



ESCUELA TEOLOYUCAN

"Constancia, Unidad y Trabajo"

BACHILLERATO TECNOLÓGICO CON LA CARRERA TÉCNICA EN PROGRAMADOR

RECONOCIMIENTO DE VALIDEZ OFICIAL DE ESTUDIOS, CON NÚMERO DE ACUERDO
DGETI20041010 CON FECHA DEL 22 DE DICIEMBRE DE 1991.

"DIAGNÓSTICO DE LAS CARACTERÍSTICAS DE UNITY, ENGINE Y GAMESMAKER EN LOS SISTEMAS OPERATIVOS ANDROID PARA CELULARES DE GAMA MEDIA"

TESINA

**QUE PARA OBTENER EL TÍTULO DE
TÉCNICO EN PROGRAMADOR**

P R E S E N T A:

**BARRENECHEA LEDEZMA ZAID
CHAVÉZ MEZA MELANI LISSETH
ESPINOZA RAMÍREZ EDUARDO
GARCÍA GARCÍA JUAN CARLOS
JOAQUIN HERNÁNDEZ ALISON ESTEFANIA
MIRANDA FLORES ALING MONTSERRATH
SALINAS RODRÍGUEZ URIEL**

ASESOR DE TESINA:

LIC. SERGIO GONZÁLEZ VARGAS

TEOLOYUCAN, ESTADO DE MÉXICO, JULIO DE 2025.

Dedicatorias y Agradecimientos

Presentación

Capítulo I. Planteamiento del Problema

1.1	Planteamiento del Problema	1
1.1.1	Preguntas de Investigación	3
1.2	Objetivos	4
1.2.1	Objetivo General	4
1.2.2	Objetivos Específicos	4
1.3	Delimitación	5
1.3.1	Delimitación Temporal	5
1.3.2	Delimitación Espacial	5
1.4	Justificación	6
1.5	Hipótesis	7

Capítulo II. Publicidad en Redes Sociales

2.1	Primeras Animaciones con Motores Gráficos.	8
2.2	Primeros Juegos Animados en 2D.	9
2.3	Evolución de la Calidad de Videojuegos 2D.	11
2.4	Primeros Videojuegos Animados en 3D.	11
2.5	Evolución de la Calidad de Videojuegos 3D.	13
2.6	Tipos de Modelos de Motores Gráficos.	15
2.7	Diseño del Motor Gráfico.	16
2.8	Características.	17
2.9	Optimización de Rendimiento.	18
2.10	Herramientas de Desarrollo.	29
2.11	Unity y su Interfaz.	20
2.12	Optimización del Rendimiento general de la Interfaz.	22
2.13	Motor de Renderizado.	23
2.14	Capa de Optimizaciones.	25
2.15	Efectos Visuales y su Integración con la Realidad Virtual.	26

2.16	Motores Gráficos.	28
2.17	Porcentaje de Efectividad.	31
2.18	Costos de Licencias de Motores Gráficos.	33
2.19	Accesibilidad a la Creación de Videojuegos.	34

Capítulo III. Consumismo

3.1	Primeros Videojuegos Hechos para Android.	36
3.2	Evolución de Juegos en Android.	38
3.3	Evolución de Marcas de Celulares Android.	40
3.4	Creación de Celulares Android para Videojuegos.	44
3.5	Versiones y sus Actualizaciones de Android.	45
3.6	Procesadores en los Dispositivos Android.	49
3.7	Factores que Influyen en una Buena Optimización en los Dispositivos Android.	50
3.8	Relación de la Memoria RAM en el Rendimiento de los Dispositivos Android.	51
3.9	Como Influye la Batería en el Rendimiento de los Dispositivos Android.	53
3.10	Memoria RAM.	54
3.11	Características.	55
3.12	Tipo de Memoria.	56
3.13	Android.	57
3.14	Desarrollo de Aplicaciones.	58
3.15	Sobrecalentamiento de Android.	58
3.16	Calidad y Precio.	61
3.17	Preferencia de Usuarios a Android.	63

Capítulo IV. Metodología de la Investigación

4.1	Enfoque y Método	66
4.2	Población y Muestra	67
4.3	Instrumento de Medición	68

Capítulo V. Análisis de Resultados

5.1	Matrices	70
-----	----------	----

5.2	Graficas	76
5.2.1	Encuestas	76
5.2.2	Escala Likert	78
5.2.3	Entrevistas	80

Informe Final

Conclusiones

Recomendaciones

Referencias Bibliográficas

Anexos

Dedicatorias y Agradecimientos

A mis padres:

Quiero agradecer principalmente a dos personas esenciales en mi formación académica: mi padre Manuel Arturo Chávez Pérez por apoyarme a seguir a delante y a no rendirme a mis logros y a mis metas, y a mi madre Erika Yanet Meza Alfaro por apoyarme y aconsejarme a como mejorar como persona, a los dos les doy gracias por esforzarse para poder darme lo mejor y que no me falte nada.

A mis hermanos:

Principalmente a mi hermano mayor Edwin Enríquez Meza, por apoyarme cada vez que lo necesito, por creer en mí en que algún día llegare a cumplir mis objetivos, y a mi hermana Astrid Saidí Meza por ser gran apoyo para mí.

A mis amigos:

A mis amigos les doy las gracias por estar para mí en todo momento, por los momentos divertidos que me regalaron a su lado, por dejarme ser parte de su vida y sobre todo en esta gran etapa que la hacen más feliz con su presencia y darme ánimos.

A mis mascotas:

Quiero agradecerles por ser gran inspiración para mí y porque me levantan el ánimo solo con verlos, son muy importantes para mí.

A mi profesor:

Me gustaría agradecer a mi asesor de tesina, el profesor: Sergio Vargas, por guiarnos y apoyarnos para la tesina.

CHAVÉZ MEZA MELANI LISSETH
Dedicatorias y Agradecimientos

A mis papás:

Quiero agradecer a mis papás por todo el sacrificio que han dedicado a mi educación desde chico hasta ahora lo que soy. Gracias por su tiempo y paciencia en guiarme por el buen camino, ya que son su ayuda no sé qué hubiera pasado y más en esta etapa tan de la vida.

A mis hermanos:

También quiero agradecer a mis hermanos ya que me han servido como ejemplo de vida, me han ayudado bastante ya sea escolar o personal, gracias por no dejarme solo en momentos difíciles y siempre entenderme y no criticarme, por consejos, por regaños soy lo que soy. Han sido parte fundamental en este proceso cada quien me ha apoyado en lo que sabe y así he mejorado en bastantes áreas.

A mis profesores:

Quiero agradecer a los profesores que he tenido en esta etapa de mi vida, de sus enseñanzas y sus consejos que me han sido útil para mi vida cotidiana, en especial a los maestros que me guiaron para hacer esta tesina, gracias a sus regaños se pudo entregar un trabajo ejemplar y casi perfecto, siempre estaré agradecido por lo que aprendí.

A mis amigos:

Por otro lado, quiero hacer un hincapié en mis amigos que hice en el salón de clases y fuera de él, apoyándome en lo que pudieron y más a mis amigos del aula que han estado para mí y trabajando juntos, aprendiendo a trabajar en equipo.

A mis compañeros:

Finalmente quiero agradecer a mis compañeros que me han acompañado en esta etapa, intercambiando ideas y apoyándonos cuando lo necesitamos, gracias por todo lo que aprendí de ellos que me servirán en el futuro.

GARCÍA GARCÍA JUAN CARLOS
Dedicatorias y Agradecimientos

A mis profesores:

Estoy profundamente agradecida por habernos guiado en cada paso

del camino. Su dedicación, paciencia y conocimientos fueron un pilar fundamental para el desarrollo de este trabajo. Gracias por dedicar tiempo a resolver nuestras dudas, corregir nuestros errores y fortalecer todas nuestras habilidades.

A mi familia:

Quienes han sido mi mayor fuente de inspiración y fortaleza durante esta etapa. Gracias por su amor incondicional, por darme siempre palabras de ánimo cuando más las necesitaba, por todo su sacrificio y apoyo que pusieron en mí, al igual que me motivaron a dar lo mejor de mí en cada momento. Ustedes son, sin duda, el motor que impulsa cada uno de mis logros.

A mis amigos:

Quiero agradecerles por haber sido un gran apoyo a lo largo de este camino. Sus palabras de aliento, sus ideas y, sobre todo, su compañía en los momentos de mayor dificultad, este proceso no solo fue un reto académico, sino también una experiencia que compartí con ustedes, y por ello estoy profundamente agradecida.

Finalmente, quiero dedicar un especial reconocimiento a todas aquellas personas que, de manera indirecta, también fueron parte de este logro. A quienes, con un gesto amable, una palabra de aliento o simplemente con su presencia, contribuyeron a que este proceso fuera más llevadero.

Este trabajo no solo representa un esfuerzo académico, sino también la suma de los esfuerzos, enseñanzas y apoyo de todos ustedes. Por ello, dedico este logro no solo a mi crecimiento personal, sino también a todas las personas que hicieron posible su realización.

A todos ustedes, mi más sincero agradecimiento.

MIRANDA FLORES ALING MONTSERRATH
Dedicatorias y Agradecimientos

A mis padres:

Quiero agradecer a mis padres por sus grandes consejos por su gran apoyo hacia a mí que siempre puedo contar con ellos y también por sus grandiosos abrazos y por decirme que está bien y que no, por estar siempre para mí por darme una buena educación y que gracias a ellos he logrado llegar hasta aquí sin ellos no hubiera llegado hasta donde

estoy ahora y a ver logrado mis metas y por eso les agradezco de todo corazón.

A mí perrito:

Quiero agradecer a mi hermoso perrito que a pesar de que no habla me demuestra mucho cariño y amor que se pone muy contento cuando me ve y quiero agradecerle por su cariño y nunca dejarme sola y más cuando lo necesito siempre está ahí para mí cuando no estoy bien siempre estaré agradecida de él aunque no toda la vida esté conmigo.

A mí abuela:

Quiero agradecer a mi abuela por el gran apoyo que me da y por las cosas bonitas que suele decirme y por sus hermosos abrazos y que siempre me cuide de las malas personas que siga adelante en cualquier circunstancia que ella nunca me va a dejar sola siempre va a estar para mí apoyándome cuando más la necesite y siempre se lo agradeceré.

A mí amiga:

Quiero agradecer a mi amiga por siempre estar ahí conmigo por darme buenos consejos por ser mi gran consejera por qué me da mucho apoyo y me dice que es lo que está bien y lo que está mal siempre se lo voy agradecer a mi gran amiga.

A mí familia:

Quiero agradecer a toda mi familia por qué siempre me han apoyado y que siempre me dicen que yo puedo con todo y que voy a llegar muy lejos y que no me importe lo que me digan que siga adelante y sé que puedo sin su apoyo yo no llegaría hasta aquí donde estoy ahora.

JOAQUIN HERNÁNDEZ ALISON ESTEFANIA

Dedicatorias y Agradecimientos

A mí:

Por nunca darme por vencido y mantenerme persistente todo el tiempo hasta cuando las cosas no van como se esperan y recordarme lo bueno que soy para hacer cosas mucho más grandes de lo que creo y se ven, también a enseñarme y darme cuenta de cosas más allá de lo que se ven y quien soy y hacia donde me dirijo.

A mis Padres:

Por apoyarme dándome aliento, motivación y guiarme en mi camino y ver cosas más allá de mí y gracias a ellos estoy donde estoy y hacen el sacrificio por mantenerme darme una buena educación.

A mis abuelos:

A ellos por haberme apoyado, motivado y aconsejando en cada cosa que hago para ser mejor cada vez y cumplir mis objetivos exitosamente a pesar de todo.

A mis amigos más cercanos:

por nunca dejarme abajo y haberme apoyado y motivado cuando lo necesite siempre siendo firmes y ayudándonos mutuamente sin ninguna diferencia y cuando se necesiten ahí estamos firmes para ayudar en lo que se pueda y se necesite.

A mis Compañeros:

Por apoyarnos y comprometernos en este proyecto y estar dispuestos a hacer este proyecto, también por el buen ambiente generado.

SALINAS RODRÍGUEZ URIEL

Dedicatorias y Agradecimientos

A mis padres:

Por haberme dado la oportunidad de haber podido pagarme la escuela y darme un mejor sistema, gracias a ellos pude seguir adelante y sin su apoyo no hubiese podido de la misma manera, siempre estaré agradecido.

A mis amigos:

Sinceramente ellos siempre han estado conmigo y a pesar de que no siempre lo están ellos me dan todo si apoyo, Mis amigos de la escuela y los externos siempre los voy a querer ya que siempre me han apoyado en todo y soy afortunado de tenerlos, me siento agradecido de tener amistades tan increíbles a mi lado y espero seguir con ellas y también formar más en el futuro.

A Everardo:

Realmente jamás creí que escribiera esto de un profesor ya que jamás me lleve también con uno, pero si hay algún profesor que le deba es a Everardo, Sus clases y su forma de ser profesor me enseñaron mucho, Las clases con El Combobox sin dudas fueron las que más disfruté, Gracias profesor Everardo, por ayudarme a querer terminar mi carrera.

ESPINOZA RAMÍREZ EDUARDO

Presentación

Presentación: Tiene como objetivo principal diagnosticar las características de estos tres motores gráficos en dispositivos Android de gama media, evaluando su rendimiento, accesibilidad y costo. La investigación se llevó a cabo en la Universidad Bancaria de México, utilizando un enfoque cuantitativo y exploratorio con una muestra de 50 alumnos de programación. Los resultados revelan que Unity es una de las plataformas más populares y accesibles, aunque Engine y GameMaker también son opciones viables.

La optimización de videojuegos para dispositivos de gama media es crucial para asegurar una buena experiencia de usuario, y la elección del motor gráfico debe basarse en las necesidades específicas del proyecto y los recursos disponibles. El presente estudio surge del interés porque: se debe a la necesidad de identificar cual de estos motores gráficos ofrece un mejor rendimiento,

accesibilidad y costos para los desarrolladores que trabajan con Android de gama media.

En la actualidad: los desarrolladores buscan motores gráficos que ofrezcan un equilibrio entre rendimiento y costos especialmente en para dispositivos de gama media que tienen limitaciones de hardware. La metodología de la investigación se basó en un enfoque cuantitativo y exploratorio, utilizando la recolección de datos para probar una hipótesis mediante medición numérica y análisis estadístico.

La población objetivo incluyó 200 alumnos de programación, con una muestra intencional de 50 alumnos seleccionados por su conocimiento en motores gráficos. Los instrumentos de medición utilizados fueron encuestas, entrevistas y una escala Likert para recopilar datos sobre las características y preferencias de los motores gráficos en el desarrollo de videojuegos para dispositivos Android de gama media.

El objetivo principal para desarrollar el presente trabajo es: es identificar y buscar cuál de estos motores gráficos ofrecen mejorar rendimiento y accesibilidad, costos para los desarrolladores, además se pretende proporcionar una guía útil para estudiantes y desarrolladores que desean optimizar sus proyectos en estas dispositivos y sistemas operativos considerando las limitaciones de hardware y las características específicas de cada motor gráfico.

Capítulo I: Se plantean preguntas de investigación sobre sus características y se establecen objetivos de evaluar y comparar su eficiencia, la investigación se delimita temporalmente hasta el 6 de febrero al 12 de julio de 2024 y se realizara en la universidad bancaria de México lo justificación destaca la importancia de optimizar proyectos y la hipótesis sugiere que el uso de Unity permite un ahorro en Android.

Capítulo II: Explora la evolución y características de estos motores gráficos, destacando su impacto en la publicidad y desarrollo de videojuegos. Se abordan temas como las primeras animaciones, la evolución de los videojuegos en 2D y 3D, los tipos de modelos de motores gráficos, el diseño y optimización de rendimiento, las herramientas de desarrollo, la interfaz de Unity, y los costos de licencias. Además, se analiza la integración de efectos visuales y la

realidad virtual, proporcionando una visión completa de los motores gráficos.

Capítulo III: Analiza la evolución de los videojuegos en Android, desde los primeros juegos móviles hasta los más avanzados, y la evolución de las marcas de celulares Android. También se aborda la creación de celulares específicos para videojuegos, las versiones y actualizaciones de Android, los procesadores en dispositivos Android, factores que influyen en la optimización de estos dispositivos, la relación de la memoria RAM con el rendimiento, el impacto de la batería en el rendimiento, y la preferencia de los usuarios por Android debido a su flexibilidad.

Capítulo IV: Se centra en la metodología de la investigación, utilizando un enfoque cuantitativo y exploratorio. La población objetivo incluye 200 alumnos de programación, con una muestra intencional de 50 alumnos seleccionados por su conocimiento en motores gráficos. Los instrumentos de medición utilizados fueron encuestas, entrevistas y una escala Likert para recopilar datos sobre las características y preferencias de los motores gráficos en el desarrollo de videojuegos para dispositivos Android de gama media.

Capítulo V: Presenta el análisis de resultados obtenidos a través de encuestas, entrevistas y una escala Likert aplicadas a 200 alumnos de programación. Los datos recopilados revelan el conocimiento y las percepciones de los estudiantes sobre los motores gráficos y su uso en videojuegos.

Las gráficas y matrices muestran que la mayoría de los encuestados están familiarizados con Unity, pero hay una falta de conocimiento general sobre las diferencias entre los motores gráficos. Los resultados también indican que los estudiantes valoran el rendimiento y la capacidad de almacenamiento al elegir sus dispositivos Android.

Capítulo VI: Presenta el informe final, que incluye las conclusiones, recomendaciones, referencias bibliográficas y anexos. Las conclusiones destacan la importancia de los motores gráficos en el desarrollo de videojuegos y la necesidad de optimizar estos motores para dispositivos de gama media.

Las recomendaciones sugieren elegir el motor gráfico más accesible y adecuado según los objetivos del proyecto, tiempo, dedicación y recursos disponibles. El capítulo también incluye una lista de referencias utilizadas en la investigación y anexos con información adicional relevante.

CAPÍTULO I

Planteamiento del Problema

Capítulo I. Planteamiento del Problema

1.1 Planteamiento del Problema

La presente investigación tiene como finalidad analizar el diagnóstico de las características de Unity, Engine y Gamemaker en los sistemas operativos Android para celulares de gama media. En este caso, se eligió los tres motores gráficos más reconocidos, esto debido a que son los más actualizados hoy en día. Por lo tanto, se investigará estos tres motores gráficos, los puntos a resaltar es su facilidad al desarrollador al crear un videojuego para el sistema operativo Android, pero que a su vez sea el más accesible y económico.

Dicho estudio se llevará a cabo en las Universidad Bancaria de México ubicada en el municipio de Teoloyucan al Norte del Estado de México. Contando con una buena reputación llevando más de 10 años impartiendo clases, teniendo 3,858 matriculados, de las cuales 43.2% fueron hombres (1,667) y 56.8% fueron mujeres (2,191). De estos alumnos la investigación se enfocará en la ingeniería en sistemas operativos.

La presente investigación estará dirigida principalmente a los alumnos que cursan la carrera de programación de la Universidad Bancaria de México entre los dieciocho a veinticinco años de edad y para aquellos que se dedican al desarrollo de videojuegos, así mismo, se espera que pueda ser de interés para otros estudiantes relacionados al ámbito de la tecnología ya que ofrece una guía útil para identificar el mejor motor gráfico para desarrollar videojuegos en sistemas operativos Android.

Según datos de Unity Technologies, más de 3.6 mil millones de descargas mensuales de aplicaciones se realizan en dispositivos que ejecutan contenido creado con Unity. Además, una encuesta de 2021 indicó que el 61% de los desarrolladores de juegos utilizan Unity como su motor principal.

Por otro lado, Unreal Engine es otro motor de videojuegos, en 2018, Epic Games informó que más de 7 millones de desarrolladores utilizaban Unreal Engine. Desde entonces, su popularidad ha seguido creciendo, especialmente con el lanzamiento de Unreal Engine 5, que ofrece herramientas avanzadas para la creación de contenido 3D en tiempo real.

Por último, GameMaker ha experimentado un notable crecimiento en su base de usuarios, Según un artículo de noviembre de 2023, desde su integración con Opera, GameMaker ha triplicado su número de usuarios activos. Además, se registró un incremento del 63% en nuevos usuarios de entre 13 y 17 años.

Con referencia a Unity es una de las plataformas de desarrollo de videojuegos más popular y ampliamente utilizando en la industria lo que lo hace que sea un tema relevante debido a la demanda de profesionales capacitados en su uso. Por lo tanto, Unity ofrece una interfaz intuitiva y herramientas accesibles

Permitiendo a los desarrolladores y crear videojuegos y experiencias interactivas sin necesidad de conocimientos avanzados de programación, lo que lo hace atractivo para proyectos de investigación y desarrollo de tal manera que Unity es compatible con múltiples plataformas lo cual permite desarrollar videojuegos para PC, consolas, realidad virtual y sistemas operativos móviles.

La consecuencia directa de estas limitaciones es que los dispositivos de gama media pueden experimentar un rendimiento subóptimo al ejecutar aplicaciones y juegos desarrollados con motores gráficos avanzados. Esto se traduce en posibles retrasos, caídas de fotogramas y una experiencia de usuarios menos fluida. Los desarrolladores deben optimizar aplicaciones de hardware, utilizando técnicas como la reducción de los recursos para garantizar un rendimiento aceptable en dispositivos gama media.

A diferencia de antes, donde se les cobraban a los desarrolladores de los videojuegos la licencia de los motores gráficos a partir de la rentabilidad de su proyecto a lo largo de un periodo de prueba el cuál varía de la empresa la cual desarrolle el motor gráfico de su preferencia. Sin embargo, evaluaremos los mejores motores gráficos que sean más accesibles para el desarrollador.

Para la posible solución del tema lo primero seria encuestar a los alumnos que estudian ingeniería técnica de sus preferencias o su pensamiento sobre los motores gráficos Engine, Unity y GameMaker, esto para identificar las problemáticas, ventajas y desventajas que el público llega a encontrar al desarrollar videojuegos para celulares Android de gama media.

1.1.1 Preguntas de Investigación

Tomando en cuenta lo anterior, el trabajo se desenvuelve en el diagnostico de los motores gráficos Unity, Engine y GameMaker para un alumno de ingeniería técnica, esta problemática puede afectar el rendimiento y la capacidad de los desarrolladoras nuevos para realizar videojuegos en los dispositivos Android de gama media que ocupan requisitos específicos para que el dispositivo pueda fluir adecuadamente el videojuego, en consecuencia, la presente investigación busca resolver las siguientes cuestiones:

- ¿Cuáles son las características de Engine, Unity y GameMaker?
- ¿Cuáles son las características de los sistemas Android?
- ¿Cuáles son las características de Unity, Engine y GameMaker en los sistemas operativos Android para celulares de gama media?

1.2 Objetivos

En toda investigación se fijan metas, y para la presente investigación se trabajará con los siguientes objetivos:

1.2.1 Objetivo General

El objetivo general de investigación es diagnosticar las características de Unity, Engine y GameMaker en los sistemas operativos Android para celulares de gama media.

A su vez este estudio cuenta con tres objetivos específicos.

1.2.2 Objetivos Específicos

- Investigar las características de Unity, Engine, GameMaker.
- Comparar se eficiencia en los sistemas operativos Android.
- Identificar las características de Unity, Engine, GameMaker a su beneficio en los sistemas operativos Android.

1.3 Delimitaciones

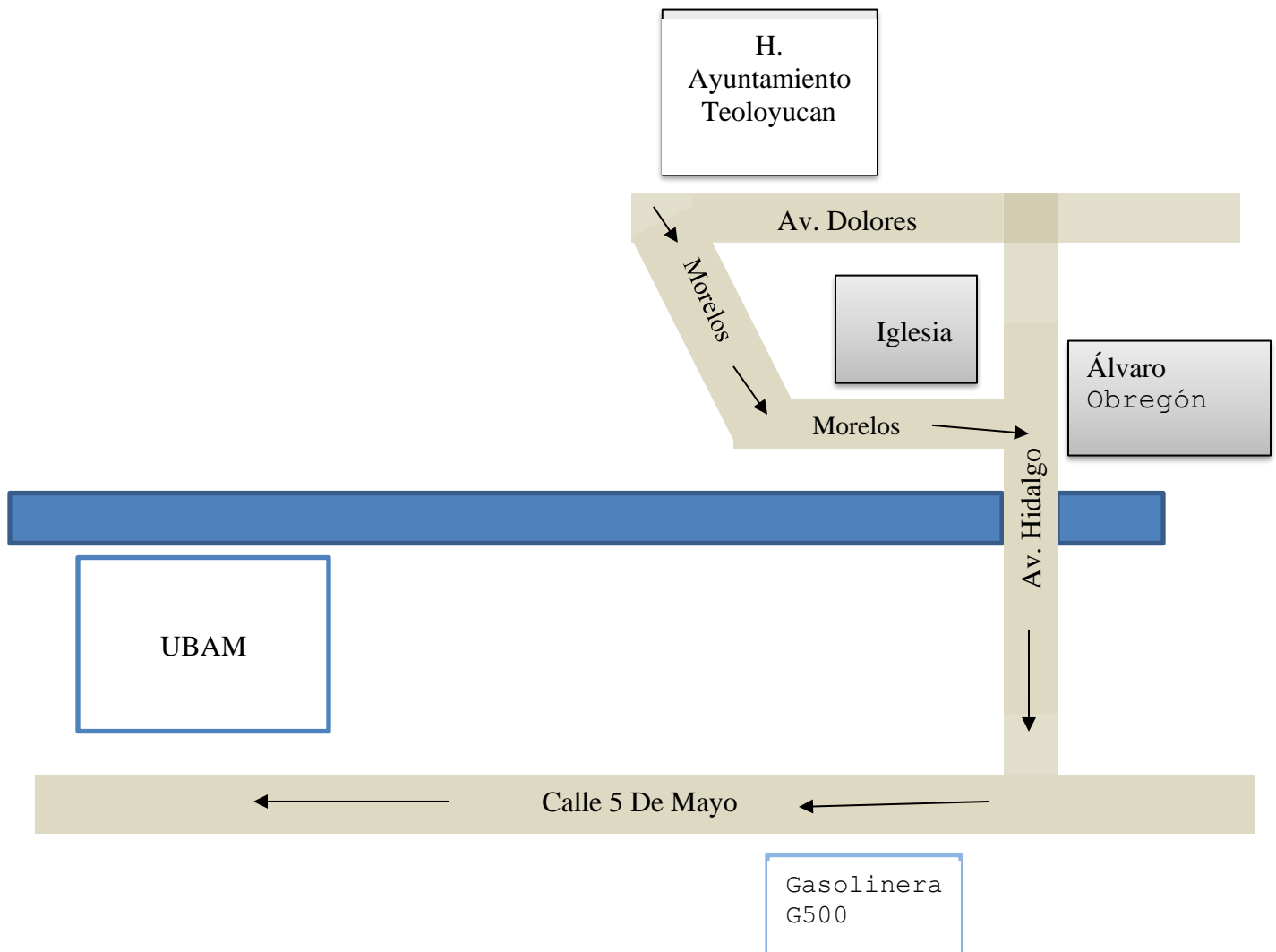
Es muy importante diagnosticar las características de Unity, Engine y GameMaker en los sistemas operativos Android para celulares de gama media y para ello se presentan los siguientes límites.

1.3.1 Delimitación Temporal

Se pretende investigar y observar al fenómeno por un periodo del 6 de Febrero de 2024 al 12 de Julio del 2024.

1.3.2 Delimitación Espacial

La presente investigación se estará efectuando en la Universidad de México de Teoloyucan, Estado de México que cuenta con un total de 3,458 jóvenes y solo se tomara en cuenta 200 que estudian ingeniería en sistemas, se observará con base a la sociedad del lugar ya que se puede obtener una mayor cantidad de resultados sobre el tema.



1.4 Justificación

Es importante hablar de las características de Unity, Engine, GameMaker para comprender cuál de estas tres opciones es la mejor hablando de rendimiento, accesibilidad y facilite a los desarrolladores en sus proyectos de videojuegos y que puedan entregar un videojuego de una excelente calidad y buena jugabilidad en los sistemas operativos Android para celulares de gama media.

De igual forma es importante hablar del tema debido a que la mayoría de las personas que inician en el mundo de la programación de videojuegos no se les da información tanto en la carrera de estudios y las mismas compañías, sin embargo, los mismos desarrolladores no se dan la tarea de explorar la amplia variedad de opciones y los beneficios que cada motor gráfico ofrece al usar su servicio.

Debido a esto, la nueva generación de programadores de videojuegos cada vez es menor, los que siguen adelante con su carrera son pocos y varios deciden comprar o hacer de 0 su propio motor gráfico y esto a consecuencia se queden estancados por la dificultad que resulta no tener un buen motor gráfico con las mejores herramientas y buena accesibilidad, esto los lleva a dedicarse a otra profesión.

Por ello, la necesidad de optimizar cuidadosamente los videojuegos para adaptarse a las restricciones de hardware de dispositivos de gama media suele ser una problemática común, ya que al no lograr una optimización adecuada puede conducir a una ejecución deficiente y a tiempos de carga prolongados, al lanzar el videojuego con estos errores su videojuego no llega a los objetivos esperados y no recauda el dinero requerido.

Si bien es un tema que no representa una gran crisis, es importante que se dé a conocer esto debido a que en un par de años se volverá más grande a la falta de apoyo e información a los nuevos desarrolladores que surgen con grandes ideas que no son vistas y quedan abandonadas, esto puede llegar a que en un futuro el mundo de los videojuegos no avance más, cosa que actualmente se ve en el catálogo de videojuegos para Android.

1.5 Hipótesis

En esta investigación se usarán los factores que son una suposición que permite establecer la hipótesis para mostrar relaciones entre hechos. El valor de la hipótesis reside en una capacidad para establecer esas relaciones entre los hechos de esa manera explicarnos porque se produce. La hipótesis para utilizar será la causal que permite explicar y predecir, con cierto margen de error en los procesos sociales. De tal forma se presenta la hipótesis a comprobar:

“Tenemos un gran porcentaje de proyectos abandonados porque los desarrolladores no saben las características y beneficios de los motores gráficos”

V.I: Tenemos un gran porcentaje de proyectos abandonados

V.D: Los desarrolladores no saben las características y beneficios de los motores gráficos

CAPÍTULO II

Características de Unity, Engine y GameMaker

Capítulo II. Características de Unity, Engine y GameMaker

2.1 Primeras Animaciones con Motores Gráficos

En 1972, los estudiantes de informática Edwin Catmull y Fred Parke digitalizaron una mano humana para su cortometraje A “Computer Generated Hand”. Dibujaron 350 polígonos y triángulos en una maqueta de una mano antes de introducir los datos en un ordenador para crear una mano realista en movimiento por primera vez en la historia. El resultado es uno de los primeros ejemplos de animación por ordenador. (Adobe, 2022, párr. 11)

Para ahondar en el tema, en 1972, estudiantes de informática realizaron un trabajo en el cual animaron una mano humana A Computer Generated Hand. Durante el desarrollo ocuparon figuras geométricas en una maqueta, acto seguido se introdujeron los datos en un ordenador, como consecuencia se obtuvo un ejemplo de animación por ordenador, un cortometraje digitalizado.

El primer motor gráfico en 3D del que se tiene conocimiento fue Freescape Engine, desarrollado por Incentive Software en 1986. Hoy en día, adquirir un Engine varía mucho dependiendo de su origen: pueden ser de shareware (código abierto) o pago, manejando todo tipo de valores. (Marín, 2020, párr.02)

Sin embargo, el Freescape Engine, desarrollado por Incentive Software en 1986, fue el primer motor gráfico 3D registrado. En la actualidad, existen motores gráficos de distintos tipos, desde opciones gratuitas de código abierto hasta versiones comerciales. Los costos pueden variar ampliamente según las características y licencias. Esto permite a los desarrolladores elegir según sus necesidades y presupuesto.

Freescape, desarrollado entre mediados y finales de los 80, por Incentive Software en un Amstrad CPC, y posteriormente portado a plataformas tanto de 8 bits (ZX Spectrum, Commodore 64); como de 16 bits (Atari ST, Amiga, PC-CGA y PC-EGA), es considerado el primer motor de 3D para juegos de ordenadores domésticos. Aunque a que podía hacer para momentos de divulgación para el público del Laboratorio Nacional principios del desarrollo, se consideraba una tarea imposible, o al menos llena de retos, los resultados del primer título que lo usaba, Driller (1987), fueron realmente buenos. (Gutiérrez, 2011, párr.2)

Por otra parte, a mediados y finales de los 80, por Incentive Software en un Amstrad CPC, portado a plataformas de 8 bits, como de 16 bits, considerado por muchos el primer motor de 3D para juegos de computadores de hogares. Aunque en un inicio de su desarrollo se consideró una tarea difícil, el producto de esto fue el primer título llamado Driller (1987), fue muy bien recibido al público.

Después, se puso a la venta 3D Construction Kit (1991), que ponía en manos de los usuarios la potencia de Freescape, para poder crear sus escenarios virtuales 3D, y desarrollar incluso juegos. Un años después, aparecía 3D Construction Kit II (1992), y de este modo terminaba la saga de Freescape. (Gutiérrez, 2011, párr.9)

Por lo tanto, en 1991 se comercializó 3D Construction Kit, una herramienta que permitía a los usuarios acceder a la tecnología del motor Freescape para crear sus propios escenarios tridimensionales y desarrollar juegos. Este lanzamiento democratizó el desarrollo en 3D, brindando nuevas posibilidades creativas. Posteriormente, en 1992, apareció 3D Construction Kit II, una versión mejorada del software. Con esta última entrega, se puso fin a la saga de Freescape, marcando un hito en la historia de los motores gráficos tridimensionales.

2.2 Primeros Juegos Animados en 2D

En 1958, el físico americano William Higinbotham estaba preguntándose qué podía hacer para llamar la atención del público las jornadas de puertas abiertas del Laboratorio Nacional de Brookhaven. El laboratorio neoyorkino, que sigue funcionando en nuestros días, fue originalmente un centro de investigación atómica y ahora pertenece al departamento de energía de los Estados Unidos. Era normal que ciertos organismos estatales tuvieran jornadas de divulgación de cara al público para que familias, colegios y particulares pudieran familiarizarse con lo que estaban pagando con sus impuestos (y, con suerte, encender la llave de la vocación en las jóvenes mentes). (Serrano, 2023, párr.2)

Para todo esto, en 1958, William Higinbotham buscaba captar la atención en las jornadas de puertas abiertas del Laboratorio Nacional de Brookhaven. Este centro neoyorkino, hoy parte del Departamento de Energía de EE. UU., nació como una instalación de investigación atómica. Las

jornadas permitían al público conocer sus actividades y, con suerte, despertar vocaciones científicas.

Para ello se fijó en su avanzada Donner Model 30, una computadora “portátil” aquí significa que se podía mover razonablemente de un lado para otro, no como los ordenadores habituales de la época como el IBM 305, que ocupaba una pared grande de una habitación-. Era un dispositivo de uso científico y entre sus capacidades estaba la posibilidad de calcular trayectorias de proyectiles bajo la influencia del aire, algo necesario para establecer una defensa antimisiles efectiva. Así que se decidió a realizar una demostración de estas capacidades, pero en un contexto menos funesto que la guerra fría y el fin del mundo por ataque nuclear. En dos horas realizó un programa que simulaba la trayectoria de una bola sobre una red y durante tres semanas, junto al técnico Robert V. Dvorak, adaptó dos mandos con una rueda y un botón, y lo conectó todo a un osciloscopio, bautizándolo como “*Computer Tennis*”, o también “*Tennis for Two*”. (Serrano, 2023, párr.3)

Por consecuencia, la Donner Model 3D (Computadora portátil) era un dispositivo de solo uso científico y portátil. Así que se puso manos a la obra y comenzó con demostraciones de esto, bajo un contexto menos funesto de las guerras ocasionadas por aquellos años. En dos horas hizo un programa de una bola atravesando sobre una red y en tres semanas junto al técnico Robert V. Dvorak, adapto dos mandos con rueda y un botón y conectado a un osciloscopio, llamándolo así *Computer Tennis*, o también *Tennis for Two*.

Desde fines del 50 e inicios del 60, se comenzó a trabajar en videojuegos, ofreciéndonos una jugabilidad nunca antes vista con roles de competencia multijugador o para batir récords en puntaje. La evolución da puntadas con Pong (1972) y Space Invaders (1978), quienes empezaron incluyeron algunas imágenes a color gracias al papel celofán. (Redacción RPP, 2018, párr. 5)

Como resultado, se comenzó a producir videojuegos con una jugabilidad distinta, nunca antes vista llegando con roles de competencia de multijugador. Ejemplo de esto esto la evolución da puntadas con Pong (1972) y Space Invaders (1978), quienes empezaron con imágenes a color gracias al papel celofan, proporcionando una revolución a la industria de los videojuegos en los años 50 y 60.

2.3 Evolución de la Calidad de Videojuegos 2D

Donkey Kong fue el primer juego en donde se introdujo este concepto de utilizar saltos para progresar a través del nivel es por eso que se considera como el primer videojuego de plataformas y sería la primera vez en que los jugadores tendrían una pequeña visión del género. Un año más tarde, Pitfall (1982) es lanzado en la consola Atari 2600 en donde se utiliza el mismo concepto de saltos esta vez en un avance horizontal. (Moena, 2019, p. 2)

A partir de esto, el primer juego donde se introdujo el concepto de saltos para el progreso de los niveles fue Donkey Kong es por esto que es considerado como el primer videojuego de plataformas y con esto los jugadores probarían por primera vez una pequeña visión del género. Al año siguiente salió Pitfall (1982) lanzado en la consola Atari 2600 el cual uso el mismo concepto, pero esta vez con un avance horizontal.

Los juegos de lucha 2D son una parte muy importante de la historia de los videojuegos en general. En este género encontramos títulos tan queridos y populares como Street Fighter o The King of Fighters, y fue sin duda uno de los grandes motores dentro de los salones recreativos de la década de los noventa. (Ros, 2017, párr.1)

Asimismo, los juegos de lucha 2D han ocupado un lugar destacado en la historia de los videojuegos, dejando una huella imborrable en la industria. Títulos icónicos como Street Fighter y The King of Fighters se ganaron el cariño y la popularidad de millones de jugadores. Durante la década de los noventa, este género fue uno de los pilares de los salones recreativos, convirtiéndose en una de las experiencias más buscadas por los entusiastas de los videojuegos .

2.4 Primeros Videojuegos Animados en 3D

Para los 90, el 2D (largo y ancho) daría el salto definitivo en una segunda época maestra: el 3D. Los personajes nacidos en la década de los 80 lograron con esta tecnología un ambiente de profundidad único e irrepetible. La consola dónde se vieron mejor estos cambios fue el Nintendo 64, dónde presenciamos como Mario Bros, Pokémon o The Legend of Zelda disfrutaban de la tridimensionalidad. (Redacción RPP, 2018, párr. 8)

No obstante, en los años 90 marcaron un gran paso a la industria de los videojuegos ya que llegó la era del 3D. Los personajes nacidos en los 80 logran con esta tecnología un ambiente único e irreplicable. Todo esto se ve en la consola Nintendo 64, donde se puede disfrutar estas animaciones en Mario Bros, Pokemon o The Legend of Zelda con una profundidad tridimensionalidad.

El resultado fue tener una generación de juegos con un look muy específico. Los primeros años de juegos en 3D supusieron para muchos una vuelta atrás en lo artístico con tal de justificar lo tecnológico, ya que la industria pasó de un estilo pixel-art colorido y detallado en 16 bits (estilo que se sigue usando actualmente) a popularizar un estilo gráfico aún por pulir, con modelados ásperos y puntiagudos debido a las limitaciones de memoria de la época. (Rodríguez, 2018, párr.6)

Por otra parte, la transición hacia los primeros juegos en 3D generó una estética muy particular. Para muchos, este cambio significó un retroceso artístico, ya que se dejó atrás el detallado y colorido pixel-art de los 16 bits, un estilo que aún perdura en la actualidad. Con el objetivo de aprovechar las nuevas tecnologías, la industria adoptó gráficos con modelados ásperos y angulosos, consecuencia de las limitaciones de memoria y capacidad de procesamiento propias de la época.

El año 1990 se erige como un hito en la cronología del diseño de videojuegos, al desplegar el uso de texturas para revestir formas tridimensionales, una innovación que transformó la estética lúdica. *Wolfenstein 3D*, una creación pionera de id Software lanzada en 1992, se adelantó a su tiempo al adoptar esta técnica, otorgando a los objetos virtuales una apariencia de materiales tangibles. Este salto tecnológico inauguró una era de realismo sin precedentes en el entretenimiento interactivo, sentando las bases para los entornos virtuales ricos en detalles y profundidad que constituyen el estándar actual. (Venegas, 2024, párr.8)

Por esta razón, en los 90 se hace un salto sin precedencias en el diseño de videojuegos, al usar texturas para revestir formas tridimensionales, que formó la estética lúdica. *Wolfenstein 3D* creación de id Software en 1992, otorgando objetos virtuales en ricos de detalles con materiales tangibles. Conteniendo realismo sentando las bases y profundidad que contribuyeron al estándar de hoy en día.

El primer videojuego con gráficos 3D, aunque simple en sus estándares actuales, representó un

salto gigante en la evolución de los videojuegos. Este hito abrió las puertas a un mundo de posibilidades, inspirando a generaciones de desarrolladores a crear experiencias de juego cada vez más inmersivas y realistas. (Venegas, 2024, párr.11)

En consideración, el primer videojuego con gráficos 3D, aunque simple para los estándares actuales, fue un gran avance en la evolución de los videojuegos. Este hito abrió nuevas posibilidades tecnológicas, inspirando a los desarrolladores a crear mundos más complejos. Con el tiempo, los gráficos 3D mejoraron, permitiendo experiencias más realistas e inmersivas, cambiando la industria para siempre.

2.5 Evolución de la Calidad de Videojuegos 3D

La introducción definitiva del 3D traía consigo la necesidad de una revisión en el rol de la cámara para estos títulos. Para ello se adoptaron tres filosofías diferentes, que tenían mucho que ver con el género del juego y sus mecánicas, y que se materializaron en forma de tres tipos de cámara que siguen utilizándose actualmente. (Rodríguez, 2018, párr.7)

Mientras tanto, en el inicio los videojuegos 3D trajo consigo el cambio y revisión en el rol de la cámara por el tipo de videojuegos. A todo esto, se adoptaron varias filosofías dependiendo de su mecánica y el género del videojuego, que se hicieron tres tipos de ángulos que actualmente forman parte de los videojuegos más famosos actualmente significando otra experiencia que daba al jugar un videojuego.

Muchos títulos optaron por una cámara de seguimiento, heredada directamente de la era 2D. Similar a como funcionaba el efecto parallax pero en un entorno 3D, esta cámara se mueve en una dirección siguiendo el movimiento del personaje. Se utilizaba en juegos donde el desplazamiento en una sola dirección es la base de la experiencia, y por tanto ese ángulo de cámara podía seguir toda la acción. (Rodríguez, 2018, párr.8)

A consecuencia de este cambio, muchos títulos optaron por usar una cámara de seguimiento, heredada de la era 2D. Funciona de manera similar al efecto parallax, pero adaptado a un entorno 3D, moviéndose en una dirección para seguir al personaje. Esta técnica se utilizaba en juegos donde

el desplazamiento en una sola dirección era fundamental para la experiencia, permitiendo que la cámara se ajustara al movimiento del jugador de forma fluida.

La cámara fija optaba por perspectivas inmóviles, que solo cambiaban cuando el jugador salía de plano o cambiaba de localización. Esta cámara ya se utilizaba en el medio y siempre ha venido acompañado de un valor cinematográfico, de ahí su uso en aventuras gráficas. Las desventajas de esto son bastante obvias, podemos ver el mundo del juego únicamente limitados por la mirada de los desarrolladores (Rodríguez, 2018, párr.9).

En este caso, la cámara fija ofrecía perspectivas inmóviles, que solo cambiaban cuando el jugador se desplazaba fuera de la vista o cambiaba de ubicación. Esta técnica, comúnmente utilizada en el medio, tiene un valor cinematográfico, lo que la hace popular en aventuras gráficas. Sin embargo, sus desventajas son claras: los jugadores experimentan el mundo del juego desde un punto de vista limitado, restringido por la visión que los desarrolladores han elegido.

La cámara interactiva era la gran inclusión del 3D. Se trata de la cámara que el jugador puede manipular (a través de rotaciones verticales y horizontales) para tener la perspectiva que desee. Estas cámaras tuvieron un uso no tan extenso como debería, y en los primeros ejemplos no estaba demasiado bien implementada, sintiéndose un elemento inorgánico, un estorbo en la experiencia. En los mejores casos, el movimiento de la cámara estaba asignado a un *joystick* o botones independientes, como ocurrió con Super Mario 64 (1996), uno de los mejores ejemplos de juego que integró cómodamente el uso de la cámara interactiva o manual. (Rodríguez, 2018, párr.10)

Al respecto de la última actualización, la gran inclusión del 3D tuvo un gran impacto. Era la cámara que podía manipular el jugador para tener la perspectiva de su conveniencia. Este tipo de cámaras no tuvieron un extenso como debería, en las primeras versiones era un estorbo para la experiencia del jugador, para en los mejores casos el movimiento era a partir de un joystick o botones independientes, como lo es Súper Mario 64.

De esta disputa, y el deseo de Sony (principalmente de Ken Kutagari), director ejecutivo en su momento de la división de Sony *Computer Entertainment*) por crear una máquina de estas características, nació PlayStation. Una consola de 32 bits con capacidad de leer CD y enfocada

especialmente para el desarrollo en 3D. Una consola de sobremesa revolucionaria que se convertiría en la principal fuerza creadora de títulos 3D y que dominó en popularidad y ventas la quinta generación de consolas, convirtiéndose además en la primera consola del mundo en vender cien millones de unidades en todo el mundo. (Rodríguez, 2018, párr.21)

Por último, el deseo de llevar los videojuegos a consolas de Sony, más del director ejecutivo Ken Kutagari, por crear una máquina de esas características llega PlayStation, capaz de leer CD con 32 bits, enfocada para el desarrollo en 3D, revolucionaria en la principal fuerza creadora en títulos 3D el cual comino en ventas, siendo así la primera consola en vender cien millones de unidades.

2.6 Tipos de Modelos de Motores Gráficos

“La plataforma Android admite varios motores gráficos, incluido OpenGL ES, que es una subcategoría del estándar OpenGL diseñada específicamente para sistemas embebidos y dispositivos móviles” (Silva, 2015, p. 123).

En resumen, la plataforma Android es compatible con varios motores gráficos, siendo OpenGL ES uno de los más utilizados. Esta subcategoría de OpenGL está diseñada específicamente para sistemas embebidos y dispositivos móviles, optimizando el rendimiento en dispositivos con limitaciones de recursos. OpenGL ES permite a los desarrolladores crear aplicaciones gráficas y juegos de alto rendimiento en entornos móviles, aprovechando al máximo las capacidades de estos dispositivos.

“Uno de los motores más populares para el desarrollo de juegos en Android es Unity3D, que permite a los desarrolladores crear juegos 3D y 2D de alta calidad que se pueden desplegar en múltiples plataformas” (Silva, 2015, p. 145).

Asimismo, Unity3D es uno de los motores más ampliamente utilizados para el desarrollo de juegos en la plataforma Android. Este motor gráfico destaca por su capacidad de permitir a los desarrolladores crear tanto juegos en 3D como en 2D con un alto nivel de calidad visual. Gracias a su versatilidad y potentes herramientas, Unity3D facilita la creación de gráficos impresionantes y detallados que mejoran significativamente la experiencia del usuario.

“LibGDX es otro marco poderoso que ofrece un entorno flexible para desarrollar juegos tanto en 2D como en 3D en Android, proporcionando un conjunto robusto de herramientas para manejar gráficos, entrada y física” (Silva, 2015, p. 167).

En resumen, LibGDX es otro marco poderoso que proporciona un entorno flexible para desarrollar juegos en 2D y 3D en Android. Ofrece un conjunto completo de herramientas que facilitan el manejo de gráficos, entradas y física, permitiendo a los desarrolladores crear experiencias de juego complejas y optimizadas para dispositivos móviles. Además, su capacidad para ser utilizado en múltiples plataformas lo convierte en una opción popular para proyectos de juegos móviles.

2.7 Diseño del Motor Gráfico

El diseño del motor gráfico de Unity se centra en la escalabilidad y la eficiencia, permitiendo a los desarrolladores crear juegos tanto para dispositivos móviles de baja potencia como para sistemas de juego de alta gama (Goldstone, 2024, p.35).

Por otra parte, la orientación hacia la escalabilidad y la eficiencia implica que los desarrolladores pueden adaptar juegos tanto a dispositivos móviles con recursos limitados como a sistemas de juego de alto rendimiento. Esta filosofía de diseño permite que los juegos creados con Unity sean accesibles para una amplia gama de plataformas, desde teléfonos inteligentes y tabletas hasta consolas de videojuegos y PC de gama alta, sin comprometer la calidad visual o el rendimiento del juego.

“La arquitectura del motor gráfico de Unity se basa en un enfoque de componentes, donde los diferentes sistemas y funcionalidades se desacoplan para facilitar la reutilización y la extensibilidad del código” (GoldStone, 2024, p.37).

De igual manera, La arquitectura del motor gráfico de Unity se fundamenta en un enfoque basado en componentes, donde los distintos sistemas y funcionalidades se desacoplan. Este diseño permite una mayor flexibilidad, ya que los componentes pueden ser reutilizados y extendidos fácilmente. Unity facilita la personalización y el mantenimiento del código, lo que mejora la eficiencia en el desarrollo de juegos y aplicaciones interactivas .

La capacidad de renderizado del motor gráfico de Unity se apoya en tecnologías modernas como el rendering pipeline Scriptable Render Pipeline (SRP), que ofrece a los desarrolladores un control granular sobre el proceso de renderizado para lograr efectos visuales personalizados y un rendimiento optimizado (Goldstone, 2024, p.152).

Del mismo modo, la eficacia del motor gráfico de Unity en el renderizado se respalda en tecnologías avanzadas como el Scriptable Render Pipeline (SRP), un sistema que brinda un nivel excepcional de control sobre el proceso de renderizado. El SRP permite ajustar de manera minuciosa los aspectos visuales de un juego. Esta capacidad de personalización detallada no solo permite a los desarrolladores lograr efectos visuales únicos y de alta calidad, sino que también optimiza el rendimiento del juego.

2.8 Características de Motores Gráficos

Unity ofrece un sistema de física robusto y fácil de usar que permite a los desarrolladores simular el comportamiento realista de objetos y personajes en el mundo del juego (Addison, Manning y Nugent, 2016, p.85).

Por esta razón, Unity proporciona a los desarrolladores un sistema de simulación física que es tanto sólido en su funcionamiento como fácil de implementar. Este sistema permite a los creadores recrear de manera precisa y sencilla el comportamiento físico de objetos y personajes en el entorno del juego, brindando una experiencia realista y envolvente para los jugadores.

“Una de las características destacadas de Unity es su potente sistema de renderizado, que permite a los desarrolladores crear efectos visuales impresionantes y escenas visualmente impactantes con facilidad” (Addison, Manning y Nugent, 2016,p .88).

Igualmente, Unity es su motor gráfico que ofrece un conjunto completo de herramientas para crear efectos visuales impresionantes. Desde gráficos 2D hasta entornos 3D complejos, Unity permite a los desarrolladores dar vida a sus ideas con facilidad, permitiendo la creación de escenas visualmente impactantes que cautivan a los jugadores y mejoran la inmersión en el juego.

La flexibilidad de Unity se ve reflejada en su capacidad para admitir una amplia gama de plataformas, incluyendo dispositivos móviles, consolas de videojuegos y computadoras de escritorio, lo que permite a los desarrolladores alcanzar audiencias diversas con sus juegos (Addison, Manning y Nugent, 2016, p.88).

Por lo tanto, la versatilidad de Unity se manifiesta en su capacidad para funcionar en una amplia gama de plataformas, desde dispositivos móviles hasta consolas de videojuegos y ordenadores de escritorio. Esta flexibilidad permite a los desarrolladores llegar a audiencias diversas con sus juegos, maximizando el potencial de distribución y asegurando que sus creaciones sean accesibles para una amplia variedad de jugadores en todo el mundo.

2.9 Optimización de Rendimiento

La optimización de rendimiento en Unity implica identificar y abordar cuellos de botella en el proceso de renderizado, como el exceso de polígonos, el uso ineficiente de texturas y los cálculos excesivos en shaders, para garantizar una experiencia de juego fluida en una variedad de dispositivos. (Ward, 2014, p.69)

Del mismo modo, el rendimiento en Unity implica la detección y solución de obstáculos que obstaculizan el proceso de renderizado, como la excesiva complejidad de los modelos, la gestión ineficaz de texturas y la carga de cálculos pesados en los shaders. Estos problemas deben ser abordados para asegurar una experiencia de juego fluida y consistente en una amplia gama de dispositivos móviles y plataformas.

Una estrategia clave para optimizar el rendimiento en Unity es la gestión eficiente de los recursos, que incluye la carga y descarga dinámica de activos, el uso de técnicas de *streaming* y la minimización de la sobrecarga de memoria para garantizar una ejecución fluida y sin interrupciones (Ward, 2014, p.72).

Por ello, un aspecto fundamental de la optimización de rendimiento en Unity es la gestión cuidadosa de los recursos del sistema, incluyendo la carga y descarga dinámica de activos, la

implementación de técnicas de streaming y la optimización del uso de memoria. Estas prácticas aseguran que el juego funcione sin problemas y sin interrupciones, incluso en dispositivos con recursos limitados.

La optimización de scripts y código en Unity es fundamental para mejorar el rendimiento del juego, mediante la implementación de técnicas como la reducción de llamadas a funciones costosas, el uso de estructuras de datos eficientes y la eliminación de bucles innecesarios para maximizar la velocidad de procesamiento y minimizar el consumo de recursos. (Ward,2014, p.779)

Asimismo, el rendimiento del juego en Unity implica también la optimización de los scripts y el código del juego. Esto implica minimizar las llamadas a funciones costosas, utilizar estructuras de datos eficientes y eliminar bucles innecesarios. Estas prácticas reducen la carga en la CPU y maximizan la velocidad de procesamiento, lo que resulta en un juego más fluido y reactivo en dispositivos móviles.

2.10 Herramientas de Desarrollo

Las herramientas de desarrollo de Unity ofrecen un conjunto poderoso de funcionalidades para los creadores de juegos, desde el Editor de Escenas que permite la construcción y edición de entornos virtuales hasta el Profiler que ayuda a identificar y solucionar cuellos de botella de rendimiento. (Thorn, 2012, p.145)

Además, en Unity contiene elementos, los cuales son una colección de funcionalidades robustas que ayudan a los creadores de juegos en diversas tareas, desde la construcción de escenarios virtuales hasta la optimización del rendimiento. Entre estas herramientas se encuentran el Editor de Escenas, que permite la creación y edición de entornos, y el Profiler, que ayuda a identificar y resolver problemas de rendimiento.

Con el Editor de Animaciones de Unity, los desarrolladores pueden dar vida a sus personajes con movimientos fluidos y realistas, mientras que el Visual Effect Graph ofrece un control sin precedentes sobre los efectos visuales, permitiendo la creación de escenas impresionantes y envolventes (Thorn,2012, p.134).

No obstante, Con el Editor de Animaciones de Unity, los desarrolladores pueden crear movimientos fluidos y realistas para sus personajes, mejorando la dinámica de sus juegos. Además, el Visual Effect Graph proporciona un control detallado sobre los efectos visuales, permitiendo la creación de escenas impresionantes y envolventes. Estas herramientas combinadas ofrecen a los creadores una gran flexibilidad y capacidad para diseñar experiencias visuales de alta calidad, elevando la interacción y el impacto visual en sus proyectos.

Las herramientas de colaboración de Unity, como Unity Collab y Git integrado, facilitan el trabajo en equipo al permitir a los desarrolladores compartir y sincronizar fácilmente sus proyectos, asegurando una colaboración eficiente y una gestión de versiones sin problemas (Thorn, 2012, p.152).

Es decir, las herramientas de colaboración de Unity, como Unity Collab y la integración con Git, simplifican el trabajo en equipo al permitir a los desarrolladores compartir y sincronizar proyectos de manera fácil y eficiente. Estas herramientas facilitan la gestión de versiones y aseguran una colaboración fluida, permitiendo que varios miembros del equipo trabajen simultáneamente en el mismo proyecto sin conflictos. Esto optimiza el flujo de trabajo y mejora la productividad al reducir los errores y mejorar la organización del proyecto.

2.11 Unity y su Interfaz

La interfaz de Unity ofrece herramientas y paneles diseñados para ayudar a los desarrolladores a crear su propio contenido uno de sus componentes es el scene view en dónde se construye la representación visual del entorno del juego por otro lado está el Game view muestra cómo se verá el juego cuando esté en ejecución permite hacer pruebas y depuración en tiempo real. (Luis Ruelas, 2018, p.2)

A pesar de que la interfaz de Unity está diseñada para ayudar a los desarrolladores a crear contenido de manera eficiente, ofreciendo herramientas y paneles clave. Uno de sus componentes principales es el Scene View, donde los desarrolladores construyen y ajustan la representación visual del entorno del juego. Por otro lado, el Game View muestra cómo se verá el juego durante la ejecución,

lo que permite realizar pruebas y depuración en tiempo real. Ambas vistas trabajan juntas para facilitar el diseño, la prueba y la optimización de los proyectos en Unity.

Unity es un Motor de videojuegos que permite crear juegos en 2D y 3D para diversas plataformas como Windows Android y IOS como principales plataformas que a su vez podrás crear videojuegos y proyectos con realidad aumentada por lo tanto su interfaz está diseñada para facilitar el flujo de trabajo del desarrollador. (Luis Ruelas, 2018, p.1)

Por otra parte, Unity es un motor de videojuegos que permite crear juegos en 2D y 3D para diversas plataformas, siendo Windows, Android y iOS algunas de las principales. Además, ofrece la posibilidad de desarrollar proyectos con realidad aumentada. Su interfaz está diseñada para optimizar el flujo de trabajo del desarrollador, proporcionando herramientas intuitivas y paneles que facilitan la creación, prueba y depuración de los videojuegos, lo que mejora la eficiencia y la productividad durante todo el proceso de desarrollo.

Unity ofrece cierto grado de personalización en su interfaz algunos usuarios pueden desear más opciones para adaptarla a sus preferencias específicas de flujo de trabajo Esto puede incluir la capacidad de mover redimensionar o personalizar los paneles y ventanas de la interfaz. (Ruelas, 2018, p.3)

Puesto a que Unity ofrece un grado de personalización en su interfaz, lo que permite a los usuarios adaptar el entorno a sus necesidades y preferencias de flujo de trabajo. Los desarrolladores pueden mover, redimensionar o incluso personalizar los paneles y ventanas, lo que facilita la organización del espacio de trabajo según las tareas específicas. Esta flexibilidad permite optimizar la experiencia de desarrollo, haciendo que cada usuario pueda configurar la interfaz de manera que se ajuste mejor a su estilo de trabajo y aumento de productividad .

2.12 Optimización del Rendimiento General de la Interfaz

Capas dependientes del dispositivo Un gran sistema de este tipo está compuesto de capas donde se establecen dependencias, en las capas superiores se encuentra la capa Hardware Destino y los

Drivers de dispositivos que ofrece especializaciones particulares para cada Sistema Operativo final: plataformas de escritorio Microsoft Windows, Linux, y MacOS; plataformas móviles como Apple iPhone, iPad, Android smart phones, Sony PlayStation Vita, Amazon's Kindle Fire, etc. Y consolas de juego como Microsoft's Xbox, Xbox 360 y Xbox One, Sony PlayStation, PlayStation 2, PlayStation 3 y PlayStation 4, y Nintendo's DS, GameCube, Wii y Wii U. Por debajo estas capas se apoyan en capas comunes a todos ellos, por lo que las capas inferiores se pueden considerar agnósticas en cuanto a la plataforma. (González, 2015, p.8)

Por lo tanto, las capas de el gran sistema de este tipo tiene la capa Hardware Destino y los Drivers de dispositivos, esto da consigo especializaciones únicas para cada Sistema Operativo, como Microsoft Windows, Linux y MacOS en computadoras o Apple iPhone, iPad, Android smartphones, Sony PlayStation Vita, etc., en móviles, y consolas de juego como Microsoft Xbox, Sony PlayStation, Nintendo DS, GameCube, Wii y Wii U. Abajo de estas capas se encuentran capas comunes a todos ellos, lo que hace que las capas inferiores sean consideradas agnósticas.

La mayoría de los motores se apoyan en código de terceros (Software Development Kits o SDK) para dar soporte a diferentes áreas. El código se expone a través de librerías que exponen un API o interfaz de programación común. Por ejemplo, algunas de las librerías comúnmente aceptadas son: La gestión de las estructuras de datos a través de librerías como STL, Boost, Loki, etc. Representación de gráficos confiando en librerías como OpenGL o DirectX, libgem, Edge (también para PlayStation 3), Glide, etc. Detección de colisiones y motores físicos para simulación de dinámicas de objetos usando librerías como Havok, PhysX (ofrecido por NVIDIA) u Open Dynamics Engine (ODE) como librería de código abierto. (González, 2015, p.8)

Además, los motores gráficos contando con sus bases son códigos de terceros, conocidos como Software Development Kits, para brindar soporte en diversas áreas. Este código se expone a través de librerías que ofrecen un API o interfaz de programación común. Algunas de las librerías aceptadas incluyen la gestión de estructuras de datos mediante librerías como STL, Boost, Loki, etc.

Animación de personajes siendo este un punto de mucha importancia para dotar a los juegos de realismo. Muchas empresas deciden desarrollar librerías propias, pero entre las disponibles se

encuentran Granny que permite trabajar con los formatos de aplicaciones de diseño 3D como Maya, 3D studio Max, etc. Manipulándolos en tiempo de diseño e importándolos en tiempo de ejecución, Havok Animation, Edge que aparte de funcionalidades de representación de gráficos ofrece API para animación. (Gonzáles, 2015, p.8)

Por otro lado, los personajes que se llegan a diseñar es un punto de gran importancia para dotar a los juegos de realismo. Aunque muchas empresas optan por desarrollar sus propias librerías, entre las opciones disponibles se encuentran Granny, que permite trabajar con formatos de aplicaciones de diseño 3D como Maya, 3D Studio Max, etc., manipulándolos en tiempo de diseño e importándolos en tiempo de ejecución. También destacan Havok Animation y Edge, esta última ofrece funcionalidades de representación gráfica además de un API para animación.

2.13 Motor de Renderizado

El motor de renderizado o de presentación es el componente más complejo de cualquier motor gráfico. Como en otras partes de la arquitectura, esta capa se divide habitualmente en subcapas para encargarse de aspectos específicos de la presentación y dependen a nivel más bajo de una capa dependiente del dispositivo. (Gonzáles, 2015, p.9)

Además, el motor de renderizado es el componente más complejo de un motor gráfico, ya que se encarga de generar y mostrar la imagen final en la pantalla. Para manejar esta complejidad, se divide en subcapas especializadas en diferentes aspectos, como la iluminación, las sombras, las texturas y los efectos visuales. Estas subcapas dependen de una capa más baja que interactúa directamente con el hardware, adaptándose a las capacidades del dispositivo.

Esta capa se optimiza para dibujar de la forma más eficiente posible un conjunto de formas geométricas básicas de la forma más rica posible; sombras, texturas, luces, etc. Se emplean aquí para dotar a dichas formas de aspecto deseado. Librerías gráficas como DirectX, OpenGL se emplean para satisfacer estos requisitos. En muchas ocasiones, estas formas se agrupan en conjuntos como mayas con miles de triángulos, nubes de puntos, partículas, textos, etc. Para optimizar su representación. (González, 2015, p.9)

Adicionalmente, esta capa se optimiza para dibujar de la forma más eficiente posible un conjunto de formas geométricas básicas, incluyendo sombras, texturas, luces, etc. Librerías gráficas como DirectX, OpenGL se emplean para satisfacer estos requisitos. Estas formas se agrupan en conjuntos como mayas con miles de triángulos, nubes de puntos, partículas, textos, etc., para optimizar su representación.

Esta capa se encarga también de trasladar la vista del mundo a través de una matriz de transformación a la vista de una o varias cámaras y a una proyección 3D tratando aspectos tales como el campo de vista y los planos de cortes cercanos y lejanos para evitar la representación de objetos no relevantes para la posición actual de las cámaras. Los materiales y luces por ejemplo se gestionan por sistemas específicos de materiales y de iluminación dinámica y el resultado final es un flujo de procesamiento que se agrupa en primitivas gráficas que se envían a la tarjeta gráfica y esta se encarga de ir procesando. (Gonzáles, 2015, p.9)

Por consiguiente, la hace el trabajo de trasladar la vista del mundo a través de una matriz de transformación a la vista de una o varias cámaras y a una proyección 3D es la capa de los aspectos como el campo de visión y planos de cortes cercanos y lejanos son tratados para evitar la representación de objetos no relevantes para la posición actual de las cámaras. Los materiales y luces, se gestionan mediante sistemas específicos de materiales e iluminación dinámica. El resultado final es un flujo de procesamiento que se agrupa en primitivas gráficas y se envía a la tarjeta gráfica para su procesamiento.

Los materiales describen qué texturas se emplean en cada forma geométrica y qué *shaders* usar para dibujar cada píxel y vértice; el sistema de iluminación calcula como las luces del entorno afectan a la representación de los materiales; superficies metálicas, rugosas, etc. Adquieren en esta fase su aspecto final. (Gonzáles, 2015, p.9)

Asimismo, los materiales definen qué texturas se aplican a cada forma geométrica y qué *shaders* se utilizan para dibujar cada píxel y vértice. El sistema de iluminación, por su parte, calcula cómo las luces del entorno afectan la apariencia de los materiales. En esta fase, las superficies metálicas, rugosas y otras características adquieren su aspecto final, ya que la interacción de la luz con los materiales determina su apariencia visual, haciendo que se vean más realistas o estilizados según

la configuración del entorno.

2.14 Capa de Optimizaciones

Se prioriza el objetivo de reducir el número de formas a representar con el consiguiente ahorro de tiempo de procesamiento que a la postre permitirá aumentar la calidad de lo que si se ha de representar. Esta reducción se produce eliminando aquellas formas que no están visibles para la cámara en cada frame o aplicando otros algoritmos; un buen ejemplo es la arquitectura OGRE plug&play que permite añadir o eliminar algoritmos ya desarrollados o desarrollar algoritmos propios. (González, 2015, p.9)

Por ende, la meta principal que se busca es reducir el número de formas a representar, lo que conlleva un ahorro de tiempo de procesamiento que permite mejorar la calidad de lo que debe ser representado. Esta reducción se logra eliminando formas que no son visibles para la cámara. Un ejemplo notable es la arquitectura OGRE plug&play, que facilita añadir o la eliminación de algoritmos ya desarrollados.

Todos los motores gráficos componen una capa normalmente 2D para presentar información relevante al entorno, así como elementos gráficos que sirvan al usuario para interactuar con el sistema; en un juego, se emplea para mostrar los puntos conseguidos, los menús, las vidas restantes. (González, 2015, p.10)

Además, todos los motores gráficos incluyen generalmente una capa 2D para mostrar información relevante del entorno, así como elementos gráficos que faciliten la interacción del usuario con el sistema. En el contexto de un juego, esta capa se utiliza para mostrar elementos como los puntos obtenidos, las vidas restantes, los menús, y otros indicadores visuales que permiten al jugador interactuar de manera efectiva con el juego y seguir el progreso en tiempo real.

2.15 Efectos Visuales y su Integración con la Realidad Virtual

Ciertos efectos requieren algoritmos específicos que fácilmente pueden ir sustituyéndose por nuevos que mejoren los resultados o incorporen más características. Sistemas de partículas (humo,

fuego, agua, etc), sistemas de reproducción o calco para mostrar gran cantidad de elementos del mismo tipo como los impactos de bala, pisadas, etc. Anti aliasing para reducir los bordes recortados en las imágenes, efectos de luz como saturación, HDR (Hight Dynamic Range), etc. Entrarían dentro de este grupo. (Gonzáles, 2015, p.10)

Por otra parte, algunos de los efectos es necesarios cumplir algoritmos específicos que fácilmente pueden ser sustituidos por nuevos para mejorar los resultados. Ejemplos de estos efectos incluyen sistemas de partículas como humo, fuego, agua, sistemas de reproducción o clonación para representar múltiples elementos similares como impactos de bala, pisadas. Anti-aliasing para suavizar los bordes de las imágenes, y efectos de iluminación como la saturación y HDR. Todos estos se clasifican dentro de esta categoría.

Realidad Virtual o VR por sus siglas en inglés han coexistido en el mundo de la tecnología desde los años 70 con el simulador Flight Simulation, pero la complejidad y elevado coste de las infraestructuras necesarias, la complejidad de cualquier desarrollo mínimo, y la falta de realidad del producto final, ha hecho que durante todos estos años el término haya aparecido en la escena tecnológica con la misma fuerza con la que ha desaparecido y se ha mantenido siempre latente, pero en reducidos círculos de desarrollo. (González, 2015, p.10)

A pesar de que la Realidad Virtual ha coexistido en el mundo tecnológico desde los años 70, cuando surgieron los simuladores de vuelo, su presencia ha sido intermitente. Esto se debe a la complejidad y los altos costos de las infraestructuras requeridas, así como a la dificultad en el desarrollo y la falta de realismo en los productos finales. A lo largo de los años, el término ha aparecido y desaparecido de la escena tecnológica, manteniéndose principalmente en círculos de desarrollo reducidos.

En los últimos años, la reducción de costes hardware, la simplificación en el desarrollo software mediante la reutilización de componentes y la evolución tecnológica a nivel de representación gráfica, procesamiento en GPUs, procesadores especializados cada vez más potentes, capacidad de almacenamiento, etc. Junto con nuevas arquitecturas de desarrollo de software (principalmente a través de Game Engine) que se han adaptado fácilmente a este paradigma han conseguido popularizar el uso de esta tecnología. (González, 2015, p.10)

Por otro lado, la reducción de los costos de hardware, la simplificación en el desarrollo el API software mediante la reutilización de componentes y la evolución tecnológica a nivel de representación gráfica, procesamiento en GPUs, procesadores especializados cada vez más potentes, capacidad de almacenamiento, etc., junto con nuevas arquitecturas de desarrollo de software principalmente mediante Game Engines han contribuido a popularizar el uso de esta tecnología en los últimos años.

VR en los motores gráficos Los distintos motores gráficos han ofrecido soporte a estos entornos tan específicos en sus distintas versiones, comprometiendo en mayor o menor medida la portabilidad de los desarrollos. Unity ha soportado históricamente distintos entornos de VR, pero con ciertos Problemas: Cada dispositivo de VR tenía un plugin específico, con lo que la misma aplicación tenía que adaptarse a cada entorno de ejecución. Distintos plugins para distintos dispositivos VR podían entrar en conflicto y el trabajo para soportar cada uno de estos dispositivos era considerable. Desarrollar una aplicación para entornos VR y no VR a la vez era costoso. (González, 2015, p.11)

Por consiguiente, los diversos motores gráficos han ofrecido soporte a estos entornos tan específicos en sus distintas versiones. En el caso de Unity, a lo largo de su historia, ha respaldado varios entornos de realidad virtual, aunque con desafíos significativos. Por ejemplo, cada dispositivo de VR requería un plugin específico, lo que implicaba que una aplicación debía adaptarse a cada entorno de ejecución. Además, desarrollar una aplicación que funcionara tanto en entornos de VR como no VR al mismo tiempo resultaba costoso.

La nueva versión de Unity (Unity 5.1) ofrece un API común para desarrollar en entornos VR dando soporte a ciertos dispositivos VR existentes en el mercado. Esto simplifica el desarrollo multi dispositivo a unos pequeños cambios en la configuración, y si bien es sencillo, se irá adaptando a las necesidades de los desarrolladores según las necesidades que vayan surgiendo. (González, 2015, p.11)

Además, la nueva versión de Unity (Unity 5.1) ofrece un API para desarrollar en entornos de realidad virtual, brindando soporte a diversos dispositivos disponibles en el mercado. Esto

simplifica el desarrollo multi-dispositivo, requiriendo solo pequeñas modificaciones en la configuración. Aunque el API es sencillo, se irá adaptando a las necesidades de los desarrolladores según vayan surgiendo nuevas demandas.

2.16 Motores Gráficos

CryEngine nació como un demostrador de la empresa Crytek de las capacidades de la tarjeta gráfica Nvidia evolucionando hacia un juego completo llamado Far Cry. CryEngine3 se ha convertido en una plataforma para desarrollo de juegos con herramientas para la creación de assets y gráficos de gran calidad en tiempo Real. Los desarrollos son multiplataforma incluyendo Xbox One, Xbox 360, PlayStation 4, PlayStation 3, Wii U y PC. (González, 2015, p.5)

Por ende, a partir de un demostrador inicial de las capacidades de la tarjeta gráfica Nvidia, surgió Far Cry, un juego completo. Posteriormente, el CryEngine3 se ha transformado en una plataforma de desarrollo de juegos que ofrece herramientas para crear activos y gráficos de alta calidad en tiempo real, siendo compatible con diversas plataformas como Xbox One, Xbox 360, PlayStation 4, PlayStation 3, Wii U y PC.

Unity entorno el desarrollo de juegos multiplataforma, así como su runtime que permite desplegar los desarrollos a un gran número de plataformas incluyendo las móviles como Apple iOS, Google Android, Windows Phone, Blackberry 10, consolas (Microsoft Xbox 360, Xbox One, Sony Playstation 3 y 4, Nintendo Wii y Wii U) así como entornos de escritorio (Microsoft Windows, Apple Macintosh y Linux) o incluso un Webplayer para el despliegue en los principales navegadores. El entorno unificado de desarrollo permite desde la creación, manipulación y visualización de assets y entidades, visualización tanto en el escritorio como en el hardware destino (móvil conectado al USB en modo depuración). Ofrece También herramientas para analizar el comportamiento en las distintas plataformas y permite la programación en lenguajes JavaScript, C# o Boo para facilitar la creación de animaciones. (González, 2015, p.5)

Asimismo, el entorno de desarrollo de juegos multiplataforma, posibilita el despliegue de desarrollos en una amplia gama de plataformas, que incluyen dispositivos móviles como iOS y Android, así como consolas como Microsoft Xbox y Sony Playstation, además de entornos de

escritorio como Microsoft Windows, Apple Macintosh y Linux, e incluso Webplayer. Este entorno unificado permite la creación, manipulación y visualización de activos y entidades. Además, proporciona herramientas para analizar el comportamiento en diversas plataformas.

Unreal Engine nacido en 1998, Epic Games empleó este motor para desarrollar Unreal. Desde entonces se ha convertido en un referente en muchos aspectos como en los juegos de primera persona (FPS). Unreal Engine 4 (UE4) es la última versión y destaca por herramientas para la creación de Shaders, programación de lógicas de juego y gran entorno gráfico de desarrollo. Su arquitectura abierta permite la modificación del motor para adaptarse a necesidades específicas de desarrolladores para optimizar su funcionamiento En distintas plataformas y destaca en el desarrollo de juegos 3D de primera persona. (González, 2015, p.5)

Ahora bien, desde 1998, Epic Games ha utilizado este motor para dar vida a Unreal. Desde entonces se ha convertido en un referente en muchos aspectos como en los juegos de primera persona. Unreal Engine 4 es la última versión y destaca por herramientas para la creación de Shaders, programación de lógicas de juego y gran entorno gráfico de desarrollo. Su arquitectura abierta permite la modificación del motor para adaptarse a necesidades específicas de desarrolladores para optimizar su funcionamiento en distintas plataformas y destaca en el desarrollo de juegos 3D.

Microsoft XNA basado en lenguajes .NET manejado, con el soporte de todas las librerías Genéricas de .NET y algunas librerías específicas XNA, es una plataforma enfocada a usuarios con un nivel medio de programación. Se desarrolla sobre Microsoft Visual Studio desde la creación de Assets hasta la programación y permite desarrollar aplicaciones para PC y para la consola Xbox 360. Existe una plataforma de comercialización de estas aplicaciones en la que se pueden publicar los desarrollos propios por una pequeña tasa, lo que ha atraído a muchos desarrolladores. (González, 2015, p.6)

Dado a que, los lenguajes .NET es el respaldo de todas las librerías genéricas de .NET y algunas específicas como XNA, esta plataforma se centra en usuarios con un nivel medio de habilidades de programación. Se utiliza en Microsoft Visual Studio para todo, desde la creación de activos hasta la codificación, y permite la creación de aplicaciones tanto para PC como para la consola Xbox 360. También ofrece una plataforma de comercialización para estas aplicaciones, donde los

desarrolladores pueden publicar sus trabajos por una pequeña tarifa.

Sony PhyreEngine en el año 2008, Sony presentó esta plataforma para hacer más accesible el desarrollo de juegos para sus consolas. PhyreEngine se ha convertido en la herramienta de referencia para la creación de juegos para la PlayStation 4, PlayStation 3, PlayStation 2, PlayStation Vita y plataformas PSP. Permite la utilización de características hardware específico de dichas consolas como la ejecución en paralelo sobre su sistema de procesadores en Celda, un conjunto de procesadores RISC PowerPC de 64 bits a 3.2GHz de frecuencia y diseñado para una ejecución real en paralelo de los procesos. Se ofrece de forma Gratuita a todos los desarrolladores licenciados por Sony como parte del SDK de PlayStation. Más de 90 juegos se han desarrollado con este motor como Flower o Journey, AMY, Souls o Dark Souls entre otros. (González, 2015, p.6)

No obstante, desde su introducción en 2008 por Sony, esta plataforma ha revolucionado el desarrollo de juegos para sus consolas. PhyreEngine se ha establecido como la herramienta principal para crear juegos para la PlayStation 4, 3, 2, Vita y plataformas PSP. Facilita el aprovechamiento de las características específicas de hardware de estas consolas, como la ejecución en el sistema de procesadores Celda. Como parte del SDK de PlayStation, se ofrece de forma gratuita a todos los desarrolladores licenciados por Sony.

Game Engine tomando como referencia la arquitectura y herramientas empleadas para desarrollar ciertos juegos que se han convertido en referentes a lo largo de la historia de los videojuegos, se han popularizado y han evolucionado ciertas plataformas empleadas en su momento para crear dichos juegos específicos. Por ejemplo, el Motor de Juegos Half-Life se basa en el código empleado para la creación de Half Life 2 y sus secuelas (Episode One, Episode Two, Team 2 o Portal) y expone un motor de gran calidad a nivel de gráficos y herramientas que lo ponen a la altura de Unreal Engine 4. (González, 2015, p.6)

Basándose en la arquitectura y herramientas utilizadas para desarrollar juegos icónicos a lo largo de la historia de los videojuegos, ciertas plataformas empleadas en su momento para crear esos juegos específicos han ganado popularidad y evolucionado. Por ejemplo, el Motor de Juegos Half-Life está inspirado en el código utilizado para desarrollar Half-Life 2 y sus secuelas (Episodio Uno,

Episodio Dos, Team Fortress 2 y Portal), ofreciendo un motor de alta calidad en términos de gráficos y herramientas que lo sitúan al nivel Unreal Engine 4.

2.17 Porcentaje de Efectividad

Los motores gráficos son fundamentales para la visualización de datos en aplicaciones de escritorio y en línea. Estos motores permiten la representación de información en forma de gráficos, diagramas y otros elementos visuales que facilitan la comprensión de los datos. Los motores gráficos pueden ser utilizados para mostrar datos estadísticos, tendencias, y relaciones entre variables, lo que ayuda a los usuarios a tomar decisiones informadas. (Foley, 1995, p.12)

Por eso mismo, los motores gráficos son esenciales para la visualización de datos en aplicaciones de escritorio y en línea. Permiten representar información de manera visual a través de gráficos, diagramas y otros elementos que facilitan la comprensión de los datos. Estos motores son útiles para mostrar estadísticas, tendencias y relaciones entre variables, ayudando a los usuarios a interpretar mejor la información y tomar decisiones más informadas. Su capacidad para hacer los datos más accesibles y fáciles de entender mejora la experiencia del usuario.

Los motores gráficos modernos han mejorado significativamente en términos de rendimiento, escalabilidad y flexibilidad. Estos motores pueden manejar grandes cantidades de datos y ofrecen una amplia variedad de opciones para personalizar la visualización de la información. Además, los motores gráficos pueden integrarse con otras tecnologías, como bases de datos y sistemas de análisis, para ofrecer una visión más completa de los datos. (Foley, 1995, p.15)

Asimismo, los motores gráficos modernos han experimentado mejoras notables en rendimiento, escalabilidad y flexibilidad. Son capaces de manejar grandes volúmenes de datos y brindan diversas opciones para personalizar la visualización, adaptándose a las necesidades específicas de cada usuario. Además, pueden integrarse con otras tecnologías, como bases de datos y sistemas de análisis, lo que permite ofrecer una visión más completa y detallada de los datos.

La elección del motor gráfico adecuado depende de varios factores, como el tipo de datos a visualizar, el tamaño de la base de datos, y las necesidades específicas de la aplicación. Algunos

motores gráficos son más adecuados para la visualización de grandes cantidades de datos, mientras que otros son más adecuados para la creación de gráficos personalizados. Es importante considerar estas variables al seleccionar un motor gráfico para una aplicación. (Foley, 1995, p.18)

Por lo tanto, la elección de un motor gráfico correcto depende de factores como el tipo y tamaño de los datos, y las necesidades específicas de la aplicación. Algunos motores son más apropiados para visualizar grandes volúmenes de datos, mientras que otros se adaptan mejor a la creación de gráficos personalizados. Es importante considerar estas variables al seleccionar un motor gráfico.

Los motores gráficos también pueden ser utilizados para crear interfaces de usuario interactivas y dinámicas. Estos motores permiten la creación de gráficos que se actualizan en tiempo real, lo que facilita la exploración y análisis de los datos. Además, los motores gráficos pueden ser utilizados para crear visualizaciones que se adaptan a las necesidades específicas del usuario, lo que mejora la experiencia de usuario. (Román, 2018, p.20)

Por lo tanto, los motores gráficos no solo se limitan a la creación de gráficos visuales, sino que también son herramientas poderosas para diseñar interfaces de usuario interactivas y dinámicas. Gracias a su capacidad para generar gráficos que se actualizan en tiempo real, facilitan un análisis y una exploración más detallada de los datos. Esto permite a los usuarios interactuar de manera más eficiente con la información.

En resumen, los motores gráficos son una herramienta fundamental para la visualización de datos y la creación de interfaces de usuario interactivas. Estos motores ofrecen una amplia variedad de opciones para personalizar la visualización de la información y pueden ser utilizados para crear gráficos que se adaptan a las necesidades específicas del usuario. (Román, 2018, p.20)

Así que, los motores gráficos juegan un papel clave en la visualización de datos y en el diseño de interfaces de usuario interactivas. Gracias a su versatilidad, estos motores permiten personalizar la presentación de la información de manera detallada, adaptándose a las necesidades específicas de los usuarios. A través de la capacidad de crear gráficos dinámicos y actualizados en tiempo real, se facilita un análisis más profundo y una interacción más eficiente con los datos.

2.18 Costos de Licencias de Motores Gráficos

Cualquier desarrollador de videojuegos o usuario en general puede comercializar lo que cree en Unreal Engine sin necesidad de pagar a Epic Games por el uso de Unreal Engine. Sólo aquellos proyectos que generen más de \$1 MDD son los que deben pagar una cuota de 5%. Así pues, esto no cambiará. (Galván. 2023. p. 8)

Asimismo, el costo del uso de Unreal 4 y 5 se basa en 3 modelos de negocio, el primero es el de uso gratuito el cual se usa para descargar y usar proyectos no comerciales sin ningún costo inicial. El modo de regalías consiste en que Epic Games la desarrolladora del motor gráfico cobra un 5% de regalías sobre ingresos que superen 1 millón de dólares en ingresos. Negocio personalizado, en algunos casos para grandes empresas es posible negociar un acuerdo en el cual se puede negociar una licencia con algún pago por adelantado en lugar de regalías.

Desde 2016, Unity pasó a un modelo de suscripción, con opciones gratuita (para uso personal o empresas pequeñas que generan menos de 100.000 dólares anuales, cantidad que posteriormente pasó a 200.00) y de pago (basado en lo que ingresan los juegos que usan el motor). Hubo un primer paso hacia el cobro más generalizado en junio de 2021. (Tones. 2023. p. 11)

Por otra parte, Unity adoptó un modelo de suscripción, ofreciendo opciones tanto gratuitas como de pago. La opción gratuita está disponible para uso personal o para pequeñas empresas cuyos ingresos anuales no superen los 100,000 dólares, cifra que luego se incrementó a 200,000 dólares. Además, la opción de pago se basa en los ingresos generados por los juegos que utilizan el motor. En junio de 2021, Unity dio un primer paso hacia un modelo de cobro más amplio, comenzando a aplicar tarifas de manera más generalizada desde 2016.

2.19 Accesibilidad a la Creación de Videojuegos

La dificultad de usar Unreal Engine depende de varios factores, como el tipo de juego que se quiera crear, el nivel de experiencia que se tenga o el tiempo y los recursos que se dispongan. No es lo mismo crear un juego 2D sencillo que un juego 3D complejo. No es lo mismo tener conocimientos previos de programación, diseño gráfico o animación que empezar desde cero. No es lo mismo

dedicar unas horas al día que unas semanas o meses al proyecto. En general, se puede decir que Unreal Engine es un motor gráfico difícil de usar, sobre todo para los principiantes o aficionados. Unreal Engine es un motor gráfico muy completo y complejo, que requiere mucho tiempo y esfuerzo para aprender a usarlo correctamente. (Meléndez.2023, p.16)

Además, su accesibilidad, Unreal Engine ofrece una gran cantidad de herramientas y funcionalidades que facilitan el proceso de desarrollo. Su sistema de materiales avanzado permite a los artistas y diseñadores crear texturas y efectos visuales complejos sin necesidad de escribir código. La integración de herramientas de terceros, como Adobe Photoshop y Autodesk.

La documentación exhaustiva y los tutoriales en línea de Unreal Engine también contribuyen a su accesibilidad. La comunidad en línea de desarrolladores de Unreal Engine es activa y colaborativa, lo que significa que los desarrolladores pueden encontrar respuestas a sus preguntas y soluciones a sus problemas con facilidad. Unity no es un lenguaje en sí, sino que es un motor para video juegos. O simulaciones, si quieres hacer eso. Realmente es bastante amplio. Aprender a utilizar la interfaz de Unity no es difícil, aprender para qué sirve cada cosa y a utilizar contenido de la tienda (o contenido incluido en Unity) tampoco lo es, pero sí necesitas tiempo para eso, sobre todo si no tienes experiencia con programas similares, incluso puedes crear juegos sin saber programar, sin embargo, serán bastante sencillos debido a que no podrás incluir más complejidad a lo que viene incluido. (Caracamo.2015, p.11)

De igual manera, se ha destacado por su accesibilidad y facilidad de uso para una amplia gama de desarrolladores, desde principiantes hasta expertos. Su interfaz intuitiva y su enfoque en la programación orientada a objetos a través de C#, Unity ofrece una gran cantidad de herramientas y funcionalidades. Su Asset Store, una tienda en línea llena de activos y recursos creados por la comunidad, permite a los desarrolladores acceder a una amplia gama de elementos prefabricados.

CAPÍTULO III

Sistemas Operativos Android para Celulares de Gama Media

Capítulo III. Sistemas Operativos Android para Celulares de Gama Media

3.1 Primeros Videojuegos Hechos para Android

El primer juego para móvil fue lanzado por Nokia, hoy propiedad de Microsoft. Era el juego de la víbora que debía seguir un cuadrito. Venía dentro del sistema del Nokia 6110.

El juego era sencillo. Mientras la gente se asombraba con los juegos de la Nintendo 64 con sus gráficos mejorados, todo dentro de un cartucho, Nokia con éste sencillo juego cambiaba totalmente el uso de los móviles para siempre. (Pérez, 2016, párr.9)

Como resultado, el primer juego para móvil fue lanzado por Nokia, una empresa que actualmente pertenece a Microsoft. Este juego era el clásico de la víbora, en el que el objetivo era hacer que la serpiente siguiera un cuadrito en la pantalla. Venía preinstalado en el sistema operativo del Nokia 6110. Aunque el juego era muy sencillo en comparación con los avanzados gráficos de la Nintendo 64, que funcionaban a través de cartuchos, Nokia logró transformar por completo el uso de los teléfonos móviles.

Después del ‘boom’ del Snake, poco a poco los teléfonos móviles fueron incluyendo más y más juegos. Aprovechando que los títulos arcade más tradicionales no eran algo muy difícil de llevar con la potencia de los dispositivos de la época (y no quedaban tan mal a blanco y negro), varios de estos juegos acabaron en las memorias internas de los aparatos, entre ellos el Space Invaders. El famoso videojuego de destruir marcianos que bajan por la pantalla tuvo gran popularidad entre los jugadores de la época e incluso fueron saliendo al mercado nuevas versiones años más tarde, con mejores gráficos y a color. (Europa Press, 2017, párr.6)

A partir del ‘boom’ del Snake, los teléfonos móviles poco a poco incluyeron más juegos. Trayendo juegos arcade más tradicionales no eran muy difíciles de llevar con la potencia de los dispositivos, varios juegos acabaron en la memoria interna de los móviles, entre ellos Space Invaders, el famoso juego de marcianos tuvo una gran popularidad en esa época, saliendo al mercado actualizaciones con mejores gráficos y a color.

Pasando ya a los juegos a color de los 2000 en adelante, un título que tuvo su audiencia en las plataformas móviles fue *Driver 3*, que primero contó con versiones en consolas y ordenador. Aunque no tenía grandes gráficos y no era todavía en 3D, con títulos como este ya se empezaba a vislumbrar el potencial que tenían los ‘Smartphone’ del futuro y los videojuegos móviles, ya que cada vez era más normal ver a jugadores casuales de cualquier género y edad aprovechar su tiempo libre dentro de este campo. (Europa Press, 2017, párr.10)

Por consecuencia, con la llegada de los juegos a color a partir de los años 2000, títulos como *Driver 3* encontraron su público en las plataformas móviles. Originalmente lanzado para consolas y ordenadores, el juego llegó a los teléfonos, aunque aún carecía de gráficos avanzados y no era completamente en 3D. Sin embargo, su presencia en el mercado marcó un paso importante al demostrar el potencial creciente de los dispositivos móviles para el entretenimiento digital.

Asphalt Urban GT salió primero para la N-Gage, el famoso híbrido de Nokia entre teléfono y videoconsola portátil que nunca consiguió despuntar, y la Nintendo DS; poco más tarde llegó a los móviles del momento. Este videojuego fue importante ya no solo por la gran popularidad que alcanzó, sino porque revitalizó la industria de los videojuegos móviles, que se había llevado una decepción con la N-Gage. Con él ya se empezaban a ver mejoras gráficas y de jugabilidad. (Europa Press, 2017, párr.12)

Por otro lado, *Asphalt Urban GT* debutó inicialmente en la N-Gage, el innovador pero fallido híbrido de Nokia entre teléfono y consola portátil, y también estuvo disponible en la Nintendo DS. Posteriormente, el juego llegó a los teléfonos móviles del momento, convirtiéndose en un título clave para la industria de los videojuegos móviles. No solo alcanzó una gran popularidad, sino que también revitalizó el sector tras la decepción que supuso el fracaso comercial de la N-Gage. Con este juego, se empezaron a evidenciar notables avances gráficos y mejoras en la jugabilidad.

Angry Birds, el juego de los pájaros que buscan derribar fortificaciones y deshacerse de los cerdos. La de cantidad de horas de diversión y muertas que habrá matado este título en alguna de sus múltiples versiones. Fácil de jugar, con múltiples niveles y con nuevo contenido (pájaros, cerdos diferentes) conforme vas avanzando en el juego, este título ha sido uno de los que más han logrado despuntar con la llegada de los ‘smartphones’. (Europa Press, 2017, párr.16)

Así mismo, la cantidad de horas de diversión y entretenimiento que este título ha proporcionado es incalculable, convirtiéndose en el compañero ideal para matar el tiempo en cualquiera de sus múltiples versiones. Su jugabilidad sencilla, combinada con múltiples niveles y la incorporación constante de nuevo contenido, como pájaros y cerdos con diferentes características, lo ha posicionado como uno de los juegos más populares tras la llegada de los *smartphones*.

3.2 Evolución de Juegos en Android

Durante la década de 1990 los teléfonos móviles se transformaron de dispositivos analógicos voluminosos en maravillas digitales más refinadas. A medida que la tecnología móvil se volvió más accesible y asequible, los terminales comenzaron a penetrar en los mercados masivos a nivel mundial. Con esta adopción generalizada llegó una nueva oportunidad para los juegos móviles. (Hervás, 2024, párr.6)

Mientras, en los años 90, los teléfonos móviles experimentaron una gran transformación, dejando atrás su diseño voluminoso y analógico para convertirse en dispositivos digitales más sofisticados. A medida que esta tecnología se hizo más accesible y económica, los terminales comenzaron a llegar a los mercados masivos en todo el mundo. Esta expansión abrió una nueva oportunidad para los juegos móviles, que emergieron como una forma innovadora de entretenimiento portátil, marcando el inicio de una industria en constante evolución.

A medida que avanzaban los años, también lo hicieron las capacidades de los juegos. Con la llegada de las plataformas basadas en Java -a finales de la década de 1990 y principios de la de 2000-, los desarrolladores obtuvieron la capacidad de crear juegos más complejos con gráficos y mecánicas mejoradas. Esto condujo a una explosión de contenido, con títulos que iban desde clásicos arcade hasta nuevas experiencias innovadoras. (Hervás, 2024, párr.8)

A partir del paso de los años, las capacidades de los juegos móviles también evolucionaron significativamente. La introducción de plataformas basadas en Java, a finales de la década de 1990 y principios de la de 2000, permitió a los desarrolladores crear títulos más complejos, con gráficos mejorados y mecánicas de juego más sofisticadas. Esto dio lugar a una explosión de contenido,

ofreciendo una amplia variedad de opciones que iban desde versiones de clásicos arcade hasta nuevas experiencias innovadoras.

Ya a finales de la década de 2000, los títulos existentes se habían establecido firmemente como un actor importante en la industria con millones de usuarios en todo el mundo y una amplia gama de desarrollos disponibles, los teléfonos móviles se habían convertido en más que dispositivos de comunicación; eran casi consolas de juegos. (Hervás, 2024, párr.9)

Por consecuencia a finales de los 2000, los juegos móviles se habían consolidado como un componente fundamental de la industria del entretenimiento, con millones de usuarios en todo el mundo y una amplia gama de títulos disponibles. Los teléfonos móviles, que originalmente eran simples herramientas de comunicación, evolucionaron para convertirse en auténticas plataformas de juego, casi comparables a consolas portátiles, ofreciendo experiencias cada vez más inmersivas y diversificadas.

La década de 2010 -por el momento todo sigue más o menos parecido hasta la fecha-, marcó un punto de inflexión en la historia de los juegos móviles, gracias a la adopción generalizada de los teléfonos inteligentes. Con la introducción de dispositivos como los iPhone y los teléfonos con sistema operativo Android, los juegos alcanzaron nuevas cotas de popularidad y accesibilidad (sin dejar de lado un aumento de la calidad impresionante). (Hervás, 2024, párr.10)

Así mismo en la década de los 2010, marcó un punto de inflexión en la evolución de los juegos móviles, un cambio que se mantiene vigente hasta la fecha. La adopción masiva de los teléfonos inteligentes transformó por completo la forma de jugar. Con la llegada de dispositivos como el iPhone y los teléfonos con sistema operativo Android, los juegos móviles alcanzaron niveles sin precedentes de popularidad y accesibilidad.

Los juegos móviles están a punto de experimentar una revolución. Varias tecnologías emergentes convergen para crear experiencias de juego más fluidas. Pero, sin duda, el 4G -pero especialmente el posterior 5G- resultan ser el cambio definitivo. Con velocidades de descarga muy rápida y una latencia mínima, se acabaron los tiempos de espera y las partidas con lag. Es posible jugar con personas de todo el mundo con una experiencia increíble propia de las consolas, y títulos de estas

llegan a los juegos móviles como Fortnite. (Hervás, 2024, párr. 12)

Como consecuencia, los juegos móviles están a las puertas de una revolución impulsada por la convergencia de tecnologías emergentes que prometen experiencias de juego más fluidas y envolventes. Sin embargo, el verdadero cambio transformador ha sido la llegada del 4G y, especialmente, del 5G. Con velocidades de descarga ultrarrápidas y una latencia prácticamente inexistente, los tiempos de espera y las partidas con *lag* han quedado atrás. Esta evolución ha permitido jugar en línea con personas de cualquier parte del mundo.

El futuro apunta a ofrecer grandes cambios con opciones como la realidad aumentada implementada en todos los rincones o, como no, el acceder a los títulos en la nube sin necesidad de tener nada instalado en el teléfono o Tablet. El caso es que los juegos móviles han pasado de ser algo residual, que apenas tenía impacto en usuarios y mercados, a ser ahora mismo una industria de entretenimiento vital y en pleno crecimiento. Y nada hace pensar que esto va a cambiar... todo lo contrario (Hervás, 2024, párr. 14).

Por otro lado, el futuro apunta a ofrecer cambios como lo es la realidad virtual o aumentada en todos los rincones. O el acceder a los títulos de la nube sin tener nada instalado en los teléfonos móviles. Para todo esto los teléfonos móviles pasaron de ser algo residual a ser una industria de entretenimiento vital y en pleno crecimiento. Y nada hace cambiar esta idea, todo lo contrario.

3.3 Evolución de Marcas de Celulares Android

El primer teléfono móvil fue inventado en 1973 por Motorola. El 3 de abril de 1973, el ingeniero de Motorola Martin Cooper hizo la primera llamada de teléfono móvil de la historia con el DynaTAC 8000X. El prototipo pesaba 1,1 kg y esa bastante grande para lo que acostumbramos hoy en día. Este aparatoso dispositivo ofrecía un tiempo de conversación de solo 30 minutos y necesitaba 10 horas para recargarse. Hasta entonces, lo más parecido a no estar atado a un teléfono fijo era tener un teléfono instalado en el coche, así que fue toda una revolución. (González, 2023, párr.4)

A partir del primer teléfono móvil fue inventado en 1973 por Motorola y el 3 de Abril de 1973, se

realizó la primera llamada en la historia a cargo de Motorola Martin Cooper con el DynaTAC 8000X, el cual tenía un peso de 1,1 kg y era bastante grande a comparación a lo que conocemos hoy en día. Este aparato ofrecía un tiempo de conversación de solo 30 minutos necesitando 10 horas para cargarse.

Por supuesto, las grandes empresas no tardaron en seguir los pasos de Motorola y comenzaron a lanzar sus propias versiones: Nokia, por ejemplo, entró en el mercado en 1987 con el Mobira Cityman 900. Con su peso de 760 gramos, el teléfono representaba una mejora significativa respecto al DynaTAC. Otro año más tarde, Samsung lanzó su primer teléfono móvil en 1988 con el SH-100. (González, 2023, párr.8)

Por otra parte, las grandes empresas no tardaron en seguir el camino trazado por Motorola, lanzando sus propias versiones de teléfonos móviles. Nokia, por ejemplo, debutó en el mercado en 1987 con el Mobira Cityman 900, un dispositivo que pesaba 760 gramos y representaba una mejora notable en comparación con el voluminoso DynaTAC de Motorola. Un año después, en 1988, Samsung se sumó a la competencia con su primer teléfono móvil, el SH-100.

El primer teléfono móvil con teclado QWERTY fue el Nokia Communicator 9000, lanzado en 1996. Además de tener teclado, también introdujo muchas funciones relacionadas con la empresa, como correo electrónico, navegación web, fax, procesamiento de textos y hojas de cálculo. (González, 2023, párr.12)

Por consecuencia, el Nokia Communicator 9000, lanzado en 1996, fue el primer teléfono móvil en incorporar un teclado QWERTY, marcando un hito en la evolución de los dispositivos móviles. Además de su innovador teclado físico, el dispositivo introdujo una amplia gama de funciones empresariales avanzadas para la época, como correo electrónico, navegación web, envío y recepción de faxes, así como herramientas para procesar textos y manejar hojas de cálculo, sentando las bases de lo que hoy conocemos como *Smartphone*.

En 2002, la innovación llegó con el Sanyo SCP-5300, un teléfono que permitía ver las fotos en la pantalla, en lugar de tener que conectarlo a un ordenador. Este innovador teléfono también incorporaba dos pantallas en color y una cámara con flash. (González, 2023, párr.15)

Por otro lado, el Sanyo SCP-5300 fue un teléfono innovador que permitió a los usuarios ver las fotos directamente en la pantalla, sin necesidad de conectarlo a un ordenador. Además de esta funcionalidad, incorporaba dos pantallas en color y una cámara con flash. Este dispositivo, lanzado en 2002, marcó un hito en la evolución de los teléfonos móviles, llevando las capacidades multimedia a un nuevo nivel.

De nuevo, el teléfono siguió haciéndose más pequeño y más funcional, con Nokia y Motorola dominando los mercados de todo el mundo. Las cámaras también subieron de nivel cuando Nokia presentó el 7610, su primer Smartphone con una cámara de 1 Mpx. También presentó el primer teléfono con itinerancia global, el Nokia 6630. Por otro lado, el primer teléfono resistente al agua apareció en 2005. Se trataba del Casio G'zOne, que podía sumergirse hasta 1 metro de profundidad. (González, 2023, párr.16)

Después del teléfono, siguió haciéndose más pequeño y con diversas funciones, Nokia y Motorola seguían dominando el mercado en todo el mundo. Subiendo la calidad de las cámaras cuando Nokia presento el 7610, el primer Smartphone con una cámara de 1 Mpx. Igualmente presento el primer teléfono con itinerancia global nombrado Nokia 6630. Asi mismo en 2005 se presentó el primer teléfono resistente al agua, llamado Casio G'zOne, que se podía sumergir un metro de profundidad.

La gran pantalla del iPhone también permitía por primera vez navegar por Internet en un dispositivo relativamente pequeño. Asimismo, fue el primer móvil de la historia en incluir aplicaciones dedicadas para YouTube y Google Maps. (González, 2023, párr. 17)

Así mismo, las cosas cambiaron poco a poco, sobre todo con la llegada del primer iPhone en 2007 por parte de Apple. También conocido como iPhone 2G, saliendo al mercado y dejando al mundo bocabierto gracias a la eliminación de los botones físicos, dando paso a una interfaz táctil. Contando con una gran pantalla del iPhone, permitiendo por primera vez, navegar por Internet en un teléfono móvil. Fue el primer móvil de la historia en incluir YouTube y Google Maps.

En 2010, Samsung lanzó su nuevo buque insignia: el Samsung Galaxy S (ahora ya hablamos del Galaxy S23). En ese momento, era el Smartphone más fino con 9,9 mm y tenía el procesador

gráfico más rápido. Ese mismo año, Samsung también lanzó el primer Smartphone 4G: el Samsung SCH-R900. (González, 2023, párr.20).

Igualmente, Samsung lanzó el Samsung Galaxy S, un modelo que se convirtió en su buque insignia al ser el smartphone más delgado de su época, con solo 9,9 mm de grosor, y al incorporar el procesador gráfico más rápido disponible. También presentó el Samsung SCH-R900, el primer smartphone 4G, marcando el inicio de la era de la conectividad móvil de alta velocidad. Todo esto ocurrió en 2010.

Motorola lanzó el Razr, batió a Samsung con el teléfono más fino con solo 7,1 mm, y Apple lanzó el iPhone 4S con Siri, el primer asistente personal de voz digital en un Smartphone. Sony Ericsson se lanzó al mercado de los juegos móviles y creó el Sony Ericsson Xperia Play, un teléfono similar a una videoconsola centrada en la experiencia de juego. (González, 2023, Párr. 22)

Sin embargo, Motorola venció a Samsung, lanzando el Razr, el teléfono más fino con tal solo 7,1 mm, por otro lado, se Apple lanzó el iPhone 4S con Siri, el primer asistente personal con voz digital en un Smartphone innovando así la industria. Para este entonces Sony Ericsson creó el Sony Ericsson Xperia Play lanzándose al mercado de videojuegos, contando con un teléfono similar a una videoconsola centrada en los videojuegos.

Este periodo también marcó el auge del formato plegable. Técnicamente, el Royole FlexPai ostenta el récord de ser el primer teléfono plegable de la historia. Sin embargo, no fue hasta el lanzamiento del Motorola Razr y el Samsung Galaxy Z Fold en 2019 cuando el mercado realmente despegó en serio. (González, 2023, párr. 24)

Así mismo, durante esta etapa, el formato plegable alcanzó un gran auge. Aunque el Royole FlexPai ostenta el récord de ser el primer teléfono plegable de la historia, no fue hasta el lanzamiento del Motorola Razr y el Samsung Galaxy Z Fold en 2019 que el mercado realmente comenzó a despegar y a ganar impulso, captando la atención de consumidores y desarrolladores por igual.

3.4 Creación de Celulares Android para Videojuegos

El Samsung Galaxy S24 Ultra 5G Dual SIM es la elección perfecta. Con su pantalla LTPO OLED de 6.8", procesador Snapdragon 8 Gen 3 y una tasa de refresco de 120 Hz, ofrece una experiencia de juego fluida y envolvente. Además, su batería de 5000 mAh con carga rápida de 45W te mantendrá en acción durante horas. Con una potente cámara principal de 200 + 50 + 10 + 12 MP y cámara frontal de 12 MP, también captura cada momento con claridad. (Quiróz, 2024, párr.3)

Para comenzar, el Samsung S24 Ultra 5G Dual SIM está diseñado para videojuegos, contando con una pantalla LTPO OLED de 6.8" y un procesador Snapdragon 8 Gen 3, ofreciendo así una experiencia de juego fluida. Además, con su batería de 5000 mAh con carga rápida de 45W te mantendrá pegado durante horas. Así mismo con una cámara potente de 200 + 50 +10 +12 MP y con cámara frontal de 12 MP.

El Apple iPhone 14 se erige como un dispositivo excepcional para los entusiastas del gaming. Su pantalla Super Retina XDR OLED de 6.1 pulgadas, acompañada de una resolución de 2532x1170p, ofrece una experiencia visual inigualable. Equipado con el potente procesador Apple A15 Bionic y una GPU propia de Apple, este teléfono garantiza un rendimiento óptimo incluso en los juegos más exigentes. Aunque su tasa de refresco es de 60 Hz, la optimización tanto del hardware como del software de Apple asegura una ejecución suave y sin interrupciones. (Quiroz, 2024, párr. 4)

Mientras tanto, se encuentra el iPhone 14 de Apple que se erige como un dispositivo para los usuarios del *gaming*. Contando con una pantalla Super Retina XDR OLED de 6.1 pulgadas, con una resolución de 2532x1170p ofreciendo una experiencia visual envidiable. Así mismo acompañado de un procesador Apple A15 Bionic y una GPU propia de Apple, garantizando un rendimiento óptimo tanto hardware como software sin interrupciones.

El Xiaomi Redmi Note 13 Pro+ 5G ofrece una experiencia de juego y rendimiento excepcional a un precio asequible. Con su pantalla AMOLED de 6.67 pulgadas y una impresionante resolución de 2712x1220p, sumérgete en colores vibrantes y detalles nítidos. El procesador MediaTek Dimensity 7200 Ultra y la GPU Mali-G610 MC6, combinados con una increíble RAM de 12GB, garantizan un rendimiento fluido incluso en los juegos más exigentes. (Quiróz, 2024, párr. 7)

Así mismo, el Xiaomi Redmi Note 13 Pro+ 5G nos proporciona una experiencia y rendimiento excepcional con un precio accesible. Con una pantalla AMOLED de 6.67 pulgadas y una resolución de 2712x1220p, contando con colores vibrantes y detalles nítidos. El procesador MediaTek Dimensity 7200 Ultra y la GPU Mali-G610 MC6, haciendo una increíble RAM, garantizando un rendimiento fluido.

El Xiaomi Redmi Note 13 Pro 5G ofrece un rendimiento sólido y características destacadas a un precio atractivo. Con una pantalla AMOLED de 6.67 pulgadas y una resolución de 2712x1220p, disfrutarás de imágenes nítidas y colores vibrantes para una experiencia visual envolvente. Impulsado por el procesador Snapdragon 7s Gen 2 y la GPU Adreno 710, junto con una generosa RAM de 12GB, este dispositivo garantiza un rendimiento suave y sin problemas, incluso en juegos exigentes. (Quiróz, 2024, párr. 10)

Por último, tenemos el Xiaomi Redmi Note 13 Pro 5G ofrece un rendimiento solido con un precio atractivo. Con la una pantalla AMOLED de 6.67 pulgadas y una resolución de 2712x1220p, con el cual disfrutaras de imágenes nítidas y colores vibrantes con una buena experiencia visual. Impulsado por el procesador Snapdragon 7s Gen 2 y la GPU Adreno 710, contando con una RAM de 12GB, garantizando un rendimiento suave.

3.5 Versiones y sus Actualizaciones de Android

Todo comenzó allá por el año 2003, cuando unos jóvenes Andy Rubin, Chris White, Rich Miner y Nick Sears se embarcaron en el proyecto Android, comenzando por la fundación de la empresa Android Inc. Su objetivo en aquellos primeros días, era dedicar sus esfuerzos a la creación de un sistema operativo orientado a mejorar la experiencia de los usuarios de cámaras digitales -- dispositivos en pleno auge por aquella época--. Más tarde, en 2005, Google se cruzó en el camino de estos tres emprendedores, adquiriendo la empresa fundada solo unos años atrás. (Martínez, 2017, párr.2)

A partir del año 2003, unos jóvenes Andy Rubin, Chris White, Rich Miner y Nick Sears se embarcaron en el proyecto llamado Android, iniciando así con su fundación de la empresa Android Inc. Buscaban crear un sistema operativo orientado a mejorar la experiencia de los usuarios de

cámaras digitales. En 2005 Google adquirió la empresa fundada por estos chicos.

Esta primera versión llegó bajo el nombre de Android 0.5 Milestone 3, la interfaz estaba adaptada a terminales de pantalla "pequeña", y solo se incorporaban aplicaciones como Google Maps, un navegador, y otras herramientas esenciales de un teléfono. Sin embargo, no sería hasta Android 0.5 Milestone 5, cuando realmente se comenzaron a asentar las bases de lo que Android es hoy en día. (Martínez, 2017, párr. 6)

Por consecuencia, la primera versión llegó bajo el nombre de Android 0.5 Milestone 3, estaba adaptada a terminales de pantalla “pequeña”, pero solamente incorporaba aplicaciones como Google Maps, un navegador y unas más esenciales de un teléfono. Pero llegó su actualización el Android 0.5 Milestone 5 donde se comenzó a asentar las bases de lo que hoy conocemos de Android.

Android 0.9 Beta fue la última versión de pruebas antes de que el sistema operativo viese la luz públicamente, y fue la versión que introdujo algunos de los añadidos característicos de la que, solo 10 años más tarde, sería la plataforma móvil más usada en el planeta. (Martínez, 2017, párr. 10)

Después de esto, la versión Android 0.9 Beta fue la última etapa de pruebas antes de que el sistema operativo se hiciera disponible para el público en general. En esta versión se implementaron varias de las características que más tarde se consolidarían como fundamentales para el sistema. Estas mejoras jugaron un papel clave en el crecimiento y la expansión de Android, que, solo 10 años después, alcanzaría el estatus de ser la plataforma móvil más utilizada en el mundo.

Llega octubre de 2008, y llega el día en el que Android por fin se sitúa en el punto de madurez que le permite dar el salto a un hardware real, orientado al público de a pie, gracias al primer teléfono en salir al mercado con Android en su interior: el *HTC Dream* o Google G1. (Martínez, 2017, párr. 14)

Ahora bien, en octubre de 2008, Android alcanzó un hito importante al situarse en el punto de madurez necesario para dar el salto a un hardware real, dirigido al público general. Este momento clave llegó con el lanzamiento del primer teléfono con Android en su interior: el HTC Dream,

también conocido como Google G1. Este dispositivo marcó el comienzo de la expansión de Android en el mercado de los *smartphones*.

Android 1.1 está considerada la primera actualización incremental del sistema operativo, dado que llegaba para solucionar la mayoría de errores descubiertos en Android 1. Aún así, Google la bautizó con un nuevo nombre, y entre sus novedades destaca el añadido de la búsqueda por voz gracias a *Google Voice Search*, cuyo funcionamiento consistía en comandos de voz que posteriormente se traducirían en búsquedas en el buscador. (Martínez, 2017, párr. 18)

Luego de esto, Android 1.1 es considerada la primera actualización incremental del sistema operativo, llegaba para solucionar los errores descubiertos en Android 1, aun así Google la nombro con su nuevo nombre, lo que la caracteriza fue el añadido de la búsqueda por voz gracias a Google Voice Search, mediante comandos de voz, que se convertirían en búsquedas en el buscador.

Android Donut fue la primera versión del sistema operativo en incluir soporte para diferentes tamaños de pantalla, sentando así los cimientos del que más tarde se convertiría en el slogan de Android: *Be together, not the same*. Y es que, de este modo, los fabricantes vieron cómo se abría una veda a la creación de teléfonos de diferentes tipos. (Martínez, 2017, párr. 27)

Para este entonces, Android Donut, el sistema operativo dio un paso importante al incluir soporte para diferentes tamaños de pantalla, lo que permitió una mayor flexibilidad para los fabricantes. Esta versión sentó las bases de lo que más tarde se transformaría en el lema de Android: "*Be together, not the same*". Gracias a esta capacidad de adaptación, los fabricantes tuvieron la libertad de desarrollar teléfonos de diversas formas y tamaños, lo que permitió una amplia variedad de dispositivos en el mercado y ayudó a consolidar la presencia de Android.

Creo que todos estamos de acuerdo en que Android 5.0 Lollipop, presentada en noviembre de 2014, fue la versión de Android más importante en toda la historia del sistema operativo. En primer lugar, se creó por primera vez el programa Developer Preview, que permitía a desarrolladores y entusiastas probar las últimas novedades antes que nadie, dejando en secreto detalles como el número de la versión o el nombre definitivo. Así nació "Android L", y junto a ella, las líneas de diseño Material Design. (Martínez, 2017, párr.74)

A partir de ese punto, Android 5.0 Lollipop llegaba en Noviembre del 2014 fue la versión más importante en toda la historia del sistema operativo, conteniendo por primera vez el programa Developer Preview, permitiendo así a los desarrolladores a probar nuevas novedades antes que nadie. Así nació Android L y junto a ella, las líneas de diseño Material Design.

Material Design fue el punto de inflexión más grande de la historia de Android, y un añadido histórico para todo el ecosistema de Google. Fue anunciado por el propio Matías Duarte en el Google I/O de 2014, y lo anunció como un nuevo estilo de diseño tanto para Android, como para el resto de productos y servicios de los de *Mountain View*. La idea principal era que el sistema operativo, y las aplicaciones, webs y diferentes plataformas, tuviesen una misma apariencia, cuyas interfaces estuvieran basadas en elementos físicos metafóricos como la tinta y el papel, incluyendo sombras, texturas y elevaciones virtuales sobre un lienzo tridimensional. (Martínez, 2017, párr. 75)

Así mismo, Material Design es el punto de inflexión más grande de Android, y añadido histórico para el ecosistema de Google. Anunciado en 2014, como un nuevo estilo de diseño tanto para Android para el resto de servicios, la idea principal era que el sistema operativo, aplicaciones, webs, etc. Tuvieran la misma estética, cuyas interfaces estuvieran basadas en elementos físicos, incluyendo sombras, texturas y demás.

En febrero de 2022, sin haber avisado a nadie, Google presentó Android 13, la última gran actualización del sistema operativo. Desde ese momento, era posible descargar la primera versión previa para desarrolladores para instalar Android 13 en un móvil compatible. Tras la gran actualización de la plataforma que supuso Android 12, Android 13 se centra, sobre todo, en introducir mejoras de seguridad y privacidad, incluyendo un nuevo menú de selección de imágenes que permite compartir contenido de una manera más segura. También se añaden cambios dirigidos a mejorar la experiencia de usuario con Material *You*, por ejemplo, dando soporte a iconos tematizados para apps de terceros. Otra novedad importante es la posibilidad de elegir un idioma diferente para cada aplicación, independiente del idioma del propio sistema operativo. (Martínez, 2017, párr. 17)

Para finalizar, en febrero del 2022, Google presento Android 13, la última gran actualización del

sistema operativo. Se centró sobre todo en introducir mejoras en seguridad y privacidad, incluyendo menú nuevo de selección de imágenes permitiendo así compartir contenido de forma más segura. También incluyendo mejoras de la experiencia de usuario con Material You, dando soporte a iconos tematizados. La última novedad importante fue la posibilidad de elegir el idioma en cada aplicación.

3.6 Procesadores en los Dispositivos Android

El NDK de Android es una herramienta poderosa que proporciona a los desarrolladores la capacidad de escribir en código nativo las partes de sus aplicaciones que son críticas en términos de rendimiento. Esto puede ser crucial para los desarrolladores de juegos, ya que los juegos a menudo requieren más potencia de procesamiento que las aplicaciones típicas, particularmente en las áreas de gráficos y física. Al usar el NDK, los desarrolladores pueden aprovechar todo el poder del hardware del dispositivo, incluyendo su CPU y GPU, para crear juegos más sofisticados y receptivos. (Kosarevsky y Latypov, 2019, p.122)

De modo que, el NDK de Android permite a los desarrolladores escribir partes críticas de sus aplicaciones en código nativo, lo cual es vital para los juegos que requieren más potencia de procesamiento, especialmente en gráficos y física. Esto les permite utilizar completamente la capacidad del hardware del dispositivo, incluyendo tanto la CPU como la GPU, para desarrollar juegos más avanzados y con mejor respuesta.

Uno de los beneficios más significativos del NDK de Android es que permite a los desarrolladores usar C++ para escribir la lógica de sus juegos. C++ es un lenguaje poderoso y eficiente que ofrece un control detallado sobre los recursos del sistema, lo cual es esencial para aplicaciones intensivas en rendimiento como los juegos. Al integrar C++ con Java a través del NDK, los desarrolladores pueden crear aplicaciones híbridas que combinan la facilidad de desarrollo en Java con las ventajas de rendimiento del código nativo. (Kosarevsky y Latypov, 2019, p.125)

Una de las principales ventajas del NDK de Android es la posibilidad de usar C++ para desarrollar la lógica del juego. C++ es un lenguaje muy eficiente que proporciona un control detallado sobre los recursos del sistema, lo cual es crucial para aplicaciones que demandan alto rendimiento como

los juegos. Al combinar C++ con Java mediante el NDK, los desarrolladores pueden crear aplicaciones híbridas que ofrecen tanto facilidad de desarrollo como mejoras en el rendimiento.

La gestión de recursos es un aspecto crítico del desarrollo de juegos, especialmente en dispositivos móviles con recursos limitados. El NDK de Android proporciona varias herramientas y bibliotecas que ayudan a los desarrolladores a gestionar la memoria, procesar hilos y optimizar el rendimiento. Al usar estas herramientas de manera efectiva, los desarrolladores pueden asegurarse de que sus juegos funcionen de manera fluida y eficiente, proporcionando una mejor experiencia a los usuarios. Técnicas como el uso de bibliotecas nativas para cálculos de física, la optimización de los canales de renderizado y la gestión de tareas en segundo plano pueden mejorar significativamente el rendimiento de los juegos en Android. (Kosarevsky y Latypov, 2019, p.129)

Por otra parte, la gestión adecuada de los recursos es esencial en el desarrollo de juegos para dispositivos móviles debido a sus limitaciones. El NDK de Android ofrece herramientas y bibliotecas que ayudan a gestionar la memoria, procesar hilos y optimizar el rendimiento. Utilizando estas herramientas, los desarrolladores pueden asegurarse de que sus juegos funcionen de manera fluida y eficiente. Esto incluye el uso de bibliotecas nativas para cálculos complejos, la optimización del renderizado y la gestión de tareas en segundo plano.

3.7 Factores que Influyen en una Buena Optimización en los Dispositivos Android

Threading es una herramienta poderosa en la caja de herramientas del desarrollador de Android, que permite a las aplicaciones realizar tareas en segundo plano mientras mantiene una interfaz de usuario receptiva. El uso adecuado de threading puede mejorar significativamente la experiencia del usuario al garantizar que las tareas de larga duración, como solicitudes de red o operaciones de base de datos, no bloquee el hilo principal de la interfaz de usuario. (Goransson, 2021, p.203)

Indicando que, el uso de hilos es una herramienta crucial para los desarrolladores de Android, ya que permite que las aplicaciones realicen tareas en segundo plano sin afectar la capacidad de respuesta de la interfaz de usuario. Utilizar correctamente los hilos puede mejorar notablemente la experiencia del usuario, asegurando que las tareas de larga duración, como solicitudes de red u

operaciones con bases de datos, no bloqueen el hilo principal de la UI.

El procesamiento asincrónico en Android no se trata sólo de ejecutar código en segundo plano. Implica una gestión cuidadosa de los recursos, el manejo de la sincronización de subprocesos y garantizar que la interfaz de usuario se actualice de forma segura desde los subprocesos en segundo plano. Esta complejidad requiere una comprensión sólida de la concurrencia y de Android. modelo de roscado. (Goransson, 2021, p.207)

Sin embargo, el procesamiento de ejecución independiente no solo se encarga de ejecutar código en segundo plano. También implica la gestión cuidadosa de los recursos, el manejo de la sincronización de hilos y la garantía de que la interfaz de usuario se actualice de manera segura desde los hilos en segundo plano. Esta complejidad requiere una comprensión sólida de la concurrencia y del modelo de hilos de Android.

Las clases Handler y Looper en Android proporcionan un mecanismo para la comunicación entre subprocesos. Al utilizar un Looper, un subproceso puede ejecutar un bucle de mensajes, lo que le permite procesar mensajes y ejecutables desde una cola. Esto es esencial para crear subprocesos de trabajo que puedan realizar tareas en segundo plano y comunicar los resultados al hilo principal. (Goransson, 2021, p.311)

Asimismo, las lecciones que dan Handler y Looper en Android ofrecen un mecanismo para la comunicación entre hilos. Al utilizar un Looper, un hilo puede ejecutar un bucle de mensajes, permitiéndole procesar mensajes y runnables desde una cola. Esto es esencial para crear hilos de trabajo que puedan realizar tareas en segundo plano y comunicar los resultados de vuelta al hilo principal.

3.8 Relación de la Memoria RAM en el Rendimiento de los Dispositivos

Android

Uno de los problemas más comunes a los que se enfrentan los desarrolladores de Android es el problema de las pérdidas de memoria. Las pérdidas de memoria se producen cuando una aplicación retiene memoria que ya no necesita, lo que impide que el sistema la recupere. Esto puede provocar

un mayor uso de la memoria y un rendimiento más lento. y, eventualmente, errores de falta de memoria. Para evitar pérdidas de memoria, los desarrolladores deben ser diligentes a la hora de publicar referencias a objetos que ya no son necesarios y utilizar herramientas como LeakCanary para detectar y corregir fugas durante el desarrollo. (Dobie y McAnlis, 2017, p. 112)

No obstante, los problemas que ocurren con más frecuencia a los desarrolladores de Android es el problema de las fugas de memoria. Las fugas de memoria ocurren cuando una aplicación retiene memoria que ya no necesita, impidiendo que el sistema la recupere. Para prevenir fugas de memoria, los desarrolladores deben ser diligentes en liberar referencias a objetos que ya no se necesitan y usar herramientas para detectar y corregir fugas durante el desarrollo.

La recolección de basura es una función de administración automática de memoria en Android, pero puede presentar problemas de rendimiento si no se maneja adecuadamente. Los ciclos frecuentes de recolección de basura pueden causar pausas notables en su aplicación, lo que lleva a una mala experiencia del usuario. Los desarrolladores deben intentar minimizar la creación de objetos de corta duración y reducir la huella de memoria general de sus aplicaciones para mantener los tiempos de recolección de basura al mínimo. (Dobie y McAnlis, 2017, p. 205)

Aunque, el recolectar la basura es una función de gestión automática de memoria en Android, pero puede introducir problemas de rendimiento si no se maneja correctamente. Los ciclos frecuentes de recolección de basura pueden causar pausas notables en tu aplicación, llevando a una mala experiencia de usuario. Los desarrolladores deben intentar minimizar la creación de objetos de corta duración y reducir la huella de memoria total de sus aplicaciones para mantener los tiempos de recolección de basura al mínimo.

Optimizar el uso de la memoria en Android no se trata sólo de evitar fugas y reducir las asignaciones. También se trata de comprender cómo interactúan entre sí los diferentes componentes del sistema. Por ejemplo, el manejo de mapas de bits, las estrategias de almacenamiento en caché y el uso de servicios pueden afectar el rendimiento de su aplicación". Perfil de memoria. Los desarrolladores deben considerar el impacto holístico de sus patrones de uso de memoria y optimizarlos en consecuencia para lograr el mejor rendimiento. (Dobie y McAnlis, 2017, p. 215)

Optimizar el uso de la memoria en Android no se trata solo de evitar fugas y reducir asignaciones. También se trata de entender cómo interactúan los diferentes componentes del sistema entre sí. Por ejemplo, el manejo de bitmaps, las estrategias de almacenamiento en caché y el uso de servicios pueden afectar el perfil de memoria de tu aplicación. Los desarrolladores deben considerar el impacto holístico de sus patrones de uso de memoria y optimizar en consecuencia para lograr el mejor rendimiento.

3.9 Como Influye la Batería en el Rendimiento de los Dispositivos Android

La gestión eficiente de la batería es crucial para asegurar un rendimiento óptimo en dispositivos Android. El consumo excesivo de energía por parte de aplicaciones mal optimizadas puede provocar una disminución significativa en la duración de la batería y afectar negativamente la capacidad de respuesta del dispositivo. Para mitigar estos problemas, es esencial implementar políticas de gestión de energía que equilibren la funcionalidad con la eficiencia energética, optimizando el uso de recursos como la CPU, la red y los sensores. (Firtman, 2014, p.297)

Además, es fundamental gestionar eficazmente el uso de la batería para mantener un rendimiento óptimo en dispositivos Android. Cuando las aplicaciones no están bien optimizadas y consumen demasiada energía, esto puede reducir drásticamente la duración de la batería y afectar la capacidad de respuesta del dispositivo. Para abordar estos problemas, es necesario implementar estrategias de gestión de energía que busquen un equilibrio entre la funcionalidad deseada y la eficiencia energética, optimizando cómo se utilizan recursos críticos como la CPU, la red y los sensores.

El impacto del consumo de batería en el rendimiento de los dispositivos Android es multifacético. Factores como la ejecución de procesos intensivos en segundo plano, la frecuencia de actualización de datos y el manejo ineficiente de la conectividad pueden llevar a una degradación significativa del rendimiento y una reducción en la vida útil de la batería. Los desarrolladores deben adoptar enfoques proactivos, utilizando herramientas de monitoreo de energía y técnicas de optimización avanzadas para identificar y corregir los puntos críticos que afectan la eficiencia energética y la experiencia del usuario. (Firtman, 2014, p.285)

En cambio, el impacto del consumo de batería en dispositivos Android es complejo y afecta diversos aspectos del rendimiento del dispositivo. Tareas como la ejecución intensiva de procesos en segundo plano y una gestión ineficiente de la conectividad pueden causar una disminución notable en el rendimiento general y en la autonomía de la batería. Es crucial que los desarrolladores adopten un enfoque proactivo mediante el uso de herramientas de monitoreo de energía y la aplicación de técnicas avanzadas de optimización para identificar y resolver eficazmente los problemas que comprometen la eficiencia energética y la satisfacción del usuario.

La optimización del consumo de batería no solo mejora la autonomía del dispositivo Android, sino que también es fundamental para mantener la estabilidad del sistema y la satisfacción del usuario. Estrategias como la gestión inteligente de procesos en segundo plano, la minimización de la frecuencia de actualización de datos y el uso eficiente de los sensores juegan un papel crucial en la creación de aplicaciones responsables que no solo funcionen bien, sino que también preserven los recursos energéticos del dispositivo. (Firtman,2014, p.305)

También, optimizar el consumo de batería en dispositivos Android es esencial no solo para prolongar la vida útil de la batería, sino también para garantizar la estabilidad del sistema y la satisfacción del usuario. Implementar estrategias como la gestión cuidadosa de procesos en segundo plano, reducir la frecuencia de actualizaciones de datos y usar de manera eficiente los sensores son prácticas fundamentales para desarrollar aplicaciones responsables que mantengan un equilibrio.

3.10 Memoria RAM

La memoria RAM es donde el ordenador almacena temporalmente los datos y los programas con los que está trabajando en un momento dado. Todo lo que hay en ella almacenado se borra cuando apagamos o reiniciamos el ordenador. La memoria RAM es un componente imprescindible para el ordenador, almacena temporalmente las instrucciones y los datos para que la CPU pueda procesarlos. (González, 2020, p.1)

Por lo tanto, el ordenador dependerá de una serie de componentes esenciales para su funcionamiento eficiente, y uno de los más importantes es el almacenamiento temporal de datos. Esto se logra mediante la memoria de acceso aleatorio (RAM), que se encarga de almacenar de

manera temporal los datos y programas con los que el sistema está trabajando en ese momento. Todo lo que se guarda en la RAM se borra cuando el equipo se apaga o reinicia. La memoria RAM permite que la CPU acceda rápidamente a las instrucciones.

Se puede comparar a un gran casillero. Cada casilla, denominada posición de memoria, está formada por ocho bits de forma que en ella se puede escribir un carácter (un byte). El microprocesador debe saber exactamente la posición en memoria de cada dato, por lo que las posiciones están identificadas por un número denominado dirección de memoria. Físicamente, la memoria RAM es una plaquita rectangular de circuito impreso con varios chips, que se acopla a la placa base a través de una ranura específica. (González, 2020, p.1)

Así, la memoria RAM puede compararse con un casillero grande, donde cada casilla o posición de memoria está compuesta por ocho bits y puede almacenar un carácter. Para que el microprocesador acceda a los datos, necesita conocer la posición exacta en la memoria, identificada por un número conocido como dirección de memoria. Físicamente, la memoria RAM es una placa rectangular con chips integrados, que se conecta a la placa base mediante una ranura específica.

3.11 Características de Memorias

La capacidad para almacenar datos expresados en MB o GB. Existen módulos de distintas capacidades (128 MB, 256 MB, 512 MB, 1 GB y 2 GB). Es preferible adquirir un ordenador con toda la memoria RAM concentrada en un solo módulo, para disponer de más ranuras libres, por si queremos ampliar la memoria RAM en un futuro. La capacidad de un módulo de memoria viene determinada por la capacidad de cada chip y por el número de chips que incluya, que puede variar entre 2 y 16. (González, 2020, p.1)

Por consiguiente, hay módulos de memoria de diversas capacidades (128 MB, 256 MB, 512 MB, 1 GB y 2 GB). Es recomendable optar por un ordenador con toda la memoria RAM en un único módulo, lo que deja más ranuras disponibles para futuras ampliaciones. La capacidad de cada módulo depende tanto de la capacidad individual de los chips como del número de chips que contenga, oscilando entre 2 y 16.

3.12 Tipo de Memoria

Actualmente, la mayoría de los ordenadores utilizan memorias del tipo DIMM, concretamente la serie DDR-SDRAM, pero también existen módulos del tipo DDR2-SDRAM, que funcionan de forma semejante a los DDR, pero transfieren más paquetes de datos por cada ciclo de reloj. (González, 2020, p.2)

Por lo tanto, aunque en la actualidad la mayoría de los ordenadores utiliza memorias DIMM, concretamente de la serie DDR-SDRAM, también están disponibles módulos DDR2-SDRAM. Estos módulos operan de manera similar a los DDR-SDRAM, pero se diferencian por su capacidad para transferir un mayor número de paquetes de datos en cada ciclo de reloj. Esta mejora en la transferencia de datos resulta en un rendimiento más eficiente y rápido, beneficiando especialmente aplicaciones que requieren alta velocidad de procesamiento y mayor ancho de banda.

Actualmente se suelen emplear las memorias DDR3-SDRAM transfiere paquetes de datos más rápido que las dos anteriores y permite usar módulos de 512 MB a 8 GB, siendo posible fabricar módulos de hasta 16 GB. Estos tres tipos de memoria son incompatibles entre sí, por lo que sólo pueden conectarse en placas bases con las ranuras de conexión adecuadas. (González, 2020, p.2)

En consecuencia, las memorias DDR3-SDRAM son comúnmente utilizadas en la actualidad debido a su capacidad de transferir paquetes de datos más rápidamente que las versiones anteriores. Estos módulos permiten capacidades que van desde 512 MB hasta 8 GB, con la posibilidad de alcanzar hasta 16 GB por módulo. Sin embargo, es importante destacar que los tres tipos de memoria (DDR, DDR2, y DDR3) no son compatibles entre sí y sólo pueden instalarse en placas base que tengan las ranuras específicas para cada tipo.

“Expresada en MHz. Hay que distinguir entre la frecuencia de reloj a la que trabaja la memoria (frecuencia interna) con la frecuencia a la que transmite los paquetes de datos (frecuencia externa efectiva)” (González, 2020, p.2).

Además, es esencial distinguir entre dos tipos de frecuencias asociadas a la memoria RAM. Por un lado, la frecuencia de reloj interna indica la velocidad a la que la memoria trabaja internamente, determinada en MHz. Por otro lado, la frecuencia externa efectiva se refiere a la velocidad real a la que la memoria transmite los paquetes de datos hacia y desde otros componentes del sistema,

también medida en MHz. Esta diferencia es crucial para entender cómo se maneja y se optimiza el rendimiento de la memoria en un sistema informático.

3.13 Android

Android no diferencia entre el núcleo del teléfono y las aplicaciones de terceros. Todas pueden ser construidas para tener igual acceso a un teléfono y tienen la capacidad de ofrecer a los usuarios un amplio espectro de aplicaciones y servicios. Con los dispositivos construidos en la Plataforma Android, los usuarios podrán adaptarse plenamente al teléfono para sus intereses. (Cruz, 2008, p.3)

Por otro lado, Android no separa el núcleo del sistema operativo de las aplicaciones de terceros, permitiendo que ambas tengan el mismo nivel de acceso al teléfono. Esto abre la puerta a una enorme variedad de aplicaciones y servicios que pueden ser creados para mejorar la experiencia del usuario. Los dispositivos que utilizan la plataforma Android permiten una personalización total, adaptándose a los intereses y necesidades individuales de cada usuario.

Los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede hacer luego uso de esas capacidades (sujeto a reglas de seguridad del framework). Éste mismo mecanismo permite que los componentes sean reemplazados por el usuario. (Cruz, 2008, p.6)

En resumen, los desarrolladores tienen acceso a los mismos APIs del framework que utilizan las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes, permitiendo que cualquier aplicación publique sus funcionalidades y que otras aplicaciones puedan utilizarlas, sujeto a las reglas de seguridad del framework. Este mecanismo también permite al usuario reemplazar los componentes.

“Android incluye un conjunto de librerías C/C++ usadas por varios componentes del sistema Android. Estas capacidades se exponen a los desarrolladores a través del framework de aplicaciones de Android” (Cruz, 2008, p.7).

En este contexto, Android proporciona una serie de bibliotecas C/C++ que son empleadas por distintos componentes del sistema operativo. Estas funcionalidades se ponen a disposición de los desarrolladores a través del framework de aplicaciones de Android, permitiéndoles acceder a capacidades avanzadas para optimizar el rendimiento y mejorar la experiencia del usuario.

Android incluye un conjunto de librerías base que proveen la mayor parte de las funcionalidades disponibles en las librerías base del lenguaje de programación Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. (Cruz, 2008, p.6)

Además, Android incluye un conjunto de librerías base que proporcionan la mayor parte de las funcionalidades presentes en las librerías estándar de Java. Cada aplicación Android se ejecuta en su propio proceso, utilizando una instancia independiente de la máquina virtual Dalvik para asegurar su funcionamiento autónomo y eficiente. Esto permite que cada aplicación tenga su propio entorno de ejecución aislado, lo que contribuye a la estabilidad y seguridad del sistema.

Android depende de un Linux versión 2.6 para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, *Stack* de red, y modelo de drivers. El núcleo también actúa como una capa de abstracción entre el hardware y el resto del *stack* de software. (Cruz, 2008, p.7)

Por consiguiente, Android incorpora un conjunto de bibliotecas base que proporcionan la mayoría de las funcionalidades encontradas en las librerías estándar de Java. Cada aplicación Android se ejecuta en su propio proceso, utilizando una instancia separada de la máquina virtual Dalvik, lo que garantiza un funcionamiento autónomo y eficiente. Este diseño permite que cada aplicación tenga un entorno de ejecución aislado, lo que contribuye significativamente a la estabilidad y seguridad del sistema operativo.

3.14 Desarrollo de Aplicaciones

Android permite a los desarrolladores obtener la ubicación del dispositivo. Android incluye un conjunto completo de herramientas que se han construido desde el inicio junto a la plataforma con los desarrolladores para proporcionar una elevada productividad y una profunda comprensión de sus aplicaciones. (Cruz, 2008, p.7)

Además de eso, Android brinda a los desarrolladores la capacidad de obtener la ubicación del dispositivo utilizando servicios y APIs especializadas. Además, la plataforma proporciona un conjunto completo de herramientas desarrolladas en estrecha colaboración con los programadores desde sus inicios. Estas herramientas están diseñadas para maximizar la productividad y ofrecer un profundo conocimiento sobre el funcionamiento de las aplicaciones, lo que facilita la creación de soluciones más eficientes y adaptadas a las necesidades del usuario.

Android permiten a los desarrolladores crear aplicaciones móviles y sacar el máximo provecho que todo dispositivo tiene que ofrecer. Está construido para ser realmente abierto. Por ejemplo, una aplicación puede llamar a cualquier función básica de un teléfono, como hacer llamadas, enviar mensajes de texto, o usar la cámara. Android se basa en el Kernel Linux. Además, utiliza una máquina virtual personalizada que ha sido diseñada para optimizar la memoria y los recursos de hardware en un entorno móvil. (Cruz, 2008, p.7)

Por otro lado, Android capacita a los desarrolladores para crear aplicaciones móviles que aprovechen al máximo todas las capacidades de cada dispositivo disponible. Esta plataforma se distingue por su enfoque abierto, permitiendo que las aplicaciones accedan a funciones esenciales del teléfono, como realizar llamadas, enviar mensajes de texto o utilizar la cámara. Android se basa en el Kernel Linux y emplea una máquina virtual personalizada diseñada específicamente para optimizar la gestión de memoria y los recursos de hardware en entornos móviles.

3.15 Sobrecalentamiento de Android

“Los dispositivos Android se sobrecalientan debido a la alta demanda de procesamiento de los últimos modelos de aplicaciones y juegos” (Smith, 2021, p.22).

Sin embargo, el sobrecalentamiento de los dispositivos Android puede ocurrir debido a la alta carga de procesamiento que exigen las aplicaciones y juegos más recientes. Estos programas, particularmente aquellos con gráficos complejos y una mayor demanda de recursos, pueden poner a prueba el rendimiento del hardware del dispositivo, lo que a menudo resulta en un incremento de la temperatura interna. También contribuyen factores como la multitarea, el uso extendido y las aplicaciones ejecutándose en segundo plano.

“El sobrecalentamiento en Android también puede deberse a problemas de software como aplicaciones mal optimizadas o procesos en segundo plano que consumen demasiados recursos” (García, 2022, p.2).

Por lo tanto, una de las principales causas del sobrecalentamiento en los dispositivos Android puede estar relacionada con el software, especialmente cuando hay aplicaciones mal optimizadas o procesos en segundo plano que consumen una cantidad excesiva de recursos del sistema. Las aplicaciones que no están diseñadas de manera eficiente pueden sobrecargar el procesador, la memoria y la batería, lo que lleva a un aumento de la temperatura del dispositivo.

“Otro factor que contribuye al sobrecalentamiento es el uso de pantallas grandes y brillantes, especialmente en dispositivos con procesadores de gama alta” (Martínez, 2020, p.3).

Asimismo, el uso de pantallas grandes y brillantes, especialmente en dispositivos con procesadores de gama alta, también juega un papel importante en el sobrecalentamiento. Las pantallas de mayor tamaño requieren más energía para mantener su brillo y resolución, lo que a su vez incrementa la carga sobre la batería y el procesador. Cuando se combinan con chips más potentes, destinados a soportar aplicaciones y juegos de alto rendimiento, esto puede generar un calor significativo.

“La falta de un sistema de refrigeración adecuado en algunos modelos de gama media y baja de Android propicia que se calienten más rápido durante el uso intensivo” (Sánchez, 2021, p.4).

No obstante, en muchos dispositivos Android de gama media y baja, la ausencia de un sistema de refrigeración adecuado puede contribuir significativamente al sobrecalentamiento durante el uso intensivo. A diferencia de los modelos de gama alta que a menudo incorporan tecnologías de disipación de calor, como sistemas de enfriamiento líquido o estructuras térmicas avanzadas, los dispositivos más asequibles no cuentan con estas características. Esto significa que, al ejecutar aplicaciones o juegos que demandan más potencia de procesamiento.

“Mantener el dispositivo Android en un estuche o funda cerrada puede impedir la disipación del calor y provocar sobrecalentamiento, especialmente en ambientes cálidos” (Ramírez, 2022, p.5).

Del mismo modo, el uso de un estuche o funda cerrada en un dispositivo Android puede afectar negativamente la capacidad del dispositivo para disipar el calor de manera adecuada. Estas fundas, especialmente las de material no transpirable, actúan como una barrera que atrapa el calor generado por el procesador y otros componentes internos. En condiciones normales, los teléfonos están diseñados para liberar el calor a través de sus bordes y la parte posterior, pero si están cubiertos por una funda cerrada, se dificulta este proceso.

“Algunos usuarios reportan que actualizar a la última versión de Android en ciertos modelos más antiguos empeora el problema del sobrecalentamiento” (Gómez, 2020, p.15).

De manera similar, los usuarios han informado que actualizar a la última versión de Android en dispositivos más antiguos puede, irónicamente, empeorar el problema del sobrecalentamiento. Esto se debe a que las nuevas actualizaciones suelen incluir mejoras en la seguridad, la optimización del sistema y nuevas funcionalidades que pueden aumentar la demanda de recursos del hardware.

3.16 Calidad y Precio

“Los precios de los dispositivos Android pueden variar significativamente dependiendo de la marca, el modelo y las características” (García, 2022, p.13).

Igualmente, la calidad y precios de Android pueden variar ampliamente debido a varios factores, como la marca, el modelo, las características, el procesador, la cantidad de almacenamiento, la calidad de la cámara, el tamaño de la pantalla, y si el dispositivo pertenece a la gama baja, media o alta. Los fabricantes de Android ofrecen una amplia gama de opciones, desde dispositivos económicos hasta teléfonos premium con características avanzadas.

“Aunque los dispositivos Android de gama alta suelen ser más caros, también ofrecen mejoras significativas en el rendimiento y la calidad de la cámara” (Martínez, 2020, p.4).

Por otra parte, aunque los dispositivos Android de gama alta suelen tener precios más elevados, ofrecen ventajas notables en cuanto a rendimiento, calidad de la cámara y características

adicionales. Estos dispositivos suelen contar con procesadores más rápidos, pantallas de mayor resolución y cámaras con capacidades avanzadas, lo que resulta en una experiencia de usuario más fluida y de mayor calidad.

La calidad de los dispositivos Android puede variar dependiendo de la marca y el modelo, con algunas marcas ofreciendo dispositivos de alta calidad a precios más asequibles. (Sánchez, 2021, p.11)

Por esta razón, los teléfonos de gama baja suelen ser más económicos y, aunque ofrecen funciones esenciales como llamadas, mensajería y navegación por internet, pueden carecer de las características más avanzadas, como pantallas de alta resolución o cámaras de gran calidad. Por otro lado, los dispositivos Android de gama alta, aunque tienen un precio más elevado, están diseñados para ofrecer un rendimiento superior con procesadores rápidos, pantallas brillantes, cámaras de alta resolución y capacidades adicionales.

“Los dispositivos Android de gama baja pueden ser más accesibles, pero a menudo carecen de características y rendimiento que los dispositivos de gama alta ofrecen” (Ramírez, 2022, p.2).

Asimismo, los dispositivos de gama baja suelen ser más asequibles, pero a menudo sacrifican características avanzadas y un rendimiento superior en comparación con los de gama alta. Los dispositivos Android de gama baja, aunque son más accesibles, generalmente cuentan con hardware menos potente, cámaras de menor calidad y pantallas con menor resolución, lo que puede limitar la experiencia del usuario, especialmente en tareas que requieren un mayor procesamiento o en aplicaciones y juegos exigentes.

“La calidad de la cámara y el rendimiento de los dispositivos Android pueden ser importantes factores a considerar al elegir un dispositivo, especialmente si se utiliza para tomar fotos y videos” (Gómez, 2020, p.7).

En cambio, al elegir un dispositivo Android, la calidad de la cámara y el rendimiento son aspectos clave a tener en cuenta, especialmente si se planea utilizar el dispositivo para tomar fotos y videos. Los dispositivos de gama alta suelen estar equipados con cámaras de alta resolución, múltiples

lentes y características avanzadas, como la grabación en 4K o modos de fotografía en condiciones de poca luz.

“Los precios de los dispositivos Android pueden variar según la región y el proveedor, lo que puede afectar la elección del dispositivo” (Smith, 2021, p12).

De modo que, los precios pueden variar según la región y el proveedor, lo que influye en la elección del dispositivo. Dependiendo de dónde vivas, un dispositivo Android de gama alta puede ser más costoso debido a impuestos, tarifas de importación o la oferta limitada en ciertas áreas. Por otro lado, en algunas regiones, las promociones y descuentos de los proveedores pueden hacer que ciertos modelos sean más asequibles, lo que ofrece oportunidades para encontrar mejores precios o beneficios adicionales.

3.17 Preferencia de Usuarios a Android

“Android es el sistema operativo más popular debido a su flexibilidad y capacidad para ser personalizado” (García, 2022, p.4).

Por esta razón, Android es utilizado en una amplia gama de dispositivos, desde teléfonos y tabletas hasta televisores y dispositivos portátiles. Esta versatilidad ha sido clave para su éxito, ya que se adapta a una variedad de mercados y usuarios. Además, la disponibilidad de aplicaciones en la Google Play Store, junto con su integración con otros servicios de Google, hace que sea una opción atractiva para quienes buscan una experiencia personalizada y accesible.

“La gran variedad de dispositivos y marcas que ofrecen Android es otro factor que contribuye a su popularidad y a la calidad del sistema” (Martínez, 2020, p.5).

Del mismo modo, cada fabricante de dispositivos Android tiene la libertad de adaptar el sistema a sus necesidades, lo que resulta en una amplia variedad de teléfonos con diferentes características, precios y diseños. Esto permite a los usuarios elegir entre una gama de opciones que se ajusten a sus preferencias personales y presupuesto. Desde teléfonos de gama baja hasta modelos premium con características avanzadas, Android ha logrado abarcar un público diverso

“La capacidad de Android para integrarse con otros dispositivos y servicios es otro aspecto que lo hace atractivo para los usuarios” (Sánchez, 2021, p.6).

Al igual que la integración de Android, con una amplia gama de dispositivos y servicios es un factor fundamental para su popularidad. Gracias a su compatibilidad con dispositivos como relojes inteligentes, televisores, tabletas, auriculares y más, los usuarios pueden tener una experiencia fluida y conectada a través de diferentes plataformas. Además, la integración con servicios de Google, como Google Drive, Google Photos y Google Assistant.

“La gran cantidad de aplicaciones disponibles en la tienda de Google Play es otro factor que contribuye a la popularidad de Android” (Ramírez, 2022, p.5).

De manera similar, la extensa variedad de aplicaciones disponibles en Google Play es una de las principales razones por las cuales Android se ha consolidado como el sistema operativo líder. Con millones de aplicaciones en diversas categorías como entretenimiento, productividad, redes sociales, juegos y educación, los usuarios pueden personalizar su dispositivo según sus necesidades y gustos. Además, la facilidad para que los desarrolladores creen y publiquen sus aplicaciones en la plataforma fomenta una constante innovación y una oferta diversa de software.

“La capacidad de Android para ser actualizado y mejorado constantemente es otro aspecto que lo hace atractivo para los usuarios” (Gómez, 2020, p.8).

En cambio, para recibir actualizaciones periódicas es uno de los aspectos clave que contribuye a su éxito. Gracias a su naturaleza abierta y a la colaboración constante entre Google y los fabricantes de dispositivos, Android puede ser optimizado y mejorado continuamente, ofreciendo nuevas funciones, mejoras de seguridad y ajustes de rendimiento. Esto significa que, a pesar de que el hardware de un dispositivo pueda ser más antiguo, los usuarios aún pueden disfrutar de las últimas innovaciones.

“La gran cantidad de desarrolladores que trabajan en Android y la comunidad de usuarios activa son otros factores que contribuyen a su popularidad” (Smith, 2021, p.13).

En resumen, el éxito de Android también se debe en gran parte a la enorme comunidad de desarrolladores y usuarios activos que lo respaldan. Gracias a este ecosistema vibrante y colaborativo, Android ha logrado mantenerse como la plataforma más popular y en constante evolución. Los desarrolladores de todo el mundo aportan nuevas aplicaciones, actualizaciones y soluciones, mientras que los usuarios contribuyen con retroalimentación.

CAPÍTULO IV

Metodología de la Investigación

Capítulo IV. Metodología de la Investigación

4.1 Enfoque y Método

Para la presente investigación se partirá del enfoque mixto, el cual según Hernández (2003), “usa la recolección de datos para probar una hipótesis con base en la medición numérica y el análisis estadístico para establecer patrones de comportamiento de una población” (p. 6).

El alcance de la investigación es exploratorio. Exploratorio porque el objetivo es diagnosticar las características de Unity, Engine y GameMaker, lo que indica según los antecedentes de la investigación que es un tema poco estudiado, sobre el cual no existe una solución definida al respecto, porque busca establecer una relación entre las características de Unity, Engine y GameMaker en los motores gráficos. Los estudios correlacionales tienen “como propósito conocer la relación que exista entre dos o más conceptos, categorías o variables en un contexto particular” (Hernández, 2003, p. 6).

Si buscas en Google “créate a videogame” recibirás 5 mil millones de resultados. De todos Unity estará en 70 millones. Unreal Engine en 36 millones. GameMaker Studio en 2 millones. Va a hacer difícil – no imposible- que a un principiante alguien le muestre otro camino. Mucho menos que le recomiende crear su propio motor. (Marques, 2021, párr. 1)

4.2 Población y Muestra

Según Selltiz, en Sampieri (2003), “una población es el conjunto de todos los casos que concuerdan con una serie de especificaciones” (p. 303). En el caso de esta investigación, la población a tener en cuenta son 200 alumnos de la ingeniería en sistemas tomando en cuenta su conocimiento en motores gráficos. Para los estudios cuantitativos, no es más que un “subgrupo de la población del cual se recolectan los datos y debe ser representativo de dicha población”. (Sabino, 1992, p.302).

El muestreo realizado fue de tipo no probabilístico. En este sentido, Hernández (2003), acota que la muestra probabilística es el “subgrupo de la población en la que la elección de los elementos no depende de la probabilidad sino de las características de la investigación” (p. 306). Por su parte, según Sabino (1992), una muestra intencional es aquella que “escoge sus unidades no en forma fortuita sino completamente arbitraria designando a cada unidad según características que para el investigador resulten de relevancia” (p. 302). Por lo tanto, la muestra representativa para esta investigación es de 50 alumnos de la ingeniería en sistemas que fueron seleccionados tomando en cuenta su conocimiento en programación.



4.3 Instrumento de Medición

ESCUELA TEOLUYUCAN

“Constancia, Unidad y Trabajo”

**BACHILLERATO TECNOLÓGICO CON LA CARRERA TÉCNICA
EN PROGRAMACIÓN**

“Diagnóstico de las Características de Unity, Engine y Gamemaker para los Sistemas Operativos Android de Gama Media”

Motivo: Esta encuesta, escala Likert y entrevista se llevó a cabo para conocer las características de Unity, Engine y GameMaker y recopilar los datos sobre el tema de investigación, y junto con ella dar respuesta a la pregunta de investigación.

ENCUESTA

INSTRUCCIONES: Subraya o coloca una X, en la respuesta que considere adecuada.

1. ¿Para qué sirve un Motor Gráfico?

- a) Programar
- b) Jugar
- c) Diseñar
- d) Todas las anteriores

2. ¿En qué contextos ha oído hablar sobre Motores Gráficos?

- a) Películas
- b) Realidad Virtual
- c) Videojuegos
- d) otros

3. ¿Ha oído hablar de ellos, Cuáles conoce?

- a) Unity
- b) Godot
- c) Unreal Engine
- d) GameMaker

4. ¿Sabe cómo se crean los videojuegos?

- a) Si
- b) No
- c) Solo los juego

5. ¿Le gustaría realizar un videojuego?

- a) Si
- b) No
- c) Solo los juego

ENTREVISTA

- 1.- ¿Qué sabe sobre los motores gráficos?
- 2.- ¿Sabe cuál es la mejor opción entre los motores gráficos?
- 3.- ¿Por qué eligió el celular que tienes?
- 4.- ¿Investiga más sobre los videojuegos?
- 5.- ¿Conoce la diferencia entre motores gráficos?

	Preguntas	Siempre	Casi Siempre	Algunas Veces	Rara Vez	Nunca
1	¿Cada cuánto juego videojuegos?					
2	¿Qué tan seguido uso el teléfono?					
3	¿Al comprar un celular checo la gama?					
4	¿Suelo analizar con qué Motor gráfico está hecho mi videojuego favorito?					
5	¿Suelo analizar las características de un videojuego?					

CAPÍTULO V

Análisis de Resultado

Capítulo V. Análisis de Resultados

5.1 Matrices

En este apartado se dará a conocer el matriz, resultado de los instrumentos de medición aplicados a las personas de 18 y 25 años. En la Escuela Universidad Bancaria De México, del municipio de Teoloyucan, Estado de México, Avenida 5 de Mayo.

La matriz es un mecanismo de codificación que ayuda a los analistas para tener en cuenta varios puntos analíticos, entre los cuales están, qué condiciones o consecuencias, deben ser parte del análisis como significativas, así como las siguientes acciones, interacciones y consecuencias, pueden respetarse en los datos. (Strauss, Corbin, 2002, p.199).

A continuación, se presenta la matriz.

Encuesta:

	Preguntas	A)	B)	C)	D)
1	¿Para qué sirve un Motor Gráfico?	0	3	1	19
2	¿En qué contextos ha oído hablar sobre Motores Gráficos?	0	4	18	1
3	¿Ha oído hablar de ellos, cuáles conoce?	14	2	1	6
4	¿Sabe cómo se crearon los videojuegos?	13	4	6	0
5	¿Le gustaría realizar un videojuego?	17	1	5	0

Entrevistas:

1.- ¿Qué sabes sobre los motores gráficos?

(R-1) Que son para diseñar.

(R-2) Crea todo lo gráfico y videojuegos 3D.

(R-3) Son útiles para mejorar la calidad de la pantalla.

- (R-4) No sé mucho.
- (R-5) Que son los que dan calidad a la pantalla.
- (R-6) Nada.
- (R-7) Nada.
- (R-8) Que sirven para los videojuegos.
- (R-9) Son los que se pueden hacer los videojuegos.
- (R-10) Son con lo que se hacen los videojuegos.
- (R-11) Son lo que abarcan el juego en nivel visual.
- (R-12) Son programas para poder desarrollar videojuegos.
- (R-13) Son parte fundamental de las computadoras en cuanto calidad gráfica.
- (R-14) Que van más fluidos los juegos.
- (R-15) Son útiles.
- (R-16) Son las herramientas que facilitan crear videojuegos.
- (R-17) Sé conceptos básicos sobre ellos.
- (R-18) Es un programa que se encarga de usar y renderizar una imagen.
- (R-19) Es un programa encargado de procesar y modificar imágenes, etc.
- (R-20) Nada.
- (R-21) No mucho.
- (R-22) Sirven para diseñar.
- (R-23) Sirven para la función de videojuegos.

2.- ¿Sabes cuál es la mejor opción entre los motores gráficos?

- (R-1) La verdad no.
- (R-2) No.
- (R-3) No.
- (R-4) Si.
- (R-5) Si.
- (R-6) Unreal Engine (en mi opinión).
- (R-7) Depende tus necesidades.
- (R-8) No.
- (R-9) No.
- (R-10) No, realmente no.

- (R-11) No.
- (R-12) No.
- (R-13) Unity.
- (R-14) Unity.
- (R-15) No.
- (R-16) Unity.
- (R-17) No.
- (R-18) Unity.
- (R-19) No.
- (R-20) No.
- (R-21) No.
- (R-22) No.
- (R-23) No.

3.- ¿Por qué elegiste el celular que tienes?

- (R-1) No tenía más dinero.
- (R-2) Porque toma buen almacenamiento.
- (R-3) Por la capacidad de almacenamiento.
- (R-4) Presento un buen rendimiento.
- (R-5) Por su RAM, procesador.
- (R-6) Por el rendimiento.
- (R-7) Por la RAM.
- (R-8) Por calidad y precio.
- (R-9) Por el espacio y rendimiento.
- (R-10) Porque me lo regalaron.
- (R-11) Barato, eficiente y de calidad.
- (R-12) Por el espacio de la RAM y la ram, procesador y precio.
- (R-13) Por el rendimiento gráfico y su memoria.
- (R-14) Nomas porque me gusto su diseño.
- (R-15) Porque tiene mucha RAM.
- (R-16) Por el procesador.
- (R-17) Ni lo elegí yo.

(R-18) Por la cámara, RAM y espacio.

(R-19) El espacio.

(R-20) Por la marca, memoria y RAM.

(R-21) Porque estaba bien.

(R-22) Por su marca.

(R-23) Por su espacio.

4.- ¿Investigas más sobre los videojuegos?

(R-1) No.

(R-2) Si.

(R-3) Solo los que me gustan.

(R-4) No.

(R-5) No.

(R-6) Si.

(R-7) Si.

(R-8) Si.

(R-9) No.

(R-10) Si.

(R-11) Si.

(R-12) De vez en cuando.

(R-13) Si.

(R-14) No.

(R-15) No.

(R-16) Si.

(R-17) No.

(R-18) No.

(R-19) No.

(R-20) Si.

(R-21) No.

(R-22) No.

(R-23) No.

5.- ¿Conoces la diferencia entre motores gráficos?

(R-1) No.

(R-2) No.

(R-3) No.

(R-4) No.

(R-5) No.

(R-6) No.

(R-7) Muy poco.

(R-8) No.

(R-9) No.

(R-10) Si.

(R-11) Si y no.

(R-12) No.

(R-13) No, eso no.

(R-14) No.

(R-15) No.

(R-16) Principalmente los límites que permite a la hora de diseñar.

(R-17) No, solo uso el que se me hace más cómodo.

(R-18) Si.

(R-19) Si.

(R-20) No.

(R-21) No.

(R-22) No.

(R-23) No.

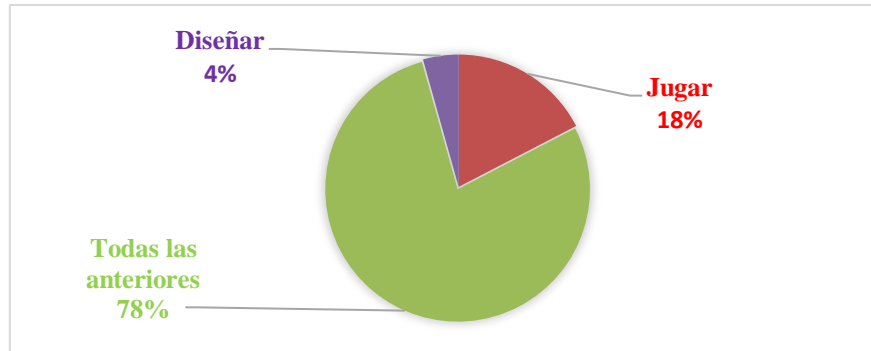
Escala Likert:

	Preguntas	Siempre	Casi Siempre	Algunas Veces	Rara Vez	Nunca
1	¿Cada cuánto juego videojuegos?	8	9	4	2	0
2	¿Qué tan seguido uso tu teléfono?	17	3	2	1	0
3	¿Al comprar un celular checo la gama?	7	12	2	2	0
4	¿Suelo analizar con qué motor gráfico este hecho mi videojuego favorito?	0	5	3	6	9
5	¿Suelo analizar las características de un videojuego?	2	6	8	6	1

5.2 Gráficas

Gráfica 1:

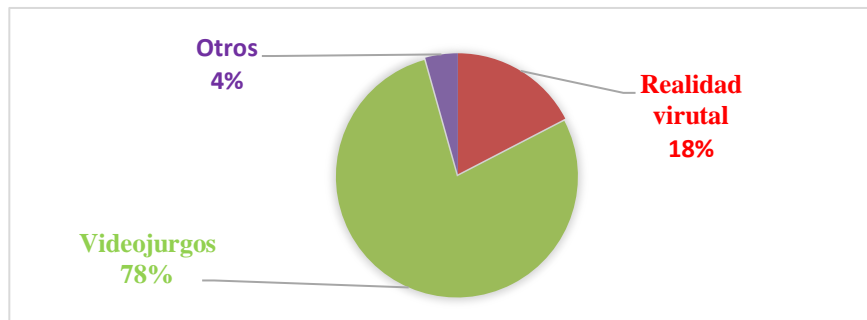
¿Para qué sirve un Motor Gráfico?



En la primera grafica se puede apreciar que el 78% de los encuestados tienen el conocimiento de que un motor gráfico sirve para, programar, jugar y diseñar, el 18% solo piensa que sirve para jugar y el último 4% dicen que solo es para diseñar.

Gráfica 2:

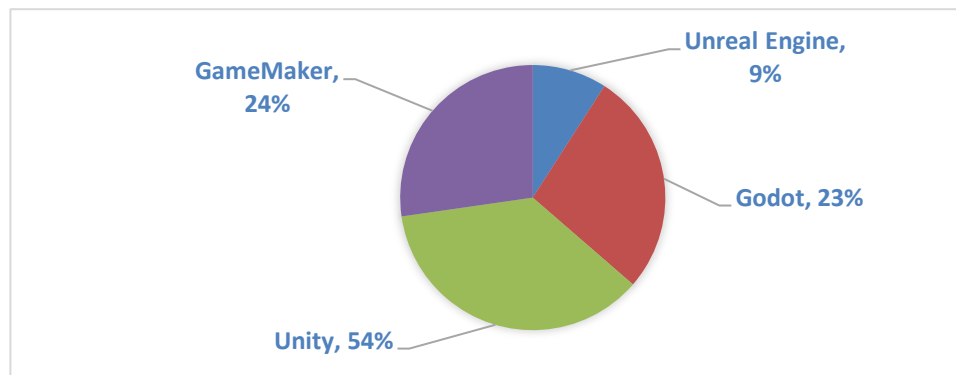
¿En qué contextos has oído hablar sobre Motores Gráficos?



La segunda grafica indica que el 78% ha oído hablar de los motores gráficos en videojuegos, mientras tanto el 18% recuerda escucharlo en el ámbito de realidad virtual y por ultimo solo el 4% lo ha escuchado hablar en otros temas.

Gráfica 3:

¿Si has oído hablar de ellos, cuáles conoces?

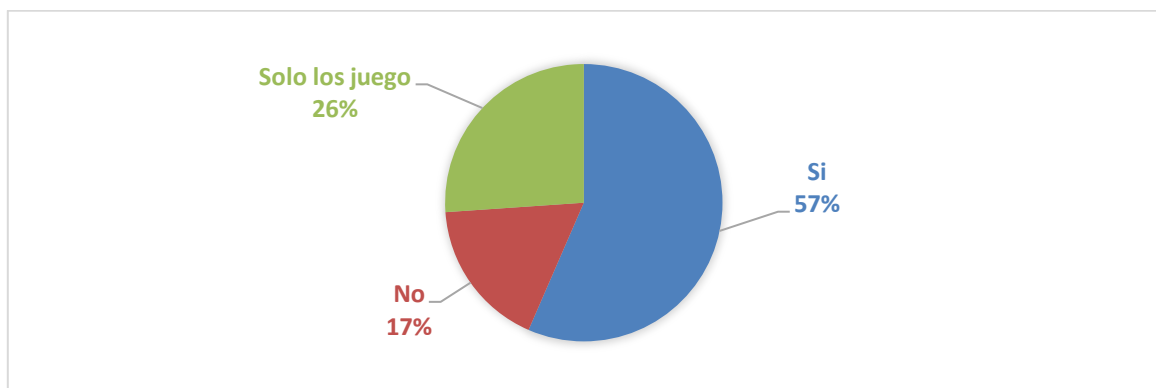


En la siguiente grafica rescataamos que el 54% conoce los motores gráficos por Unity, mientras tanto el 24% ha escuchado de GameMaker y los últimos 23% y no menos importantes conocen motores gráficos por Godot y Unreal Engine y con un último 9% para rara la vez que han oído hablar de ellos

0

Gráfica 4:

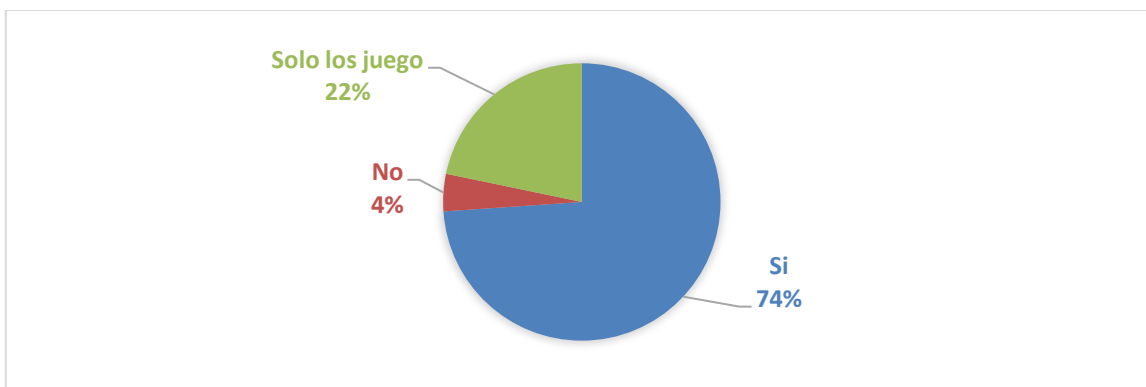
¿Sabes cómo se crean los videojuegos?



Mientras tanto, la siguiente grafica podemos ver que al 57% saben el cómo se hacen sus videojuegos favoritos, mientras el otro 26% no lo saben y el ultimo 17% solo los juega sin investigar más allá. Y un 9% rara la vez.

Gráfica 5:

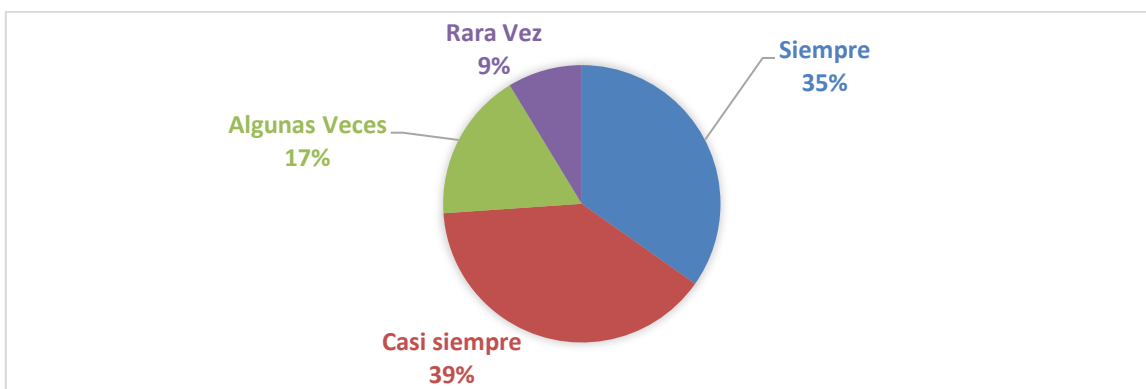
¿Te gustaría realizar un videojuego?



Después seguimos con los siguientes resultados en centrando que el 74% les gustaría realizar su videojuego, mientras que un escaso 4% no les gustaría y consecuencia de ello llegamos con un 22% que solo los juega.

Gráfica 6:

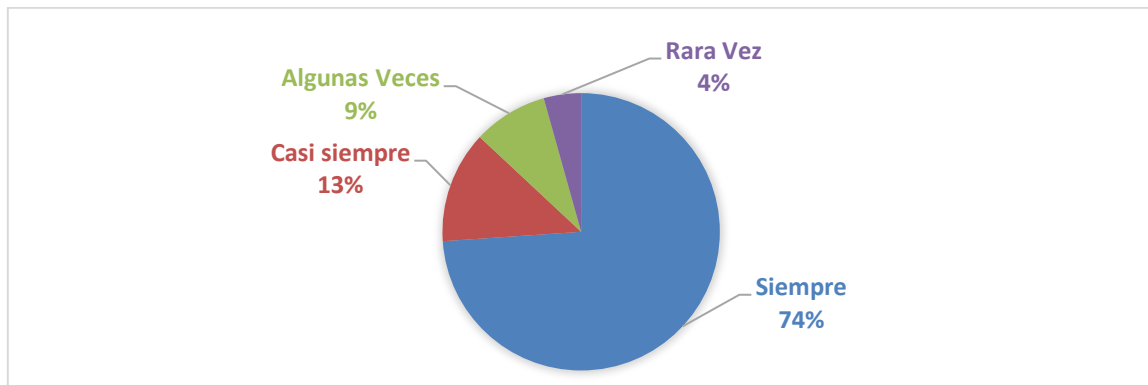
¿Cada cuánto juegas videojuegos?



En la siguiente grafica nos arroja que el 35% siempre juega videojuegos, el 39% casi siempre, con un 17% algunas veces, llegando a un 9% rara vez.

Gráfica 7:

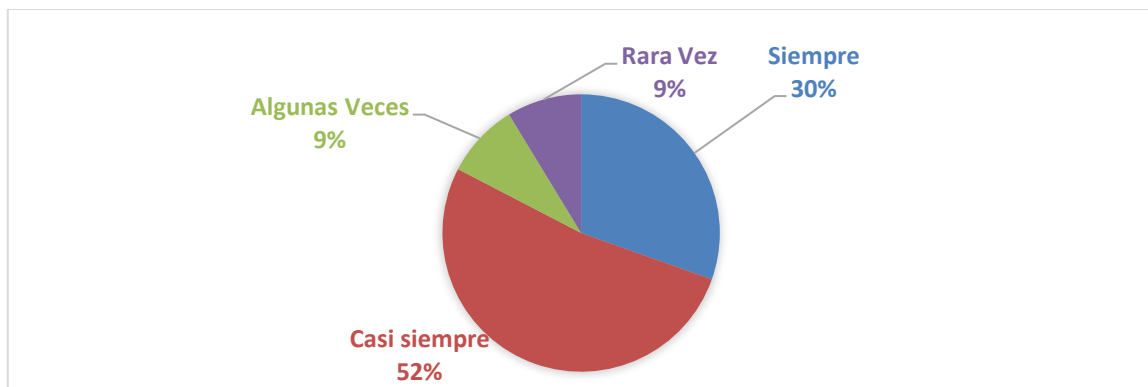
¿Qué tan seguido usas tu teléfono?



Luego observamos que el 74% siempre usa su teléfono, el 13% casi siempre y el más bajo con un 9% lo usan algunas veces, llegando a ser indispensable ya que nadie puede vivir sin su teléfono móvil y con un 4% en rara vez

Gráfica 8:

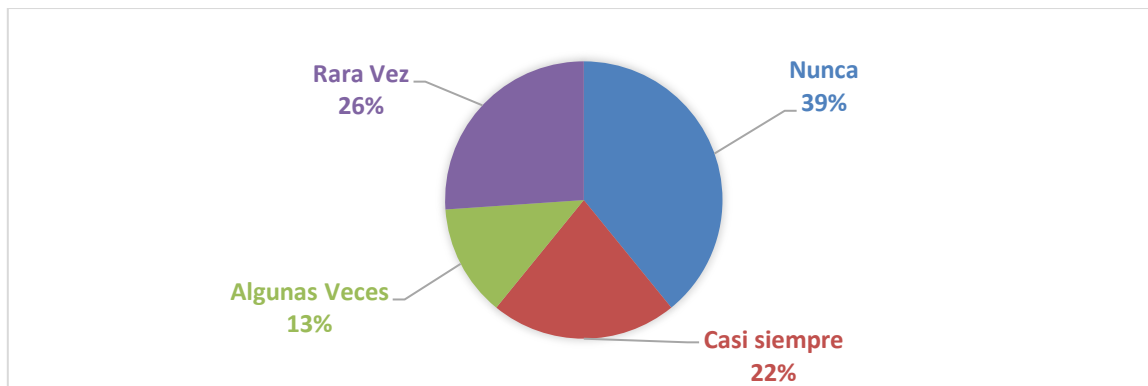
¿Al comprar un celular checas la gama?



Posteriormente los resultados a esta pregunta variaron llegando a que el 30% siempre chequeaba la gama del celular, mientras que el 52% casi siempre lo chequeaba, por último los resultados fueron que el 9% algunas veces lo cheque y de igual forma el 9% rara vez lo chequeara.

Gráfica 9:

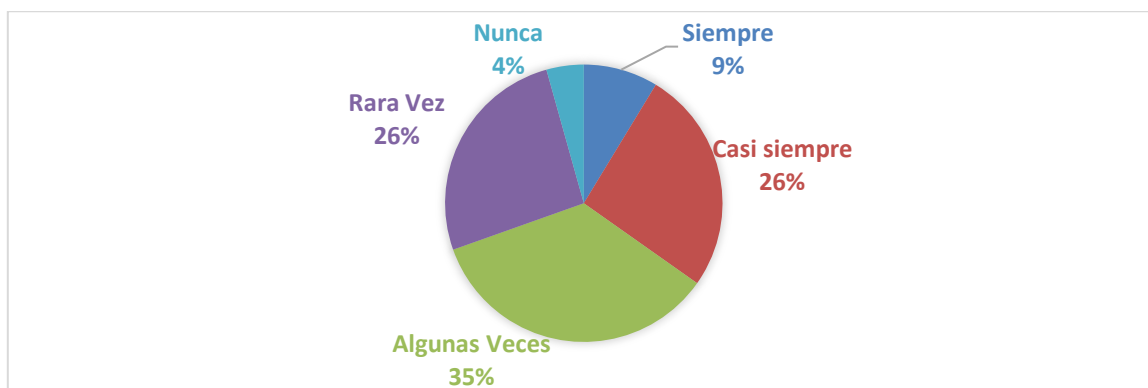
¿Suelen analizar con qué motor gráfico este hecho tu videojuego favorito?



En esta grafica apreciamos resultados muy distintos llegando el 22% casi siempre lo cheque, un 13% algunas veces lo checa, un 26% rara vez lo hace y el dato más sorprendente fue que el 39% nunca lo checa.

Gráfica 10:

¿Sueles analizar las características de un videojuego?



Por último, en esta grafica apreciamos un influyente 35% que algunas veces analiza las características de un videojuego, un 26% rara vez lo analiza, igualmente casi siempre cuenta con un 26%, lamentablemente con un 8% siempre lo checan y un 4% nunca lo analiza.

Informe Final

Conclusión y Recomendaciones

En este trabajo se observó las diferentes respuestas al uso de los motores gráficos al momento de trabajar con ellos, algunos de los usuarios lo usan como diseño para proyectos de videojuegos, ya sea 2D, 3D o realidad virtual. También pudimos notar que algunos usuarios lo ocupan como animador de escenarios y personajes, pero con todos los usos con cuerda en usarlo en los videojuegos.

Lo más importante de sus respuestas fue el saber que todas coincidían y llegaban al mismo punto que se quiere llegar, saben el funcionamiento y algunos de los encuestados usan esta herramienta en sus proyectos, aunque no tienen ña efectividad a falta de dinero o accesibilidad, por otra parte los usuarios de celulares Android supieron dar respuesta al porqué de su elección del celular que portan.

Durante todo esta investigación lo que más ayudo para poder solucionar, entender y saber a fondo el problema fue la aplicación de las encuetas realizadas al igual que la profunda investigación que se consiguió con éxito de estos tres motores gráficos ya que al hacer una análisis completo y una comparación muy meticulosa, se pudo sacar a la mejor conclusión y solución al problema planteado.

Lo que más se dificulto para que las personas pudieran tomaran en cuenta el problema va relacionada a la falta de conocimiento de la variedad de motores gráficos que hay en el mundo de la creación de videojuegos y que no estaban informaos de las ventajas y desventajas de estos a su vez los beneficios y ayuda que cada motor gráfico te aporta a tu crecimiento como desarrollador.

Recomendación.

Como recomendación general que se da para mejorar el problema es usar el motor gráfico que ofrece Unity sin descartar las otras dos opciones que es Engine y GameMaker ya que no hay mucha diferencia en sus motores gráficos cada uno de ellos tiene sus ventajas la elección de uno va depender de tus objetivos a realizar.

Haciendo mención a las recomendaciones anteriores, se sugiere elegir el motor gráfico de acuerdo a la accesibilidad que tengas para invertir en tu proyecto y la ayuda que crees necesitar en tu videojuego.

Otras recomendaciones generales que debes tomar en cuenta:

- Crear tu meta a lograr con tu videojuego a corto y a largo plazo
- Definir tu presupuesto que puedes invertir en el videojuego, esto tomando en cuenta tus metas anteriores
- Ser constante en las actualizaciones, esto para que puedan llegar ofertas de la compañía que quieran invertir en tu proyecto.

Referencias Bibliográficas

1. Acosta, F. A. S. (2023). Historia de los videojuegos: Tennis for Two, ¿el primer videojuego? Meristation. <https://as.com/meristation/noticias/historia-de-los-videojuegos-tennis-for-two-el-primer-videojuego-n/> (Acosta, 2023)
2. Cardan, I. (2023). Unity vs Unreal Engine 5. Instituto Cardan. <https://institutocardan.com/blog/diferencia-unity-vs-unreal-engine/> (Cardan, 2023)
3. Crear videojuegos con Unity nunca fue tan fácil - Tokio School. (s. f.). Tokio School. <https://www.tokioschool.com/noticias/crear-videojuegos-con-unity-nunca-fue-tan-facil/> (Crear Videojuegos Con Unity Nunca Fue Tan Fácil - Tokio School, s. f.)
4. De Joseph Hernández, V. T. L. E. (2020). ¿Qué es un motor gráfico? Sólo Es Ciencia. <https://soloesciencia.com/2020/06/24/que-es-un-motor-grafico/> (De Joseph Hernández, 2020)
5. Fernández, P. P. (2020). 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/> (Fernández, 2020b)
6. Garza, E. P. (2016). La historia de los juegos para móviles y su éxito. Más Allá de la Web 2 Cero. <https://3cero.com/historia-juegos-moviles/>
7. Gonzáles, C. (2015,). Análisis: motores gráficos y su aplicación en la industria. <https://www.aragon.es/documents/20127/674325/Estado%20del%20arte%20GameEngines%20y%20su%20impacto%20en%20la%20industria.pdf/db827568-09ef-e931-01e8-d293b9fca834>
8. Hervás, H. (2024). Así ha sido la evolución de los juegos móviles: todo comenzó en los años 60. AVPasión. <https://www.avpasion.com/evolucion-juegos-moviles-comenzo-anos-60/> (Hervás, 2024)
9. Los 7 mejores celulares gamer para juegos de alta gama 2024. (s. f.). doto.com.mx. <https://www.doto.com.mx/blog/post/mejores-celulares-gamer-para-juegos-de-alta-gama> (Los 7 Mejores Celulares Gamer Para Juegos de Alta Gama 2024, s. f.)

10. Portaltic, & Portaltic. (S. F.). La evolución de los videojuegos para el móvil: de Snake a Super Mario Run. europapress.es. <https://www.europapress.es/portaltic/videojuegos/noticia-evolucion-videojuegos-movil-snake-super-mario-run-20161004085944.html> (Portaltic & Portaltic, s. f.)
11. Rodríguez, M. (2018). La Primera Era del 3D: La generación que lo cambió todo. Meristation. https://as.com/meristation/2018/10/19/reportajes/1539948499_015444.html (Rodríguez, 2018)
12. Ros, I. (2018). Evolución gráfica de los juegos de lucha 2D a través de los años. MuyComputer. <https://www.muycomputer.com/2017/03/17/evolucion-grafica-juegos-lucha-2d/> (Ros, 2018)
13. Ros, I. (2018). Evolución gráfica de los juegos de lucha 2D a través de los años. MuyComputer. <https://www.muycomputer.com/2017/03/17/evolucion-grafica-juegos-lucha-2d/> (Chamorro, 2011)
14. Rpp, R. (2018). La historia de la animación en los videojuegos. RPP Noticias. <https://rpp.pe/tecnologia/videojuegos/la-historia-de-la-animacion-en-los-videojuegos-noticia-1165960?ref=rpp> (Rpp, 2018)
15. Tones, J. (2023). Unity, el motor gráfico de los pequeños estudios, comienza a cobrar por su uso. Los desarrolladores. https://www-xataka.com.cdn.ampproject.org/v/s/www.xataka.com/videojuegos/unity-motor-grafico-pequenos-estudios-comienza-a-cobrar-su-uso-desarrolladores-estan-que-trinan/amp?amp_gsa=1&js_v=a9&usqp=mq331AQIUAKwASCAAgM%3D#amp_tf=D%20%251%24s&aoh=17135486696728&referrer=https%3A%2F%2Fwww.google.com&share=https%3A%2F%2Fwww.xataka.com%2Fvideojuegos%2Funity-motor-grafico-pequenos-estudios-comienza-a-cobrar-su-uso-desarrolladores-estan-que-trinan (Tones, 2023b)
16. Valenzuela, C. G., & Valenzuela, C. G. (2023). Esta es la historia de los teléfonos móviles: desde su origen hasta la actualidad. Computer Hoy. <https://computerhoy.com/moviles/historia-telefonos-moviles-origen-actualidad-1181484> (Valenzuela & Valenzuela, 2023)
17. Venegas, P. (2024). Este fue el primer videojuego de la historia con gráficos 3D. Alfa Beta Juega. <https://www.mundodeportivo.com/alfabeta/juegos/este-fue-el-primer-videojuego-de-la-historia-con-graficos-3d> (Venegas, 2024)

Anexos