Materia: Programación Visual

Profesor: Emmanuel Torres Servín

Equipo:

- Erick Alejandro López Pacheco 1321124248
- Christian Uriel Ramírez Pérez 1321124315

Grupo: 4322IS

Carrera: Ingeniería en Software

Introducción.

Dentro de esta investigación se abordarán distintos temas sobre la programación visual y los videojuegos, en los temas que veremos en breve se investigo sobre el funcionamiento de los motores de videojuegos y su importancia para el funcionamiento del videojuego. Al igual que todas las bases para poder crear un video juego, desde la metodología a usar hasta el proceso y la creación de una historia entretenida y coherente para el jugador.

Conceptos de videojuegos.

Un videojuego o igualmente conocido como juego de video es un software qué se creó con el fin del entretenimiento qué se basa principalmente en la integración de 1 o más jugadores, dicho software se puede ejecutar tanto en ordenadores como en dispositivos electrónicos exclusivamente creados para los videojuegos, estos dispositivos son conocidos como consola de videojuego.

Para el funcionamiento de los videojuegos es necesario tener en cuenta los siguientes puntos:

- Software: el juego propiamente dicho que ha sido desarrollado para la diversión

- Controlador: un periférico de entrada qué permitirá realizar las distintas acciones dentro del juego. Dicho controlador varía de diseño y funciones dependiendo de qué dispositivo se utilice, siendo en el caso

del ordenador desde un teclado hasta un Mando USB especialmente diseñado para el juego.

- Dispositivo Electrónico: Este puede ser tanto un Ordenador, como las viejas Máquinas Arcade, tanto como una Consola de Videojuegos o inclusive un Dispositivo Portátil (como ha sido en los últimos años, el avance de los Juegos para Teléfonos Móviles).

Conceptos y tipos de game designer, storyboard.

- **Concepto game designer:**

  Un game designer o diseñador de videojuegos es el encargado de crear la historia del videojuego al igual que sus niveles y todos los elementos que lo compondrán. Un game designer no necesariamente debe ser programador o diseñador gráfico.

- **Tipos**

1. **Producer:**
   Generalmente el diseñador principal. Define las líneas generales del mundo del juego, sabiendo cómo va y a donde. Dueño del mundo, todas las decisiones pasan por él.

2. **ContentDesigne:**
   Encargado del contenido, crea los elementos del mundo, realiza muchas propuestas, tomándose solo las que se necesitan.

3. **GameWriter:**
   Escribe diálogos, guion, cuenta la historia, hace el concepto.

4. **SystemDesigner:**
   Hace las reglas del juego, el sistema, es quien construye el videojuego.

5. **LevelDesigner:**
   Crea el espacio del juego y construye una narrativa por medio del mismo cuerpo del juego.
6. **UsabilityDesigner (diseñador gráfico especialista en feedback):**
   Se encarga de que el sistema y todo tenga una lógica y funcione correctamente, además de asegurarse que el juego se comunique con el usuario (su centro es la comunicación).

- **Concepto Storyboard**

El Storyboard se encarga de la realización de una secuencia de dibujos realizados en una plantilla, dichos dibujos son acompañados de textos breves, todo lo anterior definirá la estructura de la historia. Un proceso imprescindible en la fase de preproducción para el desarrollo de Videojuegos que permite previsualizar el resultado final de la producción.

Un Storyboard puede realizar su trabajo de dos formas:

1. **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.
2. **Digital:** Mediante softwares que disponen de herramientas para la creación de Storyboards un ejemplo seria “Toon Boon Storyboard“ ,software estándar en la industria, que permite planificar la animación 2D y la creación de recursos para juegos.

- **Tipos**
    1. **En el Diseño del juego:** Se utiliza una plantilla por defecto que será la principal, donde se podrán reflejar las diferentes cuadriculas de la estructura base del juego, la vista previa de todos los niveles que conformarán el juego.

    2. **En el Diseño de niveles:** A partir del storyboard creado anteriormente, creamos un Storyboard para cada uno de niveles del juego por separado detallando las diferentes acciones, escenas y objetivos del juego. Su función es separar y distinguir cada nivel.

3. **En el Diseño de escenas, objetos y personajes:** Realizamos el Storyboard de los personajes, objetos y entornos del juego para describir como 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.

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

- **Características de motores de videojuegos.**

Un motor de juego se encarga de asuntos como la gestión de las colisiones y las físicas del juego es una serie de librerías programadas para la correcta representación de menús e interfaz, la reproducción de sonidos, o el manejo de la inteligencia artificial de los enemigos, entre otras muchas otras cosas.

- **Tipos**

1. **Unreal Engine 4**

   Desarrollado por Epic Games, Unreal Engine es un potente motor gráfico y motor de juego que permite el desarrollo de juegos de todo tipo y para prácticamente cualquier plataforma (PC, VR, consolas, smartphones…). Creado inicialmente en 1998 para soportar el juego Unreal, ha evolucionado para ser la herramienta ideal para crear todo tipo de juegos. Algunos de los juegos desarrollados sobre este motor

son Gears of War, Bioshock, Street Fighter V, Dishonored, Life is Strange, Sea of Thieves o el popular Fortnite.

2. **Unity**

   Unity es el principal competidor de Unreal Engine y es igualmente flexible y potente, permitiendo desarrollar para un gran número de plataformas. Con un enfoque destinado a cubrir las necesidades de los desarrolladores independientes, Unity es un motor muy popular en la industria. Juegos desarrollados sobre este motor: Heartstone, Ori and the Blind Forest, Cuphead, Firewatch, Inside o Monument Valley.

3. CryEngine

   Motor de juego desarrollado por la compañía alemana Crytek, usado por primera vez en el juego Farcry. Como Unreal o Unity, estamos frente a una suite de desarrollo completa enfocado a juegos principalmente 3D. Juegos desarrollados sobre este motor: Crysis 3, Ryse, The Climb, Robinson: The Journey, Kingdom Come: Deliverance, o el reboot de Prey de 2017.

- **Lenguajes de videojuegos.**

1. **C#**

   C# es probablemente el lenguaje que cualquier artículo para el desarrollo de videojuegos va a tener. En realidad, la razón más importante para aprender este lenguaje es este grandioso motor: Unity ya que es la herramienta más importante del desarrollo de videojuegos en la actualidad.

2. **Java**

   Antes de que Unity apareciera en el mercado, Java era el lenguaje casi por preferencia para el desarrollo de videojuegos tipo indie (que no eran realizados por grandes compañías).

   Java aprovechaba la multiplataforma a su máxima expresión, si hacemos un poco de memoria los primeros videojuegos para celulares todos eran Java, es decir, que teníamos una máquina virtual y que esta era el puente para que los videojuegos creados con Java funcionaran.

3. **Javascript**

   Dentro de esta lista podríamos pensar que C# o Java no tienen competidores, pero Javascript tiene una ventaja: es para la web. Con los alcances que puede tener ahora HTML5 y con las librerías de gráficos tanto 2D y 3D que han aparecido javascript toma un papel relevante para el desarrollo de videojuegos, en específico si nos interesa que nuestro juego sea fácil de ejecutar y que se funcione en internet.

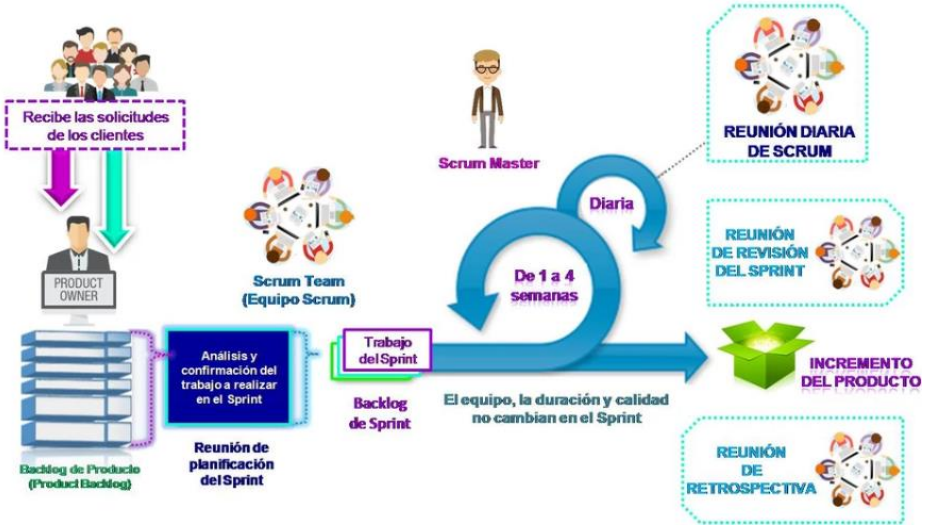
Metodologías de desarrollo de videojuegos.

1. **Programación extrema (XP)**

   Conocida por sus siglas XP (eXtreme Programming), es una metodología basada en un conjunto de reglas y buenas prácticas para el desarrollo de software en ambientes muy cambiantes con requisitos imprecisos, por ende está enfocada en la retroalimentación continua entre el equipo de desarrollo y el cliente.

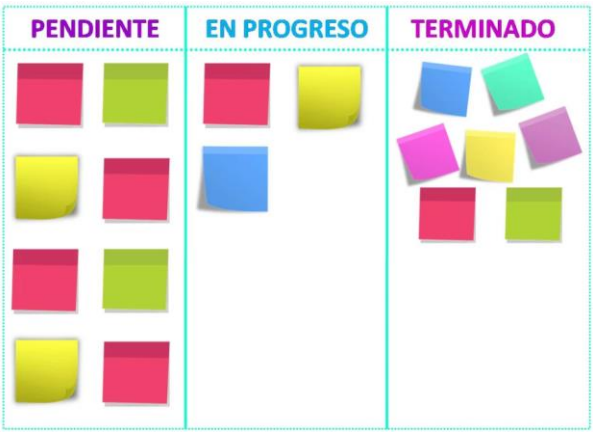
2. **Scrum**

   Esta metodología, es un marco de trabajo de procesos ágiles que trabaja con el ciclo de vida iterativo e incremental, donde se va liberando el producto por pares de forma periódica, aplicando las buenas prácticas de trabajo colaborativo (en equipo), facilitando el hallazgo de soluciones óptimas a los problemas que pueden ir surgiendo en el proceso de desarrollo del proyecto.



3. **Kanban**

   Proveniente de una palabra japonesa cuyo significado es “Tarjeta Visual” es un marco de trabajo que requiere una comunicación en tiempo real sobre la capacidad del equipo, utilizado para controlar el avance de trabajo en una línea de producción, en la cual se clasifican las tareas en sub estatus, esto con la intención de determinar los niveles de productividad en cada fase del proyecto.



.

Desarrollo de prototipos de videojuegos.

Hay tres tipos de prototipado que deberemos hacer para testear en un principioel videojuego, y son estos :

1- Prototipado del Código.

Construir experimentos con las nuevas tecnologías que se vayan a utilizar en el proyecto: engines 3D, herramientas, o incluso consolas o hardware que todavía no se han comercializado.

Frecuentemente los programadores realizan prototipos con tecnologías o lenguajes de programación diferentes a los que van a usar en el juego definitivo.

2- Prototipado del Arte.

El equipo de arte gráfico de un juego necesita al principio afinar con precisión el estilo artístico, estético y técnico de los recursos de 'art' del game. Esto implica tener claro no solamente el aspecto de los personajes escenarios y objetos por separado, sino también la forma en que se integran unos con otros, y los requisitos y limitaciones técnicas con los que se construyen.

3. Prototipo de Diseño.

Aunque la disciplina de diseño puede ser la más abstracta de las tres, centrada inicialmente en la concepción y exploración de ideas, pero no en su implementación, es de hecho la que más herramientas tiene para prototipar esos conceptos, y la que más puede beneficiarse al hacerlo, ya que el trabajo de todo el equipo va a estar basado en las decisiones del diseño.

Alpha

Se podría decir que la versión Alfa de un videojuego es la versión más primitiva del mismo, pues es la primera versión del juego que es funcional. Sin embargo, esta versión de desarrollo suele ser muy inestable, por lo que suele pasarse la versión Alfa a varios probadores de videojuegos, los conocidos como testers.

Beta

Una vez el videojuego es probado y corregido en su versión Alfa, surge la primera versión completa del videojuego: la Beta. Esta es a su vez la última versión de prueba que los desarrolladores pasar a los testers para que detecten los últimos errores antes de sacar la versión definitiva

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

Los moteres de videojuegos se pueden integrar de diferente forma, algunas de esas son:

- API: Conjunto de funciones y procedimientos, que ofrecen ciertas librerías para ser utilizado por otro software como una capa de abstracción.
- API GRÁFICO: Es un API que ofrece prestaciones para desarrollos gráficos digitales a programadores y diseñadores. En el ámbito de los videojuegos, los API gráficos mas conocidos son Direct3D y OpenGL.
- BUFFER: Es un espacio de memoria, en el que se almacenan datos para evitar que el recurso que los requiere ya sea hardware o software, se quede en algún momento sin datos.
- DirectX: Es una colección de APIs creadas para facilitar tareas relacionadas con la programación de juegos en la plataforma Microsoft Windows. Las APIs son: Direct3D, DirectInput, DirectSound, DirectGraphics, DirectMusic, DirectPlay.
- DIRECT3D: El objetivo de esta API es facilitar el manejo y trazado de entidades gráficas elementales, como líneas, polígonos y texturas, en cualquier aplicación

que despliegues gráficos en 3D, así como efectuar de forma transparente transformaciones geométricas sobre dichas entidades.

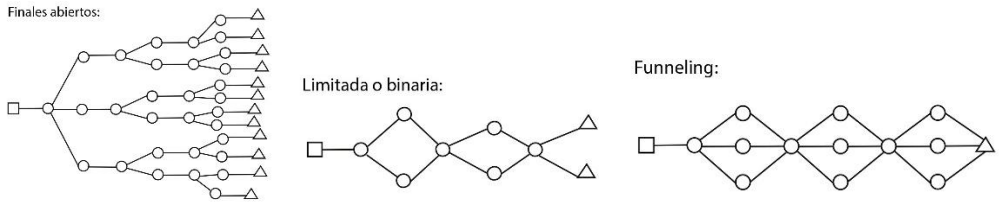
- OPENGL: Es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D.
- PÍXELES: Es la menor unidad en la que se descompone una imagen digital, ya sea una fotografía, un fotograma de vídeo o un gráfico.

Transición narrativa y lenguaje visual de videojuegos.

En los videojuegos es más que evidente el grado de participación del usuario en la creación narrativa, esta construcción se da a medida que se avanza en el juego. Dicha intervención se encuentra vinculada con el grado de interactividad que ofrecen los videojuegos lo que permite que el usuario haga un aporte, quizás ilusorio, con el relato; éste, a su vez, se convierte en aparente cocreador de la historia en la cual se halla sumergido. Un factor importante en esta relación es el papel del jugador cuando se desprende de su pasividad para transformarse en un jugador-lector-narrador. Una extraña hibridación, que no tiene tampoco nada de nuevo, porque los juegos de rol ya mantenían esos rasgos.

- Narrativa lineal: Es una línea recta la cual sigue su objetivo fijamente, esta se caracteriza por no permitir que el usuario tome decisiones para seguir cambiar la historia del videojuego.
- Narrativa ramificada: La narrativa ramificada es una de las más importantes, ya que ofrece caminos distintos al jugador, la historia será modificada de acuerdo a la decisión tomada por el usuario.
- Narrativa en embudo: En esta narrativa la historia se separa en varias opciones, pero todas las opciones terminan siguiendo al final el mismo objetivo o camino, lo que esta narrativa permite explorar otras opciones, pero con el mismo final.
- Narrativa no lineal: con este tipo de narrativa, el orden cronológico no está presente, a su manera. En la narración multinivel, que es característica de los

juegos tipo MMO’s, los bloques narrativos no guardan relación entre sí, pero sí vale para que el jugador aúne el universo y el tema del juego



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

- 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) 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.
- 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

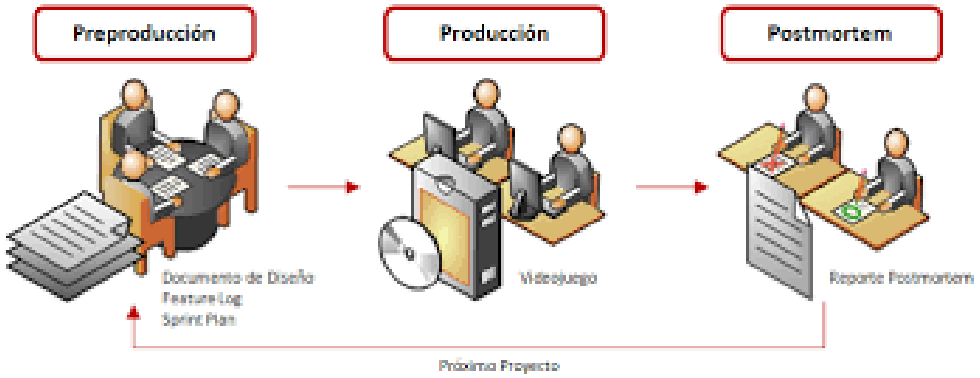
En esta etapa se corrigen los errores del proceso de programación y se mejora la jugabilidad a medida que se prueba el juego.

- 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.



Bibliografías.

D. (2022, 2 mayo). *La función de la geolocalización en tus Apps*. Doonamis. https://www.doonamis.es/la-funcion-de-la-geolocalizacion-en-tus-apps/

A. (2019, 18 junio). *El Storyboard en Diseño de Videojuegos*. Arteneo. https://www.arteneo.com/blog/storyboard-videojuegos-escuela-madrid/#:%7E:text=El%20Storyboard%20no%20es%20otra,la%20estructura%20de%20la%20historia.

Guzman, H. C. (2018, 1 marzo). *Las 7 fases más importantes en el desarrollo de juegos | Escuela de Videojuegos | Hektor Profe*. 7 fases mas importantes.

https://docs.hektorprofe.net/escueladevideojuegos/articulos/fases-del-desarrollo-de-videojuegos/

A. (2019, 18 junio). *El Storyboard en Diseño de Videojuegos*. Arteneo. https://www.arteneo.com/blog/storyboard-videojuegos-escuela-madrid/#:%7E:text=El%20Storyboard%20no%20es%20otra,la%20estructura%20de%20la%20historia.

Fernández, P. P. (2020b, febrero 18). *Motores gráficos y de juego: definición, tipos y modelos de negocio (Tesis) - HyperHype*. HyperHype | Web de videojuegos 100% independiente. https://www.hyperhype.es/motores-graficos-y-de-juego-definicion-tipos-y-modelos-de-negocio/

Mingorance, M. G. (2022, 15 enero). *Los 3 mejores lenguajes de programación para videojuegos*. Game It - Consolas, videojuegos y hardware. Gaming Culture. https://www.gameit.es/los-3-mejores-lenguajes-de-programacion-para-videojuegos/

A. (2019b, junio 18). *El Storyboard en Diseño de Videojuegos*. Arteneo. https://www.arteneo.com/blog/storyboard-videojuegos-escuela-madrid/#:%7E:text=El%20Storyboard%20no%20es%20otra,la%20estructura%20de%20la%20historia.

UNIVERSIDAD POLITÉCNICA DE TECÁMAC

Análisis de la programación visual

Materia: Programación Visual

Profesor: Torres Servin Emmanuel

Alumno: RAMIREZ PEREZ CHRISTIAN URIEL

ERICK ALEJANDRO LOPEZ PACHECO

Matricula: 1321124315

1321124248

Grupo: 4322IS

Carrera: Ingeniería en Software

Conceptos de programación orientada a objetos.

La programación orientada a objetos se basa en el concepto de crear un modelo del problema de destino en sus programas. La programación orientada a objetos disminuye los errores y promociona la reutilización del código. Python es un lenguaje orientado a objetos.

Características y aplicaciones de eventos.

Características

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.

Los eventos en sí pueden ser desde aceptar o rechazar una solicitud de préstamo, denominado evento de alto nivel, hasta que un usuario presione una tecla, esto es un evento de bajo nivel.

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. Además, los servicios se ejecutan por lo general en el trasfondo del sistema operativo.

Eventos

Es una condición que surge durante la ejecución de un programa y que requiere alguna acción por parte del sistema. Cada evento es diferente por naturaleza, algunos requieren que el programa recobre y muestre cierta información, y otros que se inicien algunos cálculos y cambios de estado.

Los eventos incluyen al mouse, al teclado, una interfaz de usuario y las acciones que se deben activar en el programa cuando ocurran. Esto significa que el usuario debe interactuar con un objeto en el programa, como hacer clic en un botón del mouse, usar el teclado para seleccionar un botón, etc.

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.

Por ejemplo, podría ser un botón que cuando el usuario haga clic en él muestre un mensaje y cuando vuelva a hacer clic en ese botón cierre el mensaje.

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.

La actualización de Windows es un ejemplo de tiempo controlado, donde el usuario puede establecer cuándo actualizar o cuándo verificar y descargar la actualización.

Aplicaciones

La **programación** dirigida por **eventos** es lo que llamamos interfaz de usuario, aunque puede emplearse también para desarrollar interfaces entre componentes de Software o módulos del núcleo.

El evento permite al usuario realizar una serie de acciones lógicas para un programa. sobre un determinado componente, elemento que presta un servicio de comunicación cuando se diseñan interfaces, se da inicio a un conjunto de acciones programadas por el usuario para ese evento concreto.

Los eventos incluyen al mouse, al teclado, una interfaz de usuario y las acciones que se deben activar en el programa cuando ocurran, esto quiere decir que el usuario debe interactuar con un objeto en el programa, como hacer clic en un botón del mouse, usar el teclado para seleccionar un botón etc.

Características de componentes y métodos visuales y no visuales.

Un componente es visual cuando tiene una representación gráfica en tiempo de diseño y ejecución (botones, barras de scroll, cuadros de edición, etc.), y se dice no visual en caso contrario (temporizadores, cuadros de diálogo -no visibles en la fase de dieño-, etc).

Todos los componentes visuales tienen un método llamado *Show* para mostrarlos y otro llamado *Hide* para ocultarlos.

Procesos de desarrollo visual en proyectos distribuidos y de escritorio.

Un sistema distribuido es un sistema de software cuyos componentes están separados físicamente y conectados entre sí por una red de computadoras. Dichos componentes interactúan entre ellos para lograr una meta común.
Las tres características principales de un sistema distribuido son:

1. Concurrencia de componentes: Los componentes pueden ejecutar sus acciones de manera concurrente e independiente.
2. No hay un reloj global: Los componentes (nodos) de un sistema distribuido no dependen de un reloj que sincronice o indique las acciones de los distintos nodos.
3. Falla independiente de componentes: La falla de un componente no afecta al resto de los componentes.

ordenadores personales. A diferencia de las aplicaciones web, se instalan directamente sobre el sistema operativo.

La portabilidad de las aplicaciones de escritorio se consigue:

- Con múltiples compiladores.
- Con Lenguajes basados en máquina virtual.

Requerimientos visuales de proyectos distribuidos y de escritorio.

Maqueta de interfaz de usuario: Una versión mas detallada y grafica del wireframe, no solo ayuda a hacerse una idea de como funciona su aplicación, también del aspecto

Herramientas y lenguajes de programación visual.

SCRATCH

El lenguaje de programación visual más popular y utilizado 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 microcontroladores en pasos simples.

APP INVENTOR

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

ARDUBLOCK

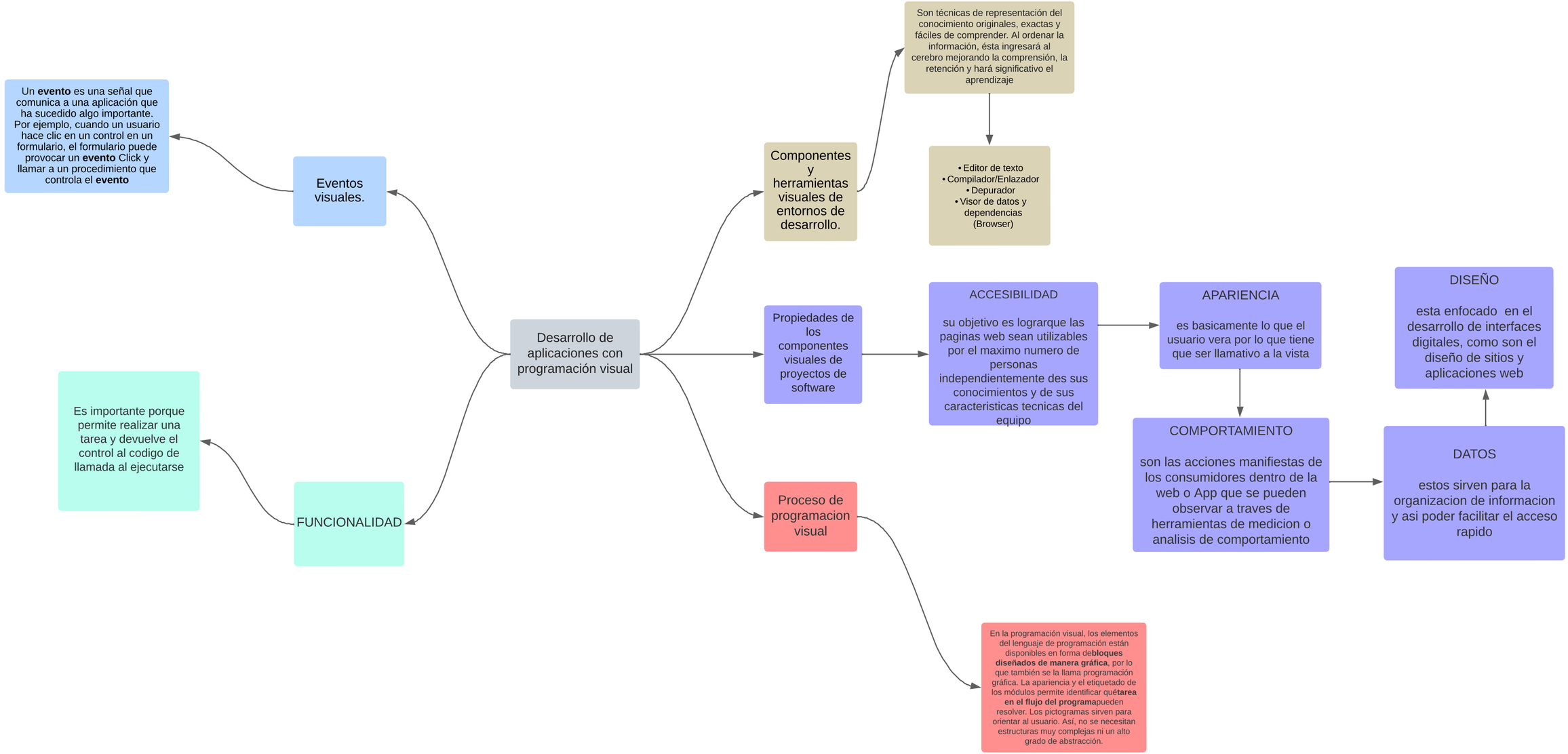
Este lenguaje de programación gráfica está especialmente diseñado para programar el **microcontrolador Arduino** sin introducir texto.

PURE DATA

Este lenguaje de programación visual está orientado tanto a flujos de datos como a los entornos de desarrollo. Pure Data permite producir **software multimedia interactivo**, por ejemplo, para sintetizadores.

LEGO MINDSTORM

Una serie de productos del fabricante de juguetes Lego, cuyo núcleo es la **pieza de Lego programable**: los motores eléctricos, sensores y piezas de tecnología propios de Lego permiten construir y programar robots y otros sistemas interactivos.


Desarrollo de aplicaciones con programación visual
Eventos visuales.
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**
FUNCIONALIDAD
Es importante porque permite realizar una tarea y devuelve el control al codigo de llamada al ejecutarse
Componentes y herramientas visuales de entornos de desarrollo.
Son técnicas de representación del conocimiento originales, exactas y fáciles de comprender. Al ordenar la información, ésta ingresará al cerebro mejorando la comprensión, la retención y hará significativo el aprendizaje
• Editor de texto
• Compilador/Enlazador
• Depurador
• Visor de datos y dependencias (Browser)
Propiedades de los componentes visuales de proyectos de software
ACCESIBILIDAD
su objetivo es lograrque las paginas web sean utilizables por el maximo numero de personas independientemente des sus conocimientos y de sus caracteristicas tecnicas del equipo
APARIENCIA
es basicamente lo que el usuario vera por lo que tiene que ser llamativo a la vista
DISEÑO
esta enfocado en el desarrollo de interfaces digitales, como son el diseño de sitios y aplicaciones web
COMPORTAMIENTO
son las acciones manifiestas de los consumidores dentro de la web o App que se pueden observar a traves de herramientas de medicion o analisis de comportamiento
DATOS
estos sirven para la organizacion de informacion y asi poder facilitar el acceso rapido
Proceso de programacion visual
En la programación visual, los elementos del lenguaje de programación están disponibles en forma de**bloques diseñados de manera gráfica**, por lo que también se la llama programación gráfica. La apariencia y el etiquetado de los módulos permite identificar qué**tarea en el flujo del programa**pueden resolver. Los pictogramas sirven para orientar al usuario. Así, no se necesitan estructuras muy complejas ni un alto grado de abstracción.

DISEÑO DE PROGRAMACION VISUAL

DISEÑO

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

Interactividad entre componentes.

.Una interactividad se genera con los componentes que se presentan en la interfaz como son: botones, listas, etiquetas, etc.

MULTIMEDIA

Combinación de texto, sonido, animación y video

INTERFACES

Conjunto de métodos y propiedades que no tiene ninguna implementación

MAQUETADO

Transformar un diseño gráfico en una interfaz funcional en términos de

Proceso de diseño

Se hace un concepto y se crea una especificacion de un artefacto de SW, pensado para cubrir objetivos.