



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN CARRERA DE INGENIERÍA DE SOFTWARE

TRABAJO DE UNIDAD DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN SOFTWARE

TEMA:

EVALUACIÓN Y SELECCIÓN DE FRAMEWORKS PARA EL DESARROLLO DE APLICACIONES MÓVILES SEGÚN LOS
CRITERIOS DE CALIDAD DE LA NORMA ISO: UN ENFOQUE EN LA CALIDAD DE USO, EXTERNA E INTERNA.

AUTORES:

CHUQUITARCO COPARA, ANGIE MARICELA
UNDA REINOSO, ANTHONY JOSUE

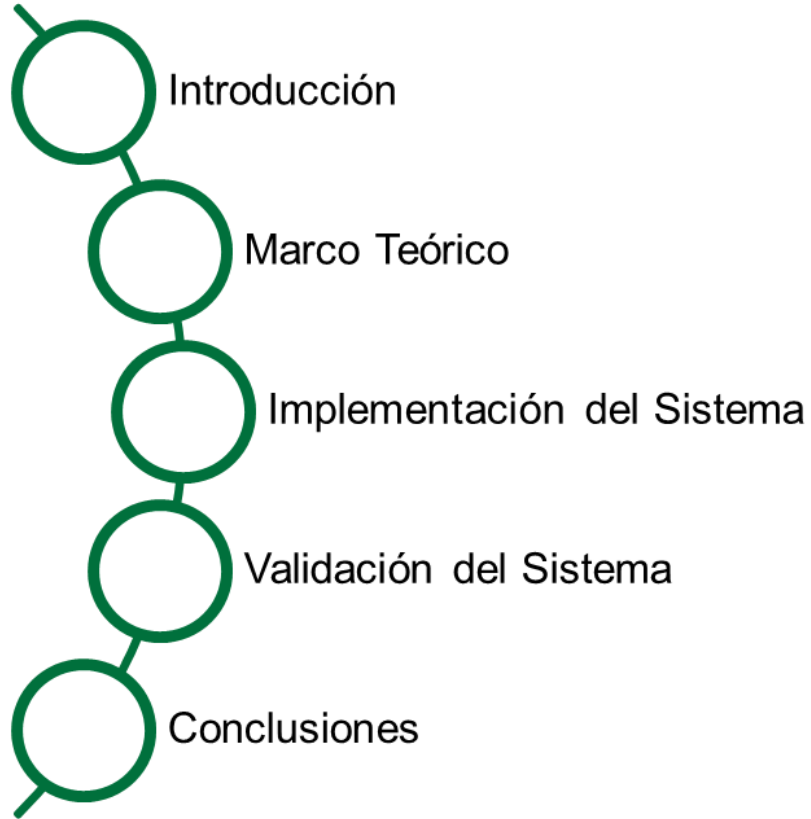
DIRECTOR:

ING. MONTALUISA YUGLA, FRANKLIN JAVIER

LATACUNGA FEBRERO, 2024



Agenda del día





Introducción

En la actualidad, las aplicaciones móviles de realidad aumentada (AR) han transformado nuestras experiencias, mejorando la eficiencia y seguridad. Este documento propone un enfoque basado en ISO 25000 para evaluar y seleccionar entornos de desarrollo de aplicaciones AR, brindando una guía objetiva para los desarrolladores y contribuyendo al avance de la realidad aumentada y la ingeniería de software.



Objetivo General

- Evaluar y comparar frameworks de desarrollo para aplicaciones móviles con realidad aumentada (RA) con el propósito de seleccionar el que mejor se adapte a las necesidades específicas de una aplicación móvil. El objetivo principal es identificar el framework que ofrezca el mejor equilibrio entre calidad interna, externa y de uso de acuerdo con los estándares de la norma ISO 25000.



Objetivos Específicos



- Asegurar que la aplicación móvil de detección de señales de tránsito cumpla con los criterios de calidad de la norma ISO 25000, garantizando precisión, confiabilidad y facilidad de uso, así como su mantenibilidad y mejora continua en el futuro.
- Establecer un marco de referencia para la evaluación de la calidad en el desarrollo de aplicaciones de realidad aumentada en seguridad vial, proporcionando una guía para proyectos futuros y contribuyendo al avance de la investigación y desarrollo tecnológico en este campo.



Importancia del estudio

Este proyecto establece una metodología para seleccionar frameworks en el desarrollo de aplicaciones móviles de realidad aumentada (AR) con enfoque en la seguridad vial. Basado en la norma ISO 25000, guía a los desarrolladores en la elección del framework adecuado, comparando características, rendimiento y facilidad de aprendizaje. Implementa un modelo de evaluación para analizar entornos de desarrollo de aplicaciones móviles AR, buscando ofrecer una guía confiable y contribuir al avance en la selección de frameworks para aplicaciones de RA.





Relevancia de la evaluación de calidad en RA

Radica en su potencial para garantizar que las aplicaciones y sistemas desarrollados cumplan con los estándares de calidad esperados, ayudando así a proporcionar una experiencia de usuario satisfactoria y eficaz.



Realidad Aumentada

“La Realidad Aumentada no es una tecnología que necesite de muchos requerimientos técnicos para ponerla en práctica, los requerimientos son mínimos” (Reinoso 2013).

Una cámara o webcam que capte la imagen del entorno.

Software de Realidad Aumentada que permita superponer contenido digital sobre la escena real.

Microprocesador con capacidad de procesamiento para modificar la señal de vídeo que se entrega a la pantalla.

Un monitor o pantalla donde visualiza la imagen real tomada por la cámara combinada en tiempo real con el contenido digital.

Según su método de operación la RA se clasifica en diferentes niveles como son:

Hiperenlaces en el mundo físico - Nivel 0.- un claro ejemplo son los códigos QR que se enlazan con sitios web.

Realidad aumentada basada en marcadores - Nivel 1.- Basados en marcadores de referencia los cuales se usan como punto de referencia o de medida.

Realidad aumentada markerless - Nivel 2.- Basado en reconocimiento de imágenes, objetos y localizaciones, en los últimos años se ha venido desarrollando aplicaciones móviles con RA para captar imágenes del entorno y superponer información sobre puntos de interés del usuario.

Visión aumentada - Nivel 3.- El propósito es visualizar información si utilizar las manos impartiendo ordenes de voz hacia gafas inteligente.



Normas ISO 25000

Tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software.



Nota. Tomado de (NORMAS ISO 25000, s. f.)



Normas ISO 25010

Modelo de Calidad ISO/IEC 25010

La ISO/IEC 25010 maneja la calidad del producto software en tres fases (Vaca & Jácome, 2018a):

- Calidad Interna: Se refiere al “desarrollo” del producto software.
- Calidad Externa: Se refiere al “funcionamiento” del producto software.
- Calidad en Uso: Se refiere al “uso” del producto software.



Descripción de los entornos de desarrollo: React Native, Flutter, Xamarin e Ionic.

Flutter

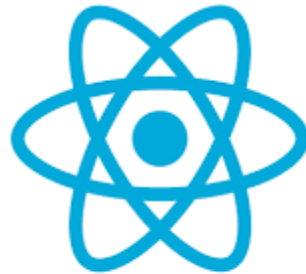
Flutter es un marco de desarrollo de código abierto creado por Google para la creación de aplicaciones móviles nativas y de alta calidad para múltiples plataformas. Flutter te permite crear apps móviles de alta calidad de forma eficiente y versátil.



Descripción de los entornos de desarrollo: React Native, Flutter, Xamarin e Ionic.

React Native

React incluye muchos bloques de código JavaScript que se pueden reutilizar para crear elementos de la interfaz de usuario (UI) llamados widgets. Esta biblioteca integra herramientas basada en JavaScript.



React Native



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Descripción de los entornos de desarrollo: React Native, Flutter, Xamarin e Ionic.

Xamarin

Reconocida por su enfoque de código abierto, destinado a la creación de aplicaciones altamente eficientes y modernas para iOS, Android y Windows mediante el uso de .NET. Esta herramienta proporciona una capa de abstracción que facilita la comunicación entre el código compartido y el código específico de cada plataforma



Descripción de los entornos de desarrollo: React Native, Flutter, Xamarin e Ionic.

Ionic

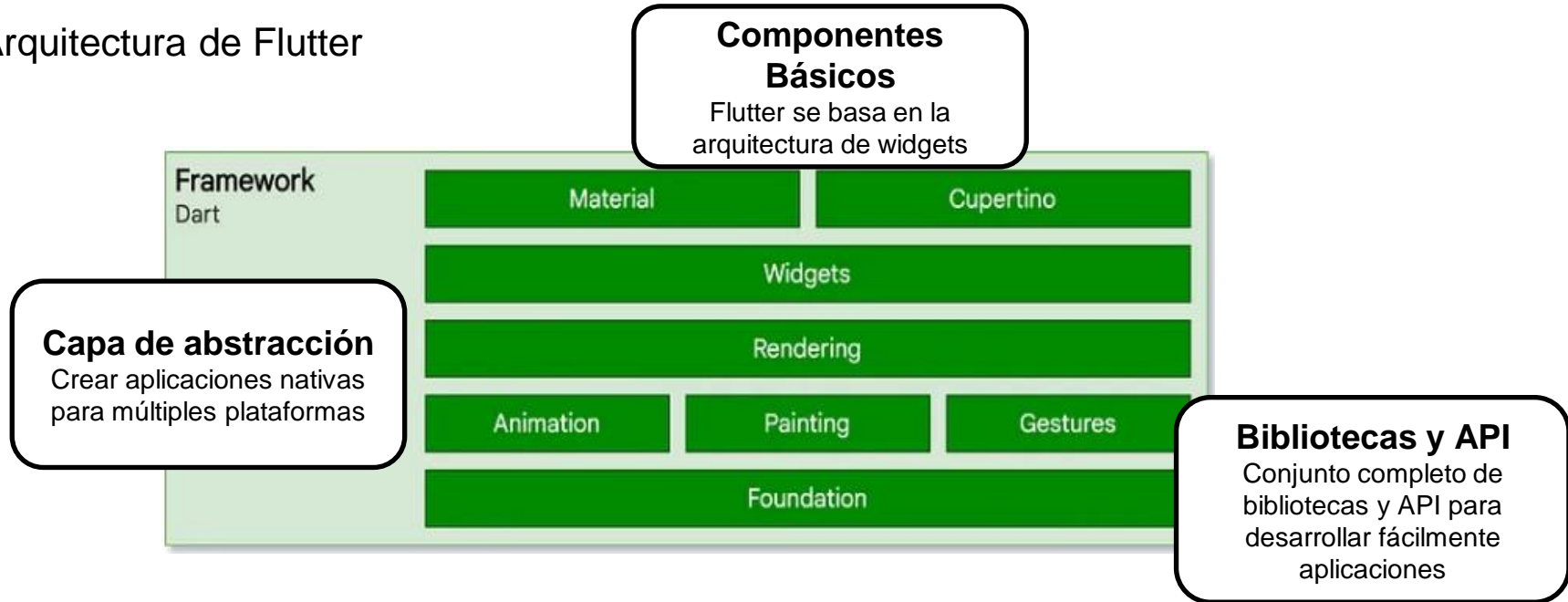
Como menciona (Yang et al., 2017) es un marco de desarrollo de código abierto basado en tecnologías HTML5, JavaScript, CSS y otras tecnologías para crear una experiencia nativa de aplicación móvil, crea aplicaciones a través de SASS “Syntactically awesome Stylesheets”





Arquitectura General en cada Framework del caso práctico

Arquitectura de Flutter

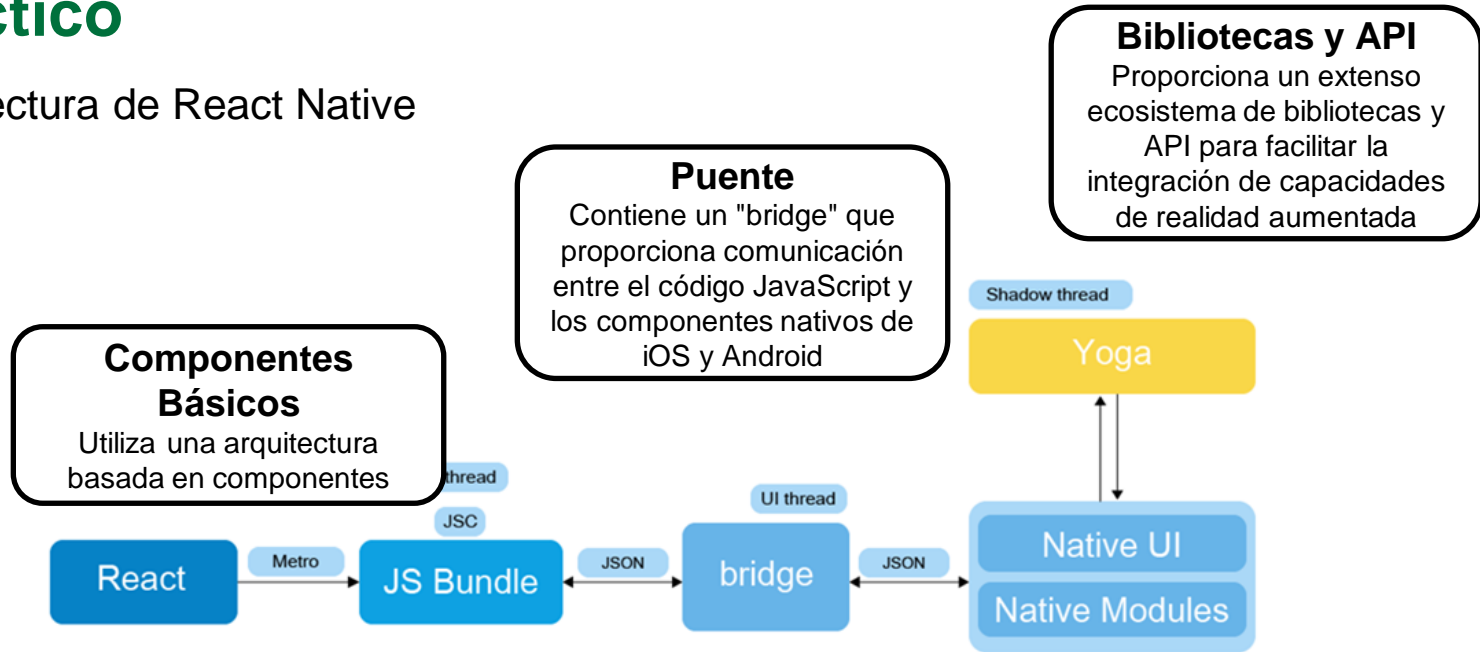


Nota. El gráfico representa la arquitectura de Flutter. Tomado de (Naik, 2022).



Arquitectura General en cada Framework del caso práctico

Arquitectura de React Native



Nota. El gráfico representa la arquitectura de React Native. Tomado de (*About React Native – Welcome – React Native Course*, s. f.)



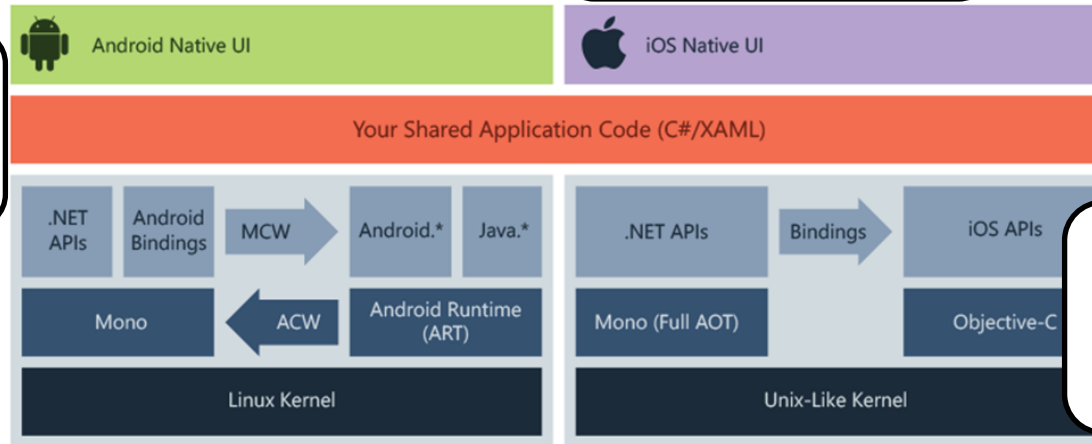
Arquitectura General en cada Framework del caso práctico

Arquitectura de Xamarin

Componentes Básicos

Xamarin sigue un enfoque de desarrollo basado en .NET

Capa de abstracción
facilita la interacción con las API nativas de iOS y Android



Bibliotecas y API
Proporciona acceso a bibliotecas y API que facilitan la integración de capacidades de realidad aumentada

Nota. El gráfico representa la arquitectura de React Native. Tomado de (Xamarin Apps with Business Central Integration – P, s. f.)



Arquitectura General en cada Framework del caso práctico

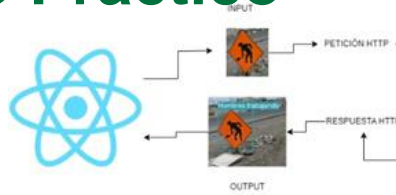
Arquitectura de Ionic



Nota. El gráfico representa la arquitectura de React Native. Tomado de (Análisis de capacidades, diferencias, roadmap y futuras posibilidades de Ionic, React Native and NativeScript, s. f.)



Funcionalidades de Reconocimiento de Señales del Caso Práctico



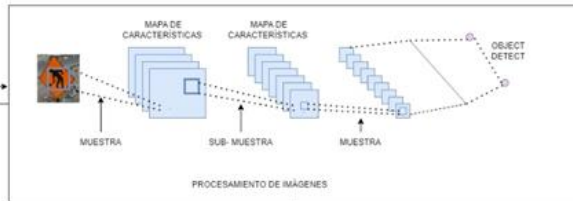
Procesamiento y reconocimiento de imágenes

El modelo utiliza técnicas avanzadas de procesamiento de imágenes y aprendizaje automático para analizar imágenes

MODELO PRE-ENTRENADO

Identificación y clasificación de signos

Esta clasificación se basa en los patrones y características específicos asociados con cada tipo de señal.



EXTRACCIÓN DE CARACTERÍSTICAS

OBJETO DETECTADO

JSON RESPONSE
Clase Name
X1,Y1
X2,Y2

Produce una respuesta en formato JSON

La respuesta contiene el nombre de la clase a la que pertenece cada señal



Correspondencia de métricas a los atributos de calidad ISO

Evaluación según el objetivo de uso

Categoría de evaluación	Descripción
Calidad interna	
Funcionalidad	¿El framework facilita agregar funciones de realidad aumentada?
Arquitectura	¿El framework es fácil de integrar con otras tecnologías?
Seguridad	¿El framework protege los datos y privacidad de usuarios?



Correspondencia de métricas a los atributos de calidad ISO

Rendimiento	¿El framework proporciona un rendimiento optimo en aplicaciones con realidad aumentada?
Calidad externa	
Usabilidad	¿El framework es fácil de usar para el desarrollo de aplicaciones con realidad aumentada?
Calidad de código	¿El código generado por el framework es claro y y esta adecuadamente documentado?
Adopción	¿El framework es ampliamente utilizado y aceptado por la comunidad y la industria?
Comunidad	¿La comunidad en base al framework es colaborativa y activa?



Correspondencia de métricas a los atributos de calidad ISO

Calidad de uso	
Escalabilidad	¿El framework puede adaptarse a diferentes niveles de demanda y tamaño de aplicación, manteniendo un rendimiento óptimo sin importar la carga de trabajo?
Apoyo	¿El framework ofrece buen soporte técnico y actualizaciones regulares?
Documentación	¿La documentación proporcionada del framework es completa y útil para los desarrolladores?
Profesionalismo	¿La organización del framework muestra un alto nivel de compromiso, integridad, mantenimiento y calidad en desarrollo?



Correspondencia de métricas a los atributos de calidad ISO

Calidad de uso	
Escalabilidad	¿El framework puede adaptarse a diferentes niveles de demanda y tamaño de aplicación, manteniendo un rendimiento óptimo sin importar la carga de trabajo?
Apoyo	¿El framework ofrece buen soporte técnico y actualizaciones regulares?
Documentación	¿La documentación proporcionada del framework es completa y útil para los desarrolladores?
Profesionalismo	¿La organización del framework muestra un alto nivel de compromiso, integridad, mantenimiento y calidad en desarrollo?



Ponderación de categorías en base a su importancia

De acuerdo con (Chandra et al., 2011), las características de calidad, que pretenden ser aplicadas a cualquier producto software deben ser específicas, puesto que no tienen la misma importancia o prioridad para cada tipo de software.

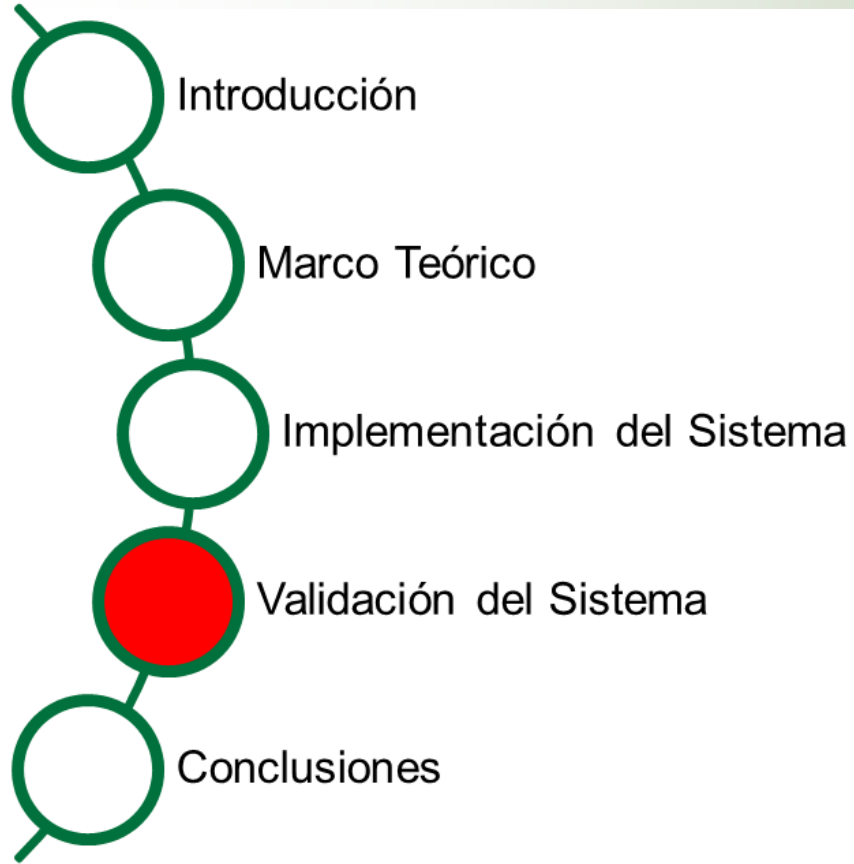
Categoría de evaluación	Clasificación
Calidad interna	
Funcionalidad	1
Arquitectura	12
Seguridad	8
Rendimiento	5



Ponderación de categorías en base a su importancia

Calidad externa	
Usabilidad	2
Calidad de código	9
Adopción	7
Comunidad	4
Calidad de uso	
Escalabilidad	11
Apoyo	6
Documentación	3
Profesionalismo	10





Pruebas Evaluadas

Adaptación de Métricas de la Norma ISO 25000 al método BBR

Como menciona (BRR, 2005), el modelo de evaluación debe incluir requisitos puntuales, que sea completo, simple, adaptable y consistente.

Evaluación rápida. – La cual consiste en identificar la lista de frameworks a ser evaluados consecuentemente medir cada componente según criterios de calidad que se proponga.

Evaluación según el objetivo de uso. – El objetivo es realizar la ponderación de categorías, asignando un porcentaje de importancia a cada métrica de las categorías seleccionadas para la evaluación,

Recolección de datos y el procesamiento. – Se debe realizar la recopilación de datos para cada métrica utilizada en la evaluación.

Traducción de datos. – Los datos deben ser interpretados en base a la evaluación realizada.



Evaluación y Resultados

Implementación del modelo BRR

Frameworks propuestos para evaluación

		IONIC	REACT	XAMARIN	FLUTTER
Facilidad de uso	G2	8.9	8.4	7.8	9.2
	Propio	6.0	8.0	6.0	10.0
Facilidad de configuración	G2	8.5	8.6	-	9.7
	Propio	4.0	8.0	6.0	10.0
Integración de desarrollo móvil	G2	-	-	-	-
	Propio	6.0	10.0	6.0	10.0
Calidad de soporte	G2	8.2	8.3	-	-
	Propio	4.0	8.0	4.0	10.0
¿Existen ejemplos de aplicaciones exitosas desarrolladas con este framework?	G2	-	-	-	8.7
	Propio	8.0	10.0	2.0	8.0
¿Qué tan personalizable (adaptable) es el framework?	G2	-	-	-	-
	Propio	6.0	10.0	2.0	10.0



Evaluación y Resultados

Implementación del modelo BRR

¿Qué tan personalizable (adaptable) es el framework?	G2	-	-	-	-
	Propio	6.0	10.0	2.0	10.0
Funcionalidad: Integración	G2	9.3	-	-	-
	Propio	4.0	10.0	4.0	10.0
Facilidad de administración	G2	8.9	8.5	-	10.0
	Propio	6.0	8.0	4.0	8.7
Cumple con los requisitos	G2	8.6	8.7	8.3	9.0
	Propio	6.0	8.0	2.0	10.0
Dirección del producto (% positivo)	G2	9.0	8.4	7.7	9.7
	Propio	6.0	8.0	4.0	10.0
Puntuación general		5.4 / 10	8.6 / 10	4.9 / 10	9.9 / 10

Nota. Esta tabla muestra la evaluación de los frameworks en base a los requerimientos propuestos



Evaluación y Resultados

Implementación del modelo BRR

Interpretación de resultados obtenidos en la evaluación

Categoría	Métricas	Fórmula por prueba	Fórmula por todas	Framework	Valor Obtenido	Interpretación del valor
CALIDAD INTERNA						
Funcionalidad	Integración con otras tecnologías	$X = \frac{A}{T\alpha}$ A =Apis integradas	N/A	React Native Xamarin	1	1 >= 1 Entre más cercano a 1 mejor



Evaluación y Resultados

Implementación del modelo BRR

Categoría	Métricas	Fórmula por prueba	Fórmula por todas	Framework	Valor Obtenido	Interpretación del valor
		$T_a = \text{Total, de Apis requeridas}$				Entre más cercano a 1 mejor (Teniendo en cuenta que no se pudo implementar de forma correcta la AR)
					$1 \geq 1$	
				Flutter	1	Entre más cercano a 1 mejor
					$1 \geq 1$	
				Ionic	1	Entre más cercano a 1 mejor
						0.7589 , React Native muestra una buena
				React Native	0.7589	cohesión y acoplamiento moderado.



Evaluación y Resultados

Implementación del modelo BRR

$$X = \frac{Cm + Am}{2}$$

Modularidad

Cm =Cohesión de módulos

Am =Acoplamiento de módulos.

$\frac{\sum X}{n}$

n =número de pruebas

Xamarin	0.8333	0.8333, lo que sugiere una alta cohesión y bajo acoplamiento (Teniendo en cuenta que no se pudo implementar de forma correcta la AR)
Flutter	0.5	0.5, Flutter muestra el valor más bajo tiene una cohesión más débil y un acoplamiento más alto
Ionic	0.7333	0.7333, Ionic se ubica en un nivel razonable de cohesión y acoplamiento



Evaluación y Resultados

Implementación del modelo BRR

Categoría	Métricas	Fórmula por prueba	Fórmula por todas	Framework	Valor Obtenido	Interpretación del valor
				React Native	0.222	0.222, sugiere que utiliza los recursos de manera eficiente
				Xamarin	0.333	0.333, todavía muestra un uso eficiente de los recursos, aunque un poco menos en relación a los otros frameworks (Teniendo en cuenta que no se pudo implementar de forma correcta la AR)
				Flutter	0.222	0.222, sugiere que utiliza los recursos de manera eficiente
Acceso a los recursos del dispositivo		$X = \frac{R}{Tr}$ R = Recursos utilizados Tr = Total, de recursos del dispositivo	N/A			



Evaluación y Resultados

Implementación del modelo BRR

Rendimiento	Tiempo de ejecución de operaciones	$X = Tt$ Tt = Tiempo de tarea	$\Sigma X / n$ n=número de pruebas	
				0.222, sugiere que utiliza los recursos de manera eficiente
				284.3329 segundos, muestra un tiempo de carga más largo
				156.9996 segundos, muestra un tiempo de carga moderado (Teniendo en cuenta que no se pudo implementar de forma correcta la AR)
				59.3673 segundos, muestra el valor más bajo tiende a cargar más rápido



Evaluación y Resultados

Implementación del modelo BRR

Categoría	Métricas	Fórmula por prueba	Fórmula por todas	Framework	Valor Obtenido	Interpretación del valor
				Ionic	255.9994 seg.	255.9994 segundos, muestra el tiempo de carga más largo entre todas
				React Native	56 %	56%, muestra el consumo muy alto entre todas
				Xamarin	18.2 %	18.2%, muestra un nivel moderado de consumo de recursos (Teniendo en cuenta que no se pudo implementar de forma correcta la AR)

$$X = A$$

A = Porcentaje de uso de CPU.

$$\frac{\sum X}{n}$$

n=número de actividades



Evaluación y Resultados

Implementación del modelo BRR

Utilización de recursos
Porcentaje de CPU y espacio de RAM

Flutter	3.3233 %	3.3233%, muestra el valor más bajo las aplicaciones tienden a consumir menos
Ionic	28 %	28%, muestra un nivel más alto de consumo de recursos
React Native	21,8 MB	21.8 MB, muestra un tamaño bastante bajo
Xamarin	2,43 GB	2.43 GB, muestra un tamaño significativamente más grande (Teniendo en cuenta que no se pudo implementar de forma correcta la AR)

$$X = \frac{\sum X}{n}$$

A = Cantidad de memoria RAM.
n=número de actividades



Evaluación y Resultados

Implementación del modelo BRR

Categoría	Métricas	Fórmula por prueba	Fórmula por todas	Framework	Valor Obtenido	Interpretación del valor
				Flutter	120.42 MB	120.42 MB, también muestra un tamaño relativamente bajo
				Ionic	177.5 MB	177.5 MB, muestra un tamaño alto
				React Native	1.1030	1.1030, muestra el valor más alto
				Xamarin	1.0166	1.0166, muestra un valor cercano a 1, lo que sugiere una actividad de desarrollo más alta en comparación (Teniendo en cuenta que no se pudo implementar de forma correcta la AR)
	Extensión de código	$X = \frac{Lc}{Tr}$ <p>Lc = Líneas de código nuevas o modificadas. Tr = Total, de</p>	$\frac{\sum X}{n}$ <p>n=número de archivos de código</p>			



Evaluación y Resultados

Implementación del modelo BRR

líneas de código
original.

Flutter	0.7855	0.7855, muestra un valor cercano a 0, lo que indica una cierta estabilidad en su código
Ionic	0.6021	0.6021, muestra el valor más cercano a 0

CALIDAD EXTERNA

Usabilidad	Experiencia de usuario final	X = conteo de respuestas	N/A	React Native	Medio	"Medio", indica una evaluación intermedia
				Xamarin	Bajo	"Bajo", esta tecnología tiene una evaluación menos favorable en comparación con las otras (Teniendo en cuenta que no se pudo



Evaluación y Resultados

Implementación del modelo BRR

Categoría	Métricas	Fórmula por prueba	Fórmula por todas	Framework	Valor Obtenido	Interpretación del valor
						implementar de forma correcta la AR)
				Flutter	Alto	"Alto", lo que sugiere que esta tecnología tiene una evaluación positiva
				Ionic	Medio	"Medio", indica una evaluación intermedia
				React Native	10 a 30	Requiere un tiempo moderado a largo para completar la actividad



Evaluación y Resultados

Implementación del modelo BRR

Tiempo de Instalación y configuración básica	$X = Tt$ Tt = Tiempo de tarea	N/A	
			Requieren un tiempo moderado para completar la actividad (Teniendo en cuenta que no se pudo implementar de forma correcta la AR)
			Requieren un tiempo moderado para completar la actividad
			Requiere significativamente más tiempo en comparación con las otras



Evaluación y Resultados

Interpretación de resultados obtenidos en la evaluación

Categoría	Métricas	Propósito	Framework	Puntaje					Interpretación del valor
				1	2	3	4	5	
				Inaceptable	Pobre	Aceptable	Muy Bueno	Excelente	
CALIDAD INTERNA									
Funcionalidad	Completitud de la implementación funcional	¿El framework proporciona todas las características y funcionalidades necesarias para desarrollar una aplicación de realidad aumentada?	React Native					X	Disponible en todos los elementos de código con una respuesta instantánea.
			Xamarin		X				Es limitado e inconsistente (Teniendo en cuenta que no se pudo implementar de forma correcta la AR)
			Flutter					X	Disponible en todos los elementos de código con una respuesta instantánea.
			Ionic		X				Es limitado e inconsistente



CALIDAD EXTERNA

Categoría	Métricas	Propósito	Framework	Puntaje					Interpretación del valor
				1 Inaceptable	2 Pobre	3 Aceptable	4 Muy Bueno	5 Exceiente	
Usabilidad	Requisitos previos de <u>instalació</u>	¿Cuán compleja es la Instalación	React Native					X	Instalación sencilla, en poco tiempo incluso para usuarios sin experiencia
			Xamarin			X			Relativamente sencillo aun con pasos y configuraciones técnicas (Teniendo en cuenta que no se pudo implementar de forma correcta la AR)



				Instalación sencilla, en poco tiempo incluso para usuarios sin experienci
			X	Flutter
				Instalación sencilla, en poco tiempo incluso para usuarios sin experienci
			X	Ionic
				Experiencia básica en el desarrollo en un tiempo aproximado de una a dos semanas.
			X	React Native
				Requiere tiempo extenso aproximadamente seis meses o más (Teniendo en cuenta que no se
			X	Xamarin
Facilidad de aprendizaje	¿Cuánto tiempo le tomo al usuario aprender a desarrollar en el framework?			



Categoría	Métricas	Propósito	Framework	Puntaje					Interpretación del valor
				1	2	3	4	5	
				Insoportable	Pobre	Aceptable	Muy Bueno	Excelente	
									pudo implementar de forma correcta la AR)
			Flutter	X					Requiere tiempo extenso aproximadamente seis meses o más.
			Ionic	X					Requiere tiempo extenso aproximadamente seis meses o más.



CALIDAD DE USO

Apoyo	Soporte	React Native		X	Responde de manera rápida y confiable
		Xamarin	X		No responde a las consultas (Teniendo en cuenta que no se pudo implementar de forma correcta la AR)
		Flutter		X	Responde de manera rápida y confiable.
		Ionic		X	Responde en un tiempo razonable, aunque existe variabilidad en el tiempo de respuesta.



Categoría	Métricas	Propósito	Framework	Puntaje					Interpretación del valor
				1 Inaceptable	2 Pobre	3 Aceptable	4 Muy Bueno	5 Excelente	
			React Native					X	Respuestas claras y útiles satisfaciendo a los usuarios de forma efectiva.
		¿Las respuestas proporcionadas por el equipo de soporte son claras, útiles y satisfacen las necesidades de los usuarios?	Xamarin	X					Respuestas confusas, incoherentes o poco claras (Teniendo en cuenta que no se pudo implementar de forma correcta la AR)
			Flutter					X	Respuestas claras y útiles satisfaciendo a los usuarios de forma efectiva.



		Ionic	X	Mayoría de respuestas útiles y claras.
Capacitación	¿Hay eventos de capacitación o meetups organizados por la comunidad de usuarios que utilizan el framework?	React		X
		Native		
		Xamarin	X	Ninguna
				Capacitaciones o meetups disponibles frecuentemente presentado por expertos.
				disponibilidad de capacitaciones o meetups (Teniendo en cuenta que no



Categoría	Métricas	Propósito	Framework	Puntaje					Interpretación del valor
				1	2	3	4	5	
				Inaceptable	Pobre	Aceptable	Muy Bueno	Exoelente	
									se pudo implementar de forma correcta la AR)
			Flutter					X	Capacitaciones o meetups disponibles frecuentemente presentado por expertos.
			Ionic	X					Ninguna disponibilidad de capacitaciones o meetups.



Documentación	Documentación de referencia	¿Existe documentación completa y fácilmente accesible para el framework?	React Native	X	Documentación exhaustiva clara y altamente accesible.
			Xamarin	X	Documentación suficiente abordando aspectos relevantes (Teniendo en cuenta que no se pudo implementar de forma correcta la AR)
			Flutter	X	Documentación exhaustiva clara y altamente accesible.
			Ionic	X	Documentación suficiente abordando



Categoría	Métricas	Propósito	Framework	Puntaje					Interpretación del valor
				1 Inaceptable	2 Pobre	3 Aceptable	4 Muy Bueno	5 Exoelente	
									aspectos relevantes.
			React Native				X		Tutoriales de referencia amplios y bien elaborados, proporcionando ejemplos prácticos.



Tutorial de referencia	¿La documentación incluye ejemplos claros y guías paso a paso para ayudar a los desarrolladores?	Xamarin	X	Los tutoriales están disponibles y cubren características principales del framework (Teniendo en cuenta que no se pudo implementar de forma correcta la AR)
		Flutter	X	Tutoriales de referencia amplios y bien elaborados, proporcionando ejemplos prácticos.
		Ionic	X	Los tutoriales están disponibles y cubren características principales del framework.





Conclusiones

Calidad interna:

React Native y Flutter se destacan por su capacidad para proporcionar una implementación integral de la funcionalidad AR y una respuesta instantánea a todos los elementos del código, como lo demuestran los datos cuantitativos recopilados. Por el contrario, la tabla cualitativa muestra limitaciones e inconsistencias entre Xamarin e Ionic en su capacidad para integrar y ejecutar de manera efectiva la funcionalidad AR.



Calidad externa:

En términos de accesibilidad y facilidad de aprendizaje, React Native, Flutter e Ionic se destacan como opciones viables ya que son relativamente fáciles de instalar y usar. Sin embargo, es importante tener en cuenta que adquirir habilidades de Xamarin puede llevar una cantidad de tiempo significativa debido a la pronunciada curva de aprendizaje que se muestra en el análisis cualitativo.



Teniendo en cuenta estos resultados, ***Flutter demuestra ser el entorno más favorable*** para el desarrollo de ***aplicaciones AR***. No solo proporciona una implementación completa de la funcionalidad AR con respuesta inmediata a todos los elementos del código, sino que también presenta una instalación sencilla, una curva de aprendizaje asequible, soporte técnico confiable y documentación completa



Calidad de uso:

En términos de soporte técnico, React Native y Flutter fueron elogiados por brindar respuestas rápidas y claras a las preguntas de los desarrolladores, lo que hizo que el proceso de desarrollo fuera más fluido



Gracias por su
atención