



PONTIFICIA UNIVERSIDAD
CATÓLICA DEL ECUADOR

SEDE
ESMERALDAS

UNIDAD ACADÉMICA
ESCUELA DE SISTEMAS Y COMPUTACIÓN

TESIS DE GRADO:
TAXONOMÍA DE TECNOLOGÍAS PARA DESARROLLO MÓVIL
MULTIPLATAFORMA

PREVIO AL GRADO ACADÉMICO DE
INGENIERÍA EN SISTEMAS

AUTOR:
JOSÉ MIGUEL LUNA INTRIAGO

ASESOR:
MSC. MARC GROB

ESMERALDAS, 2020

TRIBUNAL DE GRADUACIÓN

Título: Taxonomía de tecnologías para desarrollo móvil multiplataforma.

Autor: José Miguel Luna Intriago

Mgt. Marc Grob

f. _____

Asesor

Mgt. Gustavo Chango Sailema

f. _____

Lector #1

Mgt. Jaime Sayago

f. _____

Lector #2

Mgt. Susana Patiño

f. _____

Coordinadora de carrera

DEDICATORIA

Esta investigación se la dedico a mi familia, principalmente a mis abuelos, que siempre han estado de forma incondicional y completa para mí en todo momento. Han esperado cosas buenas de mí, las mismas que me han inculcado a través de los años.

José Luna.

AGRADECIMIENTO

Le agradezco a mi familia por el apoyo, la paciencia y la confianza durante todos mis años de educación.

Agradezco a mis profesores quienes fueron muy profesionales y me brindaron los mejores conocimientos y guías que pudieran tener.

A mis amigos, personas muy incondicionales con quienes compartí alegrías y me apoyaron para seguir adelante.

José Luna.

ÍNDICE

TRIBUNAL DE GRADUACIÓN	I
DEDICATORIA	II
AGRADECIMIENTO	III
ÍNDICE.....	IV
LISTA DE FIGURAS.....	VII
LISTA DE ILUSTRACIONES	VIII
RESUMEN / PALABRAS CLAVES.....	IX
ABSTRACT / KEYWORDS.....	X
INTRODUCCIÓN	1
Presentación del tema de investigación.....	1
Planteamiento del problema.....	1
Justificación.....	3
Objetivos	4
Objetivo general	4
Objetivos específicos.....	4
CAPÍTULO 1.....	4
MARCO DE REFERENCIA.....	4
1.1. Antecedentes	4
1.2. Bases Teórico Científicas	6

1.2.1.	Taxonomía	6
1.2.2.	Normas ISO	10
1.2.3.	Sistema operativo Android	12
1.2.4.	Sistema operativo iOS	13
1.2.5.	Framework	14
1.2.6.	Aplicaciones Nativas	15
1.2.7.	Aplicaciones basadas en la Web	15
1.2.8.	Aplicaciones Híbridas	18
CAPÍTULO 2.....		19
METODOLOGÍA.....		19
2.1.	Tipo de investigación.....	19
2.2.	Métodos y técnicas.....	20
2.3.	Descripción y validación del instrumento	21
2.4.	Técnicas de procesamiento y análisis de datos	22
2.5.	Normas éticas.....	23
CAPÍTULO 3.....		23
RESULTADOS		23
3.1.	Estructura técnica de las aplicaciones móviles	23
3.2.	Características de las Normas ISO que añadir en la taxonomía	24
3.3.	Aplicación práctica	25
CAPÍTULO 4.....		27

DISCUSIÓN	27
CAPÍTULO 5.....	29
PROPUESTA.....	29
5.1. Título.....	29
5.2. Representación.....	29
5.3. Descripción	29
CAPÍTULO 6.....	30
CONCLUSIONES Y RECOMENDACIONES	30
7. REFERENCIAS	31
7.1. Referencias bibliográficas.....	31

LISTA DE FIGURAS

Figura 1. Estructura de una taxonomía [13]	8
Figura 2. Clasificación Jerárquica Simple [14]	9
Figura 3. Clasificación Facetada o Multijerárquica [14]	9
Figura 4. Modelo de calidad del producto definido por la ISO/IEC 25010 [16].....	11
Figura 5. Interfaz de Perfetto [22]	13
Figura 6. Resultados de los test hechos por Lighthouse [28]	18
Figura 8. Registro del sistema con aplicación nativa.....	27
Figura 9. Registro del sistema con aplicación híbrida.....	27
Figura 10. Registro del sistema con aplicación PWA	27

LISTA DE ILUSTRACIONES

Ilustración 1. Relación de las características cualitativas de las tecnologías y normas ISO [36]–[38]	25
Ilustración 2. Representación gráfica de la taxonomía creada.....	29

RESUMEN / PALABRAS CLAVES

A medida que la tecnología avanza los métodos de programación también evolucionan y se trata de ser práctico a la hora de crear sistemas y aplicaciones, para esto se crearon las tecnologías multiplataforma que permiten a los desarrolladores crear aplicaciones que puedan ejecutarse en diferentes sistemas operativos, logrando así que con una sola base de código se pueda desplegar una versión de la aplicación para cada uno de estos. El problema es que el desarrollo móvil es más complejo puesto que se trata de equipos con capacidades muy limitadas y esto hace que los desarrolladores deban optimizar al máximo los procesos que realizan los dispositivos para no sobrecargarlos, por lo que se suele tener muy en cuenta la tecnología a utilizar y qué implica el uso de esta. Sin embargo, en este estudio se plantea una taxonomía que permita clasificar las tecnologías para crear una aplicación móvil desde el punto de vista de las normas ISO, centrándose así en la calidad del producto para el usuario y buscando ser agnóstico a la tecnología, sino solo tomar en cuenta sus capacidades de desarrollo. Para esto se realizó una investigación documental extensa que abarcó la documentación de las tecnologías, las métricas de uso de frameworks y lenguajes de programación y las normas ISO 25010, que trata de la calidad del software, y 25062, que trata de las pruebas para aplicaciones móviles. Esto permitió recolectar información sobre las diferencias y similitudes de las tecnologías, además de las ventajas y desventajas entre una y otra. También se realizó una aplicación con cada tecnología para contrastar los datos documentales y así refinar la taxonomía, verificando datos y esclareciendo dudas. Los resultados permitieron ver las cualidades más importantes que debe tener una aplicación móvil y cómo el uso de una tecnología u otra permite gozar de cierto número de estas, siendo las aplicaciones híbridas las más balanceadas a la hora de gestionar el software, tener un buen rendimiento y una fácil implementación de usabilidad. Todo esto nos dice que las tecnologías de desarrollo multiplataforma si son muy distintas y cada una tiene su manera de integrarse con el sistema operativo en el que se ejecute; por lo que es muy importante tener en cuenta este estudio y la taxonomía formulada para ayudarse en la toma de decisión sobre qué tecnología utilizar y sobre poner la calidad del software para el cliente ante la comodidad del equipo de trabajo al momento de desarrollar una aplicación.

Palabras clave: tecnologías multiplataforma, móvil, normas ISO, 25010, 25062, taxonomía, software, calidad del software.

ABSTRACT / KEYWORDS

As technology advances, programming methods also evolve and programmers try to be practical when creating systems and applications, for this, multiplatform technologies were created to allow developers to create applications that can run on different operating systems, achieving thus, with a single code base, create a version of the application that can be implemented for each operating system. The problem is that mobile development is more complex since it involves devices with very limited capacities and this means that developers have to optimize the processes carried out by the devices to the maximum so as not to overload them, so technology to be used and what its use implies are usually taken into account. However, this study proposes a taxonomy that allows classifying the technologies to create a mobile application from the point of view of ISO standards, thus focusing on the quality of the product for the user and seeking to be technology agnostic, just taking into account their development capabilities. To do this, an extensive documentary research was carried out that covered the documentation of technologies, the use metrics of frameworks and programming languages and the ISO 25010 standards, which deals with software quality, and 25062, which deals with tests for applications. mobiles. This made it possible to collect information on the differences and similarities of the technologies, as well as the advantages and disadvantages between them. An application was also made with each technology to contrast the documentary data and thus refine the taxonomy, verifying data and clarifying doubts. The results allowed to see the most important qualities that a mobile application should have and how the use of one or another technology allows to enjoy a certain number of these, with hybrid applications being the most balanced when managing the software, having good performance and easy usability implementation. All of this tells that multiplatform development technologies are very different and each has its own way of integrating with the operating system on which it runs; Therefore, it is very important to take into account this study and the formulated taxonomy to help you make a decision about which technology to use and to put the quality of the software for the client before the comfort of the work team when developing an application.

Keywords: multiplatform technologies, mobile, ISO standards, 25010, 25062, taxonomy, software, software quality.

INTRODUCCIÓN

Presentación del tema de investigación

El desarrollo de aplicaciones orientadas a dispositivos móviles permite que las personas tengan diversas herramientas en sus smartphones que les ayudan en su día a día, pero la creación de aplicaciones complejas o que están pensadas para realizar muchas tareas puede ser un trabajo muy pesado incluso para un equipo de programación empresarial, por lo que, con el tiempo, se han propuesto distintas alternativas que permiten hacer que los proyectos de programación traspasen las barreras de las plataformas y que así se disminuyan los costes y tiempos de programación. Sin embargo, esto ha resultado una tarea muy difícil y han existido varios intentos para poder cumplirlo; de lo cual, en la actualidad aún existen ciertas alternativas, pero por falta de interés o apoyo no han sido estandarizadas y esto hace que muchas veces sea complicado hallar las diferencias y similitudes que existen entre sí. Si bien el clasificar y distinguir cada una de estas tecnologías y métodos de programación es una tarea ardua, una forma sencilla de realizarla es a través de una taxonomía que las clasifique de forma agnóstica al cómo será programada la aplicación sino orientada en el tipo de servicio que se espera brindar al usuario, y que ayude así a decidir cuál sería la tecnología más factible para poder llevar a cabo el proceso de desarrollo de una aplicación móvil multiplataforma.

Planteamiento del problema

En los últimos años, la forma de determinar la calidad técnica de los sistemas operativos y aplicaciones móviles ha cambiado considerablemente. Las normas ISO, que ordenan la gestión de una empresa, se actualizan para mejorar la competencia internacional, de forma que los desarrolladores están en una constante búsqueda por la mejor alternativa para cambiar o mejorar sus procesos de desarrollo de software y así llegar rápidamente a su público objetivo que se divide entre usuarios de dispositivos Android y iOS, ya que éstos son los sistemas operativos de smartphones y tabletas más usados en el mundo [1]. Teniendo esto en cuenta, se puede notar la necesidad de nuevas tecnologías que apoyen el trabajo de los desarrolladores de aplicaciones los cuales optan por utilizar herramientas que mejoren el desempeño laboral y permitan desarrollar software más eficiente.

La tecnología es volátil, por lo que no implementar nuevos métodos de programación puede hacer que se pierdan importantes oportunidades de negocio, es esto lo que incrementa la competencia en la industria de las aplicaciones multiplataforma y hace relucir la importancia de las tecnologías que se usarán en el proceso de desarrollo y cómo estas influyen fuertemente al momento de elegir el sistema operativo para el cual programar y la metodología a utilizarse [2]. Esto sucede porque, si bien el desarrollo nativo de aplicaciones es muy compatible con el hardware del dispositivo, haciéndolo una muy buena opción en muchos casos, cuando se trata de equipos de trabajo pequeños, que requieren crear apps para varias plataformas, es mucho más grande el costo de producción porque se requiere de más personas y de más tiempo de trabajo comparado con una app multiplataforma para la cual se puede ocupar un solo equipo de trabajo que desarrolle para varios ambientes y enfrente los problemas de compatibilidad lo cual es menos costoso y laborioso. Entonces, estas son las razones que conducen a un creciente número de desarrolladores nativos a traspasar fronteras y crear proyectos multiplataforma que permitan realizar una implementación eficaz y con empuje de sus aplicaciones en los distintos dispositivos.

Con todo esto en cuenta es más fácil entender que las empresas que utilizaban un método de desarrollo nativo, y el escribir códigos individuales para las diferentes plataformas, van a preferir los frameworks multiplataforma ya que permiten utilizar un único código para crear aplicaciones que funcionen en varias plataformas. Sin embargo, el desarrollo multiplataforma no se limita únicamente al grupo de los sistemas móviles, sino no sería eficaz como explica Vilkomir en sus estudios [3], y las aplicaciones desarrolladas con las últimas tecnologías pueden manejarse fácilmente en un ambiente web y esto permite llegar a un número aún mayor de usuarios, haciendo del trabajo algo más efectivo y valioso.

En Ecuador, el aumento del uso de dispositivos móviles se ha vuelto una parte integral del día a día en los habitantes, lo que conduce a la expansión del desarrollo de aplicaciones a nivel nacional y el desafío de optimizarlas para las diferentes plataformas [4]. Pero las empresas y Freelancers enfrentan el reto de adaptarse a todos los estándares internacionales y al mismo tiempo buscar lo que sea más eficiente para ellos en base a los medios y capacidades que posean, esto sin perder el amplio rango de usuarios a los que se puede llegar gracias a los medios más actuales.

Por lo antes expuesto, es necesario analizar qué medios pueden ser los mejores para desarrollar sistemas multiplataforma según las necesidades, facilidad de implementación y costes que puedan caer no solo en empresas de desarrollo de software sino en cualquier tipo de entidad que cumpla con funciones o trabajos parecidos y que así la transformación digital sea lo más preciosa y efectiva posible de modo que aporte al trabajo activo de los desarrolladores de software y sus equipos.

Justificación

Una taxonomía permite la recuperación precisa de información y, gracias a su estructura jerárquica, ayuda al usuario a buscar a través de las etiquetas de las clases y subclases que tiene este tipo de clasificación de información. Teniendo esto en cuenta, las organizaciones buscan una clasificación para sus tecnologías que permita determinar cómo está su portafolio tecnológico y verificar así las brechas que deben cerrar optando por nuevas tecnologías, nuevos métodos, o distintas estrategias, y tener una mejor disposición competitiva.

La característica principal que posee una taxonomía es la claridad; mediante una clasificación que conlleve los puntos adecuados y un estudio que determine el puesto indicado de cada objeto a catalogar, derivado principalmente de la coherencia lógica, se podrá determinar cuál es la tecnología pertinente para realizar un proyecto de desarrollo multiplataforma móvil según los requerimientos de hardware que vaya a tener la aplicación o, explicado de una forma más general, de cuánto debe integrarse el software con el hardware del dispositivo.

A través de una taxonomía para los frameworks de desarrollo multiplataforma se podrá analizar, distinguir y escoger la mejor opción a utilizar para una entidad dedicada al desarrollo de software de este tipo; que busque distinguir lo más conveniente basándose en características comunes de la organización, sus recursos y capacidades de expansión gracias a que esta investigación permite una ordenación jerarquizada y sistemática encuadrando las características de las tecnologías en categorías para identificar la mejor opción a implementar para desarrollar un proyecto de desarrollo móvil multiplataforma.

Objetivos

Objetivo general

Formular una taxonomía para frameworks de desarrollo móvil multiplataforma que permita identificar la tecnología más pertinente a usar en un proyecto de desarrollo móvil basada en requisitos específicos de los estándares ISO 25010 y 25062.

Objetivos específicos

- Distinguir la estructura técnica de cada tipo de aplicación que se pueda desarrollar con una determinada tecnología.
- Identificar las categorías y subcategorías que debe tener la taxonomía en base a requisitos de las normas ISO 25010 y 25062.
- Sustentar la investigación documental con una aplicación práctica para refinar la taxonomía.

CAPÍTULO 1

MARCO DE REFERENCIA

1.1. Antecedentes

Según los estudios de mercado, hechos por MOBILESoft en 2018, actualmente los sistemas operativos móviles son dominados por el duopolio de iOS y Android. Los proyectos de desarrollo de aplicaciones intentan llegar al mayor rango de usuarios posibles lo cual crea la necesidad de orientarlas a estas dos plataformas [5].

Es ahí cuando la necesidad de usar herramientas que faciliten el arduo trabajo que significa crear aplicaciones para los distintos sistemas operativos se hace visible. Para esto están los frameworks que permiten hacer aplicaciones multiplataforma y dar soporte a este tipo de proyectos, sin embargo, se puede llegar más allá con aplicaciones móviles híbridas basadas en la web, pero esto implica el uso de otro tipo de frameworks.

Las aplicaciones basadas en la web consisten en cruzar la brecha de dispositivos móviles y de escritorio al estar construidas en estándares web como HTML5, CSS3, JavaScript.

Esto permitiría fragmentar más eficazmente la distribución de una aplicación y no limitarse solamente al ya difícil objetivo de llegar a las tiendas de dispositivos móviles [6].

Existen muchos desarrolladores expertos en la creación de proyectos web y esto hace que le puedan sacar todo el jugo a herramientas basadas en la web al ser un campo que lleva muchísimo tiempo expandiéndose, y es que, si bien se podría decir que es el mismo caso para los dispositivos móviles, los lenguajes de programación para estos han ido cambiando radicalmente sin seguir un estándar demasiado ceñido.

No obstante, los perfiles de las aplicaciones móviles híbridas y nativas son considerablemente diferentes. Usando la herramienta Hybrid-scanner [7], para rastrear y analizar los comportamientos internos de las aplicaciones móviles híbridas, se muestra que hay más rastros de APIs haciendo llamadas SS-API en una aplicación móvil híbrida que en una aplicación móvil nativa de Android. Este conocimiento es crucial para el diseño apropiado de estrategias de mitigación de vulnerabilidades o detección de programas malignos.

Todas las novedades de las aplicaciones híbridas traen nuevos retos de seguridad asociados con la integración de código diseñado para un navegador web con características nativas a un dispositivo móvil. Seth Hanson explora estas preocupaciones de seguridad y define tres formas de ataque que pueden explotar los servicios web conectados a las aplicaciones híbridas [8].

Estos estudios establecen que, a pesar de todas las ventajas como la reutilización de código, reducción de tiempo de desarrollo y apoyo en los proyectos móviles ya existentes que no dependen de los lenguajes de programación específicos de cada plataforma, aún existen muchos problemas con la implementación de las nuevas tecnologías para programación multiplataforma y todas deben ser identificadas, analizadas y depuradas.

Entonces, con tantas ventajas y desventajas, y teniendo en cuenta la convergencia tecnológica que siempre está en expansión, no es fácil aprovechar las clasificaciones tecnológicas existentes ya que las tecnologías emergentes pueden ser fácilmente ignoradas porque las taxonomías actuales no proporcionan la explicación adecuada sobre los cambios tecnológicos y su evolución. Así, y como dice Lee [9], es un buen momento

para volver a conceptualizar y desarrollar una nueva taxonomía tecnológica desde la perspectiva sistemática en tecnología.

Finalmente, y con todo lo expuesto, es notable que aún se necesita de una taxonomía que permita identificar los recursos necesarios, las vulnerabilidades a perfeccionar y que tenga un buen tiempo útil en el mercado para clasificar, identificar y seleccionar la herramienta de desarrollo de aplicaciones multiplataforma que vaya mejor con cada proyecto de programación.

1.2. Bases Teórico Científicas

Las bases teóricas son los temas principales con los que tiene relación este estudio; como es la propia taxonomía, sus partes y tipos, los tipos de aplicaciones móviles y los frameworks de desarrollo, entre otros.

1.2.1. Taxonomía

La palabra taxonomía se origina de la palabra griega taxis, cuyo significado es arreglo u orden, y nomos, que significa ley o ciencia. Hablando de la gestión de la información, el término taxonomía se utiliza tanto en un contexto específico, para referirse a una clasificación jerárquica o sistema de categorización, como en un sentido muy amplio, refiriéndose a cualquier medio de organización de conceptos de conocimiento [10].

La taxonomía radica en un sistema de categorías que se ajusta a características específicas relacionadas, a fin de poder describir la concordancia entre categorías. Usualmente hay una interdependencia entre las categorías de tal manera que en la clasificación una categoría llama al tratamiento conceptual de otra categoría [11].

Entonces, se entiende por taxonomía una estructura de información ordenada constituida por categorías y subcategorías que están relacionadas y permiten así que se puedan unir objetos o entidades que comparten una o varias características en común; así se crea la base para la organización del contenido y provee de un esquema explorable del contenido, etiquetando información para su almacenamiento, búsqueda y recuperación precisa.

Es así como varias categorías están conectadas entre sí y forman parte de una entidad superior o son la entidad del que dependen otros términos. De igual forma, los objetos o

entidades de nivel superior forman parte de un término de máximo nivel, el cual no siempre es explícito. Es así como se puede decir que lo substancial de una taxonomía es predecir; evitar indagaciones secuenciales o reducir las acciones obligatorias para clasificar, ordenar o encontrar un objeto dentro de una clasificación.

Al desarrollar una taxonomía es importante tener en cuenta el separar apropiadamente los elementos de un grupo en subgrupos que son mutuamente exclusivos, no ambiguos y que, en su conjunto, incluyen todas las posibilidades [12].

Para que una taxonomía sea aplicable en el mundo real debe ser sencilla y fácil de entender y usar. No es posible presentar una taxonomía con un esquema de clasificación absoluto, pero puede ser un punto de partida para examinar componentes centrales de las dimensiones y categorías de los frameworks multiplataforma, que podrían conducir a una taxonomía ampliamente aceptada.

1.2.1.1. Componentes y estructura

Una taxonomía está compuesta principalmente por términos representados como categorías y subcategorías; los cuales se consideran términos que se relacionan y aparecen como parte de la estructura de navegación dentro de la clasificación. Sin embargo, los términos no relacionados son llamados índices complementarios [13].

Las taxonomías tienen estructura de árbol, están constituidas por una jerarquía de clases y subclases que vienen a ser las categorías y subcategorías que se quieren clasificar. Las categorías de mismo nivel coexisten con una relación de hermanos y las categorías y subcategorías poseen una relación padre e hijo (ver Figura 1).

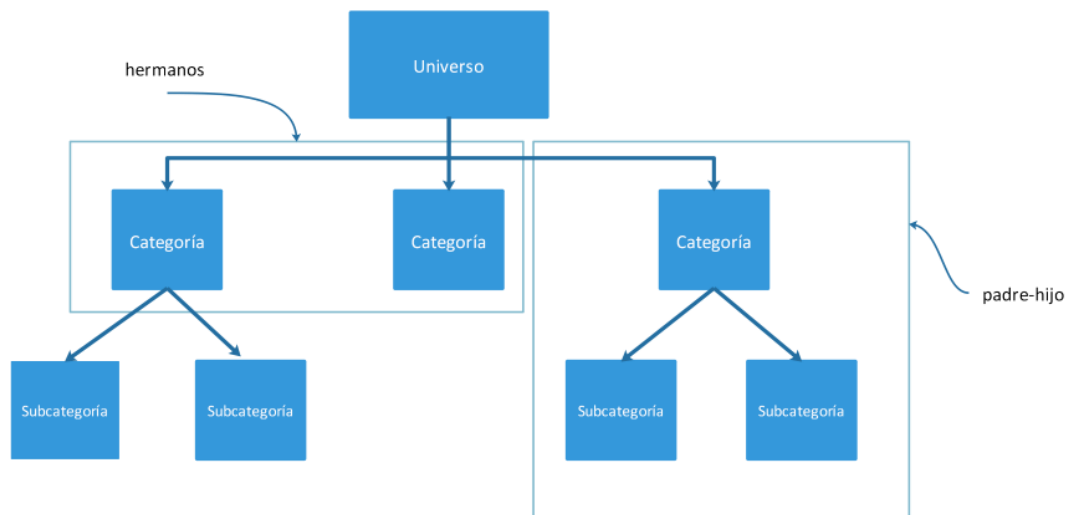


Figura 1. Estructura de una taxonomía [13]

La cantidad de niveles de una taxonomía no está predeterminada, es decir, una subcategoría puede tener a su vez más subcategorías y relacionarse entre sí como padre e hijo o tener categorías hermanas. La relación padre e hijo o categoría y subcategoría se suele justificar en torno a una relación de ‘partes de un todo’, como en tarjetas gráficas a un ordenador, y la relación entre clases podría verse como la que existe entre familiares del mismo grado, al igual que en las aves existe relación entre los loros y canarios [13].

1.2.1.2. Taxonomía facetada

Las taxonomías pueden ser unitarias o facetadas, esto último refiriéndose a que son múltiples. Por lo que, una taxonomía facetada es un conjunto de clasificaciones aplicadas a un mismo ente y que consiste de un número de clasificaciones diferentes, donde cada una de ellas es una faceta del tipo de términos u objetos que son categorizados [13].

Con esto en mente, la taxonomía de tecnologías de desarrollo móvil multiplataforma que se plantea se podría desarrollar como unitaria, pero esto obligaría a repetir determinadas subcategorías como arquitectura, lenguaje de programación, tiempo de compilación, entre otros. Por ejemplo, si existen tres categorías principales como Framework de aplicaciones nativas, Framework de aplicaciones basadas en web, Framework de aplicaciones híbridas, habría que repetir las subcategorías antes mencionadas en alguna, o en todas, de estas tres categorías.

En lugar de esto es más conveniente dividir la taxonomía en diversas facetas como las antes nombradas y de este modo cada framework queda categorizado por la intersección de un cierto número de facetas, o características, de la taxonomía que se plantea. De este modo, cada framework estará caracterizado por arquitectura, lenguaje de programación, y, en este caso, por las normas ISO 25010 y 25062.

La mejor forma de entender qué es una taxonomía facetada es mediante su comparación con sistemas de clasificación comunes, como es el sistema jerárquico simple [14].

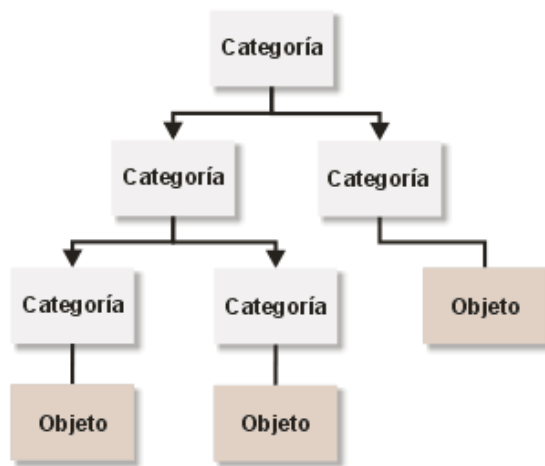


Figura 2. Clasificación Jerárquica Simple [14]

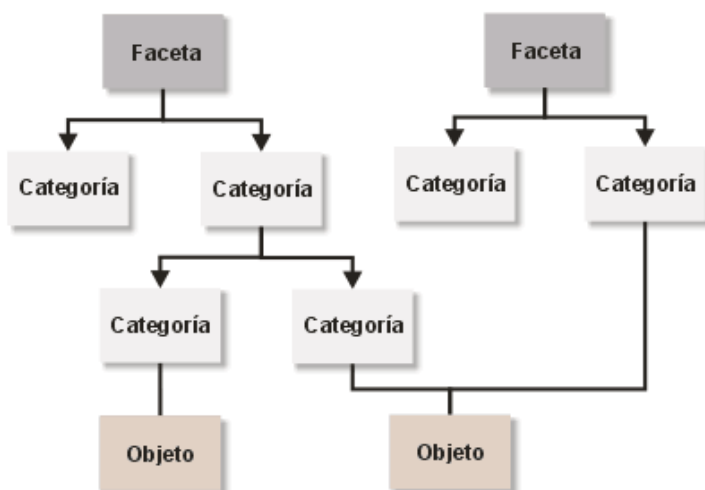


Figura 3. Clasificación Facetada o Multijerárquica [14]

1.2.1.3. Facetas, Categorías y Objetos

Las facetas contienen las categorías de forma excluyente, es decir que una misma categoría no puede existir en dos facetas diferentes. El hecho de utilizar facetas es no repetir las categorías dentro de una taxonomía, caracterizando un objeto o término desde diferentes puntos de vista para así poder proveer una descripción general y, a la vez, multidimensional de la entidad.

Una categoría puede albergar un variado número de objetos. Los cuales pueden, o suelen, estar contenidos en distintas categorías de diferentes facetas. Por tanto, un objeto es sintetizado por su ‘pertenencia’, ‘ocurrencia’ o ‘aparición’ en las diferentes categorías.

1.2.2. Normas ISO

“Las normas ISO son un conjunto de normas orientadas a ordenar la gestión de una empresa en sus distintos ámbitos. La alta competencia internacional acentuada por los procesos globalizadores de la economía y el mercado y el poder e importancia que ha ido tomando la figura y la opinión de los consumidores, ha propiciado que dichas normas, pese a su carácter voluntario, hayan ido ganando un gran reconocimiento y aceptación” [15].

Casi todos los sellos de ISO van seguidos de un número, el cual indica el tipo de producto o servicio cuya calidad certifica. Las normas ISO sirven para mejorar los resultados de una empresa, posicionarse mejor en el mercado, mejorar el servicio al cliente y demostrar innovación y liderazgo.

1.2.2.1. Norma ISO 25010

La norma ISO 25010 ofrece un modelo de calidad que es fundamental al momento de establecer un sistema de evaluación de calidad de productos. Este modelo determina las características de calidad que serán utilizadas para evaluar un producto de software específico.

“La calidad del producto de software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor. Son precisamente estos requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y sus subcategorías” [16].

“El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad que se muestran en la siguiente figura:” [16]



Figura 4. Modelo de calidad del producto definido por la ISO/IEC 25010 [16]

En este estudio se utilizará los requisitos de eficiencia de desempeño de la ISO25010 como parámetros dentro de la taxonomía que se creará, esta característica se enfoca en el desempeño del software relativo a la cantidad de recursos utilizados en condiciones específicas. Para hacer estas mediciones se tiene en cuenta 3 subcategorías:

- Comportamiento temporal. - Es la medición de los tiempos de respuesta y procesamiento del software al realizar tareas, y la comparación de estos datos con un banco de pruebas establecido.
- Utilización de recursos. - Es las cantidad y tipos de recursos del dispositivo utilizados cuando el software lleva a cabo sus tareas en condiciones específicas.
- Capacidad. - Nivel en que los límites de un parámetro en el software llenan o complacen los requisitos necesarios.

1.2.2.2. Norma ISO 25062

La norma ISO 25062 provee de un método estándar para presentar los resultados de pruebas de usabilidad. El formato está diseñado para presentar los resultados de las pruebas formales de usabilidad en las que se recolectan evaluaciones cuantitativas apropiadas para pruebas sumativas o comparativas [17]. Si bien no indica el cómo realizar una prueba de usabilidad, busca orientar sobre cómo procesar los resultados de una prueba de usabilidad. El análisis de los datos de usabilidad permite conocer las limitaciones de hardware que puede encarar un software, dicho estudio puede ser usado tanto por profesionales de usabilidad y partes interesadas de la organización para ayudar a tomar

decisiones informadas sobre el lanzamiento de productos de software o la adquisición de dichos productos.

La limitación móvil que se analiza en este estudio, y forma parte de la taxonomía, es la de característica de red que verifica cuánto puede afectar el limitado y variable ancho de banda que pueda tener el usuario final al momento de utilizar la aplicación móvil, afectando el tiempo de descarga de datos, la calidad de transmisión de audio y video [18].

Además, basándose en las referencias con la norma ISO 9126 se toma en cuenta la usabilidad que puede llegar a tener el software dependiendo del camino que se siga al utilizar la taxonomía que se desea crear. La usabilidad es influenciada por cuan limitada pueda ser la interfaz de usuario [19]. Es la capacidad del software para ser entendido, aprendido, utilizado y de ser atractivo para el usuario cuando es usado bajo situaciones predeterminadas.

1.2.3. Sistema operativo Android

Android es un sistema operativo móvil desarrollado por Google y que está basado en el kernel de Linux. La mayor parte de las aplicaciones están desarrolladas en Java; el bytecode Java no es ejecutado, primero se compila a un ejecutable Dalvik y luego este se ejecuta en la Máquina Virtual Dalvik que está diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con procesador, memoria, y batería limitados.

Los desarrolladores tienen acceso completo a las mismas API del entorno de trabajo utilizados por las aplicaciones propias del sistema operativo Android. La arquitectura está diseñada para facilitar el reciclaje de componentes; cualquier aplicación puede informar de sus capacidades y cualquier otra aplicación puede luego hacer uso de estos objetos y su información [20].

1.2.3.1. Registro del sistema y Perfetto

El proceso de registrar la actividad del dispositivo durante un periodo corto se conoce como registro del sistema. Este registro produce un archivo que se puede usar para generar un informe del sistema que ayuda a identificar la mejor manera de optimizar el rendimiento de una aplicación [21]. El registro del sistema se puede activar desde la

configuración del dispositivo en las opciones de desarrollador, una vez activado se empezará a guardar la información del dispositivo en un archivo perfetto-trace que puede ser leído desde la herramienta web de perfetto.dev [22] y así inspeccionar el uso de la CPU y la actividad de subprocesos en tiempo real mientras el usuario interactúa con las aplicaciones del dispositivo. Perfetto es una herramienta de registro para la plataforma de Android 10, y posteriores. Este es un proyecto de código abierto para el registro de datos, general y sofisticado para Android, Linux y Chrome.

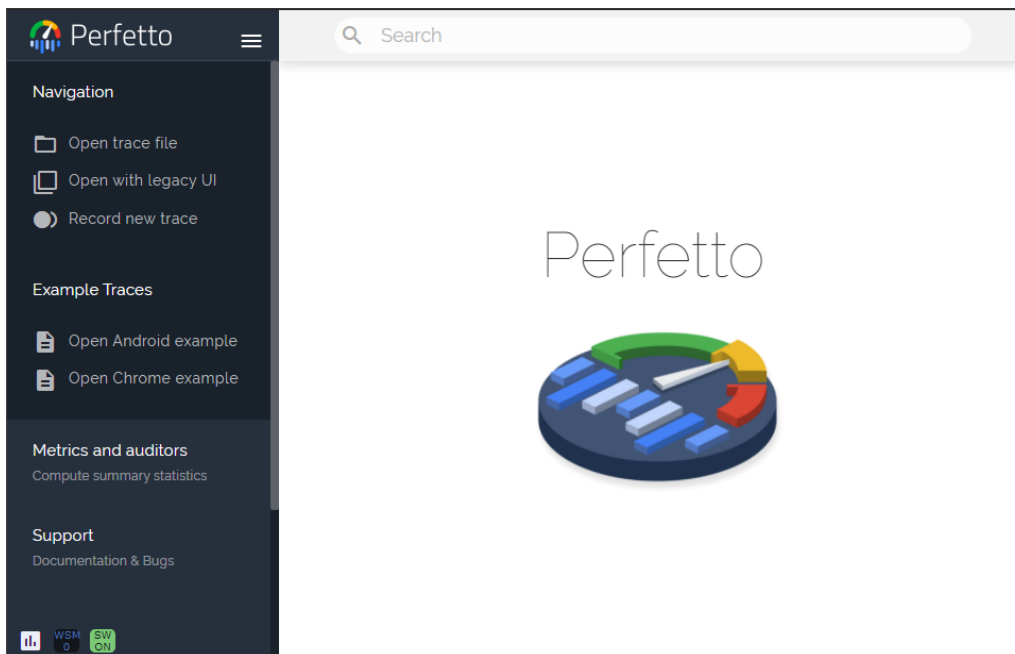


Figura 5. Interfaz de Perfetto [22]

1.2.4. Sistema operativo iOS

iOS es un sistema operativo propietario de la empresa Apple Inc. Que originalmente fue desarrollado para ser utilizado en el iPhone y que después se implementó en dispositivos como el iPod Touch y el iPad.

Este sistema operativo se deriva de macOS, el cual está basado en Darwin BSD, lo cual hace de este un sistema de Tipo Unix. iOS cuenta con cuatro capas de abstracción; el núcleo del sistema operativo, los servicios principales, middleware y la interacción táctil.

“En marzo del 2008 Apple liberó un kit de desarrollo de software para terceros y desarrolladores, permitiendo así crear aplicaciones para iPhone y iPod Touch, así como probarlas en el ‘iPhone simulator’ ” [23].

1.2.5. Framework

Un Framework de desarrollo de software es una herramienta tecnológica que brinda un apoyo y una estructura en la creación de diferente tipo de aplicaciones sirviendo de base para la organización y desarrollo, usualmente a través de artefactos o módulos de software concretos, además de incluir soporte de programas, bibliotecas y en ocasiones un lenguaje interpretado, que ayudan a desarrollar y unir las diferentes partes de un proyecto [24].

El uso de un framework permite desarrollar una aplicación de forma rápida y eficaz, haciendo el código más limpio y seguro, y es de esa forma que permite realizar procedimientos avanzados de producción y testeo a través de buenas prácticas y regidos por normas ISO, sin caer en el código repetitivo.

1.2.5.1. Framework Multiplataforma

Una forma de crear aplicaciones para diferentes plataformas es creando versiones customizadas de una sola aplicación para cada plataforma, por ejemplo, una versión para iPhone, otra para Android. Sin embargo, este camino deja múltiples versiones de código base para una sola aplicación y esto a través del tiempo dificulta la mantención y evolución de un proyecto. Es así como los desarrolladores adoptan los frameworks para desarrollo de aplicaciones multiplataforma. Estos frameworks permiten crear la lógica de una aplicación una sola vez y en un lenguaje de alto nivel, además de proveer de herramientas de soporte que permiten a la aplicación ejecutarse en un cierto número de plataformas.

Existen dos clases de frameworks multiplataforma. La primera clase son los llamados frameworks basados en la web, estos permiten a los desarrolladores construir apps utilizando lenguajes popularmente utilizados para construir aplicaciones web, como HTML5, JavaScript y CSS [25]. Sin embargo, estos lenguajes no contienen privilegios para permitir a las aplicaciones acceder a los recursos de los dispositivos, sus periféricos u otras aplicaciones y configuraciones. Estas interfaces permiten visualizar, descargar y manejar contenido dentro de la misma aplicación sin una integración al hardware del dispositivo.

La segunda clase, llamada frameworks nativos, aborda los desafíos anteriores dando soporte a la plataforma nativa y a una, o más, plataformas objetivo. Los desarrolladores crean aplicaciones para una plataforma base o *home* y el framework provee de soporte automático para producir apps que funcionen en los demás dispositivos o plataformas objetivo [25]. Al utilizar un framework nativo, el desarrollador espera de forma implícita que la aplicación se comporte de la forma esperada a través de las distintas plataformas. La realización de esta expectativa depende de cuan extendida es la traducción de las APIs del SDK nativo a los SDK de las otras plataformas objetivo. Desafortunadamente la traducción es una tarea compleja porque las distintas plataformas necesitan de un encodeo semántico específico y las fallas que pueden existir pueden dar paso a bugs.

1.2.6. Aplicaciones Nativas

“Las aplicaciones nativas se denominan así porque se desarrollan en el lenguaje nativo del propio dispositivo. Dependiendo de la plataforma para la que se quiera desarrollar la aplicación se programará con un lenguaje específico para la misma. Es así como para desarrollar en Android se utiliza Java o Kotlin, y en iOS Objective C o Swift” [26].

Estas aplicaciones utilizan eficazmente los recursos de cada dispositivo ya que poseen la capacidad de acceder a diferentes características propias del dispositivo como la cámara, el GPS, entre otras. Al estar diseñadas específicamente para el sistema operativo del dispositivo, generalmente, están optimizadas y tienen una interfaz gráfica muy adaptada al sistema operativo al que un usuario estaría acostumbrado.

Sin embargo, estas aplicaciones requieren de un desarrollo más complejo, más tiempo de desarrollo y diferentes lenguajes de programación por lo que su coste es alto. Esta es la principal desventaja en este método de desarrollo, puesto que el código utilizado para un sistema operativo no puede ser reutilizado y se debe empezar un nuevo desarrollo para llevar a cabo la implementación en otra plataforma.

1.2.7. Aplicaciones basadas en la Web

La aplicación basada en la web son desarrollos hechos con JavaScript, CSS y HTML. Este tipo de aplicaciones tienen un solo código que es interpretado y ajustado a los dispositivos. La mayor ventaja de este tipo de aplicaciones que es muy amigable para los desarrolladores web por lo que no es necesario tener un equipo de trabajo con desarrolladores especializados en Android o iOS para realizar este tipo de proyectos, sino

que se pueden reutilizar a los programadores web que ya conocen las herramientas necesarias para estos proyectos y sobre todo lo harán sin ningún tipo de capacitación extra.

Por otro lado, el inconveniente principal con estas aplicaciones es que cuando se intenta utilizar los recursos propios de un dispositivo es necesario utilizar plugin extras y ajenos a las herramientas base, esto conlleva a un mayor uso de recursos del hardware y que la velocidad de la aplicación disminuya considerablemente. Además, hay que recordar que, como todo desarrollo web, para que la aplicación pueda funcionar va a ser necesario el acceso de internet para comunicarse al servidor de la aplicación y descargar las plantillas y funcionalidades.

1.2.7.1. Aplicaciones Web Progresivas

Las aplicaciones web progresivas son aplicaciones web que se han diseñado para que sean capaces, confiables e instalables. Estos tres pilares se logran porque implementan APIs modernas lo cual las transforman y permiten ofrecer una experiencia que se siente como una aplicación nativa.

Las capacidades que puede tener la web son cada vez más semejables a las capacidades de las apps nativas [27]. Si bien algunas capacidades aún están fuera del alcance de la web las APIs buscan cambiar eso expandiendo lo que la web puede hacer con características como acceso al sistema de archivos, controles de medios, credenciales, entre otros.

Las aplicaciones web progresivas son confiables en el sentido de que pueden ser utilizadas independientemente de la conexión de red. Se configuran de tal manera que se inicien en conexiones de red lentas o inestables o incluso sin conexión. Se busca que el contenido más reciente con el que se ha interactuado esté disponible y se pueda usar incluso si es difícil obtener una solicitud a su servidor.

Este tipo de aplicaciones se pueden instalar como un acceso directo en el dispositivo y se ejecutan en una ventana independiente en lugar de una pestaña del navegador. Es posible buscarlas en un dispositivo y saltar entre ellas con el conmutador de aplicaciones dando la sensación de ser parte del dispositivo. Además, pueden registrarse para aceptar

contenido de otras aplicaciones o ser la aplicación predeterminada para manejar diferentes tipos de archivos, entre otros.

1.2.7.2. Google Lighthouse y PWA

Lighthouse es una herramienta de código abierto que es instalable en el navegador web Google Chrome o que se puede ejecutar en el computador a través de líneas de comandos al instalar el paquete de node. El propósito de esta herramienta es ejecutar una serie de pruebas contra una página web y luego generar un informe de su rendimiento, consiguiendo así métricas que permitan encontrar los puntos débiles de una aplicación web.

“Lighthouse actualmente tiene un gran enfoque sobre las funciones de las Progressive Web Apps, como ‘Add to home screen’ y soporte sin conexión. Sin embargo, el objetivo general del proyecto es ofrecer una auditoría de extremo a extremo de todos los aspectos de la calidad de la app web. [28]”

Una vez realizadas las pruebas se lanza una ventana de navegador de Chrome con el resultado del mismo en que se puede observar datos relacionados con el rendimiento, la accesibilidad, las buenas prácticas, el SEO de la web y en qué porcentaje cumple con las premisas para ser una PWA.

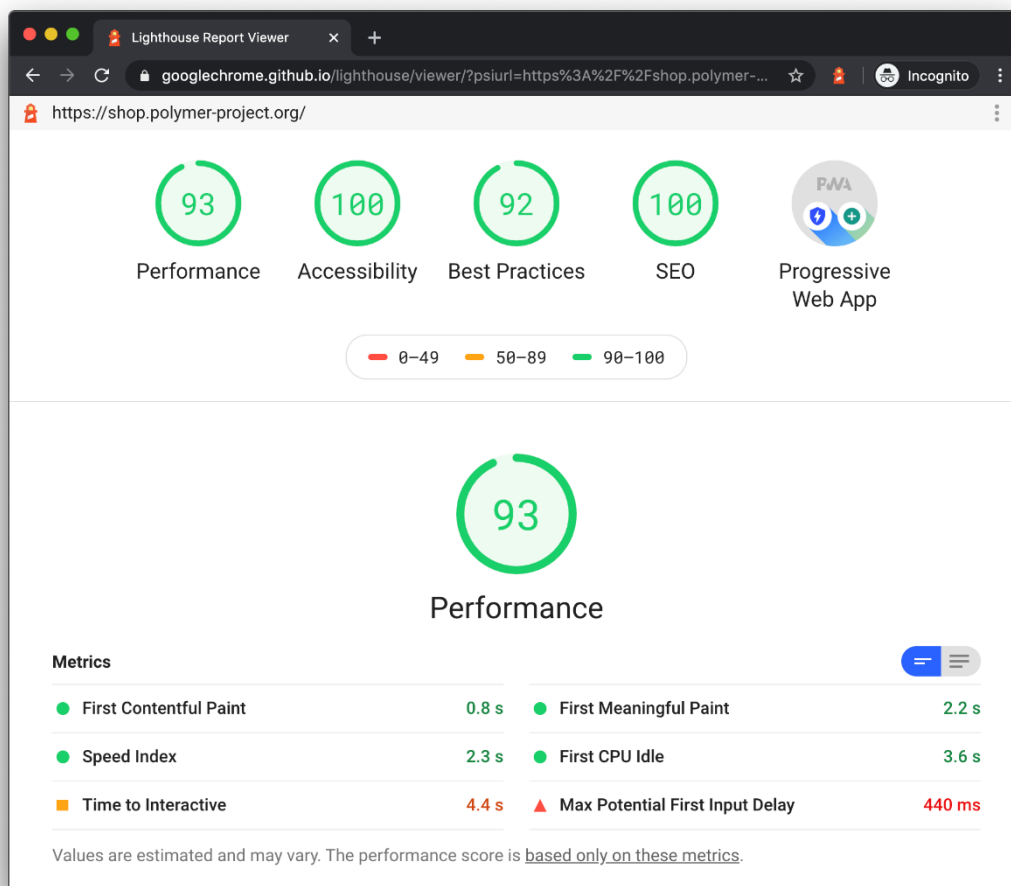


Figura 6. Resultados de los test hechos por Lighthouse [28]

1.2.8. Aplicaciones Híbridas

Las aplicaciones híbridas son la unión entre las aplicaciones nativas y las aplicaciones basadas en la web, y son desarrolladas a través de un framework que le da un valor agregado y mayor eficacia a la hora de lanzar la aplicación a las distintas plataformas objetivo. De igual manera, estas aplicaciones se crean utilizando HTML, CSS y JavaScript, sin embargo, es el mismo framework el que se encarga de hacer la compilación de los paquetes de instalación, apk para Android e ipa para iOS. El resultado es un proyecto web que puede ser ejecutado en distintas plataformas, con la ventaja de que no necesita de internet para funcionar.

Usualmente estos frameworks compilan la aplicación con un servidor local que permite el alza de las interfaces y funciones que se conectan a un servidor en la nube del cual

consumen las APIs que sean necesarias para realizar los procesos para los que fue creada la aplicación. Además, para que las aplicaciones puedan ser multiplataforma se debe poder generar un apk y una ipa para instalar el software compilado en Android e iOS respectivamente, por lo que sería necesario utilizar los SDK de estas plataforma o servicios alternativos que los generen automáticamente, como Adobe Build [29].

Aunque la usabilidad es igual a la de una aplicación nativa, el rendimiento de una aplicación híbrida es menor, ya que la primera aprovecha de forma más óptima los recursos de hardware del dispositivo móvil como los sensores integrados, entre otros. Aun así, las aplicaciones híbridas pueden utilizar estos recursos de hardware, aunque no al mismo nivel de las nativas [30] puesto que si tiene una integración a nivel nativo pero se da luego de una traducción de código o implementación de una API para comunicar la aplicación móvil híbrida con el sistema operativo como si fuera una aplicación nativa.

CAPÍTULO 2

METODOLOGÍA

2.1. Tipo de investigación

El presente estudio tiene un enfoque cualitativo, busca principalmente la expansión de los datos sobre las tecnologías para el desarrollo multiplataforma a través de la recolección de características y la distinción de categorías que conllevaron la creación de una taxonomía. El proceso de indagación es inductivo y el investigador interactuó con los datos buscando respuestas a preguntas que se centran en la experiencia social; cómo se crea y qué significado tiene [31]. Este tipo de método de investigación es de base lingüística por lo que utiliza técnicas de observación para la recolección de datos y su posterior interpretación, analizando las relaciones de significado producidas entre distintos individuos, que en este caso fueron tecnologías multiplataforma.

Además, esta investigación se define como histórica y descriptiva. Es investigación histórica ya que se basa en datos y experiencias del pasado, describe lo que eran los objetos de estudio, en este caso las tecnologías de programación móvil, y pretende entender algún hecho, desarrollo o experiencia anterior para poder representar una búsqueda crítica de la verdad que sustenta estos acontecimientos [32]. Es decir, el cómo

se ha llevado a cabo la programación móvil y cuáles han sido las limitantes a través de la historia, dependiendo de las tecnologías disponibles, para finalmente ver los cambios en la actualidad y analizar los datos de forma más sencilla y contrastada.

La investigación descriptiva trabaja sobre realidades de hecho y su característica principal es presentar una interpretación correcta a través de la descripción exacta de actividades, objetos, procesos, entre otros. El objetivo no es solo la recolección de datos, sino la predicción e identificación de relaciones que existen entre dos o más variables que posteriormente producen una hipótesis o teoría que expone y resume la información para luego analizar los resultados. [33]. Esto fue fundamental a la hora de crear la taxonomía ya que hubo que describir y desmenuzar las características claves para clasificar las diferentes tecnologías de desarrollo móvil.

2.2. Métodos y técnicas

Se realizó la recolección y análisis de datos provenientes de la documentación de tecnologías y las normas ISO 25010 y 25062 para efectuar la selección de las categorías de la taxonomía y poder revelar nuevas interrogantes en el proceso de interpretación de estas. Esto conllevó a utilizar el método de estudio de inducción-deducción que se utiliza con los hechos particulares, siendo deductivo de lo general a lo particular, e inductivo de lo particular a lo general, observando características generales con el propósito de señalar las verdades particulares. El proceso deductivo no es suficiente por sí mismo para explicar el conocimiento, sin embargo, la combinación de ambos métodos significa la aplicación de la deducción en la elaboración de una hipótesis y la aplicación de la inducción en los hallazgos [34]. Es así como se busca detectar las características de los diferentes frameworks ya que, utilizando tecnologías específicas como es el desarrollo web, los lenguajes nativos y la traducción de un lenguaje a otro, forman parte de un objeto general como es ser frameworks de aplicaciones móviles multiplataforma. Y, de la misma forma, se diseñó una estructura que también pueda pasar de lo general, y vasto, como son los frameworks multiplataforma, hacia lo específico que se divide en distintas opciones que se buscará clasificar de forma que denote la utilidad y necesidad de esta según sea el caso de uso.

La técnica de investigación que mejor se adaptó a este trabajo es el estudio documental ya que su principal función se centra en todos aquellos procedimientos que conllevan el uso óptimo y racional de los recursos documentables en las funciones de información, lo cual permitió obtener mucha información que fue necesaria para la definición de la taxonomía, además de señalar con qué etiquetas se iban a identificar los distintos objetos de estudio, como son los frameworks con sus distintas tecnologías.

2.3. Descripción y validación del instrumento

Para la recolección de datos se utilizó:

- Como parte inductiva, la revisión de documentación de frameworks y normas ISO 25010 y 25062. De esta forma se pudo:
 - Obtener información sobre los lineamientos de desarrollo y uso de aplicaciones móviles.
 - Conocer la estructura técnica de la tecnología que implementa un framework y cómo compararla con otra.
 - Definir los lineamientos necesarios para un proyecto de desarrollo.
- Como parte deductiva, lectura de comentarios y encuestas de StackOverFlow hechas a desarrolladores enfocados en este campo. De esta forma se pudo:
 - Conocer la frecuencia con la que se prefiere cada tecnología.
 - Hacer uso racional de los documentos ofrecidos por cada tecnología.
 - Identificar características que no estén bien definidas en la documentación.
 - Analizar el contraste entre la opinión general de desarrollo y uso de aplicaciones móviles con los datos documentados.

De esta manera, primero se estudió la documentación de los frameworks para conocer acerca de la estructura técnica de las aplicaciones móviles según la tecnología empleada.

Además, se definió las categorías de la taxonomía a partir de los puntos de las normas ISO que fueron más relevantes en la documentación de las tecnologías, como son: comportamiento temporal, utilización de recursos, capacidad, red y usabilidad. Esto se dio así ya que en las tecnologías multiplataforma lo más importante es la integración ligera del software con el hardware ya que, si bien la mayoría de estas logra implementarse de forma eficaz, no siempre se logra un funcionamiento eficiente porque

cada tecnología tiene su forma de interactuar con los dispositivos y dependiendo del proceso que se vaya a llevar a cabo la aplicación puede resultar más o menos ligera, rápida e independiente de la red de datos.

La mejor forma de sustentar este estudio es realizar una aplicación móvil y ver las partes fuertes y débiles de cada tecnología para así corroborar su puesto en la taxonomía. Se planteó aplicar la taxonomía en un proyecto de programación móvil utilizando frameworks de aplicaciones móviles multiplataforma para tener distintas versiones de una misma aplicación e inspeccionar su versatilidad y competencia según la tecnología implementada.

La idea es que la aplicación tenga 3 funcionalidades: uso de cámara, localización GPS y explorador de archivos. Estas son funciones que necesitan una buena implementación tanto con el hardware, así como el sistema operativo del dispositivo y dependiendo de cómo se efectúen pueden resultar en mayor o menor carga de trabajo para el CPU y la RAM. Además, se podrán verificar los puntos de la disponibilidad de la red, siendo que se verificó qué tecnología necesita de una constante conexión a internet, y la usabilidad, que trata el tema de la facilidad de implementar una interfaz gráfica fácil y eficazmente.

2.4. Técnicas de procesamiento y análisis de datos

Los datos para hacer la taxonomía fueron obtenidos a partir de las normas ISO 25010 y 25062, en base a su definición, que explica los requerimientos con los que debe cumplir una aplicación, específicamente una aplicación móvil según la norma 25062, y de las características principales con las que debe contar para ser eficiente. Es así como se buscó partir del concepto del desarrollo móvil multiplataforma, seguir con la selección características de las normas ISO y llegar hasta las tecnologías que cumplan o tengan dichos requerimientos para llevar a cabo este estudio.

Además, si bien se define que existen 3 tipos de tecnologías para el desarrollo de aplicaciones móviles; siendo nativo, híbrido y las PWA basadas en web, existen diferentes frameworks que se pueden implementar. Los frameworks seleccionados fueron Flutter, Ionic y se utilizó Kotlin para programar en Android. Estas herramientas de desarrollo fueron seleccionados gracias a las métricas de StackOverFlow[35] de 2019 que, tras la revisión de varias tablas de reporte, las señala como las más populares entre

sus usuarios. En lo referente al desarrollo web, React está en el segundo puesto de los frameworks más populares. Kotlin está como el cuarto lenguaje de programación más querido. Flutter está de tercero dentro de la categoría frameworks, librerías y herramientas más queridas.

Entonces, el hecho es que se creó una aplicación con cada tecnología para probar las ventajas y desventajas que estas poseen y constatarlas con lo que dice la documentación, además de probar si cumplen entre ellas una con las características de otras y con las principales capacidades de la que presumen cada una de ellas.

Finalmente, para realizar una comparativa entre la clasificación expuesta en la taxonomía y el resultado de la implementación de las tecnologías en aplicaciones se utilizó la función de registro del sistema que se encuentra en las opciones de desarrollador en el menú de configuraciones de los dispositivos móviles que funcionan con Android 10, lo cual aportó al refinamiento de la taxonomía planificada teniendo en cuenta principalmente las características móviles como son la utilización de recursos y la usabilidad, características que resaltan mucho en la descripción de cada tecnología, ya que estas son muy cercanas al usuario final por lo que, a pesar de que la aplicación tenga un buen rendimiento, sin un óptimo desarrollo pueden ocasionar un mal uso o rechazo hacia el software desarrollado por parte de los consumidores.

2.5. Normas éticas

Esta investigación está dentro de los rangos éticos y morales vigentes en la sociedad. Es un estudio que no lleva a cabo ninguna actividad que pudiera tornarse fraudulenta o con lo que se pueda infringir los derechos humanos.

CAPÍTULO 3

RESULTADOS

3.1. Estructura técnica de las aplicaciones móviles

La documentación de las tecnologías permitió aclarar la manera en que funcionan las aplicaciones móviles que se generan a través de cada una de ellas. Las aplicaciones nativas funcionan en una estrecha simbiosis con el sistema operativo, ya que éste está

diseñado para saber manejar y ejecutar dichas aplicaciones de forma que éstas puedan cumplir con los requerimientos del usuario y, por ende, con el propósito para el que fueron creadas. Esto se logra gracias a que se compilan en un entorno de trabajo controlado y optimizado para que todo en la aplicación siga los lineamientos del sistema operativo. Por otra parte, las aplicaciones basadas en la web están hechas para ejecutarse en un web view que es una vista de navegador la cual permite renderizar la aplicación y poder interactuar con ella. Esto permite ejecutar una aplicación web dentro de un servidor local, pero cabe decir que también existen páginas web responsive que permiten tener un icono de acceso directo a la aplicación web dando la impresión de que son aplicaciones instaladas en el dispositivo. En este caso se estudió a las aplicaciones web que funcionan localmente, como son las PWA que se pueden crear con Ionic que, a través de otras tecnologías, funcionan bajo un DOM virtual que permite un renderizado y tiempo de respuesta veloz ante los métodos tradicionales de renderizado web. El problema con este tipo de aplicaciones es la integración con el hardware del dispositivo ya que los navegadores web están enfocados al renderizado y para trabajar con los componentes físicos requieren de plugins específicos para cada plataforma lo que hace de la aplicación más pesada y lenta al necesitar diversos middlewares. Es por esto que cuando se quiere implementar una aplicación web para trabajar localmente en un dispositivo móvil, buscando aprovechar las utilidades físicas de éste, que se opta por desarrollar una aplicación híbrida ya que estas también renderizan de forma ligera y eficaz la interfaz gráfica de la aplicación y logran una comunicación más ligera y cercana con el sistema operativo al utilizar APIs de bajo nivel que permiten que los procesos se realicen de forma más rápida y que se pueda realizar configuraciones más específicas dentro de estos, logrando así solucionar las desventajas de las aplicaciones nativas y basadas en la web, pero careciendo del uso de sus virtudes individuales al máximo.

3.2. Características de las Normas ISO que añadir en la taxonomía

El estudio realizado y el uso de las tecnologías seleccionadas llevó a la construcción de la taxonomía y a tener una constante verificación a medida que ésta se iba montando. Según los datos documentados tanto de las tecnologías como en las normas ISO 25010 y 25062 se determinó que las características de comportamiento temporal, utilización de recursos, capacidad, red y usabilidad serían claves para poder diferenciar el camino más efectivo a la hora de crear una aplicación móvil dependiendo de cuáles son los

requerimientos y de la disponibilidad que puede necesitar la aplicación a la hora de funcionar. La selección de estas características se hizo teniendo en cuenta las virtudes que dice tener cada tecnología una frente a otra; como es el comportamiento temporal, la utilización de recursos y la capacidad, este último refiriéndose al cuánto se consienten los requerimientos del usuario. Mientras que, de la norma ISO 25062 se tomaron las características de la red y la usabilidad, siendo que también son factores característicos de las tecnologías estudiadas ya que unas requieren de más uso de internet que otras, siendo estas las aplicaciones basadas en la web, como también existen aquellas que son menos versátiles al crear interfaces gráficas que funcionen en distintas plataformas, a veces al punto de solo poder trabajar en una, siendo este el caso de las aplicaciones nativas.

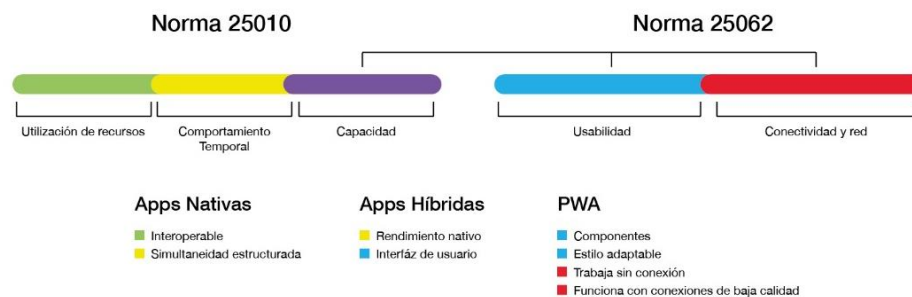


Ilustración 1. Relación de las características cualitativas de las tecnologías y normas ISO [36]–[38]

3.3. Aplicación práctica

Con el desarrollo y complicación de las 3 versiones a comparar de la aplicación de ejemplo realizado, se procedió a la ejecución de cada una y a la posterior recopilación de datos sobre el rendimiento del dispositivo. Las aplicaciones, capturas de pantalla y reportes de Perfetto están disponibles en el siguiente repositorio: https://github.com/JosLuna98/tesis_apps.

Para obtener los informes de Perfetto se ejecutó el “rastreo de acciones”, que está entre las opciones de desarrollador, para luego iniciar cada versión de una aplicación, las cuales utilizan las tecnologías estudiadas, y poder obtener los datos del uso de recursos del dispositivo móvil en el que se ejecutan las aplicaciones para obtener un archivo que permitirá ver el informe en la web de Perfetto.

Bajo el proceso realizado en la metodología se plantea en la taxonomía que aquellas aplicaciones que vayan a tener un gran uso de recursos deberían ser creadas como aplicaciones nativas, ya que éstas tienen un mejor manejo del hardware del dispositivo al estar en una estrecha relación con el sistema operativo logrando así también un buen comportamiento temporal al no necesitar de APIs que hagan una traducción de las instrucciones que envía la aplicación al dispositivo. Así mismo, las aplicaciones híbridas también logran un buen comportamiento temporal ya que estas instrucciones son creadas en un nivel considerablemente bajo de la comunicación entre aplicación y dispositivo pues esto se realiza de forma nativa y se sincroniza con el renderizado de la aplicación para cumplir una función deseada. Cabe decir que las aplicaciones híbridas logran tener mayor capacidad en la usabilidad ya que los métodos para implementar una interfaz gráfica tienen mayor estandarización que los utilizados en las aplicaciones nativas las cuales tienen métodos propios que son muy específicos para cada plataforma, mientras que en las aplicaciones híbridas se utiliza un solo método de elaboración de interfaces agnósticamente a la plataforma que se esté utilizando. Esto último también se da con las aplicaciones basadas en la web, ya que todas éstas están creadas con una base de HTML y CSS que permite una reutilización voraz en cuanto a plataformas se refiere, sin embargo, la capacidad de uso de red del dispositivo debe ser total. Esto quiere decir que estas aplicaciones son la mejor opción cuando se requiere de procesos web y de comunicación con servidores, además de un ahorro de desarrollo, ya que si no se va a utilizar las funciones físicas del dispositivo, o se va a hacer uso menor de estas, la mejor opción es reutilizar un sitio web y adaptarlo para funcionar ya no solo en computadores sino también en dispositivos móviles sin necesidad de crear proyectos desde cero, por eso su gran capacidad de usabilidad y la necesidad de la red.

Al llevar a cabo la ejecución de la aplicación móvil, en sus tres versiones, se hizo uso de la función de registro del sistema para tener un informe de las actividades que realizaba el emulador de Android al momento de realizar los procesos de la aplicación. Los resultados fueron como se esperaban, según la investigación bibliográfica realizada, la versión nativa tuvo un rendimiento normal; ni muy pesado ni muy ligero, siendo el punto de referencia. Mientras que, la versión híbrida tuvo un despliegue ligero pero un consumo mayor que la aplicación nativa, aunque no muy relevante. Sin embargo, la PWA tuvo un consumo de recursos muy bajo ya que se ejecutaba sobre un navegador web, pero esto

también provocó que no se pudo tener una estrecha interacción con el dispositivo al estar hecha con tecnologías web y por ende la aplicación no se comportó como se esperaba.

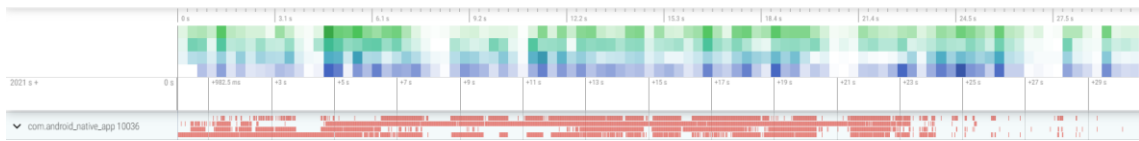


Figura 7. Registro del sistema con aplicación nativa

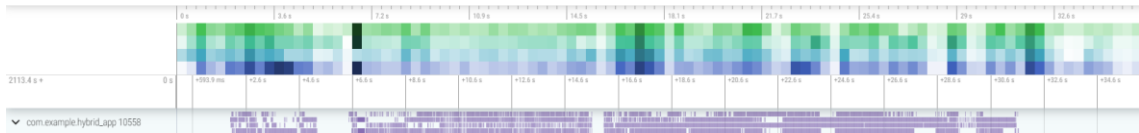


Figura 8. Registro del sistema con aplicación híbrida

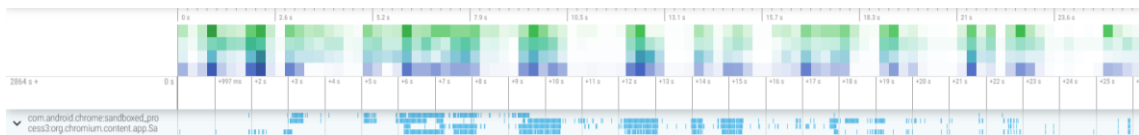


Figura 9. Registro del sistema con aplicación PWA

CAPÍTULO 4

DISCUSIÓN

Según el primer objetivo específico los resultados obtenidos a través de la documentación evidenciaron una estructura técnica muy distintiva en cada una de las tecnologías planteadas en esta investigación lo cual refleja la necesidad de clasificarlas identificando sus mayores diferencias, ya que las similitudes son las más visibles, esto además es corroborado con anterioridad por M. Rezaiee [7] al utilizar la herramienta Hybrid-Scanner; identificando que la frecuencia con la que son llamadas las APIs de comunicación con el sistema operativo son muy diferentes dependiendo del proceso a realizar y la tecnología implementada. Entonces, es ahí cuando se marcan los puntos de diferenciación; en cuanto tiempo toma realizar una acción, cuántos recursos son utilizados y concluir si al final de cuentas la fragmentación de dispositivos de la app resultó o no efectiva.

Según el segundo objetivo específico los resultados tras la lectura y análisis de las normas ISO, contrastado con lo que cada tecnología dice de sí misma, marcaron cuales serían las

principales características con las que debe contar una aplicación móvil para ser eficiente. Fue de esa manera que se seleccionaron principalmente el comportamiento temporal, la utilización de recursos y la capacidad, como los campos de mayor impacto para los usuarios ya que influyen directamente en su interacción con la aplicación, y así mismo, dentro de la capacidad están las características de red e interfaz de usuario que marcan en gran medida cuánto se puede fragmentar una aplicación para llegar a más dispositivos y por ende a más usuarios. Esta parte de la investigación se fijó a partir de los estudios de R. Nunkesser [5] e I. Malavolta [6] quienes expresaban su visión de la necesidad de más aplicaciones híbridas en el mercado tanto para apoyar al desarrollador, en no tener que aprender un nuevo lenguaje, como a la empresa, al no tener que contratar más empleados, y al usuario ya que conseguía un producto familiar que podría llevar con él a todas partes.

Según el tercer objetivo específico los resultados han sido muy favorables ya que la taxonomía planteada está diseñada para apuntar a partes críticas del desarrollo de una aplicación móvil, ya que según los estudios revisados se percibe que muchas veces solo se ven los medios que se poseen para decidirse por una tecnología u otra, cuando lo verdaderamente importante es analizar los procesos que se van a realizar y buscar una solución que se adapte a los recursos que ya posee la entidad y logre que la aplicación tenga un rendimiento óptimo dependiendo de las posibilidades o acciones que se tengan planeadas en un proyecto de programación. Esto es algo que ya había sido propuesto con anterioridad por H. Lee [9] quien explicaba que el impacto de la convergencia digital se va haciendo más significativo con las tendencias tecnológicas emergentes ya que pueden ser fácilmente ignoradas al existir taxonomías que fallan al proveer de una adecuada visión acerca de los cambios tecnológicos y cómo evolucionan los dispositivos móviles. Es así como se buscó características dentro de las normas ISO para que sean como puntos de inflexión que sirvan de base en este estudio y esclarecan y apoyen el desarrollo práctico de aplicaciones que brindan resultados directos, sustentando lo más posible la investigación documental y permitiendo así que la aplicación de esta taxonomía se pueda realizar y desarrollar a través del tiempo; ya que no mira solo las tecnologías que clasifica sino todas sus ventajas, desventajas y consecuencias de uso, de una forma casi agnóstica al basarse en estándares de calidad para el usuario.

CAPÍTULO 5

PROPUESTA

5.1. Título

Taxonomía de tecnologías para desarrollo móvil multiplataforma.

5.2. Representación

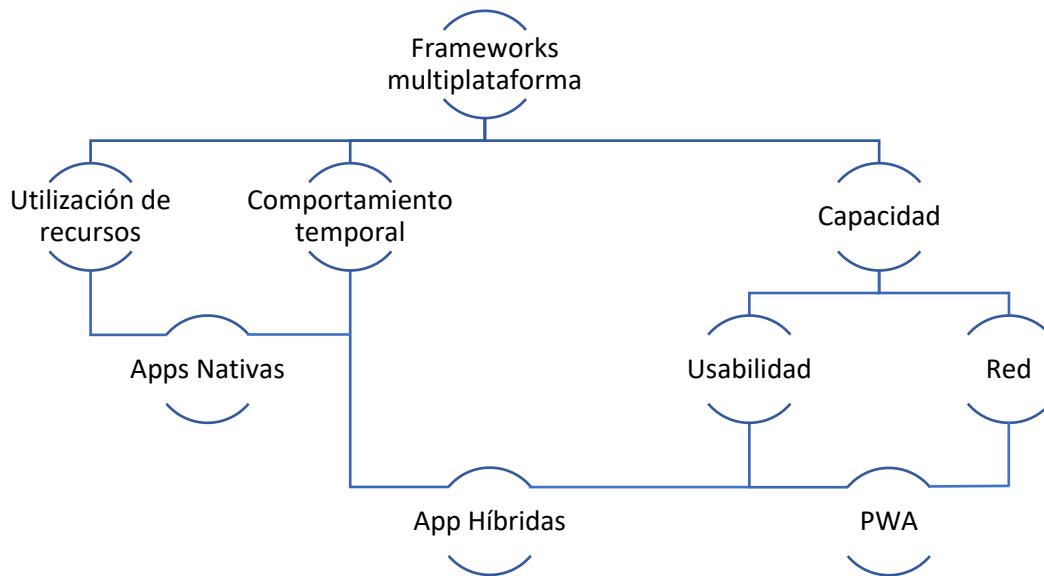


Ilustración 2. Representación gráfica de la taxonomía creada

5.3. Descripción

En esta taxonomía los frameworks de desarrollo móvil multiplataforma se dividen por las diferentes características de las normas ISO 25010 y 25062 que marcan lo más importante que debe poseer una aplicación móvil de calidad; dependiendo del conjunto de características en las que se especialice una aplicación ésta será catalogada como de un tipo u otro. Al conocer la tecnología que mejor desarrolle, o lleve a cabo, los requerimientos de la aplicación e implementar dicha tecnología, será más eficiente el rendimiento de la aplicación para el usuario final. Esto representa lo más importante para

el desarrollo de software, que el usuario final sea quien se lleve el mayor beneficio de todas las facilidades que se puedan aprovechar al momento de realizar este tipo de trabajo.

CAPÍTULO 6

CONCLUSIONES Y RECOMENDACIONES

En esta tesis se formuló una taxonomía de frameworks de desarrollo móvil multiplataforma para poder diferenciar las tecnologías que se pueden llegar a implementar en un proyecto dependiendo de los recursos disponibles y el tipo de procesos que se van a llevar a cabo, esto se logró basándose en requisitos específicos de los estándares ISO 25010 y 25062 lo cual concedió ver más allá de las tecnologías y hacer que el punto central sea la estructura técnica de una aplicación buscando sacar el mejor provecho de las ventajas más necesarias y así identificar la tecnología más acertada para usar en un proyecto de desarrollo móvil.

1. Se distinguió la estructura técnica de cada tipo de aplicación que se puede desarrollar con una determinada tecnología para remarcar las diferencias que existen entre estas y señalar la variante utilización de recursos que se da dependiendo de los procesos que realiza la aplicación para llevar a cabo un fin.
2. Se identificó las categorías y subcategorías para esta taxonomía en base a los requisitos de las normas ISO 25010 y 25062, esto se hizo buscando que la taxonomía sea agnóstica a la tecnología, a pesar de clasificarlas, y que de esta manera su enfoque sea un óptimo desarrollo móvil a través de los estándares de calidad.
3. Se sustentó la investigación documental con una aplicación práctica que permitió ver directamente cómo es la introducción, uso y resultado de utilizar las tecnologías existentes para el desarrollo móvil multiplataforma, logrando así refinar la taxonomía al constatar la información de las tecnologías con los datos las aplicaciones y obteniendo resultados que permitieron magnificar el alcance de la taxonomía.

A través de este estudio se pudo vislumbrar mucha información que señalaba a las tecnologías web como las más populares, sin embargo, no resultaron de las más utilizadas a la hora de realizar aplicaciones móviles multiplataforma ya que se quedaban muy atrás en la parte de interacción y acople con el sistema operativo del dispositivo ejecutor. Como bien dice este estudio, lo primordial es buscar la opción que brinde las mejores capacidades al cliente; los resultados obtenidos mostraron que las tecnologías web no suelen ser el mejor camino a pesar de que han existido distintos intentos por implementarlas en este campo para reciclar a los programadores orientados al ámbito web y así economizar costes. Con este contexto, se puede recomendar la ampliación de los horizontes de conocimiento en los programadores y tratar de estar a la vanguardia antes de querer sobrevivir en una zona de confort que no permite el avance y desarrollo de nuevas tecnologías que lo que buscan no es remplazar a las viejas sino brindar un camino más fácil para que los desarrolladores puedan llevar a cabo su trabajo.

7. REFERENCIAS

7.1. Referencias bibliográficas

- [1] M. A. Nazarenko, “Mobile Operating Systems and Integrated Apps Quality Management System,” in *2018 IEEE International Conference “Quality Management, Transport and Information Security, Information Technologies” (IT&QM&IS)*, 2018, pp. 63–65.
- [2] H. Yan, J. Wang, Y. Wang, and X. Zhou, “An Example for Industry 4.0: Design and Implementation of a Mobile App for Industrial Surveillance Based on Cloud,” in *2017 5th International Conference on Enterprise Systems (ES)*, 2017, pp. 331–336.
- [3] S. Vilkomir, K. Marszalkowski, C. Perry, and S. Mahendrakar, “Effectiveness of Multi-device Testing Mobile Applications,” in *2015 2nd ACM International Conference on Mobile Software Engineering and Systems*, 2015, pp. 44–47.
- [4] R. Cedeño Luna, K. Alcívar Vaca, and D. Ponce Vásquez, “Observaciones acerca de los dispositivos móviles,” *Dominio las Ciencias*, vol. 3, no. 4, pp. 89–103, 2017.
- [5] R. Nunkesser, “Beyond web/native/hybrid,” in *Proceedings of the 5th*

International Conference on Mobile Software Engineering and Systems - MOBILESoft '18, 2018, pp. 214–218.

- [6] I. Malavolta, “Web’based Hybrid Mobile Apps: State of the Practice and Research Opportunities,” 2016, p. 241,242.
- [7] S. Pouryousef, M. Rezaiee, and A. Chizari, “Let me Join Two Worlds! Analyzing the Integration of Web and Native Technologies in Hybrid Mobile Apps,” in *Proceedings - 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and 12th IEEE International Conference on Big Data Science and Engineering, Trustcom/BigDataSE 2018*, 2018, pp. 1814–1819.
- [8] M. L. Hale and S. Hanson, “A Testbed and Process for Analyzing Attack Vectors and Vulnerabilities in Hybrid Mobile Apps Connected to Restful Web Services,” in *Proceedings - 2015 IEEE World Congress on Services, SERVICES 2015*, 2015, pp. 181–188.
- [9] H. Lee, “Research on the impact of technology taxonomy for the tracking of technology convergence,” in *9th International Conference on Information and Communication Technology Convergence: ICT Convergence Powered by Smart Intelligence, ICTC 2018*, 2018, pp. 1452–1456.
- [10] G. MacGregor, *The Accidental Taxonomist*, vol. 60, no. 1. 2011.
- [11] Mohammad Naghi Namakforoosh, “Metodología de la investigación - Mohammad Naghi Namakforoosh - Google Libros,” 2005. [Online]. Available: <https://books.google.com.pe/books?id=ZEJ7-0hmvhwC&printsec=frontcover#v=onepage&q&f=true>. [Accessed: 09-Jan-2020].
- [12] P. Olla and C. Shimskey, “mHealth taxonomy: a literature survey of mobile health applications,” *Health Technol. (Berl.)*, vol. 4, no. 4, pp. 299–308, Apr. 2015.
- [13] L. Codina, “Qué es una taxonomía y cómo se aplica a un sitio web,” 2018. [Online]. Available: <https://www.lluiscodina.com/taxonomia-sitio-web/>.

[Accessed: 09-Jan-2020].

- [14] Y. Hassan Montero, F. J. Martín Fernández, and Ó. Martín Rodríguez, “Clasificaciones Facetadas y Metadatos (I): Conceptos Básicos,” 2003. [Online]. Available: http://www.nosolousabilidad.com/articulos/clas_facetadas1.htm.
- [15] “¿Qué son las normas ISO y cuál es su finalidad?” [Online]. Available: <https://www.isotools.org/2015/03/19/que-son-las-normas-iso-y-cual-es-su-finalidad/>. [Accessed: 17-Feb-2020].
- [16] “ISO 25010.” [Online]. Available: <https://iso25000.com/index.php/normas-iso-25000/iso-25010>. [Accessed: 17-Feb-2020].
- [17] “ISO/IEC 25062:2006(en), Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Common Industry Format (CIF) for usability test reports.” [Online]. Available: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25062:ed-1:v2:en>. [Accessed: 17-Feb-2020].
- [18] K. Moumane, A. Idri, and A. Abran, “Usability evaluation of mobile applications using ISO 9241 and ISO 25062 standards,” *Springerplus*, vol. 5, no. 1, 2016.
- [19] Iec, “Software engineering-Software product Quality Requirements and Evaluation (SQuaRE)-Common Industry Format (CIF) for usability test reports,” 2006.
- [20] “Philosophy and Goals | Android Open Source.” [Online]. Available: <https://web.archive.org/web/20120501080416/http://source.android.com/about/philosophy.html>. [Accessed: 17-Feb-2020].
- [21] “Descripción general del registro del sistema.” [Online]. Available: <https://developer.android.com/topic/performance/tracing>. [Accessed: 22-Aug-2020].
- [22] “Perfetto UI.” [Online]. Available: <https://ui.perfetto.dev/#/>. [Accessed: 21-Aug-2020].

- [23] “Todas las novedades de iOS 12, el nuevo sistema operativo para iPhone - AS.com.” [Online]. Available: https://as.com/meristation/2018/06/04/betech/1528132311_239830.html. [Accessed: 17-Feb-2020].
- [24] M. Yuan and L. Fan, “ATS software framework design pattern and application,” in *Proceedings - 5th International Conference on Instrumentation and Measurement, Computer, Communication, and Control, IMCCC 2015*, 2016, pp. 141–146.
- [25] N. Boushehrinejadmoradi, V. Ganapathy, S. Nagarakatte, and L. Iftode, “Testing cross-platform mobile app development frameworks,” in *Proceedings - 2015 30th IEEE/ACM International Conference on Automated Software Engineering, ASE 2015*, 2016, pp. 441–451.
- [26] “Qué es una App Nativa | Qode.” [Online]. Available: <https://www.qode.pro/blog/que-es-una-app-nativa/>. [Accessed: 20-Jan-2020].
- [27] “¿Qué son las aplicaciones web progresivas?” [Online]. Available: <https://web.dev/what-are-pwas/>. [Accessed: 23-Apr-2020].
- [28] “Auditar apps web con Lighthouse | Tools for Web Developers.” [Online]. Available: <https://developers.google.com/web/tools/lighthouse/>. [Accessed: 11-Aug-2020].
- [29] YeePLY, “Programación en HTML5 para crear apps multiplataforma.” [Online]. Available: <https://www.yeeply.com/blog/programar-apps-multiplataforma-html5/>. [Accessed: 22-Jan-2020].
- [30] D. Sandoval, “APLICACIONES HÍBRIDAS: ¿QUÉ SON Y CÓMO USARLAS?” [Online]. Available: <https://www.nextu.com/blog/aplicaciones-hibridas-que-son-y-como-usarlas/>. [Accessed: 22-Jan-2020].
- [31] UJAEN, “Metodología cualitativa.” [Online]. Available: http://www.ujaen.es/investiga/tics_tfg/enfo_cuali.html. [Accessed: 05-Feb-2020].
- [32] D. B. Van Dalen and W. J. Meyer, “LA INVESTIGACIÓN HISTÓRICA |

- noemagico.” [Online]. Available: <https://noemagico.blogia.com/2006/100101-la-investigaci-n-hist-rica.php>. [Accessed: 05-Feb-2020].
- [33] D. B. Van Dalen and W. J. Meyer, “LA INVESTIGACIÓN DESCRIPTIVA | noemagico.” [Online]. Available: <https://noemagico.blogia.com/2006/091301-la-investigaci-n-descriptiva.php>. [Accessed: 05-Feb-2020].
- [34] I. Vásquez Hidalgo, “Tipos de estudio y métodos de investigación - GestioPolis,” 18-Dec-2005. [Online]. Available: <https://www.gestiopolis.com/tipos-estudio-metodos-investigacion/>. [Accessed: 12-Feb-2020].
- [35] “Stack Overflow Developer Survey 2019.” [Online]. Available: <https://insights.stackoverflow.com/survey/2019#technology>. [Accessed: 02-Aug-2020].
- [36] “Desarrolladores de Android | Android Developers.” [Online]. Available: <https://developer.android.com/kotlin>. [Accessed: 22-Aug-2020].
- [37] “Flutter - Beautiful native apps in record time.” [Online]. Available: <https://flutter.dev/>. [Accessed: 22-Aug-2020].
- [38] “Progressive Web App Development - Ionic PWA Web Apps.” [Online]. Available: <https://ionicframework.com/pwa>. [Accessed: 22-Aug-2020].