# Resumen

## Análisis de la programación visual

UNIVERSIDAD POLITÉCNICA
DE TECÁMAC


**Profesora:**
Torres Servín Emmanuel.
Programación visual.
**Grupo:**
ingeniería en software
4322-IS

**Estudiantes:**
1321124285 | Hernández Silva Raúl Abraham.
1321124381 | Gutiérrez García Harold Guillermo.

# Contenido

## Conceptos de programación orientada a objetos.

Es un modelo o un estilo de la programación que nos guía a como trabajar con él. Se basa en el concepto de clases y objetos, este tipo de programación es utilizada para hacer una estructura de programación de software en piezas simples y reutilizables que son los códigos (clases) para crear instancias individuales de objetos.


**Características.**

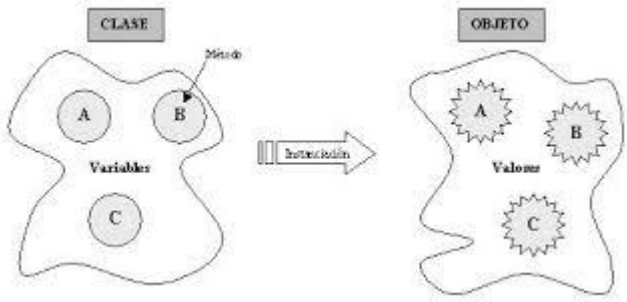
En este se manejan conceptos básicos como lo son clases, objetos, atributos, métodos y se caracteriza por emplear la abstracción de datos, herencia, encapsulamiento y polimorfismo. Como consejo estas características deben de estudiarse para ser comprendidas para poder aplicarlas en la programación orientada a objetos.

## Aplicación de eventos.

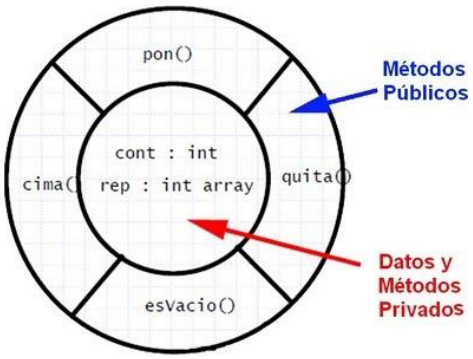
Es la base de lo que llamamos interfaz de usuario, aunque puede emplearse también para desarrollar interfaces entre componentes de Software o módulos del núcleo.

## Características de componentes.

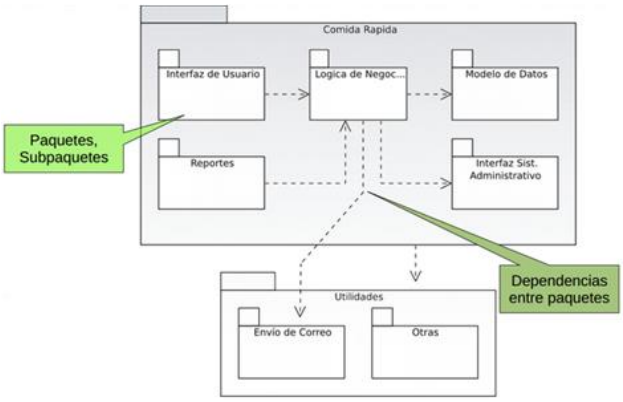
**Abstracción:** las características específicas de un objeto, aquellas que lo distinguen de los demás tipos de objetos.



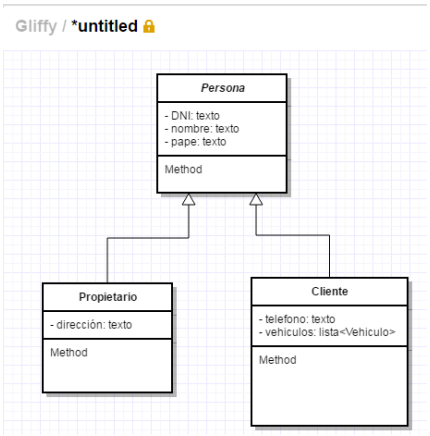

**Encapsulamiento:** Es el ocultamiento del estado, es decir, de los datos miembro, de un objeto de manera que sólo se puede cambiar mediante las operaciones definidas para ese objeto.




**Modularidad:** consiste en dividir un programa en módulos que puedan compilarse por separado, sin embargo, tendrá conexiones con otros módulos. El modularidad también tiene principios y es: la capacidad de descomponer un sistema complejo



**Jerarquía:** Se utiliza en programación orientada a objetos, para clasificar a las abstracciones y a eso se le conoce como Jerarquía.




Así mismo este modelo tiene elementos secundarios:

- Tipos (Tipificación)
- Concurrencia
- Persistencia.

## Métodos visuales.

La programación orientada a objetos (POO) es una forma de programación en donde se desarrollan soluciones utilizando componentes u objetos de software.

A diferencia de la programación estructurada, la POO se parece más al mundo real, porque se trabaja con objetos.

Hay muchos componentes de este tipo, atributos, etc. Hay muchos componentes de este tipo, como pueden ser los botones (TButton), etiquetas de cómo pueden ser los botones (TButton), etiquetas de texto (TLabel), formas (TShape), etc.

- Cronogramas
- Tableros Kanban
- Calendarios

Cada una de estas formas de ver tu trabajo tiene sus ventajas y desventajas, y elegir cuál funciona mejor para ti dependerá del tipo de proyecto que planifiques. Por ese motivo, es muy importante que el software de gestión visual de proyectos te brinde la posibilidad de cambiar entre diferentes tipos de vista. De esa forma, todos los involucrados, de colaboradores individuales a ejecutivos, pueden verse beneficiados por el software para gestión de proyectos.

## Requerimientos visuales de proyectos distribuidos y de escritos.

**Necesidad:** Un interesado demanda un requerimiento.

**Característica:** Un servicio proporcionado por el sistema, por lo general formulado por un analista de negocios.

**Caso de uso:** Una descripción del comportamiento del sistema descrito como una secuencia de acciones.

**Requisito complementario:** Otro requisito (generalmente no funcional) que no puede ser contemplado en los casos de uso.

**Caso de prueba:** Una especificación de las entradas necesarias para una prueba, las condiciones de ejecución y resultados esperados. Tiene el papel de comprobar si los casos de uso derivados de los casos de prueba y los requisitos complementarios se aplican correctamente.

**Escenario:** Una secuencia específica de acciones o una ruta de acceso específica a través de un caso de uso. Ayudan a derivar en casos de uso a partir de los casos de prueba y facilitan el diseño e implementación a través de los casos de uso.

## Herramientas y lenguajes.

Algunos ejemplos de estas herramientas son: Scratch, adecuada a partir de los 8 años, ScratchJr, que es la versión de Scratch adaptada para niños y niñas de entre 5 y 7 años, Lightbot, etc. Incluso la empresa Google ha creado un precioso doodle llamado Coding for Carrots que ejemplifica este tipo de herramientas

Los lenguajes de programación visual permiten a los usuarios crear programas mediante la manipulación de elementos gráficos, en lugar de especificarlos exclusivamente de manera textual.

Y estos son algunos lenguajes que ocupa la programación visual: HTML, CSS, JavaScript y JSON. Saque el máximo provecho de todo lo que le ofrecen LESS y Sass, y use PHP, Python o C# con ASP.NET

# Referencias.

*4.2 componentes del entorno visual*. (s. f.). CIDECAME UAEH. http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro15/42_componentes_del_entorno_visual.html?fbclid=IwAR1Zhs5QzK8VQ-n2WMc51xksYbE5DPcLB5I8zr5XFP7pmnXJyO0Bed7fk4E

*Características de la POO*. (s. f.). Portal Académico del CCH. https://portalacademico.cch.unam.mx/cibernetica1/algoritmos-y-codificacion/caracteristicas-POO#:~:text=El%20paradigma%20de%20la%20programación,,%20herencia,%20encapsulamiento%20y%20polimorfismo.

*Lenguajes - visual studio*. (s. f.). Visual Studio. https://visualstudio.microsoft.com/es/vs/features/web/languages/?fbclid=IwAR3Qj7d_WrIbkVsXGKIP4U4iZNCoohU9iwdhobLy4GNb1Z9FqoXO5gaeGmQ

*Programación orientada a eventos: Características, ejemplos, ventajas, aplicaciones*. (s. f.). Lifeder. https://www.lifeder.com/programacion-orientada-a-eventos/?fbclid=IwAR0Fh6nbnNaFkvDN7umlQuKli2gkrDMnAG3Qb0zAZP1pUt1ITKd26Qyhock

# Programación Visual

## Diseño de programación visual

### Proceso de construcción de maquetado

La maquetación o se refiere a transformar un diseño gráfico en una interfaz funcional

Luego de que hagas un análisis inicial de requisitos, ten en cuenta estos pasos para que evites reprocesos

1. Arquitectura de la información definida
2. Flujos de navegación claros
3. Insumos gráficos aprobados y optimizados
4. ¿Qué elementos son reusables en la interfaz?
5. ¿En qué navegadores y dispositivos se soportará la aplicación?
6. De lo general a lo específico
7. Busca constantemente buenas prácticas
8. Trabaja en equipo, aprende a preguntar rápido y sé recursivo
9. Todo puede cambiar
10. Atención a los detalles

### Interactividad de componentes

Las interacciones se utilizan para modelar los aspectos dinámicos de las colaboraciones que implementan la funcionalidad del sistema.

Sociedad de objetos que desempeñan roles específicos y colaboran entre sí para desarrollar un determinado comportamiento.

### Proceso de diseño e integración de contenidos y componentes

Al considerar los contenidos desde las primeras etapas de diseño, nos aseguramos de que estén bien integrados y se comunique correctamente lo que se quiere expresar.

Los componentes de software son unidades ejecutables de adquisición, implementación y producción independiente que pueden formar parte de un sistema funcional.

### Diseño, multitimedi e interfaces de programación visual

Llamamos multimedia a la combinación de varios medios de comunicación en un único espacio visual

La interfaz de usuario en un videojuego es el punto de interacción entre el jugador y el juego.

El diseño suele implicar establecer relaciones visuales, sistémicas o geométricas entre las partes de un diseño.

# Desarrollo de aplicaciones con programación visual

Para desarrollar programas, Windows provee una librería de rutinas y funciones (SDK - Kit de desarrollo de software) que permiten gestionar componentes como menús, diálogos, ventanas.

Visual C++ es un entorno integrado de desarrollo que permite la programación orientada a objetos (POO) conjuntamente con el sistema de desarrollo SDK (también denominado API) de Windows. Al ser un entorno integrado Visual C++ incluye, entre otras, las siguientes herramientas de desarrollo:

Un evento es una señal que comunica a una aplicación que ha sucedido algo importante. Por ejemplo, cuando un usuario hace clic en un control en un formulario, el formulario puede provocar un evento Click y llamar a un procedimiento que controla el evento

Herramientas

Editor de texto

Compilador/Enlazador

Depurador

Visor de datos y dependencias (Browser)

Datos: Son indispensables para darle valor a las variables

Diseño y estilo: Es para darle presentación a una aplicación

Propiedades

Accesibilidad: Sirve para garantizar el proyecto

Apariencia: Es necesario para que sea mas interactivo para el usuario

Comportamiento: Instrucciones controladas por los usuarios en las aplicaciones

Funcionalidad: Envía una señal para que la ejecute la aplicación y inicie un proceso

Proceso de programación visual: Es la creación mediante el uso de elementos gráficos y textuales

# Trabajo de investigación

## Videojuegos


UNIVERSIDAD POLITÉCNICA
DE TECÁMAC

**Profesora:**
Torres Servín Emmanuel.
Programación visual.
**Grupo:**
ingeniería en software
4322-IS

**Estudiantes:**
1321124285 | Hernández Silva Raúl Abraham.
1321124381 | Gutiérrez García Harold Guillermo.

# Contenido

## Introducción.

un Game Designer o diseñador de videojuegos es el profesional encargado de diseñar todos los elementos que componen el juego.

Aunque no existe una definición estricta sobre lo que debe hacer un game designer pues varía según el videojuego que se esté haciendo, su labor en común es ser el encargado de definir todos los aspectos del videojuego, desde mecánicas, personajes, historia, mundo del videojuego, etc.

La labor primordial del game designer es dar un equilibrio a todos los aspectos del juego, desarrollar toda la idea del videojuego. Junto a que debe liderar y revisar el proceso de desarrollo completo del juego.

En este texto abordáremos las principales características del diseño de videojuegos, así como sus componentes y funciones. De igual manera explicaremos los procesos de diseño de interfaces y estructura lógica de un videojuego.

# Conceptos de videojuegos.

## Conceptos y tipos de game designer, storyboard.

### ¿Qué hace un Game Designer?

Dependiendo del proyecto, el game designer puede asumir funciones diferentes, pero lo cierto es que como responsable del diseño del videojuego debe encargarse de los siguientes puntos concretos:

- **Definir el funcionamiento del videojuego.**

  Partiendo de una idea inicial y de la experiencia que quiere que el jugador experimente con el juego, el game designer da forma a las mecánicas principales que lo sustentan, así como a otros elementos como la historia, el guion o los personajes.

- **Establecer las reglas.**

  Es prioritario fijar las reglas y los niveles. Debe plantearse todas las posibles situaciones a las que podrá enfrentarse el usuario. También debe fijar las recompensas que recibirá el jugador cada vez que realice una determinada acción.

- **Dar vida a los personajes.**

  Toda historia tiene sus protagonistas y si hay un elemento que ha sufrido una mayor evolución dentro de los videojuegos, estos son los personajes. Hoy son mucho más complejos y elaborados que hace tan solo unos años, no solo su aspecto físico, sino también su carácter. Cada uno debe tener su propia personalidad y habilidades, también los enemigos que haya decidido incluir.

Fijar los criterios sobre el modo en el que los personajes interactuarán entre sí y no solo eso, también decidirá si los usuarios podrán establecer contacto con otros jugadores y de qué manera.

- **Situar el juego en un escenario.**

  Tan importante como el argumento son los escenarios y el momento histórico en los que tendrá lugar la acción. Han de ser coherentes con la historia.

- **Dirigir y supervisar el trabajo de los equipos del proyecto.**

  Una vez que ha trasladado al resto de grupos las ideas, debe trabajar con ellos para que todo encaje y controlar la evolución de los diferentes procesos.

**¿Que debe saber un Game designer?**

- **Saber comunicar:** Al ser el punto de unión de un equipo de trabajo con diversos perfiles, es primordial saber comunicarse y escuchar atentamente a cada área.

- **Resolución de conflictos:** En los videojuegos siempre se está innovando, muy probablemente se encontrará con obstáculos que nadie ha resuelto, así que debe ser capaz de resolver cualquier tipo de conflicto.

- **Nunca parar de aprender:** Un buen game designer tiene una gran curiosidad por sacar nuevas ideas, mejorar características, experimentar con distintos sistemas, etc. Por ello, siempre están aprendiendo algo nuevo, desde programación básica hasta las bases del diseño.

- **Entender las diversas áreas del equipo:** Es el encargado de sacar adelante el videojuego, la comunicación es primordial y por ello debe entender la forma en que se comunica cada área, saber que es un sprite, un trigger, renderizar, compilar, etc. Debe saber un poco de todas las áreas.

- **Nunca parar de jugar:** No se puede seguir tendencias e innovar sin aprender de lo que hay en la industria. Debe jugar, pero analizando el videojuego, debes jugar muchos títulos, pero sin viciarse y pasar 1000 horas en un solo juego.

- **Escribir de manera clara:** Para definir un videojuego debe llenar el Game Design Document, en ella se va a encontrar todo lo relacionado con el desarrollo del videojuego.

**Game Design Document.**

Es un documento de diseño en constante actualización cuyo contenido debe describir todo acerca del videojuego en cuestión, es utilizado como guía durante el proceso de desarrollo del juego.

Suele estar formado por las siguientes secciones:

- Historia
- Caracteres
- Diseño de nivel y del entorno
- Jugabilidad
- Arte
- Sonido y música
- Interfaz de usuario
- Controles de juego
- Accesibilidad
- Monetización

**Tipos de Game Designer.**

Existen diferentes tipos de diseñadores de juegos, teniendo en cuenta su ámbito de acción:

- **Lead designer.** Suele ser quien tiene la idea inicial o en su conjunto. Supervisa la actividad de todos los equipos, comprobando que trabajan en sintonía.
- **Game writer.** Es el autor de la historia, la persona que escribe el guion.
- **Level designer.** Se encarga del diseño de los mapas, fijando su duración, la dificultad o la ubicación y recorrido de los enemigos.
- **System designer**. Diseña las reglas del juego.

**Storyboard.**

Se trata de un conjunto de imágenes que forman una guía visual mediante la cual se pueden previsualizar las escenas que van a emplearse. Muchas veces usamos el término guion gráfico para referirnos a los storyboards. Y es que gracias a él podemos hacernos una ligera idea de lo que veremos luego en la pantalla.

En un videojuego, los storyboards son la base del diseño de escenas y transiciones que dan coherencia y un orden a seguir en el diseño gráfico del proyecto.

## Tipos y características de motores de videojuegos y lenguajes de videojuegos.

Un motor de videojuego es un término que hace referencia a una serie de librerías de programación que permiten el diseño, la creación y la representación de un videojuego.

El aspecto más destacado a la hora de elegir un motor de videojuegos entre todos los disponibles que hay en el mercado son las capacidades gráficas, ya que son las encargadas de mostrar las imágenes 2D y 3D en pantalla, así como calcular algunos aspectos como los polígonos, la iluminación, las texturas, etc. Otras características para tener en cuenta a la hora de la elección son la facilidad de aprender a usar el motor de videojuegos y la facilidad para exportar el juego a diferentes plataformas.

Algunas de las funcionalidades más importantes son:

- **El motor de físicas**

  El motor de físicas es el que hace posible aplicar aproximaciones físicas a los videojuegos para que tengan una sensación más realista en la interacción de los objetos con el entorno. En otras palabras, es el encargado de realizar los cálculos necesarios para que un objeto simule tener atributos físicos como peso, volumen, aceleración, gravedad, etc.

- **El motor de sonido**

  Los sonidos y la banda sonora de un videojuego es también una parte muy importante. El motor de sonidos es el encargado de cargar pistas, modificar su tasa de bits, quitarlas de reproducción, sincronizarlas entre otras cosas.

- **El scripting**

  Todos los motores de videojuegos tienen un lenguaje de programación que permite implementar el funcionamiento de los personajes y objetos que forman parte del videojuego.

Dentro de las diferentes opciones de motores de videojuegos podemos distinguirlos en populares y motores propietarios o privados que son los creados por empresas importante de videojuegos para diseñar sus títulos más populares.

Los motores populares más utilizados y que más posibilidades dan al desarrollador son:

- **Unreal Engine:** Fue creado por Epic Games en 1998. En 2012 se presentó Unreal Engine 4, una nueva versión del motor. Entre las empresas que lo utilizan se encuentran Electronic Arts y Ubisoft. Utiliza el lenguaje de programación C++.

- **Unity 3D:** Se trata de una de las innovaciones más importantes creadas por la comunidad científica y de videojuegos y permite jugar a complejos videojuegos en 3D sin necesidad de instalarlos en el ordenador. Los videojuegos creados con el motor Unity 3D se pueden jugar en un navegador con el reproductor Unity Web Player, eliminando la necesidad de instalar el videojuego.

Los motores propietarios o profesionales más conocidos por los juegos que han sido diseñados con ellos:

- **Frostbite Engine:** Este motor para videojuegos creado por Digital Illusions CE se utiliza para crear videojuegos de acción en primera persona. Se presentó principalmente para la serie de videojuegos Battlefield. Ha jugado un papel fundamental en prácticamente todos los videojuegos de EA. L a nueva versión del motor Frostbite Engine es Frostbite 3.

- **Decima Engine:** Alberga herramientas y características para crear inteligencia artificial, física, lógica y mundos en el desarrollo, así como compatibilidad con 4K y HDR.

- **Luminous Studio:** Es un motor de videojuegos multiplataforma desarrollado y usado internamente por Square Enix. Con este motor se desarrolla el juego Final Fantasy.

Los lenguajes de programación más usados en el Game Designer son los siguientes:

- **C++**

  Es uno de los lenguajes de programación más utilizados por los profesionales del sector. Se utiliza para algunas de las principales plataformas de hardware como PlayStation y Xbox, siendo uno de los más compatibles con la mayor parte de motores de juego. Unido a un rápido tiempo de ejecución, C++ permite a los desarrolladores tener un amplio control sobre el hardware. Dominar este tipo de programación nos facilitará en gran parte la enseñanza de otros lenguajes.

- **C#**

  Uno de los más extendidos en el entorno de Windows y de los favoritos por programadores de videojuegos. Siendo más fácil de aprender que el C++, también se caracteriza por ser algo más limitado que el anterior. No obstante, se pueden crear juegos con este lenguaje para múltiples plataformas (PlayStation, Xbox, Android, iOS, Windows). Magnífica opción para principiantes.

- **Java**

  Con cierta similitud a C++, y siendo uno de los más utilizados mundialmente, con este lenguaje se puede programar en la mayor parte de plataformas. Entre sus ‘pros’ se encuentra la gran cantidad de frameworks para el desarrollo 3D, haciendo que podamos hacer prácticamente de todo con este lenguaje. Uno de sus ‘cons’: se ejecuta dentro de su máquina virtual, llevando esto a una pérdida de rendimiento.

## Metodologías de desarrollo de videojuegos.

El desarrollo de un videojuego conlleva un proceso de diseño y desarrollo que conlleva dirigirnos por diferentes fases que serán ilustradas a continuación.

### Fase de Concepción

Todo comienza con una idea a partir de la cual se conformarán los aspectos fundamentales. Se determina el género o géneros del videojuego, cómo será el proceso de juego (game play), y también se constituye un guion gráfico (story board) en el que se tratan todo tipo de ideas preconcebidas que pueden ir adaptándose, como por ejemplo el estilo de los personajes, el ambiente, la música, etc. Una vez se sabe qué hacer entonces es el momento de diseñar.

### Fase de Diseño

Se empieza definiendo los elementos que componen el juego. Se desarrolla la historia, se crean bocetos de guiones para determinar los objetivos, se deciden los personajes principales, el contexto, etc.

Utilizando estos esbozos de guiones los artistas se ponen manos a la obra para crear conceptos del aspecto del juego, la forma en que se visualizarán los personajes, los escenarios, objetos, etc. Su trabajo es presentar propuestas visuales para ir dando forma a la idea original.

También se describen los elementos sonoros de los que consta el juego: efectos de sonidos, ambientación, música, voces, etc. Aunque todavía no se compone ni se graba nada.

Paralelamente se especifica el funcionamiento general del videojuego, algo que depende del género, ya que señalan la forma en que las entidades virtuales interactúan dentro del juego.

Finalmente, con una idea algo más clara del rumbo que tomará el juego, se hace el diseño de la programación, que describe la manera en la que se implementará el

videojuego, el lenguaje o lenguajes de programación que se utilizarán, las metodologías que se seguirán, etc.

Todo lo anterior tendrá como objetivo generar el Documento de Diseño que especificará el desarrollo del arte, las mecánicas y la programación del videojuego.

**Fase de Planificación**

Esta etapa tiene como objetivo identificar las diferentes tareas para desarrollar el videojuego. Se reparte el trabajo entre los distintos componentes del equipo de desarrollo, se fijan plazos de entregas, se planifican reuniones de seguimiento, etc.

**Fase de Producción**

Una vez se tiene claro lo que hay que hacer, cómo hacerlo, y se ha planificado el tiempo para llevarlo a cabo, entonces se empieza la producción con el objetivo de crear el juego, como mínimo en una versión inicial o prototipo a mejorar gradualmente.

Se llevan por tanto a cabo todas las tareas de la fase de planificación teniendo como guía el documento de diseño: programación, ilustración, desarrollo de interfaces, animación, modelado, desarrollo del sonido, etc.

Si finalmente se logra ensamblar correctamente todas las piezas entonces esta fase culmina (por ahora). Sin embargo, al igual que en el desarrollo de software tradicional, es muy difícil que todo salga bien a la primera, por lo que se entra en una fase para probar a fondo el videojuego.

**Fase de Pruebas**

En esta etapa se corrigen los errores del proceso de programación y se mejora la jugabilidad a medida que se prueba el juego.

Generalmente encontraremos dos tipos: las pruebas Alpha, realizadas por un pequeño grupo de personas generalmente involucradas en el desarrollo, y las

pruebas beta, realizadas por un equipo externo de jugadores. Las primeras tienen el objetivo de corregir defectos graves y mejorar características fundamentales no contempladas en el documento de diseño, mientras que las segundas se enfocan en detectar fallos menores y perfilar la experiencia de usuario.

**Fase de Distribución/Márketing**

En cuanto a la distribución es el proceso de crear las copias del juego ya finalizado y llevarlo a las tiendas (ya sean físicas o digitales) para que los jugadores puedan comprarlo o hacerse con él.

Por otro lado, el márketing es también fundamental para dar a conocer el videojuego y conseguir el mayor número de jugadores posibles. No tiene un orden concreto dentro del desarrollo, pues algunas empresas empiezan a hacer campaña de sus videojuegos meses e incluso años antes de publicarlos. La verdad es que depende de los recursos que los desarrolladores quieran destinar a promocionar la obra y no tiene por qué ser un departamento dentro de la propia empresa, sino que tanto la distribución como el márketing se pueden delegar a empresas externas especialistas en estas áreas.

Vale la pena comentar el fenómeno “hype”, que ocurre cuando una empresa hace uso de una excesiva publicidad para dar a conocer su producto, creando incluso una necesidad inexistente en los potenciales consumidores. Lo malo ocurre cuando el producto no está a la altura de lo prometido y entonces se convierte en el blanco de multitud de críticas en muy poco tiempo, algo que puede perjudicar gravemente la imagen de los creadores.

**Fase de Mantenimiento**

Pese a que el juego esté finalizado y en las manos de los jugadores, su ciclo de vida aún está lejos de terminar. La fase de mantenimiento es el momento de arreglar nuevos errores, mejorarlo, etc. Esto se hace sacando parches o actualizaciones al mercado.

Sin embargo, es también una oportunidad para seguir sacándole partido. Ya sea en forma de micro transacciones, suscripciones de pago o incluso con expansiones completas que añaden nuevas características al videojuego sin modificar en profundidad el motor de este, digamos que sería más o menos como aprovechar al máximo la base inicial.

## Proceso de diseño de interfaces de videojuegos en 2d y 3d.

La interfaz de un videojuego, la UI, es lo primero que nos atrae como jugadores en una consola. Es nuestra puerta de entrada, la invitación que nos hacen los diseñadores para entrar al mundo que han construido. Diseñar una buena UI es todo un desafío. Muchas veces, los estudios descuidan esta faceta porque no tienen tiempo o recursos suficientes para dedicarle, pero un mal diseño de interfaz de un videojuego es lastrar la experiencia del usuario. Un buen diseño nos permitirá empezar en el juego de una manera sencilla e intuitiva.

la interfaz de usuario en un videojuego es el punto de interacción entre el jugador y el juego. Su objetivo fundamental es el de brindar la información necesaria para que el usuario pueda hacer todo lo que el juego le propone de manera totalmente fluida. Un buen diseño de UI guía de manera directa o intuitiva para que el jugador pueda recorrer el mundo de tu videojuego de forma correcta.

A continuación, se enlistarán algunas cosas que se debe tener en cuenta a la hora de diseñar la interfaz de un videojuego:

- **Entorno/Plataforma.** Lo primero para tener en cuenta es dónde se va a jugar el juego que se está diseñando. Se debe tener en cuenta las posibilidades y limitaciones que ofrece la plataforma. No es lo mismo hacer juegos para smartphones que para una consola o que para un PC.

- **Contenido.** Un buen diseño de UI proporciona al jugador toda la información necesaria para que pueda interactuar con el juego y que todo sea fluido.

- **Diseño Visual.** Los videojuegos, casi siempre, entran por los ojos. Un apartado visual feo o denso en la interfaz del juego puede resultar contraproducente y sacar al jugador de la experiencia inmersiva que quieres proporcionarle. Se debe definir el estilo de arte.

- **Arquitectura de la información.** Definir qué elementos son de mayor o menor importancia para el usuario y organizarlos de tal forma que todo resulte en un diseño de interfaz coherente y relevante.

# Desarrollo de prototipos de videojuegos.

## Concepto, tipos y características de los motores de videojuego.

Un motor de videojuegos es un framework o un conjunto de herramientas que ayudan a agilizar el proceso de desarrollo de un videojuego. Los motores proveen herramientas al programador, que le permiten dedicar menos tiempo a aspectos poco importantes para la idea general del videojuego, pero que son de suma importancia para la experiencia del usuario final, es decir, con ayuda de los motores, los programadores pueden enfocarse en desarrollar buenos juegos sin perder tiempo en otras tareas.

Entre las herramientas de los motores de videojuegos podemos distinguir algunas de suma importancia como: motor de renderizado, física de videojuegos y detección de colisiones, scripting, motor de sonidos, inteligencia artificial y administradores de memoria.

- **Motor de renderizado**

  Es el encargado de mostrar las imágenes 2D o 3D en pantalla, así como de calcular algunos aspectos como los polígonos, la iluminación, difuminado, texturas, entre otros. Es muy común que un motor de renderizado esté implementado a nivel de API como DirectX u OpenGL puesto que estas proveen de abstracciones de la GPU y acceso a algunos componentes de hardware independientemente de la plataforma que se utilice.

- **Física de videojuegos y detección de colisiones**

  Este motor es el que permite aplicar aproximaciones físicas a los videojuegos para que luzcan más naturales y sean más ergonómicos para el jugador. En otras palabras, es el encargado de realizar los cálculos necesarios para que un objeto simule tener atributos físicos como peso, volumen, estado físico, gravedad etc. Algunos también se encargan de detectar colisiones entre objetos en el espacio del videojuego para simular las respuestas ante el

estímulo y brindar al programador la opción de reaccionar a las colisiones como lo desee (ejemplo: si una bala colisiona contra el jugador, la bala desaparecerá y el jugador perderá un corazón de salud).

- **Scripting**

  Los motores de videojuegos también deben permitir al programador crear sus propios scripts porque es muy común que muchas funcionalidades deseadas no estén previamente implementadas en el motor y sea necesario añadirlas por medio de scripts o, a veces, el programador simplemente desea ajustar el comportamiento de algunas configuraciones por defecto. Esta herramienta diversifica las posibilidades de desarrollo.

- **Motor de sonidos**

  Casi tan importante como los gráficos, son los sonidos y la banda sonora de un videojuego. El motor de sonidos es el encargado de cargar pistas, modificar su tasa de bits, quitarlas de reproducción, sincronizarlas etc.

- **Inteligencia artificial**

  Casi todos los videojuegos de la actualidad hacen uso de inteligencia artificial (generalmente para personajes o enemigos). Una buena inteligencia artificial es la que separa a un juego fácil y aburrido de uno entretenido y desafiante. Generalmente la implementación de la inteligencia artificial en muchos videojuegos se queda un poco corta porque la limitan a pequeños árboles de comportamiento y decisiones, pero la IA puede ser tan compleja como el programador lo desee, aunque una mejor IA requiere dedicar más tiempo a su desarrollo.

- **Administrador de memoria**

  Los enemigos, paisajes, personajes, edificios y en general, todos los componentes de un videojuego consumen recursos. La tarea del

administrador de memoria es saber cuándo liberar algunas celdas porque los enemigos que las ocupaban dejaron de existir, el paisaje ya no necesita estar cargado en memoria, o en casos extremos, porque el procesador se comienza a calentar. Este componente no es muy necesario en juegos pequeños, pero con juegos AAA que muestran miles de polígonos y partículas en pantalla en un mismo frame, este componente es quizás el principal a considerar para lograr un buen producto final.

**Motores de código abierto frente a privados**

Si alguna vez buscaste algún motor de videojuegos, posiblemente notaste que los más importantes son privados y pertenecen a grandes empresas como Epic. Aunque existen muchos motores de código abierto, la realidad es que muy pocos son tan buenos como para competir contra los motores de licencia privada. Existen muy buenas razones para que esto suceda, entre ellas tenemos:

- La programación de motores de videojuegos requiere de una inversión muy fuerte ya que son muy complejos, por lo que es difícil ver uno de código abierto que haya sido desarrollado por la comunidad.

- La industria de videojuegos tiene pocos años de haber despegado, aún es muy pronto para que un motor desarrollado por la comunidad pueda competir contra los de renombre.

- Generalmente quienes los desarrollan son empresas grandes que necesitan algunas funcionalidades específicas para sus futuros títulos que ningún otro motor del mercado les brinda. Esto se traduce en motores que permanecen de uso privado dentro de dichas empresas, es decir, no se lanzan a la venta.

- Quienes los desarrollan lo hacen para generar ganancias por licencias, no es probable que lancen su código fuente para el público puesto que perderían una buena fuente de ingresos.

- Estas fueron algunas de las características, herramientas y componentes más importantes que un motor de videojuegos necesita tener para ser considerado como tal, aunque cabe señalar que existen algunas más que también son muy importantes como los motores de animaciones, cinemáticas, procesamiento paralelo, entre otras.

A continuación, te señalo algunos de los motores más importantes de la industria, si te interesa informarte puedes consultar los enlaces (algunos de esos motores son muy importantes y conocidos, pero son de uso privado para las empresas desarrolladoras):

- Blender
- Corona SDK
- Fox Engine
- Frostbite
- GameMaker Studio
- JmonkeyEngine
- Rockstar Advanced Game Engine
- RPG Maker
- Source 2
- Unity
- Unreal Engine

**Integración de motores de videojuegos con programación visual de acuerdo con los requerimientos del videojuego.**

Los motores de videojuegos suelen estar especializados hacia el género de juego en que son usados. Hay grandes empresas de desarrollo de videojuegos que incluso trabajan con diferentes motores dependiendo del tipo juego o franquicia que desarrollen. Esto se debe principalmente a que los motores muchas veces son desarrollados con características especiales para atender necesidades específicas del género de videojuego en el que se trabaja. Por ejemplo, hay motores especializados para desarrollar first person shooters, otros para juegos de plataformas, otros para juegos de peleas, otros para RPGs, otros para MMOs, otros para deportes, otros para carreras de autos, etc.

Es posible encontrar motores especializados que serían difíciles usar para el desarrollo de otros géneros de videojuegos sin tener que recurrir a la programación, por ejemplo, hacer juegos de plataformas en un motor especializado en first person shooters, hacer un juego RPG en un motor especializado en juegos carreras de autos, o un juego MMO en un motor especializado en juegos single player. Esto no quiere decir que sea imposible utilizarlos, simplemente algunas herramientas o características tendrán que ser desarrolladas -programadas- aparte.

Por todo lo anterior, cuando un desarrollador o artista quiere hacer su videojuego es importante tener claro el tipo de juego que se va a desarrollar y, en base a las necesidades técnicas, escoger el motor pertinente. Dando un par de ejemplos extremos: si se busca desarrollar un juego con elementos 3D en el motor Gamemaker Studio 2, sería bastante complejo ya que está especializado en juegos 2D. Por el contrario, hacer un juego de plataformas 2D utilizando Cryengine podría complicar demasiado los procesos.

Entre otros factores que podrían impactar en la selección de un motor de juegos se podrían encontrar:

- Presupuesto
- Tamaño del equipo de desarrollo
- Demográfica del equipo de desarrollo (proporción entre diseñadores, artistas y programadores)
- Experiencia del equipo de desarrollo
- Tiempo de desarrollo proyectado
- Utilización de programas externos de desarrollo (Maya, 3DsMax, Blender, Houdini, Illustrator, Photoshop, etc)
- Dependencia a librerías o plugins específicos (AWS, Steamworks, Speedtree, Havok, etc)
- Portado a plataformas destino (PC, macOS, HTML5, Xbox, PlayStation, Switch, Web, iOS, Android, etc.)

## Transición narrativa y lenguaje visual de videojuegos.

Narración y juego ofrecen al videojuego una base de especificidad que la diferencia de otras formas de representación simbólica de las cuales, no obstante, hereda sus patrones de configuración, como el cine. El componente lúdico planteado por el videojuego conlleva, en su propia génesis, una narración específica, dado que el jugador participa del entorno digital actuando sobre el mismo gracias a un proceso de inmersión pronunciado, que permite incluso su transformación como sujeto simbólico integrante y participante de la narración. Esta actuación inmersiva y transformadora habla de un tiempo presente para el relato, el del jugador que, transformado en un narrador excepcional, modifica el curso de la acción a través del personaje en una constante actualización del relato en el proceso narrativo mismo: “En estos entornos los saberes del narrador y del personaje emergen de la fractura del espacio dramático y se dispersa la percepción de lo narrado bajo el predominio de una nueva perspectiva” (ESNAOLA y LEVIS, 2008).

Detrás de un juego no hay uno, si no varios programadores, que tienen un alto conocimiento de diferentes metodologías y lenguajes. De igual manera hay artistas que se encargan del diseño, el apartado gráfico y el sonoro. También es indispensable que haya alguien con facilidad en la creación de un guion, entre muchas otras cosas. Sin embargo, la programación visual ah ayudado a la democratización del desarrollo de videojuegos, sobre todo orientado a plataformas móviles, donde la demanda es muy alta para títulos sencillos.

## Explicar el proceso de desarrollo de videojuego acorde a los elementos de programación visual.

El documento elaborado por el diseñador del juego es lo que sirve de base para que programadores, artistas y modeladores puedan dar forma al producto final.

### Determinar los objetivos

El primer paso por seguir es determinar cuáles son los objetivos del proyecto que se va a desarrollar. ¿Qué quieres hacer con el juego? ¿Qué historia quieres contar y cómo la quieres contar? ¿Qué tipo de experiencia vas a ofrecer a los jugadores?

### Determinar el público objetivo

Una vez que se han definido los objetivos es hora de determinar qué tipo de público puede estar interesado en el juego. No es lo mismo diseñar un juego de la saga Souls que hacer Super Mario. Parte del público puede coincidir, pero hay juegos que solo atraen a un nicho concreto de jugadores.

### Pensar en qué consola/dispositivo se va a hacer

Obviamente, antes de ponerte a describir nada sobre el funcionamiento del juego, sus sistemas o mundo, debes definir para qué plataforma se va a desarrollar el juego. Esto es importante de cara al trabajo del resto del equipo de desarrollo de un videojuego.

**Pensar en el género del juego**

Otro de los pasos importantes a dar en cómo diseñar un videojuego es pensar en cuál va a ser el género de este. No es lo mismo pensar en mecánicas para un shooter que pensar en los sistemas de un RPG.

**Diseñar el mundo del juego**

El mundo del juego es uno de los aspectos importantes en el diseño de videojuegos. ¿Qué ofrece el mundo al jugador? ¿Qué cuenta? ¿Es necesario que sea rico en detalles? No todos los juegos necesitan un mundo detallado y diseñado hasta el más mínimo detalle.

**Diseñar las mecánicas**

La parte jugable es lo que realmente hace diferentes a los videojuegos de otros medios culturales y artísticos.

Diseñar la interactividad de un juego es algo complejo y que requiere de mucho trabajo previo. Y también se necesita haber jugado mucho para conocer todo tipo de propuestas para modificar, combinar o añadir.

**Diseñar los niveles**

Otro de los puntos importantes en el diseño de videojuegos es el diseño de niveles.

Esto es algo que está estrechamente relacionado con el diseño del mundo del juego. ¿Va a ser un juego de mundo abierto o estructurado en niveles? Si está estructurado en distintos niveles, ¿va a estar estos interconectados entre ellos?

**Diseñar a los personajes**

El diseño de personajes se refiere a todo. Ya sea al protagonista del juego, cómo es, qué personalidad tiene, cómo se mueve, etc. ¿Es un avatar hueco o tiene personalidad propia?

Además, también hablamos de personajes secundarios y de personajes que solo son extras dentro del juego. Puede parecer que, muchas veces, los NPC's son parte de un decorado, pero un buen juego les da una personalidad propia.

Por supuesto, dentro del diseño de personajes también entra el diseño de enemigos, de ser el caso.

**Diseñar el contenido**

En el proceso de diseño de un juego, como hemos insistido a lo largo de todo el artículo, es necesario que tengas en cuenta muchas cosas. Una de ellas es el contenido del juego.

Con esto nos referimos al diseño de los ítems con los que puede interactuar el jugador con otros personajes que no son el avatar o protagonista del juego, o a elementos del mundo del juego, ya sea para destrucción o construcción de estos.

## Conclusión.

Para concluir, los videojuegos, actualmente, son una parte fundamental de la vida, a pesar de las controversias dentro del tema, han traído aportes culturales y artísticos al mundo, así como avances tecnológicos que son incluso empleados fuera del mundo del Gaming para funcionalidades variadas.

Un Game Designer es el artista y encargado de realizar este producto con los requerimientos necesarios para el éxito. El debe conocer cada área, artística como tecnológica para poder desarrollar su papel correctamente. De igual manera debe saber manejar cada herramienta y lenguaje necesario para la culminación de su proyecto.

En el desarrollo de un videojuego es indispensable la selección de un motor grafico adecuado, que cumpla con las características precisas del modelo de juego que se desea realizar y así poder generar una interfaz gráfica, precedida por un storyboard que siente las baces del diseño y obtener una experiencia de usuario satisfactoria.

# Referencias.


*Game Designer: Requisitos y funciones en el desarrollo de juegos*. (s. f.). UNIR. https://www.unir.net/ingenieria/revista/game-designer/

*¿Qué es un game designer?* (s. f.). La Nave Madrid. https://www.lanavemadrid.com/event/que-es-game-designer/#:~:text=Como%20bien%20dice%20su%20nombre,diseñador%20gráfico%20o%20artista%20conceptual.

*¿Qué es y cómo crear un storyboard?* (s. f.). ESDESIGN. https://www.esdesignbarcelona.com/actualidad/diseno-grafico/que-es-y-como-crear-un-storyboard

*¿Qué hace un game designer?* (s. f.). Platzi. https://platzi.com/blog/que-hace-un-game-designer/

*Los 3 mejores lenguajes de programación para videojuegos*. (s. f.). Game It - Consolas, videojuegos y hardware. Gaming Culture. https://www.gameit.es/los-3-mejores-lenguajes-de-programacion-para-videojuegos/

*¿Qué es un motor de videojuegos? – Observatorio del Gabinete de Tele-Educación*. (s. f.). UPM [Blogs], servicio de blogs UPM – Sitio dedicado a blogs para la docencia e investigación de la Universidad Politécnica de Madrid. https://blogs.upm.es/observatoriogate/2018/07/04/que-es-un-motor-de-videojuegos/

*Las 7 fases más importantes en el desarrollo de juegos | Escuela de Videojuegos | Hektor Profe*. (s. f.). Documentos | Hektor Profe | Formación sobre Python y Unity. https://docs.hektorprofe.net/escueladevideojuegos/articulos/fases-del-desarrollo-de-videojuegos/

*El diseño de interfaz de usuario en un videojuego | Tokio School*. (s. f.). Tokio School. https://www.tokioschool.com/noticias/diseno-interfaz-videojuego/

*¿Qué es un motor de videojuegos (game engine)?* (s. f.). Blog de tecnologías de la información. https://codingornot.com/que-es-un-motor-de-videojuegos-game-engine

*Programación visual: crear un videojuego sin tener ni idea de código*. (s. f.). ADSLZone. https://www.adslzone.net/2018/07/14/programacion-visual-videojuegos/