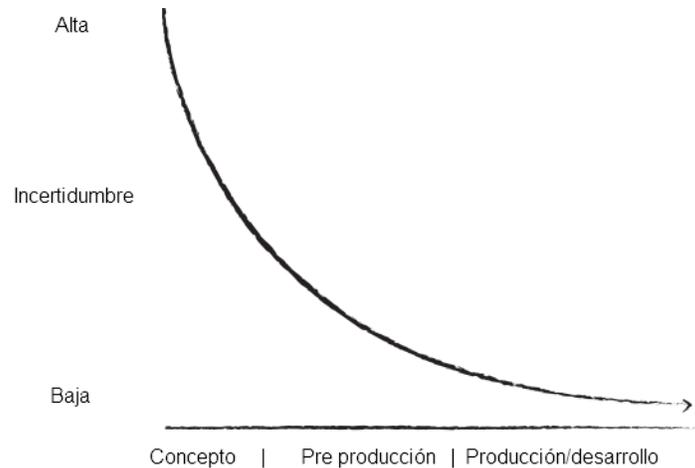


Figura 5.2.1 Incertidumbre en el desarrollo de videojuegos.

Una vez concluido el desarrollo videojuego, y una vez que este ya ha salido al mercado, comienza la etapa de post producción o post *release*. La reparación de los errores descubiertos luego del lanzamiento del juego junto con actualizaciones del mismo es entregada gratuitamente a los usuarios por medio de lo que se conoce como parches. También durante la etapa de post producción es necesario brindar a los jugadores actividades de soporte técnico para los problemas que puedan surgir y resolver las dudas de los usuarios.

En el caso de juegos MMO la etapa de post producción requiere de un equipo constante que gestione los pagos mensuales (si utiliza un modelo de negocios de suscripción, ver capítulo 7.5.4), asegurarse que los servidores tanto en hardware como en software funcionen correctamente, y realice actualizaciones y agregue nuevo contenidos para mantener el producto interesante para sus usuarios [98].

5.2.1 Desarrollo del concepto.

El equipo en esta etapa se reduce a unos pocos roles: diseñador de juegos, el líder de tecnología, artista conceptual y productor. Sin embargo la participación de todo el equipo de desarrollo puede ser útil durante el "*brainstorming*" o tormenta de ideas no solamente por la posibilidad de un aporte creativo a la idea sino también para fomentar la integración y el compromiso de todo el equipo [95].

El objetivo de esta etapa es generar la documentación necesaria para poder transmitir, en forma inmediata, la idea general del juego. Se deciden los elementos principales de

gameplay, se desarrolla el arte conceptual como forma de explicar y mostrar el estilo del juego y se realiza un bosquejo acerca de la historia del juego (si la hay). También se determina el alcance del juego y la audiencia objetiva [99].

El entregable que se debe confeccionar en esta etapa es denominado “Documento de Concepto”. Este documento, en general tiene de 10 a 20 hojas, y debe tener una muestra del arte conceptual del juego [95][35]. Su objetivo es comunicar rápidamente la idea del juego al equipo de *publishers* (o a un inversor externo) e informar al equipo encargado del desarrollo. En él se deben detallar [38]:

- **Concepto de alto nivel:** una descripción de pocos renglones para explicar todo el juego.
- **Genero:** Que genero, o cuales elementos de distintos géneros tendrá el juego.
- **Características:** características que lo distinguen de otros juegos.
- **Escenario:** describe el mundo en que se desarrolla el juego
- **Historia:** la historia (si existe), con los personajes y problemáticas principales.
- **La audiencia objetivo:** A quienes esta orientado el juego y porque puede ser interesante para esta audiencia.
- **Estimaciones de tiempos, presupuestos, balance entre ingresos y egresos:** Si bien la incertidumbre es grande durante esta etapa se debe intentar dar información aproximada sobre el costo y las ganancias que tendrá el proyecto hasta la puesta de este en el mercado.
- **Análisis competitivo:** Se debe explicar la competencia que tendrá este juego en el momento de las ventas, y debe explicarse porque éste juego es potencialmente mejor. Si el juego guarda similitudes con otro titulo exitoso anterior deben especificarse cuales son las similitudes que tiene con este, y las ventas que tuvo este otro juego.
- **Equipo:** Se debe especificar quienes son las personas que trabajaran en este proyecto, detallando credenciales y la experiencia del equipo. El objetivo

en este punto es mostrarle al *publisher* que el equipo es capaz de desarrollar el proyecto.

- **Análisis de riesgo:** Un resumen acerca de las cosas que pueden salir mal en el proyecto, y planes de contingencia.
- **Un resumen ejecutivo.**

Para desarrollos propios (sin un *publisher* externo) este documento puede realizarse en forma parcial (o incluso, si el equipo es realmente pequeño, puede obviarse). En estos casos el documento lleva simplemente el nombre de “Concepto de alto nivel”, y solamente incluye la premisa del juego, cual es la audiencia, a cuales géneros pertenece, las características distintivas, la plataforma objetiva y un resumen de la historia (si corresponde) [38].

No obstante, independiente del tamaño del equipo, si el proyecto se realizará con capital externo o con un *publisher* externo este documento debería confeccionarse obligatoriamente y con el mayor detalle posible [95].

5.2.2 Pre-producción.

El objetivo de esta etapa es determinar y planificar en detalle cuales son las tareas, el tiempo, y personas asignadas al proyecto junto con una descripción completa y profunda acerca del juego. Se crean los documentos más importantes de planificación y caracterización del juego que permitirán a los miembros del equipo conocer cuales serán sus tareas y sobre qué estarán trabajando [38].

Durante esta etapa (ver figura 5.2.2.1) se suele desarrollar elemento técnico que es creado con el objetivo de probar el juego, saber que este es tecnológicamente viable y principalmente que la idea del juego resulta divertida. Éste elemento es llamado prototipo, y no necesariamente debe estar desarrollado con la misma tecnología en que se desarrollara el juego [11].

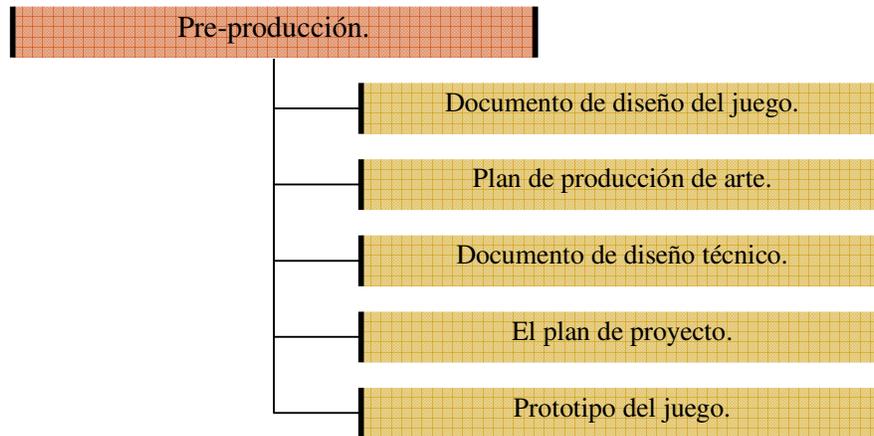


Figura 5.2.2.1 Entregables de la etapa de preproducción.

5.2.2.1 El documento de diseño del juego.

El documento de diseño del juego o GDD (por sus siglas en inglés) es el documento que con mayor nivel de detalle define al juego. Es la referencia definitiva a todos los temas relacionados con la estructura y la organización del juego, contiene todas las decisiones de diseño, y todos los aspectos del juego: la jugabilidad, la historia, los personajes, la interfaces de usuario, etc. [9].

Este documento, sin embargo, no incluye el diseño técnico del juego, es decir, contiene la parte creativa, conceptual y funcional del juego (podría incluir especificaciones técnicas solo cuando sea necesario) y no contiene ni incluye especificaciones acerca de cómo el juego debe ser implementado en el software [38]. El GDD es análogo a lo que se conoce como documento de especificación de requerimientos en ingeniería de software donde se plantean en detalle el problema que se busca solucionar y de que manera será solucionado sin especificar como será implementado.

El GDD por lo tanto está dirigido a todo el equipo que trabajará en el juego en la etapa de producción y contendrá información que permitirá [35][95][93]:

- Al equipo de programación relevar los requerimientos técnicos y realizar el diseño del software que será necesario para desarrollar el juego.
- Al equipo artístico, encabezado por el director de arte, entender la apariencia, así como el alcance de los recursos (conocidos como assets) de arte, que participarán en la creación del juego.

- A los diseñadores en el equipo, entender cuales áreas del juego requerirán mayor cuidado por parte de ellos, tales como niveles, diálogos, y eventos scripteados (Ver Glosario).
- A los diseñadores de audio, entender que efectos de sonido, voces, y música necesitarán desarrollar para el juego.
- Al equipo de marketing, entender cuales son los temas y mensajes que usarán para construir el plan de marketing.
- A los productores, comprender los variados componentes del juego que deberán desarrollar en un plan de producción.
- Al equipo de administradores ejecutivos, ser capaces de leer el documento y reconocer cual es la naturaleza del proyecto que se esta financiando [93].

Dada la complejidad del documento (dependiendo del proyecto), la extensión del mismo puede llegar a ser demasiado grande y puede resultar ilegible e inmantenible en forma de un documento de texto. Por eso es usual que se utilicen herramientas como Wikis para posibilitar el acceso y la actualización rápida para todos los miembros del equipo [38].

El documento de diseño no permanece intacto una vez que es creado sino que evoluciona constantemente a través la etapa de pre-producción [95]. El GDD se nutre de la interacción con todo el equipo.

5.2.2.2 Plan de producción de arte.

En la etapa de preproducción de arte o ADD (Art design document), se debe definir como se verá el juego, cuál es el estilo gráfico que tendrá y que contenidos deben ser desarrollado por los artistas [95]. El diseñador, el director de arte y el artista conceptual deben colaborar para definir este estilo artístico. La mayoría de los ejemplos que acompañan el plan de producción de arte pueden ser bosquejos en lápiz o incluso pueden llegar a ser imágenes de otros juegos o películas.

El principal objetivo de este plan es unificar el estilo artístico final del juego y definir que es lo que va a desarrollar cada uno de los artistas. Al igual que el documento de

diseño del juego, este documento servirá de referencia futura en la etapa de producción [38].

5.2.2.3 El documento de diseño técnico.

El objetivo de este documento es especificar un plan para transformar el diseño del juego en un componente de software para ser ejecutado en la plataforma destino. Establece quien estará involucrado en el desarrollo y estima tiempos para la finalización de cada tarea. El documento técnico también describe las herramientas que serán usadas para construir el juego (o bien que requieren ser desarrolladas), los detalles de cada requerimiento de software y hardware que serán utilizados por el equipo y quienes formarán este equipo. También define el diseño del software, el estilo de codificación y los estándares a utilizar [38].

5.2.2.4 El plan de proyecto.

El plan de proyecto es un conjunto de documentos que ayuda a los administradores de proyectos a controlar el desarrollo del proyecto, calcular costos, ganancias estimadas, tiempos y eventos importantes del proyecto

Algunos de estos documentos son [38][99]:

- **El plan de recursos humanos:** es una hoja de cálculo que tiene una lista del personal del proyecto, cuando ellos deben comenzar, cuando terminar y el salario aplicado al proyecto.
- **El plan de recursos:** Calcula todos los costos externos del proyecto. La compra de equipamiento de hardware, y las de herramientas de software, así también como todo aquello que se desarrolle externamente como podrían ser voces, música, cinemáticas etc.
- **Documento de seguimiento del proyecto:** se suele utilizar software asistido para la administración de proyectos, diagramas de Gant con camino crítico, y cualquier herramienta o mezcla de ellas para lograr hacer un seguimiento de las tareas que se deben hacer, cuales fueron hechas y cuales están atrasadas.

- **Presupuesto:** Utilizando la información del plan de recursos, y el plan de recursos humanos se calculan los gastos mensuales, y el presupuesto total de proyecto de desarrollo.
- **Balance de pérdidas y ganancias:** Se debe ampliar y refinar la definición que se hizo durante la etapa de concepto, y debe mantenerse actualizado según la evolución del proyecto.
- **Cronograma de desarrollo:** se debe mantener la información de cada una de las tareas, entregables y sus fechas para permitir a equipos de post producción saber cuando empezar con tareas de marketing.
- **Definición de hitos (o Milestone):** se deben definir fechas para los cuales eventos importantes, concretos y bien definidos del desarrollo deben ser completados.

5.2.2.5 El prototipo.

El prototipo es un resultado tangible que surge de la etapa de la preproducción [38]. Dado que el componente más importante de un videojuego es la interactividad, resulta más útil comunicar la experiencia del juego lo más tempranamente posible mediante un prototipo que solamente por medio de documentos escritos.

La utilidad de este componente no solo radica en la posibilidad de hacerlo más atractivo a la vista de un *publisher*, sino también porque permite que todo el equipo de desarrollo tenga una idea más precisa acerca del proyecto que se va desarrollar y permite hacer ajustes en las planificaciones o hasta incluso pueden surgir modificaciones en el diseño mismo del juego [38].

Durante el desarrollo de un prototipo se produce por primera vez el desarrollo funcional de los más importantes elementos del *gameplay* [11]. Sin embargo, dependiendo del proyecto el objetivo de un prototipo puede ser mostrar una aproximación a las características técnicas que tendrá el juego o a la forma en que este se verá. Es posible también que se desarrollen por separado varios prototipos para mostrar diversos aspectos del juego [38].

Por ejemplo, el juego **American McGee's Alice** (Electronic Arts) posee un estilo artístico muy particular, y un *gameplay* bastante tradicional. En la etapa de producción,

se demoró 3 meses en tener el tema y estilo visual definido. Por el contrario, el juego “*The Sims*” (Maxis Games) al contener un gameplay innovador, el diseñador del juego tardó alrededor de dos años entre investigación y desarrollos para poder conceptualizar la idea del juego [11].

5.2.3 Producción/desarrollo.

Esta es la etapa más larga y compleja en el proceso de desarrollo de un videojuego. En desarrollos pequeños esta etapa puede extenderse alrededor de 6 meses, y en proyectos grandes este tiempo puede llegar a estirarse alrededor de 2 o 3 años o incluso aún más [38].

Durante esta etapa son aplicadas distintas metodologías de desarrollo, que serán propiamente descritas en la sección 5.3, para ayudar a completar las tareas propuestas por los distintos planes desarrollados en la etapa de producción. La administración de los tiempos de producción es vital, para evitar gastos de tiempo y recursos innecesarios. No obstante en esta etapa surgen inconvenientes que no habían sido previstos en etapas anteriores y que amenazan con no poder cumplir lo pactado a tiempo [96].

El control de tareas se puede delegar para que cada uno de los implicados tenga a su cargo su propia lista de tareas para hacer. Se establecen hitos para fomentar la coordinación entre el equipo interdisciplinario y para permitirles, a los administradores del proyecto, tener una visión global del mismo. Cuando el *publisher* es externo, el seguimiento del proyecto es pactado y el incumplimiento de fechas puede significar pérdidas económicas para el equipo desarrollador.

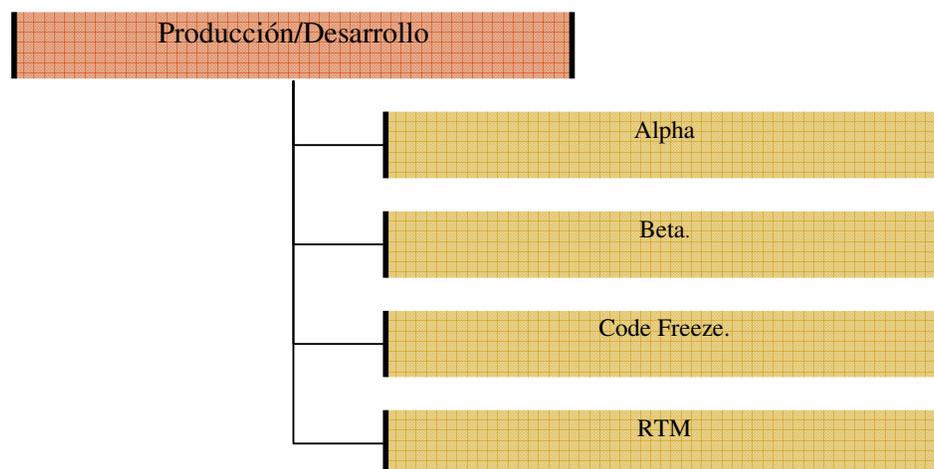


Figura 5.2.3.1 Entregables de la etapa de producción y desarrollo.

Los entregables de esta etapa pueden verse en la figura 5.2.3.1, y comprenden los siguientes:

- **Alpha:** la definición de alpha varia entre las distintas compañías, pero se refiere a un estado en el cual el juego puede ser jugado con las interfaces de usuario definidas, la mayor parte del engine funcionando y los subsistemas más importantes integrados al juego, puede que algunos assets no sean los definitivos o se encuentren incompletos, y es muy probable que pueda contener algunas fallas y errores que, en general, no arruinan la experiencia de juego [99]. Esta es la primera vez que el juego es probado por gente que no pertenece al equipo de desarrollo, y es puesto a prueba por un equipo de testadores de juegos (*gametesters* o simplemente *testers*) que deben ejecutar un plan de pruebas descrito en el documento técnico, para buscar errores, y reportarlos [38]. Dichos reportes suelen ser descripciones escritas muy detalladas, acompañadas de videos y *screenshots* (imágenes que se toman de la pantalla de juego al momento del error) para facilitar la búsqueda y reproducción de los errores. Durante esta etapa, si bien la búsqueda de errores comienza, no es el foco principal de atención de los desarrolladores.
- **Beta:** Esta versión del juego cuenta con todas las funciones desarrolladas, todos los errores que surgieron de la revisión de la versión alpha corregidos y todos los *assets* del juego incorporados. En esta versión el juego es vuelto a poner a prueba con el objetivo de descubrir nuevos errores ya que el equipo de desarrollo centrará su atención el arreglo de estos problemas [38] y aunque no existe un juego sin errores [95], se intenta encontrar y reparar la mayoría de ellos. También se busca la compatibilidad del juego entre distintas configuraciones de hardware y software, si el desarrollo es para PC, o se envía para su aprobación a los encargados de control de calidad a la/las consolas objetivo. Esta es una de las etapas más presionadas por el tiempo ya que se busca tener el producto listo para distribuirlo lo más rápido posible. Algunas empresas de videojuegos también utilizan el concepto de beta cerrada (el ya expuesto) y beta abierta (más usual en juegos multiplayer online). La beta abierta es una versión del juego que es entregada a un grupo reducido de jugadores con el objetivo de ver donde el juego podría presentar fallas. El objetivo de esta versión es probar el juego en un entorno de pruebas más

realista y encontrar estas fallas antes de que el producto sea puesto a la venta[99].

- **Code Freeze:** se fija un límite de en donde no se realizarán más cambios al código del juego más que para la reparación de errores menores. La idea es dejar el código lo más intacto posible [38].
- **RTM:** de las siglas en ingles “*ready to Manufacture*” (listo para fabricación) es la versión final del juego que será lanzada al mercado. Entre la versión beta, y la versión RTM existirán varias versiones denominadas “candidato para la liberación”, (*release Candidate*) que se ponen a prueba para saber si pueden o no ser la última versión del juego. El RTM va a ser aquel *release candidate* que cumpla con la calidad requerida para la distribución ya sea impuesta por el *publisher*, por el propio equipo de desarrollo, por el propietario de la consola a la cual se desarrolla o por todos ellos [38][95].

5.3. Roles.

Como se ha mencionado anteriormente, existe una gran variedad de roles, de diversas disciplinas, que componen el equipo de desarrollo de un videojuego (ver **Figura 5.3.1**). En esta sección se explicará la división más representativa y cada uno de los roles más importantes basándose en [98][95][38][96][117].

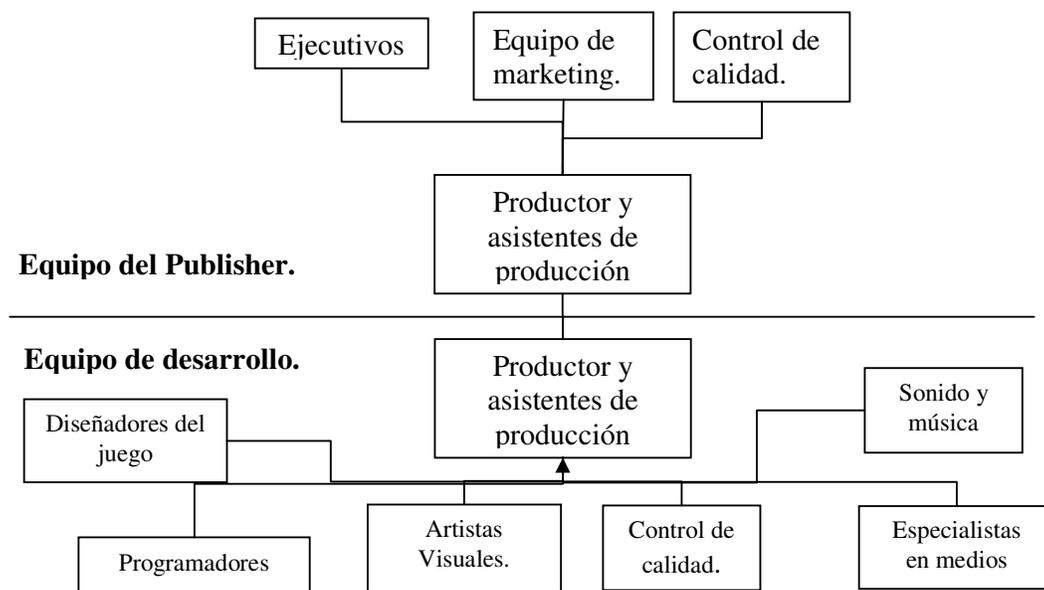


Figura 5.3.1 Equipos de desarrollo y de producción.

Una persona puede seguir más de un rol en el proyecto cuando su experiencia, su formación, y el tamaño del equipo del proyecto así lo requieran.

5.3.1 División de producción.

En esta división se incluyen los roles de administración que están directamente involucrados con la producción del juego. Los límites de los niveles más bajos de la administración tienden a ser borrosos con los roles de lo más altos niveles del equipo técnico y de diseño y esta situación es mucho más frecuente y confusa en empresas pequeñas [38].

- **Productor:** actúa como líder de proyecto, y es quien debe entregar el juego prometido al *publisher*. Tiene la responsabilidad, por lo tanto, de crear un plan de entrega, desarrollar esquemas y calendarios, presupuestos y la asignación de recursos. Funciona como nexo entre el equipo de desarrollo y el *publisher* asegurándose de que el equipo de desarrollo complete a tiempo sus tareas y se encuentre motivado. También tiene la responsabilidad de resolver problemas relacionados con la producción.

5.3.2 Diseño del juego.

En esta división se encuentran aquellos roles que tienen la responsabilidad de diseñar el juego desde el punto de vista lúdico. Principalmente se destacan dos roles:

- **Diseñador de juego (Game designer):** El rol del diseñador de juegos es diseñar todos los aspectos del juego que el equipo producirá. Genera los diseños iniciales en las fases de pre-producción, así también como el diseño detallado de todo el juego en el GDD. Debe tener una visión integradora del juego ya que su responsabilidad es hacer que el juego brinde la mejor experiencia de juego y, al mismo tiempo, debe proveerles esta visión al resto del equipo. También es quien desarrolla (junto a un equipo de programadores) y prueba los prototipos, y realiza el diseño de los niveles junto con los diseñadores de niveles [117]. Junto con el planificador de software también analiza la factibilidad de la implementación del diseño del juego.
- **Diseñador de niveles:** Tiene la responsabilidad de crear los niveles para el juego ya sea desde sus propios diseños o de diseños creados por el game

designer. Debe poseer tanto conocimientos técnicos como artísticos, y tiene relación con los equipos de arte, programación y diseño del juego. También trabaja junto al diseñador del juego para mejorar el gameplay.

5.3.3. La división de programación.

Esta división incluye a los miembros del equipo que realizan la tarea de programación del proyecto. En general, tiende a estar dividida en grupos internos pequeños de programadores que se especializan en distintos subsistemas de juego tales como: el subsistema de gráficos, el engine de IA, o los sistemas de control.

- **Planificador del software:** Tiene la responsabilidad de desarmar el diseño del juego en un conjunto detallado de requerimientos técnicos, y estimar los tiempos y el esfuerzo requeridos para implementarlos.
- **Arquitecto líder:** Tiene la responsabilidad, junto con el planificador de software, de producir un conjunto de especificaciones modulares a partir de los requerimientos técnicos. Es también, el responsable de toda la arquitectura del proyecto pero no del diseño detallado de los módulos a implementar.
- **Programador líder:** tiene la tarea de coordinar y monitorear los esfuerzos del equipo de programación para asegurarse que el cronograma de tareas sea respetado. Actúa como intermediario entre los administradores de proyecto, y reporta los progresos para su seguimiento. También es responsable de la integridad total del software, y suele pasar un tercio de su tiempo programando, y el resto tratando con temas de administración del proyecto [38].
- **Programador:** tiene la obligación de transformar las especificaciones técnicas detalladas por el arquitecto líder y el planificador del software en código que constituirá el software. Tiene la responsabilidad de trabajar en un sector particular del programa principal y su responsabilidad persiste en esa área durante todo el proyecto. También son responsables de todas las partes estándar del proceso de desarrollo de software tal como el desarrollo inicial, la prueba de unidades, las pruebas de integración y la reparación de bugs o errores [117].

5.3.4. La división de Arte.

Esta división comprende el conjunto de todos los artistas visuales que trabajarán en el proyecto. Estos incluyen: artistas conceptuales, modeladores 3D, animadores, especialistas en iluminación y texturas, etc.

- **Líder de arte:** cumple un rol similar al líder de programación, pero en una manera más definida. Un líder de arte no necesita monitorear constantemente el trabajo de los artistas (a diferencia del líder de programación), ya que la calidad de los trabajos es apreciable con mayor facilidad. Su principal tarea, es servir de intermediario entre el líder de programación, el diseñador del juego y el resto de los artistas para asegurar que el trabajo artístico producido para el juego sea el adecuado [38].
- **Artista 2D/3D/Modelador:** El rol de un artista es producir arte para uno o más proyectos. Este arte, puede comprender: arte para el juego, para el diseño de manuales, para el marketing, para el empaquetado del juego o para cualquier otra tarea relacionada. Cada artista puede realizar trabajos para distintos proyectos al mismo tiempo, respondiendo, en un mismo momento, a distintos líderes artísticos [117]. La tarea que realiza cada artista depende mucho del tipo de arte que debe desarrollar, y escapa al alcance de este trabajo.

5.3.5. La división de Música y sonido, y medios especializados.

Esta división incluye personal que produce tanto música y sonido, como otro tipo de especialistas como pueden ser estudios de captura de movimiento. En general este tipo de roles suelen desempeñarlos individuos o empresas de modo tercerizado (outsourcing) [93].

- **Músico/compositor:** Es quien se encarga de producir la música del juego. En general este trabajo suele realizarse individualmente y fuera del equipo del desarrollo, teniendo pocas interacciones con este ya que el músico solo necesita conocer la ambientación y situaciones en que la música va a ser usada [93].
- **Técnicos en diseño de sonido:** son el encargado de conseguir o producir todos los efectos de sonidos usados en el juego, con la calidad, los efectos y las características técnicas necesarias para poder ser implementadas en el juego.

5.3.6. La división de control de calidad (QA).

Esta división incluye el equipo de testeo que se asegura que la experiencia del juego sea correcta y que el juego tiene la calidad necesaria para el lanzamiento.

- **Líder de Control de calidad (QA):** Tiene la responsabilidad de supervisar el equipo de técnicos de QA, y de cooperar con el administrador de proyecto y el diseñador del juego para asegurarse se ha alcanzado la calidad suficiente desde dos puntos de vista: desde el de la jugabilidad y de la funcionalidad.
- **Técnico de QA:** tiene la responsabilidad de testear el código escrito por el equipo de programación. Cada una de las pruebas que realiza fue diseñada en por el líder de QA. Los técnicos de QA tienen conocimiento sobre el código que están ejecutando y es por ello que interactúan con los responsables de la programación de cada uno de los módulos en los que se descubrieron errores.
- **Probador de juego (game tester):** Tiene la tarea de jugar el juego con el objetivo de encontrar errores de diseño del juego (demasiado difícil, niveles mal balanceados etc.) o de programación (errores en el chequeo de colisión, *glitches* gráficos, etc.).

5.4. Metodologías de desarrollo.

En los comienzos de los videojuegos el uso de metodologías de desarrollo para proyectos de una sola persona (o de un equipo muy chico) no era ni aplicable ni necesario. Sin embargo con el crecimiento de las capacidades del hardware, el crecimiento de los equipos de desarrollo y el aumento de la complejidad de los proyectos, la falta de metodologías se tradujeron en retrasos, pérdidas, y productos de calidad inferior a las planificadas [96]. La necesidad de administrar el desarrollo de los proyectos de maneras no arbitrarias se hizo evidente y obligo a que la industria comience a utilizar metodologías de desarrollo más formales [93].

Lo que se conoce como el triángulo del proyecto (ver figura 5.4.1) establece que para todo proyecto de software solamente dos de los tres objetivos presentes en el triángulo pueden ser cumplidos al mismo tiempo en un proyecto. Los objetivos comprendidos en el triángulo son: respetar el presupuesto asignado, terminar a tiempo, y conseguir un producto de alta calidad.

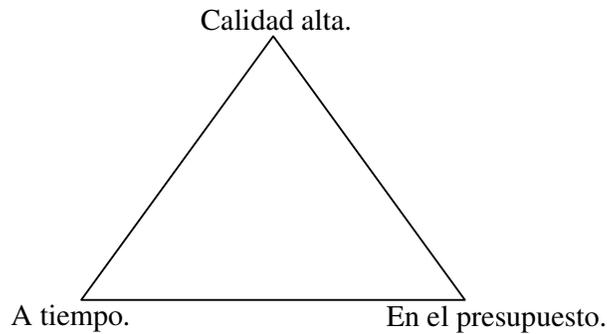


Figura 5.4.1 “The Project triangle”

Juegos como *The Sims* (Maxis) y *Ultima Online*, si bien no cumplieron con los objetivos de llegar a tiempo y cumplir con el presupuesto asignado, lograron ser juegos de una buena calidad y lograron alcanzar un nivel de críticas y de ventas exitosas. Por otra parte juegos como *Diablo* (Blizzard), *Quake* (ID) y *Baldurst Gate* (Bioware) lograron respetar el presupuesto y alcanzar una buena calidad, y al mismo tiempo conseguir un éxito similar.

Juegos como *Duke Nukem Forever* con 14 años de desarrollo (acompañado de muchos problemas, traspasos de tecnologías y diversas empresas desarrolladoras) y *StarCraft II* (Blizzard) con 8 años de desarrollo se alejaron completamente del objetivo de terminar a tiempo. Sin embargo ambos juegos cuentan con una calidad totalmente dispar y la expectativa generada por los años de espera de los jugadores subió el nivel de calidad esperado. *Duke Nukem Forever* con una calidad cuestionable según la crítica logró una aceptación muy pobre y un nivel de ventas bajo. Por el contrario, el juego de Blizzard fue un éxito tanto en ventas como en críticas convirtiéndose rápidamente en un nuevo clásico [133][134][93].

Las metodologías de desarrollo son una de las herramientas con las cuales cuentan los desarrolladores de videojuegos para combatir este problema [93]. En la actualidad existen diversas metodologías que se suelen aplicar al desarrollo de videojuegos, sin embargo, no hay recetas para saber que metodología se debe utilizar en cada juego. La elección de una u otra recae en el criterio y la experiencia de los administradores del proyecto [96]. La capacidad y las características del equipo y sus líderes, junto con la flexibilidad y la capacidad de integración de los mismos son esenciales para lograr llevar un proyecto a buen puerto independientemente de la metodología utilizada.

5.4.1.Cascada.

Una de las primeras metodologías adoptadas con el fin disminuir el riesgo de desarrollo fue el modelo de cascada tomado de la industria del software y basado en el trabajo de Winston Roce escrito en 1970 [132].

El modelo de cascada (ver figura 5.4.1.1) separa al desarrollo de un videojuego en fases bien definidas y ordenadas que abarca desde la planificación hasta el testeo del juego. Una característica de esta metodología es que cada fase debe completarse para poder continuar con la siguiente. La idea detrás del modelo es que cada fase completada reduce un poco más el riesgo y lo acerca a la meta final [96].

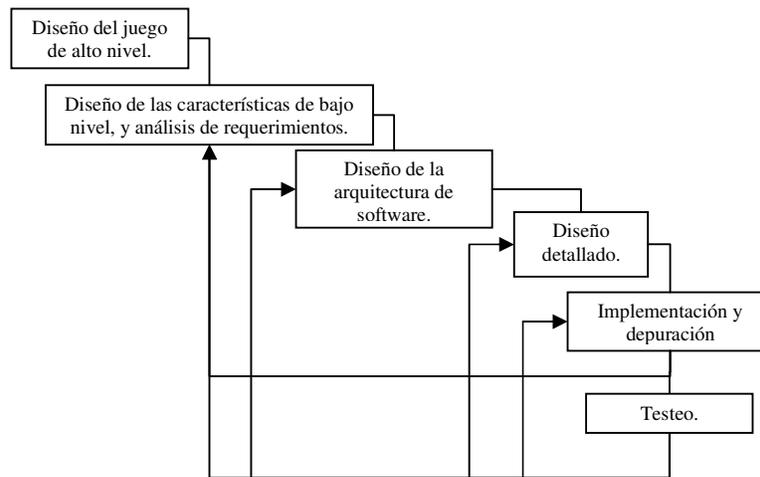


Figura 5.4.1.1 Modelo de cascada.

Dado que este modelo es poco flexible a cambios y que el éxito del desarrollo se basa en que no existan problemas o imprevistos durante el proyecto no se suele recomendar para el desarrollo de juegos en general [99]. La inflexibilidad del modelo [132] lo hace poco recomendable para proyectos en donde se requiere la exploración de nuevas mecánicas de juego que implique la necesidad de la experimentación con prototipos y ajustes continuos.

Este modelo, sin embargo, podría aplicarse en casos muy particulares como pueden ser: agregados a juegos ya publicados y la realización de secuelas de juegos que se lanzan en forma episódica, cuyos subsistemas mayores ya han sido desarrollados y probados (gameplay, engine gráfico, interfaz de usuario, etc.) [38].

5.4.2. Prototipo Iterativo.

Esta metodología es una de las mejores para el desarrollo de videojuegos con mecánicas nuevas o poco exploradas. La idea de esta metodología es obtener un prototipo funcionando lo más pronto posible para poder ponerlo a prueba, analizar los resultados y planificar los ajustes correspondientes para trabajar en el prototipo siguiente [38]. Es una de las metodologías más flexibles y permite la exploración y la prueba del núcleo del juego desde etapas muy tempranas [99].

La clave en este modelo es refinar el diseño continuamente, partiendo de las pruebas que se hacen sobre los prototipos en lo que se conoce como “*feedback* temprano”. A partir de esta retroalimentación se escriben los casos de uso o historias de usuarios para reconocer los requerimientos que emergen y deben ser considerados para próxima iteración (ver figura 5.4.2.1).

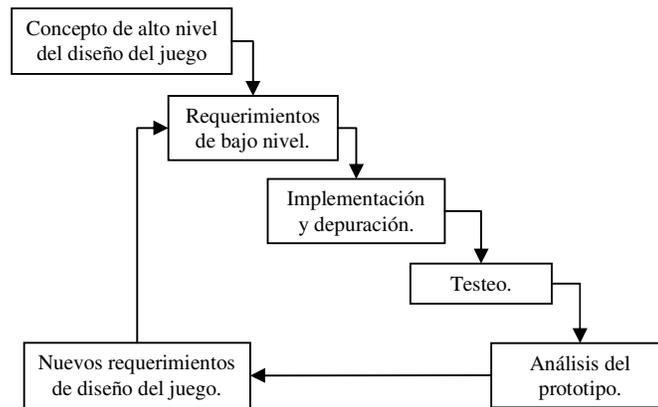


Figura 5.4.2.1 Metodología de prototipo iterativo.

Este tipo de metodología suele ser útil también en juegos de estrategia en tiempo real que requieren de un gran balanceo de dificultad y de unidades. Se crean unidades y se agregan sus características, se prueban las estrategias y las dinámicas del juego que pueden emerger, y se preparan los requerimientos en base a estos resultados para refinar las unidades o bien, para crear nuevas a partir de estas [38].

Los juegos de acción también son compatibles con esta metodología si se la combina al prototipo iterativo para el desarrollo de las mecánicas del juego, y se utiliza otra metodología distinta para la creación de los assets del juego [38].

5.4.3 Metodológica Incremental.

Esta metodología, se basa en dividir al proyecto en pequeños incrementos. El proyecto se desarrolla por partes partiendo de un concepto del juego en alto nivel, La documentación de bajo nivel del diseño del juego se crea al mismo tiempo que se desarrolla el juego, de esta manera diseñadores y programadores acuerdan cuales son las características que se pueden implementar y como deberían hacerlo [99].

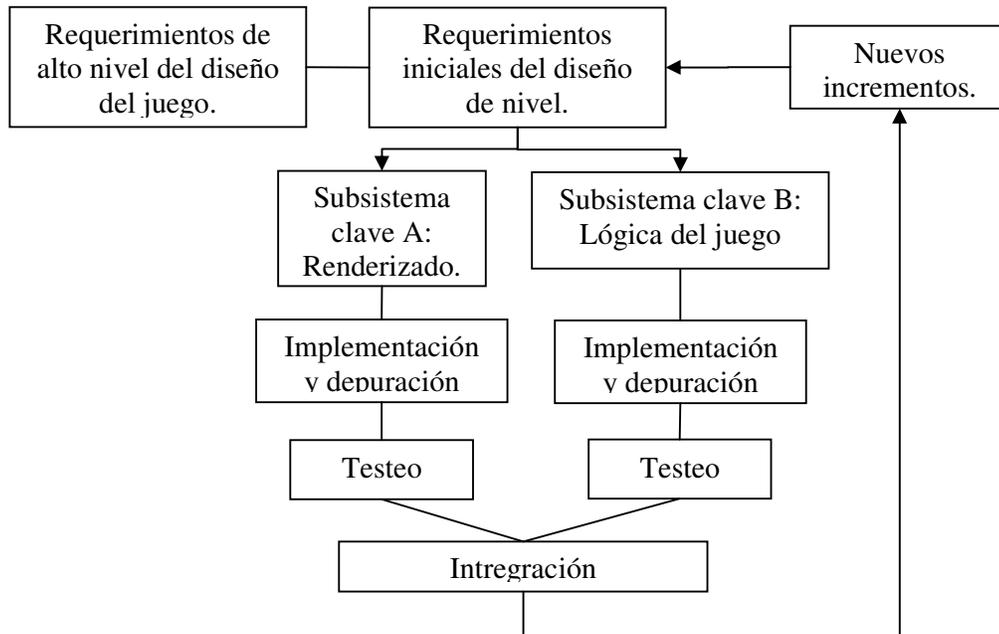


Figura 5.4.3.1. Ejemplo de subsistemas en paralelo.

Una de las ventajas de usar el modelo incremental, es que varios subsistemas pueden desarrollarse en paralelo si son lo suficientemente independientes (Ver figura 5.4.3.1) [38]. Representa un desafío, en la práctica, implementar el modelo de esta manera ya que se requiere de un gran esfuerzo de coordinación y estructuras de código lo suficientemente flexibles.

5.4.4. Metodologías ágiles.

El núcleo de las metodologías ágiles se basa en: “la satisfacción del cliente a través de la temprana y continua entrega de componentes de software útiles”. El desarrollo ágil es útil en proyectos donde los requerimientos no están bien comprendidos desde el comienzo y es necesaria la creación de nuevas tecnologías para ayudar a su descubrimiento. Esto requiere, por parte del cliente (el *publisher* generalmente), que se encuentre involucrado profundamente en el proceso de desarrollo. Estas

características definen perfectamente al desarrollo de la mayoría de los juegos de la actualidad [38].

La propuesta de la metodología ágil es priorizar los elementos del desarrollo del proyecto de la siguiente manera:

- Individuos e interacciones frente a procesos y herramientas.
- Trabajo de software frente a una documentación comprensiva.
- Colaboración con el cliente frente a un contrato de negocios.
- Responder al cambio frente a seguir un plan.

Se basa en la distribución en pequeños equipos de pequeñas secciones del proyecto que pueden ser concluidos en pocas semanas y se encuentran organizados por una línea de tiempo organizada mediante el establecimiento e hitos [99].

En la metodología ágil del Scrum, Cada equipo recibe tareas del denominado “*Scrum Master*” (un líder con experiencia en la metodología). Éste tiene la tarea de gestionar varios de los equipos pequeños, asignando proyectos y supervisando los progresos. A cada miembro del equipo se le asigna un aspecto diferente del proyecto y debe guiarlo hasta su terminación. Una vez que se establecen los deberes, el equipo comienza un “*Spring*”, el cual consiste en un proceso donde todo el equipo trabaja al unísono sobre un aspecto específico del proyecto coordinando entre sí sus tareas. Al final, cuando el ítem de desarrollo está listo, se integra dentro del juego y es presentando para el hito establecido [95].

6.0 Tecnología de videojuegos.

Los videojuegos crearon la necesidad del desarrollo de diferentes tecnologías, tanto en hardware como en software, que les permitiera a sus desarrolladores crear los videojuegos y a sus jugadores poder ejecutarlos. Desde los primeros experimentos en la década del 40' y 50' hasta el día de hoy, la evolución de esta tecnología llevó principalmente a la exploración del poder de procesamiento, del procesamiento gráfico, implementaciones de simulaciones físicas, el desarrollo de protocolos en redes de comunicación e incluso al desarrollo de interfaces de usuario innovadoras [11]. El objetivo de este capítulo es exponer entonces dichas tecnologías haciendo particular hincapié en la actualidad. Para esto, se divide el capítulo en dos partes principales: la parte de hardware que se ocupará de explicar las características del mismo; y la parte de software que se encargará de caracterizar y describir las técnicas y algoritmos utilizados para el desarrollo de videojuegos.

6.1 Hardware.

El hardware en videojuegos ha seguido diversos caminos dado que son ejecutados prácticamente en todo tipo de dispositivo con capacidades multimedia. Existen dispositivos de hardware dedicados exclusivamente a la ejecución de videojuegos, así también como dispositivos de hardware de propósito general con hardware y software heterogéneos [135][93]. La mayor parte del desarrollo de hardware está enfocado hacia el aumento del poder de procesamiento gráfico, sin embargo, algunos desarrollos se centraron también en la creación de diferentes interfaces de usuario.

Los equipos destinados a la ejecución de videojuegos, ya sean las consolas o computadoras personales, tienen la reputación de necesitar una renovación tecnológica mayor y más rápida que en cualquier otro grupo de usuarios [40]. La búsqueda para lograr representaciones gráficas cada vez más realistas ha empujado a la utilización de procesadores de varios núcleos tanto en el CPU de computadoras y consolas, como en los procesadores gráficos dedicados presentes en ellas [40][135].

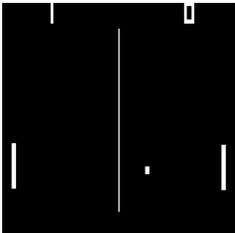
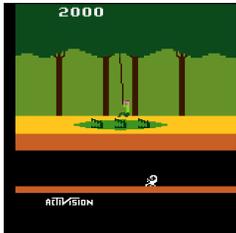
6.1.1 Consolas.

Las consolas son hardware dedicado a la ejecución de videojuegos, aunque en la actualidad también cuentan con capacidades de reproducción de contenido

multimedia. Si bien cada consola cuenta con sus peculiaridades, comparten un conjunto de características que se consideran estándares [118]:

- Utilizan como dispositivo principal de salida a un televisor o monitor (En la actualidad, las consolas pueden restringir que tipo de televisor requieren debido a la utilización de la tecnología HDMI como puerto de salida).
- En general el sistema de sonido es similar al utilizado por los reproductores de DVD con calidad Dolby Digital y sonido envolvente 5.1 (o mayor).
- Cuentan con controles que combinan botones, sensores de movimientos, y procesamiento de visión artificial.
- Cuentan con puertos USB o un sistema de entrada adicional que posibilite el agregado de más accesorios a la consola que no fueron planificados en el momento de construcción de la misma.
- Cuentan con conexión a Internet mediante diversas tecnologías, aunque la tendencia se inclina hacia la utilización de tecnología inalámbrica.

A continuación se presentará, de manera resumida, la evolución de las consolas (tomando encuentra la más representativa de cada generación) junto con una descripción técnica y ejemplos de videojuegos para ilustrar su capacidad gráfica.

Consola: Atari 2600 [119].	Generación: Segunda.	
Año: 1977	Empresa: Atari.	
Microprocesador: MOS 6507, 1.19 MHz		
Memoria: 128 bytes RAM, 6 KB ROM		
Soporte: Cartuchos con memoria ROM.		
Juegos de ejemplo:		
		
Pong.	Space Invaders.	Pitfall.

Consola: Nintendo Entertainment System **Generación:** Tercera.

[120][121].

Año: 1985

Empresa: Nintendo.

Microprocesador: Procesador Ricoh 2A03 de 8-bit 1.76 MHz

Memoria:

- 2 KB de memoria RAM on board.
- 2 KB para gráficos.
- Expansión de memoria hasta 8KB en los cartuchos

Soporte: Cartuchos con memoria ROM y RAM.

Juegos de ejemplo.



Super Mario Bros.



The legend of Zelda



Contra.

Consola: Super Nintendo Entertainment System. **Generación:** Cuarta Generación.

[122]

Año: 1990

Empresa: Nintendo.

Microprocesador: 16-bit 65c816 Ricoh 5A22 3.58 MHz

Memoria:

- 128 KB onboard.
- 64 KB para gráficos.
- 64 KB para música.

Soporte: Cartuchos con memoria ROM (opcionalmente con memoria también RAM).

Juegos de ejemplo.



Donkey Kong Country.



Yoshi's Island.



Megaman X3.

Consola: PlayStation. [123]

Generación: Quinta Generación.

Año: 1995

Empresa: Sony.

Microprocesador: MIPS R3000A-family R3051 33.8688 MHz

Memoria: 2 MB onboard.

Procesador Gráfico:

- **Memoria:** 1mb.
- **Máxima resolución:** 740x480
- **Polígonos por segundo:** 360.000.

Soporte: CD-Roms.

Soporte para partidas: Memory Card. (memoria flash)

Juegos de ejemplo.



Metal Gear Solid.



Resident Evil 2.



Final Fantasy VII.

Consola: PlayStation 2. [123]

Generación: Sexta Generación.

Año: 2000.

Empresa: Sony.

Microprocesador: 128 Bits Chip Emotion Engine 294.912 Mhz⁹

Memoria: 32 MB onboard.

Procesador Gráfico:

- **Frecuencia de reloj:** 150 MHz

- **Memoria:** 32mb
- **Máxima resolución:** 1280x1024
- **Polígonos por segundo:** 75 millones.

Soporte: DVDs/CD-Roms.

Soporte para partidas: Memory Card. (memoria flash)

Conectividad: Placa de red Ethernet.

Juegos de ejemplo.



Black.



We love Katamari.



Shadow of Colossus.

Consola: Xbox 360 [124].

Año: 2005.

Microprocesador: 3.2 GHz(PowerPC) Tri-CoreXenon

Memoria: 512 MB DDR3 onboard.

Procesador Gráfico:

- **Frecuencia de reloj:** 500 MHz
- **Memoria:** 512 MB GDDR3.
- **Polígonos por segundo:** 500 millones.

Soporte: DVDs, CDs.

Soporte para partidas: Disco duro interno.

Conectividad:

- Placa de red Ethernet
- Wifi 802.11

Juegos de ejemplo.

Generación: Séptima Generación.

Empresa: Microsoft.



Gears of War 3



Call
of duty: Modern warfare



Dante's Inferno.

Consola: PlayStation 3 [123] [125].

Generación: Séptima Generación.

Año: 2006.

Empresa: Sony.

Microprocesador: Cell Broadband Engine 3,2 GHz.

Memoria: 32 MB onboard.

Procesador Gráfico:

- **Frecuencia de reloj:** 550 MHz
- **Memoria:** 256 MB GDDR3.
- 1.8 TFLOPS.

Soporte: DVDs, CDs, Blue Rays Discs.

Soporte para partidas: Disco duro interno.

Conectividad:

- Placa de red Ethernet
- Wifi 802.11
- Bluetooth.

Juegos de ejemplo.



KillZone 3.



God of War 3.



Flower.

Consola: Nintendo Wii.

Generación: Séptima
Generación.

Año: 2006.

Empresa: Nintendo.

Microprocesador: IBM Broadway 729 MHz.

Memoria: 32 MB onboard.

Procesador Gráfico:

- **Frecuencia de reloj:** 243 MHz

Soporte: Wii Disc, 8 cm GameCube disc.

Soporte para partidas: Memoria flash interna.

Conectividad:

- Wifi 802.11
- BlueTooth.

Juegos de ejemplo.



Okami.



Super Mario Galaxy.



Shadows Tale.

Exceptuando a la consola Nintendo Wii (cuya estrategia comercial llevo al diseño de un hardware distinto) el poder de procesamiento de de las consolas ha crecido a un ritmo similar al predicho por la ley de Moore [96]. La consola PlayStation 3, por su parte, dado su gran poder de procesamiento y la posibilidad que ofrece de soportar sistemas operativos abiertos (en versiones del *firmware* anteriores al 2010), ha sido utilizada por investigadores para la construcción clusters con 8 de estas consolas y consiguiendo una gran poder de procesamiento a un bajo costo [105].

En la actualidad, las consolas que lideran el mercado cuentan en común obligatoriamente con las siguientes características de hardware:

- Gran poder de procesamiento.
- Un Procesador Gráfico con memoria destinada a él.
- Soporte de datos por discos ópticos o memoria flash.
- Conexión a Internet por Ethernet o Wifi.

- Posibilidad de agregar accesorios por puertos USB o bluetooth.
- Disco duro o memoria flash para el guardado de juegos y partidas.
- Un sistema de control innovador (tema que será tratado en la sección 6.1.4).

6.1.2 Consolas portátiles.

Las consolas portátiles, son como su nombre lo indica, hardware dedicado para la ejecución de videojuegos con un tamaño adecuado para que puedan ser fácilmente transportadas y en consecuencia, cuentan con su propia fuente de energía, pantalla, y salida de sonido. Tradicionalmente el mercado de las consolas portátiles estuvo dominado por la empresa Nintendo con sus populares consolas Game Boy y Game Boy Color [118].

Dos referentes actuales son: la Playstation Portable (PSP) de Sony y la Nintendo 3ds. La PS Vita, la sucesora de la PSP, no fue lanzada al mercado al momento de la escritura de este trabajo, pero las especificaciones de hardware son más representativas de la actualidad que las de su predecesora. Del mismo modo la consola 3DS fue lanzada este mismo año (2011) en febrero, no obstante a fines prácticos del trabajo, es más importante detallar las características técnicas de estas consolas.

La Nintendo 3ds incorpora la posibilidad de gráficos con efecto estereoscópico sin la necesidad de lentes. Tiene 2 pantallas, una de ellas táctiles, y la otra con el efecto estereoscópico anteriormente mencionado. Tiene un almacenamiento interno con memoria flash de 2GB, y puede ser expandida con memorias flash SD. Cuenta con un microprocesador de doble núcleo ARM de 266 Mhz y un procesador gráfico PICA220 de 133 Mhz [129]. También incorpora conectividad Wifi 802.11, infrarrojo, e incorpora dos cámaras para permitir tomar fotografías también con efecto estereoscópico.

La Ps Vita, por otro lado, cuenta con un procesador de 4 núcleos de 800Mhz y un procesador gráfico de 200 Mhz que le permite mostrar 133 millones de polígonos por segundo. Cuenta con una pantalla táctil, y con un reverso táctil también, junto con acelerómetros y giroscopios para permitir deducir la inclinación y la posición de la consola [128].

6.1.3 PC.

El análisis del hardware actual en equipos de escritorio y equipos portátiles escapa al alcance del trabajo ya que es extenso y heterogéneo. Sin embargo, este mismo desafío es con el que se encuentran los desarrolladores de videojuegos para PC, no solamente con el hardware sino también con el software [118].

Las herramientas con que cuentan los desarrolladores para combatir en mayor o menor grado esta heterogeneidad son: los engine de videojuegos y las librerías estandarizadas para el manejo de gráficos, de interfaces de dispositivos de entrada y de sonido [40][118]. Este tema será tratado con mayor detenimiento en las siguientes secciones.

En cuanto al hardware especializado en videojuegos en computadoras personales se encuentran las denominadas placas aceleradoras de video [40]. Antes de la creación de dichas placas, la responsabilidad de actualizar individualmente cada píxel presente en la pantalla era responsabilidad del microprocesador principal. El objetivo de estas placas, por lo tanto, es liberar al microprocesador principal de esta tarea y acelerar la capacidad de procesamiento gráfico en tiempo real para conseguir experiencias visuales más complejas y realistas [135].

Cuentan con un microprocesador gráfico (GPU) especializado en operaciones de cálculo en punto flotante necesarias para el procesamiento de imágenes y con un conjunto de instrucciones especializadas en el tratamiento de matrices y texturas. Se conectan al motherboard de la computadora por medio de interfaces AGP, o más común en la actualidad, por medio del puerto PCI-Express.

Las placas de video también cuentan con su propia memoria de trabajo y se intenta que la interfaz de la memoria sea de una tecnología que permita la mayor transferencia posible en gigabytes por segundo. Usualmente se utilizan bus de tecnologías como DDR2, GDDR3, GDDR4 y GDDR5, para permitir el manejo de texturas y resoluciones más grandes.

Las primeras generaciones de placas de videos contenían un conjunto finito y estático de instrucciones para operar gráficos, sin embargo, hoy en día los GPU cuentan con un conjunto de instrucciones mucho más grande y son programables permitiendo aprovechar mejor el poder de cálculo de estos procesadores [135]. Los programas

escritos con instrucciones de los GPU son denominados Shaders, y tienen por objetivo realizar transformaciones en la representación visual de los elementos gráficos. Gracias al gran poder de procesamiento de las GPU, también existen técnicas para enmascarar datos de cualquier índole en texturas, para poder procesarlos como si fueran gráficos y así aprovechar este poder de procesamiento para otros fines [108]. Este tema también será tratado con mayor detenimiento más adelante.

El mercado de las chips gráficos se encuentra dominado por las empresas Intel/AMD con un 40.6% del mercado, NVIDIA con el 59%, mientras que el resto de las empresas ocupan solamente un 0.4% del mercado [107]. Las GPU o chips gráficos actuales cuentan con decenas y hasta centenares de núcleos de proceso por placa y con frecuencias de reloj que varía de 500 a 800 Mhz, y con una memoria varía de 128MB a los 8GB.

La mayor parte de los videojuegos para PC están orientados a su ejecución sobre el sistema operativo Windows. Si bien existen juegos para otros sistemas operativos basados en Linux, por lo general, estos son versiones portadas de otros videojuegos de Windows [118].

La cantidad de juegos disponibles para Mac, si bien no alcanzan a los existentes para Windows, es bastante grande, y a diferencia de lo que sucede con las computadoras personales, el hardware de una Mac es mucho más homogéneo facilitando el desarrollo para esta plataforma [118].

6.1.3 Móviles.

Los teléfonos celulares o móviles, han experimentado un rápido crecimiento tecnológico que fue transformando su finalidad solamente de comunicación a ser un centro multimedia [118]. Las características de software y de hardware presentes en cada dispositivo, al igual que sucede con las computadoras personales, es heterogéneo. Existen 3 grandes divisiones que engloban la mayor parte de los juegos para los teléfonos celulares: Teléfonos celulares con una máquina virtual de Java 2 Micro Edition (J2me), los dispositivos con el sistema operativo de Google Android, y por último los dispositivos de Apple: Iphone, Ipod y el Ipad [111][130]. Existen otras plataformas para dispositivos móviles tales como Windows Mobile, y Brew, pero son plataformas poco extendidas y por lo tanto poco utilizadas [130].

Los teléfonos celulares con j2me, son variados y si bien cuentan con una máquina virtual para solucionar el problema de la heterogeneidad, cada modelo de celular (además de tener distinto hardware) puede implementarla de una manera distinta separándose del estándar y dificultando el desarrollo de las aplicaciones. La respuesta por parte de las empresas que se dedican a desarrollar juegos utilizando j2me es la realización de “ports”. Un “port” consiste en una adaptación del software, en este caso el juego, para que funcione bajo una condición de hardware/software distinta en la cual se encuentra implementada [130][131].

Por lo general los dispositivos con j2me cuentan con hardware de poco poder de procesamiento y almacenamiento, más reducidos aún, si son comparados los denominados “*smartphones*” los cuales utilizan Android o el sistema operativo propietario de Apple [130].

Un *smartphone* cuenta con un poder de procesamiento similar al de una netbook, con un microprocesador que varía de 800Mhz a 1 GHz, y memoria RAM entorno a los 512 MB, y almacenamiento por medio de memorias flash desde los 4 a los 32GB. Todos ellos cuentan con conexión a Internet ya sea por medio de Wifi 802.11 o tecnologías 3g o 4g (estándares de comunicación de los teléfonos celulares) [109][110].

Tanto Android como Apple, poseen un mercado online donde se adquieren las aplicaciones y juegos. En Android es conocido como Android Market, y en Apple, el Apple Store. La adquisición de los juegos se realizan a través de estas tiendas online, y sus características serán detalladas más adelante en el capítulo 7.

6.1.4 Interfaces.

A lo largo de las generaciones de consolas se han desarrollado diversas formas de controlar los juegos. Durante la séptima generación en particular, los fabricantes de consolas, centraron su atención el desarrollo de nuevas interfaces bajo la premisa de hacerlos más interactivos incluyendo información del entorno, del usuario y su posición en el espacio.

Los controles, por lo general, constan de una colección de botones de diversas formas que ajustan su disposición a convenciones implícitas (Ejemplo: los botones para mover al personaje se ubican siempre a la izquierda [118]) y sobre un soporte ergonómico

para hacer más cómodo su uso. Los botones simplemente cuentan de dos estados: presionado y no presionado [115].

El joystick de la consola Nintendo 64 introdujo el concepto de analógico, donde el valor devuelto es continuo y permite saber “cuan presionado” está el botón. Este tipo de botones se utilizaron en los joystick posteriores también en forma de palancas (Ej.: los joystick de PS2 y el de Xbox). El joystick de la Xbox 360, por su lado, incluyo también valores analógicos para los botones ubicados en la parte superior del joystick denominados triggers o botones de hombrera [116][118].



Figura 6.4.1.1 Wii mote.

El Wii mote (ver figura 6.4.1.1), el joystick de la consola de la Nintendo Wii, posee la característica de interpretar la posición, la inclinación y el lugar en donde el joystick está apuntando. Para lograrlo hace uso de acelerómetros, giroscopios y sensores que le dan la posibilidad de detectar:

- La inclinación y la rotación hacia arriba y abajo
- La inclinación y la rotación en sentido izquierda y derecha
- Rotación a lo largo del eje principal.
- Aceleración hacia arriba y abajo.
- La aceleración de la izquierda y derecha
- Aceleración hacia la pantalla y la distancia hacia ella [112][113].

El control de Nintendo, basó su tecnología en componentes electrónicos ya existentes para obtener esa información del entorno. Por otro lado Microsoft, si bien también

desarrollo un periférico con un objetivo similar, se basó en una tecnología completamente distinta: la visión artificial y el reconocimiento de voz.

El accesorio Kinectic, de Microsoft, está compuesto por un conjunto de 4 micrófonos, una cámara a color VGA y una cámara monocromo CMOS que permite capturar imágenes (y profundidad) independientemente de los niveles de luz. También cuenta con un software de procesamiento basado en máquinas de aprendizaje para permitirle al dispositivo reconocer rostros, voz, y 18 partes del cuerpo distintas por cada persona presente [114].

En sensor Kinectic, no solamente abrió la posibilidad a nuevas interfaces de juego, sino también a una serie de “hacks” (modificaciones no autorizadas al software) para realizar experimentos de visión artificial [108].

Existen también otros tipos de interfaces utilizados en videojuegos como lo son: guitarras (joystick con formas de guitarras), tambores, micrófonos, volantes, pedaleras, cajas de cambios y pistolas [118]. Pero son utilizados bajo condiciones y juegos particulares.

6.2 Software

La programación de videojuegos involucra, al igual que toda disciplina, conocimientos de diversas áreas de estudio: computación gráfica, redes, inteligencia artificial, simulaciones físicas y procesamiento de imagen y sonido, entre algunas de las áreas más comunes [98].

En esta sección se explicarán las herramientas y las técnicas de software utilizadas en el desarrollo de videojuegos para solucionar los problemas comunes de implementación y facilitar el desarrollo de los juegos.

6.2.1 Breve reseña histórica.

El desarrollo de videojuegos, desde el punto de vista del software, ha experimentado también cambios conforme lo hizo la capacidad de las consolas y computadoras personales. El hardware limitado en la década de los 80 obligó a que, en búsqueda de performance, las herramientas utilizadas para el desarrollo del hardware fueran lo más cercano posible al código que interpretan los microprocesadores. La utilización de

assembler para escribir los juegos y la mezcla de datos, la lógica del juego y del manejo de las entradas y salidas era frecuente en los desarrollos de videojuegos [98]. El desarrollo del código del juego, del diseño del mismo, y hasta del desarrollo del arte del juego era producido generalmente por una sola persona [136][11].

El desarrollo de videojuegos para consolas fue posible para empresas externas a los fabricantes, a partir del momento en que un gran catálogo de videojuegos representaba una ventaja comparativa frente a competencia [136]. Sin embargo hasta el día de hoy (y a partir del *crash* de la industria comentado en el capítulo 2), el desarrollo para consolas ha sido controlado por las empresas propietarias de consolas. La utilización de *assembler* acompañó al desarrollo de videojuegos en consolas hasta la llegada de la PS2, la cual fue la primera en usar principalmente al lenguaje C para el desarrollo de sus videojuegos [98].

La separación de los datos (tales como audio, mapas, y gráficos) del juego en sí se hizo posible a partir de la utilización de los sistemas de archivos [136], sin embargo, la separación más importante de la lógica del juego y del motor del juego propiamente dicho (*game engine* o simplemente *engine*), se logró un tiempo después [98]. La primera aproximación fue el desarrollo de una herramienta conocida con el nombre de SCUMM desarrollada y utilizada por Lucas Arts para generar una parte de los datos que determinaban, de cierta manera, la lógica del juego. El SCUMM permitió a los desarrolladores de aventuras gráficas definir escenarios, diálogos, y objetos para poder realizar las aventuras populares de dicha empresa. El juego **Doom** de Id software fue uno de los primeros en separar completamente la lógica del juego del *engine*; y al mismo tiempo uno de los primeros en programarse bajo el lenguaje C en su gran mayoría. Esto permitió que varios juegos tales como **Hexen** y **Heretic**, fueran creados a partir del mismo *engine* reemplazando el archivo de datos de la lógica del juego bajo el formato WAD [136].

6.2.2 Estado del arte.

Hoy en día el desarrollo de videojuegos ha avanzado lo suficiente para que la cantidad de herramientas existentes logren facilitar la tarea del desarrollo [98]. Por un lado existen herramientas de software destinadas a la edición y creación de gráficos, sonidos, música, niveles, texturas, modelos 3D, animaciones, etc.; mientras que, por el otro lado, el concepto de reusabilidad (utilizar software de otros proyectos e incluso de

otras empresas) brinda posibilidades que permiten reducir los costos y tiempos de desarrollo.

La utilización de motores de juegos (*game engine*) popularizado por ID software[98] hoy en día ha avanzado a un nuevo nivel desarrollando todo tipo de *middleware* para solucionar distintos tipos de problemas tales como IA, manejo de gráficos, simulaciones físicas, lógica de red, etc [40]. La idea detrás de la fragmentación en la creación del software para videojuegos es que especialistas en cada área sean los encargados de estos desarrollos [137].

Existe la creencia de que la utilización de las mismas herramientas para el desarrollo de los juegos tenga como consecuencia que todos los videojuegos se vean y sientan iguales [98][136][40], sin embargo el uso de un *game engine* es, hoy en día, una practica común en el desarrollo de juegos [40].

La necesidad de conseguir performance, si bien sigue siendo una necesidad común en la mayoría de los videojuegos, no es primordial para todos los desarrollos [40]. Dependiendo del proyecto, de la plataforma, y de la audiencia objetiva se puede prescindir de lenguajes y herramientas de alta performance y cambiarlos por lenguajes que aceleren el tiempo de desarrollo de los videojuegos gracias al aumento de la capacidad de hardware, y la demanda de juegos sencillos como pueden ser los juegos hechos en flash o bien juegos casuales, tanto para plataformas como PC, consolas, y *smartphones* [40][49].

La cantidad y variedad de plataformas existentes crearon también la necesidad de herramientas que permitan independizar el software del juego de la plataforma. Herramientas como Unity, CryEngine, UDK (Unreal development kit), GameBryo, Vision engine, entre otros, permiten compilar los proyectos para que funcionen en más de una plataforma a partir del mismo código fuente. En el caso de Unity permite exportar por ejemplo: a Windows, MAC, Iphone, Ipad, Wii, Xbox360 y PlayStation 3 [138].

6.2.3 Estructura principal de un videojuego.

Un videojuego, desde el punto de vista del software, forma parte de los denominados sistemas de tiempo real [136]. Este tipo de sistemas tiene la característica de considerar al tiempo como un requerimiento crítico. Si el sistema no actúa conforme a la restricción temporal el sistema corre un alto riesgo de fallar (no satisfacer con el objetivo para el cual fue desarrollado) [136][132]. Dada su naturaleza, en este tipo de sistemas se busca que el tiempo que tome pensar, procesar los datos y dar respuesta sea el mínimo, y es por ello que se da preferencia a la eficiencia en el uso de la CPU frente al uso de memoria Ram [139], un criterio que en el desarrollo de videojuegos marcará en mayor o menor grado el diseño de las estructuras y de los algoritmos que componen los juegos.

Los videojuegos cuentan, además, con la característica de ser interactivos y de contener, al mismo tiempo, simulaciones de mundos virtuales [136]. Para poder resolver estas características, la abstracción que es generalmente utilizada en videojuegos se denomina *game loop* (ver figura 6.2.3.1) [136][98]. Esta estructura consiste en un ciclo en el cual se ejecutan, de forma ordenada, los siguientes procesos: procesamiento de los dispositivos de entrada, procesamiento de los elementos que componen el mundo virtual y la representación en pantalla del estado actual del mundo simulado.

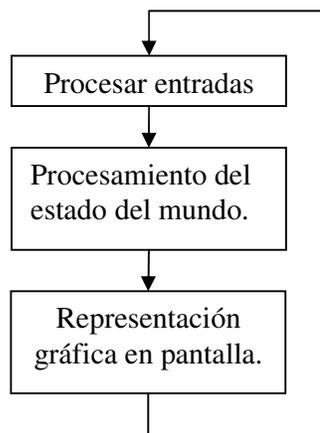


Figura 6.2.3.1 El *game loop*.

Otro criterio que marca el diseño técnico de los videojuegos, tal como se ha mencionado anteriormente, es buscar la independencia con el hardware. Las secciones más dependientes son el procesamiento de la representación gráfica y el procesamiento de los dispositivos de entrada [40].

Dada la heterogeneidad del hardware y que la cantidad de elementos a procesar no es siempre es la misma, el tiempo que demora la ejecución de estos procesos no es constante. Para asegurar una experiencia de juego homogénea se debe intentar controlar o bien, manejar, estas variaciones de tiempo. Existen al menos dos técnicas para solucionar este problema y varias formas posibles de implementación [98][136][139].

Una de las posibles soluciones conocida como “*fixed step*” que consiste en determinar una duración máxima para cada *step* (cada ejecución del *loop*). Al comienzo del *step* se debe tomar una medida de tiempo, y al finalizar, se debe calcular con este valor, el tiempo transcurrido desde el comienzo del *loop*. El juego esperará hasta que el tiempo haya sido mayor o igual al predefinido para continuar con el siguiente [136] (Ver figura 6.2.3.1). El tiempo predefinido usualmente se fija para que el videojuego muestre 60 o 30 cuadros por segundo (FPS). El tiempo de espera se define entonces como $1/\text{frecuencia}$ deseada de fps.

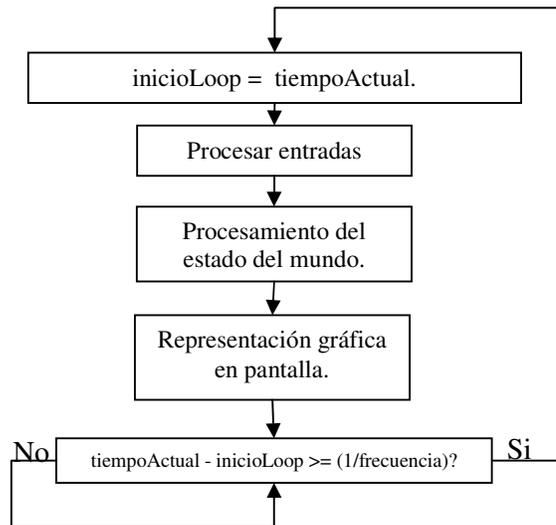


Figura 6.2.3.1 El game loop con *fixed step*.

El problema de esta técnica es que si el videojuego tarda más en procesar que lo esperado (superando el tiempo predefinido), el juego presentará una tasa de cuadros por segundo (fps) variable e interrumpirá, inadecuadamente, con la experiencia de juego. Otra consecuencia es que no se aprovechará por completo la capacidad del hardware. Esta técnica fue mayormente utilizada para videojuegos de consolas donde las características de hardware son totalmente homogéneas, sin embargo, también se

ha utilizado en videojuegos en computadoras personales. En la actualidad, esta técnica, no es utilizada con tanta frecuencia [98][136].

La técnica llamada “tiempo delta variable” busca lograr que la ejecución de cada *step* se haga tan rápido como sea posible, de manera que no se determina una duración mínima para cada ejecución del *loop* [98]. Para asegurar que la velocidad con que se mueven los objetos y se actualizan las animaciones en el juego sea homogénea se debe calcular el tiempo que el juego demora en la ejecución de cada *step* (Ver figura 6.2.3.2). Este tiempo, denominado tiempo delta, determina la velocidad de movimiento de los objetos y de las animaciones. Las velocidades se basan entonces en función del tiempo y no de los fps.

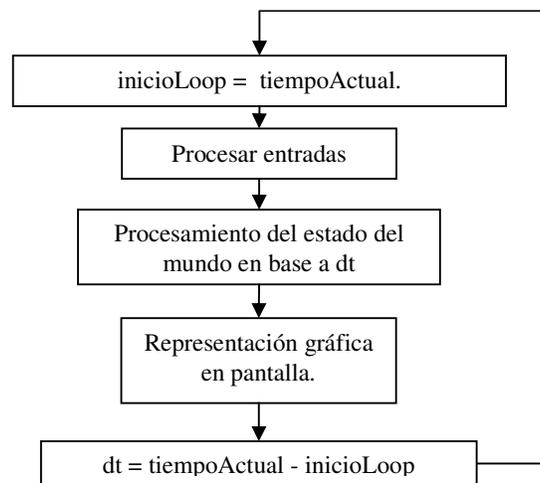


Figura 6.2.3.2 El *game loop* con tiempo delta.

Sin embargo, las consecuencias de utilizar esta técnica surgen en la detección de colisiones entre objetos cuando el tiempo delta es muy grande. Es decir suponiendo que un objeto viaja a una velocidad de 300 pixeles por segundo, y se encuentra a 5 pixeles de un objeto, si el tiempo delta fuera de 0.02 segundos, el objeto lo atravesaría sin haber detectado la colisión (ver figura 6.2.3.3) [139]. Existen soluciones a este tipo de problemas pero en general se pueden presentar o no dependiendo de las características de cada juego.

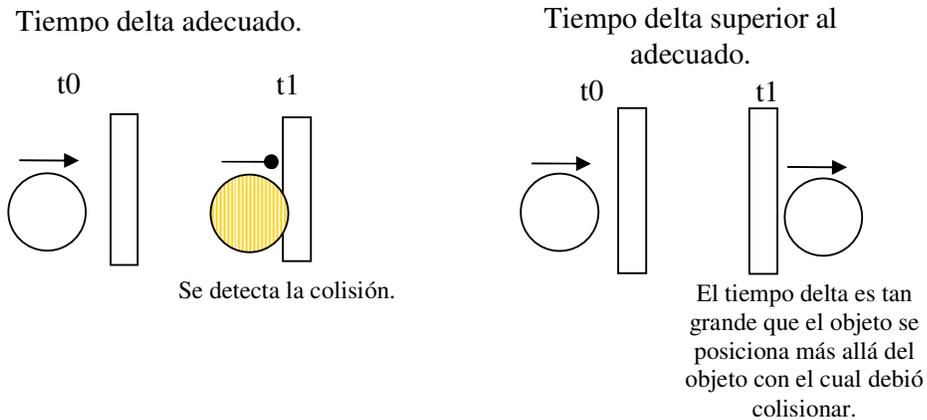


Figura 6.2.3.3 Error de colisión por tiempo delta superior.

Una última opción surge de la combinación de utilizar las dos técnicas anteriormente explicadas. Se busca un valor máximo al cual el tiempo delta permita que el juego se ejecute con normalidad. Este valor máximo se puede fijar en base a un análisis matemático o en base a la experimentación. Una vez fijado, se utiliza como tope, y si el tiempo delta lo supera, entonces, la ejecución del procesamiento del estado del mundo se realiza N veces hasta que se alcanza el tiempo delta real [139].

6.2.3.1 Procesamiento de entrada.

En todos los juegos, la necesidad de una sensación confortable al utilizar el dispositivo de entrada es fundamental [138]. Como se ha mencionado anteriormente, la dependencia del hardware es grande debido a que éste es muy variado [40]. Los elementos que se pueden llegar a utilizar como dispositivos de entrada, normalmente abarcaban:

- Teclado.
- Mouse
- Joystick.

Sin embargo, en la actualidad esta lista se ha ampliado:

- Pantallas táctiles.

- Visión artificial.
- Reconocimiento de voz.
- Acelerómetros y giroscopios.

No obstante todos los dispositivos de entrada se basan en indicadores de presión, posición, y orientación, y la tarea dentro del procesamiento de los dispositivos de entrada es, además de capturar las entradas, abstraerse las características del dispositivo y transformarlas en acciones del control que sean importantes para el juego. Por ejemplo, en el caso de la utilización de un teclado, la tecla de la letra “A”, puede ser referenciada dentro del juego como la “tecla disparo”.

En el caso del teclado, existen dos formas de leerlo. Por un lado la lectura de cadenas de texto (como el que se produce al escribir dentro de un campo de texto), y por otro lado, la capacidad de obtener el estado de cada tecla individualmente.

Es usual que se creen capas de abstracción sobre el teclado para crear eventos como “una tecla acaba de soltar” (*just release*) o “una tecla se presiona y antes no estaba presionada” (*just pressed*), para lo cual es necesario mantener en memoria el estado actual del teclado, junto con el estado del mismo en el *step* anterior. El manejo en el caso del joystick es muy similar. No obstante el joystick también puede contar con valores analógicos, y en estos casos, es usual (para determinadas ocasiones) fijar un umbral para el cual se considera o no, que se apunta hacia una dirección o bien se pulsa un determinado botón [138].

En el caso del mouse, al igual que en una pantalla táctil, también se necesita obtener información sobre la posición que los sensores obtienen. La diferencia principal entre una pantalla táctil, y el mouse (además del concepto de *multitouch*) es que el mouse puede sensor su posición todo el tiempo, mientras que una pantalla táctil se maneja mediante eventos donde solo se puede conocer la posición una vez que se haya hecho contacto con el dedo o el lápiz. Es frecuente en ambos casos implementar abstracciones para situaciones como el “*drag and drop*” (arrastrar y soltar) usual en el manejo de interfaces.

Cada plataforma debe proveer una abstracción para acceder a los dispositivos de entrada ocultando la complejidad de los *drivers* (programas escritos en lenguajes de

bajo nivel con el objetivo de controlar y hacer posible la comunicación con un elemento de hardware y el sistema operativo). Por ejemplo para usar joystick en la plataforma Windows se puede hacer uso de la API Win32, o bien de DirectX Input (la explicación de DirectX será considerada en la sección 6.2.4.6). Utilizando estas librerías, el juego puede utilizar de interfaz diferentes joystick y, al mismo tiempo, no tener la necesidad de conocer la marca o modelo particular del mismo [138][40].

6.2.3.2 Procesamiento del estado del mundo simulado.

La cantidad e índole de elementos a procesar para simular el mundo de juego, son variados y comprenden elementos como:

- **Procesamiento del jugador:** Actualizar el estado del avatar del jugador conforme a las reglas del juego, las restricciones y los datos de los dispositivos de entrada.
- **Procesamiento de los personajes y elementos que no son el jugador (NPC):** aplicar las restricciones y reglas del juego en los elementos que no son el jugador, tales como enemigos, puertas, llaves, etc.
- **Inteligencia Artificial:** Ejecutar los algoritmos de inteligencia artificial utilizados por los elementos controlados por el juego.
- **Cálculo de colisiones:** Verificar cuando existe colisión entre los objetos del mundo y realizar las acciones correspondientes.
- **Procesamiento de simulaciones físicas:** Aplicar las restricciones de la simulación física para actualizar la posición de los objetos en el mundo (tanto del avatar del jugador como de los elementos controlados por el juego).
- **Procesamiento de la lógica del juego:** Aplicar las reglas del juego a todos los objetos y su relación entre ellos. Por ejemplo, se debe chequear, si existe, la condición de victoria en el juego.

Estos temas serán tratados en las secciones siguientes individualmente.

6.2.3.3 Procesamiento gráfico.

El procesamiento gráfico tiene por objetivo principal mostrar en pantalla una representación gráfica del mundo simulado. Los algoritmos utilizados difieren si el juego utiliza una representación en 2D o en 3D. Sin embargo, sea cual sea la representación, el procesamiento gráfico (al igual que el procesamiento de las entradas) guarda cercanía con el hardware sobre el cual es ejecutado. Para ocultar esta complejidad es frecuente la utilización de librerías diseñadas para tal fin, ocultando la heterogeneidad tecnológica y ofreciendo una interfaz más abstracta. En la actualidad las API gráficas más utilizadas son OpenGL y DirectX [138][98].

Los elementos utilizados para la representación gráfica usualmente son creados por artistas (o bien son generados proceduralmente) por lo que el juego deberá ser capaz de interpretar distintos formatos de archivos ya sea para texturas e imágenes, o modelos 3D y sus animaciones. Durante el procesamiento gráfico, sin embargo, todo el procesamiento realizado debe comprender solo los esfuerzos requeridos para mostrar la representación en pantalla ya que el resto de las tareas se ejecutan en el procesamiento la simulación del mundo. Los archivos deberán estar cargados en memoria previamente, bajo la estructura apropiada, para evitar demoras.

6.2.4 Representación 2D.

La representación en 2D fue ampliamente utilizada en el mundo de los videojuegos antes del auge de las representaciones 3D, si bien la variación de la jugabilidad entre ambas representaciones es poca (o hasta incluso algunos juegos y géneros de juegos resultan más cómodos en 2D) la tecnología en videojuegos ha empujado desde siempre a la búsqueda de experiencias audio visuales cada vez más realistas [136] [98]. En la actualidad el uso de 2D en juegos no es muy frecuentando en la industria AAA de videojuegos, no obstante, sigue siendo la elección para muchos videojuegos en las siguientes condiciones:

- Juegos para dispositivos móviles
- Juegos online (Flash).
- Juegos casuales [48].
- Juegos independientes.

- Juegos Serios [40].

Existen distintos tipos de variantes de esta representación:

- Una sola pantalla (*single screen*): El juego transcurre en una sola pantalla sin movimientos de cámara. Por ejemplo el juego **Pacman** (ver figura 6.2.4.1).

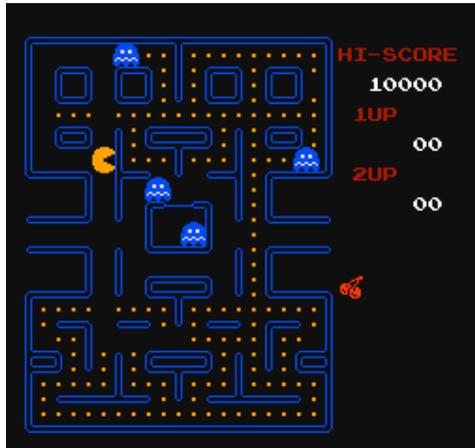


Figura 6.2.4.1 **Pacman** (Namco).

- Juegos con *scrolling*: El mapa o área de juego es más grande de lo que puede mostrar en la pantalla por lo que es necesaria la simulación de una cámara para seguir la acción del jugador. Por ejemplo el juego **Gradius**.



Figura 6.2.4.2 **Gradius**.

- Juegos Isométricos: Los juegos con vista isométrica son aquellos que intentan simular un entorno 3D al utilizar una vista con perspectiva. Por ejemplo el juego **Diablo** (Blizzard).



Figura 6.2.4.3 **Diablo** (Blizzard).

Las estructuras principales utilizadas en los juegos 2D para tratar la información gráfica, independientemente de la variación, son los *bitmaps*. Los *bitmaps* consisten en una matriz de valores numéricos donde cada uno representa un color. Este valor numérico lleva el nombre de pixel y según la cantidad de bits destinados a este valor varía la profundidad del color (cantidad diferente de colores que puede representar). Por ejemplo utilizando valores numéricos de 24 bits, la convención general para representar un color como un número (en hexadecimal), es representarlo como RRGGBB (ver imagen 6.2.4.4), es decir los primeros 8 bits determinan el valor del color rojo, el grupo siguiente de 8 bits representa valores de verde, y el ultimo representa los valores de azul. Con esta representación se pueden mostrar 16,777,216 diferentes colores. En la actualidad se suelen utilizar las representaciones de 32 bits siendo para esta la convención AARRGGBB, donde los primeros 8 bits sirven para representar la transparencia, y el resto es equivalente a la representación en 24 bits.

24 bits 00000000 00000000 00000000
Rojo Verde Azul

Número	Color
FF 00 00	
00 FF 00	
00 00 FF	
FF FF 00	
00 FF FF	
FF 00 FF	

Figura 6.2.4.4 Codificación de colores en 24 bits.

De esta manera por ejemplo, una imagen de 8x8 pixeles, queda representada en una matriz de pixeles (de 8x8) de la manera que se ilustra en la figura 6.2.4.5.



Figura 6.2.4.5 Representación de una imagen en pixeles de 24 bits.

Para mostrar los gráficos en pantalla se realiza la operación conocida como “*blitting*” (Blit resulta de la contracción de las iniciales en ingles para transferencia de imágenes en bloque). La operación de *blitting* consiste en copiar la matriz de la imagen deseada a la matriz que corresponde a la memoria de la placa de video, que luego a su vez, será volcada al monitor [136].

Para poder superponer dos imágenes, por ejemplo un personaje sobre un fondo, es necesario definir un color de transparencia (*color key*). De esta manera, si se definiera en la imagen del ejemplo anterior al color FFFFFFFF (blanco) como transparente, la bola de la imagen puede superponerse con el fondo perfectamente sin mostrar el recuadro blanco que la acompaña (Ver figura 6.2.4.6).

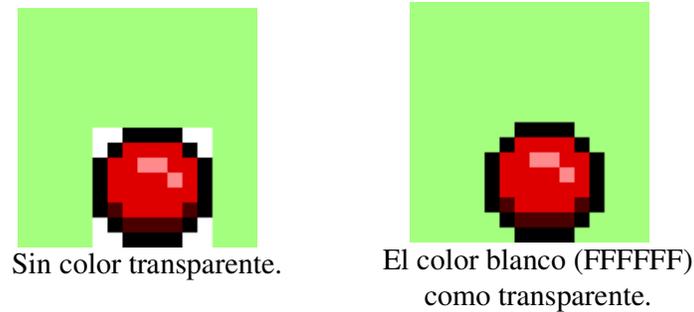


Figura 6.2.4.6 Color key.

Al elegirse un color de transparencia y al realizar el *blit*, se copian todos los colores de la imagen excepto el color transparente. Si se utiliza una codificación de 32 bits con canal alfa no es necesario definir un color key, ya que se cuenta con los bits para indicar la transparencia, de modo que un color completamente transparente queda representado por ceros en los primeros 8 bits del pixel.

Para realizar animaciones se crea la estructura conocida como *sprite*. Un *sprite* es una estructura compuesta por una imagen que se encuentra dividida, de manera lógica, en distintas secciones que se denominan cuadros (*frames*). Cada uno de estos *frames* corresponde a un elemento distinto de la animación (ver figura 6.2.4.7).



Figura 6.2.4.7 Ejemplo de *sprite*.

Esta técnica de animación es la misma utilizada por el cine de animación tradicional donde se realizan los dibujos en diferentes cuadros que se intercalan para crear la sensación de movimiento.

Otra técnica muy utilizada en juegos 2D es dividir un elemento (análogo al *sprite*) en partes lógicas (llamados *tiles*) solo que en lugar de estar destinado a mostrar una animación, se lo utiliza para crear imágenes mucho más grandes utilizando menos recursos. Un *tilemap* es una matriz de números que indexa a cada uno de *tiles* de una imagen (ver figura 6.2.4.8).

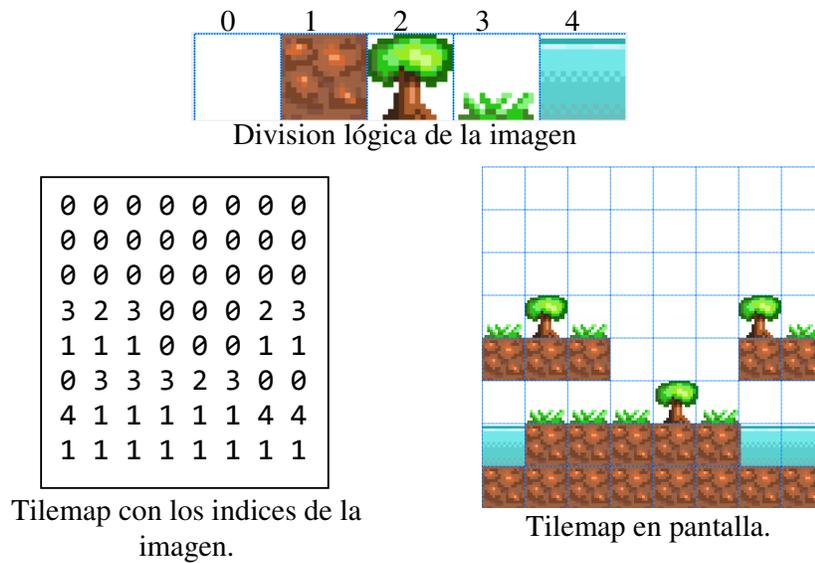


Figura 6.2.4.8 Tilemap.

Para lograr la ilusión de movimiento de cámara (el lugar que se muestra en pantalla) se utiliza un par de coordenadas x e y, junto con una definición del alto y ancho del área a mostrar. Luego se dibujan en pantalla aquellos *sprites*, y *tiles* (elementos del *tilemap*) cuyas posiciones y tamaños estén contenidas dentro del área a mostrar en las posiciones correspondientes desplazadas por las coordenadas x e y de la cámara. De esta manera se puede generar el efecto conocido como *scrolling* (desplazamiento de la pantalla) variando las coordenadas x e y del área a dibujar.

Es común la utilización de capas para manejar la representación gráfica del juego. De esta manera se puede designar una capa diferente para dibujar los fondos, otra para dibujar los *sprites*, estas divisiones, en consolas de generaciones antiguas estaban ya determinadas por hardware. Cada una de las capas no es más que una colección de elementos gráficos que se dibujan en un orden predeterminado.

Para lograr el efecto conocido como "*Parallax Scrolling*", el cual consiste en un desplazamiento de las diferentes capas a distintas velocidades para lograr la ilusión de distancia (las capas más superiores se mueven con mayor rapidez) se utiliza un desplazamiento distinto para cada una de las capas. Cuanto más alejada este la capa (La primera capa en dibujarse es la más lejana) menor será su velocidad de desplazamiento.

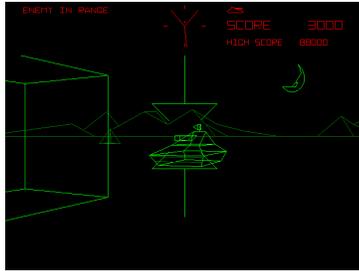
Existen técnicas para aprovechar la eficiencia desarrollada en las placas de video para el procesamiento de gráficos en 3D, la cual consiste, en general, en la utilización de planos o polígonos (geometrías sencillas) a los cuales se les aplica una textura con el gráfico que se pretende dibujar [140].

Un ejemplo de librería utilizada para el tratamiento de gráficos 2D es SDL. SDL (Simple Direct Media), es una librería multimedia diseñada para proveer acceso de bajo nivel, pero con abstracción de hardware, a los componentes de audio, teclado, mouse, joystick. Se distribuye bajo la licencia GNU LGPL versión 2, y es multiplataforma soportando varios sistemas operativos como Linux, Windows, Mac, Solaris. [141]

6.2.5. Representación 3D.

La representación en 3D, la más popular hoy en día, ha evolucionado desde los primeros videojuegos en 3D simplemente con vectores de las primeras máquinas de arcade, hasta la actualidad, donde las consolas son capaces de renderizar millones de polígonos multitexturados con luces y sombras en tiempo real y con efectos de post procesado en lenguajes programables.

El hardware destinado al procesamiento gráfico experimentó una evolución acelerada, incluso, desafiando la Ley de Moore, que llevo no solamente a un aumento en la capacidad de procesamiento, sino también, a un cambio en la estructura del proceso de renderización que permite lograr en tiempo generar imágenes fotorealistas [146] (ver figura 6.2.5.1 para un ejemplo ilustrativo).



Battle Zone. Atari. 1980.



Doom ID. 1993



Quake II. ID. 1997.



Halo 2. Microsoft. 2005



Assasins Creed. Ubisoft. 2008



Crysis 2. EA. 2011

Figura 6.2.5.1 Evolución de los gráficos 3D a través de distintos juegos y años.

Existen una serie de estructuras características de la representación en 3D que es de uso común en los distintos *engines* y APIs y es el objetivo de esta sección describirlos y explicar el proceso por el cual estas estructuras se transforman en gráficos en la pantalla.

La unidad más básica utilizada en 3D es el vector. El vector es un par ordenado de tres números que permiten determinar una posición en el espacio tridimensional. En general se utilizan el tipo de dato *float* de 24 bits, ya que provee una precisión suficiente, y además, es utilizada en la mayoría de las placas de video actuales.

El uso más frecuente de los vectores (además de determinar posiciones y velocidades) es agruparlos secuencialmente en la estructura conocida como triángulo.

Cuando es así el vector lleva el nombre de vértice. Los triángulos son la estructura en la cual se puede representar un modelo 3D completamente. Sin embargo para representar figuras complejas con un elevado nivel de detalle (Ver figura 6.2.5.2) estos modelos pueden llegar a ocupar mucho espacio en disco y en memoria.

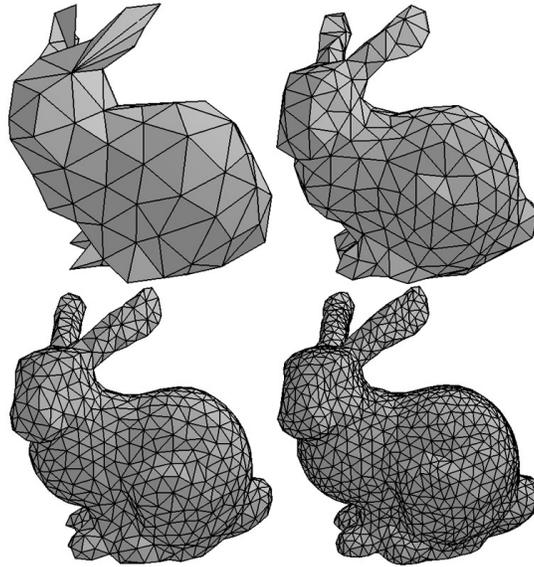


Figura 6.2.5.2 Distintas cantidades de polígonos para un modelo de un conejo.

Es por ello que se desarrollaron diversas técnicas para reducir este espacio. La indexación de los vértices de los triángulos permite eliminar la redundancia en aquellos casos en los que dos o más triángulos comparten vértices. Para ello se crea una lista de vértices a los que se le asigna un índice, y luego, los vértices de cada triángulo del modelo son referenciados a esta lista [136].

Otra técnica de compresión llamada cuantización, consiste en utilizar un tipo de dato de menor precisión, con la consecuente pérdida de información, para almacenar a los valores a comprimir [164]. El tiempo de procesamiento que requiere esta técnica tanto para comprimir como para descomprimir es despreciable, lo que la hace muy versátil. La pérdida de información, si bien existe, puede ser manejada de manera tal que no afecte significativamente a la calidad del modelo, para ello se calculan el mínimo y el máximo de los valores a comprimir, y se los hace coincidir con la precisión del tipo de dato de menor precisión [136].

Para lograr un mayor realismo y poder representar diversos tipos de materiales, a cada uno de los triángulos se le asigna una textura. Una textura consiste en una imagen que se aplica al modelo conforme la geometría del objeto. Las texturas se

almacenan en memoria y se les asignan un índice para luego ser referenciadas. Para cada triángulo no solo se debe especificar el índice de la textura sino también, un punto referencial de la textura por cada vértice para indicar que parte de la textura corresponde a cada uno de los puntos (ver figura 6.2.5.3). Este punto referencial consta de dos coordenadas, normalmente denominadas U y V. Existen diversas formas de aplicar e interpretar las coordenadas de una textura, e incluso, un mismo vértice puede contener índices a más de una textura, en lo que se conoce como multitexturas [136].

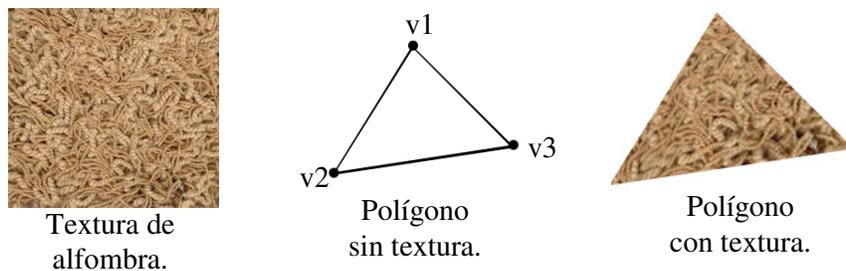


Figura 6.2.5.3 Polígono texturado.

Para lograr transformar los datos contenidos en las estructuras explicadas a una imagen en pantalla se deben ejecutar una serie de procesos conocido como el “*graphic pipeline*” o proceso de renderización [136] Las responsabilidades de este proceso son tres [143]:

- Aplicar una transformación a la información en 3D de los componentes gráficos del mundo utilizando una transformación para representar la información de acuerdo al punto de vista del observador.
- Eliminar la mayor cantidad de datos que no serán visibles desde el punto de vista del observador con el objetivo de mejorar la eficiencia del proceso de renderización.
- Mostrar en pantalla, gracias a transformaciones del espacio 3D hacia un espacio 2D, los gráficos procesados. Está es una operación muy costosa, y para lo cual los GPU han sido optimizados.

Para lograr cumplir con estas responsabilidades, el proceso de renderización deberá atravesar las siguientes etapas (descriptas en las secciones siguientes):

- Determinación de la visibilidad.
- Determinación de la resolución.
- Transformaciones y luces.
- Rasterización.

6.2.5.1 Determinación de la visibilidad.

Para lograr una optimización en la utilización del hardware y en el tiempo de procesamiento, se debe intentar minimizar los elementos a procesar. El objetivo de esta etapa es filtrar aquellos objetos que efectivamente aparecerán directa o indirectamente en pantalla, ya que, de otro modo se gastarían recursos en el procesamiento de objetos invisibles [136][143].

La técnica conocida como *clipping* consiste en eliminar aquella geometría que no se verá en pantalla utilizando una medida de volumen. La obtención de este volumen, y el testeado deben ser lo suficientemente eficientes para que sea provechosa su implementación y no signifique, en cambio, una pérdida de performance [136]. Es importante también, como en cualquier técnica de determinación de visibilidad, que no se produzcan falsos negativos. Es decir, se puede permitir el error de incluir geometrías que no serán visibles, sin embargo, se considera una falla grave que una geometría que debería presentarse en pantalla no se muestre por culpa de un falso negativo.

Existen diversas formas de calcular este volumen:

- El ángulo de visión del observador, comúnmente es de 60° por lo que, solamente se deben mostrar aquellas geometrías que queden dentro de este ángulo.

- **Recorte de triángulos:** el chequeo de cada uno de los triángulos que componen cada uno de los modelos de la imagen es una operación costosa de realizarse por software, por lo que en general, se relega este método a las placas de video que tienen hardware especializado para dicha tarea. Si bien se podría dejarse que todo el recorte de geometrías no visibles se haga por el hardware de la placa de video, se debe considerar también, que el envío de toda la geometría completa se traduciría en un uso ineficiente del bus de datos.
- **Recorte por objetos:** En lugar de procesar cada uno de los triángulos de cada uno de los modelos, se pueden utilizar representaciones mucho más fáciles y rápidas de procesar, asumiendo cierto error que no posibilite falsos negativos, para acelerar el proceso de *clipping*. La utilización de esferas que cubran la totalidad del objeto proporciona la ventaja de ser invariantes a las rotaciones y que el cálculo para saber si una esfera aparece o no en el área que se mostrará finalmente, se reduce solo al cálculo de una ecuación de productos y sumas. Los objetos con formas alargadas, sin embargo, harían que la esfera resultante sea muy grande en relación del tamaño del objeto y ocasione demasiados falsos negativos (Ver figura 6.2.5.1.1).

La utilización de cajas, por otro lado, puede ofrecer la ventaja de ajustarse mejor a distintas formas de objetos, y por lo tanto garantiza una mejor calidad en la discriminación de objetos visibles e invisibles. El chequeo, no obstante, es más complejo, y a diferencia de las esferas, las cajas no son invariantes respecto a la rotación.

La técnica conocida como "*culling*", permite eliminar porciones de un objeto (o el objeto mismo) tomando en cuenta su orientación con respecto al observador [140]. Los modelos en 3D tienen la particularidad de ser del grupo llamado "objetos bien formados" cuyos triángulos son capaces de delimitar un volumen interno y separarlo del resto del mundo. Una característica importante de este grupo, es que cada uno de los vértices se encuentra en el orden contrario de las agujas del reloj. Realizando simplemente el producto vectorial entre dos vértices se puede obtener la normal a cada uno de los triángulos; si esta normal apunta fuera del punto de vista de la sección a renderizar, entonces, esta se encuentra ocluida por otras caras. Este tipo de testeo, a nivel de polígono, es usualmente realizado por el GPU ya que es una operación costosa de realizar por software [136].

El testeo de oclusión basa en determinar la visibilidad conforme a si la geometría está o no tapada (respecto al punto de origen de la vista) por otro objeto. Si bien las técnicas de clipping y de culling pueden haber eliminado un gran número de triángulos, esta técnica permite evitar que se rendericen objetos que no estarán a la vista, reduciendo el tiempo total de renderización y evitando el sobre dibujado. Las técnicas y políticas aplicadas para determinar una mejor implementación del algoritmo de testeo de oclusión dependen del tipo de juego [136]. No se aplican las mismas optimizaciones para juegos con grandes espacios abiertos, que para juegos FPS con escenarios interiores y pequeños, sin embargo, escapa al alcance del trabajo discutirlas en profundidad.

Al igual que el *clipping* de objetos, una forma eficiente de implementar el testeo de oclusión es asignar un borde de volumen (BV) (ya sea una caja, una esfera o un AABB) del menor tamaño posible que cubra completamente el objeto, luego se envían, en formato de consulta de oclusión, al proceso de renderizado del GPU. Durante esta consulta los BV no se renderizan sino que se chequean sobre el buffer Z (de profundidades). Aquellas geometrías cuyos BV que no modifiquen el buffer no deben ser renderizado ya que se encuentra ocultos por otros objetos. Para hacer más efectivo el test, se realizan los chequeos desde el más cercano, para lograr que los objetos más grandes (en relación al punto del observador) se testeen primero [136][143].

6.2.5.2 Determinación de la resolución.

Otra estrategia para conseguir eliminar aún más triángulos y geometrías consiste en considerar el efecto conseguido por la distancia de los objetos hacia el punto de vista, su tamaño e importancia.

Sabiendo que el tamaño percibido de un objeto es inversamente proporcional a su distancia al cuadrado, renderizar objetos complejos que no se verán en detalle debido a su tamaño relativo en pantalla, significa un uso ineficiente de los recursos.

La multiresolución consiste en utilizar distintos modelos (ya sean creados por un artista o un software de modelado, o bien generados dinámicamente) para un mismo objeto dependiendo de su distancia hacia el observador. Básicamente, las técnicas utilizadas en la multiresolución se dividen de dos procesos diferentes: la determinación de la

resolución (puede ser mediante un algoritmo o mediante heurísticas) y el dibujado de estos modelos dependiendo de la resolución deseada.

Las técnicas de multiresolución pueden basarse en aproximaciones discretas y continuas para determinar el nivel de detalle (LOD por sus siglas en ingles) del modelo:

- **Políticas discretas para la determinación del LOD:** Las políticas discretas dividen el espacio (formado por la distancia hacia el observador o el área de renderizado) en segmentos en los cuales para cada uno le corresponde un modelo con distinto LOD (usualmente de 2 a 5 diferentes niveles) (Ver figura 6.2.5.1.1). De esta manera el modelo con mayor detalle será el utilizado en el primer segmento (cuanto más cerca al jugador, mayor LOD). La ventaja de utilizar las políticas de LOD discretas residen en la ausencia de costo para el procesamiento, sin embargo estas políticas, producen el efecto denominado *popping* (El punto en el cual un modelo es reemplazado por otro de diferente LOD puede ser visiblemente molesto).

Para solucionar esto se suelen dibujar ambos modelos, durante la transición, pero con distintos niveles de alpha. De esta manera el cambio de los modelos es casi invisible, pero aumentan el uso de los recursos al renderizar los dos modelos con diferente LOD.

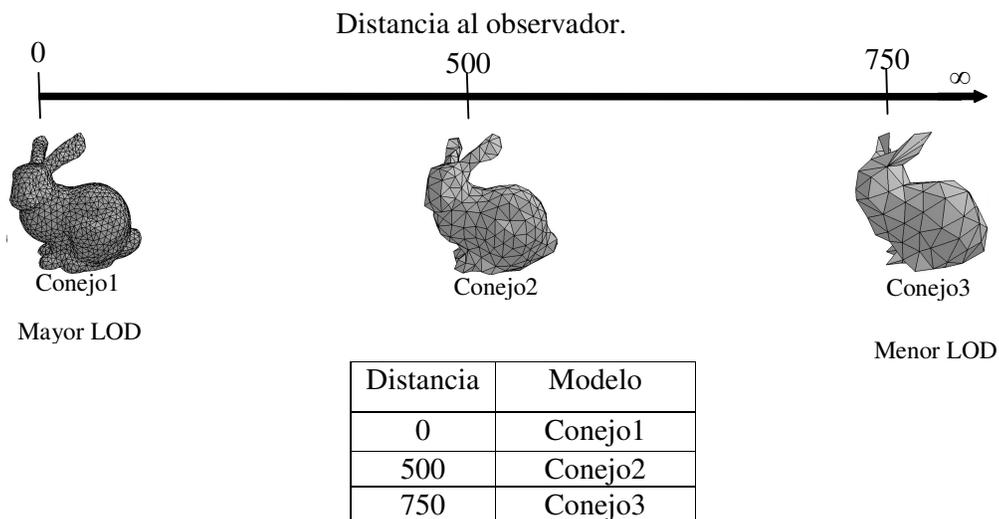


Figura 6.2.5.1.1 Distancia al observador con LOD discreto.

- **Políticas continuas para la determinación del LOD:** Las políticas continuas para la determinación del LOD utilizan algoritmos para determinar, en tiempo real, el modelo a renderizar con un menor número de triángulos. Al ser realizado en tiempo real tiene un impacto en el uso de CPU, por lo que se suelen utilizar técnicas como la coherencia temporal. Estas técnicas consisten en almacenar los datos ya calculados (junto a la distancia), y solamente volver a ser recalculados pasado un umbral establecido.

Para implementar políticas continuas para la determinación de LOD, se necesita almacenar información extra sobre cada modelo, como pueden ser información sobre bordes, o vértices vecinos, a fin de acelerar los cálculos que determinarían la reducción de la cantidad de polígonos.

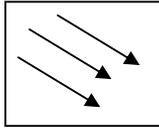
Uno de los métodos más populares, llamado plegado de Bordes (*Edge collapsing*), se basa en unir aquellos triángulos adyacentes que aportan poca información al modelo. Para ello se buscan los bordes donde los planos de apoyo de sus triángulos vecinos son casi co-planares o bien que por lo menos el ángulo entre los planos sea menor que un umbral establecido. Una vez que se realiza la selección del borde se pueden abordar dos estrategias distintas: se eliminan y pliegan los dos vértices en uno, o se eliminan ambos vértices y se crea uno nuevo ubicado a lo largo del borde.

6.2.5.3 Transformaciones y luces.

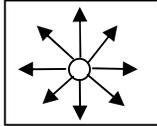
Antes de ser mostrados en pantalla se aplican a los modelos las rotaciones, los cambios de escalas, y otras transformaciones. Las transformaciones son representadas como una matriz 4x4 la cual multiplica cada uno de los vértices para obtener la transformación de la geometría individualmente por cada vértice. Estas operaciones se llevan a cabo por lo general en el GPU teniendo un costo nulo para el CPU. En esta etapa se transforman también cada uno de los vértices a coordenadas de la pantalla usando la matriz de la proyección [136].

La aplicación de luces también es realizada generalmente por la GPU, y en combinación con efectos logrados gracias a los *pixel shaders* (explicados más adelante) que logran una apariencia aún más realista de los modelos, aplicando diferentes niveles de brillo a las texturas de los modelos dependiendo de fuentes de luz y de los materiales de las texturas [136].

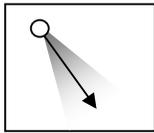
Existen tres tipos estándares de luces:



Luces direccionales (*directional light*): son luces que simulan estar lo suficientemente lejos como para que todos los rayos de luz sean paralelos. Un ejemplo clásico es el sol.



Punto de luz (*point light*): La fuente de luz tiene lugar dentro del espacio y emite luz en todas direcciones.



Foco de luz (*focal light*): La fuente de luz tiene lugar dentro del espacio y emite luz en una dirección en particular.

El tipo de luz que produce una aproximación más razonable a las fuentes de luz reales (las cuales emiten luz desde un área o volumen) son los puntos de luz. No obstante la aproximación no produce siempre una representación fiel ya que las sombras de un punto de luz tienen bordes más definidos [143]. Las luces también pueden tener asignado un color RGB, o bien ser monocromáticas, y poseer una intensidad (solo los puntos de luz y los focos de luz). En el caso de los puntos de luz, la intensidad de la luz suele ser inversamente proporcional al cuadrado de su distancia. Las fuentes de luz con color deben especificar los valores que aportan a [143][144]:

- La luz ambiental: aportación a la luz ambiental de la escena
- La luz difusa: color de la fuente de luz.
- La luz especular: color del brillo especular que produce la fuente de luz.

6.2.5.4 *Rasterización.*

El proceso de *rasterización* es la parte final del proceso donde los píxeles están listos para ser mostrados en pantalla. Esta tarea es primordialmente realizada en el GPU y el deber de los desarrolladores es entregarle dicha información a la placa de video, haciendo un uso eficiente del bus de comunicación. Éste representa el cuello de botella de este proceso.

Las APIs de video tales como DirectX y OpenGL proveen, bajo distintos nombres, herramientas para hacer más eficientes el uso del bus de datos. Enviar la información mediante pequeños llamados produce la fragmentación en el envío de datos por lo que el uso de *buffers* para almacenar geometrías es provisto por ambas APIs. No obstante para hacer aún más eficiente el uso del bus de datos, las placas de video incorporan un espacio considerable de memoria RAM para almacenar geometrías, texturas y materiales que son utilizados más frecuentemente. El acceso de esta memoria por parte de la interfaz de programación provee un control total de lo que reside en la memoria de la GPU, sin embargo, es importante considerar que si bien la escritura de datos es muy veloz, la lectura de estos es mucho más lenta, haciéndola inadecuada para almacenar datos y estructuras dinámicas (con frecuente cambio).

6.2.5.5 Shaders.

La evolución del hardware destinado al procesamiento gráfico transformó su enfoque desde el de una funcionabilidad fija hacia una naturaleza programable [146]. En un principio los GPU estaban diseñados para ejecutar un *pipeline* predefinido que limitaba lo que los desarrolladores podían utilizar del GPU. El avance tecnológico se dirigió hasta entonces, hacia la mejora de la performance y de la integración de nuevas características. Sin embargo llegó un momento en que la cantidad de polígonos a renderizar no producía un aumento significativo en el realismo de los gráficos [146], y con la llegada del concepto del *shader* al hardware, se produjo una evolución aún más grande en cuanto los efectos y el realismo que son capaces de mostrar los videojuegos de hoy en día [145].

RenderMan, uno de los primeros lenguajes de *shaders* que fue desarrollado por Pixar Animation Studios, permitió lograr efectos fotorealistas en las renderizaciones ofreciendo a programadores y artistas la posibilidad de controlar por completo los resultados del proceso de renderización mediante un lenguaje de programación simple pero poderoso. Sin embargo, dada su complejidad y el hardware contemporáneo, no estaba diseñado para utilizarse en tiempo real [145].

A partir de la versión 8, DirectX incluyó en su API a la versión 1.0 de los *shaders* para acceder a esta funcionabilidad dentro de las GPU. El lenguaje de los *shaders* 1.0 no poseía elementos de control de flujo, algo que cambió en la segunda versión. En la actualidad la mayoría de las APIs gráficas y el hardware soportan la versión 2.0 y 2.x de *shaders* [145].

Mediante el uso de *shaders* es posible [146]:

- Incrementar el realismo: el de los materiales (tal como piedra, metal, madera etc) el realismo de los efectos de luces (reflejos, sombras suaves), y de los fenómenos naturales tales como humo, agua y nubes.
- Desarrollar efectos avanzados de renderizado en la iluminación global.
- Crear efectos avanzados (no fotorealistas) como pueden ser efectos de dibujado en tinta y en lápiz, remarcado caricaturesco, etc.
- Nuevos usos de la memoria de almacenamiento de texturas normales, valores de brillo, coeficientes polinomiales, etc,
- Uso de filtros de convolución para imágenes, mascarar de suavizados, *blending* avanzado, etc.
- El uso de texturas 2D y 3D generadas de manera procedural.
- Efectos de interpolación de cuadro en animación, sistemas de partículas, movimiento definido proceduralmente, etc.
- Ejecutar algoritmos de computación general como orden, modelado matemático, dinámica de fluidos, etc.

Existen dos tipos generales de *shaders* que corresponden a momentos distintos en el *graphic pipeline* y que, por lo tanto, pueden tener acceso a diferente información y lograr efectos diferentes.

Tal como se explico anteriormente, los vértices son las estructuras por medio de las cuales se especifican los polígonos que componen los modelos de la representación 3D. Estos vértices son enviados por medio de las APIs gráficas hacia el hardware donde por cada vértice son ejecutados los conocidos como *vertex shaders* (*shaders* de vértices) [145]. El procesador de vértices ejecuta los *shaders* y se los entrega al subsistema de rasterización (ver figura 6.2.5.5.1).

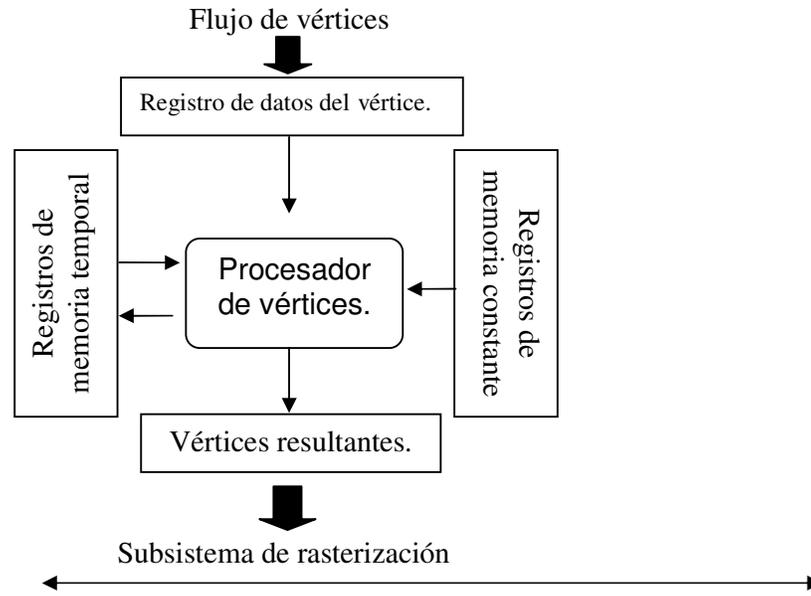


Figura 6.2.5.5.1 Diagrama funcional del procesador de vértices según [145].

El procesador de vértices soporta usualmente algoritmos para transformación de vértices, normalización, generación de las coordenadas de las texturas, transformación de las coordenadas de las texturas, iluminación, y aplicación del color del material [146].

En cuanto a la programación cuenta con [145]:

- Registro de datos del vértices. El cual puede contener 16 datos diferentes para cada vértice que se lee del flujo de vértice enviado por la API gráfica.
- Una memoria de registros constante (de solo lectura) para datos estáticos de los *shaders*. Puede almacenar números en punto flotante, enteros o valores booleanos. Cada uno de los registros está compuesto por vectores de 4 dimensiones posibilitando procesarlos todos al mismo tiempo o bien utilizar máscaras para procesarlos uno a uno.
- Una memoria de almacenamiento temporal para almacenar los datos de los cálculos (en la cual se puede leer y escribir).
- Los valores finales de los vértices se almacenan en el vector de vértices resultantes para ser entregados al proceso de rasterización.

En el proceso de rasterización se realiza la interpolación (donde la información definida en los vértices de los polígonos se traslada a los valores correctos de cada pixel) y el proceso de oclusión (anteriormente descrito). Una vez concluido este procesamiento se invoca al *pixel shader* por cada uno de los pixeles en pantalla a dibujar (Ver figura 6.2.5.5.2).

El procesador de pixel shader usualmente realiza las operaciones tradicionales como: operaciones sobre los valores de interpolación, acceso a texturas, aplicación de texturas, efecto de niebla, y operaciones sobre colores [146].

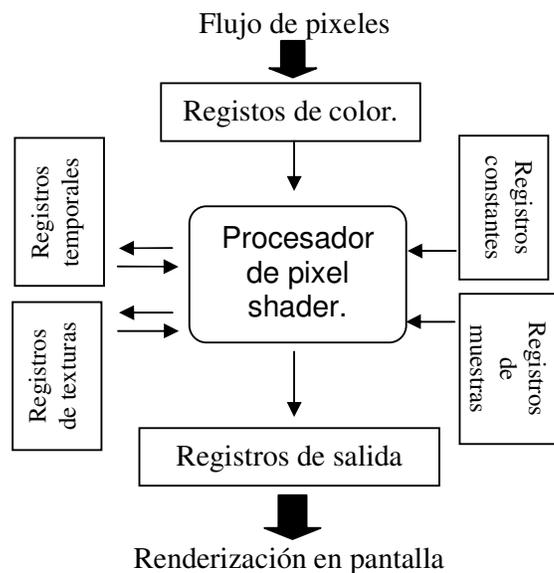


Figura 6.2.5.5.2 Diagrama funcional del procesador de pixel shader según [145].

En cuanto a la programación cuenta con [145]:

- Registros constantes (solo lectura): son equivalentes a los registros constantes del procesador de vértices almacenan información útil y estática para los *shaders*.
- Los registros de color: donde se proveen los datos de la textura y el color.
- Los registros de texturas: llevan una tabla de acceso rápido de las coordenadas de los valores interpolados de las texturas.

- Los registros temporales sirven para almacenar información intermedia de los procesos de los *shaders*.
- Los registros de muestras (solo lectura): llevan la información de puntos a texturas que el pixel shader va a muestrear de la textura durante el procesamiento actual del píxel.
- Los registros de salida son usados por el hardware para renderizar el pixel final. Define en color, la niebla y la profundidad (los valores de salida del pixel shader).

6.2.6 APIs gráficas.

Las APIs para el manejo de gráficos 3D surgieron de la necesidad de poseer una interfaz estandarizada que permitiera acceder a las características del hardware de procesamiento gráfico proveniente de fabricantes heterogéneos. Los fabricantes en un principio habían optado por desarrollar para cada uno una API diferente. Esto hacía más difícil el desarrollo de aplicaciones que hicieran uso de estas características sin tener que re-implementarse bajo las diferentes APIs (para lograr asegurar la compatibilidad), y aún más complicado, lograr que en todas ellas las aplicaciones se comporten de igual manera.

Como se ha nombrado anteriormente existen dos grandes APIs gráficas con enfoque, alcance y estrategias diferentes, pero con un mismo objetivo. DirectX y OpenGL son estas dos grandes APIs.

6.2.6.1 DirectX

DirectX inicialmente surgió como una propuesta de Microsoft para incentivar a los desarrolladores de videojuegos a migrar desde el entorno DOS hacia el de Windows 95. La dificultad con la que se encontraban los desarrolladores de videojuego es que la API gráfica de Windows nativa estaba destinada al manejo de GUIs (con muchas capas de abstracción) y por lo tanto resultaba extremadamente lento para manipular gráficos a bajo nivel [143][149]. DirectX se compone de varios módulos que resuelven no solamente el procesamiento de los gráficos, sino también el manejo de entradas (mouse, teclado, joystick) y el sonido.

No fue hasta la segunda versión (en 1996) que DirectX incluyó módulos para el manejo de gráficos 3D conocido como DirectX3D, y a partir de la tercera versión DirectX comenzó a ser popularmente aceptado por los desarrolladores [143]. Con DirectX 6 y DirectX 7 se introdujo un cambio en la política de su interfaz pasando de *retained mode* (la API gestiona el *pipeline* de los elementos los cuales son agregados mediante llamadas) hacia *immediated mode* (se hacen llamadas a la API para realizar los diferentes procesamiento, gestionando los elementos el programador). A partir de la versión 8 se introdujo finalmente la posibilidad de utilizar pixel y vertex *shaders*.

La versión actual de Directx es DirectX 11, en esta nueva versión las características principalmente destacadas son [148][150];

- **Teselación:** Si bien la capacidad de eliminar y crear vértices en el GPU la brindó anteriormente DirectX 10 con el concepto de geometry Shader, en la versión de DirectX 11 se incorpora finalmente el proceso conocido como teselación (no programable) junto con dos nuevos procesos programables *hull shader* (anterior al proceso de teselación) y *domain shader* (que toma la salida de la teselación) para hacer uso de el. La implementación de ese proceso en el *pipeline* gráfico permite manejar el LOD en el GPU y hacer un mejor uso del bus de datos enviando menos información al GPU y aprovechando esta característica.
- **DirectX Compute:** Incorpora un *shader* que permite utilizar el GPU para programar bajo el lenguaje de *shader* HLSL para el procesamiento de propósito general.
- **Nuevo y mejorado formato de compresión:** El algoritmo de compresión de texturas utilizado por DirectX 11 BC7 produce una pérdida menor de información y ocupa menor memoria que el algoritmo anteriormente utilizado, lo que permite un uso más eficiente de la memoria del GPU.
- **Procesamiento paralelo:** Con DirectX 11 se introducen funciones que posibilitan aprovechar el procesamiento paralelo para la carga o creación asincrónica de recursos y la creación de listas de de renderización paralela.

- **Shader Model 5:** En la nueva versión del lenguaje de *shader* HLSL Microsoft incorpora algunos conceptos de programación orientada a objetos que facilita la organización y la creación del código.

6.2.6.2 OpenGL.

Open GL surge como un estándar de la industria multiplataforma (principal falencia de DirectX) para el tratamiento de gráficos 2D y 3D utilizando características de hardware dedicado. Silicon Graphics junto con compañías de hardware fueron los responsables de la creación de esta API gráfica abierta [146].

OpenGL tuvo el apoyo de muchos desarrolladores a través de las primeras versiones, ya que estaba mucho mejor estructurado que su contraparte. No obstante la lentitud con que OpenGL logra adaptarse a las nuevas tecnologías (debido a que pone un fuerte hincapié en la compatibilidad con versiones anteriores) provocó que la mayoría de los desarrolladores optaran por DirectX, especialmente a partir de la versión 9. Desde entonces, las versiones de DirectX marcan la tendencia en la funcionabilidad que, con el tiempo, logra agregar el estándar abierto [148].

La versión más actual de OpenGL es la versión 4.2, que incorpora al igual que DirectX el algoritmo de teselación y la posibilidad de utilizar el GPU para programación de propósito general en su *pipeline*. Al igual que DirectX a partir de la versión 3, también incluye los *geomety shaders* que permiten la creación y eliminación de vértices a través del uso de un lenguaje de *shaders* [147].

OpenCL, por otra parte, es una API estándar para el desarrollo de aplicaciones de propósito general que pueden hacer uso tanto de CPU como de GPU (compatibles con chips de Amd, Intel y Nvidia). Utiliza en un lenguaje propio y ofrece facilidades para la programación en paralelo [192] y si bien no forma parte de OpenGL, pueden utilizarse en forma conjunto, y representa a su vez una alternativa abierta a DirectX compute.

6.2.7 Manejo de audio.

El sonido y la música son una parte muy importante de los videojuegos ya que complementan la experiencia del juego, realzan el realismo, proporcionan un feedback más claro de lo que sucede [157], aumentan la inmersión en el mundo del juego, e

incluso, inducen o promueven distintos estados anímicos durante la diversas situaciones que los videojuegos plantean [163].

Existen dos enfoques principales en lo que respecta a la utilización de audio: el audio generado mediante sintetizadores de modulación de frecuencia, y el audio obtenido a partir de *samples*.

El audio generado mediante sintetizadores, utilizado especialmente en los primeros videojuegos, tiene la ventaja de ser generado “sobre la marcha” mediante aproximaciones con funciones matemáticas, y por lo tanto tienen un requerimiento muchísimo menor de espacio a costa de generar sonidos de mucho menor realismo.

El audio sampleado, por el contrario, es audio que fue capturado por medio de un micrófono y procesado para poder ser almacenado. Este proceso consiste en una etapa de captura del sonido (mediante un micrófono) en forma de señales analógicas, que luego es convertida mediante un conversor de señales analógicas a digitales (ADC), para poder luego, ser interpretada y almacenada por un programa [164] (Ver figura 6.2.7.1).

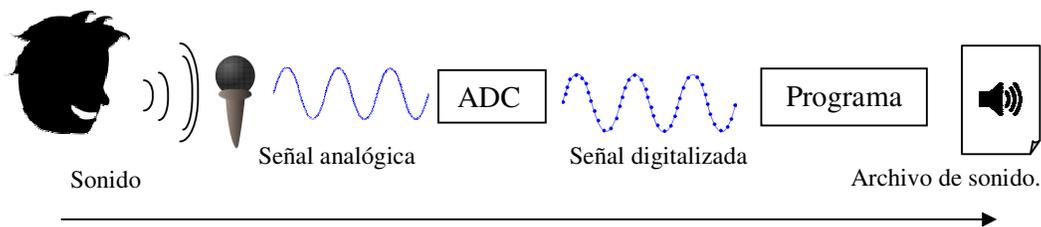


Figura 6.2.7.1 Adquisición de sonido.

Para poder capturar un sonido se deben tomar varias muestras por segundo (lo que se conoce como frecuencia de muestreo) ya que es imposible capturar por completo un sonido sin realizar una discretización del tiempo. El teorema del muestreo establece que para capturar un sonido correspondiente a una determinada frecuencia, se debe capturar al menos, el doble de esta frecuencia [164]. Esta frecuencia de muestreo se mide en Herz, y las frecuencias generalmente utilizadas son 11025Hz, 22,050Hz, y 44,100Hz (44.1KHz). Dado que la música requiere un ancho de banda de 20Kz, y por el anteriormente citado teorema de muestreo, la frecuencia 44.1Kz es una de las más populares [164].

La calidad de la muestra, por otro lado, se refiere a la precisión con que cada muestra en particular es procesada y almacenada. Es decir, del mismo modo en que se discretiza el tiempo, se debe hacer lo mismo con la amplitud de los sonidos. Si se utilizan 8 bits para cada una de las muestras, entonces, se podrán representar 256 valores distintos de amplitud, en cambio si por ejemplo si se utilizaran 16 bits, esta capacidad de representación asciende a 65.536 valores distintos [162].

Almacenar estos valores en archivos sin compresión requiere de un gran espacio de almacenamiento. Por esta razón es frecuente que los archivos de audio utilicen algoritmos de compresión de datos para reducir este espacio. Los formatos MP3 y ogg son los más populares para la compresión de sonidos. Sin embargo, una vez comprimido y para poder ser reproducidos se deben aplicar algoritmos de descompresión, lo cual puede suponer un esfuerzo extra de procesamiento.

Para ser reproducido entonces, un archivo de sonido debe ser cargado a memoria, descomprimido (si hubiese sido comprimido antes) y luego ser enviado mediante una llamada a una librería o directamente al sistema operativo, para que este mande a su vez el flujo de samples utilizando el respectivo *driver* a la placa de sonido que hará el trabajo inverso al de la captura del sonido. Éste transformará las señales digitales en analógicas (Mediante un DCA) para poder ser transmitidas al usuario por medio de parlantes o auriculares (ver figura 6.2.7.2).

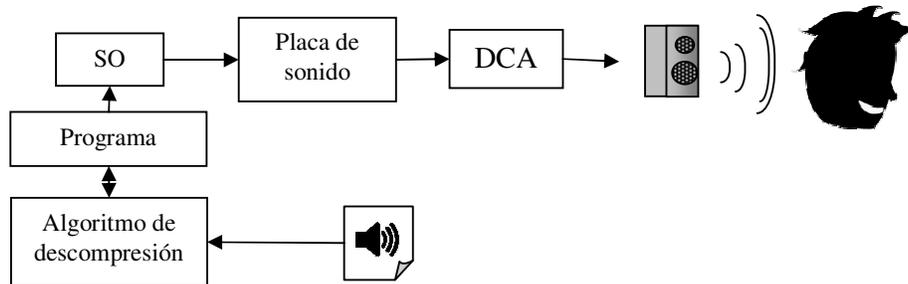


Figura 6.2.7.2 Reproducción de un sonido.

En cuanto al manejo de la música existen tres enfoques diferentes:

- Música generada en tiempo real: Es producida a partir de la reproducción de sonidos durante el juego y como resultado de decisiones por parte del jugador.

- **Música interpretada:** Esta se produce interpretando un archivo de configuración en donde se especifica cuando, con que duración, con que tono, y en que orden deben ser reproducidos una serie samples de instrumentos (o bien instrumentos generados por sintetizadores de modulación de frecuencia, en cuyo caso recibe el nombre de *tracked music*). El formato MIDI es un buen ejemplo de formato de archivo de música interpretada a partir de samples pregrabados [162]. El problema de este tipo de música en general, es que, el conjunto de samples utilizados es propio de cada dispositivo por lo que la música puede no escucharse siempre igual [162].
- **Música en formato de archivo de sonido:** Puede haber sido grabada a partir de una interpretación con instrumentos reales y así también puede ser el resultado de algún programa de composición de música digital [162]. Sea cual esa el caso, la música en este formato puede ser tratada, desde el punto de vista técnico, como un efecto de sonido.

Un sistema de audio diseñado para videojuegos debe tener en cuenta, por lo tanto, las siguientes cuestiones:

- Carga de sonidos desde archivos.
- Algoritmos de descompresión.
- Reproducción simultanea de sonidos.
- Manejo de música.
- Aplicación de filtros para lograr efectos de sonido.

Es característico, al igual que sucede con la representación gráfica, que se utilice una API creada por terceros para el manejo de los sonidos y la música [165][162][163]. Las más populares son:

- **Directx Audio:** Es una de las más distribuidas librerías de audio, y forma parte de las tecnologías de DirectX de Microsoft. Por esta razón es de uso exclusivo para el sistema operativo Windows, y las consolas Xbox y Xbox360. Proporciona una interfaz que permite manejar recursos de audio tanto a bajo

como a alto nivel, soportando funciones para la carga de sonidos en diferentes formatos, como para la aplicación de efectos, mezclas y simulación de sonidos en 3D [162][165].

- **OpenAL:** Es una API de audio tridimensional multiplataforma disponible para PC y Mac. Esta principalmente enfocada en la utilización de fuentes de sonidos en espacios tridimensionales. Provee un acceso de bajo nivel a la aplicación a los recursos de audio, y provee un conjunto de funcionalidades y facilidades comunes a diferentes plataformas [165].

- **Mac OS X Core Audio:** Es una API eficiente y flexible para la utilización de audio que sirve para manejar audio digital, y al mismo tiempo procesamiento de señales digitales (DSP) a través de bloques conocidos como unidades de audio [165] para el sistema Mac.

- **Open Sound Systems (OSS):** Es una API que articula un conjunto de *drivers* de dispositivos a través de la mayor parte de las arquitecturas Unix, y provee capacidades para la reproducción de audio sincronizado para video, y reproducción de animaciones.

6.2.8 Inteligencia Artificial.

El campo de inteligencia artificial (IA) es muy amplio en cuanto a técnicas y aplicaciones, sin embargo, el campo que concierne a los videojuegos es más acotado y pertenece a lo que se conoce como IA débil. La IA débil (IA *weak*) involucra una serie de algoritmos y procesos que tienen por objetivo brindar soluciones especializadas a problemas que requieren de cierta inteligencia [151].

La IA en videojuegos es utilizada para una variedad de problemas que comprenden ejemplos como la simulación del comportamiento de seres vivos, búsquedas de caminos (*path finding*), simulación de grupos de individuos, e incluso la generación de contenido procedural [152].

La complejidad de los algoritmos utilizados se encuentra acotada por varias razones:

- Los algoritmos utilizados cuentan con una restricción temporal y de recursos de procesamiento y memoria. Estos recursos deberán ser

compartidas con el resto del juego (física, colisiones, lógica del juego, renderización) [151].

- Los algoritmos utilizados deben brindar facilidades para permitir hacer ajustes (para conseguir los comportamientos deseados) y para la depuración [151].
- No es necesario lograr que los comportamientos simulados sean lo más realistas posible, es suficiente en cambio, simplemente con dar la ilusión de ello [153]. El criterio, en general, es mantener los algoritmos utilizados lo más simples posible [152][153].
- Un comportamiento predecible o simple puede ser suficiente según las restricciones que impone el diseño del juego [153].

La IA para videojuegos hace uso principalmente de técnicas como las máquinas de estados finitos, de algoritmos para la búsqueda de caminos (comúnmente A*), lógica difusa, técnicas de vida artificial (A-life) y árboles de decisión entre otras [151][152][153].

Distintos géneros de videojuegos tienen en consecuencia distintas necesidades de IA, los géneros como los RTS y FPS son particularmente los que más hincapié hacen en este campo. La necesidad de oponentes inteligentes para asegurar una variedad de juegos y situaciones es muy importante en ese tipo de juegos [152][153]. No obstante, aunque muy escasos, también existen juegos particularmente centrados en IA como **Creatures** (Cyberlife) y **Black and White** (Lionhead)[153].

Existen dos grupos de técnicas de IA en videojuegos:

- Las determinísticas: Son aquellas donde el comportamiento generado es predecible. Ejemplos más comunes pueden ser las máquinas de estados finitos y algoritmos para planificar y anticipar movimientos.
- No determinísticas: Son al contrario de las anteriores, de alguna manera, impredecibles. Las técnicas como redes neuronales, las técnicas bayesianas, o los algoritmos genéticos son ejemplos de ellos.

Las técnicas determinísticas son en general, más fáciles de implementar, de entender, de probar y de depurar [151]. Es por ello que son más populares dentro del campo de IA para juegos, sin embargo pueden ser no adecuadas para todos los tipos de juegos. La necesidad de una inteligencia de adaptación o de aprendizaje o incluso, la necesidad misma de impredecibilidad puede ser importante en algunos juegos.

Existen algunas técnicas para agregar impredecibilidad a un comportamiento determinístico mediante el uso de lógica difusa, introduciendo incertidumbre a través del uso, por ejemplo, de números aleatorios para la definición de umbrales en la toma de decisiones.

Las técnicas no determinísticas por el contrario son más complejas de desarrollar y de depurar pero ofrecen la posibilidad de presentar nuevos desafíos al jugador gracias a su impredecibilidad y la capacidad de algunas de estas técnicas de aprender y de adaptarse al jugador. Las situaciones que se presentan logran lo que se conoce como comportamiento emergente (es decir comportamiento que no fue directamente programado o diseñado) [151].

Las aplicaciones más usuales de las técnicas de IA son destinadas para: planificación de movimiento, la toma de decisiones, y para tomar decisiones conforme a tácticas y estrategias [153].

Como se ha mencionado anteriormente los videojuegos siempre han tenido como tendencia la búsqueda de representaciones gráficas más realistas, por lo que en general, el avance en otros campos como el de IA no ha sido aún el objeto de un gran esfuerzo. No obstante, los siguientes juegos se destacaron esencialmente por haber hecho hincapié en distintos aspectos de IA: **The Sims** (Maxis) usando conceptos de A-Life para simular seres humanos, **Creatures** (Cyberlife) por utilizar las mismas técnicas para la simulación de seres vivos (incluido un ADN digital simulado), **Black and White** (Lionhead) por utilizar técnicas no determinísticas para simular aprendizaje de una criatura, **Thief: The Dark Project** (Looking Glass Studio) por ser uno de los primeros en incorporar el concepto de “sensar” donde la IA solo puede tener conocimiento sobre lo que percibe (en lo que concierne al jugador), y finalmente **Warcraft** (Blizzard) y **Age of Empires II** por contar con IA desafiante en términos de desarrollo y ejecución de estrategias, conforman algunos buenos ejemplos de estos juegos [151][152][153].

A continuación se describirán algunas de las técnicas anteriormente mencionadas.

6.2.8.1 Maquinas de estados finitos.

Uno de los primeros juegos en utilizar esta técnica fue **Pac-man** (Namco)[151][153]. Los fantasmas del juego tenían un comportamiento programado en tres estados distintos: uno que le permitía vagar por el laberinto; otro que le permitía perseguir al jugador; y el último estado que lo hacía huir del jugador cuando este comía la bola que hacía invencible a Pac-Man y vulnerables a los fantasmas.

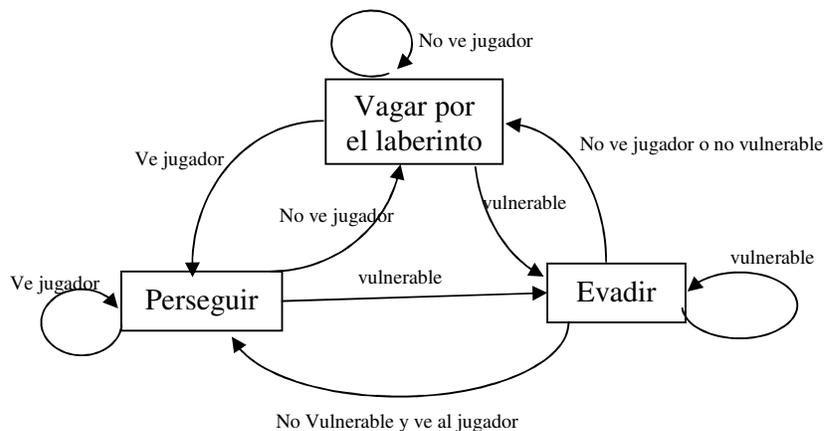


Figura 6.2.8.1.1 Ejemplo de máquina de estados finitos para un fantasma de Pac-Man (Namco)⁷.

Estos tres estados tan simples lograban darle al jugador la sensación de que los fantasmas contaban con cierta inteligencia. No obstante el comportamiento de los cuatro fantasmas no es idéntico y su forma de moverse por el laberinto y de tomar la decisión de perseguir o no, es distinta en cada uno. Esto muestra que por medio de la elección de unas cuantas reglas simples de movimiento se lograba mostrar un comportamiento lo suficientemente complejo [154] para presentar desafíos al jugador mediante la utilización de una técnica de IA determinística.

Las máquinas de estados finitos están formadas básicamente por dos conceptos: estados que definen acciones preprogramadas para cada uno de ellos, y un conjunto de reglas que determina la transición entre estos estados [151]. En la figura 6.2.8.1.1 se puede observar un ejemplo ilustrativo de la maquina de estados.

⁷ Este esquema es solo a modo ilustrativo de basado en [151] con algunas correcciones. No intenta representar el comportamiento real de los fantasmas del juego Pac-Man (Namco).

Cada uno de los rectángulos representa un estado distinto, y cada una de las flechas son las transiciones de los estados acompañados por las condiciones. Por ejemplo estando en el estado “vagando por el laberinto” si se cumple la condición de que el fantasma esta vulnerable, entonces cambia al estado “evadir”.

El uso de maquinas de estados finitos es muy popular todavía hoy en día [151][152][153] y es utilizado e implementado de diversas formas: mediante la definición de clases abstractas para manejar FSM (Maquinas de estados finitos), programado de manera explicita y embebida en el código, mediante la utilización de éstas junto un lenguaje de *scripting*, o bien lo que se conoce como jerarquía de FSM donde dentro de cada estado puede existir a su vez otra FSM. Sin embargo la lógica detrás de la FSM se mantiene constante como ha sido caracterizada.

6.2.8.2 Lógica difusa.

La lógica difusa es un concepto desarrollado originalmente por Lotfi Zadeh en 1965, y que plantea que un elemento tiene cierto grado de pertenencia a un conjunto (en lugar de utilizar lógica booleana para decidir si “si o no” pertenece). Esto se asemeja a la manera en que los humanos pensamos permitiendo operar con cierto margen de error y sobre datos menos precisos [151]. Si bien los problemas que se pueden resolver con lógica difusa, pueden ser también resueltos utilizando la lógica tradicional, en general los sistemas que utilizan lógica difusa producen menor cantidad de reglas (entre un 50 y un 80% menos [151]), son más legibles, más fáciles de escribir y de mantener, por lo que coinciden positivamente con el criterio dado al inicio de la sección.

En la lógica difusa el grado de pertenencia varía de manera continua (por convención) de 0 a 1. Por ejemplo en el caso de valores de verdad, 0 representaría completamente falso, y 1 completamente verdadero.

A diferencia de la lógica tradicional donde dos conjuntos de elementos pueden ser mutuamente excluyentes, no sucede así en la lógica difusa [152]. Por ejemplo una unidad puede considerarse saludable y dañada al mismo tiempo (cada uno en un grado distinto), no siendo verdadero en el caso de la lógica tradicional donde esto no tendría sentido (Ver figura 6.2.8.2.1).

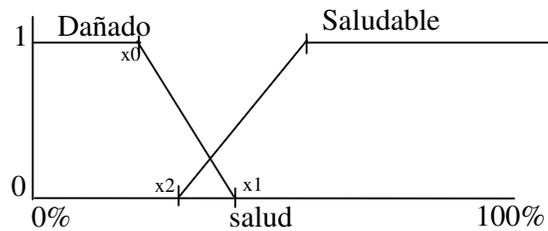


Figura 6.2.8.2.1 Determinación de valores de salud en lógica difusa.

Dado que los datos y el procesamiento en videojuegos no se realiza por medio de la lógica difusa para utilizarla se necesita de mecanismos de conversión en ambos sentidos. Es decir, se necesita de un proceso de para convertir los datos hacia un formato entendible por la lógica difusa (proceso conocido en ingles como “*fuzzification*”) y una vez procesados mediante la aplicación de reglas de lógica difusa, se deben volver a poder interpretarse por el juego mediante el proceso de de-*fuzzification* [151][152] (ver figura 6.2.8.2.2)

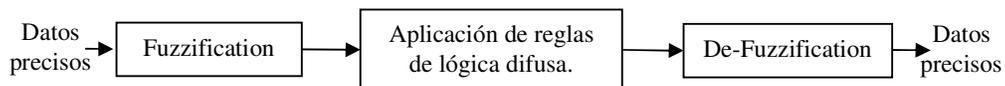


Figura 6.2.8.2.2 Proceso para la integración de lógica difusa.

Los valores generalmente a convertir son enumeraciones y datos booleanos. La solución más frecuente es utilizar para ello valores predeterminados para cada conjunto (Ver figura 6.2.8.2.1) [152]. De esta manera, en el ejemplo, una unidad se considera dañada en grado 1 de 0% de salud hasta x_0 , y desde x_0 y x_1 el valor de grado de daño disminuye hasta alcanzar 0 en x_1 . Se puede observar en este ejemplo también que el rango en que se considera saludable (con grado >0) sucede a partir de x_2 (el cual es menor x_1 , el valor en que el grado de pertenencia de dañado comienza a ser 0).

El proceso inverso, por el contrario, es un poco más complejo, ya que existen áreas de solapamiento, totalmente validas en lógica difusa, pero sin sentido en la lógica tradicional [152]. Existen además soluciones preestablecidas y librerías completas, bajo distintas licencias, para el tratamiento de lógica difusa [153].

La lógica difusa en el campo de videojuegos puede ser aplicada en:

- **Control:** Gracias al concepto de grado de pertenencia, la lógica difusa puede ser utilizada para sistemas de control que den respuestas graduadas para posibilitar mostrar comportamientos más “suaves” y realistas. Por ejemplo el control de movimiento de una unidad, que en lugar de determinar bruscamente un ángulo y una velocidad, pueden determinarse dentro de un rango continuo estás en base a las decisiones tomadas por el sistema de ilógica difusa [151].
- **Clasificación:** La utilización de la lógica difusa para la clasificación de situaciones u objetos permite la determinación más simple de las reglas de la clasificación (donde por definición están comprendidas por grados de pertenencia), y además, la utilización de estos rangos puede permitir exhibir comportamientos menos predecibles. Por ejemplo utilizado en una IA de un enemigo para evaluar la amenaza de un conjunto de unidades conforme cantidad y calidad de estas [151].
- **FSM y Lógica difusa:** Existen varias formas de combinar estás dos técnicas, desde incorporar lógica difusa para el sistema de decisión para el cambio de estado, hasta la definición de estados difusos (por ejemplo estado agresivo de una unidad en valuado en diferentes grados) [152].

6.2.8.3 Árboles de decisiones

Los árboles de decisiones son otra técnica sencilla utilizada en el desarrollo de videojuegos. Los árboles de decisión son fáciles de implementar, de entender y de usar.

Esta técnica consiste en una serie de decisiones conectas que comienzan por una condición raíz, y que a través de sucesivas decisiones se termina designando una acción. Las decisiones pueden ser de tres tipos: booleanas, de enumeraciones y de valores numéricos [153].

- Las decisiones booleanas: dividen el árbol en dos, según la condición de verdadero o falso.

- Las decisiones de enumeraciones: dividen al árbol en tantas ramas como elementos de la enumeración. Por ejemplo la enumeración colores primarios, tiene tres elementos: rojo, amarillo y azul; por lo tanto dividiría a esa rama en tres posibles caminos.
- Las decisiones de valores numéricos: se divide al árbol en rangos de valores. Por ejemplo para la distancia hacia un enemigo, se puede utilizar el criterio: menos de 50, más de 50 y menos de 100, y más de cien. Los rangos deben ser mutuamente excluyentes.

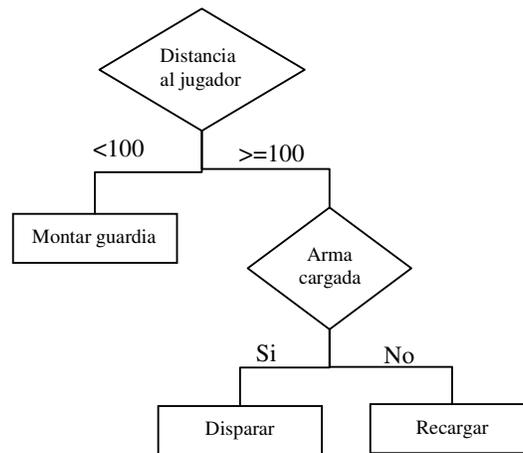


Figura 6.2.8.3.1 Ejemplo de árbol de decisión.

En la figura 6.2.8.3.1 se puede ver un ejemplo de árbol de decisión para un enemigo. Al ejecutarse la IA utilizará su conocimiento para tomar las decisiones, comenzando por la raíz, en el ejemplo, comparando la distancia al jugador, si esta fuese menor a 100, pasará a montar guardia, de lo contrario, si tiene el arma cargada disparará, y si su arma esta descargada tomará la acción de recargar.

Para lograr la mayor eficiencia en la toma de decisiones en un árbol es necesario que este esté balanceado. Un árbol balanceado es aquél que para cada decisión la cantidad de hijos (decisiones y acciones) que tienen sus decisiones (o acciones) que delega no difieren en más que uno (en la práctica que sea la menor diferencia posible). De esta manera, se logra que el árbol de decisión requiera menos decisiones para tomar una acción en promedio, y por lo tanto menos tiempo de ejecución. No obstante el tiempo requerido para tomar cada decisión no es constante ni homogéneo (No es lo mismo, por ejemplo, comparar el valor de una variable, que consultar si en una lista existe X valor), por lo que lograr aumentar en la práctica la eficiencia del árbol es una

tarea difícil. No obstante, aún sin optimizaciones es una técnica muy eficiente para la toma de decisiones [153].

6.2.8.4 Pathfinding: A* (A estrella).

En la mayoría de los videojuegos (si no en todos) existe la necesidad de que las unidades se desplacen por el espacio de manera inteligente. En aquellos juegos donde las unidades no tengan un camino predefinido por el diseñador es necesaria la utilización algoritmos que permitan planificar los caminos que utilizarán las unidades para desplazarse. Este camino debe ser optimo (menor distancia posible) y con la ausencia de obstáculos para que el jugador perciba que el movimiento de la unidad cuenta con cierta inteligencia [153].

Un algoritmo muy utilizado para este fin es el A*, este algoritmo asegura encontrar el camino con la mínima distancia (si existe) entre dos puntos y además logra hacerlo de forma eficiente [151].

Para poder utilizar sin embargo, cualquier algoritmo de búsqueda de caminos, es necesario crear una representación simplificada del mundo del juego de manera que se minimicen la cantidad de elementos a procesar. La estructura utilizada normalmente para ello es un grafo dirigido con peso [153].

Un grafo es una estructura matemática consistente de dos elementos: nodos y aristas. Los nodos pueden tener un conjunto de aristas que los conectan con otros nodos, mientras que una arista solo puede conectar dos nodos. Si el orden en que los nodos están conectados por una arista importa, se dice que el grafo es dirigido. Si además se le asigna un valor numérico a cada arista, se lo llama grafo con peso (Ver figura 6.2.8.4.1) [153].

Los nodos son utilizados para representar elementos lugares en el nivel, las aristas por otro lado son utilizadas para representar la adyacencia de posiciones, mientras que los pesos de las aristas codifican distancia (o bien podría ser tiempo) que hay de un nodo a otro.

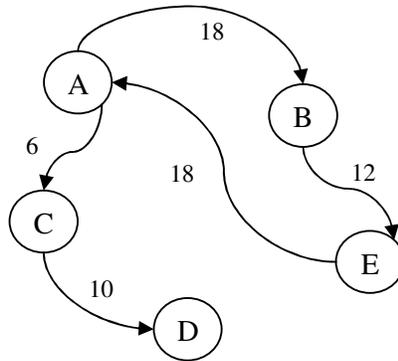


Figura 6.2.8.4.1 Ejemplo de grafo dirigido.

En la figura 6.2.8.4.2 se puede ver un ejemplo de la representación del mundo del juego en un grafo dirigido. Los obstáculos (en el ejemplo las rocas) se indican con la ausencia de aristas (caminos entre nodos) o bien con la ausencia de nodos. El peso de las aristas es 1, y si bien se obviaron las conexiones diagonales para simplificar el diagrama, se pueden agregar con peso aproximado a $\sqrt{2}$.

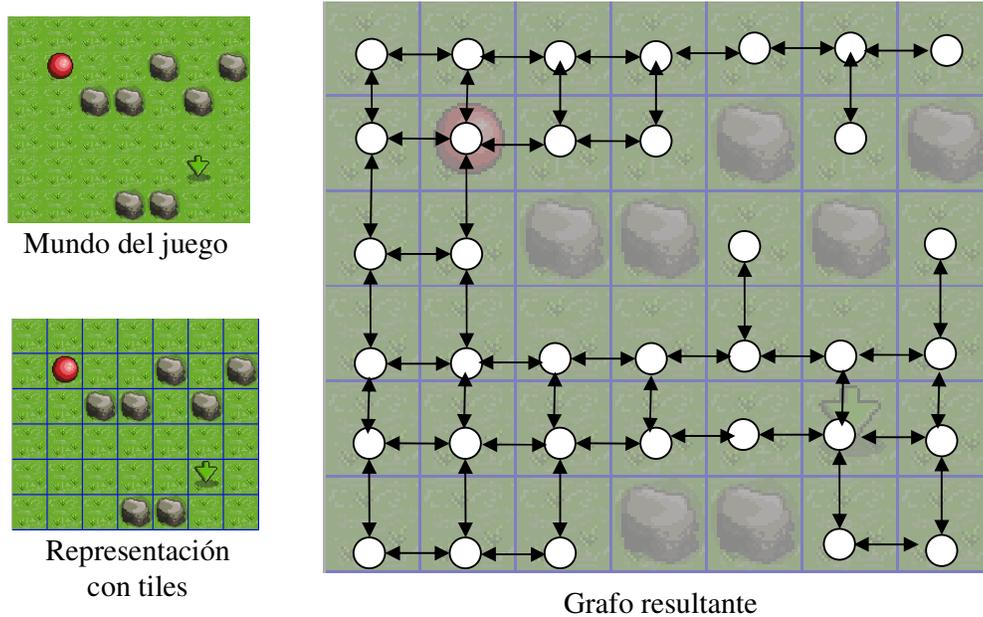


Figura 6.2.8.4.2 Representación del mundo del juego con un grafo dirigido.

El algoritmo A*, a diferencia de un algoritmo similar conocido como Dijkstra diseñado también para la búsqueda de caminos, utiliza heurísticas para orientar la búsqueda y hacerla más eficiente. La determinación de esta heurística no forma parte del algoritmo y en la práctica es una estimación de la distancia que tienen los nodos hasta el nodo

destino. De esta manera la heurística es única por cada búsqueda, y la misma para el mismo par nodo partida y nodo destino.

El algoritmo funciona bajo el criterio primero el mejor, donde se analiza (partiendo del nodo de inicio) primero aquellos nodos que tienen un menor costo asociado. El costo asociado es el costo acumulado (la suma de la distancia de los nodos visitados en el camino analizado) con adición a la heurística de cuanto falta para llegar al nodo objetivo. Antes de agregar nodos al camino se descartan aquellos nodos que formen un *loop* (que vuelvan al nodo inicial). Por cada nodo agregado se genera un camino nuevo y se agrega a la lista de caminos a explorar, siempre y cuando este camino tenga nodos por explorar, o bien no alcance un nodo con mayor distancia que otro camino previamente almacenado. Una vez agregado el nuevo camino, se elige para analizar aquel camino con menor costo asociado, de esta manera el algoritmo finaliza cuando se encuentra el nodo objetivo y siendo la respuesta el primero de estos caminos. En caso de no encontrarse, entonces el algoritmo termina cuando se vacía la lista de nodos a explorar [153][155].

6.2.8.5 Técnicas no determinísticas para aprendizaje.

Las técnicas de aprendizaje ofrecen la posibilidad de crear una IA con la capacidad de adaptarse a entornos, a situaciones e incluso al modo en que distintas personas juegan [152]. El jugador, en consecuencia, puede encontrar interesante enfrentarse ante una IA que cambia su comportamiento en búsqueda de una mejora continua ofreciendo situaciones desafiantes en cada partida [152][153][151].

Alternativamente algunos juegos utilizan lo que se conoce como falso aprendizaje (*fake learning*) donde por medio de probabilidades y tasas de error se simula un comportamiento menos inteligente que el que la IA es capaz de ofrecer, aparentando de esta manera distintos niveles de conocimiento sobre el problema [152].

Por otro lado, las expectativas generadas por las verdaderas técnicas de aprendizaje son en general mal interpretadas y no han logrado conseguir el potencial que demuestran en la teoría debido a que en la práctica presentan algunas dificultades propias de la naturaleza del aprendizaje [152]. Cuánto más se delega a la tarea de los algoritmos de aprendizaje, más difícil resulta depurar y controlar la IA resultante [152][153]. Para lograr que una IA de aprendizaje se comporte de la misma manera, es necesario someterlo al mismo conjunto de entrenamiento (situaciones de juego) y esto

a menudo puede resultar impracticable y al mismo tiempo necesario para la búsqueda y detección de errores [153].

A si mismo, algunas técnicas de aprendizaje como las redes neuronales, pueden requerir de un conjunto de aprendizaje muy grande para lograr resultados interesantes, algo que en la práctica (dependiendo del tamaño de la red) [155] puede ser muy difícil de lograr [152][153]. Por el contrario, las técnicas de aprendizaje que requieren de un conjunto menor de entrenamiento (como los árboles de decisiones con aprendizaje) pueden ser más susceptibles a incorporar errores como conocimiento. Otra cuestión relacionada es el denominado sobre aprendizaje, que resulta de la especialización de una IA sobre un conjunto de situaciones haciéndola ineficiente para situaciones nuevas que difieran de éstas [153].

No obstante, teniendo en consideración las particularidades de cada una de las técnicas es posible utilizarlas en el ámbito de los videojuegos en diversas aplicaciones [153]:

- Para el ajuste de parámetros.
- Predicción de acciones.
- Aprendizaje de decisiones.

La mayoría de las técnicas de aprendizaje no son utilizadas en tiempo real, ya que gran parte de estas se entrenan con grandes grupos de entrenamiento [153]. Un criterio utilizado en IA de videojuegos para facilitar y acelerar el ritmo de aprendizaje es el de incluir *a priori* tanta información como sea posible [152]

Analizar con detenimiento las distintas técnicas de IA para aprendizaje escapa al alcance del trabajo, por lo que se presentan a continuación las más utilizadas junto a una breve caracterización:

- Árboles de decisiones con aprendizaje: se basan en los árboles de decisión, donde a partir de un conjunto de observaciones y acciones, se infiere por medio de un algoritmo la estructura del árbol. Tienen la característica de requerir de un conjunto de entrenamiento pequeño, pero al mismo tiempo, de

ser susceptible a errores. Por estas razones, requieren por lo general, de supervisión [153][152].

- **Aprendizaje por refuerzo:** Son un conjunto de técnicas que se basan en el concepto de retroalimentación. Por cada decisión tomada se califica al sistema de IA para saber si debe realizar o no ajustes en su forma de actuar [153].
- **Redes Neuronales:** Están inspiradas en la biología, e intentan simular el funcionamiento de las neuronas, constituidas por componentes relativamente sencillos que aplican los mismos algoritmos pero que están conectados por distintos tipos de enlaces (que pueden activarlas o inhibirlas) [155]. Las redes neuronales no son utilizadas por lo general en videojuegos, pero son versátiles en cuanto a rango de aplicación y se destacan en tareas de clasificación [153].
- **Algoritmos genéticos:** tienen un funcionamiento análogo al de la selección natural. Se generan poblaciones de individuos (conjuntos de parámetros para dar solución a un problema) que luego se evalúan para poder seleccionar los mejores. Estos combinan sus valores de parámetros bajo un criterio establecido para la cruce, para crear así una nueva generación para ser evaluada nuevamente. Cuenta con mecanismos como las mutaciones (variación aleatoria de un parámetro), y distintas variantes para la selección de individuos, de mecanismos de cruce y de mutación [155] Los algoritmos requieren de mucho entrenamiento para lograr resultados, no obstante los valores iniciales pueden ser configurados a mano para acortar los tiempos de búsqueda.

6.2.8.6 Scripting.

El *scripting* surge como una necesidad de desarrollar con facilidad aquellas partes del juego que no requieren de mucha performance, pero si en cambio, de facilidad y rapidez de creación, de acceso y de modificación. El *scripting* consiste en la creación de scripts (pequeños archivos de código) en un lenguaje simple para poder resolver problemas específicos de un juego. Se pueden integrar lenguajes como Lua, o Python de carácter más general o bien, desarrollar un lenguaje simplificado propio para realizar dicha tarea [151]. La ventaja que ofrece esta técnica es que además de ayudar a separar y organizar el código, permiten realizar cambios sin necesidad de recompilar el juego siendo mucho más versátiles para hacer pruebas y ajustes en poco tiempo.

Otra consecuencia es que permite a diseñadores probar e introducir cambios sin la necesidad de acudir al equipo de programación, acelerando en consecuencia el flujo de trabajo [156].

El *scripting* es hoy en día una herramienta muy popular entre los motores de juegos [156] y una herramienta muy importante al momento de:

- Definir y ajustar el comportamiento de IA, como por ejemplo los estados de una FSM [151].
- Atributos de unidades, enemigos, y entidades del juego.
- Eventos: sucesos a partir de la posición del jugador, del tiempo que transcurre desde determinada acción etc.
- Diálogos: árboles de diálogos para facilitar su creación, mantenimiento y traducción.
- Particularidades de los niveles.

6.2.9 Juegos multiplayer y Redes.

La posibilidad de jugar con otras personas, característica conocida como multi-jugador o *multiplayer*, puede brindarse de dos maneras.

En la misma consola/PC utilizando estrategias como: turnos (un jugador debe esperar a que otro concluya su jugada), compartir la misma pantalla (ambos jugadores juegan al mismo tiempo y visualizan los resultados en la misma pantalla con la misma GUI), o bien en pantalla dividida (ambos jugadores en simultaneo sobre una misma pantalla dividida cada uno con su respectiva GUI). Sea cual fuere la estrategia, las técnicas para procesar y dar soporte a varios jugadores en el mismo dispositivo son las mismas que las anteriormente descritas. Las entradas de los jugadores, sus interacciones y su visualización son manejadas por el juego y sus respectivos subsistemas.

Otra posibilidad diferente es brindar soporte multiplayer a personas que utilizan diferentes consolas/PCs. Cuando el juego se ejecuta en diferentes máquinas, es necesario contar con un mecanismo de comunicación para lograr que jugadores en

lugares físicos diferentes (y probablemente distantes), disfruten de una experiencia de juego compartida.

Para lograr este fin es necesario utilizar una infraestructura que permita a los dispositivos comunicarse entre sí de manera confiable y eficiente ya que problemas tales como demoras, o pérdidas de información se traducen directamente en experiencias de juego insatisfactorias [157] [158].

Los videojuegos se apoyan sobre la infraestructura ya existente de redes conocida como la pila o familia de protocolos TCP/IP [157], la cual consiste en una extensa colección de protocolos que forman parte de los estándares de Internet (según la IAB Internet Architecture Board) [159].

Esta familia de protocolos estructura al problema de la comunicación en cinco capas diferentes e independientes que se relacionan con las adyacentes y prestan servicio a su capa de nivel superior (Ver figura 6.2.9.1) y se definen a continuación [159]:

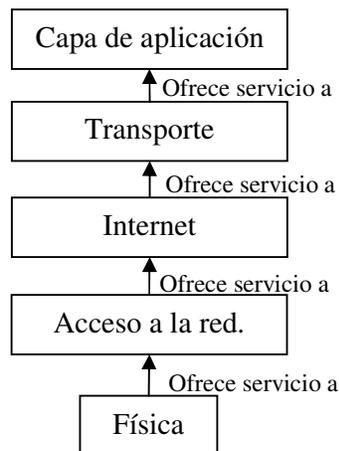


Figura 6.2.9.1 Familia de protocolos TCP/IP.

La capa física: Es la que resuelve el problema a más bajo nivel, encargándose de ocultar y manejar eficientemente la interfaz física, lidiando con cuestiones como el medio de transmisión y la naturaleza de las señales [159].

- **Capa de acceso a la red:** tiene la responsabilidad de permitir el intercambio de datos entre sistemas finales en la red a la cual está conectado

[159]. El protocolo Ethernet y el protocolo 802.11 (Wifi) son ejemplos de esta capa y los más populares en el campo de videojuegos [157][158].

Si bien por excelencia el dispositivo de juegos multiplayer con características de red es la PC, esta situación se está revirtiendo con las consolas de la actual generación todas ellas con conexiones de una o más interfaces Ethernet, Wifi, 3g y con sus respectivas plataformas online (tema ya tratado en la sección de plataformas).

- **Capa de Internet:** Tiene la responsabilidad de permitir el intercambio de datos entre sistemas que se encuentran en diferentes redes ofreciendo mecanismos para el encaminamiento de datos a través de equipos conocidos sistemas encaminadores (con dicha responsabilidad) hacia los sistemas finales. El protocolo utilizado en esta capa es IP y tiene por lo tanto la tarea del direccionamiento y entrega de los datos a través de las redes [159]. La versión más reciente de IP, la IPV6, fue diseñada con el objetivo de cubrir las carencias de su predecesora, ampliando el rango de direccionamiento e incorporando nuevas funcionalidades debido principalmente a que el protocolo IPV4(la versión actual) está alcanzando su límite de capacidad[160], no obstante, aún no es sujeto de estudio (ni de práctica) en el campo de videojuegos [157].

- **Capa de transporte:** la responsabilidad de esta capa es de proveer mecanismos para el intercambio de datos entre extremos (sistemas finales) aislando a las capas superiores de los detalles y complejidad de la red o redes que haya de por medio [158]. Mediante una interfaz de programación conocida como *socket* (identificadas por el par dirección IP, puerto) [158] esta capa va a proporcionar funcionalidad a todas las aplicaciones [159] que hagan uso de la red. Los protocolos principales en esta capa son TCP y UDP.

TCP tiene la particularidad de ser un protocolo orientado a la conexión que garantiza que la información enviada llegue a su destino, en orden y sin duplicados. Para ello emplea mecanismos de serialización y de buffering mediante una ventana deslizante. Las tramas perdidas se retransmiten, y cuando llegan, se almacenan en un buffer [159]. Estas características lo hacen ideal para su uso en aplicaciones que no tienen restricciones temporales importantes, sin embargo, dado que la latencia (tiempo que tardan en llegar los

mensajes desde el origen al destino) es variable (retransmisión, organización de las tramas), resulta inadecuado para su utilización en la mayoría de los videojuegos [157].

El protocolo UDP por contrario, es un protocolo mucho más simple que no implementa los controles y funciones que provee TCP. Por esta razón existe la posibilidad de que se pierdan mensajes o que estos lleguen desordenados por lo que se requiere de un esfuerzo mayor de programación en la capa superior para controlar estos problemas.

- **Capa de aplicación:** esta capa contiene toda la lógica necesaria para posibilitar a las aplicaciones dar servicio a los usuarios [159]. El protocolo que se creará para lograr la comunicación necesaria por el juego está, por lo tanto, situado en esta capa.

Existen diversas formas de estructurar un sistema que opera a través de la red. Estas formas son conocidas como modelos o arquitecturas, y comprenden básicamente dos enfoques:

- **El modelo *peer to peer* (p2p):** En este modelo todos los nodos del sistema tienen las mismas responsabilidades y cooperan entre sí mediante el intercambio de mensajes para lograr el objetivo del sistema (ver figura 6.2.9.2) [161]. La ventaja de esta arquitectura es que el sistema es tolerante a fallas por naturaleza, ya que todos los nodos tienen la misma importancia, sin embargo, en la práctica en el campo de videojuegos, es más costoso mantener un estado consistente del juego dado que todos los jugadores implicados en una misma partida deben percibir del mismo modo el mundo del juego, y al mismo tiempo tienen y producen modificaciones sobre este [158].

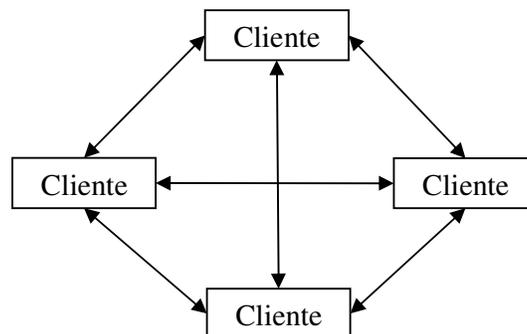


Figura 6.2.9.2 Modelo *peer to peer*.

Este modelo es principalmente útil en redes de área local (LAN) ya que los protocolos como Ethernet soportan mensajes de *broadcasting* (mensaje de un nodo hacia todos los demás) y la cantidad de participantes es relativamente baja. Algunos juegos como **Doom**, utilizaron este modelo, sin embargo, el modelo p2p aplicado para videojuegos no escala bien (al menos sin modificaciones importantes) para redes de área amplia (WAN) como internet [157].

- **El modelo Cliente-Servidor:** Es el modelo más popular y estudiado en sistemas distribuidos, y se basa en la definición de dos nodos con diferentes funciones y responsabilidades (Ver figura 6.2.9.3). El Servidor tiene la tarea de ofrecer funcionalidad a los clientes, y se comunica con ellos por medio del intercambio de mensajes. Los clientes solicitan funciones de los servidores, y solo se comunican con estos ya que no existen comunicación directa entre los nodos clientes [161].

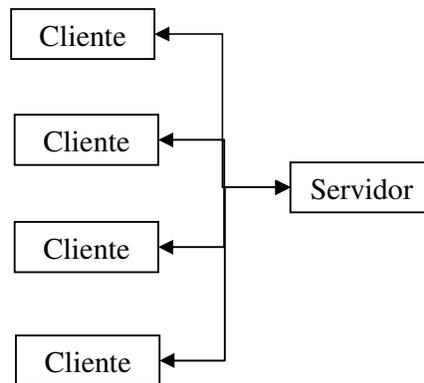


Figura 6.2.9.3 Modelo Cliente Servidor.

Este modelo es el más utilizado en videojuegos donde el servidor es el encargado de mantener un estado consistente del mundo e informarlo a los clientes que están conectados, quienes a su vez, mandan información sobre las acciones del jugador [158]. Por esta razón, el servidor representa un cuello de botella para la escalabilidad, debido a que cada cliente conectado afecta en su performance, y en consecuencia a las demoras en el sistema completo. Existen estrategias como utilizar redes de servidores para dar servicio a los clientes, donde los servidores se comunican entre sí ya sea mediante una

arquitectura p2p o bien cliente/servidor, de esta manera, se puede mejorar la relación de escalabilidad a costo de complejizar el sistema [157].

Sea cual fuere el protocolo de comunicación, o la arquitectura del sistema, existen una serie de problemáticas que son comunes y están relacionadas a la naturaleza de los videojuegos multiplayer en red, y a internet. Estos problemas surgen de las características del protocolo IP basado en el criterio del “mejor esfuerzo”: el tiempo de entrega de los mensajes, la variación de este tiempo y la pérdida de mensajes plantean cuestiones que afectan a la experiencia de juego.

La latencia (anteriormente mencionada) se refiere al tiempo que toma un paquete en llegar desde el origen hasta el destino. Dado que este tiempo no es constante, también se utiliza la medida conocida como *jitter* que refleja las diferencias de tiempo que existen de paquete a paquete. Otra métrica utilizada para describir el comportamiento de la comunicación entre cliente y servidor es el lag, el cual es el tiempo que le lleva al cliente en conseguir la respuesta del servidor (Conocido en el mundo de la teleinformática como *round trip time* o RTT).

Para compensar los efectos de las demoras en las comunicaciones entre el cliente y el servidor, se desarrollaron técnicas con el objetivo de ocultar y minimizar el daño que se produce en la experiencia de juego:

- **Predicción:** antes de aceptar las entradas por parte del jugador y de mostrarlas en pantalla, el cliente queda a la espera de la confirmación por parte del servidor de que ésta acción que ha tomado es una acción válida.

Sin embargo el tiempo que le toma al cliente en conseguir esta confirmación puede interferir con la experiencia de juego provocando un retraso entre las acciones y su visualización en pantalla. Para solucionar este problema se puede implementar un sistema de predicción en el cual el juego, mantiene su propio estado en lo que concierne al jugador, prediciendo la respuesta del servidor (basándose en su posición, velocidad, dirección y obstáculos), y una vez llegada la información correcta del servidor se realizan los ajustes pertinentes. El problema que existe en general con cualquier sistema de predicción es que el mundo de juego que presenta el cliente es diferente al del servidor creando una inconsistencia que al momento de corregirse puede generar algunos artificios que le resulten molestos al jugador.

Este mismo sistema de predicción también puede aplicarse para unidades u objetos que no pertenecen al jugador utilizando los mismos principios, sin embargo también existen problemas de inconsistencias que según el género del juego pueden interferir negativamente el gameplay [157]. La figura 6.2.9.4 ilustra los efectos de utilizar o no predicción en el cliente del juego.



Figura 6.2.9.4 Uso de predicción [157].

- **Manejo del tiempo:** Dado que la latencia entre los distintos clientes no es la misma, puede darse el caso que los clientes que se encuentran más cerca al servidor (en términos de latencia) reciban con anterioridad el nuevo estado del juego. Según el tipo de juego esto puede crear una gran desventaja en aquellos clientes que tienen una actualización más lenta, por ello existen técnicas para poder manipular el tiempo en el juego y sortear algunos de estos problemas a costa de hacer menos respondente al juego.

La técnica conocida como *time delay* consiste en imponer, ya sea en el servidor o en los clientes, una demora para el envío o procesamiento de los mensajes de manera de tratar de igualar los tiempos de latencia en cada cliente.

Por otro lado la técnica conocida como *time warp* implementa un mecanismo mediante el cual calcula el tiempo en el que los mensajes llegan desde los clientes (considerando también la latencia) y ordena los eventos de estos en tiempo en que acontecieron. Una vez ordenados el servidor procesa los eventos, ahora en orden cronológico, reduciendo inconsistencia total del mundo del juego ya que permite que las acciones de los clientes se ejecuten conforme al estado que ellos percibieron del mundo del juego [157].

El mayor desafío en el desarrollo de videojuegos con características de multijugador es, por lo tanto, ofrecer a los jugadores una experiencia de juego fluida que oculte de manera eficiente la complejidad de la comunicación y que logre al mismo tiempo

mantener la coherencia del estado del juego para todos los jugadores implicados en la partida por igual.

7. Industria de los videojuegos.

En este capítulo se ofrece una descripción de la parte económica de los videojuegos, la caracterización de los aspectos principales de la industria que acompañada de estadísticas y números permiten dar una apreciación concreta de ésta. Al finalizar el capítulo, se exponen algunos de los modelos de negocios más utilizados por la industria de los videojuegos para ilustrar las características y la variedad de estos modelos.

7.1 Caracterización general.

La industria de videojuegos ha experimentado muchos cambios desde su creación, transitando un camino que comenzó con las primeras máquinas de arcade, hasta llegar en la actualidad a estar presentes en consolas, PC, tablets, teléfonos celulares, lectores de libros digitales, televisión digital, y en general en todo tipo de dispositivo con capacidades multimedia [11][14][167].

La industria global de videojuegos en el 2010 fue valuada entre 100 y 105 miles de millones de dólares. Nintendo, por su parte, posee aproximadamente el 33% de la industria global, siendo la empresa de videojuegos más grande del mundo. Las consolas y juegos de éstas, junto con los videojuegos de PC ocupan el 32%, mientras que los juegos online (tanto para consolas como para PC) ocupan el 22%. Los juegos para móviles, por su parte, ocupan un 8% del mercado global [177]. (Ver figura 7.1.1).

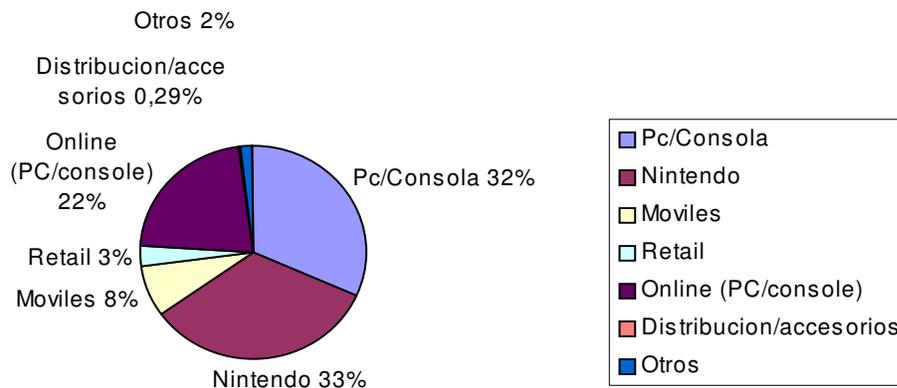


Figura 7.1.1 Distribución del mercado global en el 2010.

La industria de los videojuegos puede ser considerada como parte de las conocidas como industrias culturales [171][167]. Si bien originalmente este término “industrias culturales” nace en 1940 en EEUU como una forma de crítica hacia el fenómeno de la industrialización de la cultura propia del sistema económico capitalista, en la actualidad, este se utiliza para referirse a aquellas industrias que participan directamente en la producción, distribución y circulación de significados a través de formas simbólicas y que generan a su vez ganancias con dicha actividad. Dentro de este tipo de industrias se ubican entonces a la televisión, la radio, los diarios y revistas, industrias cinematográficas, discográficas, y a las editoriales.

Las industrias culturales se caracterizan además por tres fenómenos: el riesgo asociado a la producción de un bien cultural, costos de producción generalmente muy elevados y un costo relativamente bajo de reproducción junto con el carácter semipúblico de los bienes o servicios que ofrecen. Estas características por lo tanto son validas también para caracterizar a la industria de los videojuegos [117].

Al igual que otras industrias culturales esta industria se ve caracterizada por la existencia “hits” (productos muy exitosos) donde solo un pequeño numero de títulos son los responsables de gran parte de los ingresos [166] [167] [169].

Los costos y los procesos de producción han crecido los últimos años en la industria AAA, los presupuestos para desarrollar estos juegos llegan a varios millones de dólares y con tiempos de desarrollo que oscilan entre 18 y 24 meses empleando como mínimo a 20 personas. Estos valores, incluso, son superados por la industria japonesa cuyos presupuestos oscilan entre los 15 y 80 millones de dólares, empleando grupos de hasta 70 personas [171]. No obstante el mercado se ha diversificado en los últimos años en géneros, plataformas y también en su público objetivo. La distribución digital de la mano de empresas como Steam, los juegos casuales, los desarrollos para dispositivos móviles, los juegos sociales y los juegos serios, protagonizan esta diversificación del mercado y de la industria [169][171]. Los presupuestos, tiempos y cantidad de personas involucradas en estos sectores son en general mucho menores [167].

La industria de los videojuegos tiene un público global y en crecimiento. Según la ESA el 64% de los jefes de familia en EEUU juegan videojuegos, la edad promedio es de 34 años y la proporción masculino y femenino es 60% y 40% respectivamente (Ver el capítulo 3.5 para más información) [51]. Los videojuegos han pasado de estar

enfocados solamente a un público conformado por niños, a tener un público que abarca un amplio rango de edades [167]. Según una encuesta realizada en Europa a personas que no son jugadores sobre las razones por las cuales la que no juegan videojuegos, solamente el 8% respondieron que es porque son para niños [170]. Además según la ESA, en estados unidos el 26% de los jugadores tienen más de 50 años [51].

La industria de los videojuegos ha experimentado un crecimiento a través de los años luego del *crash* que sufrió en 1983 (Ver el capítulo 2.3.4). Según la ESA el crecimiento de la industria del software de videojuegos (en EEUU) creció un 16.7% entre los años 2005 y 2008, mientras que para la industria total de videojuegos fue de un 10.6%. Por otro lado en esos mismos años el crecimiento de la economía de EEUU fue solamente de 2.8% (de 2005 a 2008) [94]. En Japón por otro lado debido al surgimiento y fortalecimiento de los competidores occidentales, a disminuido las tasas de exportación de consolas y software de videojuegos [172]. En la actualidad el crecimiento general de la industria se ha frenado [173][174], sin embargo el sector dedicado a juegos online y juegos para móviles se ha incrementado levemente con respecto a 2009 y 2010 [174]. El segmento de juegos sociales por otro lado, ha experimentado un rápido crecimiento. El 30% de los usuarios de Facebook (la principal plataforma de los juegos sociales [190]), son usuarios de este tipo de juegos lo que representa más de 500 millones de usuarios. En 2009 la industria de juegos sociales tuvo una ganancia de mil millones de dólares, y se espera que para el año 2012 este número crezca hasta alcanzar los 3 mil millones de dólares [189].

La industria de los videojuegos es en la actualidad un sector económico muy grande que supera incluso a otras industrias culturales como el cine y la música [169].

7.2 Estructura.

Es necesario, para comprender mejor la industria de los videojuegos, entender como está esta formada. Existen tres grupos con características y dinámicas diferentes: el del hardware, el de los videojuegos en si mismos, y el de la infraestructura y la tecnología [169].

El grupo de hardware se refiere aquél sector que engloba a los fabricantes de las plataformas de hardware sobre los cuales son jugados los videojuegos.

La infraestructura y tecnología, se refiere a aquel sector que abarca el apoyo necesario para la distribución y la oferta de la tecnología necesaria para jugar.

Por último los videojuegos, también llamada la parte del software de la industria [167], comprenden a los desarrolladores, los publicadores, y los distribuidores (caracterizados en la sección 5.1 y vueltos a tratar en la sección siguiente 7.3) quienes integran el proceso de creación y de distribución de los juegos [167][169][170].

En cuanto a los dispositivos propios de juegos (las consolas), el sector del hardware se caracteriza por ser un oligopolio donde pocas empresas son las que tienen protagonismo [167]. Nintendo, Sony y Microsoft son las empresas que lideran el mercado de las consolas de sobre mesa, mientras que solamente Nintendo y Sony tienen el liderazgo en las consolas portátiles [167][169][180][40]. Dado que estos dispositivos son en lo general plataformas privadas, cerradas y no compatibles entre sí, el desarrollo del software se ve condicionado y regulado también por estas mismas empresas. Un número pequeño de publicadores están autorizados para llevar los juegos a las diferentes consolas [169]. Nintendo se destaca en particular, por implementar dos barreras ante los desarrolladores, por un lado un chip destinado a comprobar la validez de los juegos, y por otro lado un sistema riguroso de control para la aprobación de los juegos [11][167][169].

El hardware se caracteriza por tener una renovación cíclica aproximadamente cada 5 años, lo que representa un esfuerzo económico tanto para los clientes como así también para los desarrolladores que deben adaptarse a la nueva plataforma [167][169]. No obstante en la actualidad la característica de retrocompatibilidad (la posibilidad de jugar juegos de la consola predecesora) es más común en las nuevas consolas ya que permiten que estas salgan al mercado con un cierto catálogo de juegos disponibles.

Las estrategias comerciales generalmente utilizadas por las empresas en el segmento de hardware implican reducir los costos de venta de las consolas, para luego recuperar el dinero a través del cobro de un porcentaje por cada juego publicado para la misma [167].

El sector de software por su parte se encuentra fragmentado principalmente en tres: videojuegos para PC (también llamados juegos de computadora o *computer games*), videojuegos para consolas (usualmente llamados videojuegos a secas), y los videojuegos para dispositivos portátiles [167][169]. En la figura 7.2.1 se puede observar la participación en ventas de cada uno de los sectores, siendo la más predominante la de las consolas [167], sin embargo, en Europa la participación de los juegos de PC es mucho más elevada [170].

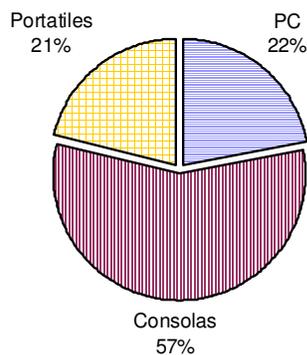


Figura 7.2.1 Ventas globales por plataforma.

En PC no existe un control tan estricto como en los juegos para consolas dentro de la plataforma dado que el hardware proviene de distintos fabricantes y los SO son de propósitos generales y no están enfocados en el consumo de videojuegos [167][169]. En general los juegos de PC son menos costosos de producir, y tiene equipos y tiempos de desarrollos ligeramente menores que los videojuegos de consolas [167].

No obstante, existe una tendencia creciente que los juegos AAA sean publicados por pocos *publishers* generando un oligopolio similar al existente en la industria de los juegos de consolas [169]. Uno de los más grandes es EA (Electronics Arts) la cual ha experimentado un crecimiento del 123% durante el año 2005 [169] Otros *publishers* importantes de la industria son Blizzard, Ubisoft, Konami, Take-Two, THQ, Infogrames y Sega [166].

7.3 Cadena de valor.

La cadena de valor de un videojuego coincide en esencia, con el de otras industrias culturales [167], ya que está basada en la creación, publicación y distribución de productos de propiedad intelectual (ver figura 7.3.1) [171].

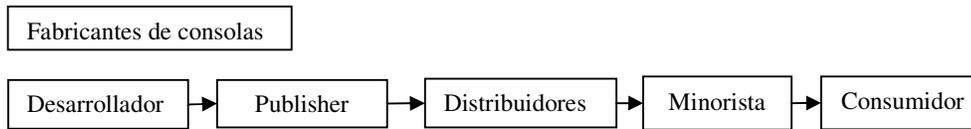


Figura 7.3.1 Cadena de valor

El equipo de desarrolladores interactúa directamente con los fabricantes de consolas, y con el *publisher*. Como fue definido en el capítulo 5.1, este tiene la tarea de desarrollar desde el punto de vista del diseño e implementación del juego y del videojuego en si mismo, en base a ideas y propiedad intelectual propia, o bien, como es la mayoría de los casos, en base a ideas y propiedad intelectual creada o controlada por el *publisher* [166].

Existen básicamente tres tipos de desarrolladoras [167]:

- Las llamadas *first party*: son parte de una empresa *publisher*, por lo que desarrollan los juegos a pedidos y voluntad de esta.
- Las *second party*: son empresas que realizan los juegos a partir de contratos y por pedido de *publisher*.
- Las *third party*: son empresas independientes de los *publisher*, que desarrollan sus propios juegos pero que luego intentan vender a un *publisher* para lograr concretar o bien distribuir el videojuego.

La mayor parte de los desarrollos de videojuegos provienen del primer y segundo tipo de desarrolladoras, lo que significa, que la mayor parte de los videojuegos son controlados o son propiedad de un *publisher* [166].

En el caso de *second* y *third party* el trato con el *publisher* se produce mediante un contrato en donde el *publisher* le otorga dinero a la empresa desarrolladora y luego a cambio, recibe un porcentaje por cada copia del juego [166].

El *publisher* por otra parte es el responsable de financiar los proyectos, de negociar con los distribuidores, de las campañas de marketing, de la manufactura, de las localizaciones (traducción y adaptación de los contenidos a otros idiomas y culturas), y en un numero de casos crecientes, negociar con los minoristas o *retailers* [171][166][169]. Es una tendencia en la industria de software de videojuegos la convergencia vertical del proceso de producción, y se ve reflejado tanto la integración de un equipo de desarrollo a los *publishers* como a la falta de necesidad de distribuidoras. A partir de esto, los *publishers* ganan experiencia en el desarrollo, propiedad sobre nuevas tecnologías, y en consecuencia ventajas competitivas frente a otros *publishers* al asegurarse el control de las tecnologías, videojuegos y propiedades intelectuales que su equipo desarrollador produce [169].

El *publisher* es, hoy en día, la parte más grande de la industria del software de los videojuegos, el papel del distribuidor en la mayoría de los casos es realizado por éste [169]. Del mismo modo que existen desarrolladoras como parte de *publishers*, los fabricantes de consolas como Sony, Microsoft, y Nintendo tienen sus propios *publishers* y empresas de desarrollo [170].

A continuación en la tabla 7.3.2, se muestra un listado de listado de *publishers*, ordenados conforme a su participación en el mercado [175].

Empresa	Independiente	Empresa	Independiente
1. Nintendo	No	11. Konami	Si
2. Electronic Arts	Si	12. Sega	Si
3. Activision / Blizzard	Si	13. Capcom	Si
4. Ubisoft	Si	14. MTV Games	Si
5. Take-Two	Si	15. Namco Bandai Games	Si
6. Sony Computer Entertainment	No	16. Warner Bros. Interactive	Si
7. Bethesda Softworks	Si	17. Disney Interactive	Si
8. THQ	Si	18. Atari	Si
9. Square Enix	Si	19. Atlus	Si
10. Microsoft	No	20. LucasArts	Si

Tabla 7.3.2 Top 20 publishers en el 2009.

Las distribuidoras por su parte tienen el papel de hacer llegar los juegos a los minoristas, y si bien el papel del distribuidor es muchas veces solapado por el del *publisher*, existen hoy muchas empresas distribuidoras alrededor del mundo [176].

La presencia de la distribución digital, el aumento en las velocidades de conexión a internet junto al crecimiento de la penetración de la banda ancha, y el creciente mercado de los juegos móviles hace necesario presentar un esquema alternativo a la cadena de valor tradicional [169][176]. Parte de la infraestructura creada para lograr la distribución de copias físicas se torna irrelevante [176]. En la figura 7.2.3 se representa una versión modificada de la cadena de valor tradicional.

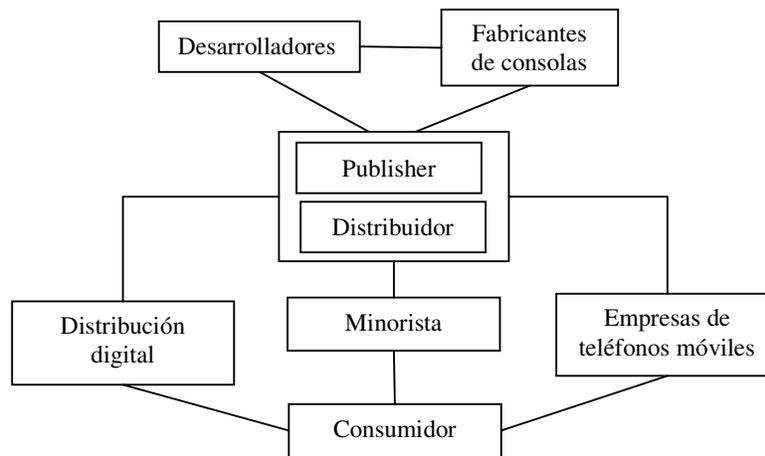


Figura 7.3.3 Cadena de valor con la inclusión de la distribución digital.

La distribución digital permite acortar la cadena de valor, y los precios los cuales el cliente debe pagar por los juegos. El uso de la nube en la distribución de videojuegos permite además alcanzar más fácil y rápidamente, con la necesidad de menos intermediarios, a clientes en distintos lugares del mundo [176]. Este tema será retomado en la sección 7.5.

7.4 El mercado.

Tal como fueron caracterizados en la sección 3.5, los jugadores de videojuegos hoy en día abarcan un espectro muy amplio de edades, y con diferencia cada vez menor entre los jugadores de distintos géneros. La cantidad y tipos de juegos y plataformas se diversificaron para cubrir este espectro amplio de consumidores, logrando capturar la atención de todos los sectores [48][170].

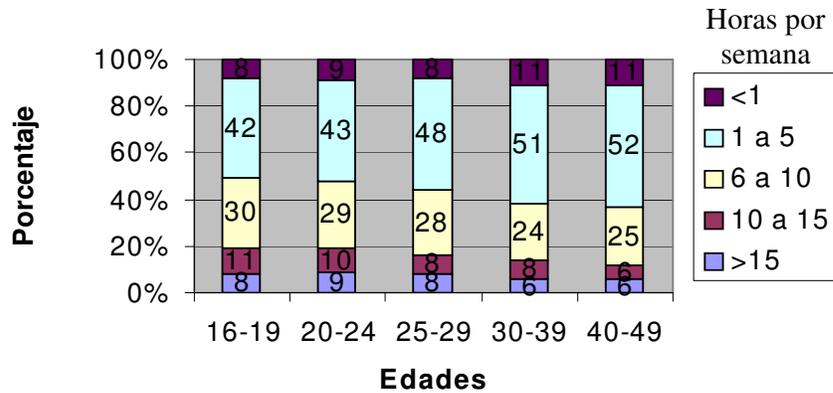


Figura 7.4.1 Horas de juego durante la semana según edades.

En la figura 7.4.1 se puede observar el tiempo que emplean las personas de distintas edades para jugar en una semana, demostrando que la industria pudo cubrir las necesidades de sus consumidores en distintas edades sin descuidar el sector tradicional, dado que la distribución de las horas de juego son muy similares entre sí [170].

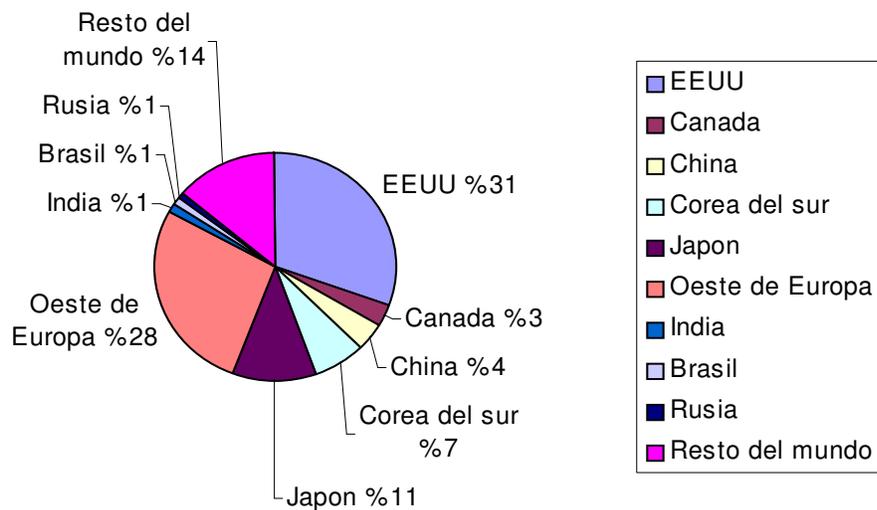


Figura 7.4.2 Distribución del mercado global de videojuegos.

En cuanto a distribución geográfica el mercado de videojuegos (Ver figura 7.4.2) es predominante la posición de EEUU en el mercado mundial, sin embargo se prevé para los próximos años un crecimiento rápido de China, junto al crecimiento en el sector de la India y Brasil [177].

En cuanto a los géneros de videojuegos y las ventas en EEUU según la ESA [51] en la figura 7.4.3 se muestra la participación que tienen cada uno.

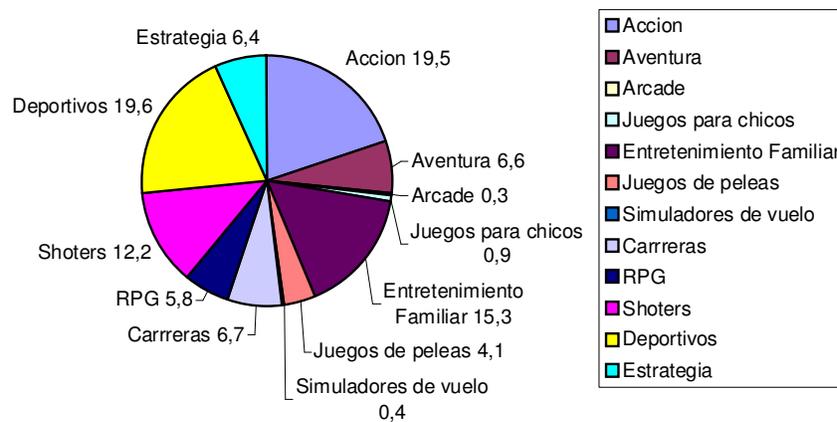


Figura 7.2.3 Ventas según géneros ESA 2010.

Los juegos de acción juntos con los de deportes son los más comprados en EEUU, seguido por los juegos de carácter de entretenimiento familiar.

7.5 Modelos de negocios.

Los modelos de negocios utilizados por la industria de software de videojuegos es variado y depende de diversos factores tales como el tipo de juego, el tamaño de la empresa, el público objetivo, la plataforma objetiva, etc. Dado que escapa al alcance del trabajo la caracterización de todos los modelos de negocios utilizados por la industria de videojuegos [178] se caracterizarán los más representativos.

7.5.1 Retail.

El modelo de negocios de retail (venta minorista), es el modelo tradicional utilizado desde el comienzo de la industria. Se venden copias del juego almacenado en un soporte físico como pueden ser cartuchos, CD, DVDs u otros medios de almacenamientos. El *publisher* es el que se encarga de realizar la manufactura y preparación de la parte física del producto y de contactar con el minorista o bien con la distribuidora para que llegue el producto al consumidor [166][178]. Cada eslabón de esta cadena agrega un costo adicional al juego acrecentando el costo final con el cual llega al consumidor (Ver figura 7.5.1.1) [167].

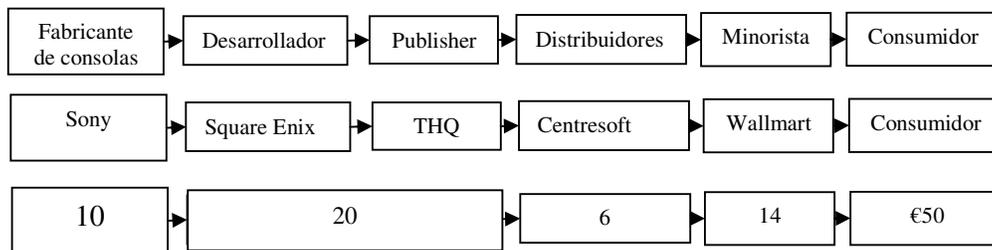


Figura 7.5.1.1 Costos agregados en el modelo de distribución Retail.

7.5.2 Distribución digital.

La distribución digital, es similar al modelo de negocios de retail con la diferencia de que no existe la distribución de un elemento físico para almacenar el juego. La venta y la distribución del juego se realizan a través de Internet y en forma mucho más directa [178][166]. El aumento en la velocidad de banda ancha y de la penetración de la misma, junto con el afianzamiento de sistemas de ventas online hacen que la distribución digital tome un papel cada vez más importante [169].

La plataforma de ventas y comunidad online más importante es Steam, que ofrece a los usuarios la posibilidad de comprar fácilmente en un catalogo de más de 1800 títulos, y mantenerse en contacto con una base de usuarios activa de 35 millones de usuarios. Steam se encuentra disponible en 21 idiomas distintos y en 237 países [29]. Origin de EA, y Desura son otras plataformas de distribución digital que están intentando entrar a este negocio.

También existen casos de juegos que se distribuyen directamente por Internet sin la necesidad de contar con una gran plataforma, y utilizando paquetes de servicios diseñados para el comercio electrónico [166].

En el sector consolas, las tres empresas líderes del mercado Nintendo, Sony y Microsoft, poseen sus respectivas tiendas online llamadas Nintendo WiiStore, Playstation Network (PSN), y Xbox Live Arcade (XBLA) respectivamente.

PSN por su parte, ofrece no solamente la venta de juegos para la PS3 sino también para la PSP.

7.5.3 Videojuegos en *Smartphones* y Dispositivos móviles.

Resulta necesario hacer una división entre los juegos para dispositivos móviles convencionales y los juegos para *smartphones* ya que los canales y la estructura es de cada uno es diferente.

En cuanto a *smartphones*, tal como se adelantó en el capítulo 6.1.3, Android y Apple son quienes dominan el mercado, y utilizan la distribución digital como único medio para la compra de videojuegos. El Apple Store, y el Android Market son las plataformas utilizadas por estos. El mercado de videojuegos en *smartphones* experimenta un crecimiento que amenaza a las consolas portátiles de Nintendo y Sony [179]. En la figura 7.5.3.1 se puede observar la distribución del mercado compartido por *smartphones* y las consolas portátiles, y el crecimiento de la participación de juegos en *smartphones*.

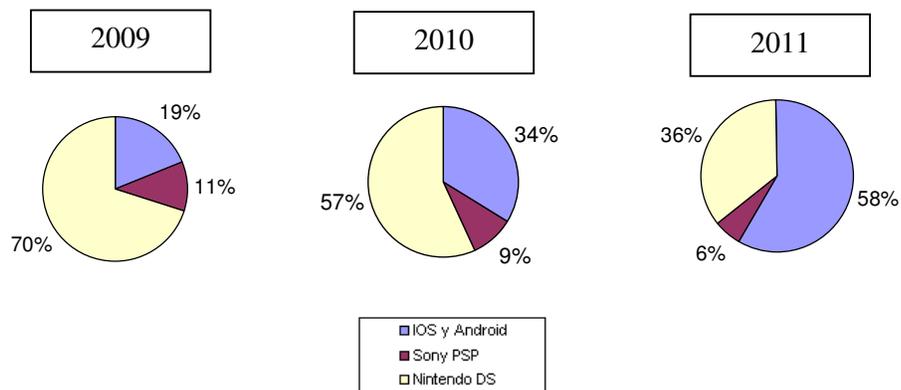


Figura 7.5.3.1 Distribución del mercado de videojuegos en dispositivos portátiles.

Tal como sucede en las tiendas online XBLA, WiiWare y PSN, los desarrolladores deben pasar por un proceso de aprobación de los propietarios de las plataformas, y pagar un porcentaje a estos por cada unidad vendida.

En cuanto a dispositivos móviles existen alternativas que utilizan las empresas que proveen el servicio de telefonía móvil. Por medio de un portal el usuario solicita contenido, y la empresa de telefonía le envía la solicitud de contenido y el pago al propietario del juego, mientras que al cliente, se lo cobra por medio de la factura mensual o bien el crédito que tiene disponible [166]. Existen diferentes variantes a este modelo, pero esta es una de las más populares.

7.3.4 Suscripción.

Este es un modelo de negocio utilizado comúnmente por juegos multijugadores online masivos (MMO). El cliente paga una cuota fija mensual para poder continuar jugando y acceder a las capacidades de multijugador [178]. El juego en si mismo, conocido como programa cliente (Ver sección 6.2.9), puede ser de distribución gratuita en formato digital o bien ser pago utilizando el modelo retail [178]. Los juegos MMO también utilizan frecuentemente lo que se conocen como micro-transacciones tratado en la sección siguiente.

Los juegos por suscripciones generaron en 2008 ganancias de mil millones de dólares, repartidos entre juegos MMO como **World of Warcraft**, **Runescape**, **Lord of the Rings Online**, **Final Fantasy XI**, y **City of Heroes**. Parte de estas ganancias corresponden también a suscripciones a sitios en los cuales, en lugar de enfocarse en un solo juego, proveen al usuario suscripto una amplia gama de juegos (generalmente casuales). Los portales más importantes de este estilo son Pogo.com, Realarcade.com, BigFishGames.com y Gametap.com [180].

7.5.5 Micro-transacciones.

Las micro-transacciones se utilizan generalmente en juegos gratuitos (conocidos como free-to-play) o de suscripción, donde existe contenido al que el jugador no puede acceder normalmente a menos que pague por él. Este contenido pueden ser ítems, niveles, vestuarios, personalizaciones, ventajas para jugar, o cualquier cosa que las

características del juego permitan. El usuario puede comprar mediante dinero real una moneda virtual que puede usar en el juego, o bien, comprar directamente los ítems o el contenido deseado [178]. De esta manera si bien una parte del juego se puede acceder sin pagar un adicional o bien completamente gratis, hay partes del juego a las que no se puede acceder sin pagar a cambio.

7.5.6 Sponsorships y publicidad.

Este modelo de negocios es usualmente utilizado por los juegos que pueden ser jugados directamente en el explorador web mediante la utilización de un plugin. Flash es el ejemplo de la tecnología predominante en este tipo de juegos [178][181]. También es utilizado en las aplicaciones distribuidas gratuitamente en *smartphones*.

Los juegos generalmente se distribuyen y llegan al usuario de manera gratuita y a través de internet. Las ganancias llegan al desarrollador gracias a la publicidad que los juegos llevan dentro de él [178] y generalmente se utilizan infraestructuras y herramientas ya establecidas como por ejemplo Mochimedia [30]. Los portales de juegos (sitios webs que agrupan juegos y usuarios) pueden pagarle a los desarrolladores para incluir sus logos y links para dirigir el tráfico hacia estos portales. A partir de las visitas los portales obtienen ganancias mostrando publicidades alrededor del juego [178][181]. Se estima que uno de cada tres usuarios de Internet visita estos portales de juegos, lo que representa una potencial audiencia de 237 millones de personas alrededor del mundo [30].

7.5.7 Advergaming.

El advergaming si bien, al igual que el modelo de negocios anterior, consisten en la distribución de juegos generalmente gratuitos, y solventados por medio de publicidad, se diferencian de los primeros en que, los juegos para advergaming son hechos a pedido y a medida de la empresa que quiere utilizar el videojuego para publicitarse [40][178]. Esta es una de las actividades más frecuentes de la industria nacional, y será tratado en la sección siguiente. La empresa desarrolladora obtiene ganancias directamente por medio de un contrato con la empresa a la que hace publicidad, y en general es esta última quien se encarga de la distribución.

7.5.8 Juegos sociales

Si bien los modelos de negocios utilizados por los juegos sociales se encuentran descriptas básicamente con los modelos de negocios basados en publicidad y en micro transacciones (el 90% de los ingresos provienen de este modelo [190]), los juegos sociales poseen características y un mercado particular que hace necesario su tratamiento por separado.

Una característica muy importante en los juegos sociales es la viralidad la cual es una característica intrínseca de los medios sociales [191]. En juegos sociales, se refiere a la capacidad que tiene el juego de crecer en cantidad de usuarios gracias a que sus propios jugadores comparten contenido e información sobre juego con sus contactos y allegados, que luego juegan y vuelven a integrar este ciclo alcanzando tasas de crecimientos exponenciales. La empresa Playfish dedicada a juegos sociales, explica que el 95% de sus jugadores se deben gracias al uso de estrategias aplicadas a la viralidad [189][190]. Algunas de las estrategias utilizadas en estos juegos, es la entrega de bonificaciones, ítems, o insignias, a cambio de que el usuario comparta el información del juego con sus amigos; otra estrategia utilizada consiste en publicar en la información personal del usuario información sobre los avances que este produce en el juego [189].

La característica conocida como *sticky*, se refiere a la capacidad que tiene el juego de retener a sus jugadores. Cuanto más *sticky* sea un juego, más tiempo por semana o incluso al día, pasa el usuario en el juego. Un índice elevado de *sticky* por lo tanto, aumenta las probabilidades de “conversión”. La conversión por su parte, se refiere a la probabilidad que tiene un jugador de gastar dinero en el juego. Si bien las tasas de conversión en juego sociales son menores (de 1% a 6%) a las que resultan en juegos como los MMORPG (de 5% a 10%) se espera que este porcentaje aumente a partir de la integración una mejor integración de la monetización en las redes sociales [190].

8. Desarrollo de videojuegos en argentina.

El desarrollo de videojuegos es una actividad relativamente nueva en nuestro país (cerca de 11 años de antigüedad) y es desarrollada aproximadamente por un total de 65 empresas que emplea alrededor de 2000 personas. La industria nacional esta enfocada en el mercado externo y comenzó a crecer profesionalmente a partir del año 2000, donde la diferencia cambiara abarataba los costos de la mano de obra. Capitales extranjeros y empresas multinacionales, junto con el impulso de jóvenes emprendedores, fueron los que forjaron el inicio de la industria nacional [182][183][184][185].

En cuanto al capital de origen de las empresas argentinas, es predominante el impulso nacional siendo un 83% solamente de Argentina, un 13% de capital mixto, y solamente un 3% de capital extranjero. No obstante, el sector de videojuegos de Argentina recibió interés internacional por parte de distintas firmas extranjeras, en 2009 la empresa nacional Sabarasa fue parcialmente adquirida por capitales mexicanos, Playdom (luego adquirida por Disney) en 2010 adquirió en su totalidad a la empresa Three Mellons y por último la empresa argentina Vértigo, a partir del 2010, recibe una inyección de dinero del fondo de inversión *Modena Technologies Capital Partners* [182].

En la actualidad los costos de producción crecieron y ya no se cuenta con la ventaja comparativa de la devaluación, lo que impulsa a un cambio en el enfoque de la mayoría de las empresas nacionales [183].

Es frecuente que las empresas que desarrollan videojuegos desarrollen al mismo tiempo diferentes tipos de negocios como actividades secundarias. El 40% de las empresas argentinas desarrollan otras actividades complementarias que incluyen servicios de diseño y animación, de programación y diseño web, y de publicidad. En el año 2008 el 43% de la facturación total de las empresas provino de estas actividades complementarias [182].

8.1 Características de la industria nacional.

Las empresas que forman el sector nacional de videojuegos tienen un carácter asimétrico. Conviven empresas de características diversas como los son empresas

unipersonales así también como empresas que emplean a cientos de personas. No obstante, la industria nacional esta mayormente formada por pequeñas y medianas empresas (PYMES). Aproximadamente el 86% de las empresas tienen menos de 20 empleados [168][182].

La industria nacional esta orientada a satisfacer necesidades del mercado externo con un fuerte perfil exportador. En el 2008 el 76% de la facturación de las empresas nacionales provino de ventas al mercado extranjero. Los principales países de exportación son EEUU y Canadá (un 76% de las empresas exportan a estos países), Europa (un 73%) y en menor medida a America Latina y el Caribe (40%) [182]. Por esta razón, la forma en que se desenvuelva la crisis económica de Europa, y EEUU, condicionará el crecimiento de la industria nacional en los próximos años [183].

Al estar sujetas a las demandas extranjeras, las empresas nacionales se caracterizan también por una alta tasa de natalidad y mortandad [182].

El mercado interno no es foco de atención de las empresas Argentinas debido a su reducido tamaño. Según la opinión de las empresas argentinas, un 75% lo consideraron como uno de los problemas más graves para desarrollar productos pensando en Argentina [183].

Las plataformas de desarrollo más recurrentes en las empresas son: PC y Mac presentes en el 71% de las empresas, el desarrollo de videojuegos web con un 46%, un 33% para desarrollo para juegos en dispositivos móviles, y en mucha menor medida los desarrollos para consolas: 17% consolas de sobre mesa, y 8% consolas portátiles. Los costos de producción, y las dificultades de negociación con las empresas propietarias de las consolas son algunas de las principales razones por las cuales pocas empresas encaran estos desarrollos.

En cuanto a la finalidad y temática de los videojuegos locales 85% de las empresas crean videojuegos para entretenimiento, un 37% para Advergaming, y un 33% para educación y capacitación [182].

8.2 El empleo y los profesionales.

El empleo por su parte, discriminando por roles profesionales, esta conformado mayoritariamente por un 26% de programadores, un 14% por testers y un 13% por

artísticas gráficos (Ver figura 9.2.1). Sin embargo, el porcentaje elevado de personas dedicadas al testing se explica con las tareas que realiza una empresa en particular de manera tercerizada para empresas extranjeras [182]. En general, los servicios de música y sonido son provistos por terceros en el 64% de las empresas tal como es característico de la industria de los videojuegos en general [93][182].

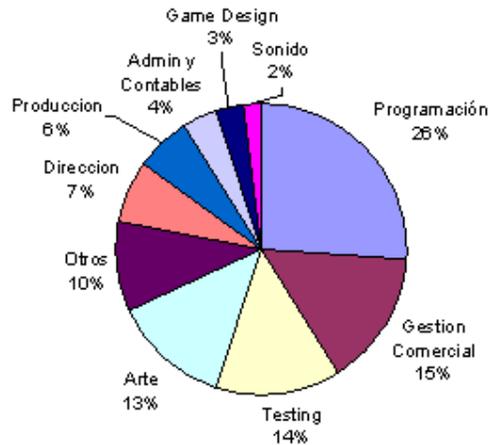


Figura 8.2.1 Distribución del empleo.

El empleo, al igual que en la industria del software [132], se caracteriza también por poseer un alto nivel de rotación entre diferentes empresas [182].

En cuanto a la formación académica del sector, la mayor parte corresponde al área de programación, mientras que las actividades relacionadas con la producción de arte (tales como Diseñador de imagen y sonido, Diseñador Gráfico y Artista 3D) ocupan un 46% del total [182]. La capacitación específica en videojuegos es mínima (solo un 6.5%), y la baja disponibilidad de recursos humanos calificados es considerada por el 87% de las empresas como una problemática importante.



Figura 8.2.2 Formación académica del sector.

La formación académica específica de videojuegos es nueva también en nuestro país y se encuentra poco extendida. Esta cubierta por institutos privados como Image Campus, La escuela Da Vinci, y la Universidad Nacional del Litoral (UNL), en el marco de carreras terciarias, y en cuanto a carreras de nivel universitario el Instituto Tecnológico de Buenos Aires (ITBA) ofrece una diplomatura en Diseño y Desarrollo de Videojuegos, y la formación de postgrado específica en la materia [182].

En el año 2010 se desarrolló el primer Workshop Argentino sobre Videojuegos (bajo el nombre de WAVI), el cual brindó un marco para que investigadores de temas afines al desarrollo del entretenimiento digital y los videojuegos divulguen sus logros en el área y con el objetivo de reducir la brecha que existe entre la universidad y la industria de videojuegos [186].

8.2 Organizaciones y eventos.

La Asociación Argentina de Desarrolladores de Videojuegos de Argentina (ADVA), es una organización sin fines de lucro creada en el año 2000 con el objetivo de promover y fomentar el desarrollo de videojuegos en la Argentina [187].

La financiación de esta asociación legal se consigue principalmente a través del cobro de cuotas de las empresas afiliadas a ella, las cuales a cambio reciben diversos beneficios y descuentos [188].

Las principales tareas desarrolladas por ADVA son:

- Mantener una comunidad desarrollada a través de un foro público y una lista de correo.
- Ofrecer facilidades de negocio a las empresas que la conforman tanto local como internacionalmente.
- Tener actuación en el marco gubernamental de manera activa en la formación de leyes y en planificación estratégica del software.
- Desarrollar anualmente la Exposición Argentina de Videojuegos (EVA).

La EVA es el evento más importante para los desarrolladores locales, y organizado por ADVA, participan tanto disertantes locales como internacionales. Esta exposición conforma un espacio de encuentro, diálogo y capacitación para personas afines al desarrollo de videojuegos. Año a año la exposición crece en cantidad de visitantes, sponsors y charlas. La última exposición al momento de la escritura del trabajo fue la 9ª edición de la misma en noviembre de 2011. Durante la misma se ofrecieron workshops, mesas redondas, conferencias, talleres, rondas de negocios, charlas y un expofloor donde las empresas locales tienen la oportunidad de mostrarse [186][187].

En cuanto a concursos el más importante a nivel nacional es CODEAR el cual consiste en una restricción (generalmente tecnológica) en la cual se invita a los desarrolladores locales e internacionales a participar realizando un videojuego en aproximadamente un mes. Durante la EVA se hace entrega de las menciones de primer, segundo y tercer puesto. El CODEAR con mayor record de concursantes fue el codear con temática "5 elementos" donde participaron un total de 39 juegos. [186]

El concurso COREAR por su lado, es un concurso dedicado a los músicos de la industria los cuales pueden participar en dos categorías: remix donde deben remixar un tema de algún videojuego en el primer caso; y la categoría original donde deben crear una música original para reemplazar la de un videojuego existente [186].

9. Consideraciones finales.

A lo largo de este trabajo se ha caracterizado al desarrollo de videojuegos como una amplia área de estudio que combina disciplinas y problemáticas de múltiples naturalezas. El trabajo intentó, sin perder de vista la extensión del mismo, dar la mayor profundidad posible en cada uno de estos temas para permitir comprender y abordar la actividad del desarrollo de videojuegos con las herramientas necesarias para comprenderla.

En cuanto al diseño de los videojuegos, es necesario entender las necesidades y la naturaleza humana, las propiedades del juego mismo, las características de la audiencia objetivo, junto con la de las plataformas y sus limitaciones. También es importante tener en consideración los significados y mensajes que los videojuegos son capaces de transmitir a la sociedad que los consume.

La audiencia, por otro lado, se ha diversificado al mismo tiempo que lo han hecho los diferentes géneros y tipos de videojuegos. La caracterización popular del videojuego como objeto de uso de los niños es desafiado por las estadísticas que demuestran que una generación de personas que han jugado videojuegos sigue haciéndolo. Aún más allá, la penetración de los videojuegos en la mayoría de los dispositivos multimedia, principalmente en celulares y smartphones, los juegos sociales y los juegos casuales (en web y en consolas) han alcanzado distintos grupos demográficos que señalan que los videojuegos se han masificado.

La capacidad de los videojuegos de transmitir ideas y mensajes ha sido estudiada en diversas aplicaciones. La utilización de estos para la reclutación de tropas, para la educación y el estímulo de esta, para el cuidado de la salud y el tratamiento de enfermedades, son algunos de estos ejemplos. Sin embargo, este tipo de estudios se encuentra todavía en una edad temprana de maduración, por lo que resulta un área de interés para estudios futuros. Los juegos artísticos, por otro lado, son una prueba más de la capacidad que tienen de los videojuegos para transmitir mensajes, y en la actualidad han sido reconocidos como arte por distintos organismos con autoridad en la materia.

Por otro lado, en cuanto el proceso de desarrollo de videojuegos, las metodologías utilizadas varían principalmente dependiendo del tamaño del equipo y del proyecto. Las metodologías iterativas, incrementales y las metodologías ágiles son algunas de

las más utilizadas. El cumplimiento del presupuesto, del tiempo de desarrollo, o de la calidad objetivo del juego, es una tarea difícil de lograr en simultáneo y que no siempre garantiza el éxito del proyecto en términos comerciales. Esta incertidumbre que puede ser explicada por la naturaleza de los videojuegos de ser productos culturales.

En cuanto a la tecnología, el aspecto de software en videojuegos se enfoca en la creación de mundos simulados haciendo un uso eficiente del hardware sobre el cual corre para lograr que los tiempos de respuesta sean mínimos y conseguir así una sensación de juego fluido. Para agilizar, acelerar y abaratar los costos de producción se han desarrollado herramientas con el concepto de reutilización de componentes de software que cubren las diversas partes de un videojuego: APIs gráficas, APIs de sonidos, engines de IA, engines de físicas y game engines, son algunos de estos ejemplos. La creciente variedad de plataformas con un poder de procesamiento cada vez más grande (PC, Mac, Consolas, Smartphones) hizo necesaria también la creación de herramientas con capacidades de generar aplicaciones para diversas plataformas.

La mayor parte de la industria de los videojuegos se ha orientado en conseguir representaciones visuales cada vez más realistas, por lo que los esfuerzos del desarrollo de tecnologías han estado orientados en el campo de la computación gráfica. Áreas como el sonido y la IA entre otras, no han sido exploradas con profundidad por lo que representan áreas de interés para su estudio futuro.

El hardware, por su parte, ha evolucionado en la búsqueda de una mayor capacidad de procesamiento, de almacenamiento y de conectividad. No obstante, la introducción de hardware innovador en materia de controles abre una nueva rama de posibilidades no solo para distintos tipos de videojuegos sino también para su utilización en interfaces de usuarios en otras aplicaciones. Kinetic, en particular, posee un gran potencial para el desarrollo de aplicaciones de visión artificial por lo que también representa un potencial objeto de estudio.

La industria de videojuegos, considerada como parte de las industrias culturales, es caracterizada principalmente por estar impulsada por hits, y estar controlada por grandes empresas tanto en los sectores de hardware como en el sector de publishers. No obstante, la diversificación del mercado, y el tamaño y variedad de este, permite el desarrollo de nuevas empresas dedicadas a diferentes sectores. Los videojuegos para móviles y los juegos sociales, representan hoy en día oportunidades de negocios

debido a que tienen un mercado caracterizado por un rápido crecimiento y los presupuestos y equipos de desarrollo necesarios para este tipos de proyectos son mucho menores en comparación, por ejemplo, con la industria AAA.

La distribución digital comienza a tomar importancia en la industria de los videojuegos tanto para pequeñas como a grandes empresas. La introducción de EA (uno de los publishers más grandes de la industria) a la distribución digital con su portal Origin, junto con las tiendas virtuales de los diferentes fabricantes de consolas, y las tiendas virtuales de los smartphones, refuerza y pronostica que este tipo de distribución sin lugar a dudas seguirá creciendo y marcará el futuro desarrollo de la industria.

Por otro lado, el desarrollo de videojuegos en Argentina es una actividad joven que nació y se fortaleció en la oferta de servicios al extranjero y de productos al mercado externo, ayudada por la diferencia cambiaria. Hoy en día los costos de producción en Argentina crecieron a la par al que los costos en otros países subdesarrollados bajaron. Las empresas argentinas hacen hincapié en que será necesario demostrarle al mundo que la Argentina es capaz de producir productos que compiten no solamente en costos sino también en calidad. La oferta académica de formación específica de videojuegos es también reciente y se encuentra poco extendida, la ausencia de personal capacitado, junto con las necesidades de inversión, son algunas de las problemáticas actuales con las que se enfrentan las empresas argentinas.

Como conclusión final del trabajo, el desarrollo de videojuegos es un área compleja que requiere análisis desde diversos puntos de vista. La diversificación de la industria, la masificación de la audiencia y el crecimiento en el interés por la exploración de la capacidad de los videojuegos para transmitir ideas, denotan que este medio de entretenimiento y expresión seguirá creciendo no solamente en términos económicos sino también de importancia cultural y en la forma en que aprendemos y nos comunicamos.

“El hombre no deja de jugar porque se hace viejo, sino que se hace viejo porque deja de jugar.” - Bernard Shaw

10. Glosario.

AAA	Un denominación utilizada para caracterizar a aquellos juegos que cuentan con un gran presupuesto y un nivel de calidad alto.
Advergames	Son juegos creados a pedido con el objetivo de mostrar publicidad. Generalmente distribuidos en forma gratuita.
Assets	Son los recursos de arte que utiliza un videojuegos como texturas, imágenes, sonidos, música, etc.
Bitmap	Un bitmap consiste en una imagen formada por una matriz de pixeles.
Blit	Es una operación por medio de la cual una matriz de imagen es copiada a otra.
Blitting	La acción de realizar un blit.
Buffer	Es una memoria de almacenamiento temporal utilizada para usar dos procesos que suceden a velocidades o en momentos diferentes.
Canal alfa	Es un valor número que caracteriza la transparencia con que se ve un píxel.
Cluster	Es un conjunto de computadoras conectadas entre sí con el objetivo de procesar información como si fuera una sola.
Drag and drop	Se refiere a una interfaz de usuario en donde se manejan ítems mediante la acción de arrastrar y soltar.
Driver	Es un programa con el objetivo de controlar y hacer posible la comunicación con un elemento de hardware y el sistema operativo
Estereoscópico	Es un efecto mediante el cual las imágenes planas parecen tener relieve.
Eventos scripteados	Son eventos determinados por el diseñador de niveles que suceden cuando se cumple determinada condición en el juego.
Feedback	Retroalimentación.
Firmware	Son programas almacenados en dispositivos electrónicos que tiene el objetivo de controlar y proveer algunas de las funciones de este.
Frames	Cuadros.
Game design	Diseño del videojuego.
Game designer	Diseñador de videojuegos.
Game loop	Ciclo de programación en el cual se procesa el videojuego, las entradas y las salidas.
Game tester	Persona que tiene la tarea de probar un videojuegos conforme a un plan de pruebas.
Gameplay	Se refiere a la forma en que es jugado un juego.
Glitch	Es artefacto gráfico que se produce por un error en el programa o bien por una anomalía en el hardware.

Pipeline	Proceso complejo.
Port	Una aplicación o videojuego implementando en una plataforma diferente a la original.
Procedural	Generación de assets a partir de algoritmos.
Publisher	Es una empresa dedicada a gestionar el desarrollo de proyectos de videojuegos y de llevarlos al mercado.
Puzzle	Es desafío intelectual.
Renderizar	Transformar una representación lógica a una imagen mediante procesamiento gráfico.
Screenshot	Es una imagen capturada de la pantalla de juego.

11. Abreviaturas.

AABB	Caja de colisión alineada a los ejes.
ADC	Conversor analógico a digital.
ADD	Documento de Diseño de Arte.
ADVA	Asociación de desarrolladores de videojuegos Argentina.
AGP	Interfaz de hardware para placas aceleradora de video.
AI	Inteligencia Artificial.
API	Interfaz de programación de aplicación.
BV	Volumen de borde.
CODEAR	Concurso nacional de videojuegos.
COREAR	Concurso nacional de música para videojuegos.
CPU	Unidad de procesamiento.
DCA	Conversor digital a analógico.
DSP	Procesamiento digital de señales.
ESA	Asociación del software de entretenimiento.
ESRB	Entertainment Software Rating Board
EVA	Exposición de videojuegos Argentina.
FPS	Cuadros por Segundo/ First Person shooter.
FSM	Máquinas de estados finitos.
GDD	Documento de diseño de juegos.
GPU	Unidad de procesamiento gráfico.
GTA	Grand Thief Auto
GUI	Interfaz gráfica de usuario.
HLSL	Lenguaje de programación de shaders.
IAB	Internet Architecture Board
IP	Protocolo de Internet.
IPV6	Protocolo de Internet versión
ITBA	Instituto Tecnológico de Buenos Aires.
J2ME	Java 2 micro edición.
LAN	Red de area local.
LGPL	Licencia de software gratuito permisiva.
LOD	Determinación del nivel de detalle.
MIT	Instituto Tecnológico de Massachusetts
MMO	Multiplayer masivo online.
MMORPG	RPG multiplayer masivo online.
NDS	Nintendo DS
NEA	El Fondo nacional de Artes de Estados Unidos
NES	Nintendo Entertainment System.
NPC	Personaje no controlado por el jugador.
OSS	Open Sound Systems
P2P	"Peer to peer" o "par a par"
PC	Computadora personal.
PEGI	Pan European Game Information
PS1,PS2,	
PS3	Playstation 1, Playstation 2, Playstation 3
PSP	Playstation portátil.
PYMES	Pequeñas y medianas empresas..
QA	Control de Calidad.
RAM	Memoria de acceso aleatoria.
ROM	Memoria de solo lectura.

Desarrollo de Videojuegos.
Página 179 de 193

RPG	Role playing game, o juegos de rol.
RTM	Listo para manufactura.
RTS	Juegos de estrategia en tiempo real
SDL	Simple direct media layer.
SNES	Super Nintendo Entertainment System.
SO	Sistema operativo.
TCP	Protocolo de transmisión de datos controlado
TDAH	Trastorno por déficit de atención con hiperactividad
UDP	Protocolo de datagramas de usuario
UNL	Universidad Nacional del Litoral
WAD	Formato utilizado por el juego Doom para almacenar lógica del juego y assests
WAN	Red de area amplia.
WAVI	Workshops argentinos de videojuegos.
XBLA	Xbox live arcade.

12. Bibliografía.

1. **"Theory of Fun on Game Design"** Ralph Koster, Paraglyph Press, ISBN: 1-932111-97-2
2. **"The video game Theory Reader 2"**, Bernard Perron y Mark J. P. Wolf, Routledge Taylor & Francis Group, ISBN 0-203-88766-2
3. **"Art of Game design"** Chris Crawford, McGraw Hill, ISBN 0-88134-117-7
4. **"The Game Design Reader, A rules of play Anthology"** ISBN 0-262-19536-4
5. <http://bibliotecaignoria.blogspot.com/2009/06/hohan-huizinga-homo-ludens-cap-i.html> Mayo 2011.
6. [http://en.wikipedia.org/wiki/Jesper_Juul_\(game_researcher\)](http://en.wikipedia.org/wiki/Jesper_Juul_(game_researcher)) Mayo 2011
7. <http://www.half-real.net/dictionary/> Mayo 2011
8. http://en.wikipedia.org/wiki/Johan_Huizinga Mayo 2011
9. **"Andrew Rollings and Ernest Adams on Game Design"** Andrew Rollings y Ernest Adams, New Riders Publishing, ISBN: 1-592-73001-9
10. <http://www.ign.com/all-game-platforms.html> Mayo 2011
11. **"The video Game Explosion."** Mark J.P. Wolf, Greenwood Press, ISBN: 978-0-313-33868-7
12. <http://classicgames.about.com/od/classicvideogames101/p/CathodeDevice.htm>
Mayo 2011
13. <http://www.cs.clemson.edu/~mark/edsac.html> Mayo 2011
14. **"The ultimate History of Videogames"** Steven L. Kent, Three Rivers Press, ISBN: 0-7615-3643-4

15. <http://www.ign.com/top-25-consoles/25.html> Mayo 2011
16. <http://www.ign.com/top-25-consoles/2.html> Mayo 2011
17. <http://www.computernostalgia.net/articles/HistoryofComputerGames.htm> Mayo 2011
18. <http://www.nintendo.com/corp/history.jsp> Mayo 2011
19. <http://www.ign.com/top-25-consoles/1.html> Mayo 2011
20. <http://www.pokemon.com/ar/> Mayo 2011
21. http://3dfx-interactive.co.tv/#Early_history Mayo 2011
22. http://www.scei.co.jp/corporate/data/bizdataps_e.html Mayo 2011
23. <http://www.whatconsole.co.uk/psone.php> Mayo 2011
24. http://www.maximumpc.com/article/features/graphics_extravaganza_ultimate_gpu_retrospective?page=0,1 Mayo 2011
25. <http://www.scei.co.jp/corporate/release/pdf/051130e.pdf> Mayo 2011
26. http://www.scei.co.jp/corporate/release/110214_e.html Mayo 2011
27. <http://www.xbox.com/en-GB/kinect> Mayo 2011
28. <http://us.playstation.com/ps3/playstation-move/> Mayo 2011
29. <http://www.valvesoftware.com/company/index.html> Mayo 2011
30. http://www.mochimedia.com/aboutus/release/MochiAds_Launch_10_16_07.pdf
Mayo 2011
31. “**Casual games and gender**”, Ulrich Tausend, 2006, Ludwig Maximilian University of Munich, Institute of Sociology.

32. <http://www.nintendo.com/3ds/es/> Mayo 2011
33. http://aii.lgrace.com/documents/Game_types_and_genres.pdf **“Game Type and Game Genre”** Lindsay Grace, 2005.
35. **“Game Design Foundations”** Roger E. Pedersen, Wordware Publishing, ISBN: 1-55622-973-9.
36. **“Video Games Industry Overview, An Analysis of the Current Market and Future Growth Trends”**, Aoife Cunningham Holger Langlotz, Marc Rhode, Clayton Whaley.
37. <http://www.lostgarden.com/2010/12/happy-2011-celebrating-frontiers-in.html> Mayo 2011
38. “Game Design” Bob Bates, Thomson Course Technology PTR. ISBN: 1-59200-493-8
39. http://www.gamasutra.com/view/feature/4074/the_designers_notebook_sorting_ph_p Mayo 2011
40. **“Developing Serious Games”**, Bryan Bergeron, Cengage Learning, ISBN: 1-58450-444-7
41. **“Serious games: online games for learning”** Anne Derryberry, Adobe Systems Incorporated 2007
42. **“Serious Games in Defence Education”**, Caspian Learning Engaging Minds copyright 2008.
43. **“Serious games Advergaming, edugaming, training and more”** Julian Alvarez, IDATE Consulting and research ISBN 978-2-84822-169-4
44. <http://www.tizapapelbyte.org/informes-2009/> Junio 2011
45. <http://www.newsgaming.com/games/index12.htm> Junio 2011

46. <http://www.nielsen.com/us/en/insights/reports-downloads/2009/Insights-on-Casual-Games-Analysis-of-Casual-Games-for-the-PC.html> **“Insights on Casual Games Analysis of Casual Games for the PC”** Agosto 2009 – Junio 2011.
47. **“Casual games and gender”** Ludwig Maximilian y Ulrich Tausend, Universidad de Munich, 2006.
48. **“Casual Games White Paper”** 2006 IGDA Casual Games SIG
49. **“A CASUAL REVOLUTION: Reinventing Video Games and Their Players”** Jesper Juul, The MIT Press Cambridge, Massachusetts London, England ISBN: 978-0-262-01337-6
50. **“Casual Games White Paper”** 2008-2009 IGDA Casual Games SIG
51. **“Essencial Facts about the computer and videogames Sales, Demographic and usage data”**. <http://www.theesa.com/facts/> Junio 2011
52. <http://www.pegi.info/es/index/id/89> Junio 2011
53. <http://www.esrb.org/ratings/faq.jsp#1> Junio 2011
54. <http://www.censorship.govt.nz/> Junio 2011
55. http://www.gamasutra.com/php-bin/news_index.php?story=8213 Junio 2011
56. <http://www.usk.de/> Junio 2011
57. **“Growing Up Online: Young People and Digital Tecnologies.”** Sandra Weber y Shandy Nixon. Palgrave MacMillan ISBN-13: 978-1-4039-7814-1.
58. **“The ethics of computer games”** Miguel Sicart. MIT. ISBN 978-0-262-01265-2
59. **“Playing with Videogames”** James Newman Routledge ISBN 0-203-89261-5
60. **“Videogames”** Gale, Cengage Learning ISBN-13: 978-1-4205-0170-4

61. **"Videogames and Art"** Andy Clarke y Grethe Mitchell. Intellect Books. ISBN 978-1-84150-142-0
62. <http://www.pbs.org/kcts/videogamerevolution/impact/myths.html> Junio 2010
63. [http://en.wikipedia.org/wiki/Grand_Theft_Auto_\(series\)#Sales](http://en.wikipedia.org/wiki/Grand_Theft_Auto_(series)#Sales) Junio 2011
64. **"Parents, children & Media: Kaiser family foundation survey June 2007"**
Victoria Rideout, M.a. Kaiser family foundation.
65. <http://www.kff.org/about/index2.cfm> Junio 2011
66. <http://www.theesa.com/facts/violence.asp> Junio 2011
67. **"The effects of violent video game habits on adolescent hostility, aggressive behaviors, and school performance"** Douglas A. Gentile, Paul J. Lynchb, Jennifer Ruh Linder, David A. Walsh, Academic Press Journal of Adolescence 27 (2004).
68. <http://www.theesa.com/gamesindailylife/health.pdf> Junio 2011
69. **"Video Games & Your Kids: How Parents Stay in Control"** Hilarie Crash y Kim McDaniel . Issues Press ISBN: 9781930461055
70. <http://www.raisesmartkid.com/3-to-6-years-old/4-articles/34-the-good-and-bad-effects-of-video-games> Junio 2011
71. **"Game Addiction"** Neils Clark y P. Shavaun Scott. McFarland & Company ISBN 978-0-7864-4364-2
72. <http://www.olganon.org/> Junio 2011.
73. <http://www.wired.com/gaming/gamingreviews/news/2002/04/51490> Junio 2011
74. http://es.wikipedia.org/wiki/Trastorno_por_d%C3%A9ficit_de_atenci%C3%B3n_con_hiperactividad Junio 2011
75. <http://archsurg.ama-assn.org/cgi/content/short/142/2/181> Junio 2011

76. <http://www.theesa.com/gamesindailylife/health.asp> Junio 2011.
77. <http://www.behavioralandbrainfunctions.com/content/6/1/9> Junio 2011.
78. <http://wiifit.com/> Junio 2011
79. <http://es.wikipedia.org/wiki/Cultura> Junio 2011
80. **“Rules of Play: Game Design Fundamentals”** Katie Salen y Eric Zimmerman,
The MIT Press, ISBN:0262240459
81. http://es.wikipedia.org/wiki/Baraja_espa%C3%B1ola junio 2011
82. <http://www.muyminteresante.es/will-wright-sims> Junio 2011.
83. http://www.indiegames.com/2011/02/browser_game_pick_american_dre.html
Junio 2011
84. <http://edition.cnn.com/2010/WORLD/europe/11/08/spain.super.mario.subdivision/index.html?hpt=T2> Junio 2011
85. <http://youbentmywookie.com/wtf/a-gallery-of-awesome-old-school-8-bit-video-game-graffiti-12809> Junio 2011.
86. <http://hyperallergic.com/8176/theater-of-the-arcade/> Junio 2011
87. <http://iamstillwater.wordpress.com/2011/05/08/national-endowment-for-the-arts-recognizes-video-games-as-art/> Junio 2011
88. <http://www.nea.gov/grants/apply/AIM/grant-program-description.html> Junio 2011
89. <http://www.objeto-a.com.ar/blog> Junio 2011
90. <http://www.americanart.si.edu/exhibitions/archive/2012/games/> Junio 2011
91. <http://www.gamescenes.org/> Junio 2011
92. http://en.wikipedia.org/wiki/Independent_video_game_development Julio 2011

93. **"Game Development and Production"** Erik Bethke, Wordware Publishing, Inc.
ISBN 1-55622-951-8
94. **"Video Games in the 21 century 2010 report."** Stephen Siwek, ESA
95. **"Producing Games: From Business and Budgets to Creativity and Design"**
D.S. Cohen, Sergio A. Bustamante II, Focal Press ISBN : 978-0-240-81070-6
96. **"Agile game development with Scrum"** Clinton Keith, Addison-Wisley ISBN-10:
0321618521 ISBN-13: 978-0321618528
97. <http://www.intel.com/technology/mooreslaw/index.htm> Julio 2011
98. **"Game Architecture and Design"** Andrew Rollings y Dave Morris, New Riders,
ISBN: 0-7357-1363-4
99. **"The game producer HandBook"** Dan Irish, Thompson Course Tecnology. ISBN:
1-59200-617-5
100. <http://www.gameproducer.net/2006/05/26/what-are-aaa-titles/> Julio 2011
101. **"Secrets of the Game Business"** François Dominic Laramée, Charles Rives
Media. ISBN: 1-58450-282-7
102. <http://electronics.howstuffworks.com/wii1.htm> Julio 2011
103. http://www.nytimes.com/packages/khtml/2006/12/21/business/20061221_HOWW_GRAPHIC.html
104. <http://www.nintendowii.es/showthread.php?t=2171> Julio 2011
105. <http://gravity.phy.umassd.edu/ps3.html> Julio 2011
106. <http://www.kinecthacks.com/> Agosto 2011
107. <http://jonpeddie.com/publications/add-in-board-report/> Diciembre 2011

108. **“Real-Time Shader Programming”** Ron Fosner, Morgan Kaufman Publishers
ISBN: 1-55860-853-2
109. <http://support.apple.com/specs/#iphone> Agosto 2011
110. <http://www.google.com/nexus/tech-specs.html> Agosto 2011
111. http://www.pcworld.com/article/173033/evolution_of_the_cell_phone.html Agosto 2011
112. http://www.nintendo.co.uk/NOE/en_GB/systems/technical_details_1072.html
Agosto 2011
113. <http://electronics.howstuffworks.com/wii.htm> Agosto 2011
114. <http://electronics.howstuffworks.com/microsoft-kinect.htm> Agosto 2011
115. http://www.nowgamer.com/features/988413/the_history_of_the_joystick.html
Agosto 2011
116. <http://gear.ign.com/articles/647/647295p1.html> Agosto 2011
117. **“Game Design Workshop: Designing, Prototyping, and Playtesting Games”**
Tracy Fullerton, Christopher Swain y Steven Hoffman, CMP Books. ISBN: 1578202221
118. **“Video Game Design Revealed”** Guy W. Lecky-Thompson, Cengage Learning
ISBN: 1-58450-562-1.
119. <http://www.classic-games.com/atari2600/specs.html> Agosto 2011
120. http://www.nintendo.co.za/NOE/en_ZA/systems/technical_details_1168.html
Agosto 2011
121. http://en.wikipedia.org/wiki/Nintendo_Entertainment_System#cite_b Agosto 2011
122. <http://www.8-bitcentral.com/nintendo/snes.html> Agosto 2011

123. http://www.jlsnet.co.uk/index.php?page=ps_specs Agosto 2011
124. <http://www.whatconsole.co.uk/ps3.php> Agosto 2011
125. <http://www.whatconsole.co.uk/ps3.php> Agosto 2011
126. <http://www.whatconsole.co.uk/wii.php> Agosto 2011.
127. <http://www.1up.com/features/all-about-wii> Agosto 2011.
128. <http://us.playstation.com/psvita/tech-specs/> Agosto 2011
129. <http://www.nintendo.com/3ds/hardware/specs> Agosto 2011
130. **“Mobile Software Development and Mobile Porting –Technology and Cost Issues”** White Paper, Rapidsoft Systems 2009
131. **“Portability Analysis in Mobile Gaming using J2ME”** Hari Menon, Lane Department of Computer Science and Electrical Engineering, 2006
132. **“Ingeniería del Software”** Ian Sommerville, Pearson. ISBN:978-84-7829-074-1
133. <http://pc.ign.com/articles/117/1176231p1.html> Agosto 2011
134. <http://pc.ign.com/articles/111/1117338p1.html> Agosto 2011
135. **“Real-Time Rendering Tricks and Techniques in DirectX”** Kelly Dempksi, Stacy L. Hiquet ISBN: 1-931841-27-6
136. **“Core Techniques and Algorithms in Game Programming”** Daniel Sánchez-Crespo Dalmau, New Riders Publishing. ISBN : 0-1310-2009-9
137. **“Software Engineering for Game Developers”** John P. Flynt y Omar Salem, Thomson Course Technology. ISBN: 1-59200-155-6
138. <http://www.esenthel.com/?id=compare> Septiembre 2011
139. <http://gafferongames.com/game-physics/fix-your-timestep/> Septiembre 2011

140. "**Advanced 2D Game Development**" Jonathan S. Harbour Course Technology PTR, ISBN: 1-59863-342-2
141. <http://www.libsdl.org/> Septiembre 2011
142. http://www.ceremade.dauphine.fr/~peyre/geodesic_computations/ Septiembre 2011
143. "**3D Game Engine Design, A practical aproach to real time Computer Graphics**" David H Eberly, Magic Software Inc, Morgan Kaufmann Publisher. ISBN: 0122290631
144. "**3D Game Engine Architecture Engineering Real-Time Applications with Wild Magic**" David H. Eberly, Morgan Kaufmann Publisher. ISBN: 0-12-229064-X
145. "**Shaders for game programming and Artists**" Sebastian St-Lauren Thomson Course Technology PTR, ISBN: 1-59200-092-4
146. "**OpenGL® Shading Language, Second Edition**" Randi J. Rost, Addison Wesley Professional, ISBN-10: 0-321-33489-2.
147. <http://www.opengl.org/about/overview/> Octubre 2011
148. <http://www.tomshardware.com/reviews/opengl-DirectX,2019-10.html> Octubre 2011
149. "**Beginning Direct3D Game Programming 2nd Edition**" Wolfgang F. Engel, Premier Press, ISBN: 1-931841-39-x
150. http://www.nvidia.com/content/GTC-2010/pdfs/2157_GTC2010.pdf Octubre 2011
151. "**AI for Game Developers**", David M. Bourg y Glenn Seeman, O'Reilly, ISBN 0-596-00555-5
152. "**AI Game Programming Wisdom**" Steve Rabin, Charles River Media, ISBN 1584500778.

153. “**Artificial Intelligence for Games**” Ian Millington, Morgan Kaufmann, ISBN 13: 978-0-12-497782-2
154. <http://gameinternals.com/post/2072558330/understanding-pac-man-ghost-behavior> Octubre 2011
155. “**Inteligencia Artificial, Un enfoque moderno**” Russell Norvig, Pearson Prentice Hall. ISBN: 978-84-205-4003-0
156. “**Programming Game AI by Example**” Mat Buckland, Wordware Publishing, Inc. ISBN: 1-55622-078-2
157. “**Networking and online games understanding and engineering multiplayer internet games**” Grenville Armitage, Mark Claypool, y Worcester Polytechnic Institute. John Wiley & Sons, LTd. ISBN-10: 0-470-01857-7
158. “**Multiplayer Game Programming**” Todd Barron, Prima Tech. ISBN: 0-7615-3298-6
159. “**Comunicaciones y Redes de Computadores**” William Stallings, Pearson Prentice Hall. ISBN: 978-84-205-44110-5
160. <http://tools.ietf.org/html/rfc2460#section-1> Noviembre 2011
161. “**Distributed Systems: Concepts and Design**” George Coulouris, Jean Dollimore y Tim Kindberg, Addison Wesley. ISBN-10: 0201619180
162. “**Beginning Game Audio Programming**” Mason McCuskey, Premier Press, ISBN: 1-59200-029-0
163. “**Game Sound Technology and Player Interaction: Concepts and Developments**” Mark Grimshaw, University of Bolton UK Advisory Board ISBN: 9781616928285
164. “**The Scientist and Engineer's Guide to Digital Signal Process, Second Edition**” Steven W. Smith, California Technical Publishing. ISBN: 0-9660176-6-8.

165. **“Audio Programming for Interactive Games”** Martin D. Wilde, Focal Press.
ISBN: 0-240-51941-8
166. **“Secrets of the Game Business”** Francois Dominic Laramee, Charles River
Media ISBN: 1-58450-282-7.
167. **“The business and culture of digital games Gamework/Gameplay”** Aphra
Kerr, Sage Publications, ISBN-10: 1412900476.
168. **“Estructura y evolución de la reciente industria del videojuego”** Martín
Raposo, Universidad de Palermo.
169. **“The Video Game Industry An Industry Analysis, from a VC Perspective”** Nik
Shah y Charles Haigh, TUCK at darthmout.
170. **“Video Games Industry Overview An Analysis of the Current Market and
Future Growth Trends”** Aoife Cunningham, Holger Langlotz, Marc Rhode y Clayton
Whaley INTERNATIONAL BUSINESS PROJECT 2008
171. **“Industria de Desarrollo de Videojuegos en Argentina”** Asociación de
Desarrolladores de Videojuegos Argentina 2004
172. http://www.jetro.go.jp/en/reports/market/pdf/2005_49_r.pdf **“Trends in the
Japanese Videogame Industry”** Industrial Report. JETRO Japan Economic Monthly,
Agosto 2005
173. http://www.cnbc.com/id/41062675/Video_Game_Sales_Drop_6_in_2010_Second_Year_of_Declines Noviembre 2011
174. http://www.theesa.com/facts/pdfs/ESA_EF_2011.pdf Noviembre 2011
175. http://www.gamasutra.com/php-bin/news_index.php?story=25506 Noviembre
2011
176. <http://sites.duke.edu/soc142-videogames/international-trade-patterns/industry-growth/> Noviembre 2011

177. http://www.gamasutra.com/view/news/30127/Avista_Game_Industry_Global_Market_Cap_At_100105_Billion.php Noviembre 2011
178. <http://lsvp.wordpress.com/2008/07/02/29-business-models-for-games/> Diciembre 2011
179. <http://venturebeat.com/2011/11/09/apple-and-android-are-now-dominating-mobile-game-market/> Diciembre 2011.
180. http://www.gamasutra.com/php-bin/news_index.php?story=18583 Diciembre 2011
181. http://www.flashgamelicense.com/view_library.php Diciembre 2011
182. **“La industria de videojuegos en la Argentina, Un análisis en base a la Segunda Encuesta Nacional a empresas desarrolladoras de videojuegos”**
Dirección general de Comercio Exterior, Ministerio de desarrollo económico, Centro de Estudios para el Desarrollo Económico Metropolitano 2010.
183. <http://www.lanacion.com.ar/1385771-la-argentina-crece-en-produccion-de-videojuegos> Diciembre 2011
184. <http://www.emol.com/noticias/tecnologia/2011/09/19/504049/industria-de-videojuegos-argentina-busca-un-hit-para-seguir-en-alza.html> Diciembre 2011
185. **“About independent development in Latin America, How can a game developer make games and earn experience in a region with a high percentage of digital illiteracy?”** Agustin Perez Fernandez.
186. <http://eiffel.itba.edu.ar/wavi2011/> Diciembre 2011
187. <http://www.adva.com.ar/informacion/> Diciembre 2011
188. <http://www.adva.com.ar/public/downloads/estatuto.pdf> Diciembre 2011
189. <http://www.verticportals.com/socialgaming/> **“Social Gaming, A white paper by Vertic Portals”** Vertic Portals 2010

190. http://www.digibusiness.fi/uploads/reports/1271751069_GPB_Social_Gaming_Report_-_March_2010.pdf **“Social Gaming: The fastest growing segment of games market”** Alec Dafferne, Justine Chan y Remy Valett, GP bullhund 2010.
191. **“REDES SOCIALES”** Lambertucci, Osvaldo D. Universidad Nacional de Luján 2008.
192. <http://es.wikipedia.org/wiki/OpenCL> Diciembre 2011.