



UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO

CARRERA:

INGENIERÍA DE SISTEMAS

Trabajo de titulación previo a la obtención del título de:

Ingeniera de Sistemas

TEMA:

**IMPLEMENTACIÓN DE UN JUEGO EDUCATIVO EN JAVASCRIPT PARA UNA
PLATAFORMA MÓVIL.**

AUTORA:

JEZZREL NICOLE VACA CABRERA

TUTOR:

JULIO RICARDO PROAÑO ORELLANA

Quito, agosto del 2020

CESIÓN DE DERECHOS DE AUTOR

Yo, JEZZREL NICOLE VACA CABRERA, con documento de identificación N° 1722521158, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autora del trabajo de titulación intitulado: “IMPLEMENTACIÓN DE UN JUEGO EDUCATIVO EN JAVASCRIPT PARA UNA PLATAFORMA MÓVIL”, mismo que ha sido desarrollado para optar por el título de: INGENIERA DE SISTEMAS, en la Universidad Politécnica Salesiana, que dando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autora me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en digital a la Biblioteca de la Universidad Politécnica Salesiana.



JEZZREL NICOLE VACA CABRERA

CI:1722521158

Quito, agosto de 2020

DECLARATORIA DE COAUTORÍA DEL TUTOR

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: “IMPLEMENTACIÓN DE UN JUEGO EDUCATIVO EN JAVASCRIPT PARA UNA PLATAFORMA MÓVIL.”, realizado por Jezzrel Nicole Vaca Cabrera, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, agosto del 2020



Julio Ricardo Proaño Orellana

C.I: 0103909412

DEDICATORIA

A mi madre por su infinito amor que siempre me anima a hacer todo lo que me propongo, su confianza en que lo lleve a cabo con responsabilidad, y sus consejos inigualables que hacen que cada día sea mejor persona.

A mi hermano por ser mi inspiración, el apoyo y el cariño que me brinda en mi vida y en los momentos más importantes.

A mis abuelos por estar siempre para mí apoyándome en cada etapa de mi vida.

Jezzrel Nicole Vaca Cabrera

AGRADECIMIENTOS

A mis padres por apoyarme en todas las etapas de mi formación personal y profesional.

Agradezco a la Universidad Politécnica Salesiana por contribuir en mi formación académica, a mi tutor de proyecto de titulación Ingeniero Julio Proaño por motivarme y orientarme con su conocimiento y experiencia a culminar este trabajo satisfactoriamente.

Jezzrel Nicole Vaca Cabrera

ÍNDICE

INTRODUCCIÓN.....	1
Antecedentes	1
Estado del arte	2
Justificación	5
Objetivo General	6
Objetivos Específicos.....	6
1 CAPÍTULO 1: MARCO TEÓRICO.....	7
1.1 Sistema Operativo Móvil.....	7
1.2 Android.....	7
1.2.1 Historia.....	8
1.2.2 Características	9
1.2.3 Arquitectura	9
1.3 iOS.....	13
1.3.1 Historia.....	14
1.3.2 Arquitectura	14
1.4 Comparación de plataformas móviles	16
1.5 Aplicaciones Móviles	17
1.6 Tipos de aplicaciones móviles	18
1.6.1 Aplicaciones Nativas	19
1.6.1.1 Consideraciones.....	20
1.6.2 Aplicaciones Web	21

1.6.3	Aplicaciones Híbridas	22
1.6.3.1	Interfaz de programación de aplicaciones (API)	24
1.6.4	Comparación de tipos de aplicaciones móviles	25
1.7	Apache Cordova	25
1.7.1	Historia.....	27
1.7.2	Componentes.....	28
1.7.3	Arquitectura	29
1.8	Metodologías para el desarrollo de videojuegos	29
1.9	Cascada.....	30
1.9.1	Fases.....	31
1.9.1.1	Especificación del Juego	31
1.9.1.2	Biblia del Arte	31
1.9.1.3	Especificación Técnica	32
1.9.1.4	Construcción.....	32
1.9.1.5	Aseguramiento de Calidad.....	32
1.9.1.6	Golden Master	33
1.9.2	Ventajas y Desventajas	33
1.10	Scrum.....	34
1.10.1	Roles de Scrum.....	35
1.10.2	Ventajas y Desventajas.....	35
1.11	Modelo de Calidad de experiencia del usuario basado en jugabilidad.....	36
2	CAPÍTULO 2: DESARROLLO DEL VIDEOJUEGO	39

2.1	Especificación del juego	39
2.2	La biblia del arte	39
2.3	Especificaciones Técnicas	45
2.4	Construcción	53
2.5	Aseguramiento de calidad	54
2.6	Golden Master	60
3	CAPÍTULO 3: EVALUACIÓN Y ANÁLISIS DE RESULTADOS	61
3.1	Análisis de resultados	62
3.1.1	Jugabilidad	62
3.1.2	Conocimientos	64
3.2	Análisis de Software	65
	CONCLUSIONES	67
	RECOMENDACIONES	68
	LISTA DE REFERENCIAS	69
	ANEXOS	74

ÍNDICE DE TABLAS

Tabla 1: Estado del arte de juegos en puzzle	2
Tabla 2. Criterios de plataformas móviles	16
Tabla 3. Criterios de los tipos de aplicaciones	25
Tabla 4. Ventajas y desventajas de la metodología en cascada	33
Tabla 5. Ventajas y desventajas de la metodología scrum	36
Tabla 6. Ejemplo de métricas por el modelo PQM	37
Tabla 7. Concepto del juego.....	39
Tabla 8. Especificación del caso de uso	46
Tabla 9. Especificación del caso de uso	47
Tabla 10. Especificación del Caso de Uso	47
Tabla 11. Especificación del caso de uso	49
Tabla 12. Especificación del caso de uso	50
Tabla 13. Especificación del caso de uso	50
Tabla 14. Especificación del caso de uso	51
Tabla 15. Especificación del caso de uso	52
Tabla 16. Resultados de las métricas del modelo basado en la jugabilidad.....	63

ÍNDICE DE FIGURAS

Figura 1. Volumen de envío global de Smartphones.	8
Figura 2. Arquitectura del sistema operativo Android.....	10
Figura 3. Actual arquitectura del sistema Android	11
Figura 4. Arquitectura del sistema iOS	15
Figura 5. Capacidad multiplataforma.....	19
Figura 6. Esquema de aplicaciones nativas	20
Figura 7. Esquema de las aplicaciones web	22
Figura 8. Esquema de las aplicaciones híbridas	23
Figura 9. Proceso de creación de aplicaciones según plataforma	26
Figura 10. Arquitectura de Apache Cordova	29
Figura 11. Metodología en cascada en el desarrollo de videojuegos	31
Figura 12. Elementos de Scrum	34
Figura 13. Pantalla de inicio.....	41
Figura 14. Selección de un país.....	42
Figura 15. Interfaz rompecabezas	42
Figura 16. Interfaz menú del rompecabezas.....	43
Figura 17. Interfaz imagen original del rompecabezas	43
Figura 18. Interfaz información de la imagen del rompecabezas.....	44
Figura 19. Interfaz final al terminar un rompecabezas.....	44
Figura 20. Diagrama de flujo	45
Figura 21. Caso de uso pantalla principal	46
Figura 22. Caso de uso pantalla rompecabezas.....	49
Figura 23. Código Fuente.....	54

Figura 24. Captura de pantalla de la instalación y nombre de la aplicación.	55
Figura 25. Captura de pantalla de la pantalla de inicio	56
Figura 26. Captura de pantalla de la elección del país	56
Figura 27. Captura de pantalla del rompecabezas	57
Figura 28. Captura de pantalla del botón atrás	58
Figura 29. Captura de pantalla del botón menú	58
Figura 30. Capturas de pantalla del menú opción ver imagen	59
Figura 31. Capturas de pantalla del menú opción ver información	60
Figura 32. Resultados de la primera encuesta	64
Figura 33. Resultados de la segunda encuesta	65
Figura 34. Rendimiento del software	66

ÍNDICE DE ANEXOS

Anexo 1: Primera encuesta.....	74
Anexo 2: Segunda encuesta.....	74
Anexo 3: Datos recolectados de la encuesta de jugabilidad.....	74

RESUMEN

En este trabajo se diseña y desarrolla un videojuego educativo, didáctico que aumente en las personas la capacidad de observación, concentración y atención con el rompecabezas y el aprendizaje sobre los países con datos en forma de comentarios, para la creación del videojuego se tomó en cuenta el modelo en cascada modificado para los videojuegos, además se aplicó dos tipos de pruebas para determinar si el juego cumple con los objetivos propuestos, estas pruebas fueron realizadas en 17 voluntarios de diferentes edades. Los resultados obtenidos evidencian una mejoría en los conocimientos de los distintos países luego de jugar el videojuego, demostrando así que este tipo de videojuegos pueden ser una opción factible para obtener conocimientos generales de los países así como también una herramienta pedagógica para la enseñanza de la Geografía.

Palabras clave: Videojuegos, Jugabilidad, Geomedia, Modelo Cascada

ABSTRACT

This thesis aims to design and development of an educational and didactic video game that increases concentration and attention abilities in people with a puzzle and learning ability about countries with some facts. To create this video game, it was considered a modified waterfall model, also two types of tests were carried out to determine if the video game gets the proposed aims. Assessments were carried out into 17 volunteers with different age ranges.

The result of this thesis illustrates the augment a player's knowledge of countries and demonstrates that this type of video games can be a feasible option to improve general knowledge as well as pedagogical tool for Geography education.

Keywords: Video games, Playability, Geomedia, Waterfall Process

INTRODUCCIÓN

Antecedentes

Actualmente el aprendizaje memorizado está quedando obsoleto, se ha usado sobre todo en la asignatura de Ciencias Sociales, es decir, Historia y Geografía, estudios realizados por pedagogos concluyen que este modelo de enseñanza-aprendizaje puede causar problemas en los alumnos o simplemente olvido luego de cierto tiempo por falta de razonamiento. Debido a la falta de recursos didácticos e interactivos que mantengan la atención de los alumnos, optan por el modelo de aprendizaje memorizado, repitiendo y reteniendo información por un corto periodo de tiempo y sin lugar a ningún análisis (Evaristo Chiyong et al., 2016).

Los videojuegos educativos se concibieron con la premisa de ser educativos y entretenidos, dirigidos para niños de edades tempranas y para ser jugados en el hogar, con el tiempo estos fueron evolucionando para servir como una herramienta de apoyo al docente (Evaristo Chiyong et al., 2016).

Estado del arte

Se recopilan el resumen con los resultados de algunos estudios realizados sobre juegos de puzzle y la importancia con la enseñanza-aprendizaje que se muestran a continuación en la

Tabla 1: Estado del arte de juegos en puzzle.

Tabla 1: Estado del arte de juegos en puzzle

Autor	Título	Año publicación	Resumen
Montoya, Carlos Flores, Pablo	Los puzzles en alambre como recurso didáctico para la enseñanza de las matemáticas	2003	En este artículo los autores presentan los beneficios de los puzzles de alambres y realizan un taller en una escuela, una vez realizado el estudio concluyen: “Estos puzzles plantean verdaderos retos, a través de actividades lúdicas que despiertan la curiosidad y el interés de los alumnos, haciéndolos propensos a explorar algunas cualidades del espacio tridimensional.”
FERNÁNDEZ PÉREZ, BLANCA	PUZZLES Y JUEGOS TOPOLÓGICOS	2009	El artículo tiene como objetivo analizar las utilidades del puzzle y los juegos topológicos como herramientas didácticas y concluye con los beneficios de estos juegos como el dominio de los cuerpos, además afirma que, aunque los juegos ayuden al aprendizaje se debería complementar con lecturas para mejorar el uso de estos juegos en la vida cotidiana

Elaborado por: La autora

Fuente: Adaptada de (FERNÁNDEZ PÉREZ, 2009; Montoya & Flores, 2003)

Descripción del problema

En la actualidad la tecnología ha tomado mucha importancia en todos los aspectos laborales, sociales y educativos, las empresas cambian su estructura, sus estrategias de mercadeo, es mucho más fácil la comunicación con personas situadas al otro lado del mundo, las escuelas, colegios, unidades educativas en general incluyen cada vez más la tecnología en los métodos de enseñanza.

Se ha demostrado que los videojuegos tienen buenos resultados al momento de enseñar, si se logra complementar el videojuego con alguna clase docente los resultados son aún mejores. La geografía es una disciplina que se encarga de describir gráficamente al planeta Tierra, de esta forma la geografía se va adaptando con el paso del tiempo a las nuevas tecnologías, el uso de cualquier elemento multimedia en la geografía, posicionamientos y georeferenciación con fines educativos se lo conoce como geomedia (de Miguel González & y Torres, 2012).

Con la evolución de la tecnología las nuevas generaciones de personas se sienten cada vez más atraídas a estas herramientas, sobre todo a los dispositivos inteligentes que han ido desplazando a los libros, debido a esto el problema de atención y concentración en clases es cada vez más común, pero gracias a los estudios realizados sobre los videojuegos como método de enseñanza, se ha comprobado los beneficios cognitivos que estos traen a los estudiantes (de Miguel González & y Torres, 2012).

Aunque los videojuegos educativos en el campo de la Geografía ya son una realidad, estos tienden a definirse a un solo territorio o pretenden abarcar la historia de dicho territorio. Por esto algunos de estos juegos no despiertan el interés suficiente por la Geografía, esta ha quedado solo a la necesidad, para viajar sea por negocios o vacaciones se consulta en el mapa (de un dispositivo móvil) cuál es el destino del viaje y al llegar al país de destino se realiza diferentes transacciones económicas, para esto debe realizar una nueva búsqueda de la moneda de dicho

país, lo mismo sucede con la fauna, animales que si no se visita una reserva o un zoológico no se sabe cuál es la importancia de su existencia en el ambiente en el que se desarrolla, lo simbólicas o endémicas que son.

Justificación

Muchos docentes de unidades educativas ya sean escolares o universitarios tienen una ideología y/o método de enseñanza muy definido por el cual no están en constante actualización con la tecnología y el alumnado, esto también varía de acuerdo con el pensum oficial que muestra la institución educativa o se ven incapacitados de hacer uso de nuevas tecnologías por la falta de infraestructuras o los equipos informáticos necesarios que desencadena el impedimento en el desarrollo digital de la educación como las geomedias (I. B. Sánchez, 2011).

Actualmente la educación 2.0 busca abrir oportunidades a los niños y jóvenes en el campo de la tecnología, desarrollando sus destrezas cognitivas, sociales y motoras con los videojuegos educativos reconocidos como un recurso didáctico para el aprendizaje que además resulta ser un medio llamativo para los alumnos (I. B. Sánchez, 2011).

Debido a los aspectos negativos a los que se les asocia a los videojuegos con los niños y jóvenes se tiene un temor de generar agresividad, falta de atención, etc. Sin embargo, los diseñadores de videojuegos educativos deben considerar este tipo de aspectos y de esta forma crear videojuegos con el enfoque de enseñanza que facilite que los alumnos obtengan habilidades cognitivas, que sirvan a su desarrollo personal y sean aplicables en cada área de su vida.

Objetivo General

Diseñar e implementar un juego educativo en JavaScript para una plataforma móvil utilizando el paradigma de aplicación híbrida.

Objetivos Específicos

Diseñar una aplicación híbrida utilizando tecnologías web como HTML, CSS y JavaScript.

Desarrollar un juego para la compilación en Android y iOS utilizando el entorno de desarrollo Cordova.

Diseñar un juego didáctico que aumente en las personas la capacidad de observación, concentración y atención con el rompecabezas y el aprendizaje sobre los países con datos en forma de comentarios.

Evaluar la calidad del juego siguiendo un modelo de calidad basado en métricas de jugabilidad

Evaluar la calidad del juego en el aspecto educativo usando métricas pertinentes.

1 CAPÍTULO 1: MARCO TEÓRICO

1.1 Sistema Operativo Móvil

Un sistema operativo móvil es un conjunto de programas (Software) que tienen un control directo con los componentes físicos, sirve como capa intermedia entre el conjunto de elementos y circuitos físicos (hardware) y los programas que interactúan con el usuario al igual que con las de escritorio, pero estos sistemas para móviles son más simples y se orientan a la conectividad inalámbrica. Existe una gran variedad de sistemas operativos móviles como Android, iOS, BlackBerry OS, Windows Phone, entre otros (Aponte Gomez & Davila Ramirez, 2011). Esta sección se enfoca en los sistemas operativos iOS y Android.

1.2 Android

El código de Android es abierto y está basado en el Kernel de Linux este fue liberado por Google bajo la licencia Apache, GPL v2. Cuenta con una tienda oficial virtual donde se recopilan aplicaciones para su descarga con el nombre de Play Store antes conocida como Android Market (Gironés, 2011).

Para tener una buena calidad de audio y gráficos, Android usaba códecs además para garantizar la calidad de las gráficas mostraba imágenes vectoriales suavizados y basados en una especificación estándar, OpenGL (Gironés, 2011).

Según la IDC (Corporación Internacional de Datos) una compañía proveedora de mundial de inteligencia de mercado subsidiaria de IDG, muestra en la Figura 1. Volumen de envío global de Smartphones. el volumen de envío global de los principales fabricantes de smartphones del primer trimestre del año 2019 y 2020 (IDC Quarterly Mobile Phone Tracker, 2020). Ya que los

fabricantes Samsung y Huawei lideran la lista se puede concluir que Android es el sistema con mayor uso seguido de iOS.

Estadísticas de envío, las cifras dadas son millones de unidades

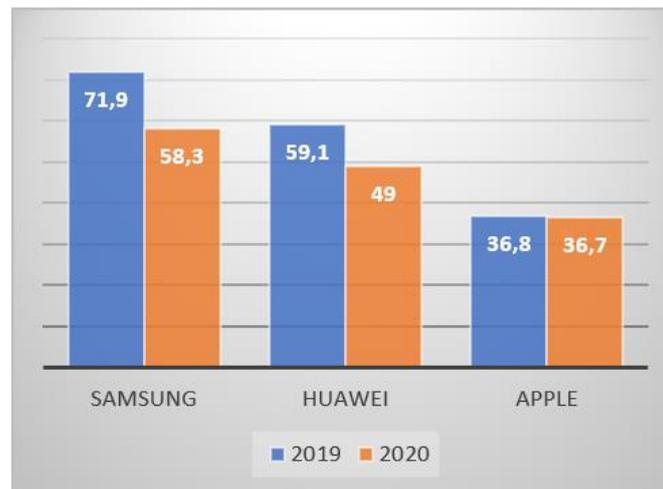


Figura 1. Volumen de envío global de Smartphones.
Fuente: Adaptada de (IDC Quarterly Mobile Phone Tracker, 2020)

1.2.1 Historia

Este sistema operativo fue desarrollado por Android Inc. una empresa fundada en el 2005 con el fin de realizar un sistema operativo para los dispositivos móviles, dos años después Google compra la empresa Android Inc. Al año siguiente Google publica una versión del SDK de Android que cambia por completo la forma en la cual estaban constituidos los dispositivos móviles, en su versión 1.5 se incorpora el teclado en pantalla (Báez et al., 2012).

Google une el proyecto de Android a la Open Handset Alliance que es una alianza de varias compañías que se encargan de desarrollar estándares abiertos para dispositivos móviles entre las compañías están Motorola, Samsung, Nvidia, LG, Dell, Intel, Google, entre otros. (Aponte Gomez & Davila Ramirez, 2011).

El fabricante T-Mobile lanza su primer smartphone con sistema operativo Android en 2008, meses después Google lanza también su primer smartphone con Android sin estar sujeto a

ningún operador de telefonía móvil, este teléfono fue lanzado para que los desarrolladores programen sus aplicaciones y realicen las pruebas en un dispositivo físico y no simulado (Aponte Gomez & Davila Ramirez, 2011).

1.2.2 Características

Android es un entorno de software integrado en lugar de una plataforma de hardware, ya que los componentes del sistema operativo están escritos en C, las aplicaciones están escritas en Java y las aplicaciones integradas también están escritas en Java, la mayoría de estas aplicaciones son desarrolladas en Java usando el SDK; Software Development Kit o herramientas de desarrollo de Software de Android (Aponte Gomez & Davila Ramirez, 2011).

Las aplicaciones incorporadas en el sistema operativo y las aplicaciones desarrolladas por el SDK no tienen diferencias por este motivo el desarrollador puede crear aplicaciones en las cuales se puede usar los recursos del dispositivo sin embargo existen excepciones como las aplicaciones exploratorias de Linux que se desarrollan con el NDK; Native Development Kit o herramientas de desarrollo nativas de Android (Ableson et al., 2011).

Otra característica de Android y una de las más importantes es el código abierto, gracias a la licencia Apache existe una comunidad global de desarrolladores de Android, por ejemplo, nuevos códecs multimedia pueden ser desarrollados por esta comunidad en lugar de esperar a una nueva actualización de Android con la nueva funcionalidad (Ableson et al., 2011).

1.2.3 Arquitectura

Como se puede observar en la Figura 2. Arquitectura del sistema operativo Android, la arquitectura antigua del sistema operativo Android se compone de cuatro capas que son: Kernel, Librerías, Armazón de Aplicaciones y Aplicaciones. Esta arquitectura estuvo vigente desde el inicio del

proyecto hasta el Android versión 8.0 llamada Oreo, en las versiones posteriores a este se aumentó una capa a la arquitectura.

Arquitectura del sistema Android



Figura 2. Arquitectura del sistema operativo Android

Fuente: Tomada de Arquitectura del sistema Android, por Madrid, n.d, Software de comunicaciones.

Como se puede observar en la Figura 3. Actual arquitectura del sistema Android, la arquitectura actual del sistema operativo Android se compone de cinco capas que son: Kernel, HAL (Hardware Abstraction Layer), Librerías y Tiempo de ejecución, Marco de API de Java y Aplicaciones.

Actual arquitectura del sistema Android

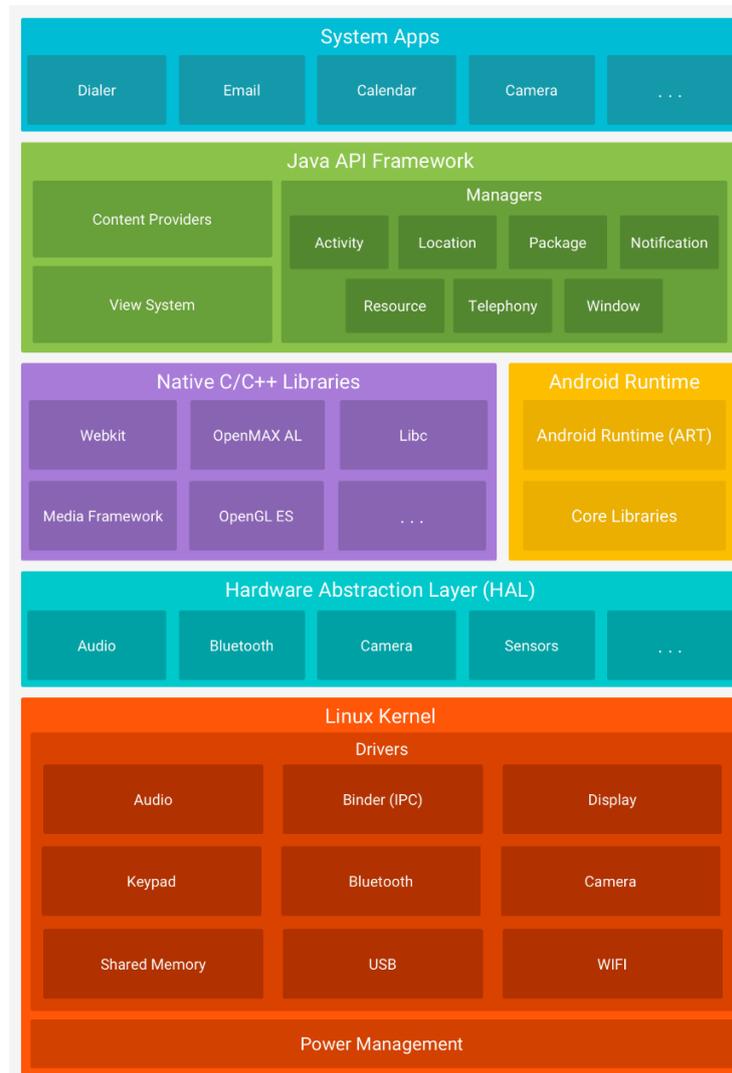


Figura 3. Actual arquitectura del sistema Android

Fuente: Tomada de Arquitectura del sistema Android, por Developers, 2020, Android.

Núcleo o Kernel de Linux: Esta capa de bajo nivel interactúa directamente con el hardware, esto permite que Android use las funciones de seguridad, además permite el desarrollo de controladores por los fabricantes de dispositivos. Esta capa es la encargada de los controladores de los componentes electrónicos del hardware como el manejo de la memoria, la administración del multiproceso, el audio, la pantalla, el Wi-Fi y demás componentes electrónicos (Ableson et al., 2011).

Capa de Abstracción de Hardware (HAL): Hardware Abstraction Layer o capa de abstracción de hardware, dedicada para los proveedores y fabricantes de hardware con el principal objetivo de ser más rápido y menos costosa ya que provee de interfaces estándares que muestra la capacidad del dispositivo a la capa de nivel más alto (marco de API de Java). Esta capa contiene módulos de biblioteca y cada uno de estos tiene un interfaz para cada componente de hardware específico como por ejemplo cámara, audio, bluetooth. Cuando la capa del marco de trabajo de API hace una petición para obtener acceso a un componente en específico del hardware del dispositivo, Android carga el módulo de biblioteca del componente solicitado (G. Developers, 2020).

Tiempo de ejecución: Como se observa en la Figura 2. Arquitectura del sistema operativo Android en el tiempo de ejecución de Android se muestra máquina virtual Dalvik, este entorno aloja las aplicaciones de Android y estuvo disponible en todas las versiones anteriores a Android versión 5.0 llamada Lollipop con nivel API 21. A partir de Android 5.0 en adelante el tiempo de ejecución de Android cambia a ART como se puede observar en la Figura 3. Actual arquitectura del sistema Android, este entorno consiste en la ejecución de los procesos propios de cada aplicación en sus propias instancias (Ableson et al., 2011; G. Developers, 2020).

Los archivos DEX son un formato de código de bytes para Android y diseñado para ocupar el menor espacio de memoria posible, el ART crea cadenas de herramientas y compila fuentes Java en archivos DEX de esta manera puede ejecutar varias máquinas virtuales aun cuando el dispositivo sea de memoria baja (G. Developers, 2020).

Librerías: Estas librerías y componentes como ART y HAL son nativas es decir escritas en C y C++, las aplicaciones que son desarrolladas en este lenguaje pueden usar directamente el

NDK es decir Native Development Kit o herramientas de desarrollo nativas de Android, de esta forma se tiene acceso a la biblioteca de plataforma nativa (G. Developers, 2020).

Marco de trabajo de API de Java: Para los desarrolladores de aplicaciones todas las funciones del sistema operativo se encuentran disponibles en esta capa ya que se acceden mediante API's algunos de estos servicios son:

- Administrador de recursos: acceso a los recursos
- Administrador de actividad: ciclo de vida de las aplicaciones
- Administrador de notificaciones: las aplicaciones muestran alertas en la barra de estado
- Proveedores de contenido: Aplicaciones que tienen acceso a otras aplicaciones (G. Developers, 2020).

Aplicaciones: No hay diferencia entre las aplicaciones de sistema y las del usuario final, el desarrollador puede usar las API que usan las aplicaciones de sistema, es decir si la aplicación desarrollada usa una funcionalidad como la de enviar o recibir mensajes SMS se puede llamar directamente a la aplicación de mensajería (G. Developers, 2020).

1.3 iOS

iOS conocido como iPhone OS fue lanzado en el 2007, tiene compatibilidad con el iPod Touch, un reproductor de audio y video además de ser una agenda digital y el iPad una Tablet diseñada por Apple (Aponte Gomez & Davila Ramirez, 2011).

El kit de desarrollo de software (SDK) liberado en el 2008 permite el desarrollo de aplicaciones para iPhone, utiliza el programa Xcode para desarrollar las mismas, era necesario

pagar por una cuenta de desarrollador para realizar pruebas en el dispositivo, pero desde el Xcode7 es posible realizar pruebas en el dispositivo sin necesidad de tener la cuenta desarrollador (Aponte Gomez & Davila Ramirez, 2011).

El costo de distribución de las aplicaciones consta de dos partes, la membresía y la comisión. La membresía para publicar en la tienda oficial “App Store” es de \$99 dólares americanos por año y la comisión consiste en que la empresa recibe un porcentaje por la aplicación de pago que se publica, aunque si la aplicación es gratuita la empresa no recibe ningún porcentaje por la aplicación (A. Developers, n.d.).

1.3.1 Historia

Este sistema operativo fue desarrollado por Apple Inc. La misma que cuando se fundó en 1976 llevaba el nombre de Apple Computer en esta comenzaron a construir computadoras a mano su primera computadora fue Apple I, esta fue presentada en un club y a los participantes les llamó la atención así fue como la empresa comenzó a crecer tanto en la década de los 80's y 90's, aunque estuvo a punto de quebrar en 1996 (Yoffie & Rossano, 2012).

Luego de esto Apple Computer se convierte en Apple Inc. En este resurge lanzando gran cantidad de dispositivos innovadores, para el año 2012 las acciones de Apple crecen y se convierte en una de las compañías con más valor a nivel mundial (Yoffie & Rossano, 2012).

1.3.2 Arquitectura

Como se puede observar en la Figura 4. Arquitectura del sistema iOS, la arquitectura del sistema operativo iOS se compone de cuatro capas de abstracción que son: Núcleo del sistema, Núcleo de servicios, Capa de medios y Toque de cacao.

Arquitectura del sistema iOS

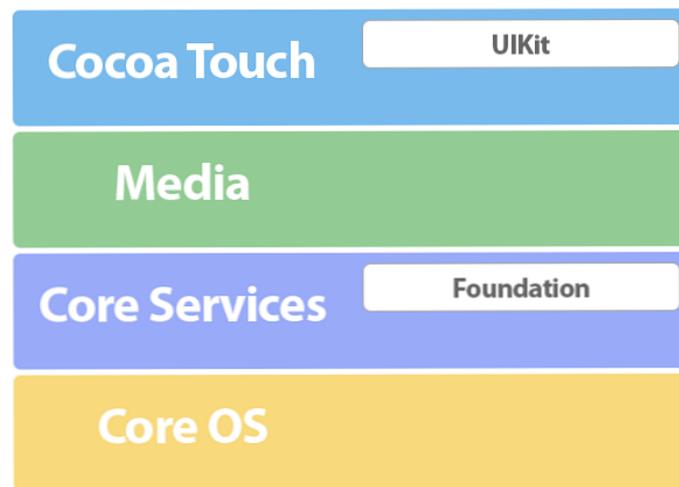


Figura 4. Arquitectura del sistema iOS

Fuente: Tomada de “Exploring the iOS SDK”, por Jacobs, 2012, iOS

Núcleo o Core del sistema: Es el núcleo del sistema operativo es decir la capa de bajo nivel que se comunica con los componentes físicos del hardware, también encapsula el Kernel y las interfaces, a esta capa los desarrolladores no tienen acceso para mantener la seguridad del dispositivo sin embargo existe una librería a la cual tienen acceso “libSystem” esta librería proporciona funciones a servicios DNS, Proxis, etc. (A. Developers, n.d.).

Núcleo de Servicios: Este núcleo administra los servicios que los desarrolladores usan para las aplicaciones nativas como por ejemplo el almacenamiento de iCloud, in-app purchase que son las compras integradas en la aplicación, ARC contador automático de referencias esta funcionalidad fue agregada desde el lanzamiento de iOS 5 y administra la memoria en Objective-C (A. Developers, n.d.).

Capa de Medios: Esta capa como su nombre lo indica son de medios multimedia es decir esta capa administra el audio, video y gráficos que usan estándares como OpenGL ES, OpenAL

y CoreGraphics. Además, administra las librerías con las cuales las aplicaciones tienen acceso a las fotos o videos del dispositivo (A. Developers, n.d.).

Toque de Cacao: Esta capa interactúa directamente con el desarrollador, proporciona tecnologías y estructura básica para las aplicaciones además contiene frameworks o marcos que usan las aplicaciones, el marco más usado es UIKit que es un marco para el desarrollo de interfaces de esta forma las aplicaciones nativas funcionan solo si se vincula con los frameworks UIKit y Foundation. UIKit es un marco para el sistema operativo iOS en su lugar el marco de desarrollo equivalente a UIKit es AppKit para el desarrollo para OS X (A. Developers, n.d.).

1.4 Comparación de plataformas móviles

Existen criterios y características comparables con las distintas plataformas móviles como Apple con iOS versión 13, Android versión 10.0, Windows Phone versión 10, BlackBerry versión 10, se muestra información resumida sobre las plataformas en la Tabla 2. Criterios de plataformas móviles.

Tabla 2. Criterios de plataformas móviles

Característica	iOS 13	Android 10	Windows Phone 10	BlackBerry 10
Compañía	Apple	Open Handset Alliance	Microsoft	BlackBerry
Núcleo del Sistema	Mac OS X	Linux	Windows NT	QNX
Licencia	Propietaria	Libre, Abierto	Propietaria	Propietaria
Año de Lanzamiento	2007	2008	2010	1999
Fabricante único	Si	No	No	Si

Soporte Memoria Externa	No	Si	Si	Si
Motor del navegador	WebKit	WebKit/ Chromium	Trident	WebKit
Tienda de aplicaciones	App Store	Google Play	Windows Marketplace	BlackBerry World
Coste publicación	\$99 / año	\$25 una vez	\$99 / año	Sin coste
Tiendas sin supervisión	No	Si	No	Si
Familia CPU soportada	ARM	ARM, MIPS, x86	ARM	ARM
Máquina virtual	No	Dalvik / ART	.net	No
Lenguaje de Programación	Objective-C, Swift	Java, C++, Kotlin	C#, Visual Basic, C++	C, C++, Java
Plataforma de desarrollo	Mac	Windows, Mac, Linux	Windows	Windows, Mac
Modo invitado	Si	Si	No	No

Fuente: Tomada de (Statista, n.d.)

1.5 Aplicaciones Móviles

Las aplicaciones móviles diseñadas para teléfonos y tabletas no siempre fueron diseñadas para teléfonos inteligentes las aplicaciones móviles como tal existían desde que los teléfonos existen, como la típica aplicación calculadora, una aplicación es un software que interactúa con el sistema operativo y el usuario. Los primeros teléfonos tenían un objetivo mejorar la vida personal por ejemplo el calendario digital en el cual se agendaban las citas y el usuario podía colocar recordatorios, la calculadora esencial para momentos de cálculos, alarmas, etc. (Cuello & Vittone, 2013).

Las aplicaciones móviles en realidad no son más que cualquier software desarrollado para los dispositivos móviles, como su nombre lo indica móviles significa acceso en cualquier momento o lugar. Las aplicaciones se desarrollan teniendo en cuenta ciertos aspectos de los

dispositivos como el sistema operativo o versión del mismo, procesador, memoria RAM, consumo de datos, etc. (Enriquez & Casas, 2013).

Debido a estas consideraciones antes de desarrollar una aplicación, las aplicaciones se pueden clasificar en categorías como aplicaciones nativas, aplicaciones web y aplicaciones híbridas.

1.6 Tipos de aplicaciones móviles

Debido al incremento de las ventas de los dispositivos móviles inteligentes se tiene la necesidad de crear aplicaciones para instituciones académicas, gobierno, empresas, etc. Lo que conlleva la existencia de los diferentes tipos de aplicaciones móviles o tipos de estrategias utilizadas en el desarrollo de estas que puedan ser usadas en las distintas plataformas o sistemas operativos móviles como Android, iOS, Windows Phone, BlackBerry, entre otros (M de Andrade et al., 2015).

Con la constante evolución de la tecnología móvil existen varios fabricantes de dispositivos con diferentes plataformas, nuevas versiones de las plataformas, mayor capacidad en los componentes del hardware como en la memoria RAM o procesador, nuevos sensores como giroscopios o acelerómetros; las empresas deben tomar en consideración todos estos aspectos y más criterios en el desarrollo de una aplicación móvil (Nahuel Delía, 2017).

Como se puede observar en la Figura 5 los tres tipos de aplicaciones móviles con su respectiva capacidad multiplataforma es decir las aplicaciones nativas no tienen capacidad multiplataforma ya que se desarrolla para cada plataforma, las aplicaciones híbridas tienen mayor capacidad multiplataforma y las aplicaciones web tienen una alta capacidad multiplataforma.

Capacidad multiplataforma



Figura 5. Capacidad multiplataforma

Fuente: Adaptado de "cross platform app a comparative study" (p. 33), por M de Andrade et al., 2015, International Journal of Computer Science & Information Technology.

1.6.1 Aplicaciones Nativas

Este tipo de aplicaciones nativas son aplicaciones que se desarrollan únicamente para cada sistema operativo por lo que necesitan un alto nivel de conocimiento en el lenguaje de programación que admite el sistema operativo en el cual va a ser instalado como por ejemplo para el sistema operativo Android el lenguaje de programación es Java, para Windows Phone el lenguaje es C#, para iOS el lenguaje es Objective-C, para BlackBerry el lenguaje es C++, entre otros (Jobe, 2013).

Una característica de las aplicaciones nativas es que son más eficientes ya que son desarrolladas para una plataforma en específico de esta forma las aplicaciones tienen acceso directo a los sensores de los dispositivos y se controla el comportamiento de estos como cámara, acelerómetro. Como se puede observar en la Figura 6 el esquema de las aplicaciones nativas, el código fuente se encapsula en un ejecutable este se compila en el sistema operativo haciendo llamadas a las API's que tienen el acceso directo a todos los componentes del hardware (Glera Aransay, 2013).

Esquema de aplicaciones nativas

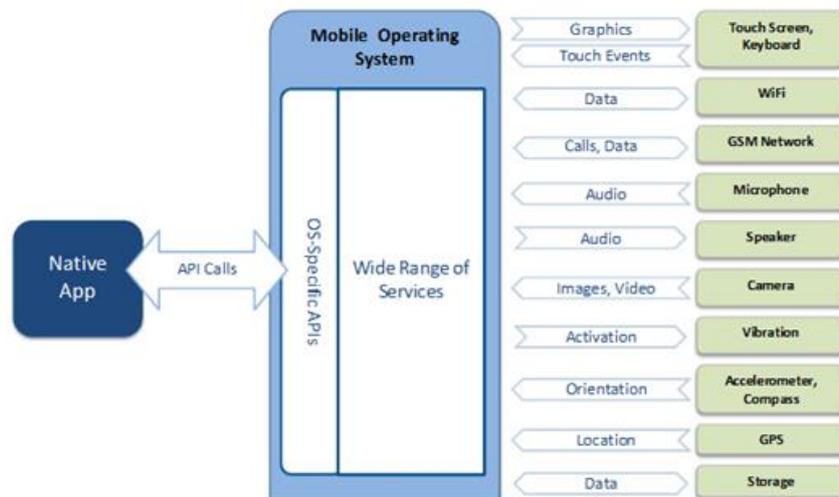


Figura 6. Esquema de aplicaciones nativas
Fuente: Tomada de (D. IBM, n.d.)

1.6.1.1 Consideraciones

Antes de desarrollar una aplicación nativa se debe tener en cuenta dos consideraciones como la población objetivo y los requisitos técnicos

Población objetivo: Antes de desarrollar una aplicación la empresa o los desarrolladores deben tomar en cuenta el público o grupo de personas hacia la cual va dirigida la aplicación y los dispositivos que utilizan, un lenguaje conocido que soporta algunos sistemas operativos como Android o BlackBerry es Java que se optimiza para ejecutarse en dispositivos con recursos limitados (Glera Aransay, 2013).

Requisitos Técnicos: Otro criterio que se debe tomar en consideración a la hora de decidir sobre una aplicación nativa son los requisitos técnicos de cada dispositivo por ejemplo ciertos sistemas operativos requieren un alto nivel de conocimiento en el lenguaje de programación o los componentes electrónicos del dispositivo como el procesador de un smartphone de gama alta, media o baja (Glera Aransay, 2013).

Pese a la gran desventaja de las aplicaciones nativas que es el costo de desarrollo, en la Figura 6. Esquema de aplicaciones nativas se muestra las llamadas a las API's o interfaz de programación de aplicaciones desde las aplicaciones nativas, estas API's son únicas en cada plataforma de modo que desarrollar una aplicación nativa garantiza el correcto uso de todas las funciones del dispositivo haciendo de esta una aplicación completa (W. IBM, 2012).

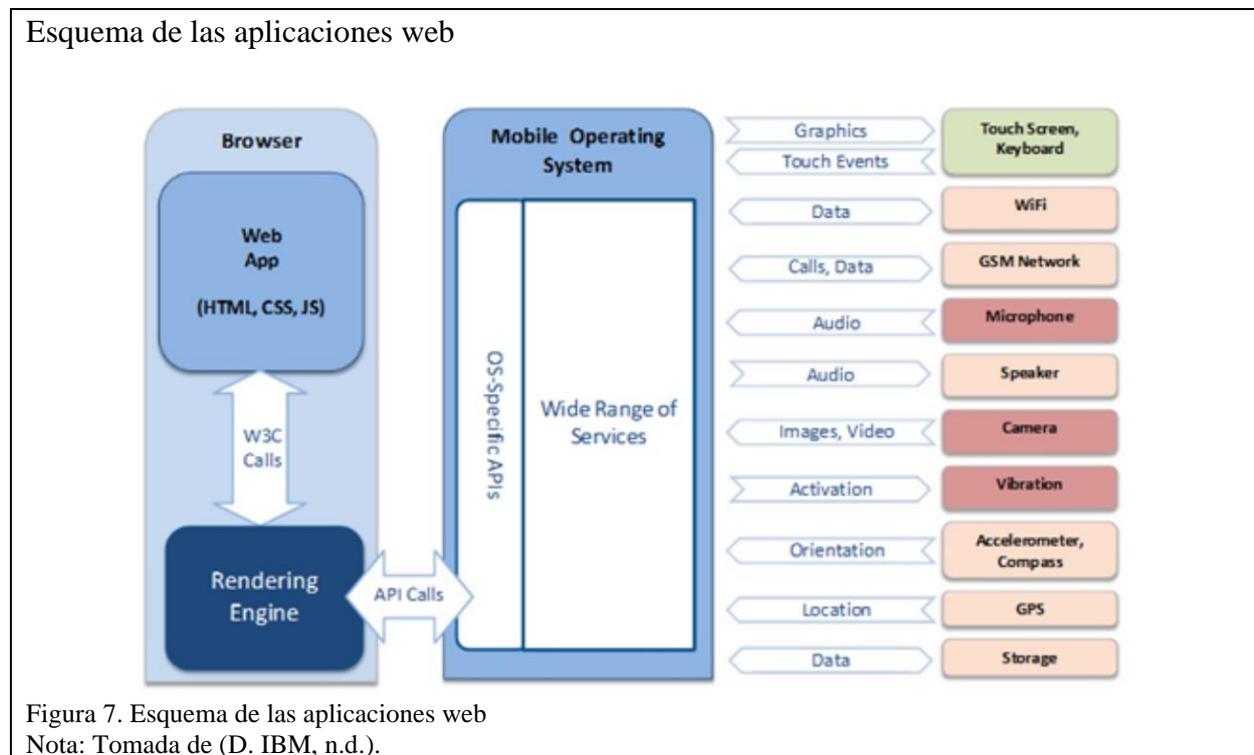
1.6.2 Aplicaciones Web

Las aplicaciones Web son aplicaciones que tienen una compilación única para distintas plataformas esto se debe a que todas las plataformas tienen un navegador web incorporado en el sistema operativo. Se diferencia de una página web responsiva ya que estas aplicaciones se comportan como las aplicaciones nativas, pero son ejecutadas en un acceso directo, estas aplicaciones se desarrollan usando tecnologías web como CSS, HTML y JavaScript (Jobe, 2013).

HTML5 es un lenguaje de marcado de páginas que ha ido evolucionando funcionalmente ya que se puede considerar un estándar de desarrollo de aplicaciones basadas en el navegador ya que tiene acceso a los servicios de Georeferenciación, interfaz gráfica, entre otras, de esta forma los desarrolladores pueden crear aplicaciones avanzadas usando tecnologías web (W. IBM, 2012).

La ventaja más notoria de las aplicaciones web es su capacidad multiplataforma ya que la mayor parte de las plataformas incluyen en su sistema operativo un navegador web, además el costo de desarrollo es bajo y no ocupan espacio en disco. Como se puede observar en la Figura 7, las aplicaciones son ejecutadas en el navegador predeterminado que envía y recibe peticiones,

estas se renderizan de modo que el motor de renderizado realiza peticiones a las API's del sistema operativo (Glera Aransay, 2013).



1.6.3 Aplicaciones Híbridas

Las aplicaciones híbridas combinan características de las aplicaciones nativas y las aplicaciones web ya que se comportan como aplicaciones web al usar tecnologías web como HTML, CSS y JavaScript en el desarrollo embebidas en aplicaciones nativas, en otras palabras, también tienen acceso al hardware nativo (Jobe, 2013).

Como se puede observar en la Figura 8 el esquema de las aplicaciones híbridas muestra como la aplicación tiene dos enfoques en porciones, la porción web de la aplicación utiliza las tecnologías web que son JavaScript, CSS, HTML, estas pueden ser solo un conjunto de archivos o alojarse en un servidor. Con el objetivo de ser multiplataforma y mantener la API's nativas del sistema. La porción nativa de la aplicación hace uso las API's nativas para generar un motor

de búsqueda incorporado que se comporta como un puente entre el navegador web y las API's del sistema operativo (W. IBM, 2012).

El puente puede ser codificado por los desarrolladores o usar un entorno de desarrollo que genere el puente automáticamente como Apache Cordova o PhoneGap que provee un interfaz JavaScript para las funcionalidades de los dispositivos (W. IBM, 2012).

La mayor ventaja de las aplicaciones híbridas es el puente antes mencionado ya que de esta forma obtiene las características más favorables entre una aplicación nativa (API's nativas) y una aplicación web (multiplataforma), Apache Cordova es un entorno de desarrollo de aplicaciones móviles híbridas que genera automáticamente el puente o las llamadas a los API's entre la aplicación híbrida que contiene la porción web y la porción nativa, y el sistema operativo móvil que accede directamente a los componentes del hardware como el Wi-Fi, acelerómetro, micrófono, cámara, etc. (W. IBM, 2012).

Esquema de las aplicaciones híbridas

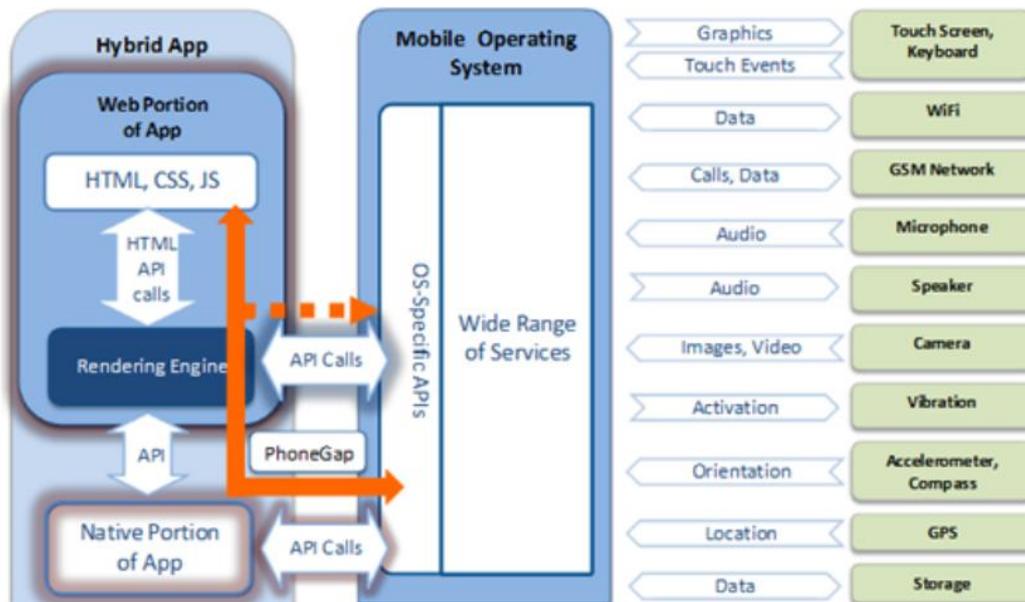


Figura 8. Esquema de las aplicaciones híbridas
Fuente: Tomada de (D. IBM, n.d.)

1.6.3.1 Interfaz de programación de aplicaciones (API)

Las API's antes mencionadas en todos los tipos de aplicaciones hacen referencia a la interfaz de programación de aplicaciones, estas se dividen en dos categorías: de bajo y de alto nivel (W. IBM, 2012).

API's de bajo nivel: Las API's de bajo nivel permiten la interacción directa de la aplicación con los componentes del hardware como la pantalla táctil, procesamiento de audio por micrófono o reproducción de audio por los auriculares, tiene acceso al GPS obteniendo la información de posicionamiento además de obtener permisos de escritura y lectura del disco (W. IBM, 2012).

API's de alto nivel: Para mejorar la experiencia del usuario las plataformas ofrecen a los desarrolladores servicios de alto nivel como por ejemplo navegación del explorador, gestión del calendario, llamadas telefónicas, algunas API's permiten acceso a los servicios en la nube como las compras en las tiendas oficiales de aplicaciones (W. IBM, 2012).

Interfaz gráfica de usuario: Cada plataforma ofrece un conjunto de API's conocidas como toolkit GUI, en otras palabras, componentes para la interfaz gráfica de usuario como botones, cuadros de dialogo, notificaciones, etc. Estas GUI no son genéricas, es decir, cada plataforma tiene sus propios componentes de interfaz gráfica por este motivo el diseñador debe conocer los componentes de cada plataforma a desarrollar (W. IBM, 2012).

1.6.4 Comparación de tipos de aplicaciones móviles

Existen criterios y características que se deben tomar en cuenta al momento de escoger el tipo de aplicación móvil a desarrollar, se muestra información resumida en la Tabla 3. Criterios de los tipos de aplicaciones.

Tabla 3. Criterios de los tipos de aplicaciones

Característica	App. Nativa	App. Híbrida	App. Web
Esfuerzo en compatibilidad de plataforma	Alto	Medio	Bajo
Acceso a las capacidades del dispositivo	Completo	Completo	Parcial
Experiencia del usuario	Completo	Completo	Medio
Funcionamiento	Muy Alto	Muy Alto	Alto
Actualización en el cliente	Necesario	Necesario	No necesario
Facilidad de publicación / distribución	Medio	Medio	Alto
Ciclo de aprobación	Obligatorio	Algunos casos	No requerido
Monetización en la tienda de aplicaciones	Disponible	Disponible	No disponible

Fuente: Tomada de (Serrano et al., 2013).

1.7 Apache Cordova

Apache Cordova es un entorno de desarrollo de aplicaciones híbridas en el cual primero se desarrolla la porción web usando tecnologías web luego se usa las herramientas de Apache

con el SDK de cada plataforma para el empaquetado de la porción web en la porción nativa para cada plataforma de destino (Wargo, 2015).

Una vez completa la aplicación Web se empaqueta en la porción nativa, cada plataforma tiene sus propias herramientas patentadas de crear aplicaciones móviles, para esto la porción web con sus archivos se agregan a un proyecto propio por plataforma que se construye usando estas herramientas. Debido a las herramientas propias los archivos de configuración varían y la estructura de las carpetas, por lo tanto se debe tener en consideración la compatibilidad que se muestra en la Figura 9 del proceso de creación de las aplicaciones compatibles según las plataformas en las cuales se desarrollan, por ejemplo: El SDK de Android de la misma forma que el SDK de BlackBerry está disponible en sistemas operativos como Windows, Mac OS y Linux, la línea de comandos Xcode de iOS está disponible solo en Mac OS y el SDK de Windows Phone está disponible solo en Windows (Wargo, 2015).

Proceso de creación de aplicaciones según plataforma

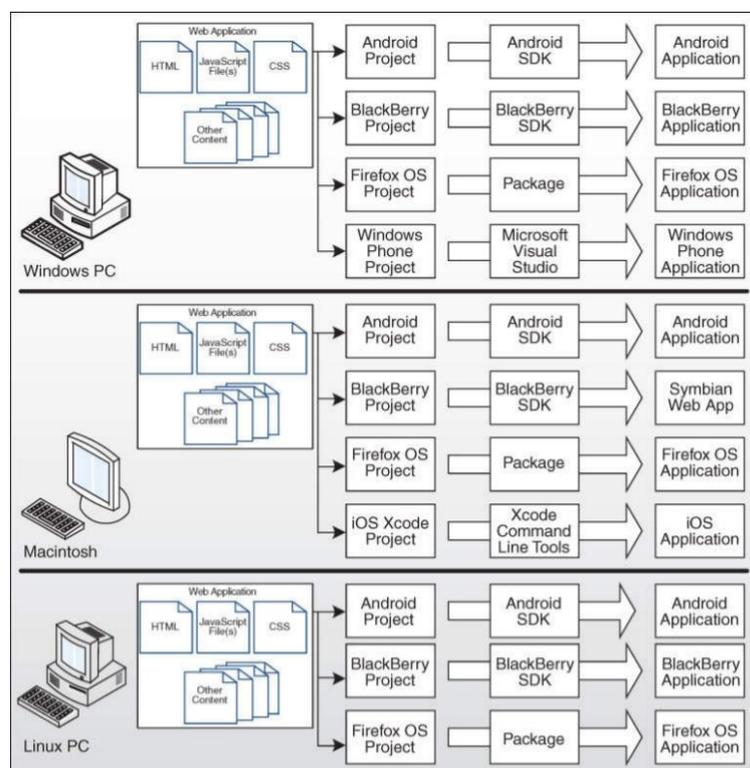


Figura 9. Proceso de creación de aplicaciones según plataforma
Fuente: Tomada de (Wargo, 2015).

1.7.1 Historia

Apache Cordova fue creado por Nitobi en 2008, en el evento iPhoneDevCamp su objetivo era crear una forma simplificada de desarrollar una aplicación multiplataforma, el equipo de desarrollo encargado del proyecto PhoneGap en 2 días de trabajo crearon la estructura del framework incluyendo la funcionalidad del contenedor de la aplicación nativa. En poco tiempo el proyecto fue creciendo y agregando el soporte para las plataformas Android y BlackBerry, con el paso de los años Apple aprobó al framework, después de hacerse público el proyecto PhoneGap, IBM y otras empresas comenzaron con el desarrollo de su propio entorno de desarrollo para aplicaciones móviles. En 2011 Nitobi anuncia la donación del proyecto a Apache, pero poco tiempo después Adobe anuncia la compra de la empresa Nitobi junto con el proyecto PhoneGap discontinúa el soporte del conocido Adobe Flash y tiempo más tarde Adobe libera el código para Apache llamándolo inicialmente Apache Callback, luego Apache DeviceReady y desde la versión 1.4 se conoce con el nombre de Apache Cordova que en realidad es el nombre de la calle de la oficina de Nitobi cuando PhoneGap fue creado. Apache Cordova fue consolidada como uno de los proyectos de Apache más utilizados por el cual las actualizaciones fueron de frecuencia mensual (Wargo, 2015).

Cuando Apple aceptó el framework de PhoneGap era necesario el uso de una computadora de Apple para desarrollar aplicaciones para iOS y una computadora con Windows para desarrollar aplicaciones para Windows Phone, desde el 2012 Adobe lanza una actualización de PhoneGap que marca la diferencia con Apache Cordova, PhoneGap tiene un compilador en la nube la cual permite que los desarrolladores carguen la porción web desde cualquier ordenador con cualquier sistema operativo y este devuelve las aplicaciones para las plataformas solicitadas (Wargo, 2015).

1.7.2 Componentes

La aplicación tiene una porción web en la cual se encuentran todos los archivos HTML, CSS, JavaScript, y archivos multimedia como las imágenes, etc. Estos archivos listos para su ejecución se presentan en un WebView dentro de una aplicación nativa que esta lista para su distribución en las tiendas oficiales de las plataformas. Los componentes de Apache Cordova son: el WebView, la aplicación Web y los complementos (Wargo, 2015).

WebView: Un WebView es un componente que representa contenido web de una aplicación nativa, está habilitado para Cordova y permite el acceso a interfaces de usuario (Wargo, 2015).

Aplicación Web: Esta porción web de la aplicación es donde se encuentre la codificación del desarrollo de la aplicación ya que esta se implementa como una página web con el archivo local “index.html” en el que hace referencia a todos los archivos multimedia, estilo, JavaScript, etc. El desarrollador puede cambiar esto. Esta porción de la aplicación corre en un WebView que esta encapsulado en un contenedor de aplicación nativa (Wargo, 2015).

Complementos: Contiene los enlaces directos a las API's del dispositivo y se puede llamar desde JavaScript a código nativo, estos complementos pueden ser proporcionados por Cordova llamados Core Plugins y por terceros, al desarrollar una aplicación se debe agregar manualmente los plugins o complementos (Wargo, 2015).

1.7.3 Arquitectura

Como se puede observar en la Figura 10 se muestra la arquitectura de una aplicación Cordova, estas aplicaciones tienen dos contenedores uno con la porción Web, con los archivos de codificación y multimedia que se comunica con el Motor de renderizado o WebView a su vez este se comunica con los Plugins de Cordova o complementos ya mencionados estos contenedores se encapsulan en una aplicación Cordova que se comunica mediante enlaces API's al sistema operativo (Apache Software Foundation, n.d.).

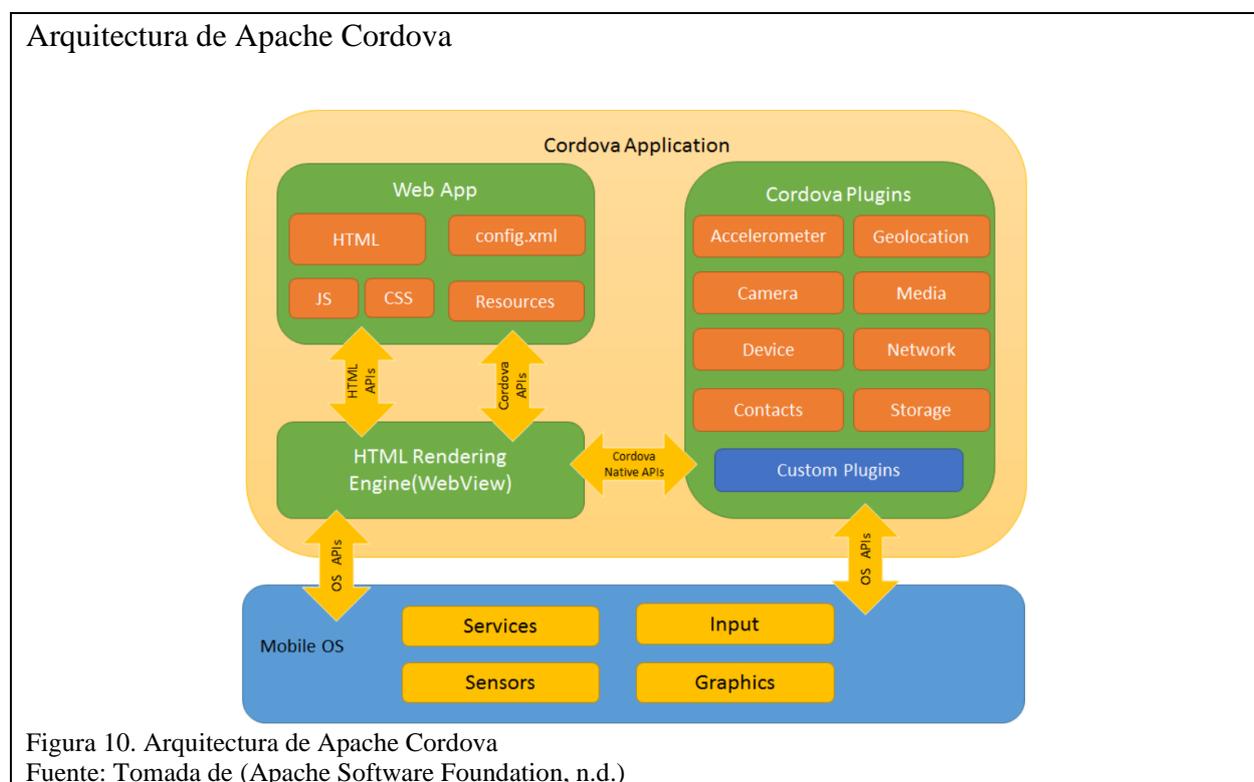


Figura 10. Arquitectura de Apache Cordova
Fuente: Tomada de (Apache Software Foundation, n.d.)

1.8 Metodologías para el desarrollo de videojuegos

La ingeniería de software determina la forma de organizar el trabajo requerido con metodologías o enfoques formales para el desarrollo y diseño de software, según el tamaño del proyecto, se requiere mayor trabajo o personal, para la mayor parte de los videojuegos los equipos son de 4 a 8 desarrolladores y con miles de líneas de código. Debido a estas

características las metodologías más usadas en el desarrollo de videojuegos son: cascada o tradicional y scrum (Bates, 2004).

La metodología en cascada tiene un enfoque lineal definido por etapas secuenciales aumentando la complejidad y el costo, cuando todas las etapas se han cumplido se obtiene el producto final. La metodología ágil tiene un enfoque basado en equipos, predomina la colaboración y la comunicación en la evolución de los requisitos (Bates, 2004; Keith, 2010).

1.9 Cascada

En la metodología en cascada o tradicional se desarrollan los proyectos de software pequeños y medianos donde puede existir un equipo pequeño o incluso solo un desarrollador a través de una serie de etapas, en la Figura 11 se muestra las etapas de la metodología para el desarrollo de videojuegos, en las etapas iniciales se constituye el concepto del software por ello los requisitos deben estar completamente definidos en estas etapas, en las siguientes etapas se diseña la arquitectura del software junto con el desarrollo y finalmente en las etapas finales se integra el software y se realiza las pruebas del sistema. El objetivo de esta metodología es reducir el riesgo antes de pasar a la etapa más costosa, de esta forma en la mayor parte de proyectos que siguen este modelo de desarrollo todos los requisitos del sistema son definidos en la etapa inicial, de esta forma si la mecánica de los proyectos cambia y se aumentan los requisitos con el desarrollo esta metodología no es adecuada en este tipo de proyectos (Bates, 2004; Keith, 2010).

Metodología en cascada en el desarrollo de videojuegos

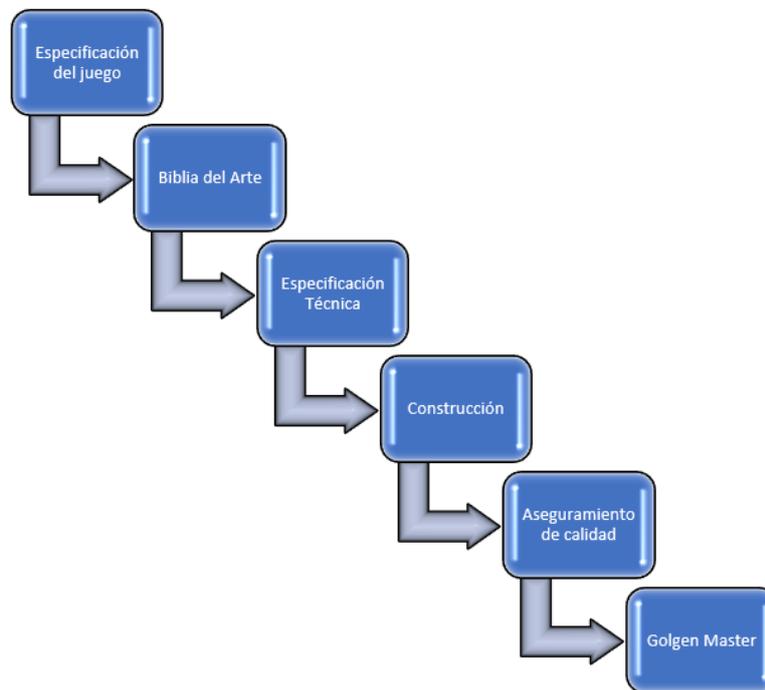


Figura 11. Metodología en cascada en el desarrollo de videojuegos

Nota: Adaptado de Ingeniería de software aplicada al desarrollo de Videojuegos por Chaparro García, 2015.

1.9.1 Fases

La metodología en cascada se define por las etapas o fases que se describen a continuación

1.9.1.1 Especificación del Juego

En esta fase se tiene un entregable este consta del detalle del juego desde la perspectiva del usuario final o jugador. Esta fase es el equivalente a la etapa de planeación en el modelo tradicional de cascada en el desarrollo de software (Chaparro García, 2015).

1.9.1.2 Biblia del Arte

El entregable en esta fase es determinado por los productores artísticos en el cual se define la historia, el guion, herramientas a usar y el diseño del juego. Esta fase es el equivalente

a la etapa de análisis en el modelo tradicional de cascada en el desarrollo de software (Chaparro García, 2015).

1.9.1.3 Especificación Técnica

El entregable en esta fase describe las herramientas de ingeniería, contenido técnico, etc. Define la implementación del videojuego por lo tanto se presenta diagramas UML (Lenguaje unificado de modelado) del funcionamiento y las interacciones del jugador con el juego, se define también las características técnicas del proyecto y los equipos de trabajo. Esta fase es el equivalente a la etapa de diseño en el modelo tradicional de cascada en el desarrollo de software (Chaparro García, 2015; Moya et al., 2019).

1.9.1.4 Construcción

En esta fase comienza el desarrollo, programación, modelos, texturas y las tareas que se describen en las etapas: biblia del arte y especificación técnica; se crean librerías y código reutilizable con el fin de agilizar el desarrollo, se desarrolla un demo corto del juego. Esta fase es el equivalente a la etapa de codificación en el modelo tradicional de cascada en el desarrollo de software (Chaparro García, 2015).

1.9.1.5 Aseguramiento de Calidad

En esta fase el equipo de QA (control de calidad) se encarga de comprobar el funcionamiento y cumplimiento de los requisitos del juego, se realizan reuniones con el equipo de producción y dirección para obtener un producto estable y la fase se divide en dos partes una primera con las pruebas alfa en la que un grupo previamente seleccionado se encargan de evaluar la calidad del juego y se determinan pequeños cambios, una vez terminada esta parte

de la fase comienza la siguiente que consiste en las pruebas beta, el juego se libera para una gran cantidad de personas donde se determina el tipo de mercado y los aspectos favorables del juego. Esta fase es el equivalente a la etapa de pruebas en el modelo tradicional de cascada en el desarrollo de software (Chaparro García, 2015).

1.9.1.6 *Golden Master*

Esta es la fase final en la cual el juego se considera estable y listo para su liberación a un público masivo. Esta fase es el equivalente a la etapa de mantenimiento en el modelo tradicional de cascada en el desarrollo de software (Chaparro García, 2015).

1.9.2 Ventajas y Desventajas

Existen criterios y características que se deben tomar en cuenta al momento de escoger una metodología como se puede observar en la Tabla 4. Ventajas y desventajas de la metodología en cascada se muestra las ventajas y desventajas de la metodología en cascada

Tabla 4. Ventajas y desventajas de la metodología en cascada

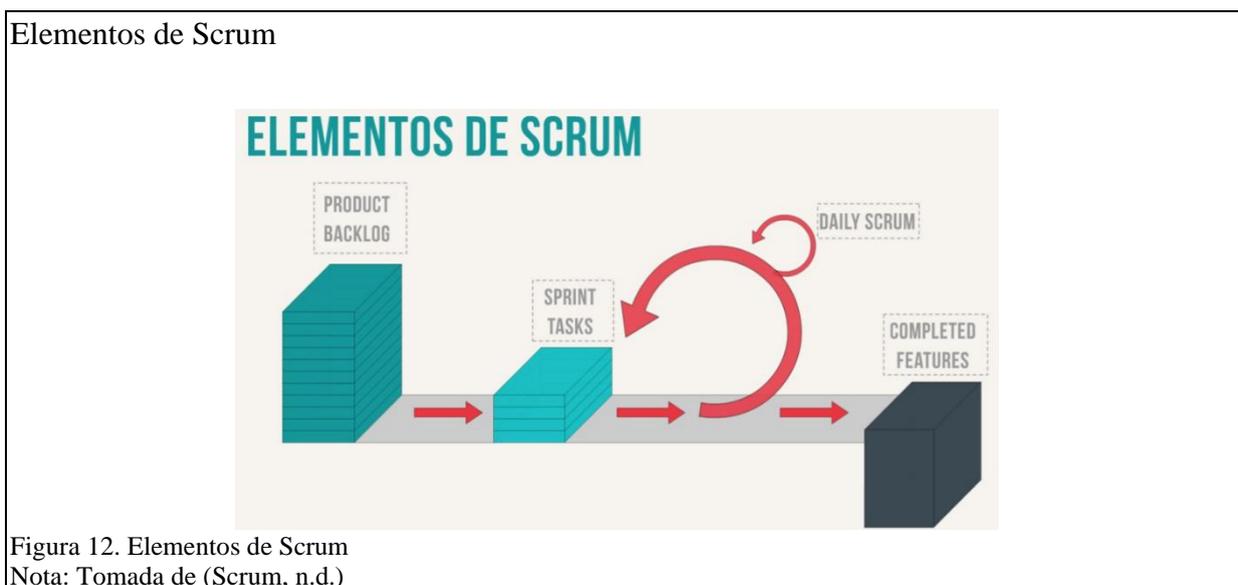
Ventajas	Desventajas
La planificación del juego es simple.	La duración de todo el ciclo es muy larga.
Si no surgen cambios grandes en las fases el producto final es de alta calidad.	Existe poca o nula comunicación con el usuario final.
Fácil organización al ser un flujo.	Los videojuegos muy grandes y complejos son difíciles de revisar.
Las fases son separadas y no se realizan en paralelo.	El orden de las fases debe mantenerse o el producto final será de menor calidad.

Fácil uso de esta metodología si el juego tiene bien definido los requisitos y las herramientas a usar.	Difícil respuesta a los cambios de requerimientos.
---	--

Fuente: Tomada de (Urrutia et al., 2015).

1.10 Scrum

Esta metodología tiene un enfoque colaborativo entre equipos de trabajo de seis a diez personas, con lo que se conoce la evolución del producto, el desarrollo del sistema es incremental por ello en lugar de usar el enfoque secuencial como el de la metodología de cascada se basa en la iteración, cada una de estas iteraciones contiene todas las fases de una metodología secuencial. La principal diferencia es la duración del proyecto, cada iteración puede durar entre dos a cuatro semanas y en secuencial cada fase puede durar de un mes a un año. El objetivo de esta metodología es la comunicación del equipo en el desarrollo del sistema y el ajuste de los interesados a la planificación del proyecto en función de la realidad, si surgen cambios en la producción estas no influyen en el desarrollo (Aalto, 2015). En la Figura 12 se muestra los elementos de Scrum.



1.10.1 Roles de Scrum

Ya que en Scrum se divide el trabajo en equipos los equipos deben tener claro su rol y responsabilidad. Los roles descritos por la metodología son:

Equipo Scrum-Scrum Master: El equipo Scrum se compone de un Scrum Master, el dueño del producto, y el equipo de desarrollo. El Master es el encargado de que el equipo conozca sobre la metodología y se siga las reglas establecidas con los tiempos determinados incluso es el que debe resolver los problemas que se presenten, se asegura del éxito de Scrum en el proyecto aplicando buenas prácticas y principios (Keith, 2010).

Dueño del producto: El dueño del producto o product owner es el encargado de la comunicación, prioriza las tareas a realizar. El product owner es el encargado de manejar el retorno de la inversión (ROI), establece que se va a construir además establece el orden y comparte la visión general del juego al equipo de desarrollo, realiza una retroalimentación sobre las tareas que se van realizando por el equipo (Keith, 2010).

Equipo de desarrollo: El equipo de desarrollo es el encargado de desarrollar y completar la tarea que fueron asignados por el product owner, el equipo debe estar conformado por profesionales como QA, programadores, modeladores de personajes con un nivel alto de experiencia que se aseguran de terminar todo el proyecto. (Keith, 2010).

1.10.2 Ventajas y Desventajas

Existen criterios y características que se deben tomar en cuenta al momento de escoger una metodología como se puede observar en la Tabla 5. Ventajas y desventajas de la metodología scrum se muestra las ventajas y desventajas de la metodología Scrum

Tabla 5. Ventajas y desventajas de la metodología scrum

Ventajas	Desventajas
Aumenta la productividad.	En algunos casos debe ser complementada con otro proceso ágil
El producto final son en realidad partes pequeñas dinámicas de fácil manejo.	La documentación es menor en comparación con otras metodologías.
Visibilidad para todo el equipo.	Una mala organización puede tener un producto inconcluso.
Las mejoras del producto se comunican al cliente.	Si los equipos de trabajo son muy grandes se debe dividirlos en grupos pequeños
Después de cada ciclo o iteración se entrega un producto funcional.	El scrum master debe confiar en su equipo o de lo contrario un control demasiado exigente causará el fracaso del proyecto.
Ajuste de funcionalidad en base a los criterios de los jugadores.	Los equipos deben estar conformados por profesionales con un nivel de experiencia, no es para equipos junior
Posibilidad de modificar los requerimientos sin interrumpir el desarrollo.	Existen problemas con la finalización del proyecto si en el contrato consta una fecha límite.

Fuente: Tomada de (Urrutia et al., 2015).

1.11 Modelo de Calidad de experiencia del usuario basado en jugabilidad

El modelo PMQ (Playability Quality Model) representa el rango de valor del juego mediante metas específicas logradas por los jugadores (J. L. G. Sánchez & Vela, 2014).

El modelo se basa en factores de calidad (Efectividad, eficiencia, flexibilidad, seguridad y satisfacción) y métricas asignadas a estos factores con las que se puede obtener una estimación cuantificable de cada factor, el modelo presenta un ejemplo de las métricas que se puede observar en la Tabla 6. Ejemplo de métricas por el modelo PQM (J. L. G. Sánchez & Vela, 2014).

Tabla 6. Ejemplo de métricas por el modelo PQM

Factor	Métrica	Propósito	Fórmula	Interpretación
Efectividad	Efectividad en la meta	¿Qué porcentaje de metas y retos se han alcanzado correctamente?	$X = 1 - \sum A_i $ A _i : Valor proporcional de cada acción incorrecta	X ∈ [0, 1], cercano 1, lo mejor
	Complejidad de la Meta	¿Qué porcentaje de metas y retos se han completado?	$X = A/B$ A = n. de metas completadas B = n. total de metas intentadas	X ∈ [0, 1], cercano 1, lo mejor
	Frecuencia de Intentos por Meta	¿Cuál ha sido la frecuencia de intentos?	$X = A/T$ A = No de intentos realizados por jugador T = tiempo o número de metas	Jugador experto cercano a 0. Al comienzo > 0
Eficiencia	Tiempo de Meta	¿Cuánto tiempo requiere el jugador para lograr una meta?	$X = T_a$	Jugadores novatos necesitan más tiempo
	Eficiencia de Meta	¿Cómo de eficiente es el usuario?	$X = M1/T$	X ∈ [0,1], cercano a valores intermedios
	Eficiencia Relativa al Nivel del Usuario	¿Cómo de eficiente es un jugador experto frente a un jugador nuevo?	$X = A/B$ A = eficiencia del jugador normal B = eficiencia del jugador experto	X ∈ [0, 1], cercano 1, lo mejor
Flexibilidad	Accesibilidad	¿Qué porcentaje de metas se logran utilizando distintas formas de interacción diferentes a las usadas por defecto?	$X = A/B$ A = metas con diferentes métodos de interacción B = No total de metas	X ∈ [0, 1], cercano 1 lo mejor
	Personalización	¿Qué proporción de la personalización disponible utiliza el jugador?	$X = A/B$ A = elementos personalizables B = elementos en el juego	X ∈ [0, 1], cercano a 1 métodos de interacción originales quizás deban ser cambiados
Seguridad	Seguridad y Salud del Jugador	¿Cómo incide en la salud del jugador el uso del producto?	$X = 1 - A / B$ A = No de jugadores que informan de problemas relacionados con la seguridad B = número total de jugadores	X ∈ [0, 1], cercano 1 lo mejor
	Daño software	¿Cómo incide la corrupción del software en el juego?	$X = 1 - A / B$ A = número de veces que el videojuego falla y es detectado por el jugador. B = No total de situaciones de uso	X ∈ [0, 1], cercano 1 lo mejor
	Escala de Satisfacción	¿Cómo de satisfecho está el jugador?	$X = A/B$ A = escala B = media popular	X > 0 el mayor, lo mejor
	Cuestionario de Satisfacción	¿Cómo de satisfecho está el jugador con las características	$X = \sum A_i / B$ A _i = respuesta a la pregunta B = número de respuestas	Comparar con valores previos, o con la media popular

Satisfacción		propias del videojuego?		
	Preferencia de Uso	¿Qué porcentaje de usuarios prefieren el videojuego frente a otro?	$X = A/B$ A= No de veces que características propias del juego es usada B= No de veces que jugadores intentan jugar a un juego	$X \in [0, 1]$, cercano 1, lo mejor
	Socialización	¿Qué porcentaje de los retos son resueltos jugando en grupo?	$X = A/B$ A = No de veces que el juego se usa en un contexto social B = No de veces que el juego es usado	$X \in [0,1]$, cercano 1, juego social, cercano a 0, juego individual

Fuente: Tomada de (J. L. G. Sánchez & Vela, 2014).

2 CAPÍTULO 2: DESARROLLO DEL VIDEOJUEGO

En este capítulo se muestra el desarrollo del videojuego siguiendo las fases de la metodología en Cascada.

2.1 Especificación del juego

El videojuego comienza con una página principal en el cual se muestra los países del mundo, el jugador es capaz de escoger el país que desea, acto seguido escoge el rompecabezas que desea realizar y el nivel de este.

Los rompecabezas son de tipo deslizante cuyo objetivo es reconstruir la imagen original deslizando las piezas de forma vertical u horizontal hasta conseguir el objetivo, estos rompecabezas tienen niveles de dificultad que son: 3x3, 4x4 y 5x5. Cada rompecabezas corresponde a categorías por país como bandera, moneda, flora y fauna.

Una vez el jugador finaliza todos los rompecabezas vuelve a la pantalla principal en el cual puede escoger otro país y/o rompecabezas.

2.2 La biblia del arte

En esta sección se muestra la historia, el guion, herramientas a usar, el diseño y los conceptos artísticos del juego

Concepto: El concepto y descripción del juego son descritas en la Tabla 7. Concepto del juego.

Tabla 7. Concepto del juego

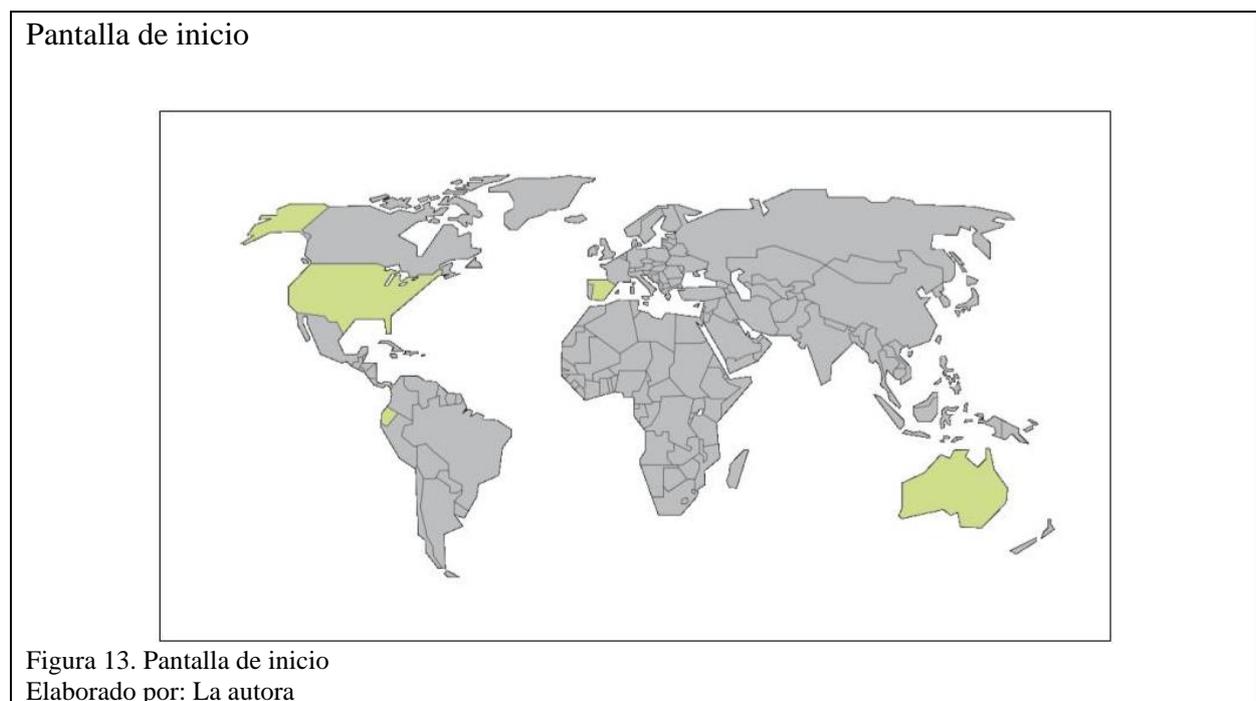
Característica	Descripción
----------------	-------------

Nombre del Juego	GEO-PUZZLE
Tipo de Juego	Educativo, Casual
Género	Juego de Agilidad Mental
Pilares	Lógica, Rompecabezas, Geografía.
Estudio Desarrollador	Desarrollado por Jezzrel Nicole Vaca Cabrera
Motor de Desarrollo	Apache Cordova
Plataforma	iOS y Android
Clasificación	(E) Contenido dirigido para todo público
Jugadores	Modalidad de juego individual
Objetivos	Resolver todos los rompecabezas
Características	<p>Continentes</p> <p>Por el modelo de 5 continentes se describe a: América, Europa, Asia, África y Oceanía.</p> <p>Países</p> <p>Cada continente tiene 3 países disponibles</p> <p>Rompecabezas</p> <p>Los rompecabezas tienen características propias como la bandera, moneda, flora y fauna endémica del país.</p>

Diseño de Niveles	<p>Rompecabezas, consta de 3 niveles de complejidad, esta se define por la dimensión del tablero</p> <p>Nivel 1: Rompecabezas de tamaño de 3x3</p> <p>Nivel 2: Rompecabezas de tamaño de 4x4</p> <p>Nivel 3: Rompecabezas de tamaño de 5x5</p>
-------------------	--

Elaborado por: La autora

Bocetos o Mockup: En esta sección se muestra el diseño del maquetado del juego, la página principal como se observa en la Figura 13, la selección del país como se observa en la Figura 14, la página del rompecabezas como se observa en la Figura 15, la selección del menú como se observa en la Figura 16, la visualización de la imagen original como se puede observar en la Figura 17, la visualización de la descripción de la imagen del rompecabezas como se observa en la Figura 18 y la página final una vez realizado el rompecabezas como se observa en la Figura 19.



Interfaz siguiente

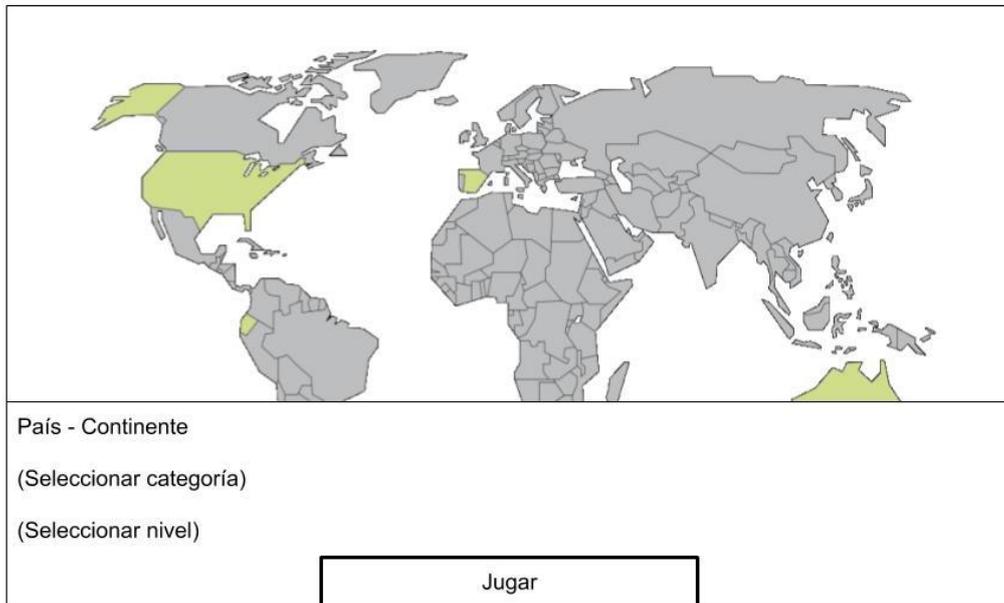


Figura 14. Selección de un país
Elaborado por: La autora

Interfaz siguiente

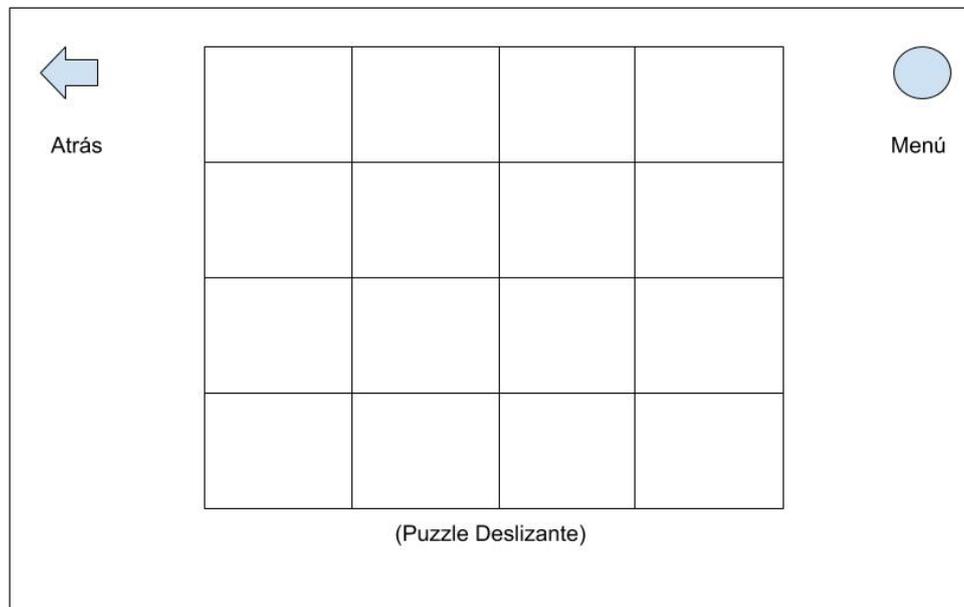


Figura 15. Interfaz rompecabezas
Elaborado por: La autora

Interfaz siguiente

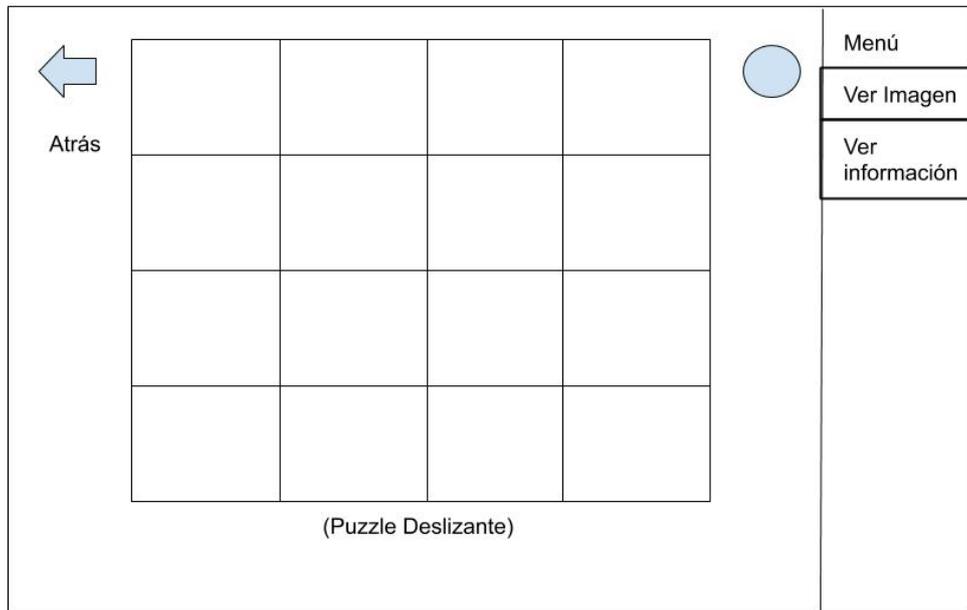


Figura 16. Interfaz menú del rompecabezas
Elaborado por: La autora

Interfaz siguiente

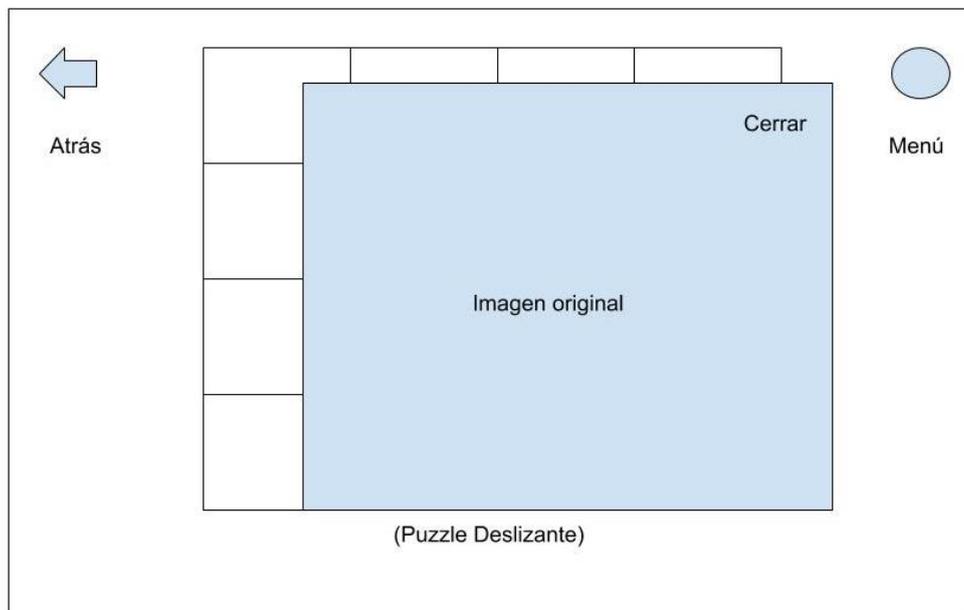


Figura 17. Interfaz imagen original del rompecabezas
Elaborado por: La autora

Interfaz siguiente

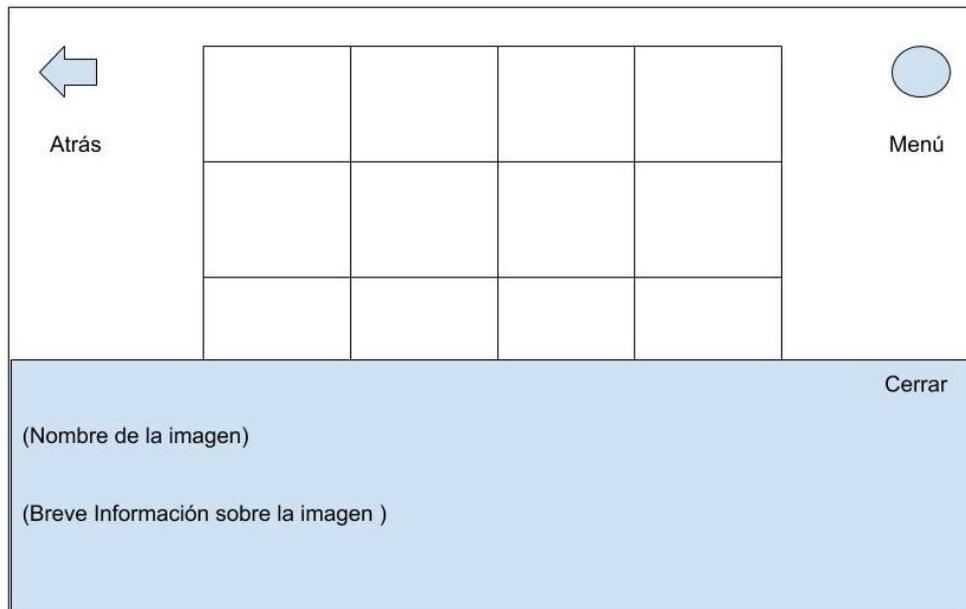


Figura 18. Interfaz información de la imagen del rompecabezas
Elaborado por: La autora

Interfaz siguiente

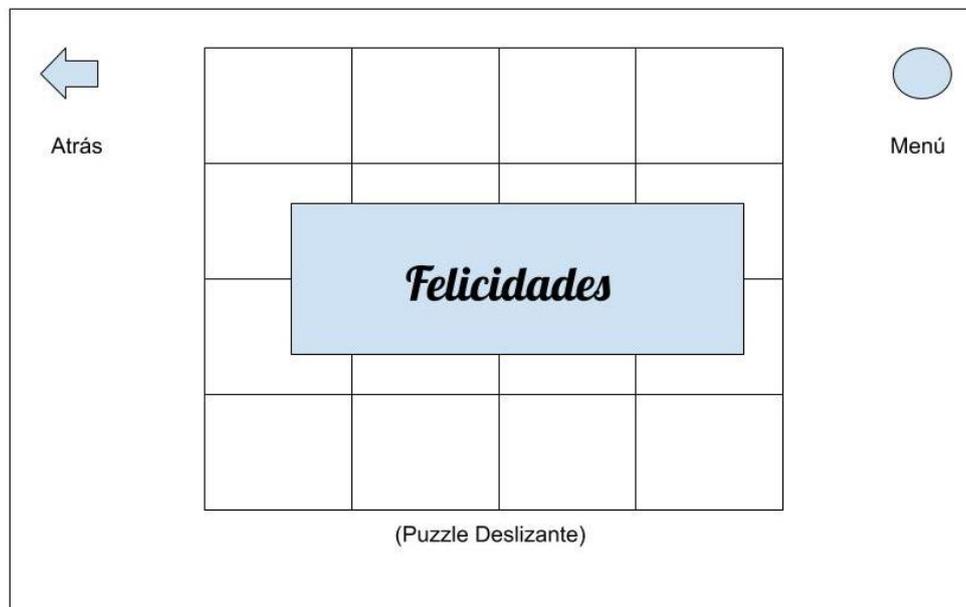
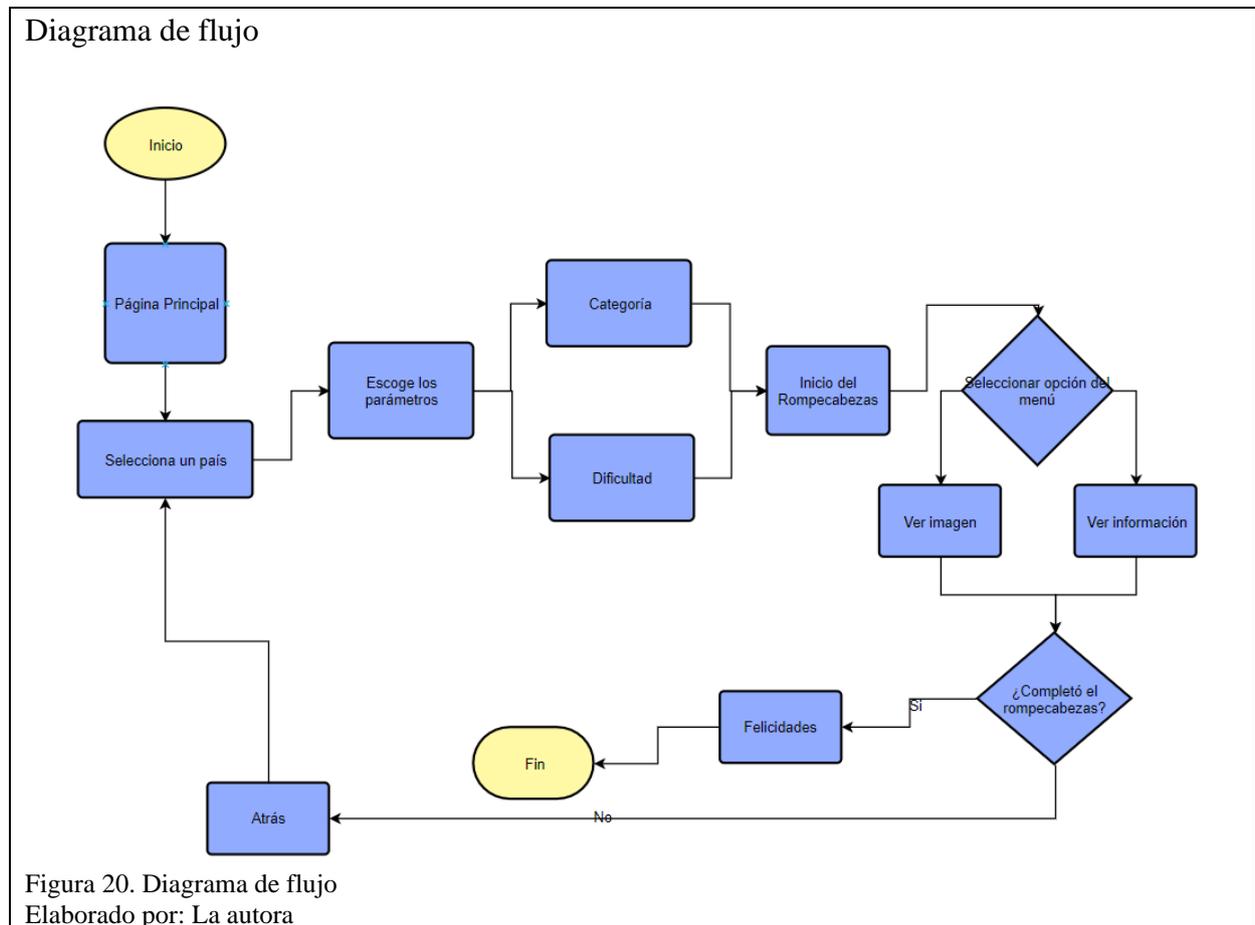


Figura 19. Interfaz final al terminar un rompecabezas
Elaborado por: La autora

2.3 Especificaciones Técnicas

Las especificaciones técnicas del juego se evidencian con el diagrama de flujo, los casos de uso y las especificaciones de dichos casos de uso. El diagrama de flujo del juego se muestra en la Figura 20.



En la Figura 21. Caso de uso pantalla principal se muestra las acciones del actor cuando este inicia la aplicación y se encuentra en la pantalla principal. En la Tabla 8. Especificación del caso de uso se muestra la especificación del caso de uso número 1, Tabla 9. Especificación del caso de uso se muestra la especificación del caso de uso número 2 y Tabla 10. Especificación del Caso de Uso se muestra la especificación del caso de uso número 3.

Caso de uso de la pantalla principal

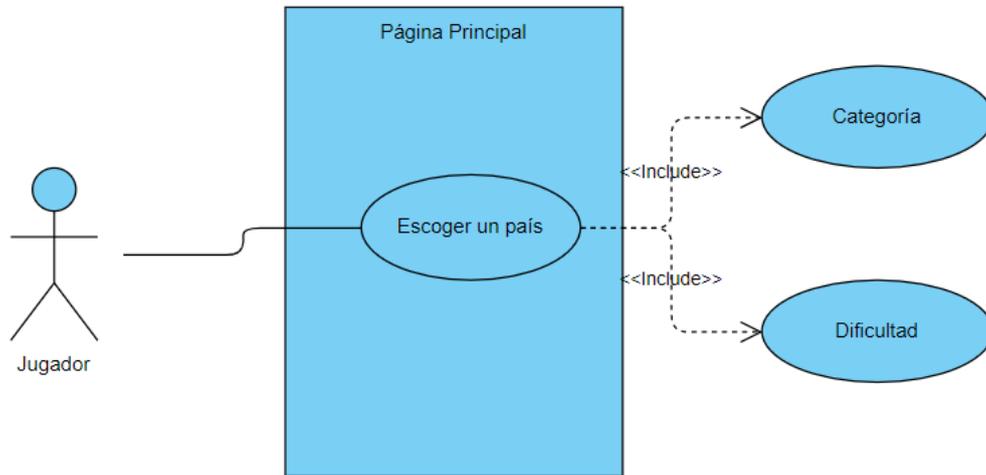


Figura 21. Caso de uso pantalla principal
Elaborado por: La autora

Tabla 8. Especificación del caso de uso

Identificador CU1	
Nombre	Escoger un país
Descripción	El jugador tiene la capacidad de escoger cualquier país disponible de la aplicación.
Actor	Jugador
Precondición	Iniciar la aplicación
Postcondición	Muestra una barra en la cual el jugador debe escoger una categoría y la dificultad del rompecabezas.
Escenario	El jugador inicia la aplicación y se muestra un mapa interactivo en el que se puede escoger un país, se muestra una barra con la información sobre el país es decir el nombre y el continente

	al que pertenece. Posteriormente el jugador debe escoger el nivel y la categoría
--	--

Elaborado por: La autora

Tabla 9. Especificación del caso de uso

Identificador CU2	
Nombre	Categoría
Descripción	El jugador tiene la capacidad de escoger la categoría de cada país que son: Bandera, Moneda, Fauna y Flora.
Actor	Jugador
Precondición	Escoger un país
Postcondición	Muestra una barra de selección en la cual el jugador debe escoger la dificultad del rompecabezas.
Escenario	El jugador escoge un país del mapa interactivo y elegir la categoría a jugar

Elaborado por: La autora

Tabla 10. Especificación del Caso de Uso

Identificador CU3	
Nombre	Dificultad

Descripción	El jugador tiene la capacidad de escoger la dificultad de cada país que es el tamaño del tablero, los niveles son: 3x3, 4x4 y 5x5.
Actor	Jugador
Precondición	Escoger un país
Postcondición	Muestra un botón de Jugar el cual el jugador debe presionar cuando escoge sus opciones.
Escenario	El jugador escoge un país del mapa interactivo, elige la categoría y la dificultad del rompecabezas a jugar.

Elaborado por: La autora

En la Figura 22 se muestra las acciones del actor cuando se encuentra en la pantalla del rompecabezas. En la Tabla 11. Especificación del caso de uso se muestra la especificación del caso de uso número 4, Tabla 12. Especificación del caso de uso se muestra la especificación del caso de uso número 5, Tabla 13. Especificación del caso de uso se muestra la especificación del caso de uso número 6, Tabla 14. Especificación del caso de uso se muestra la especificación del caso de uso número 7 y Tabla 15. Especificación del caso de uso se muestra la especificación del caso de uso número 8.

Caso de uso de la pantalla del rompecabezas

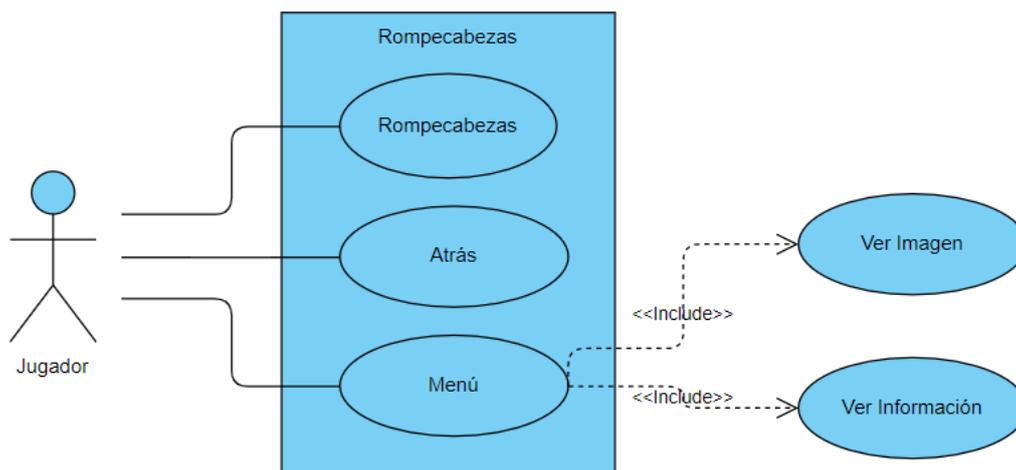


Figura 22. Caso de uso pantalla rompecabezas
Elaborado por: La autora

Tabla 11. Especificación del caso de uso

Identificador CU4	
Nombre	Rompecabezas
Descripción	El jugador debe resolver el rompecabezas deslizante con el nivel de dificultad seleccionado.
Actor	Jugador
Precondición	El jugador debe presionar el botón jugar
Postcondición	Una vez completado el rompecabezas se muestra un mensaje felicitando al jugador por completar el rompecabezas

Escenario	Luego de escoger la categoría y dificultad, se muestra una pantalla con el rompecabezas de tipo deslizante.
------------------	---

Elaborado por: La autora

Tabla 12. Especificación del caso de uso

Identificador CU5	
Nombre	Atrás
Descripción	El jugador tiene la capacidad de cancelar en cualquier momento.
Actor	Jugador
Precondición	Debe estar en la pantalla para resolver el rompecabezas
Postcondición	Al presionar atrás se mostrará la pantalla principal
Escenario	Una vez el jugador pulsa el botón atrás, vuelve a la pantalla principal en el cual debe escoger un país

Elaborado por: La autora

Tabla 13. Especificación del caso de uso

Identificador CU6	
Nombre	Menú
Descripción	El jugador tiene la capacidad de ver menú en cualquier momento.

Actor	Jugador
Precondición	Debe estar en la pantalla para resolver el rompecabezas
Postcondición	Al presionar menú se mostrará dos opciones para escoger, el jugador debe elegir entre ver imagen y ver información.
Escenario	Una vez el jugador pulsa el botón menú se abre una barra lateral en la cual tiene dos opciones para escoger entre ver imagen y ver información, para salir de la barra lateral el jugador solo debe dar un toque afuera de la barra.

Elaborado por: La autora

Tabla 14. Especificación del caso de uso

Identificador CU7	
Nombre	Ver Imagen
Descripción	El jugador tiene la capacidad de ver la imagen original sin fragmentar del rompecabezas en cualquier momento.
Actor	Jugador
Precondición	Debe estar en la pantalla para resolver el rompecabezas y pulsar en menú

Postcondición	Debe cerrar la imagen original para continuar con el rompecabezas
Escenario	Una vez el jugador pulsa el botón menú se abre una barra lateral en la cual tiene dos opciones para escoger que son ver imagen y ver información, al presionar ver imagen se muestra una nueva pantalla superpuesta al rompecabezas donde se muestra la imagen original.

Elaborado por: La autora

Tabla 15. Especificación del caso de uso

Identificador CU8	
Nombre	Ver Información
Descripción	El jugador tiene la capacidad de ver la información sobre la imagen del rompecabezas en cualquier momento.
Actor	Jugador
Precondición	Debe estar en la pantalla para resolver el rompecabezas y pulsar menú
Postcondición	Debe cerrar la barra de información para continuar con el rompecabezas
Escenario	Una vez el jugador pulsa el botón menú se abre una barra lateral en la cual tiene dos opciones

	para escoger que son ver imagen y ver información, al presionar ver información se muestra una nueva pantalla superpuesta al rompecabezas donde se muestra la información sobre la imagen del rompecabezas.
--	---

Elaborado por: La autora

2.4 Construcción

Esta fase del videojuego se refiere a las herramientas usadas para que el juego sea realizado con éxito.

Entorno de Desarrollo: El entorno de desarrollo utilizado para la creación de la aplicación híbrida es Apache Cordova.

Frameworks: Para la realización de este juego se utilizó Framework 7, un marco de código abierto para el desarrollo de la aplicación híbrida con la apariencia nativa, es decir, el desarrollo del menú; además se usó Phaser, un marco de código abierto para juegos en JavaScript.

Programación: La programación se realiza con tecnologías web que son HTML, CSS y JavaScript, en la Figura 23 se puede observar la llamada al mapa por el cual se puede escoger un país.

Código Fuente

```
var world = mapa_mundi;

var svg = d3
  .select("div#container")
  .append("svg")
  .attr("viewBox", "0 0 " + w + " " + h)
  .classed("svg-content", true);

var projection = d3
  .geoMercator()

var path = d3.geoPath().projection(projection);

var g = svg.append("g");

g.attr("class", "boundary")

.selectAll("boundary")
  .data(topojson.feature(world, world.objects.countries).features)
  .enter()
  .append("path")
  .attr("fill", function(d) {

    try {

      return map_details[d.id]['color'];

    } catch (error) {

    }})
```

Figura 23. Código Fuente
Elaborado por: La autora

2.5 Aseguramiento de calidad

En esta fase el videojuego se somete a pruebas de funcionamiento, en el siguiente capítulo se toma en cuenta dos tipos de parámetros como medida de calidad del videojuego que son: jugabilidad y conocimientos.

Prueba 1:

Luego de instalar la aplicación, debe tener el nombre de “GeoPuzzle”.

Como se comprueba en la Figura 24.

Captura de pantalla de la instalación y nombre de la aplicación



Figura 24. Captura de pantalla de la instalación y nombre de la aplicación.
Elaborado por: La autora

Prueba 2:

Al iniciar la aplicación se debe mostrar una bienvenida y una breve explicación del funcionamiento.

Como se comprueba en la Figura 25

Captura de pantalla de la pantalla de inicio



Figura 25. Captura de pantalla de la pantalla de inicio
Elaborado por: La autora

Prueba 3:

Al presionar sobre un país disponible (color amarillo) se despliega un menú en el cual se muestra el nombre del país seguido del continente al que pertenece y las opciones para resolver el rompecabezas como la categoría y la dificultad.

Como se comprueba en la Figura 26. Captura de pantalla de la elección del país

Captura de pantalla de la elección del país

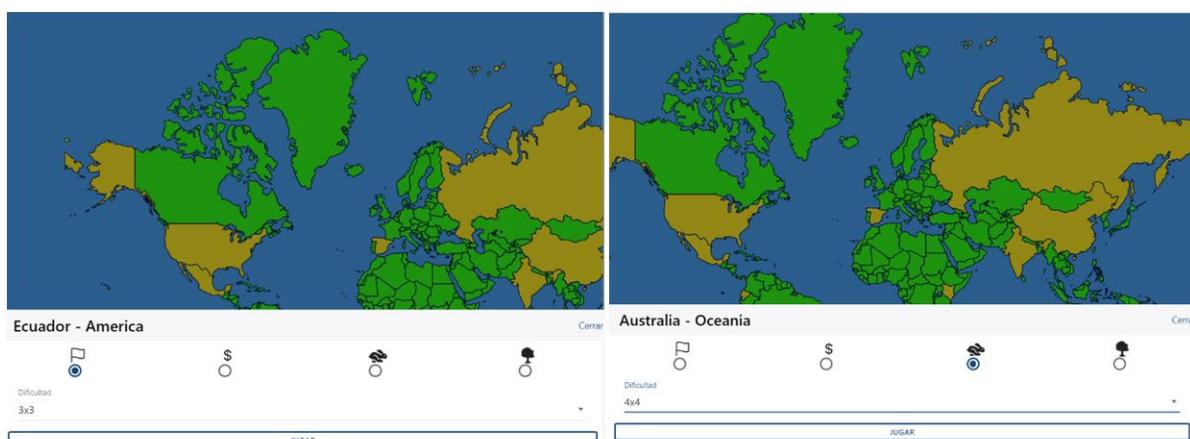


Figura 26. Captura de pantalla de la elección del país
Elaborado por: La autora

Prueba 4:

Una vez seleccionada la categoría y el nivel o dificultad se debe mostrar el rompecabezas listo para jugar, sin embargo, se debe mostrar también en la esquina superior derecha un icono de menú y en la esquina superior izquierda un icono con la flecha atrás.

Como se comprueba en la Figura 27

Captura de pantalla del rompecabezas



Figura 27. Captura de pantalla del rompecabezas
Elaborado por: La autora

Prueba 5:

Escenario 1:

Una vez en la pantalla del rompecabezas se muestran dos iconos, el primero en la esquina superior izquierda con una flecha que es botón hacia atrás, cuando se presione el botón la pantalla debe volver a la inicial en la que se puede escoger un país.

Como se comprueba en la Figura 28. Captura de pantalla del botón atrás.

Captura de pantalla del botón atrás



Figura 28. Captura de pantalla del botón atrás
Elaborado por: La autora

Escenario 2:

Una vez en la pantalla del rompecabezas se muestran dos iconos, el segundo en la esquina superior derecha con el diseño de menú, cuando se presione el botón un panel lateral derecho debe desplegarse mostrando dos opciones que son: ver imagen y ver información.

Como se comprueba en la Figura 29.

Captura de pantalla del botón menú



Figura 29. Captura de pantalla del botón menú
Elaborado por: La autora

Prueba 6:

Escenario 1:

Una vez desplegada las opciones del menú, se selecciona la opción con el nombre “Ver Imagen”, al presionar esta opción se debe mostrar una ventana en la cual se observa la imagen original del rompecabezas que se está realizando, adicional en la ventana que se muestra en la esquina superior derecha se debe encontrar la opción de cerrar para salir de la ventana y continuar con el rompecabezas.

Como se comprueba en la Figura 30

Capturas de pantalla del menú opción ver imagen

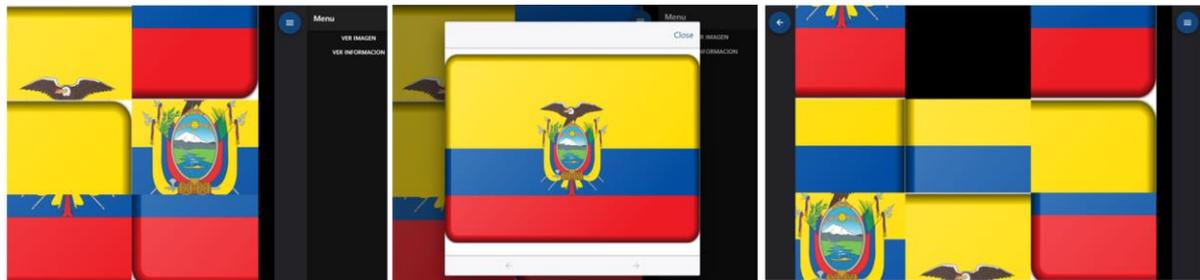


Figura 30. Capturas de pantalla del menú opción ver imagen
Elaborado por: La autora

Escenario 2:

Una vez desplegada las opciones del menú, se selecciona la opción con el nombre “Ver Información”, al presionar esta opción se debe mostrar una ventana inferior en la cual se despliegue la información sobre la imagen que está en el rompecabezas.

Como se comprueba en la Figura 31. Capturas de pantalla del menú opción ver información

Capturas de pantalla del menú opción ver información

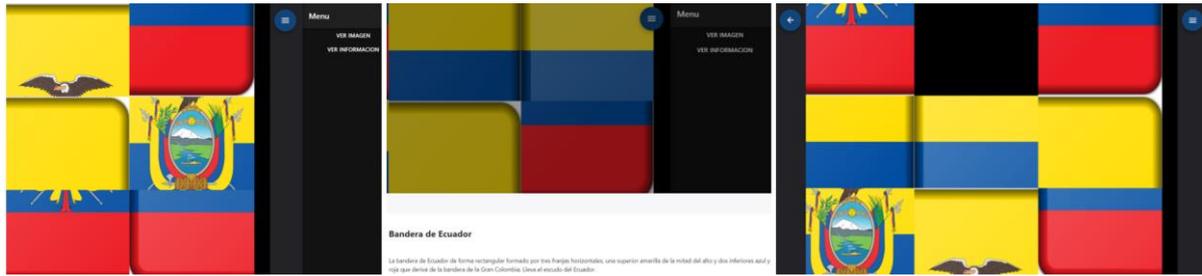


Figura 31. Capturas de pantalla del menú opción ver información
Elaborado por: La autora

2.6 Golden Master

Una vez realizadas las pruebas respectivas al videojuego, se considera que el juego en su primera versión es estable y está lista para su instalación en los dispositivos.

3 CAPÍTULO 3: EVALUACIÓN Y ANÁLISIS DE RESULTADOS

Con el modelo de calidad de experiencia del usuario basado en la jugabilidad se procede a contextualizar y adaptar las métricas a los objetivos propios del juego tomando como ejemplo la Tabla 6. Ejemplo de métricas por el modelo PQM. De esta forma las métricas quedan de la siguiente forma:

Efectividad

- Efectividad de la meta: Número de rompecabezas resueltos en un país
- Completitud de la Meta: Países resueltos (se considera país resuelto cuando se completó todas las categorías del rompecabezas de dicho país)
- Frecuencia de Intentos por Meta: rompecabezas completados sobre tiempo requerido

Eficiencia

- Tiempo de Meta: tiempo para completar un rompecabezas
- Eficiencia de Meta: eficiencia del usuario para completar un rompecabezas

Flexibilidad

- Accesibilidad: Rompecabezas resueltos con diferentes niveles de dificultad

Seguridad

- Seguridad: Número de daños o accidentes de seguridad con el dispositivo
- Daño software: Numero de caídas o problemas con la aplicación

Satisfacción

- Escala de Satisfacción: Satisfacción del jugador
- Socialización: Rompecabezas resueltos en grupo

La evaluación se realizó con 17 voluntarios de distintas edades que decidieron formar parte del proyecto, de esta forma se puede clasificar según su rango de edad:

- 5 voluntarios con un rango comprendido entre 7 y 11 años
- 7 voluntarios con un rango comprendido entre 14 y 19 años
- 5 voluntarios con un rango comprendido entre 20 y 23 años

Tomando en cuenta las pruebas a realizar según el modelo de calidad también se realiza una prueba de conocimientos antes y después de jugar. Los voluntarios tienen 5 minutos para realizar la evaluación de conocimientos antes de jugar y 10 minutos para realizar las 2 evaluaciones (conocimientos y jugabilidad) al finalizar el juego.

3.1 Análisis de resultados

Una vez realizadas todas las encuestas con el procedimiento antes mencionado a los voluntarios, los resultados se clasifican en dos partes por jugabilidad y por conocimientos obtenidos.

3.1.1 Jugabilidad

Se realizaron las encuestas con el modelo de calidad basándose en la jugabilidad a todos los voluntarios y se muestra en el Anexo 3 los datos recolectados de los mismos siguiendo la contextualización adaptada para el juego, en la Tabla 16. Resultados de las métricas del modelo basado en la jugabilidad se muestra los resultados de la aplicación usando las métricas antes mencionadas.

Tabla 16. Resultados de las métricas del modelo basado en la jugabilidad

Factor	Métrica	Resultados
Efectividad	Efectividad en la meta	59,31%
	Compleitud de la Meta	66,27%
	Frecuencia de Intentos por Meta	46,76%
Eficiencia	Tiempo de Meta (minutos)	4:31
	Eficiencia de Meta	65,88%
Flexibilidad	Accesibilidad	49,41%
Seguridad	Seguridad (sin fallas)	100%
	Daño software	94,11%
Satisfacción	Escala de Satisfacción	71,76%
	Socialización	10,78%

Elaborado por: La autora

Como se puede observar los resultados en la Tabla 16. Resultados de las métricas del modelo basado en la jugabilidad y teniendo en cuenta la interpretación de cada métrica se puede afirmar que el juego tiene una efectividad del 60%, los voluntarios de todas las categorías de edad en promedio necesitaban alrededor de 4 minutos y medio para resolver un rompecabezas con una buena eficiencia del usuario al resolver dichos rompecabezas.

En cuanto a la flexibilidad se colocó por métrica la cantidad de rompecabezas resueltos con el nivel de dificultad más alto y en base a esto se puede obtener un 49% de flexibilidad en el juego, en cuanto a la seguridad, el software no presento ningún daño al dispositivo y se reportó solo 1 problema con el software resultando así con una fiabilidad en seguridad del 97%.

El punto final es la satisfacción del usuario en el cual se pudo obtener un promedio del 72% de satisfacción con respecto al juego, según la métrica de socialización se muestra que el juego tiende a ser jugado más de forma individual que grupal.

3.1.2 Conocimientos

Se realizaron las encuestas antes mencionadas de conocimientos a todos los voluntarios, estas encuestas fueron realizadas una antes de jugar (Anexo 1) y otra después de jugar (Anexo 2). El análisis se realiza comparando las preguntas que fueron contestadas correctamente, como se puede observar en la Figura 32 ninguna de las preguntas fue contestada correctamente por todos de los 17 voluntarios, la pregunta que mayor cantidad de respuestas correctas obtuvo fue la segunda pregunta relacionada con la flora de Ecuador.

Resultados de la primera encuesta

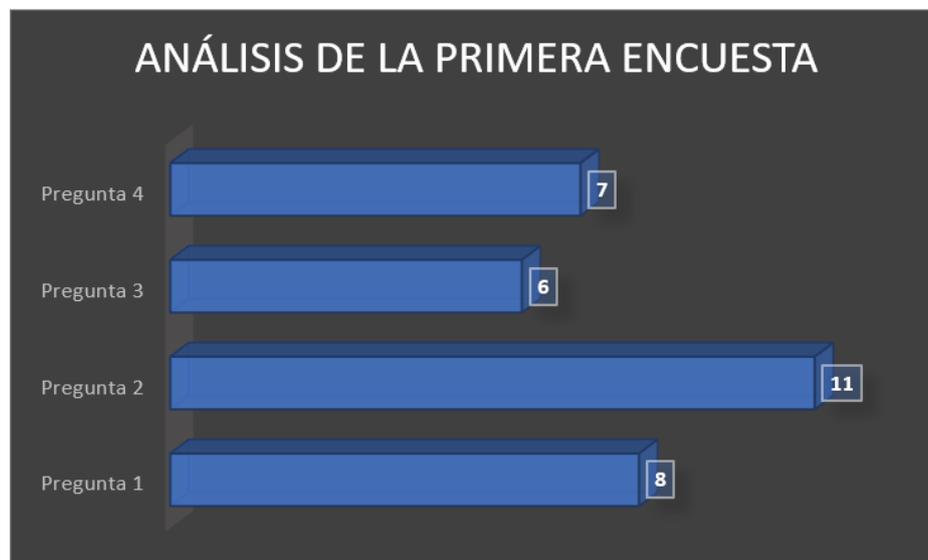


Figura 32. Resultados de la primera encuesta
Elaborado por: La autora

Como se puede observar en la Figura 33, el análisis de la segunda encuesta realizada después del periodo de prueba del juego muestra un notable cambio y mejoría ya que una pregunta fue

contestada correctamente por el 100% los voluntarios, otra pregunta fue contestada por el 94% de los voluntarios, seguido de otra pregunta contestada correctamente por el 76% de los voluntarios, la última pregunta fue contestada por el 52% de los voluntarios.

Resultados de la segunda encuesta

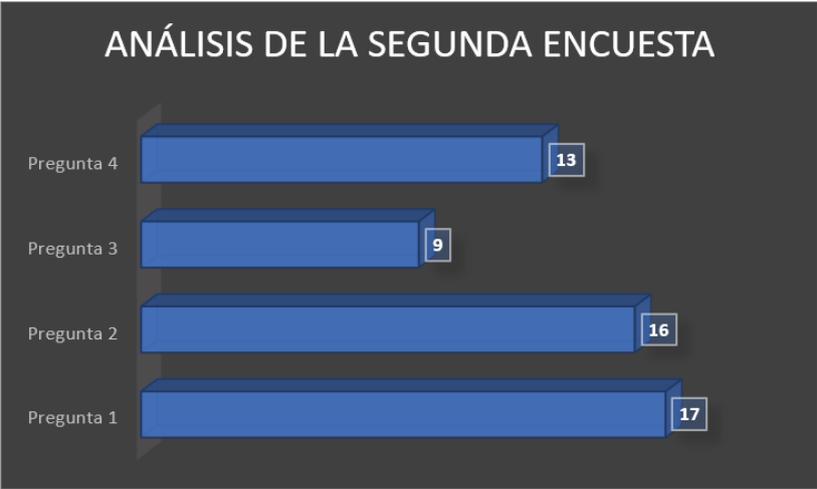


Figura 33. Resultados de la segunda encuesta
Elaborado por: La autora

3.2 Análisis de Software

Para esta sección se toma en cuenta métricas como la utilización del CPU, consumo de la memoria RAM, procesamiento gráfico, temperatura de la batería y consumo de batería. Estas mediciones son tomadas en un dispositivo smartphone Samsung con un tiempo estimado de 2 horas de uso del juego.

Rendimiento del software

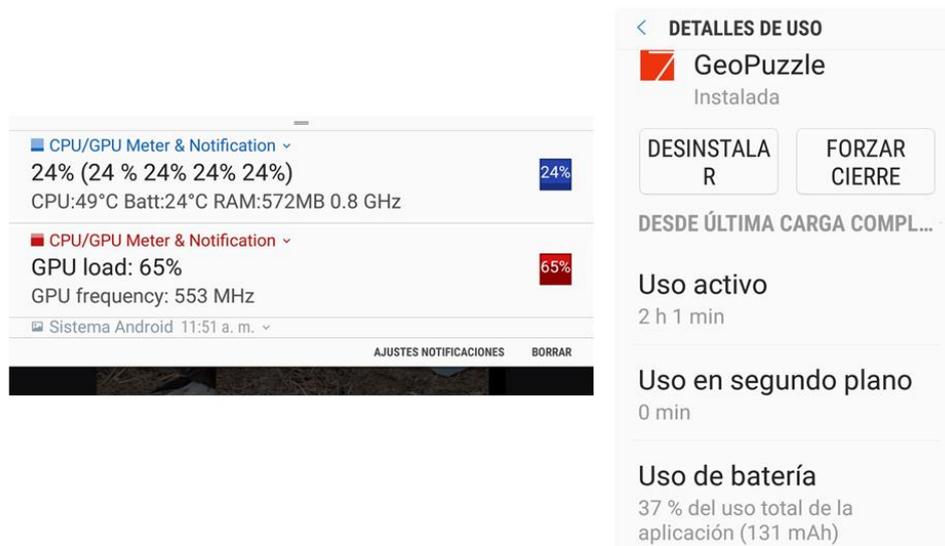


Figura 34. Rendimiento del software
Elaborado por: La autora

En la Figura 34. Rendimiento del software se puede observar que usando la aplicación CPU/GPU Meter & Notification, una herramienta de medición del rendimiento en tiempo real del dispositivo a las 2 horas de uso el porcentaje de uso del CPU fue el 24%, un consumo de 572 MB de memoria RAM, también muestra el uso del 65% de las GPU (unidad de procesamiento gráfico), adicional a esto se muestra que la temperatura de la batería es de 24 grados centígrados, es decir la batería presenta una temperatura normal y dentro de los parámetros adecuados, además se muestra el consumo de batería de un 37% (63% de batería restante) con una tasa de consumo de 131mAh a las 2 horas de uso activo de la aplicación.

CONCLUSIONES

- El desarrollo de aplicaciones usando tecnologías web como HTML, CSS y JavaScript resulta ser eficiente en la etapa de Construcción del modelo en Cascada, ya que de esta forma los frameworks usados en el desarrollo fueron únicos, es decir, no es necesario el uso de distintos frameworks con diferentes lenguajes.
- El videojuego fue creado bajo el paradigma de aplicación híbrida, obteniendo como resultado la compilación para la plataforma Android y la emulación para la plataforma iOS, sin embargo, debido a las restricciones de Apple, todos los voluntarios realizaron las pruebas en dispositivos Android.
- Los resultados obtenidos por el modelo de Jugabilidad evidencian los porcentajes de acuerdo con las métricas usadas que son: Efectividad del 60%, Flexibilidad del 49%, Seguridad del 97% y Satisfacción del usuario del 71%. Además, se puede afirmar por métrica de socialización que el juego tiende a ser jugado más de forma individual que grupal.
- Los resultados obtenidos evidencian una mejoría en los conocimientos de los distintos países luego de jugar el videojuego, demostrando así que este tipo de videojuegos pueden ser una opción factible para obtener conocimientos generales de los países así como también una herramienta pedagógica para la enseñanza de la Geografía.

RECOMENDACIONES

- El videojuego fue planteado para tener tres países disponibles por continente, teniendo en cuenta el modelo de cinco continentes. El videojuego puede ser completado con todos los países del mundo.
- Es importante mantenerse actualizado en cuanto a las categorías usadas para resolver los rompecabezas, es decir, si un país cambia su moneda oficial es importante realizar los cambios respectivos a la aplicación.
- El modelo en cascada para la creación del videojuego se debe usar si solo si se tiene todas las especificaciones, diseños, guion, historia, etc. Ya que de lo contrario el modelo puede presentar problemas al alargar el tiempo de desarrollo indefinidamente siendo más probable que el videojuego quede incompleto.

LISTA DE REFERENCIAS

Artículos Académicos

- Báez, M., Borrego, Á., Cordero, J., Cruz, L., González, M., Hernández, F., Palomero, D., de Llera, J. R., Sanz, D., Saucedo, M., & others. (2012). Introducción a android. *EME Madrid, España*, 121.
- Enriquez, J. G., & Casas, S. I. (2013). Usabilidad en aplicaciones móviles. *Informes Científicos Técnicos-UNPA*, 5(2), 25–47.
- Evaristo Chiyong, I., Navarro Fernández, R., Vega Velarde, V., & Nakano Osore, T. (2016). Uso de un videojuego educativo como herramienta para aprender historia del Perú. *RIED: Revista Iberoamericana de Educación a Distancia*, 19(2), 35–52.
- FERNÁNDEZ PÉREZ, B. (2009). PUZZLES Y JUEGOS TOPOLÓGICOS. *INNOVACIÓN Y EXPERIENCIAS EDUCATIVAS*.
- IBM, W. (2012). *El desarrollo de aplicaciones móviles nativas, Web o híbridas*.
ftp://ftp.software.ibm.com/la/documents/gb/commons/27754_IBM_WP_Native_Web_or_hybrid_2846853.pdf
- Jobe, W. (2013). Native Apps Vs. Mobile Web Apps. *International Journal of Interactive Mobile Technologies (IJIM)*, 7(4), 27. <https://doi.org/10.3991/ijim.v7i4.3226>
- M de Andrade, P. R., Albuquerque, A. B., Frota, O. F., Silveira, R. V, & da Silva, F. A. (2015). CROSS PLATFORM APP A COMPARATIVE STUDY. *International Journal of Computer Science & Information Technology (IJCSIT)*, 7(1).
<https://doi.org/10.5121/ijcsit.2015.7104>
- Montoya, C., & Flores, P. (2003). Los puzzles en alambre como recurso didáctico para la enseñanza de las matemáticas. *Gaceta de La Real Sociedad Matemática Española*, 6(3),

665–684.

Sánchez, I. B. (2011). RED EUROPEA COMENIUS DIGITAL EARTH. EU. *Didáctica Geográfica*, 12, 157–161.

Serrano, N., Hernantes, J., & Gallardo, G. (2013). Mobile Web Apps. *IEEE Software*, 30(5), 22–27.

Urrutia, G. A. M., López, C. E. N., Martínez, L. F. F., & Corral, M. A. R. (2015). Procesos de desarrollo para videojuegos. *Cultura Científica y Tecnológica*, 37.

Conferencias

Sánchez, J. L. G., & Vela, F. L. G. (2014). Jugabilidad como medida de calidad en el desarrollo de videojuegos. *CoSECivi*, 147–158.

Imágenes

Apache Software Foundation. (n.d.). *Descripción arquitectónica de la plataforma Cordova - Apache Cordova*. Retrieved June 21, 2020, from <https://cordova.apache.org/docs/en/9.x/guide/overview/index.html>

IBM, D. (n.d.). *Desarrollo de Aplicaciones Móviles – IBM Developer*. Retrieved July 2, 2020, from <https://developer.ibm.com/es/technologies/mobile/>

IDC Quarterly Mobile Phone Tracker. (2020, April 30). *Worldwide Smartphone Market Suffers Its Largest Year-Over-Year Decline in Q1 2020 Due to COVID-19, According to IDC*. <https://www.idc.com/getdoc.jsp?containerId=prUS46264320>

Jacobs, B. (2012). *Exploring the iOS SDK*. <https://code.tutsplus.com/tutorials/exploring-the->

ios-sdk--mobile-13959

Madrid, U. C. I. de. (n.d.). 2.2. *Arquitectura Android - Software de Comunicaciones*.

Retrieved June 15, 2020, from

<https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>

Scrum. (n.d.). *Home / Scrum.org*. Retrieved August 12, 2020, from <https://www.scrum.org/>

Statista. (n.d.). *The Statistics Portal for Market Data, Market Research and Market Studies*.

Retrieved June 12, 2020, from <https://www.statista.com/>

Libros

Ableson, W. F., Sen, R., Chris, K., & Ortiz, C. E. (2011). *Android in Action*. Manning Publications.

Bates, B. (2004). *Game Design* (2nd ed.). Premier Press.

<https://books.google.com.ec/books?id=f7XFJnGrb3UC>

Cuello, J., & Vittone, J. (2013). *Diseñando apps para móviles*. www.appdesignbook.com

de Miguel González, R., & y Torres, M. L. de L. (2012). *La educación geográfica digital*. AGE.

Gironés, J. T. (2011). *El gran libro de Android*. Marcombo.

Keith, C. (2010). *Agile Game Development with Scrum* (1st ed.). Addison-Wesley Professional.

Moya, F., Gonzalez, C., & Villa, D. (2019). *Creación de Videojuegos en Español: Técnicas Avanzadas*. Cursos en Español.

<https://books.google.com.ec/books?id=JmDABAAAQBAJ>

Wargo, J. M. (2015). *Apache Cordova 4 Programming* (1st ed.). Addison-Wesley Professional.

Yoffie, D. B., & Rossano, P. (2012). *Apple Inc. in 2012*. Harvard Business School.

Sitios Web

Developers, A. (n.d.). *Apple Documentation Archive*. Retrieved June 11, 2020, from <https://developer.apple.com/library/archive/navigation/#section=Platforms&topic=iOS>

Developers, G. (2020). *Android Official Documentation*. <https://developer.android.com/guide/platform>

Tesis / trabajos de titulación

Aalto, O. (2015). *MOBILE GAME PRODUCT DEVELOPMENT MODELS*. Kajaani University of Applied Sciences.

Aponte Gomez, S., & Davila Ramirez, C. (2011). *SISTEMAS OPERATIVOS MOVILES: FUNCIONALIDADES, EFECTIVIDAD Y APLICACIONES UTILES EN COLOMBIA*. UNIVERSIDAD EAN.

Chaparro García, B. (2015). *INGENERÍA DE SOFTWARE APLICADA AL DESARROLLO DE VIDEOJUEGOS* [UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA SUR]. <http://biblio.uabcs.mx/tesis/te3314.pdf>

Glera Aransay, C. (2013). *DESARROLLO DE UNA GUÍA PARA DISPOSITIVOS MÓVILES DE ESTABLECIMIENTOS PARA CELÍACOS EN LOGROÑO*. Universidad Pública de

Navarra.

Nahuel Delía, L. (2017). *Desarrollo de Aplicaciones Móviles Multiplataforma*. Universidad Nacional de La Plata.

ANEXOS

Anexo 1: Primera encuesta

Anexo 2: Segunda encuesta

Anexo 3: Datos recolectados de la encuesta de jugabilidad