

# TAREAS PROGRAMACION VISUAL

**UNIVERSIDAD POLITECNICA DE TECAMAC**

**PROFR: TORRES SERVIN EMMANUEL**

**EQUIPO:**

- **GEORGE SANCHEZ NORA ISABEL**
- **SALINAS REYES JOSE GUADALUPE**
- **VEGA OJEDA ALVARO JOSHUA**

**GRUPO: 5322IS**

**PERIODO: MAYO-AGOSTO 2022.**



## Programación orientada a objetos

La programación orientada a objetos es un modelo de programación informática que organiza el diseño de software en torno a datos u objetos, en lugar de funciones y lógica. Un objeto se puede definir como un campo de datos que tiene atributos y comportamiento únicos

La programación orientada a objetos se centra en los objetos que los desarrolladores quieren manipular en lugar de enfocarse en la lógica necesaria para manipularlos. Este enfoque de programación es adecuado para programas que son grandes, complejos y se actualizan o mantienen activamente

Para crear POO se hacen clases y luego se crean objetos a partir de dichas clases, que constituyen el modelo a partir del que se estructuran los datos y los comportamientos

## Programación orientada a eventos

La programación dirigida por eventos es un paradigma de programación en el que el flujo del programa está determinado por eventos o mensajes desde otros programas o hilos de ejecución.

Dependencia de eventos: El flujo del programa viene dado por eventos que pueden ser acciones del usuario, mensajes de otros programas, etc., separando la lógica de procesamiento de los eventos del resto del código de un programa, contrastando así con el procesamiento por lotes

Orientada al servicio: Se utiliza para escribir programas diseñados para el servicio sin ralentizar la computadora, ya que la orientación al servicio solo consume poco poder de procesamiento

Eventos: Es una condición que surge durante la ejecución de un programa y que requiere alguna acción por parte del sistema

Controlador de eventos: Es una determinada unidad del programa que se activa para reaccionar ante un evento. Es decir, es un tipo de función o método que ejecuta una acción específica cuando se activa un evento determinado

Funciones de activación: Son funciones que deciden qué código ejecutar cuando se produce un evento específico. Se utilizan para seleccionar qué controlador de eventos emplear al producirse un evento

Tiempo controlado: Es un código específico que se ejecuta en un momento determinado. Esto significa que es una tarea preestablecida por hacer

Ya que la programación orientada a eventos resulta ser más un enfoque que un tipo de lenguaje, las aplicaciones que se basan en eventos pueden crearse en cualquier lenguaje de programación.

Un caso de una situación impulsada por un evento se parecería a un temporizador de activación configurado para que haga una tarea específica en una hora determinada, como sonar y comunicar algo que diga: ¡Despierta!

### Componentes y métodos visuales y no visuales

Un componente es un elemento de un sistema software que ofrece un conjunto de servicios, o funcionalidades, a través de interfaces definidas. Un componente de software debe poseer las siguientes características:

- Ser reutilizable.
- Ser intercambiable.
- Poseer interfaces definidas.
- Ser cohesivos

Los componentes visuales son aquellos que, al utilizarlos, muestran algún elemento (o dibujo) en la pantalla y es el usuario de nuestros programas el que interactúa con él. El componente es el principal responsable de dibujar en la pantalla lo que sea oportuno, dependiendo de su estado del valor de sus atributos, etc. Hay muchos componentes de este tipo, como pueden ser los botones (button) etiquetas de texto (label), formas (shape), etc.

Los componentes no visuales se pueden colocar en los formularios de la misma manera que los controles, aunque este caso su posición es irrelevante.

Ejemplo: los componentes no visuales incluyen timer control, serial port service controller, entre otros

### Herramientas y lenguajes de programación visual

De la gran variedad de lenguajes de programación visual, aquí presentamos una pequeña selección.

Scratch: El lenguaje de programación visual más popular y utilizada es Scratch. Fue lanzado por primera vez en 2007 y tiene como propósito enseñar a los niños a programar de manera sencilla

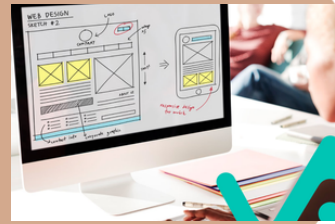
Blockly: Blockly es una llamada biblioteca. Proporciona un editor de programación visual al que se añaden aplicaciones Android, iOS y web. Blockly también utiliza bloques gráficos que encajan entre ellos

NEPO: NEPO es gratuito, está basado en Scratch y utiliza la biblioteca Blockly. Esta biblioteca se ha ampliado con funcionalidades propias. Una ventaja de NEPO son sus interfaces abiertas, que permiten controlar otros sistemas de hardware o robots

Grape: Grape es un entorno de desarrollo gráfico. Permite incluso a los principiantes en programación programar con micro controladores en pasos simples

App Inventor: App Inventor proviene originalmente de Google. Esta interfaz gráfica permite programar aplicaciones para teléfonos móviles Android con bloques gráficos

Es el diseño digital de una web o aplicación, estas se utilizan para visualizar las ideas y conceptos del diseño, estructura y los elementos en detalle.



El diseño son las relaciones visuales, sistematicas o geometricas entre el programa y su diseño.

Es toda información o archivos con la que el usuario puede interactuar,  
-textos -imágenes  
-audio -vídeos



Diseño

Multimedia

Conceptos

Interfaces

Diseño de programación visual

Consiste en proporcionar un entorno visual sencillo para permitir al usuario interactuar con el programa



Procesos de diseño e integración de contenidos y componentes

Un componente tiene un comportamiento que se define por los eventos ante los que responde y los métodos que ejecuta. Las propiedades de un componente pueden examinarse y modificarse mediante métodos o funciones miembro, que acceden a dicha propiedad,

El objetivo de diseñar componentes e integrarlos es traducir el modelo del diseño al software que se estará construyendo, éste se elabora por completo ya que incluya todos los atributos y operaciones relevantes para la aplicación

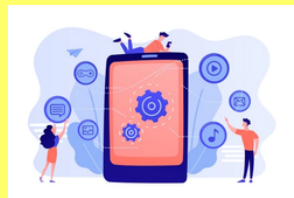
Creación de componentes (visuales y no Visuales) definidos por el usuario



construcción de maquetado.

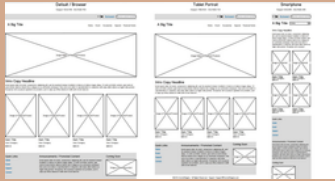
Interactividad entre componentes

Una aplicación puede tener distintos componentes interactivos. Un componente interactivo es un elemento que puede cambiar dependiendo de cómo actúe el usuario.



La interactividad se puede dar entre los diferentes elementos que contiene la aplicación, ya sea un formulario, etiquetas, botones, cajas de texto, casillas de verificación, etc.

Evaluar los objetivos y la funcionalidad  
Establecer la estructura del sitio web  
Definir que elementos y componentes habrá, así como su estilo.  
Definir los colores, la tipografía y todo lo que incluye el diseño final de la plataforma.



# DESARROLLO DE APLICACIONES CON PROGRAMACIÓN VISUAL

## EVENTOS VISUALES

Cuando el usuario opere sobre el programa, algo deberá suceder, para ello se ocupan los eventos, un evento es cualquier suceso que puede ocurrirle a un componente y que pueden condicionar el comportamiento y apariencia del programa.

Desde un click de ratón hasta la pulsación de una tecla de teclado o la reproducción de un audio o video, entre muchas otras cosas son las que pueden activar un evento

## COMPONENTES Y HERRAMIENTAS VISUALES DE ENTORNOS DE DESARROLLO

Los componentes son elementos con una funcionalidad muy concreta. Cada uno de ellos está destinado a realizar una tarea en la aplicación.

Cada componente poseerá una serie de eventos que puede recibir o generar.

Hay varias clasificaciones para los componentes. Una de ellas es los visuales y no visuales.

Un componente es visual cuando tiene una representación gráfica a la hora del diseño y ejecución (boton, barra de scroll, etc.),

Un componente no es visual en caso contrario (temporizadores, cuadros de diálogo (no visible), etc)

## PROPIEDADES DE LOS COMPONENTES VISUALES

Pueden establecerse y modificarse las propiedades de los componentes tanto a la hora de diseño como durante la ejecución

**Accesibilidad**  
Son las propiedades que permiten poder modificar la formade usar los determinados componentes

**Comportamiento**  
Hay propiedades que pueden controlan su comportamiento que pueden modificar de que formava a funcionar un componente

**Diseño.**  
Son las propiedades que modifican su color, su tamaño, su tipografia, etc.

**Apariencia**  
Esta modifica el como se vera cierto componente, el espacio que ocuapara, con que detalles y en que posicion

**Datos**  
Las propiedades de los datos, son los datos que esterecibira, asi como lo que puede almacenar o mostrar

**Estilos**  
Aqui se pueden declara los diferentes tipos de estilos que puede tener cada uno de los componenetes

# DESARROLLO DE APLICACIONES CON PROGRAMACIÓN VISUAL

## FUNCIONALIDAD DE LOS EVENTOS VISUALES

Los eventos esperan sin realizar ninguna tarea hasta que se produzca un evento. Una vez producido, ejecutan alguna tarea asociada a la aparición de ese evento y cuando concluye, el programa vuelve al estado de espera. aunque no es necesario que el usuario interactúe con el programa, puede ocurrir un evento incluso al cargar la página.  
Cada elemento del programa tendrá sus propios eventos, lo que nos permite tener un sinfín de posibilidades

Su funcionalidad es demasiado importante en la programación visual, debido a la interacción del usuario con los componentes visuales y sus respectivos eventos.

## PROCESO DE PROGRAMACIÓN VISUAL.

La programación visual usada para diseñar y desarrollar aplicaciones con un entorno visual amigable y fáciles de utilizar por el usuario.  
Con el paso del tiempo, aparecieron herramientas que incluían código que automatiza parte de las tareas más comunes en la detección y tratamiento de eventos.

La Programación Orientada a Eventos facilita el proceso de programación visual por su practicidad al generar rápidamente algunas aplicaciones basadas en objetos visuales.  
De igual manera ayuda al programador novato en el rápido aprendizaje de desarrollo de nuevos programas con alta calidad en muy poco tiempo.

Los videojuegos son un tipo particular de juego, una de esas actividades más o menos estructuradas que hacemos para divertirnos. Se caracterizan por utilizar aplicaciones software con gran capacidad multimedia y de interacción gracias a los sistemas electrónicos o informáticos sobre los que se implementan. Fueron inventados a mediados del siglo XX, aunque es en la década de los 70 cuando llegaron al gran público y pasaron a formar parte de la llamada Industria del Entretenimiento.

A lo largo de sus más de 30 años de evolución, los videojuegos han ido incorporando las características y capacidades de las nuevas tecnologías como la combinación de varios lenguajes audiovisuales en un mismo soporte, la interactividad, la capacidad para procesar información y la conectividad.

### Game designer

Tienen un rol muy importante en el proceso de creación y deben ser conocedores de las tendencias en videojuegos para poder construir un producto atractivo para los consumidores.

Su labor en común es ser el encargado de definir todos los aspectos del videojuego, desde mecánicas, personajes, historia, mundo del videojuego, etc.

Se trata de esa capacidad que tienen los consumidores de intervenir, de interactuar, e incluso de llegar a alterar la experiencia que se le plantea a través de las mecánicas de juego. Con mecánicas de juego nos referimos a todas las acciones que puedo llevar a cabo. Desde andar, correr, saltar, hasta golpear, disparar, absolutamente todo.

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.

Dentro de estas tareas desglosadas tenemos que:

- Define la mecánica del videojuego

Definir el argumento del videojuego. Qué historia quiere contar y plantear la mecánica que va a seguir el juego para poder establecer, más tarde, todos los otros aspectos.

- Gestiona proyectos con otros departamentos

Llevar las ideas a los departamentos y que el concepto sea entendido por todos los trabajadores inmersos en la creación del mismo.

- Delimita las reglas y niveles

Delimitará las normas del videojuego, es decir, diferenciará lo que sí que está permitido, de lo que no. Además, establecerá los distintos niveles de los que consta el videojuego.

- Crea los personajes

Tendrá el poder de definir el número de jugadores, así como qué rol tiene cada uno. Depende de la temática del videojuego, tendrá que pensar en cómo van a ser los personajes: humanos, animales, figuras mitológicas, etc.

- Define el escenario



Son responsables de decidir cómo va a ser el entorno en el que se va a desarrollar la acción. Siempre ligado al argumento, puede ser un escenario natural, futurista, realista e incluso ser tan original que no tenga un carácter definido.

-Fija las recompensas

Cuando un jugador hace bien una acción dentro del videojuego, suele obtener una recompensa. El Game Designer tendrá que pensar qué tipo de recompensas recibirán los usuarios: cuantificar la recompensa y clasificarlas por grupos, definir en qué momento las recibirá el usuario.

-Elige las habilidades

Deberá pensar qué habilidades tiene cada personaje. Clasificar los diferentes personajes en distintas categorías según las habilidades que presentan.

-Marca los patrones de interacción

Marcará los criterios sobre cómo interactúan los personajes entre sí y cuál es el nivel de interacción que el usuario tiene con el juego.

Storyboard

El storyboard es un concepto que puede aplicarse a distintos usos. Se utiliza, por ejemplo, en cine. Con esta herramienta se representan, en forma de esquema, cómo se va a desarrollar una escena. Ayuda a visualizar ideas y conceptos o cómo se interactúa con un cliente. También se utiliza en publicidad, para ilustrar los anuncios, y es una herramienta muy útil tanto para directores de arte como para diseñadores gráficos.

La creación de un Storyboard para videojuegos es muy similar al que se emplea en otros sectores como publicidad, cine o televisión. Sin embargo, difiere en un elemento principal: A diferencia de otros sectores donde solo hay una línea de acción, normalmente establecida por el director del guion, el diseño de un videojuego requiere que el Storyboard cuente con diferentes líneas de acción, en función de los movimientos del jugador.

Tipos de Storyboard

En el Diseño del juego: Aquí se establece una plantilla principal donde se refleja en las diferentes cuadrículas la estructura base del juego, la vista previa de todos los niveles que conformarán el juego.

En el Diseño de niveles: A partir del storyboard creado anteriormente, creamos un Storyboard para cada uno de los niveles del juego por separado detallando las diferentes acciones, escenas y objetivos del juego.

En el Diseño de escenas, objetos y personajes: Realizamos el Storyboard de los personajes, objetos y entornos del juego para describir cómo interactúan entre sí. El Storyboard de un personaje es muy importante, ya que los jugadores serán los que interactuarán con estos personajes durante el juego.

Tradicional: Pueden ser en blanco y negro, a color, bocetos e incluso ilustraciones más elaboradas, todo en función del artista y las necesidades de la producción.

Digital: Mediante softwares que disponen de herramientas para la creación de Storyboards, software estándar en la industria, que permite planificar la animación 2D y la creación de recursos para juegos.

### 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. 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.

### 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. 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, etc.

### 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.

Varios de los mas conocidos son:

Unity: Es uno de los mejores motores 2D y 3D que encontrarás actualmente, ofreciendo respuesta a las necesidades de los desarrolladores independientes, con licencia gratuita y precios asequibles. Permite exportar a un montón de sistemas y tiene soporte para DirectX 11, mecanim animation, juegos Linux, etc.

GameMaker: Es un motor enfocado al 2D aunque también incluye algunas funcionalidades para el desarrollo en 3D. Creado originalmente por Mark Overmars para ayudar a sus estudiantes allá en los años 90, liberado al público como herramienta para el desarrollo de videojuegos en el 1999.

Construct: Es un motor de desarrollo de juegos 2D que no requiere programación. Permite exportar a HTML5 y a otras plataformas utilizando PhoneGap, AppMobi, etc. Incluye un editor de eventos innovador, fácil de aprender.

Stencyl: Es una herramienta que permite crear juegos en 2D sin saber programación, utilizando un sistema de drag & drop de eventos y acciones. En el pasado estaba enfocado en

la creación de juegos en Flash y para iOS, pero en 2012 expandió su "target" a Android y a HTML5.

Detrás de cada videojuego está el código que lo ejecuta. Los lenguajes para programar videojuegos son una parte integral del desarrollo de este tipo de software.

C++ es el lenguaje de programación que más se usa en el desarrollo de videojuegos triple A.

No obstante, muchos lenguajes de programación se superponen y son compatibles entre sí, cada uno de los cuales contribuye a la experiencia general del juego. Por ese motivo no es raro, que los programadores, en el desarrollo de determinados juegos lleguen a emplear varios de los lenguajes

Android y iOS se han convertido en el mercado más grande para la industria de los videojuegos.

Java

Un juego desarrollado bajo esta tecnología es Minecraft, al menos en sus primeras versiones.

Una razón de la popularidad de Java entre los programadores es que su modelo se puede actualizar constantemente lo que le permite competir con tecnologías más nuevas.

JavaScript

JavaScript nace influenciado por Java y es el lenguaje de programación más utilizado para juegos de navegador y basados en la web.

Se trata de un lenguaje para programar juegos que se usa, sobre todo, para diseño y desarrollo web.

Python

Igual que pasa con C++ y Java, Python también se basa en la programación orientada a objetos, pero, en este caso, con una sintaxis y una ejecución mucho más simple.

Python te permite crear prototipos más rápido, lo que puede ayudar a que el proceso de desarrollo del juego sea más fluido.

Metodologías de desarrollo de videojuegos.

Existen diversas metodologías para el desarrollo de videojuegos, cada una con sus particularidades, propósitos, procesos y condiciones para ser aplicadas correctamente.

Como toda metodología para la creación de algún software, es necesario de varios pasos, tales como:

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 guión gráfico (story board).

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.

#### 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.

#### 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.

#### 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.

#### 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. Ésto se hace sacando parches o actualizaciones al mercado.

#### 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.

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.

Se tienen que tomar en cuenta los siguientes puntos a la hora de llevar a cabo el proceso de interfaz:

1. Entorno

2. Contenido

### 3.Diseño Visual

### 4.Arquitectura de la información

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

Hace referencia a una serie de rutinas de programación que permiten el diseño, la creación y el funcionamiento de un videojuego.

Los componentes que posee un motor de videojuego en la actualidad tienen una gran variedad de características muy diferentes entre sí. En principio, pueden distinguirse cinco grandes áreas:

#### Programa de juego principal

La lógica del videojuego debe ser implementada a través de diversos algoritmos. Es una estructura distinta de cualquier trabajo de renderizado o de sonido.

#### Renderización

La renderización es el proceso en el cual se generan los gráficos 3D por computadora a fin de mostrar en pantalla el aspecto visual del videojuego. Genera gráficos en 3D por varios métodos (como la rasterización gráfica, el trazado de rayos, la partición binaria del espacio, entre otros) y se ocupa de mostrar escenarios, modelos, animaciones, texturas, sombras, iluminaciones y materiales.

#### Audio

El audio de un videojuego se llega a manejar de muchas maneras y esto depende de las capacidades que tenga el motor. Hoy en día los motores de última generación soportan muchos formatos de sonido, siendo los más populares el WAV y el OGG. En algunos casos, los motores de videojuegos pueden exigir configuraciones exactas, aunque el método más conocido es la administración de audio mediante el bucles de música, o bien modificar el tono cuando se trata de voces o efectos de sonido.

#### Motor físico

El motor físico es responsable de emular las leyes de física en forma realista dentro del motor de videojuego.<sup>1</sup> Específicamente, proporciona un conjunto de funciones para simular acciones reales a través de variables como la gravedad, la masa, la fricción, la fuerza, la flexibilidad<sup>18</sup> y las colisiones, actuando sobre los diversos objetos dentro del juego al momento de su ejecución.

#### Inteligencia artificial

La inteligencia artificial es quien provee de estímulo al videojuego. Su elaboración es crítica a la hora de lograr un sistema de juego pulido y que entretenga. Puede tornarse muy compleja y es necesario tener en cuenta ciertas variables, tales como crear comportamientos programados

Transición narrativa y lenguaje visual de videojuegos.

Los videojuegos son otra forma de contar historias con la peculiaridad de que el receptor tiene la posibilidad de tomar las decisiones que determinan el camino del protagonista. La interactividad es muy grande pero el jugador no tiene libertad total para determinar el futuro de su personaje ya que las acciones están marcadas por el objetivo final de la partida.

Un personaje, un objetivo y una serie de obstáculos que superar para conseguirlo. Esta es la fórmula básica y simplificada que debe seguirse a rajatabla para articular una historia con eficiencia, y que tanto cine como videojuegos llevan aplicando desde sus orígenes en la inmensa mayoría de sus productos.

narrativa no se refiere al género narrativo, sino al conjunto de técnicas y al lenguaje comunicativo empleado a la hora de contar una historia.

En el caso del medio digital, la evolución de las herramientas de desarrollo y la potencia de las consolas y ordenadores actuales ha favorecido la sofisticación de los relatos, pasando de propuestas tan sencillas

El videojuego ha convertido al lector o espectador tradicional en un jugador que explora el texto de forma no lineal, y que no sólo debe interpretar una historia y atribuirle una serie de significados, sino que también es necesario que intervenga sobre la misma. Los mundos ficticiales que recrean los videojuegos a partir de la imagen sintética, donde las posibilidades de creación y representación van más allá de lo que se puede reproducir por medio de la imagen real, son cada vez más complejos y ofrecen mayores posibilidades de interacción

Explicar el proceso de desarrollo de videojuego acorde a los elementos de programación

Visual

Ciertamente todos los juegos son diferentes, ya sea por el género, arte, motor de juegos, temáticas, etc., por lo que no todos siguen la misma ruta de desarrollo. Sin embargo, las etapas del proceso de creación de videojuegos son similares para todos.

Planeación

Un videojuego nace de una idea. A partir de este momento inicia la etapa de planeación y los desarrolladores se enfrentan a todo tipo de preguntas: ¿Qué tipo de juego desarrollaremos? ¿Quiénes son los personajes? ¿Será en 2D o 3D? ¿Quién es nuestra audiencia?

Preproducción

En esta etapa, los escritores, diseñadores, artistas, ingenieros, leads de proyecto y otros departamentos trabajan y colaboran entre sí para definir la manera en que darán vida al videojuego.

Producción

Ciertamente la etapa de producción es la más compleja. La mayoría de los recursos, tiempo y trabajo que conlleva desarrollar un videojuego se los lleva la producción. Usualmente, en esta etapa:

Los modelos de personajes se diseñan y renderizan hasta que luzcan como deben.

El diseño de audio crea todos los sonidos del mundo del juego.

Los diseñadores de nivel crean los ambientes de forma que sean atractivos para los jugadores.

Se graba el doblaje con los actores.

Los programadores escriben enormes códigos para darle vida a los elementos del juego.

#### Periodo de prueba

Cada elemento, detalle y mecánica del juego debe someterse al control de calidad antes del lanzamiento. Para que un videojuego esté listo para su versión alpha, primero debe pasar por las manos de testers para que se identifiquen diferentes cuestiones.

#### Pre-lanzamiento

Esta es una de las etapas del desarrollo de videojuegos más estresantes para los grandes estudios y para los creadores independientes, pues tienen que elaborar una estrategia de mercadotecnia para vender su juego.

#### Lanzamiento

Por fin se define una fecha de lanzamiento y el equipo debe trabajar muy duro para pulir todos los detalles que sean necesarios.

Durante esta etapa, los desarrolladores hacen una lista de bugs que tienen que erradicar, de mayor a menor gravedad. Comienzan con aquellos bugs que pueden “crashear” el juego, hasta llegar a aquellos que representan problemas menores.

#### Post-lanzamiento

El juego ya está en manos de los jugadores, pero el trabajo aún no acaba.

Generalmente, en esta etapa temprana post-lanzamiento, los jugadores comienzan a identificar bugs u otros errores dentro del videojuego, por lo que los estudios tienen que solucionar estos problemas.

## REFERENCIAS

Arenas, D. (2019). ¿Qué hace un Game Designer? Platzi. <https://platzi.com/blog/que-hace-un-game-designer/>

U-tad. (marzo 22, 2022). ¿Qué es un game designer?. U-TAD. <https://u-tad.com/que-es-game-designer>

IFP. (junio, 2022). ¿Qué hace un Game Designer? Innovación en Formación Profesional. <https://www.ifp.es/blog/que-hace-un-game-designer>

Gonzalo Frasca's. (2011). Diseño de videojuegos/Introducción a los videojuegos y su historia. Wikiversidad. [https://es.wikiversity.org/wiki/Dise%C3%B1o\\_de\\_videojuegos/Introducci%C3%B3n\\_a\\_los\\_videojuegos\\_y\\_su\\_historia](https://es.wikiversity.org/wiki/Dise%C3%B1o_de_videojuegos/Introducci%C3%B3n_a_los_videojuegos_y_su_historia)

Agustin P. (Enero, 2022). ¿Qué es y cómo crear un storyboard? ESDESIGN. <https://www.esdesignbarcelona.com/actualidad/disenio-grafico/que-es-y-como-crear-un-storyboard>

Costa H. (2018). 10 engines y motores. HEKTOR DOCS. <https://docs.hektorprofe.net/escueladevideojuegos/articulos/engines-motores-recopilacion-programas/>

Tokio School. (abril, 2021) Lenguajes para programar videojuegos. <https://www.tokioschool.com/noticias/lenguajes-programar-videojuegos/>

Lopez D. (2022) Árcade. <https://www.riti.es/ojs2018/inicio/index.php/riti/article/view/388>

Lopez V. (abril 2018) Cine y videojuegos <https://www.espinof.com/otros/cine-y-videojuegos-las-herramientas-tecnicas-y-narrativas-que-diluyen-la-separacion-entre-los-dos-medios>