**Facultad de Ingeniería**

**Implementación de librerías gráficas.**

**Fases iniciales para el desarrollo de un juego**

**CURSO: PROGRAMACIÓN DE VIDEOJUEGOS CON C**

**DOCENTE: COTA SENCARA, DAVID WILLIAM**

**INTEGRANTES:**

* **COCHACHIN QUITO, FRANK RONALD 1512159**
* **ESPINOZA GARCÍA, JOSÉ LUIS 1411784**
* **GONZALES RUEDA, JUNIOR MIGUEL 1512188**
* **GONZALES RUEDA, SUSANA IRENE 1512187**

**2018**

**Tabla de contenido**

[**CAPITULO 1 4**](#_Toc526260561)

[**MARCO HISTÓRICO 4**](#_Toc526260562)

[**CAPITULO 2 5**](#_Toc526260563)

[**MARCO TEÓRICO 5**](#_Toc526260564)

[**CAPITULO 3 6**](#_Toc526260565)

[**DESARROLLO 6**](#_Toc526260566)

[**DEFINICIÓN DESARROLLO DE UN JUEGO 6**](#_Toc526260567)

[**FASES DEL DESARROLLO DE VIDEOJUEGOS 6**](#_Toc526260568)

[**Fase de Concepción 6**](#_Toc526260569)

[**Fase de Diseño 7**](#_Toc526260570)

[**Fase de Planificación 8**](#_Toc526260571)

[**Fase de Producción 9**](#_Toc526260572)

[**Fase de Pruebas 10**](#_Toc526260573)

[**Fase de Distribución y Marketing 11**](#_Toc526260574)

[**Fase de Mantenimiento 11**](#_Toc526260575)

[**DEFINICION DE LIBRERÍA 12**](#_Toc526260576)

[**OBJETIVOS DE LAS LIBRERÍAS 12**](#_Toc526260577)

[**TIPOS DE LIBRERÍAS 12**](#_Toc526260578)

[**Librerías de Bajo Nivel 12**](#_Toc526260579)

[**PROCESSING 13**](#_Toc526260580)

[**D3 13**](#_Toc526260581)

[**Bokeh 14**](#_Toc526260582)

[**Librerías de Alto Nivel 15**](#_Toc526260583)

[**Allegro 15**](#_Toc526260584)

[**SFML 15**](#_Toc526260585)

[**SDL 16**](#_Toc526260586)

[**GOSU 16**](#_Toc526260587)

[**CAPITULO 4 17**](#_Toc526260588)

[**MARCO CONCEPTUAL 17**](#_Toc526260589)

[**CONCLUSIONES 19**](#_Toc526260590)

[**BIBLIOGRAFÍA 20**](#_Toc526260591)

# **CAPITULO 1**

## **MARCO HISTÓRICO**

El desarrollo de videojuegos comenzó en 1947, cuando la idea de un videojuego fue concebida y patentada por Thomas T. Goldsmith Jr y Estle Ray Mann, los cuales llenaron una aplicación de patente en Estados Unidos el 25 de enero de 1947. El éxito de los videojuegos se ha extendido hasta el día de hoy y posee un futuro prometedor para beneficio de la Industria de los videojuegos y de los usuarios finales.

La evolución de la industria de los videojuegos ha estado ligada a una serie de hitos, determinados particularmente por juegos que han marcado un antes y un después, o por fenómenos sociales que han afectado de manera directa a dicha industria.

La expansión del videojuego es tan relevante que actualmente se trata de una industria multimillonaria capaz de rivalizar con las industrias cinematográfica y musical.

A pesar de la vertiginosa evolución de la industria del videojuego, hoy en día existe un gran número de retos que el desarrollador de videojuegos ha de afrontar a la hora de producir un videojuego. En realidad, existen retos que perdurarán eternamente y que no están ligados a la propia evolución del hardware que permite la ejecución de los videojuegos. El más evidente de ellos es la necesidad imperiosa de ofrecer una experiencia de entretenimiento al usuario basada en la diversión, ya sea a través de nuevas formas de interacción, como por ejemplo la realidad aumentada o la tecnología de visualización 3D, a través de una mejora evidente en la calidad de los títulos, o mediante innovación en aspectos vinculados a la jugabilidad.

Actualmente la evolución de los videojuegos está estrechamente ligada a la evolución del hardware que permite la ejecución de los mismos. Esta evolución atiende, principalmente, a dos factores: la potencia de dicho hardware y las capacidades interactivas del mismo. En el primer caso, una mayor potencia hardware implica que el desarrollador disfrute de mayores posibilidades a la hora de, por ejemplo, mejorar la calidad gráfica de un título o de incrementar la IA (Inteligencia Artificial) de los enemigos. Este factor está vinculado al multiprocesamiento. En el segundo caso, una mayor riqueza en términos de interactividad puede contribuir a que el usuario de videojuegos viva una experiencia más inmersiva (por ejemplo, mediante realidad aumentada) o, simplemente, más natural (por ejemplo, mediante la pantalla táctil de un smartphone).

# **CAPITULO 2**

## **MARCO TEÓRICO**

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.

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.

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.

Por otra parte, se conoce como librerías (o bibliotecas) a cierto tipo de archivos que podemos importar o incluir en nuestro programa. Estos archivos contienen las especificaciones de diferentes funcionalidades ya construidas y utilizables que podremos agregar a nuestro programa, como por ejemplo leer del teclado o mostrar algo por pantalla entre muchas otras más.

Las librerías no son únicamente archivos externos creados por otros, también es posible crear nuestras propias librerías y utilizarlas en nuestros programas. Las librerías pueden tener varias extensiones diferentes, las más comunes son:

.lib,. bpl, .a, .dll, .h y algunas más ya no tan comunes.

También una librería se define como un conjunto de subprogramas utilizados para desarrollar software. Las bibliotecas contienen código y datos, que proporcionan servicios a programas independientes, es decir, pasan a formar parte de estos. Esto permite que el código y los datos se compartan y puedan modificarse de forma modular. Algunos programas ejecutables pueden ser a la vez programas independientes y bibliotecas, pero la mayoría de estas no son ejecutables. Ejecutables y bibliotecas hacen referencias (llamadas enlaces) entre sí a través de un proceso conocido como enlace, que por lo general es realizado por un software denominado enlazador.

# **CAPITULO 3**

## **DESARROLLO**

### **DEFINICIÓN DESARROLLO DE UN JUEGO**

Es el proceso de creación de un videojuego, desde el concepto inicial hasta el videojuego en su versión final. Es una actividad multidisciplinaria, que involucra profesionales de la programación, el diseño gráfico, la animación, el sonido, la música, la actuación, etc.

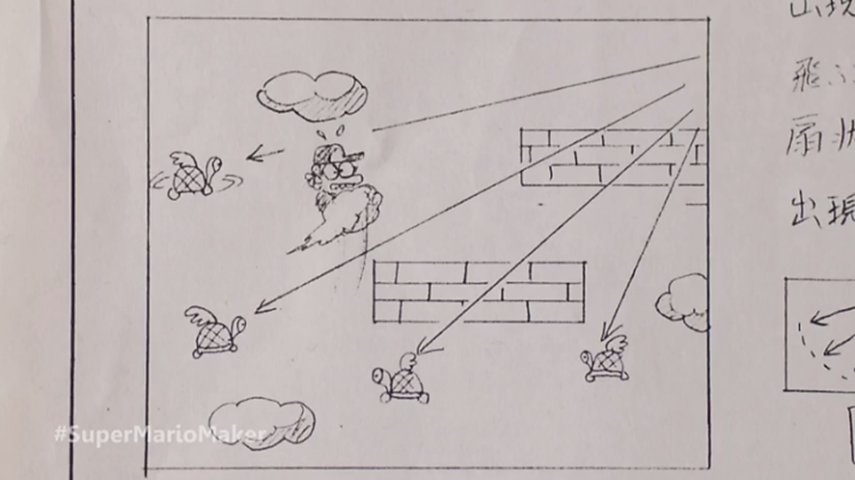
Es similar a la creación de software en general, aunque difiere en la gran cantidad de aportes creativos (música, historia, diseño de personajes, niveles, etc.) necesarios. El desarrollo también varía en función de la plataforma objetivo (PC, móviles, consolas), el género (estrategia en tiempo real, RPG, aventura gráfica, plataformas, etc.) y la forma de visualización (2D, 2.5D y 3D).

Cabe mencionar que el diseño de juegos es usualmente considerado un proceso de creación iterativo, esto quiere decir que los diseñadores tendrán que pasar por cada uno de estos pasos repetidas veces hasta que consideren que el resultado sea el mejor.

## **FASES DEL DESARROLLO DE VIDEOJUEGOS**

### **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 (gameplay), y también se constituye un storyboard 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.



**Ilustración 1: Viñeta de un Storyboard de los primeros Super Mario**

### **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.

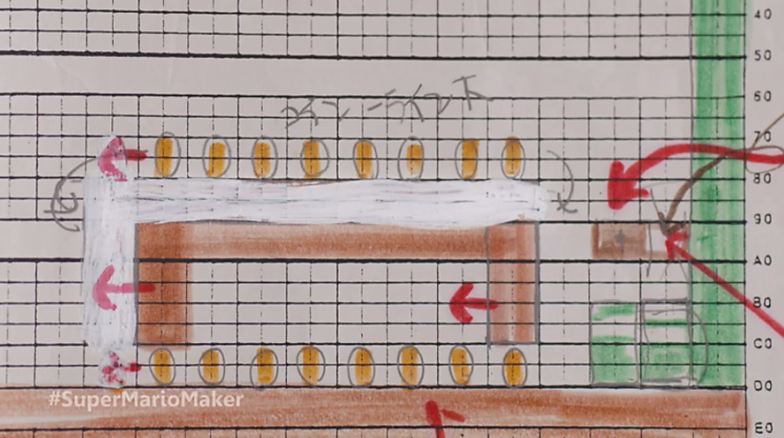
Utilizando estos esbozos de guiones los artistas se ponen manos a la obra para crear conceptos del aspecto del juego, la forma en que se visualizarán los personajes, los escenarios, objetos, etc. Su trabajo es presentar propuestas visuales para ir dando forma a la idea original.

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.

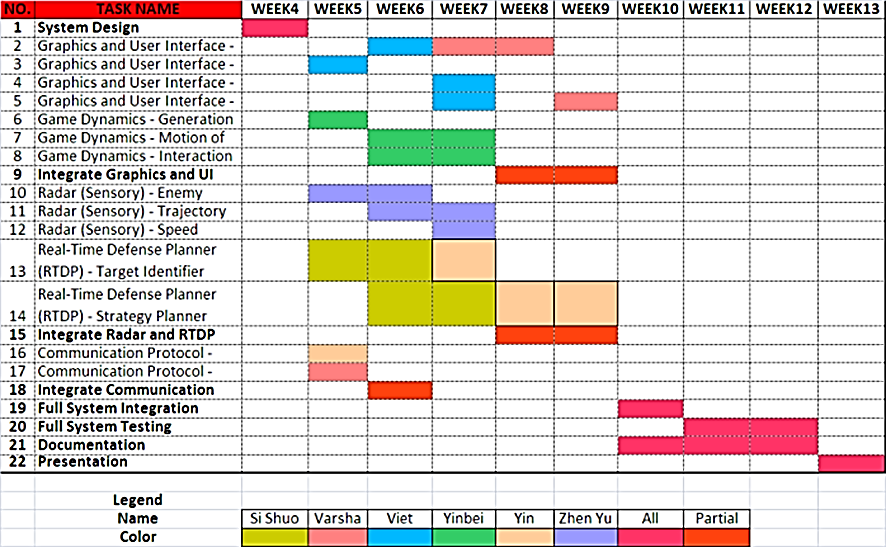
Todo lo anterior tendrá como objetivo generar el Documento de Diseño que especificará el desarrollo del arte, las mecánicas y la programación del videojuego.



**Ilustración 2: Diseño original de un escenario de Super Mario**

### **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.



**Ilustración 3: Planificación de un proyecto mediante el Diagrama de Gantt**

### **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 de sonidos, 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.

****

**Ilustración 4: Oficinas de Ubisoft en Toronto(Canadá)**

### **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.

****

**Ilustración 5: Comparación entre Alpha y Beta de Battlefield 3**

### **Fase de Distribución y Marketing**

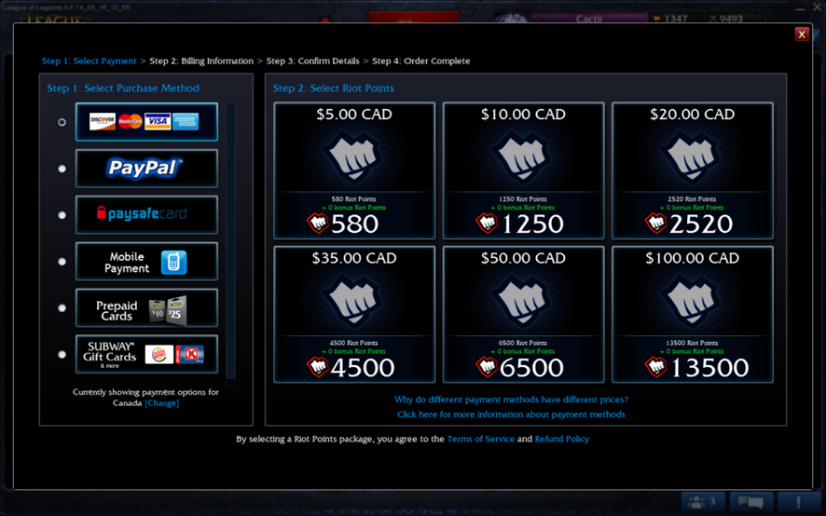
En cuanto a la distribución es el proceso de crear las copias del juego ya finalizado y llevarlo a las tiendas (ya sean físicas o digitales) para que los jugadores puedan comprarlo o hacerse con él.

Por otro lado, el marketing 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 porqué ser un departamento dentro de la propia empresa, sino que tanto la distribución como el marketing se pueden delegar a empresas externas especialistas en estas áreas.

### Fase de Mantenimiento

A pesar de 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.



**Ilustración 6: League of Legends con sistema de micropagos**

# **DEFINICION DE LIBRERÍA**

También llamada biblioteca, es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.

A diferencia de un programa ejecutable, el comportamiento que implementa una biblioteca no espera ser utilizada de forma autónoma, sino que su fin es ser utilizada por otros programas, independientes y de forma simultánea. Por otra parte, el comportamiento de una biblioteca no tiene por qué diferenciarse demasiado del que pudiera especificarse en un programa. Es más, unas bibliotecas pueden requerir de otras para funcionar, pues el comportamiento que definen refina, o altera, el comportamiento de la biblioteca original; o bien la hace disponible para otra tecnología o lenguaje de programación.

Las librerías graficas generan imágenes en base a unos modelos matemáticos y unos patrones de iluminación, texturas, etc, que son funcionalidades ya construidas y utilizables para poder agregar importando dichas librerías a nuestro programa.

## **OBJETIVOS DE LAS LIBRERÍAS**

* Independencia del hardware (tanto dispositivos de entrada como de salida).
* Independencia de la aplicación (la librería es accedida a través de una única interface (al menos para cada lenguaje de programación) para cualquier aplicación.

## **TIPOS DE LIBRERÍAS**

### **Librerías de Bajo Nivel**

Librerías que son más complejas de programar y controlar, pero permiten sacar el máximo partido al hardware.

El paralelismo con lenguajes de programación es que Java sería un lenguaje de alto nivel ya que no se programa en función a parámetros físicos (direcciones de memoria, registros) mientras que Ensamblador sería un lenguaje de bajo nivel (se programa asignando valores a registros, etc.). C y C++ se consideran más bien en un nivel intermedio porque en realidad también permite programar a bajo nivel. Entre las librerías de bajo nivel se encuentran:

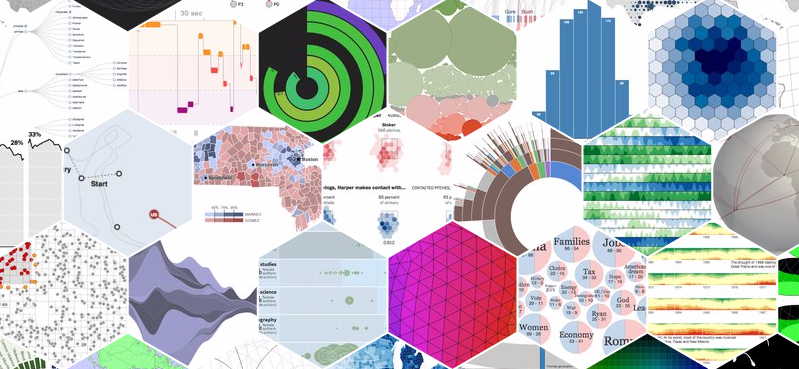
### **PROCESSING**

[Processing](http://processingjs.org/) ya tiene varios años en el mercado. Es una herramienta que puede ser descargada e instalada en cualquier plataforma, además cuenta con un lenguaje bastante fácil de usar al visualizar las líneas de código que permite hacer animaciones y gráficos interactivos.



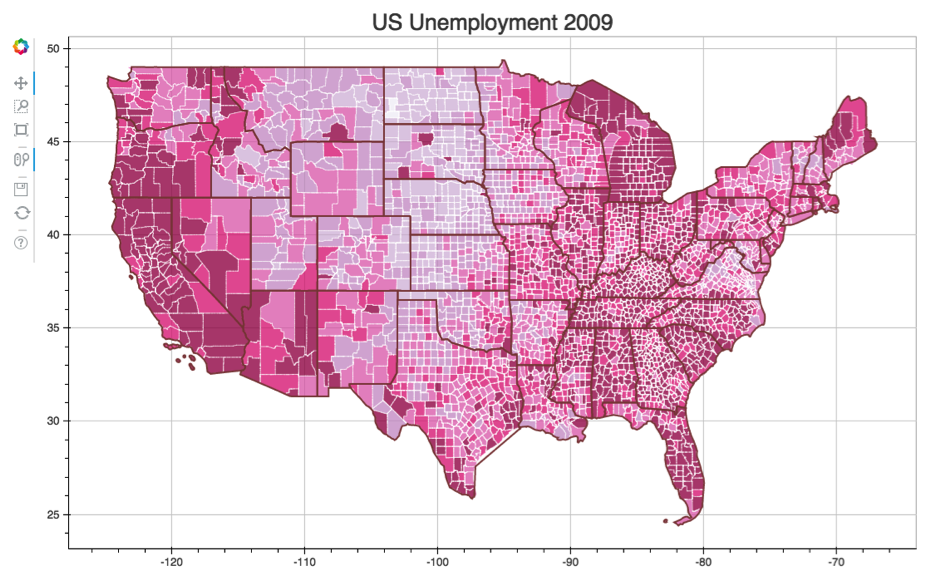
### **D3**

Librería que se ha convertido en la herramienta dominante para hacer visualizaciones basadas en gráficos de vectores SVG. Las ventajas de este tipo de gráficos es que no se ven alterados por la resolución de las pantallas, sin importar el zoom las imágenes nunca se pixelan. La librería permite hacer una variedad de gráficos avanzados como redes, árboles, mapas o burbujas, además de los gráficos usuales de barras o dispersión. Aprender bien la librería requiere bastante tiempo, sin embargo, es una gran inversión.



### [**Bokeh**](http://bokeh.pydata.org/en/latest/)

El objetivo de Bokeh es ofrecer gráficos elegantes, atractivos y sencillos, al estilo de la librería de JavaScript [D3.js](http://d3js.org/), pero también proporcionar una interactividad de alto nivel con grandes volúmenes de datos. Es una opción interesante si se quieren crear visualizaciones gráficas, aplicaciones con datos o tableros de mandos. Se pueden hacer visualizaciones de todo tipo, enfocadas fundamentalmente para navegadores modernos: [mapas de coropletas](http://bokeh.pydata.org/en/latest/docs/gallery/choropleth.html), [mapas de calor](http://bokeh.pydata.org/en/latest/docs/gallery/unemployment.html), [gráficos de líneas](http://bokeh.pydata.org/en/latest/docs/gallery/stocks_timeseries_chart.html), [de áreas](http://bokeh.pydata.org/en/latest/docs/gallery/area_chart.html), [de barras](http://bokeh.pydata.org/en/latest/docs/gallery/stacked_bar_chart.html).



### **Librerías de Alto Nivel**

Librerías que simplifican el desarrollo de los juegos y la gestión de los recursos del sistema, haciendo que el programador se centre en crear los efectos, movimiento, el diseño de escenarios, etc.

Esto tiene el inconveniente de que estas librerías de alto nivel toman ciertos atajos para simplificar el desarrollo, y por tanto no están especialmente optimizadas por lo que no se exprime al máximo el hardware sobre el que corren. Entre las librerías de Alto Nivel para LenguajeC/C++:

### **Allegro**

Biblioteca multiplataforma destinada principalmente a videojuegos y programación multimedia de fácil uso. Maneja tareas comunes de bajo nivel como crear ventanas, aceptar entradas de los usuarios, cargar datos, dibujar imágenes, reproducir sonidos, etc. y abstraer en general la plataforma subyacente.

Allegro es una intuitiva API C utilizable desde C++ y otros lenguajes, Hardware bitmap acelerado y soporte de dibujo gráfico primitivo, soporte de grabación de audio, carga y dibujo de fuentes, reproducción de video, etc.



### **SFML**

Simple and Fast Multimedia Library (SFML) es una biblioteca que proporciona una interfaz simple para facilitar el desarrollo de juegos y aplicaciones multimedia. Está compuesto por cinco módulos: sistema, ventana, gráficos, audio y red.

Con SFML, la aplicación puede compilarse y ejecutarse de manera predeterminada en los sistemas operativos más comunes: Windows, Linux, MacOS.

Su propósito principal es ofrecer una biblioteca alternativa a SDL, usando un enfoque orientado a objetos.



### **SDL**

Simple DirectMedia Layer (SDL) es una biblioteca de desarrollo multiplataforma diseñada para proporcionar acceso de bajo nivel a hardware de audio, teclado, mouse, joystick y gráficos a través de OpenGL y Direct3D. Es utilizado por software de reproducción de video, emuladores y juegos populares, incluido el galardonado catálogo de Valve y muchos juegos de Humble Bundle.

Compatible oficialmente con Windows, Mac OS X, Linux, iOS y Android. El soporte para otras plataformas se puede encontrar en el código fuente.

SDL está escrito en C, funciona de forma nativa con C ++, y hay enlaces disponibles para varios otros idiomas, incluidos C # y Python.



### **GOSU**

Biblioteca para el desarrollo de videojuegos en dos dimensiones en C++ y Ruby, disponible para los sistemas Mac OS X, Microsoft Windows y GNU/Linux.

Gosu es enfocado, liviano y tiene pocas dependencias. Proporciona:

* Una ventana y un ciclo principal.
* Gráficos 2D y texto, con tecnología OpenGL o OpenGL ES.
* Sonidos y música.
* Entrada de teclado, mouse y gamepad.

La versión para C++ también está disponible para iPhone, iPod e iPad. Esta biblioteca se ha utilizado en toda clase de juegos 2D, especialmente en competiciones cronometradas, incluyendo juegos de disparos, de scroll horizontal, juegos en perspectiva isométrica, etc.

Imagen que contiene imágenes prediseñadas

Descripción generada con confianza alta

# **CAPITULO 4**

## **MARCO CONCEPTUAL**

1. Videojuego: Aplicación interactiva orientada al entretenimiento que, a través de ciertos mandos o controles, permite simular experiencias en la pantalla de un televisor, una computadora u otro dispositivo electrónico.
2. Gameplay: Conjunto de acciones que puede realizar un jugador para interactuar con juego o la forma en la que este interactúa con el propio jugador.
3. Storyboard: conjunto de ilustraciones a modo de secuencia que acompaña al guion de una producción audiovisual como guía para entender mejor la idea del spot publicitario o film antes de realizarse.
4. Librería gráfica: Conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.
5. Interfaz gráfica de usuario: Conocida también como GUI (del inglés graphical user interface), es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.
6. Framework: Entorno de trabajo​ o marco de trabajo​ es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.
7. Software: El software de una computadora es todo aquel código ordenado de tal manera que le permite al usuario ordenarle a la misma que realice una tarea. También se deben subdividir en diversas categorías en base a las funciones que realizan en el sistema.
8. Hardware: es la parte física de un ordenador o sistema informático, está formado por los componentes eléctricos, electrónicos, electromecánicos y mecánicos, tales como circuitos de cables y circuitos de luz, placas, utensilios, cadenas y cualquier otro material, en estado físico, que sea necesario para hacer que el equipo funcione.
9. Diseño interactivo: Es el arreglo de gráficos, [texto](https://es.wikipedia.org/wiki/Texto), [videos](https://es.wikipedia.org/wiki/Videos), [ilustraciones](https://es.wikipedia.org/wiki/Ilustraci%C3%B3n), [sonidos](https://es.wikipedia.org/wiki/Sonidos), [animación](https://es.wikipedia.org/wiki/Animaci%C3%B3n), imágenes tridimensionales (3D), [realidad virtual](https://es.wikipedia.org/wiki/Realidad_virtual) y otros medios en un documento interactivo. Un documento interactivo puede ser una página [web](https://es.wikipedia.org/wiki/Web) con texto y enlaces a otras páginas web, o una aplicación autónoma más compleja que use [tipografía](https://es.wikipedia.org/wiki/Tipograf%C3%ADa), gráficos, sonido y video con controles interactivos como pueden ser botones, enlaces, etc.
10. Plugin: Es una aplicación informática que añade funcionalidades específicas a un programa principal. Su nombre procede del inglés (plug-in significa “enchufable”) y su presencia es muy habitual en los navegadores web, en reproductores de música y en sistemas de gestión de contenidos. Los plugins no son parches ni actualizaciones, sino propiedades añadidas a los programas originales, aparecidas por primera vez a mediados de los años 70 y conocidas también como complementos, extensiones y addons (del inglés add-on, “agregado”).

# **CONCLUSIONES**

Existe una variedad de información por estudiar y analizar dependiendo de la estructura lógica que estos tienen, consideramos por ello que las librerías gráficas son una herramienta eficiente para acceder a los datos de una manera más ágil y sencilla a través de las representaciones gráficas para una mejor interpretación de los datos.

Las librerías gráficas toman mayor importancia en la actualidad dada que la gran mayoría de estilos de aprendizaje está inmersa a la experiencia sensorial, donde los sentidos que toman relevancia son lo auditivo y lo visual para generar y procesar la información.

# **BIBLIOGRAFÍA**