



**UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD INGENIERÍA
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA
INFORMACIÓN**

**APLICACIÓN MÓVIL MULTIPLATAFORMA PARA EL CONTROL
DE INVENTARIO EN LA BOUTIQUE NICOLETH, UTILIZANDO LA
PLATAFORMA XAMARIN**

**Trabajo de Titulación para optar al título de Ingeniero en
Tecnologías de la Información**

Autores:

Chacon Robalino Carlos David
Estrada Arias Dennis Adrian

Tutor:

Ing. Ana Elizabeth Congacha Aushay

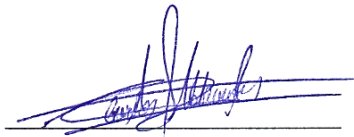
Riobamba, Ecuador. 2023

DECLARATORIA DE AUTORÍA

Nosotros Chacon Robalino Carlos David y Estrada Arias Dennis Adrian, con cédula de ciudadanía 0604549352 y 1722036652, autores del trabajo de investigación titulado: **Aplicación móvil multiplataforma para el control de inventario en la boutique Nicoleth, utilizando la plataforma Xamarin**, certificamos que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de nuestra exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 24/03/2023.



Carlos David Chacon Robalino
C.I: 0604549352



Dennis Adrian Estrada Arias
C.I: 1722036652



ACTA FAVORABLE - INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN

En la Ciudad de Riobamba, a los 09 días del mes de febrero de 2023, luego de haber revisado el Informe Final del Trabajo de Investigación presentado por los estudiantes **DENNIS ADRIAN ESTRADA ARIAS** con CI: **1722036652** y **CARLOS DAVID CHACON ROBALINO** con CI: **0604549352**, de la carrera **TECNOLOGIAS DE LA INFORMACION Y COMUNICACIÓN** y dando cumplimiento a los criterios metodológicos exigidos, se emite el **ACTA FAVORABLE DEL INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN** titulado **“APLICACIÓN MÓVIL MULTIPLATAFORMA PARA EL CONTROL DE INVENTARIO EN LA BOUTIQUE NICOLETH, UTILIZANDO LA PLATAFORMA XAMARIN”**, por lo tanto se autoriza la presentación del mismo para los trámites pertinentes.



Firmado electrónicamente por:
**ANA ELIZABETH
CONGACHA AUSHAY**

Ing. Ana Elizabeth Congacha Aushay

TUTOR(A)

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal para la evaluación del trabajo de investigación Aplicación móvil multiplataforma para el control de inventario en la Boutique Nicoletth, utilizando la plataforma Xamarin por Dennis Adrian Estrada Arias con cédula de identidad número 1722036652 y Carlos David Chacon Robalino con cédula de identidad número 0604549352, bajo la tutoría de Mgs. Ana Elizabeth Congacha Aushay; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

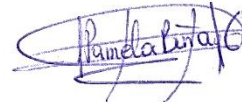
De conformidad a la normativa aplicable firmamos, en Riobamba 24 días del mes de marzo del 2023

Presidente del Tribunal de Grado
Mgs. Lady Espinoza



Firma

Miembro del Tribunal de Grado
Mgs. Pamela Buñay



Firma

Miembro del Tribunal de Grado
Mgs. Milton López



Firma



CERTIFICACIÓN

Que, CHACON ROBALINO CARLOS DAVID con CC: 0604549352 y ESTRADA ARIAS DENNIS ADRIAN con CC: 1722036652 , estudiantes de la Carrera **INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**, Facultad de **INGENIERIA** han trabajado bajo mi tutoría el trabajo de investigación titulado " **Aplicación móvil multiplataforma para el control de inventario en la Boutique Nicoleth, utilizando la plataforma Xamarin.**", cumple con el 9%, de acuerdo al reporte del sistema Anti plagio **URKUND**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 07 de marzo de 2023



Firmado electrónicamente por:
ANA ELIZABETH
CONGACHA AUSHAY

Ing. Ana Elizabeth Congacha Aushay
TUTORA PROYECTO DE INVESTIGACIÓN

DEDICATORIA

Dedico el presente trabajo de investigación a Dios quien nunca me ha permitido rendirme en este largo camino de formación profesional, por siempre protegerme con su manto y darme la sabiduría necesaria en mis momentos más difíciles.

A mis padres quienes con mucho esfuerzo han logrado que yo pueda cumplir con una más de mis metas, también por ser ese sustento anímico incondicional durante todo este camino de preparación.

Dennis Adrian Estrada Arias

Quiero dedicar esta tesis en primer lugar a Dios por ayudarme a cumplir esta gran meta de mi vida, además de haberme brindado su bendición y haberme dado una excelente familia.

A mis padres Carlos Enrique Chacon y Gladis Natividad Robalino que siempre me apoyaron y nunca me abandonaron, a mis hermanas y hermano que de igual manera siempre me apoyaron y nunca me dejaron en los malos momentos.

También a Cristina Vásquez que fue de gran apoyo y ayuda ya que siempre estuvo dispuesta a ayudarme a cumplir mi meta.

Carlos David Chacon Robalino

AGRADECIMIENTO

En primer lugar, quiero agradecer a Dios por regalarme una familia tan maravillosa que con su esfuerzo, apoyo, confianza y amor me han dado todo lo necesario para poder seguir tras mis metas y cumplirlas.

Agradecido con la Ing. Ana Elizabeth Congacha nuestro tutor de tesis quien con mucha paciencia y dedicación ha sabido guiarnos y motivarnos arduamente durante la realización de nuestro proyecto de investigación, también quiero hacer mención a los docentes quienes me impartieron cátedras regalándome su conocimiento durante toda la carrera.

Agradecerles a todos mis amigos y compañeros con quienes he podido compartir todo este proceso de formación ya que sin ellos no habría podido tener tantos momentos extraordinarios en esta etapa de mi vida, sin duda son personas que jamás olvidare.

Dennis Adrian Estrada Arias

En primer lugar, agradecer a mis padres por haberme dado su apoyo incondicional a mi familia por siempre haberme apoyado y a Dios por ser un pilar fundamental de mi vida.

De manera especial a la Ing. Ana Elizabeth Congacha nuestra tutora de tesis que con paciencia y constancia nos guió y motivo para lograr culminar este proyecto de investigación. Mencionar a todos los docentes que me acompañaron en mi vida universitaria que sin sus enseñanzas no hubiera logrado cumplir mi meta.

Agradecer de igual manera a mis compañeros con los cuales compartimos este proceso académico y a mis amigos que siempre me apoyaron y con los cuales compartimos momentos geniales y sin duda los recordare para siempre.

Carlos David Chacon Robalino

ÍNDICE GENERAL

DECLARATORIA DE AUTORÍA.....	2
DICTAMEN FAVORABLE DEL PROFESOR TUTOR.....	3
CERTIFICADOS DE LOS MIEMBROS DEL TRIBUNAL	4
CERTIFICADO ANTIPLAGIO	5
DEDICATORIA.....	6
AGRADECIMIENTO	7
ÍNDICE GENERAL.....	8
ÍNDICE DE TABLAS.....	11
ÍNDICE DE FIGURAS	12
RESUMEN	13
ABSTRACT	14
INTRODUCCIÓN.....	15
CAPÍTULO I.....	16
1. PLANTEAMIENTO DEL PROBLEMA.....	16
1.1 Problema y Justificación.....	16
1.2 Formulación del Problema.....	17
1.3 Objetivos.....	17
1.3.1 Objetivo general	17
1.3.2 Objetivos específicos.....	17
CAPÍTULO II.....	18
2. MARCO TEÓRICO.....	18
2.1 Aplicaciones móviles.....	18
2.1.1 Android.....	18
2.1.2 iOS.....	19
2.2 Aplicaciones móviles multiplataforma	19
2.3 Xamarin	22
2.3.1 Xamarin Forms.....	22
2.3.2 XAML de Xamarin Forms	25
2.4 Metodologías Ágiles	26
2.4.1 Metodología Scrum.....	26

2.5	Normas ISO	27
2.5.1	ISO/IEC 25010	28
2.5.2	Eficiencia de desempeño	28
2.5.3	Utilización de recursos	28
2.6	JMeter	29
CAPÍTULO III		30
3.	METODOLOGÍA.....	30
3.1	Tipo y diseño de la investigación	30
3.1.1	Según la fuente de investigación.....	30
3.1.2	Según el objeto de estudio.....	30
3.1.3	Según el tipo de variable	30
3.2	Unidad de Análisis.....	30
3.3	Población y muestra.....	31
3.4	Técnicas e instrumentos de recolección de datos	31
3.5	Procesamiento de datos.....	31
3.6	Operacionalización de variables	32
3.7	Desarrollo de la aplicación móvil multiplataforma	33
3.7.1	Inicio.....	33
3.7.2	Planificación.....	34
3.7.3	Desarrollo.....	40
3.7.4	Implementación.....	43
3.7.5	Lanzamiento	45
CAPÍTULO IV		47
4.	RESULTADOS Y DISCUSIÓN	47
	Dispositivos utilizados para la medición de las métricas	51
	Análisis del dispositivo Android	52
	Análisis del dispositivo iOS	52
	Evaluación de las métricas dispositivo Android	53
	Evaluación de las métricas dispositivo iOS	53
4.1	Análisis de los resultados obtenidos	54
4.1.1	Tiempo de Respuesta	54
4.1.2	Tiempo de Espera.....	55

4.1.3 CPU	56
4.1.4 RAM.....	56
4.1.5 Eficiencia de Desempeño	57
CAPÍTULO V.	57
5. CONCLUSIONES y RECOMENDACIONES	57
CONCLUSIONES	57
RECOMENDACIONES	59
BIBLIOGRAFÍA	60
ANEXOS	62
Anexo 1 Pantallas de la aplicación móvil multiplataforma.	62
Anexo 2 Entrevista documento requerimientos.....	64
Anexo 3 Apis utilizadas para la aplicación móvil.	70
Anexo 4 Configuración JMeter en los dispositivos móviles	71

ÍNDICE DE TABLAS

Tabla 1 Comparativa de los sistemas operativos móviles.	18
Tabla 2 Herramientas para el desarrollo multiplataforma.	20
Tabla 3 Controles de Xamarin Forms Material Visual.....	24
Tabla 4 Ejemplos de utilidades proporcionadas por Xamarin Essentials.....	24
Tabla 5 Ventajas y Desventajas de XAML.	25
Tabla 6 Fases de la Metodología SCRUM.	27
Tabla 7 Operacionalización de variables.....	32
Tabla 8 Roles Scrum.....	34
Tabla 9 Historia de Usuario.....	34
Tabla 10 Refinamiento del Product Backlog.....	36
Tabla 11 Actividades del Sprint 1	38
Tabla 12 Actividades del Sprint 2	38
Tabla 13 Actividades del Sprint 3	39
Tabla 14 Indicadores a evaluar.....	50
Tabla 15 Ecuaciones sugeridas por la norma para obtener los valores de los indicadores.	50
Tabla 16 Porcentajes de ponderación	51
Tabla 17 Descripción de los dispositivos usados.	51
Tabla 18 Medición indicadores dispositivo Android.	52
Tabla 19 Medición indicadores dispositivo iOS.....	53
Tabla 20 Resultados dispositivo Android.....	53
Tabla 21 Resultados dispositivo iOS.....	54
Tabla 22 Resultados obtenidos de la evaluación de los dispositivos Android y iOS.....	54

ÍNDICE DE FIGURAS

Figura 1	Arquitectura general de una aplicación Xamarin multiplataforma.....	22
Figura 2	Metodologías ágiles más utilizadas.....	26
Figura 3	Características de calidad de las normas ISO/IEC 25010.....	28
Figura 4	Entorno de instalación Visual Studio.....	33
Figura 5	Adecuación de los módulos de la aplicación móvil.....	39
Figura 6	Diseño conceptual de la aplicación.....	41
Figura 7	Diseño de Interfaz de la aplicación.....	42
Figura 8	Principales pantallas de la Implementación del diseño elaborado.....	43
Figura 9	Diagrama base de datos.....	44
Figura 10	Código de implementación que servirá para ambos sistemas operativos.....	45
Figura 11	Principales pantallas de la aplicación móvil Android y iOS.....	49
Figura 12	Resultado del tiempo de respuesta.....	55
Figura 13	Resultado del tiempo de espera.....	55
Figura 14	Resultado del uso del CPU.....	56
Figura 15	Resultado del uso de la RAM.....	56
Figura 16	Resultado de la eficiencia de desempeño en los dispositivos evaluados.....	57

RESUMEN

El presente proyecto de investigación tiene como objetivo desarrollar una aplicación móvil multiplataforma en la boutique Nicoleth de la ciudad de Riobamba, para realizar el control de inventario que permita llevar acabo un control de sus ingresos y egresos de productos, estableciendo un orden detallado por su valor e importancia.

Para el desarrollo de la aplicación móvil multiplataforma, se utilizó Xamarin debido a que nos permite compartir código y la interfaz de usuario entre plataformas además de obtener una aplicación moderna y con un buen rendimiento para los sistemas operativos móviles Android y iOS. Además, se utilizó la metodología ágil SCRUM la cual cuenta con cinco etapas para la producción de la aplicación. Por otra parte, mediante la norma ISO/IEC 25010 se realizó la evaluación del rendimiento de la aplicación móvil, utilizando la característica eficiencia de desempeño la cual consta de ciertas subcaracterísticas como: comportamiento temporal y utilización de recursos. Para las pruebas de rendimiento de la aplicación móvil se utilizó el software Apache JMeter en la cual se realizó simulaciones con peticiones en tiempo real, esto sé lo hizo para un dispositivo Android y iOS.

Con los resultados obtenidos mediante la aplicación de la norma se pudo evidenciar que el rendimiento de la aplicación es óptimo, obteniendo calificaciones de 8,43 para Android y 8,48 en iOS, con lo cual se pudo verificar que resulta eficaz el uso de la aplicación al momento de realizar el proceso de control de inventario, logrando también la compatibilidad para Android y iOS.

Palabras claves: Aplicaciones móviles, Aplicaciones móviles multiplataforma, Xamarin, Control de inventarios, Scrum.

ABSTRACT

This current research project has as the main objective to develop a mobile multiplatform app at the Nicoleth Boutique in Riobamba, to do the stocktaking control, which is very important and fundamental in a commercial organization since it allows to carry out a control of its incomes and expenses of the supplies and products setting a detailed order because of its value and importance.

For the mobile multiplatform app, it was used Xamarin, due to, it permits to share codes and share the user interface between platforms besides that, to obtain a modern application and with a high performance for the mobile operating system Android and iOS. Additionally, it was used the SCRUM methodology which consists of five stages, maintaining an agile and collaborative work for the application production. On the other hand, through the rule ISO/IEC 25010 an evaluation of the mobile app performance was carried out, using the efficient characteristic which consist on certain sub characteristics like: temporal behavior and utilization resources. For the performance test the mobile app it was used the Apache JMeter software in which some simulations with repetitions were done in real time, it was developed for Android and iOS devices.

With the obtained results, through the rule app we could make evident that the app performance is excellent, obtaining grades like 8,43 for Android and 8,48 with iOS, for that we could verify that results effective the use of the application at the moment to realize the control process of the stocktaking, achieving also the compatibility for Android y iOS.

Key words: mobile applications, mobile multiplatform application, Xamarian, Stocktaking control, Scrum

Reviewed by:



Lic. Andrea Rivera
ENGLISH PROFESSOR
C.C 0604464008

INTRODUCCIÓN

Actividades que en un tiempo atrás solían realizarse utilizando una computadora, en la actualidad se llevan a cabo desde los dispositivos móviles. El desarrollo de aplicaciones para dispositivos móviles es un campo en constante evolución con gran interés económico y científico (Delía, 2017). Una de las dificultades al desarrollar aplicaciones para móviles es la diversidad de dispositivos y sistemas operativos, teniéndose que construir una versión diferente para cada caso en un lenguaje y una herramienta diferente. El desarrollo multiplataforma se centra en la reutilización de código, compartiendo la misma codificación entre las versiones para las distintas plataformas, es el proceso que crea una aplicación en un lenguaje de programación que facilita su exportación y visualización en cualquier tipo de dispositivo (Delia et al., 2014).

Según Johnson et al. (2022), Xamarin es una plataforma de código abierto para el desarrollo de aplicaciones móviles multiplataforma con rendimiento comprobado para Android, iOS y Windows .NET, además permite a los desarrolladores compartir un promedio del 90% de la aplicación entre plataformas.

Por otra parte, el control de inventarios es de gran utilidad dentro de las organizaciones, permite conservar un control de ingresos, consumo y comercialización de insumos o productos, estableciendo un orden detallado por su valor e importancia, generando un alcance positivo en la rentabilidad de las empresas (Arroba et al., 2018).

En este contexto, en la microempresa Nicoleth ubicada en la ciudad de Riobamba, el control de inventario es realizado de forma manual ocasionando errores que impiden una correcta observación, comprobación y verificación del inventario, además en el escenario se maneja diferentes dispositivos con distintos sistemas operativos móviles.

Por lo tanto, el desarrollo e implementación de una aplicación móvil multiplataforma utilizando Xamarin permite a la microempresa Nicoleth realizar un control eficiente de su inventario manteniendo el correcto balance de existencias en su almacén.

La estructura del documento presenta la siguiente forma: El capítulo I describe el planteamiento del problema, formulación del problema y objetivos, el capítulo II desarrolla el marco teórico que permite conocer los conceptos relacionados con aplicaciones móviles, aplicaciones móviles multiplataforma, control de inventarios, Xamarin, norma ISO/IEC 25010, el capítulo III indica la metodología usada, en el capítulo IV se representan los resultados de la evaluación de la eficiencia de desempeño de la aplicación móvil multiplataforma utilizando el criterio comportamiento temporal y utilización de recursos de la norma ISO/IEC 25010, y finalmente en el capítulo V se presenta las conclusiones y recomendaciones.

CAPÍTULO I.

1. PLANTEAMIENTO DEL PROBLEMA

1.1 Problema y Justificación

El internet, los teléfonos inteligentes, las aplicaciones móviles y la nueva forma de comunicarnos han cambiado la manera en que vivimos, en América Latina nueve de cada diez personas utilizan un teléfono inteligente según datos recabados por la segunda edición del estudio IMS Mobile en Latam, 81% de la población utiliza el sistema operativo Android, mientras que iOS tiene un impacto menor con 17% (Lazcano et al., 2019).

Es necesario resaltar que el uso de aplicaciones móviles actualmente es indispensable en el Ecuador, esta tendencia crece en el país y a escala global de la mano del desarrollo de aplicaciones móviles y de la inserción de los teléfonos inteligentes. Según el Instituto Nacional de Estadísticas y Censos (INEC) revelo que hasta mayo de 2019 se contaron 1 261 944 de ecuatorianos que contaban con un smartphone (Grupo El Comercio, 2019).

En este contexto, el desarrollo de aplicaciones para dispositivos móviles es cada vez indispensable para cualquier empresa u organización, debido a que las aplicaciones móviles son utilizadas en distintas áreas tales como la educación, salud, industria, finanzas, etc. Al hacer uso de estos aplicativos, los usuarios tienen diversas funciones permitiéndoles realizar trabajos más eficientes.

Resulta innegable que el impacto de las aplicaciones multiplataforma está creciendo incesantemente en el mundo de las aplicaciones móviles. Cada vez aparecen más framework de desarrollo capaces de facilitar la creación de un único aplicativo que pueda ser usado por los principales sistemas operativos móviles, haciendo frente a los ya tradicionales desarrollos nativos (Ferrer Sanz, 2021).

Por lo expuesto Xamarin es una plataforma para el desarrollo de aplicaciones móviles multiplataforma contiene codificación que puede ser reutilizable en un 90% para los sistemas de iOS y Android (Johnson et al., 2022).

Por otra parte, Arroba et al. (2018) mencionan que “el control de inventarios es el corazón de cualquier compañía que compra y vende bienes o servicios; por lo tanto, la importancia de un manejo de inventario adecuado permite obtener mejores resultados financieros”.

El inventario al ser gestionado de forma manual aumenta la probabilidad de cometer errores humanos como: pérdida de productos, exceso de existencias, problemas de stock, datos inexactos, seguimientos inconsistentes. Esto ocurre cuando las empresas se acostumbran a los viejos métodos de hacerlo todo uno mismo creyendo que la percepción humana es más confiable, pero lo cierto es que no es más que un pensamiento prejuicioso con respecto a la

ayuda tecnológica. Actualmente una de las problemáticas que presenta la microempresa Nicoleth de la ciudad de Riobamba, es que no usa nuevas tecnologías para realizar un correcto control de su inventario.

Por lo antes mencionado la investigación plantea desarrollar una aplicación móvil multiplataforma utilizando Xamarin para mejorar los procesos de: ingreso de los artículos, la salida de los artículos con su debido proceso de extracción del inventario al ser vendidos, búsqueda de la existencia de los artículos, y notificaciones sobre el abastecimiento del stock.

1.2 Formulación del Problema

¿Cómo la aplicación móvil multiplataforma mejorará el control del inventario en la microempresa Nicoleth de la ciudad de Riobamba?

1.3 Objetivos

1.3.1 Objetivo general

- Desarrollar una aplicación móvil multiplataforma para el control de inventario en la boutique Nicoleth utilizando la plataforma Xamarin.

1.3.2 Objetivos específicos

- Analizar la plataforma Xamarin para el desarrollo de aplicaciones móviles multiplataforma.
- Implementar la aplicación móvil multiplataforma utilizando la plataforma Xamarin.
- Evaluar la eficiencia de desempeño de la aplicación móvil multiplataforma con la norma ISO/IEC 25010 utilizando el criterio utilización de recursos.

CAPÍTULO II.

2. MARCO TEÓRICO.

2.1 Aplicaciones móviles

Las aplicaciones móviles son programas informáticos o software instalado en dispositivos móviles que permiten una gama amplia de usos y funcionalidades, incluidos el teléfono, el video, la música, la televisión y servicios de internet. En un principio las apps se desarrollaron como herramientas de trabajo que ofrecían información general, como el calendario o correo electrónico. Con el paso del tiempo se ha incrementado rápidamente el desarrollo y variabilidad gracias a nuevas tecnologías logrando obtener apps de pago y gratuitas (Pérez et al., 2019).

“Las aplicaciones móviles son una tendencia para las generaciones digitales pues estas se han popularizado porque funcionan como herramientas para realizar tareas de todo tipo y así facilitar las actividades de quienes las usan” (García & Cáceres, 2019). En la tabla 1 se observa una comparativa entre los sistemas operativos móviles.

Tabla 1 Comparativa de los sistemas operativos móviles.

Sistema Operativo	Tipo de SO	Lenguaje de Programación	Kernel	Estándar Soportado	Tienda
Android	Abierto	Java, Kotlin	Linux	GSM, CDMA, 4G	Play Store
iOS	Cerrado	Objective C, Swift	OS X	GSM, CDMA, 4G	App Store
Windows Phone	Cerrado	C#	Windows NT	GSM, CDMA, 4G	Microsoft Store

Fuente. elaboración propia.

Los sistemas operativos móviles son responsables de definir y identificar funciones y características de los dispositivos móviles, incluidos la sincronización de aplicaciones, correo electrónico, mensajes de texto, la rueda de control y el teclado (techopedia, 2022).

2.1.1 Android

Este sistema operativo está basado en Linux para teléfonos móviles, además utilizan este sistema tablets, reproductores de música, netbooks e incluso PC's. Android permite programar en un entorno de Java aplicaciones sobre Dalvik; una variación de la máquina de Java con compilación en tiempo de ejecución.

La gran diferencia de android con otros sistemas operativos es que cualquier persona que tenga conocimientos acerca de programación puede crear widgets, aplicaciones o modificar el mismo sistema operativo, esto gracias a que su código es libre. Por lo tanto teniendo conocimientos en en lenguaje Java, va ser sencillo comenzar a programar en esta plataforma. (Báez et al., 2019).

2.1.2 iOS

Este sistema fue lanzado por Apple y su nombre proviene de iPhoneOS que quiere decir iPhone Operative System en inglés. Este sistema es cerrado quiere decir que solo se puede utilizar en dispositivos de la marca Apple. La diferencia que este sistema presenta respecto a Android es que en este sistema se puede instalar en una amplia gama de telefonos de todas las marcas, a diferencia de iOS que es un sistema cerrado y de uso exclusivo para dispositivos de la marca (Rocio, 2023).

2.2 Aplicaciones móviles multiplataforma

Una aplicación móvil multiplataforma está diseñada para funcionar en muchos sistemas móviles. Estas aplicaciones suelen crearse utilizando tecnologías que proporcionan una capa adicional de abstracción por encima de la API del sistema, uniforme para cada plataforma (Grzmil et al., 2017). La tabla 2 nos muestra herramientas para el desarrollo multiplataforma y sus características principales.

Tabla 2 Herramientas para el desarrollo multiplataforma.

Herramienta	Creado	Características Principales
<ul style="list-style-type: none"> Xamarin 	<p>Xamarin es el framework desarrollado por Microsoft</p>	<ul style="list-style-type: none"> Cuenta con enlaces completos para los SDK, además proporciona durante el desarrollo y el tiempo de compilación una sólida comprobación de tipos, y genera menos errores en tiempo de ejecución y nos brinda aplicaciones de mayor calidad. Ofrece interoperabilidad con Objective C, Java, y C++, permite utilizar un extensa variedad de Código de terceros y unir bibliotecas nativas de Java y Objective, mediante una sintaxis declarativa. Construcciones de lenguaje moderno: Se escriben en C#, características de lenguaje dinámico, construcciones funcionales como lambdas, LINQ, características de programación en paralelo, genéricos y mucho más. Biblioteca de clases base (BCL) sólida: El código de C# existente se puede compilar para usarlo en una aplicación, lo que proporciona acceso a miles de bibliotecas que agregan funcionalidad más allá de la BCL. Presenta un entorno de desarrollo integrado (IDE), esto incluye sistema de administración de soluciones y proyectos culminación automática de código, un control para el código fuente que viene integrado, etc.

➤ Ionic	Fue creado en 2013 por Drifty Co.	<ul style="list-style-type: none"> • Se puede escribir aplicaciones para Android, iOS y Windows Phone que compartan código hasta un 90%, gracias a esto se reduce el tiempo de salida al mercado y el costo de desarrollo para dispositivos móviles. ➤ Independiente de la plataforma ofrece un aspecto parecido a una aplicación nativa, podrá determinar la plataforma en la que está trabajando, basándose en CSS. ➤ Ofrece compatibilidad entre plataformas con un desarrollo de aplicaciones veloz y ágil que reduce tiempo y esfuerzo para desarrollar aplicaciones. ➤ Cuenta con cualidades de AngularJS esto debido a que el framework se basa en AngularJS, utilizando la mayoría de las funcionalidades, lo cual ayuda a desarrolladores a crear aplicaciones web y móviles atractivas con variedad de características.
✓ React Native	Fue creado por Facebook	<ul style="list-style-type: none"> ✓ Los desarrolladores no necesitan mantener dos códigos separados haciendo el proceso más sencillo. ✓ Los desarrolladores pueden reutilizar el código escrito para una plataforma mientras crean la aplicación para otra.
o Flutter	Este framework ha sido desarrollado por Google	<ul style="list-style-type: none"> o Ayuda a que los desarrolladores tengan una experiencia nativa a la hora de crear aplicaciones móviles esto lo hace resolviendo problemas que puedan surgir como el desplazamiento, la navegación, las fuentes, los iconos, etc. o Permite a los desarrolladores diseñar y desarrollar aplicaciones con facilidad ya que

reducen el tiempo total de desarrollo de la aplicación.

- o Ayuda a los desarrolladores a crear y diseñar interfaces de usuario intuitivas.
-

Fuente. Elaboración Propia

2.3 Xamarin

Según, Grosso (2021) es una herramienta muy útil y utilizada por los desarrolladores de aplicaciones móviles. Esta herramienta permite a los desarrolladores escribir su app en lenguaje C# y que el mismo código sea traducido para su ejecución en iOS, Android y Windows Phone.

2.3.1 Xamarin Forms

Xamarin.Forms es un marco de interfaz de usuario de código abierto además permite a los desarrolladores crear aplicaciones Xamarin.Android, Xamarin.iOS y Windows a partir de una única base de código compartida, se puede crear interfaces de usuario en XAML con código subyacente en C#. Estas interfaces se representan como controles nativos de alto rendimiento en cada plataforma (Microsoft, 2021). En la figura 1 se muestra la arquitectura general de una aplicación Xamarin multiplataforma.

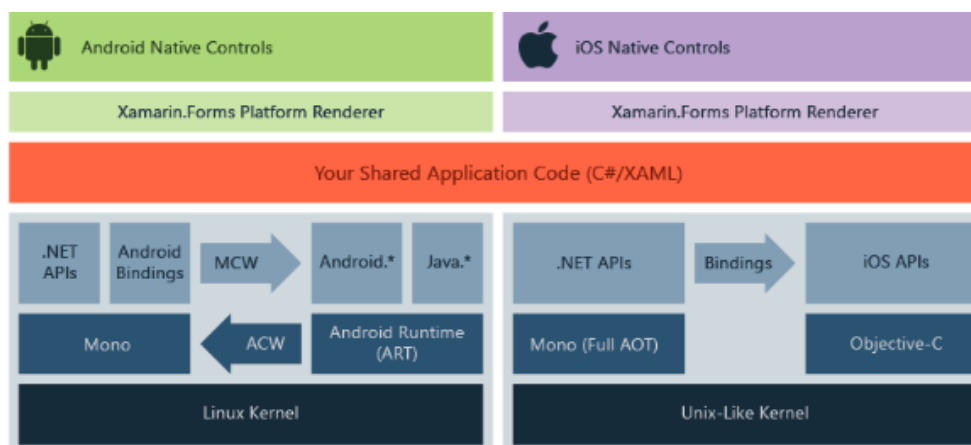


Figura 1 Arquitectura general de una aplicación Xamarin multiplataforma.

Fuente: Elaborado por (Microsoft, 2022).

Material Visual

Xamarin Forms Material Visual se utiliza para aplicar reglas de diseño de material a las aplicaciones de Xamarin.Forms. Utiliza la propiedad Visual para aplicar selectivamente representadores personalizados a la interfaz de usuario, lo que da como resultado una aplicación con un aspecto similar en iOS y Android (Microsoft, 2021). A continuación, en la tabla 3 se muestra controles que admite material visual.

Tabla 3 *Controles de Xamarin Forms Material Visual.*

Controles de Material Visual	Descripción
<ul style="list-style-type: none">• ActivityIndicator	Un control visual utilizado para indicar que algo está en curso.
<ul style="list-style-type: none">• Button	Un botón que reacciona a los eventos táctiles.
<ul style="list-style-type: none">• ProgressBar	Control View que muestra el progreso.
<ul style="list-style-type: none">• DatePicker	Vista que permite seleccionar fechas.
<ul style="list-style-type: none">• Editor	Un control que puede editar varias líneas de texto.
<ul style="list-style-type: none">• Entry	Un control que puede editar una sola línea de texto.

Fuente. Elaboración propia.

Funcionalidades adicionales

Xamarin.forms tiene un gran ecosistema de bibliotecas que agregan diversas funcionalidades a las aplicaciones (Microsoft, 2021). A continuación, se describen algunas funcionalidades:

Xamarin Esenciales

Es una biblioteca que ofrece API multiplataforma para el uso de características de dispositivos nativos. Al igual que Xamarin, Xamarin Essentials es una abstracción que simplifica el proceso de acceso a las utilidades nativas. La tabla 4 nos muestra algunos ejemplos de utilidades proporcionadas por Xamarin Essentials.

Tabla 4 *Ejemplos de utilidades proporcionadas por Xamarin Essentials.*

Utilidades Proporcionadas	Descripción
<ul style="list-style-type: none">• Información del dispositivo	<ul style="list-style-type: none">• Permite conocer acerca de la información sobre el dispositivo con facilidad.
<ul style="list-style-type: none">• Información de la aplicación	<ul style="list-style-type: none">• Sirve para averiguar información sobre la aplicación.
<ul style="list-style-type: none">• Acelerómetro	<ul style="list-style-type: none">• Ayuda a recuperar los datos de aceleración del dispositivo en el espacio tridimensional.
<ul style="list-style-type: none">• Marcador telefónico	<ul style="list-style-type: none">• Permite abrir el marcador telefónico.

- Conectividad
- Seguimiento de versiones
- Muestra la comprobación del estado de la conectividad y detecta los cambios.
- Se puede realizar un seguimiento de la versión de las aplicaciones y los números de compilación.

Fuente. Elaboración propia

Shell

Xamarin Forms Shell permite reducir la complejidad del desarrollo de aplicaciones móviles ya que proporciona las características fundamentales que la mayoría de las aplicaciones requieren (Microsoft, 2021). Algunos ejemplos de características proporcionadas por Shell incluyen:

- Experiencia de navegación común
- Esquema de navegación basado en URI
- Controlador de búsqueda integrado.

2.3.2 XAML de Xamarin Forms

XAML es un lenguaje basado en XML que fue creado por Microsoft para dar una alternativa a la programación de código y lograr inicializar objetos, crear instancias y ordenar los objetos por jeraquias padre-hijo.

Dentro de un archivo XAML, el desarrollador de Xamarin Forms puede definir interfaces de usuario utilizando todas los diseños, las vistas, y páginas de Xamarin Forms, así como clases personalizadas. Este archivo se puede compilar o incrustar en el ejecutable (Microsoft, 2021). En la tabla 5 se puede observar las ventajas sobre el código equivalente de XAML y sus desventajas en su mayoría relacionadas con limitaciones que son intrínsecas a los lenguajes de marcado.

Tabla 5 *Ventajas y Desventajas de XAML.*

Ventajas	Desventajas
<ul style="list-style-type: none"> • Es a menudo más conciso y legible que el código equivalente. • La jerarquía padre-hijo inherente a XML permite que XAML imite con mayor claridad visual la 	<ul style="list-style-type: none"> • Todos los controladores de eventos se deben definir en un archivo de código por lo tanto XAML no puede tener código. • XAML no puede contener bucles para el procesamiento repetitivo.

jerarquía padre-hijo de los objetos de la interfaz de usuario.

- Es fácil de escribir a mano para los programadores, pero también es moldeable y generado por herramientas de diseño visual.
- No puede crear instancias de clases que no definan a un constructor que este sin parámetros.

Fuente. Elaboración Propia

2.4 Metodologías Ágiles

Son aquellas que se adaptan a las formas de trabajo de las condiciones del proyecto, consiguiendo inmediatez y flexibilidad. Además, trabaja mucho más eficiente y rápido lo que provoca un aumento en la productividad, se ahorra costos y lo más importante tiempo. Recoge técnicas para la gestión de proyectos donde se utiliza un enfoque incremental e iterativo para gestionarlos (Toledo, 2020). La figura 2 muestra las metodologías ágiles más utilizadas.

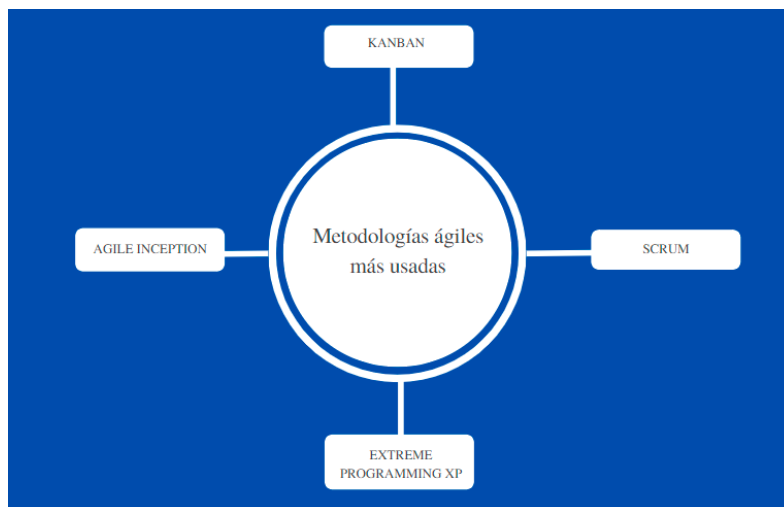


Figura 2 Metodologías ágiles más utilizadas.

Fuente: Elaboración propia.

2.4.1 Metodología Scrum

Scrum es un marco de trabajo diseñado de tal forma que logra la colaboración eficaz del equipo de trabajo, emplea un conjunto de reglas y se definen roles para generar una estructura de correcto funcionamiento (Molina et al., 2018). La Tabla 6 muestra los procesos de Scrum que abordan las actividades y el flujo específico de un proyecto Scrum. En total hay 19 procesos que se agrupan en cinco fases.

Tabla 6 Fases de la Metodología SCRUM.

Fases	Descripción
Inicio	<ol style="list-style-type: none">1. Crear la Visión del Producto2. Identify Scrum Master and Stakeholder(s)3. Formar el Equipo Scrum4. Desarrollo de Épica(s)5. Crear el Backlog Priorizado del producto6. Realizar la Planificación del lanzamiento.
Planificación	<ol style="list-style-type: none">7. Crear Historias de Usuarios8. Estimar Historias de los Usuarios9. Comprometer Historias de los Usuarios10. Crear Tareas11. Estimar Tareas12. Crear Sprint Backlog
Desarrollo	<ol style="list-style-type: none">13. Crear Entregables14. Realizar Daily Standup15. Refinar el Backlog Priorizado del Producto
Implementación	<ol style="list-style-type: none">16. Demostrar y validar el sprint17. Retrospectiva del sprint
Lanzamiento	<ol style="list-style-type: none">18. Enviar entregables19. Retrospectiva del proyecto

Fuente. Elaborado por (ScrumStudy, 2016)

2.5 Normas ISO

Son documentos que detallan los requerimientos que entidades pueden hacer uso para garantizar los productos y/o servicios que ofrecen. Hasta la fecha ISO, ha publicado un aproximado de 19 500 normas internacionales, su principal objetivo es asegurar la calidad deseada de los productos y/o servicios. Para las entidades estas normas sirven para minimizar costos y es posible la reducción de errores y lo mas importante favorecen al incremento de la productividad (Isotools, 2022).

2.5.1 ISO/IEC 25010

Es una norma que está centrada en las características de calidad que se deben tener en cuenta en el momento de evaluar las propiedades de un producto software terminado, provee un marco de referencia para medir la calidad del producto software y describe 8 características y 38 subcaracterísticas de calidad de producto software. Requisitos como eficiencia de desempeño, compatibilidad, funcionalidad, usabilidad, seguridad, fiabilidad, portabilidad y mantenibilidad son los que se encuentran en esta norma (Iso25000, 2021). 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 figura 3.

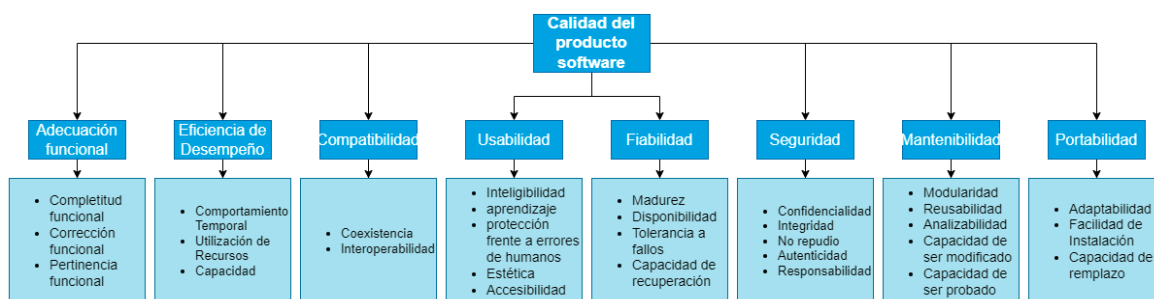


Figura 3 Características de calidad de las normas ISO/IEC 25010.

Fuente: Elaboración propia.

2.5.2 Eficiencia de desempeño

La Norma ISO/IEC 25010 representa el desempeño relativo a la cantidad de recursos usados bajo condiciones estables, la define como característica de calidad del producto de software que tiene en cuenta el rendimiento de un dispositivo en función de su comportamiento temporal, el uso de recursos y la capacidad o límites máximos de funcionamiento (Piñero González et al., 2021).

2.5.3 Utilización de recursos

Esta subcaracterística de eficiencia de desempeño se refiere al uso de recursos durante la ejecución de una tarea en condiciones preestablecidas. Se mide a través de cuatro métricas: 1) líneas de código (N/A), 2) uso de CPU, 3) uso de memoria y 4) utilización de dispositivos E/S. (Salazar et al., 2019)

2.6 JMeter

Es un software de código abierto, hecho en Java diseñada para cargar, probar el comportamiento funcional y medir el rendimiento. En principio fue creado para probar solo aplicaciones web, pero desde entonces se ha ido expandiendo a otras funciones de prueba (ApacheSoftwareFoundation, 2022).

CAPÍTULO III.

3. METODOLOGÍA

3.1 Tipo y diseño de la investigación

El tipo de investigación puede catalogarse como aplicada, tratándose de un diseño de investigación cuyo objetivo fue realizar una aplicación móvil multiplataforma para controlar los procesos que se llevan a cabo en un control de inventarios dentro de la organización comercial Nicoleth utilizando la plataforma Xamarin.

Este proyecto tiene un enfoque cuantitativo porque se analizó el rendimiento de la aplicación. El método utilizado para la investigación es inductivo, que consistió en el análisis de los problemas presentados por la microempresa al realizar un control de inventarios.

3.1.1 Según la fuente de investigación

Es una investigación bibliográfica porque se realizó una búsqueda de información, accediendo a revistas, libros, base de datos científicas, tesis, entre otras fuentes referentes al tema de investigación, permitiendo comprender conceptos relacionados a las aplicaciones móviles multiplataforma y las herramientas a utilizarse, permitiendo tener una visión más amplia del tema.

3.1.2 Según el objeto de estudio

Es una investigación de campo ya que se realizó un proceso de recolección y análisis de los requerimientos, además de un levantamiento de información sobre la clasificación y los inventarios.

3.1.3 Según el tipo de variable

Es una investigación cuantitativa porque se analiza el rendimiento y la utilización de los recursos del dispositivo móvil al momento de utilizar la aplicación, también es una investigación cualitativa de modo que se ha observado y estudiado otras investigaciones para crear un nuevo sistema.

3.2 Unidad de Análisis

En la unidad de análisis se estableció las 10 peticiones por segundo realizadas a la aplicación móvil multiplataforma en las distintas plataformas: Android y iOS para evaluar tiempo respuesta, tiempo espera, rendimiento, CPU y RAM esto se lo realizó con la herramienta JMeter.

3.3 Población y muestra

De acuerdo con el tipo de investigación planteada, se trata de una población infinita puesto que se obtendrán datos de diferentes mediciones utilizando JMeter.

3.4 Técnicas e instrumentos de recolección de datos

Las técnicas e instrumentos que se utilizaron en la investigación fueron entrevistas.

Entrevistas: Con el objetivo de recolectar información sobre el control de inventarios y los requerimientos para la aplicación móvil se ha realizado una entrevista a la dueña de la boutique Nicoleth. Ver anexo 3.

3.5 Procesamiento de datos

Una vez obtenida la información de las entrevistas se realizó un análisis para definir una orientación sobre el desarrollo de la aplicación móvil multiplataforma, teniendo en cuenta ciertos criterios para el correcto funcionamiento del mismo, en el que también se ha considerado algunas recomendaciones sobre la clasificación de los artículos. Posteriormente con la herramienta de evaluación JMeter y el aplicativo se realizó la medición de criterios de desempeño con enfoque la utilización de recursos establecidos por la norma ISO/IEC 25010.

3.6 Operacionalización de variables

Tabla 7 Operacionalización de variables

Problema	Tema	Objetivos	Variables	Indicadores
Control de inventario de la mercadería de la boutique Nicoleth.	Aplicación móvil multiplataforma para el control de inventario en la Boutique Nicoleth, utilizando la plataforma Xamarin.	<p>Objetivo general</p> <p>Desarrollar una aplicación móvil multiplataforma para el control de inventarios en la boutique Nicoleth utilizando la plataforma Xamarin.</p> <p>Objetivos específicos</p> <p>Analizar la plataforma Xamarin para el desarrollo de aplicaciones móviles multiplataforma.</p> <p>Implementar la aplicación móvil multiplataforma utilizando la plataforma Xamarin.</p> <p>Evaluar la eficiencia de desempeño de la aplicación móvil multiplataforma con la norma ISO/IEC 25010</p>	<p>Independiente:</p> <p>Aplicación móvil multiplataforma utilizando la plataforma Xamarin.</p> <p>Dependiente:</p> <p>Control de inventario en la boutique “Nicoleth”.</p>	<p>Independiente:</p> <p>Criterios de eficiencia de desempeño de la Norma ISO25010</p> <p>Utilización de recursos. Los tipos y cantidades de recursos usados bajo condiciones determinadas cuando el software lleva a cabo su función</p> <p>Dependiente:</p> <p>El número de Módulos de la aplicación.</p> <p>Módulo de artículos</p> <p>Módulo de entrada de inventario.</p> <p>Módulo de salida de inventario.</p>

utilizando el criterio utilización de recursos.

Módulo de clasificación de artículos.

Reportes generales y notificaciones.

Fuente. Elaboración propia

3.7 Desarrollo de la aplicación móvil multiplataforma

Para el desarrollo de la aplicación móvil multiplataforma se inició con la instalación y configuración de los recursos y herramientas de software, a continuación, en la figura 4 se puede observar Visual Studio y Xamarin.forms utilizado para el desarrollo de aplicaciones móviles multiplataforma.

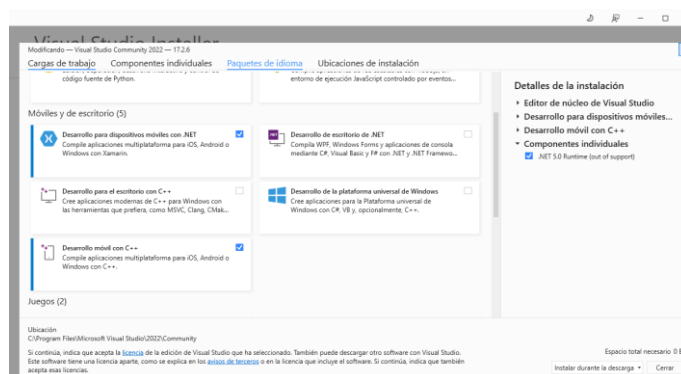


Figura 4 Entorno de instalación Visual Studio.

Fuente: Elaboración propia.

El proyecto de investigación se llevó a cabo utilizando la metodología ágil Scrum, aprovechando su enfoque dinámico para entregar un proyecto, centrándose en iteraciones rápidas, respondiendo a los clientes con entrega temprana y evitando resultados finales insatisfactorios, la misma que cuenta con cinco fases y a continuación se detallan las actividades desarrolladas en cada una:

3.7.1 Inicio

En esta fase se realizó un análisis de las necesidades a la entidad receptora por medio de reuniones, entrevistas y encuestas con la dueña de la microempresa, la cual manifestó la problemática con respecto al control de su inventario. Además, se estableció al equipo de trabajo.

A continuación, en la tabla 8 se puede evidenciar los roles para la ejecución del proyecto en base a la metodología Scrum.

Roles para la ejecución del proyecto

Tabla 8 Roles Scrum

Rol	Competencia	Descripción
Product Owner	Dra. Flor del Rocío Rojas	Representante del cliente dentro del equipo.
Scrum Master	Ing. Alexis Villegas	Guía al equipo en la metodología Scrum.
Scrum Team	Carlos Chacon, Adrián Estrada.	Encargados de diseñar y desarrollar la aplicación.

Fuente. Elaboración propia

3.7.2 Planificación

La fase de planificación describe los procesos relacionados que se realizarán durante el proceso de desarrollo del presente proyecto, por lo antes expuesto se detallan las siguientes tareas:

Product Backlog

Todo lo realizado por el Scrum Team se basa en el Product Backlog proporcionado por el Product Owner, esta es la única fuente de requisitos para la gestión del proyecto. El Product Backlog son historias de usuario de prioridad, que con el tiempo se vuelve más amplia y específica.

A continuación, en la tabla 9 se muestra la historia de usuario para la realización de la aplicación móvil multiplataforma.

Tabla 9 Historia de Usuario

Historia de Usuario			
ID	Como	Quiero	Para
HU001	Administrador	Iniciar Sesión	Identificarme como usuario Administrador.
HU002	Administrador	Registrar Usuarios	Tener un control de los usuarios que van a poder acceder a la aplicación móvil.
HU003	Administrador	Administrar Usuarios	Para modificar, agregar o eliminar un usuario.
HU004	Administrador	Administrar Marcas	Para modificar, agregar o eliminar una marca.

HU005	Administrador	Administrar Productos	Para modificar, agregar o eliminar productos. Administrar los accesos del sistema por tipo de usuario(Administrador, Usuario Normal).
HU006	Administrador	Otorgar permisos de la aplicación a usuario	
HU007	Administrador	Tener control de todos los módulos del sistema	Administrar todos los módulos.
HU008	Administrador	Realizar procesos de control de inventario: ingresos, egresos y movimientos.	Para llevar un control de inventario.
HU009	Administrador	Observar un reporte de los movimientos	Para verificar los ingresos y egresos de los productos.
HU010	Administrador	Observar notificaciones de los productos que están sin stock.	Para llevar un control de los productos sin stock y abastecer de nuevos productos.
HU011	Usuario	Iniciar Sesión	Identificarme como usuario Normal.
HU012	Usuario	Tener control de ciertos módulos del sistema	Para que solo el administrador tenga acceso total de todos los módulos.
HU013	Usuario	Actualizar Marcas	Para modificar algún dato de ser necesario.
HU014	Usuario	Actualizar Productos	Para agregar o modificar algún dato que sea necesario.
HU015	Usuario	Realizar procesos de control de inventario: ingresos, egresos y movimientos.	Para llevar un control de inventario.
HU016	Usuario	Observar un reporte de los movimientos	Para verificar los ingresos y egresos de los productos.
HU017	Usuario	Observar notificaciones de los productos que están sin stock.	Para llevar un control de los productos sin stock y abastecer de nuevos productos.

Fuente. Elaboración propia

Refinamiento del Product Backlog

El principal beneficio de esta actividad es mejorar el producto backlog inicial, por lo que el Scrum Team se compromete a:

- Conservar el Product Backlog ordenado.
- Rebajar o eliminar ítems de menos importancia.
- Ascender o añadir ítems que surgen y se consideren importantes.
- Separar ítems que se consideren pertinentes.
- Unir ítems que se consideren necesarios.
- Evaluar ítems

A continuación, en la tabla 10 se puede observar el product backlog refinado en el cual se eliminaron algunos ítems que no se consideran importantes o se encontraban redundantes, y a cada actividad se le agrego una prioridad. Y, además hay que considerar que los módulos establecidos en la tabla 7 operacionalización de variables que serán adecuados de acuerdo al diseño de la aplicación móvil.

Tabla 10 Refinamiento del Product Backlog

Historia de Usuario				
ID	Prioridad	Como	Quiero	Para
HU001	1	Administrador	Iniciar Sesión	Identificarme como usuario Administrador.
HU011	1	Usuario	Iniciar Sesión	Identificarme como usuario Normal.
HU002	2	Administrador	Registrar Usuarios	Tener un control de los usuarios que van a poder acceder a la aplicación móvil.
HU006	3	Administrador	Otorgar permisos de la aplicación al usuario.	Administrar los accesos del sistema por tipo de usuario(Administrador, Usuario Normal).
HU003	4	Administrador	Administrar Usuarios	Para modificar, agregar o eliminar un usuario.
HU004	5	Administrador	Administrar Marca	Para modificar, agregar o eliminar una marca.
HU012	5	Usuario	Actualizar Marca	Para agregar o modificar algún dato que sea necesario.
HU005	6	Administrador	Administrar Productos	Para modificar, agregar o eliminar productos.
HU013	6	Usuario	Actualizar Productos	Para agregar o modificar algún dato que sea necesario.

HU007	7	Administrador	Realizar procesos de control de inventario: ingresos, egresos y movimientos.	Para llevar un control de inventario.
H0014	7	Usuario	Realizar procesos de control de inventario: ingresos, egresos y movimientos.	Para llevar un control de inventario.
HU008	8	Administrador	Observar un reporte de los movimientos.	Para verificar los ingresos y egresos de los productos.
HU015	8	Usuario	Observar un reporte de los movimientos.	Para verificar los ingresos y egresos de los productos.
HU009	9	Administrador	Observar notificaciones de los productos que están sin stock.	Para llevar un control de los productos sin stock y abastecer de nuevos productos.
HU016	9	Usuario	Observar notificaciones de los productos que están sin stock.	Para llevar un control de los productos sin stock y abastecer de nuevos productos.

Fuente. Elaboración propia

Planificación del Sprint

Para empezar con el desarrollo de los Sprints, el Scrum Team mantuvo una reunión para elegir y entender el trabajo a realizar. El Product Owner y el Equipo de desarrollo dialogaron y llegaron a un acuerdo de ambas partes respecto a los trabajos que van a realizar y el plazo de tiempo que tendrán para entregar, esto tomando en cuenta las habilidades y conocimiento del equipo de desarrollo y la cantidad de tareas que tienen los ítems del product backlog.

Teniendo en cuenta lo mencionado se planteó los siguientes Sprints:

SPRINT 1

Para el primer Sprint se estimó 4 semanas de desarrollo para los siguientes ítems. En la tabla 11 se aprecia la Historia de Usuario para el Sprint 1.

Tabla 11 *Actividades del Sprint 1*

ID	Como	Quiero	Para
HU001	Administrador	Iniciar Sesión	Identificarme como usuario Administrador.
HU011	Usuario	Iniciar Sesión	Identificarme como usuario Normal.
HU002	Administrador	Registrar Usuarios	Tener un control de los usuarios que van a poder acceder a la aplicación móvil.
HU006	Administrador	Otorgar permisos de la aplicación al usuario.	Administrar los accesos del sistema por tipo de usuario(Administrador, Usuario Normal).

Fuente. Elaboración propia

SPRINT 2

Para el segundo Sprint se estimó 1 semana de desarrollo para los siguientes ítems. En la tabla 12 se puede observar la Historia de Usuario para el Sprint 2.

Tabla 12 *Actividades del Sprint 2*

ID	Como	Quiero	Para
HU004	Administrador	Administrar Marca	Para modificar, agregar o eliminar una marca.
HU012	Usuario	Actualizar Marca	Para agregar o modificar algún dato que sea necesario.
HU005	Administrador	Administrar Productos	Para modificar, agregar o eliminar productos.
HU013	Usuario	Actualizar Productos	Para agregar o modificar algún dato que sea necesario.

Fuente. Elaboración propia

SPRINT 3

Para el tercer Sprint se estimó 3 semanas de desarrollo para los siguientes ítems. En la tabla 13 se puede observar la Historia de Usuario para el Sprint 3.

Tabla 13 Actividades del Sprint 3

ID	Como	Quiero	Para
HU007	Administrador	Realizar procesos de control de inventario: ingresos, egresos y movimientos.	Para llevar un control de inventario.
H0014	Usuario	Realizar procesos de control de inventario: ingresos, egresos y movimientos.	Para llevar un control de inventario.
HU008	Administrador	Observar un reporte de los movimientos.	Para verificar los ingresos y egresos de los productos.
HU015	Usuario	Observar un reporte de los movimientos.	Para verificar los ingresos y egresos de los productos.
HU009	Administrador	Observar notificaciones de los productos que están sin stock.	Para llevar un control de los productos sin stock y abastecer de nuevos productos.
HU016	Usuario	Observar notificaciones de los productos que están sin stock.	Para llevar un control de los productos sin stock y abastecer de nuevos productos.

Fuente. Elaboración propia

A continuación, en la figura 5 se puede observar la adecuación de los módulos descritos inicialmente en la tabla 7 de manera que estos se implementaran de acuerdo con los diseños conceptuales y estructuración esquemática desarrollados más adelante para la aplicación móvil multiplataforma.

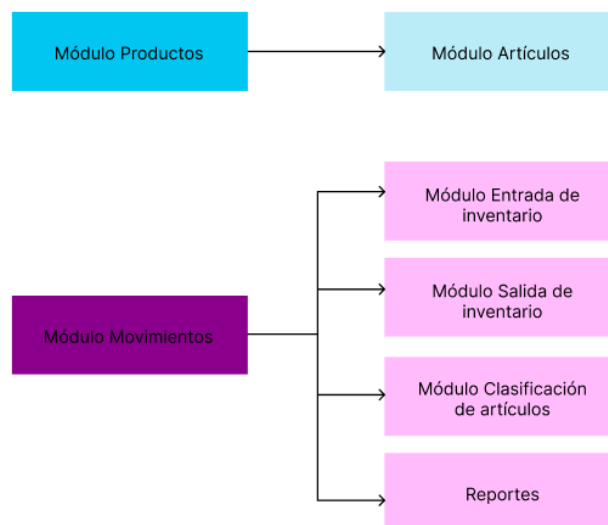


Figura 5 Adecuación de los módulos de la aplicación móvil.

Fuente: Elaboración propia.

Incremento del producto

Una vez finalizado cada uno de los sprints el equipo de desarrollo mostró avances de la aplicación móvil al Product Owner. Finalmente, luego de haber sido verificado y aprobado por el equipo de trabajo en conjunto con el Product Owner se procedió a la fase de producción.

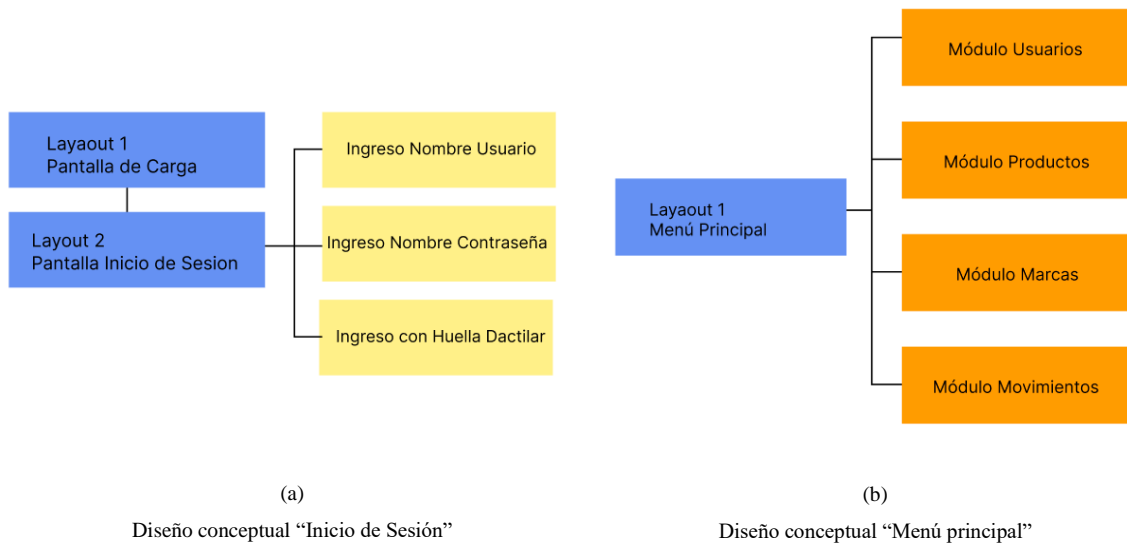
3.7.3 Desarrollo

La fase de desarrollo consistió en la ejecución de las tareas y actividades para crear el producto, por lo consiguiente se detallan las siguientes tareas.

3.7.3.1 Diseño

Diseño Conceptual

En esta fase se ha realizado un diseño conceptual y la estructuración esquemática de los módulos de la aplicación que se detallan a continuación en las figuras 6.



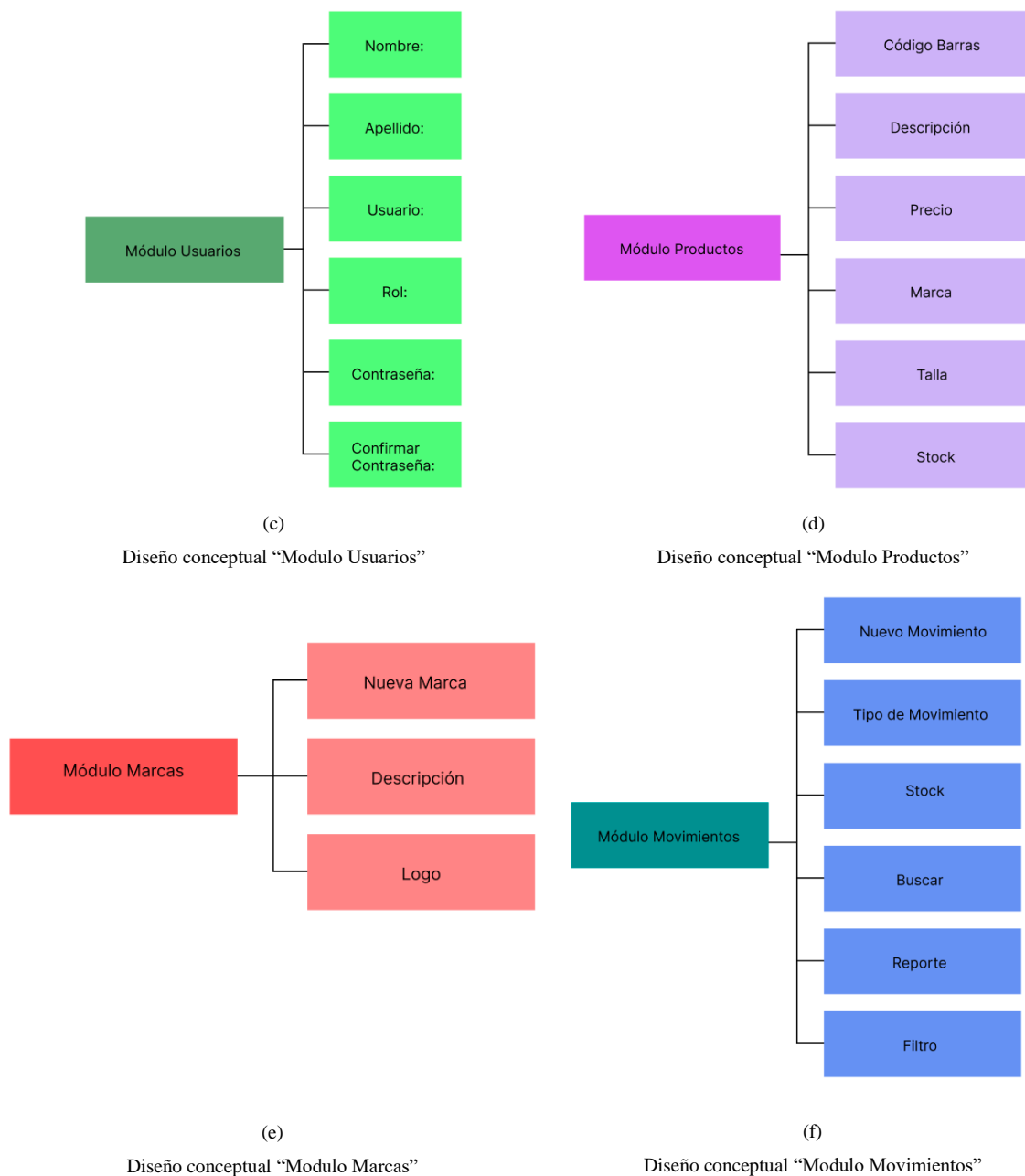


Figura 6 *Diseño conceptual de la aplicación*

Fuente: Elaboración propia.

Diseño de Interfaz

Después de completar el "Diseño conceptual" de cada módulo, se realizó la fase de diseño de la interfaz, la estructura esquemática de la navegación en la aplicación, la estructura específica de las vistas, la navegabilidad, la interfaz de usuario y sus interacciones en los distintos módulos que contiene la aplicación. A continuación, en la figura 7 se puede observar el diseño de la interfaz de nuestra aplicación móvil.



Figura 7 *Diseño de Interfaz de la aplicación.*

Fuente: Elaboración propia.

3.7.4 Implementación

Esta fase consistió en la implementación de los diseños elaborados en la fase anterior para Android y iOS, se detallan las siguientes tareas.

Implementación de los modelos que se crearon previamente teniendo en cuenta que deber ser *responsive* es decir que permita al usuario una correcta visualización de la aplicación móvil en diferentes dispositivos móviles. En la figura 8 se puede observar la implementación del diseño que fue elaborado.



Figura 8 Principales pantallas de la Implementación del diseño elaborado.

Fuente: Elaboración propia.

Implementación de la Base de Datos y Apis

Se diseño la base de datos que fue hecha en el sistema gestor de base de datos SQL server. A continuación, en la figura 9 se puede observar el modelo relacional.

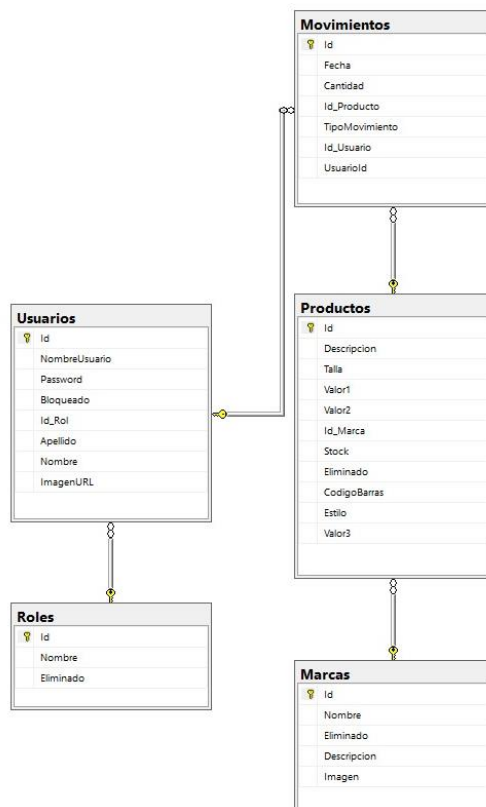


Figura 9 Diagrama base de datos.

Fuente: Elaboración propia.

Implementación Multiplataforma

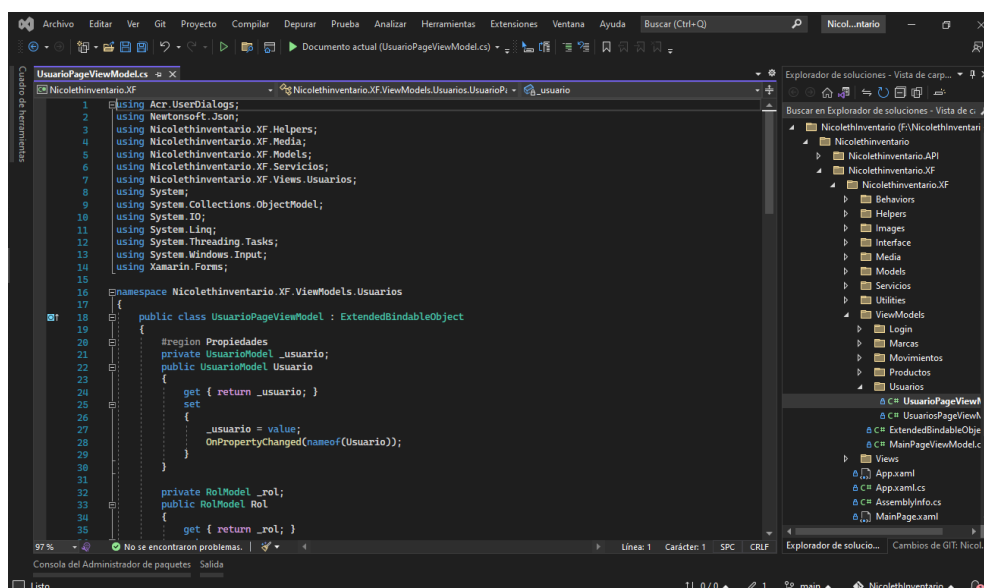
Para verificar que la aplicación móvil es multiplataforma se desarrolló la aplicación para dos de los sistemas operativos más utilizados que son Android y iOS. Esto se logra gracias a xamarin.forms ya que nos permite trabajar en un entorno de desarrollo integrado con un solo lenguaje. Al crear una aplicación en Visual Studio con Xamarin se obtiene 3 carpetas iniciales, la primera carpeta es compartida para Android y para iOS esta es la más importante ya que aquí se tiene nuestro código compartido que se compila en ambos sistemas operativos. Además, se muestran 2 carpetas más, las cuales son generadas de manera automática por Visual Studio ya que son necesarias para que funcionen en Android y iOS, en estas carpetas se agrega imágenes, fuentes, clases customizadas, etc.

La aplicación móvil multiplataforma se basa en el “Modelo – Vista - ViewModel” (MVVM), esto debido a que este modelo sugiere separar la lógica de presentación de datos de la parte principal de la lógica de la aplicación.

Este modelo funciona mediante capas separadas, en las cuales se realiza la abstracción de las fuentes de datos y además esta capa funciona en conjunto con la capa ViewModel para obtener y guardar datos. La capa vista tiene como propósito informar a la capa ViewModel acerca de la acción del usuario y no cuenta con ningún tipo de lógica de aplicación. La capa

ViewModel sirve como nexa entre la capa modelo y la vista, además manifiesta los flujos de datos que son relevantes para la capa vista.

Para la aplicación el ViewModels está dividido por pantallas (Login, Marcas, Movimientos, Productos y Usuarios), en las views se encuentran las vistas de estas pantallas, en cada ViewModel está la lógica la cual se conecta con models y en este caso con servicios siempre y cuando sea necesario. A continuación, en la figura 10 se puede observar el código utilizado en el ViewModel de la pantalla Usuarios.



```
1 using Acra.UserDialogs;
2 using Newtonsoft.Json;
3 using NicoleshInventario.XF.Helpers;
4 using NicoleshInventario.XF.Models;
5 using NicoleshInventario.XF.Services;
6 using NicoleshInventario.XF.Views.Usuarios;
7 using System;
8 using System.Collections.ObjectModel;
9 using System.IO;
10 using System.Linq;
11 using System.Threading.Tasks;
12 using System.Windows.Input;
13 using Xamarin.Forms;
14
15 namespace NicoleshInventario.XF.ViewModels.Usuarios
16 {
17     public class UsuarioPageViewModel : ExtendedBindableObject
18     {
19         #region Propiedades
20         private UsuarioModel _usuario;
21         public UsuarioModel Usuario
22         {
23             get { return _usuario; }
24             set
25             {
26                 _usuario = value;
27                 OnPropertyChanged(nameof(Usuario));
28             }
29         }
30
31         private RolModel _rol;
32         public RolModel Rol
33         {
34             get { return _rol; }
35         }
36     }
37 }
```

Figura 10 Código de implementación que servirá para ambos sistemas operativos

Fuente: Elaboración propia.

3.7.5 Lanzamiento

Con la aplicación móvil culminada se procedió a probar en ambos sistemas operativos en el caso de Android se obtuvo el apk de la aplicación (acrónimo de Android Application Package es la forma en la que está empaquetada una aplicación de Android que contiene, como instalable todos los archivos necesarios para instalar una aplicación en nuestro dispositivo basado en Android.) Posteriormente con el apk se realizó la instalación de la aplicación en un dispositivo Android.

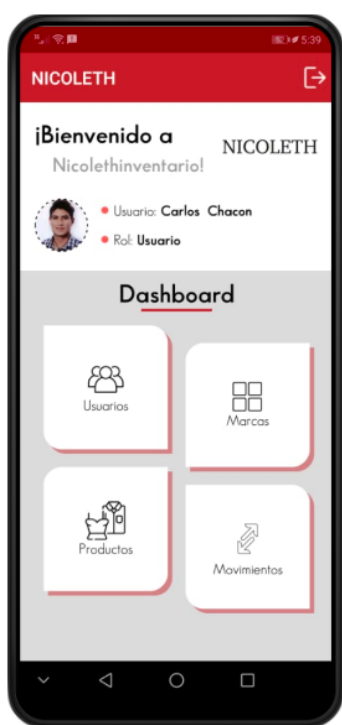
Para dispositivos iOS el proceso a realizar es diferente, lo primero a tener en cuenta es que se necesita una cuenta Apple Developer Program para tener acceso a las opciones disponibles destinadas a desarrolladores, en este caso se utilizó TestFlight.

Lo primero que se realiza es, dentro de la página principal de tiendas seleccionar App Store Conectar Usuarios, a continuación, se selecciona en Publicar en TestFlight, posteriormente en el primer paso del asistente se carga el archivo .ipa (siglas de iOS AppStore Package que es el formato que utilizan las aplicaciones de Apple). Para poder continuar con el proceso escribimos la versión de nuestra aplicación y se publica para que el archivo. ipa se entregue a la App Store Connect para su evaluación. Cuando la App Center complete la entrega de la aplicación a la App Store Connect el estado de la aplicación se encuentra como publicada y está lista para descargarse a través de TestFlight.

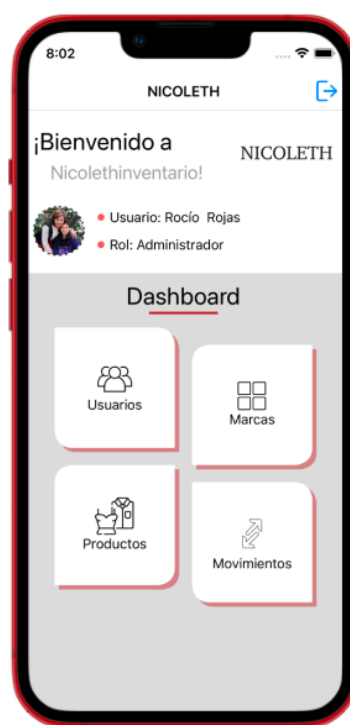
CAPÍTULO IV.

4. RESULTADOS Y DISCUSIÓN

Se planteó como uno de los objetivos de esta investigación el desarrollo de una aplicación móvil multiplataforma para el control de inventario en la boutique Nicoleth de la ciudad de Riobamba, la cual se la desarrollo con la herramienta xamarin.forms para dos de los sistemas operativos más utilizados Android y iOS. Además, se implementaron los siguientes módulos: Módulo Usuarios, Módulo Productos, Módulos Marcas y Módulo Movimientos para que los usuarios puedan realizar el control del inventario de la boutique haciendo uso de estos módulos. A continuación, en la figura 11 se muestran las principales pantallas de la aplicación. Para observar todas las pantallas ver anexo 1.



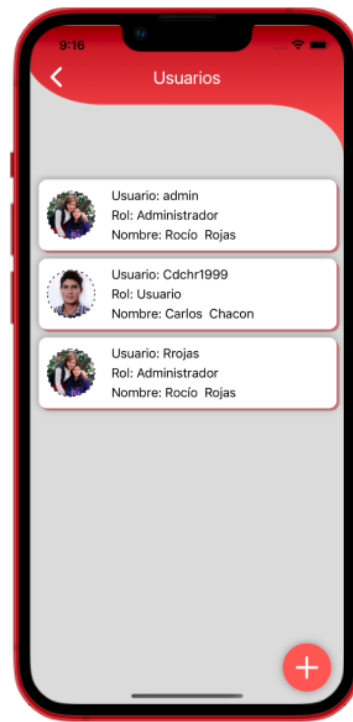
Menú Principal (Android)



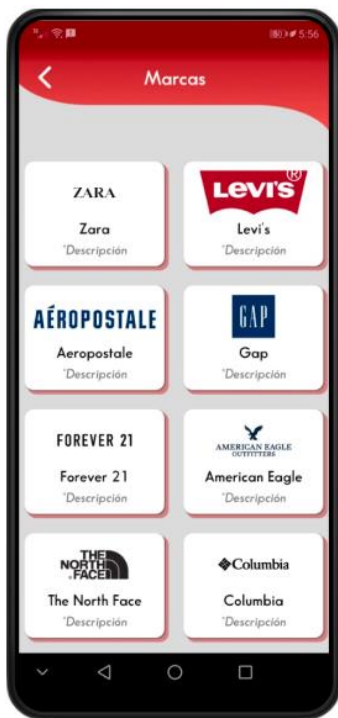
Menú Principal (iOS)



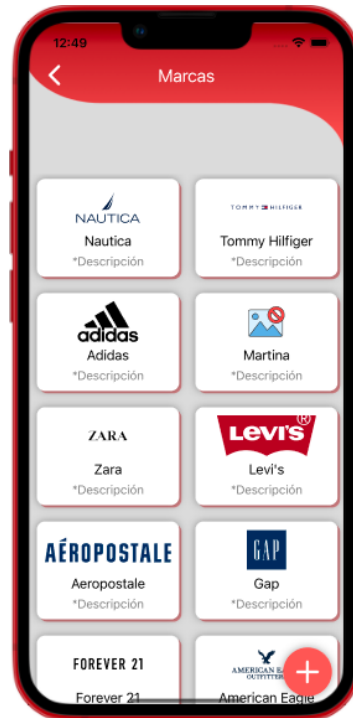
Módulo Usuarios (Android)



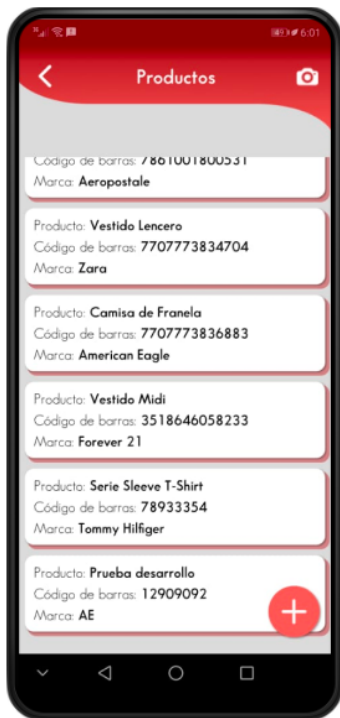
Módulo Usuarios (iOS)



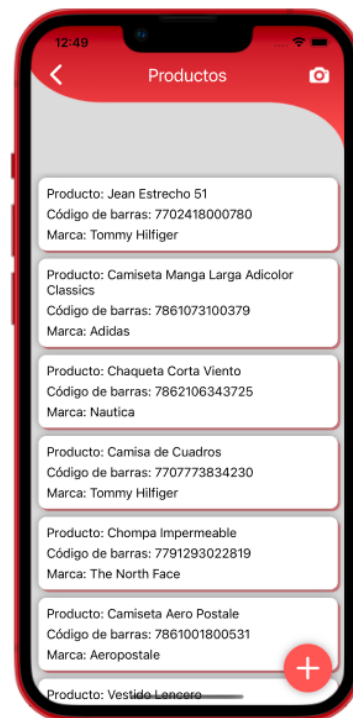
Módulo Marcas (Android)



Módulo Marcas (iOS)



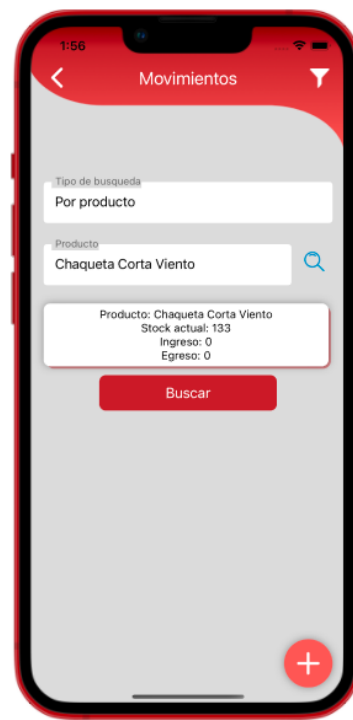
Módulo Productos (Android)



Módulo Productos (iOS)



Módulo Movimientos (Android)



Módulo Movimientos (iOS)

Figura 11 Principales pantallas de la aplicación móvil Android y iOS.

Fuente: Elaboración propia.

Para la evaluación de la eficiencia de desempeño de aplicación móvil multiplataforma, se utilizó la norma ISO/IEC 25010 la cual establece un modelo de calidad del producto de software, para este caso se destinó evaluar la eficiencia de desempeño y se utilizó las siguientes subcaracterísticas: Comportamiento temporal y Utilización de recursos para obtener resultados más reales. A continuación, en la tabla 14 se observa los indicadores a evaluar según la norma.

Tabla 14 *Indicadores a evaluar.*

Sub Característica	Indicadores
Comportamiento Temporal	<ul style="list-style-type: none"> • Tiempo de Respuesta • Tiempo de Espera • Rendimiento
Utilización de recursos	<ul style="list-style-type: none"> ▪ Uso de CPU ▪ Uso de Memoria RAM

Fuente. Elaboración propia

Para obtener los valores de los indicadores la norma ISO/IEC 25010 sugiere el uso de las siguientes ecuaciones para obtener dichos valores. Véase en la tabla 15.

Tabla 15 *Ecuaciones sugeridas por la norma para obtener los valores de los indicadores.*

Comportamiento Temporal	Utilización de recursos
<p>Indicador 1: Tiempo de Respuesta</p> <p>Entradas:</p> <p>A= Tiempo envió petición .</p> <p>B= Tiempo en recibir la primera respuesta.</p> <p>Ecuación:</p> <p>X=B-A</p>	<p>Indicador 3: Rendimiento</p> <p>Entradas:</p> <p>A= Número tareas completadas</p> <p>T= Intervalo de tiempo donde: $T > 0$.</p> <p>Ecuación:</p> <p>X=A/T</p>
<p>Indicador 1: Tiempo de Respuesta</p> <p>Entradas:</p> <p>A= Tiempo envió petición .</p> <p>B= Tiempo en recibir la primera respuesta.</p> <p>Ecuación:</p> <p>X=B-A</p>	<p>Indicador 1: Uso de CPU</p> <p>Entradas:</p> <p>A= Cantidad de CPU usado para realizar una tarea</p> <p>Ecuación:</p> <p>X=A</p>
<p>Indicador 2: Tiempo de espera</p> <p>Entradas:</p> <p>A= Tiempo de inicio de un trabajo</p> <p>B= Tiempo en completar un trabajo</p> <p>Ecuación:</p> <p>X=B-A</p>	<p>Indicador 2: Uso de Memoria RAM</p> <p>Entradas:</p> <p>A= Cantidad de memoria usado para realizar una tarea</p> <p>Ecuación:</p> <p>X=A</p>

Fuente. Elaboración propia

Para poder continuar con el proceso de evaluación, se planteó designar un 20% de ponderación en base a las necesidades de la aplicación, con el objetivo de darle la misma importancia a cada indicador y lograr obtener una aplicación eficiente que cumpla con los estándares de la norma ISO/IEC 25010. Véase la tabla 16.

Tabla 16 *Porcentajes de ponderación*

Sub Característica	Métrica	Porcentaje	Total
Comportamiento Temporal	• Tiempo de Respuesta	20%	60%
	• Tiempo de Espera	20%	
	• Rendimiento	20%	
Utilización de recursos	▪ Uso de CPU	20%	40%
	▪ Uso de Memoria RAM	20%	

Fuente. Elaboración propia

Dispositivos utilizados para la medición de las métricas

Tabla 17 *Descripción de los dispositivos usados.*

	Android	iOS
Dispositivo Móvil	Vivo	iPhone 7 plus
Procesador	Snapdragon 665	Apple A10 Fusion
Versión Sistema Operativo	Android 10	iOS 10
RAM	8 GB	3 GB LPDDR4
Almacenamiento	128 GB + microSD	128 GB

Fuente. Elaboración propia

Para realizar las pruebas se utilizó la herramienta Apache JMeter, en la cual se realizó una simulación del uso de la aplicación móvil multiplataforma en los dispositivos mencionados anteriormente, con un total de 10 peticiones simuladas en un 1 segundo para las distintas peticiones.

Posteriormente con los resultados obtenidos se realizó un promedio global de cada indicador, para la respectiva utilización en la medición por medio de la matriz de calidad como indica la norma ISO/IEC 25010. Para ver la configuración de JMeter ver anexo 5.

Análisis del dispositivo Android

En la tabla 18 se puede observar los resultados obtenidos al realizar la simulación en el software JMeter, en el dispositivo Android.

Tabla 18 *Medición indicadores dispositivo Android.*

N.º	Comportamiento Temporal		Utilización de recursos	
	Tiempo Respuesta (s)	Tiempo Espera (s)	CPU%	RAM%
1	0,01417	0,18858	35	44
2	0,02533	0,59192	33	44
3	0,01225	0,16567	33	44
4	0,01325	0,17333	34	44
5	0,0125	0,16633	33	43
6	0,0125	0,17292	33	43
7	0,01242	0,17283	34	45
8	0,02158	0,18067	33	43
9	0,0125	0,16650	33	44
10	0,0125	0,17225	34	43

Fuente. Elaboración propia

Análisis del dispositivo iOS

En la tabla 19 se puede observar los resultados obtenidos al realizar la simulación en el software JMeter, en el dispositivo iOS.

Tabla 19 Medición indicadores dispositivo iOS.

N.º	Comportamiento Temporal		Utilización de recursos	
	Tiempo Respuesta (s)	Tiempo Espera (s)	CPU%	RAM%
1	0,0115	0,18858	43	30
2	0,01058	0,59192	43	32
3	0,01008	0,16567	43	33
4	0,01017	0,17333	42	32
5	0,01058	0,16633	43	32
6	0,01083	0,17292	44	32
7	0,01042	0,17283	44	31
8	0,01042	0,18067	43	32
9	0,01058	0,16650	43	31
10	0,01058	0,17225	44	31

Fuente. Elaboración propia

Evaluación de las métricas dispositivo Android

En la tabla 20 se puede observar los resultados obtenidos en la medición del dispositivo Android obteniendo un resultado de 8,43/10.

Tabla 20 Resultados dispositivo Android.

Métrica	Valor Deseado	Peor Caso	Resultado	Valor Métrica(10)	Resultado Final	Subtotal	Total
Tiempo Respuesta	1s	>10s	0,0149	9,98	1,99		
Tiempo Espera	1s	>10s	0,2151	9,78	1,96	5,95	8,43
Rendimiento	10p	0p	10 p	10	2		
CPU	30%	100%	33%	6.7	1,34	2,48	
RAM	30%	100%	43%	5.7	1,14		

Fuente. Elaboración propia

Evaluación de las métricas dispositivo iOS

En la tabla 21 se puede observar los resultados obtenidos en la medición del dispositivo iOS obteniendo un resultado de 8,48/10.

Tabla 21 Resultados dispositivo iOS.

Métrica	Valor Deseado	Peor Caso	Resultado	Valor Métrica(10)	Resultado Final	Subtotal	Total
Tiempo Respuesta	1s	>10s	0,01058	9,98	1,99		
Tiempo Espera	1s	>10s	0,23938	9,76	1,95	5,94	8,48
Rendimiento	10p	0p	10 p	10	2		
CPU	30%	100%	43%	5,7	1,14		
RAM	30%	100%	31%	6,9	1,38	2,54	

Fuente. Elaboración propia

4.1 Análisis de los resultados obtenidos

A continuación, en la tabla 22 se muestran los resultados obtenidos en los dispositivos utilizados para la medición de las métricas de la norma ISO/IEC 25010 acerca de las características comportamiento temporal y utilización de recursos.

Tabla 22 Resultados obtenidos de la evaluación de los dispositivos Android y iOS.

Métricas	Dispositivo Android	Dispositivo iOS
Tiempo de Respuesta	0.0149 s	0.01058 s
Tiempo de Espera	0.2151 s	0.23938 s
Rendimiento	10 p	10 p
CPU	33%	43%
RAM	43%	31%

Fuente. Elaboración propia

4.1.1 Tiempo de Respuesta

En la figura 12 se muestra el tiempo promedio de la subcaracterística tiempo de respuesta que fue medida en un dispositivo Android y en iOS, este nos muestra que existe una pequeña diferencia del tiempo de respuesta respecto a cada dispositivo, por lo cual el tiempo de respuesta en un dispositivo iOS es más eficiente sin embargo en ambos dispositivos el tiempo de respuesta es adecuado.

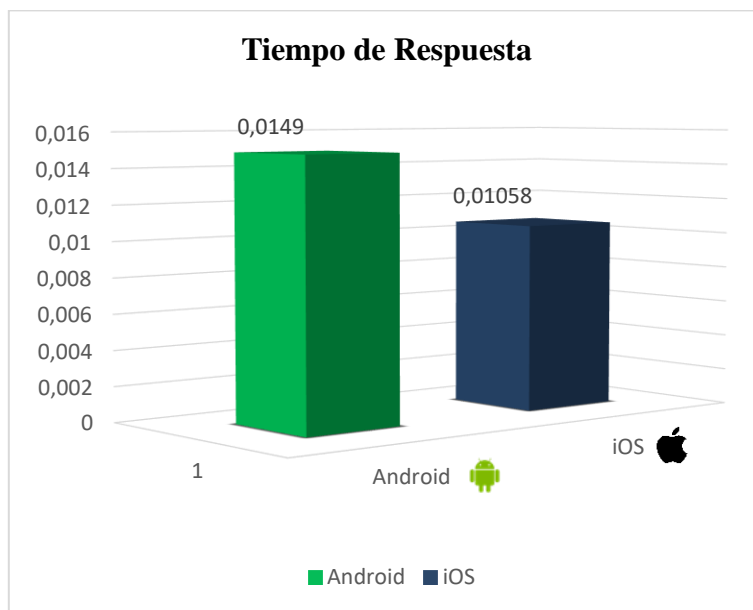


Figura 12 Resultado del tiempo de respuesta.

Fuente: Elaboración propia.

4.1.2 Tiempo de Espera

En la figura 13 se presenta el resultado sobre el tiempo promedio del tiempo de espera que fue medida en un dispositivo Android y en iOS, además se puede ver que existe cierta diferencia del tiempo de espera respecto a cada dispositivo, por lo cual el tiempo de respuesta en un dispositivo Android es más eficaz sin embargo el tiempo en ambos dispositivos es adecuado.

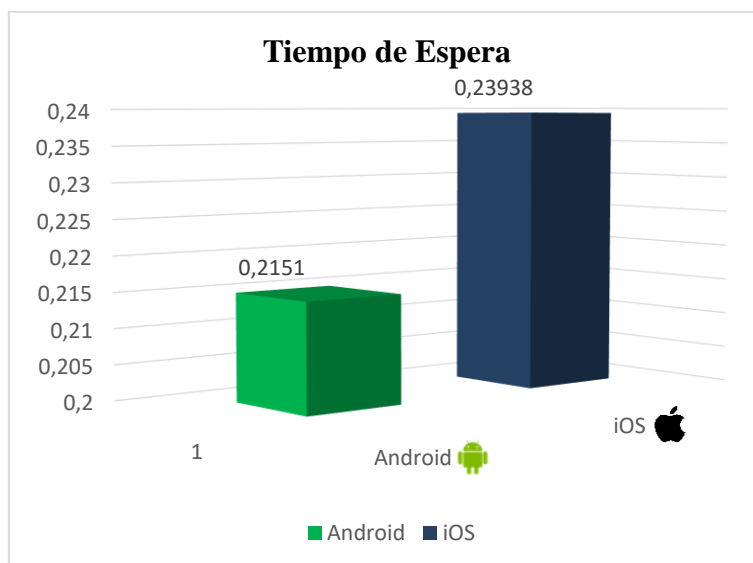


Figura 13 Resultado del tiempo de espera.

Fuente: Elaboración propia.

4.1.3 CPU

Se observa en la figura 14 el resultado del porcentaje promedio del uso del CPU en el dispositivo Android y en iOS, además se puede apreciar una ligera diferencia en el porcentaje de uso en cada dispositivo, en Android se evidencia que el uso del CPU es más eficiente, sin embargo, el uso del CPU en ambos dispositivos es adecuado.

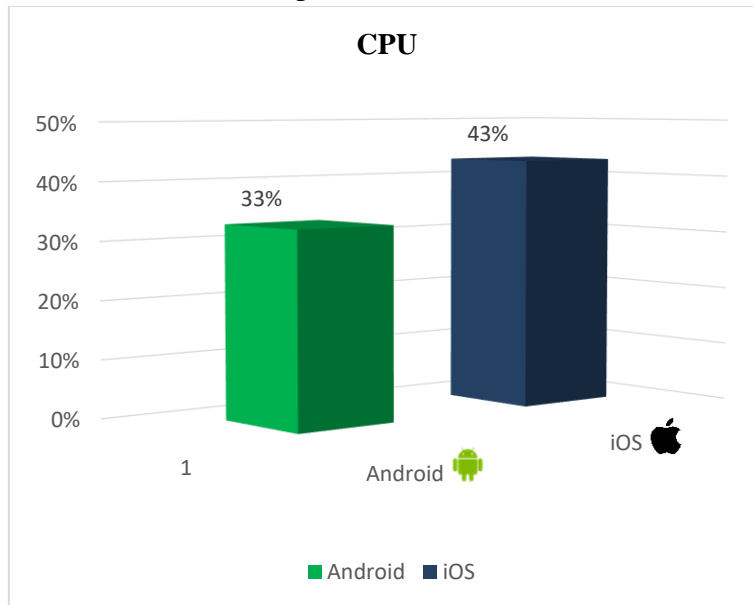


Figura 14 Resultado del uso del CPU.

Fuente: Elaboración propia.

4.1.4 RAM

En la figura 15 se muestra el porcentaje promedio del uso de la memoria RAM en el dispositivo Android y en iOS, se puede observar una ligera diferencia en el porcentaje de uso en cada dispositivo, en el dispositivo Android se evidencia que el uso del CPU es más eficiente, sin embargo, el uso del CPU en ambos dispositivos es adecuado.

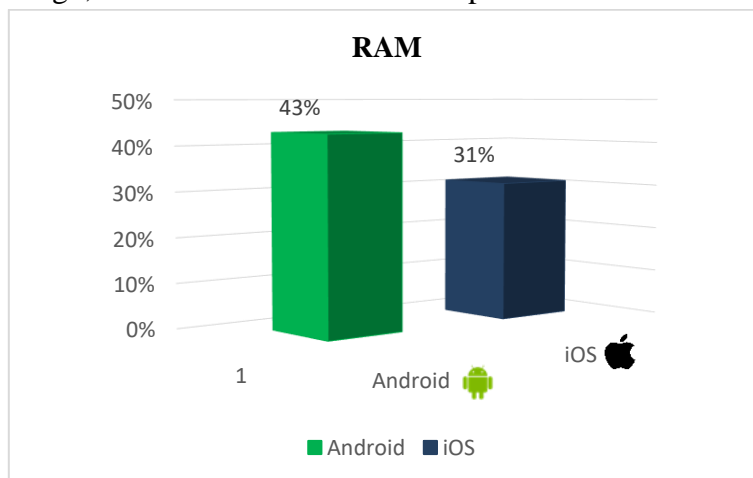


Figura 15 Resultado del uso de la RAM.

Fuente: Elaboración propia.

4.1.5 Eficiencia de Desempeño

En la figura 16 se puede apreciar el resultado que obtuvo nuestra aplicación en los diferentes dispositivos evaluados, con respecto a la matriz de calidad que la norma ISO/IEC 25010 presenta. Obteniendo como resultado 8,43/10 para el dispositivo Android y 8,48/10 para el dispositivo iOS demostrando así que la aplicación desarrollada funciona correctamente y es eficaz en ambos dispositivos y demostrando que se logró que la aplicación sea multiplataforma.

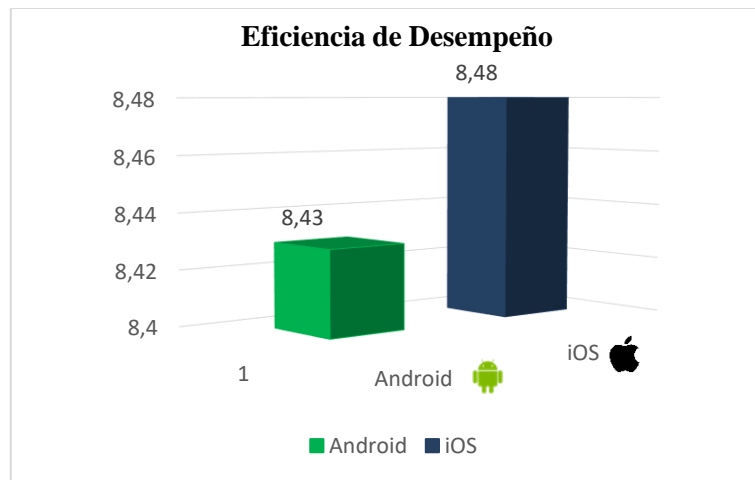


Figura 16 Resultado de la eficiencia de desempeño en los dispositivos evaluados.

Fuente: Elaboración propia.

CAPÍTULO V.

5. CONCLUSIONES y RECOMENDACIONES

CONCLUSIONES

- El uso de Xamarin para la creación de aplicaciones móviles multiplataforma es eficaz, permitiendo reducir el tiempo de desarrollo de la aplicación, además del uso del modelo MVVM nos permitió trabajar de manera simultánea e independiente en sus componentes durante el proceso de desarrollo, mientras uno se centró en la vista, el otro puede trabajar en el modelo de vista y los componentes del modelo.
- Gracias al uso de la metodología ágil SCRUM se pudo adaptar de manera natural a las condiciones cambiantes y los requisitos de la beneficiaria logrando reducir el tiempo de desarrollo del proyecto y obteniendo excelentes resultados.
- El uso del software JMeter facilita la obtención de los valores necesarios para la medición según los indicadores de la norma ISO/IEC 25010, y gracias a esto se determinó que el rendimiento de nuestra aplicación es óptimo según el modelo

calidad de la norma ISO obteniendo resultados de 8,43/10 para el dispositivo Android y 8,48/10.

RECOMENDACIONES

- La aplicación de una metodología ágil para el desarrollo de software es importante para mayor facilidad de implementación ya que permite mantener una forma innovadora de trabajar y organizar flujos, dividir proyectos en partes, lo que permite una rápida adaptación, finalización y resolución de etapas en un corto periodo de tiempo.
- Para acceder a todos los servicios que ofrece iOS y debido a la seguridad que este sistema operativo cuenta se recomienda tener una cuenta de desarrollador para poder probar la aplicación en iOS.
- Se recomienda utilizar el software JMeter ya que su uso es sencillo, además de contar con una variedad de documentación y tutoriales para el aprendizaje, permite realizar pruebas de rendimiento en tiempo real y entrega valores precisos, y sobre todo es gratis.
- El presente proyecto de investigación maneja un escenario en el cual se utiliza diferentes dispositivos cada uno con su sistema operativo diferente, por lo tanto, se aspira que este trabajo motive a los estudiantes optar por el desarrollo multiplataforma, teniendo en cuenta que se puede implementar en distintos campos laborales.

BIBLIOGRAFÍA

- Báez, M., Borrego, Á., Cordero, J., Cruz, L., González, M., Hernández, F., . . . Zapata, Á. (2019). *Introducción a Android*. E.M.E.
- Chatterjee, N., Chakraborty, S., Decosta, A., & Nath, D. (2018). Real-time Communication Application Based on Android Using Google Firebase. *International Journal of Advance Research in Computer Science and Management Studies*, 6(4).
- Delía, L. N. (2017). *Desarrollo de Aplicaciones [Tesis de Ingeniería, Universidad Nacional de la Plata]*. Repositorio Institucional. https://digital.cic.gba.gob.ar/bitstream/handle/11746/6601/11746_6601.pdf?sequence=1&isAllowed=y
- Ferrer Sanz, I. (2021). *Desarrollo de una aplicación móvil multiplataforma [Tesis de Ingeniería, Escola Tècnica Superior d'Enginyeria Informàtica]*. Repositorio Institucional.
- García, I., & Cáceres, M. (2019). Las generaciones digitales y las aplicaciones móviles como refuerzo educativo. *Revista Metropolitana de Ciencias Aplicadas*, 2(1), 25-31.
- González, N., Ignacio, M., & Hernández, N. (2020). Aproximación al Análisis de Benchmark sobre Asistentes Virtuales. *Interconectando Saberes*(9).
- Grupo El Comercio. (2019). Los servicios de las aplicaciones se expanden en Ecuador. *Líderes*. <https://www.revistalideres.ec/lideres/servicios-aplicaciones-expanden-ecuador.html#:~:text=Solicitar%20un%20taxi%2C%20pedir%20comida,un%20teléfono%20o%20una%20tableta>.
- Ionos Cloud S.L.U. (10 de Mayo de 2019). *ionos*. ionos: <https://www.ionos.es/startupguide/gestion/software-de-inventarios/>
- Iso25000. (2021). *iso25000*. iso25000: <https://iso25000.com/index.php/normas-iso-25000/iso-25010>
- Isotools. (2022). *isotools*. isotools: <https://www.isotools.org>
- Khawas, C., & Shah, P. (2018). Application of Firebase in Android App Development-A. *International Journal of Computer Applications*, 179, 48-53.
- Microsoft. (18 de marzo de 2022). *microsoftdocs*. microsoftdocs: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>

- Molina, B., Vite, H., & Dávila, J. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espirales*, 2(17), 114-121.
- Oracle. (2022). *oracle.com*. oracle.com: <https://www.oracle.com/mx/database/what-is-database/#link5>
- Pérez, V., González, M., Guilabert, M., & Joaquín, J. (2019). Mobile Apps for Increasing Treatment Adherence: Systematic. *Journal of medical internet research*, 21. <https://doi.org/10.2196/12505>
- Rocio, G. (2023). *¿Qué es iOS? Todo sobre el sistema operativo de Apple. ¿Qué es iOS? Todo sobre el sistema operativo de Apple:* <https://www.adslzone.net/reportajes/software/que-es-ios/>
- Rodriguez, J. (1 de Septiembre de 2021). *hubspot*. hubspot: <https://blog.hubspot.es/sales/que-es-control-de-inventarios>
- Salazar, F., Manosalvas, P., Rodríguez, N., & Landeta, P. (2019). Análisis de la eficiencia de desempeño en aplicaciones de Realidad Aumentada utilizando la normativa ISO/IEC/25010. *Revista Ibérica de Sistemas e Tecnologías de Informação*, 22, 256-267.
- ScrumStudy. (2016). *Una guía para el cuerpo de conocimiento de Scrum (Guía SBOK™)*. ScrumStudy.
- SurveyMonkey. (2022). *surveymonkey*. surveymonkey: <https://es.surveymonkey.com/mp/sample-size-calculator/>
- techopedia. (2022). *theastrologypage*. theastrologypage: <https://es.theastrologypage.com/mobile-operating-system>
- Tecnológico, C. (21 de Junio de 2020). *Qué es Antutu?* chiptecnologico: <https://chiptecnologico.com/que-es-antutu/>

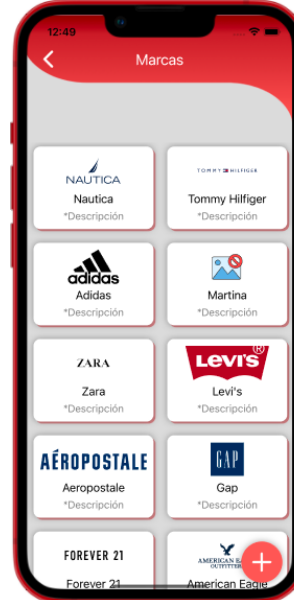
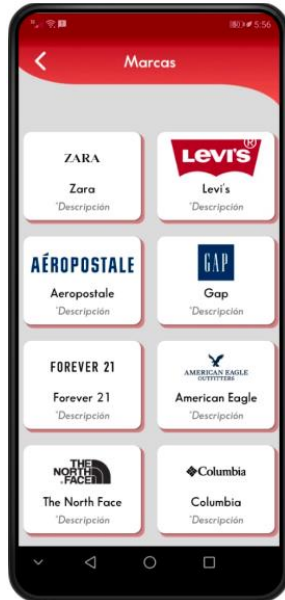
ANEXOS

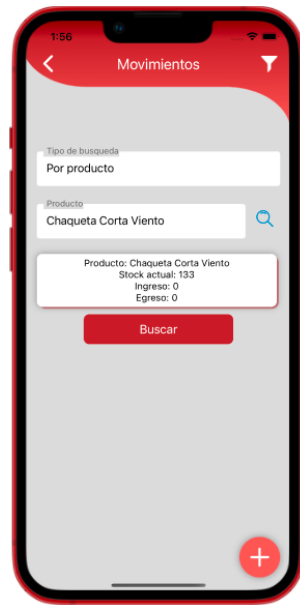
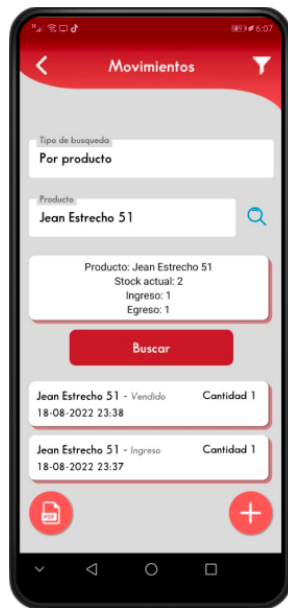
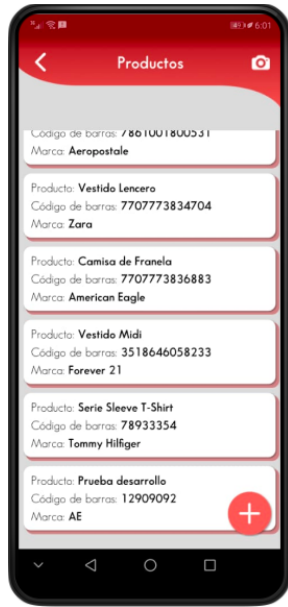
Anexo 1 Pantallas de la aplicación móvil multiplataforma.

Android



iOS





Anexo 2 Entrevista documento requerimientos.

Especificación de requisitos de software

Proyecto: "Aplicación móvil multiplataforma para el control de inventarios de la Boutique Nicoleth, utilizando la plataforma Xamarin."

1. Introducción

Entender todos los requisitos y requerimientos que nuestro cliente desea, además de como este constituye una entrevista útil, describiendo lo que realmente desea obtener como producto final, de esta manera lograr tener un documento necesario cuya información en el futuro servirá para el desarrollo del software, es decir en la codificación correcta del mismo.

Se describirá de forma detallada el propósito, el alcance, el personal involucrado, el diseño conceptual de la aplicación móvil, y los requisitos específicos.

1.2 Alcance

- Objetivos de la aplicación móvil.
 - Módulo de artículos.
 - Módulo de entrada de inventario.
 - Módulo de clasificación de artículos.
 - Reportes generales y notificaciones.

1.3 Personal involucrado

Nombre	Dennis Adrian Estrada
Rol	Scrum team
Responsabilidad	Desarrollo móvil
Información de contacto	dennis.estrada@unach.edu.ec

Nombre	Carlos David Chacon
Rol	Scrum team
Responsabilidad	Desarrollo móvil y comunicación directa con el cliente
Información de contacto	carlos.chacon@unach.edu.ec

Nombre	Alexis Villegas Pasos
Rol	Scrum máster
Responsabilidad	Guía, colaborador
Información de contacto	alexis.villegas@unach.edu.ec

- Introducción: En esta sección se detalla los objetivos que tiene el SRS y de nuestro sistema en forma general.
- Descripción General: Describe una perspectiva general del producto a desarrollarse, como también las características del usuario y las limitaciones que podría tener.
- Requerimientos Específicos: Muestra paso a paso todos los requerimientos que el usuario desea en el producto final.

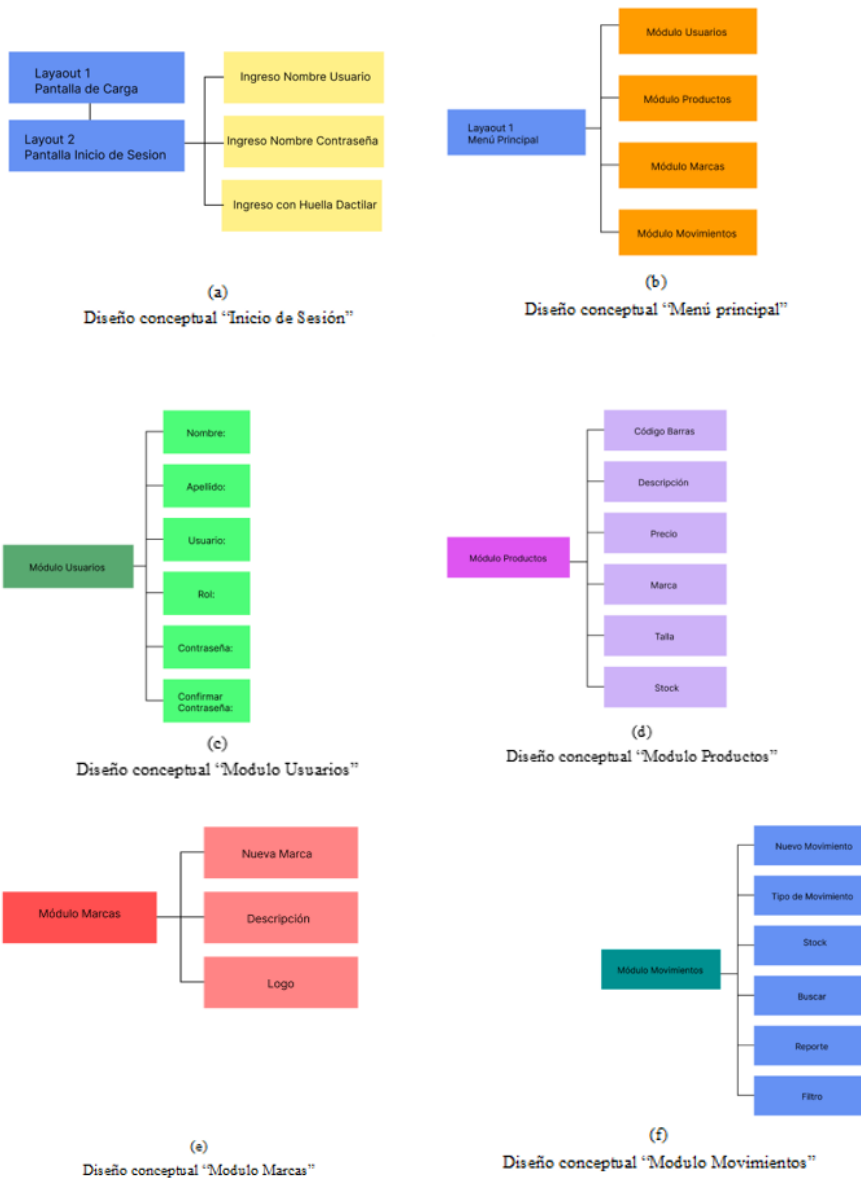
2. Descripción General

2.1 Perspectiva del producto

El sistema que se va a desarrollar es independiente, y tendrá un diseño *responsive* para que pueda ser utilizado en distintos dispositivos móviles multiplataforma.

2.2 Diseño del producto y funcionalidades

Diseño conceptual y la estructuración esquemática de los módulos de la aplicación.



2.3 Características de los usuarios

Tipo de usuario	Administrador
Información	Gerente, líder
Actividades	Administrador total de la app

Tipo de usuario	Usuario
Información	Empleado
Actividades	Realizador de movimientos en la microempresa

2.4 Restricciones

Solamente podrá ser utilizador para el administrador y los usuarios registrados, los usuarios tendrán ciertas restricciones es decir no tienen acceso total a todas las funcionalidades de la aplicación móvil.

2.5 Suposiciones y dependencias

Registro de usuarios.

2.6 Evolución previsible del sistema

Ampliación de la base de datos.

3. Requisitos específicos

N Requisito	Como	Quiero	Para
RE001	Administrador/Usuario	Iniciar Sesión	Identificarme como usuario Administrador.
RE002	Administrador	Registrar Usuarios	Tener un control de los usuarios que van a poder acceder a la aplicación móvil.
RE003	Administrador	Modificar usuarios	Para modificar, agregar o eliminar un usuario.
RE004	Administrador/Usuario	Modificar Marcas	Para modificar, agregar o eliminar marca.
RE005	Administrador/Usuario	Modificar Productos	Para modificar, agregar o eliminar productos.
RE006	Administrador	Otorgar permisos de la aplicación a usuario	Administrar los accesos del sistema por tipo de usuario (Administrador, Usuario Normal).

3.1 Requisitos funcionales

Requisito funcional 1

Número requisito	de	R001
Nombre requisito	de	Iniciar Sesión.
Descripción		Identificarme como usuario Administrador.
Prioridad		Alta.

Requisito funcional 2

Número requisito	de	R002
Nombre requisito	de	Registrar Usuarios.
Descripción		Tener un control de los usuarios que van a poder acceder a la aplicación móvil.
Prioridad		Alta.

Requisito funcional 3

Número requisito	de	R003
Nombre requisito	de	Modificar Usuarios.
Descripción		Para modificar, agregar o eliminar un usuario.
Prioridad		Alta.

Requisito funcional 4

Número requisito	de	R004
Nombre requisito	de	Modificar marca.
Descripción		Para modificar, agregar o eliminar marca.
Prioridad		Alta.

Requisito funcional 5

Número requisito	de	R005
Nombre requisito	de	Realizar procesos de control de inventario: ingresos, egresos y movimientos.
Descripción		Para llevar un control de inventario.
Prioridad		Alta.

Requisito funcional 6

Número de requisito	R006
Nombre de requisito	Reporte de los movimientos.
Descripción	Para llevar un control de inventario.
Prioridad	Alta.

Requisito funcional 7

Número de requisito	R007
Nombre de requisito	Notificaciones de los productos que están sin stock.
Descripción	Para llevar un control de los productos sin stock y abastecer de nuevos productos.
Prioridad	Alta.

3.2 Requerimientos no funcionales

- Seguridad
Uso de contraseñas para cada usuario (administrador, usuario). Esto permitirá que tengan acceso al sistema solo la persona que tienen autorización.
Registro de ingreso al sistema encargado por el administrador.
Creación de roles asignarlos a cada usuario dependiendo su funcionalidad.
- Disponibilidad
La aplicación móvil ha sido desarrollada tomando en cuenta las necesidades, requerimientos, reglas, objetivos etc. Estará en funcionamiento de acuerdo con la funcionalidad que se necesite.
- Compatibilidad
La aplicación móvil será multiplataforma puesto que podrá ejecutarse para Android y IOS de esta forma la aplicación móvil deberá ser *responsive* para cualquier medida de pantalla.

Anexo 3 Apis utilizadas para la aplicación móvil.

Nicolethinventario.API v1 GA53
/swagger/v1/swagger.json

Marcas ▼

- GET** /api/Marcas
- POST** /api/Marcas
- PUT** /api/Marcas/{idMarca}
- DELETE** /api/Marcas/{idMarca}

Movimientos ▼

- GET** /api/Movimientos
- POST** /api/Movimientos
- PUT** /api/Movimientos
- GET** /api/Movimientos/Producto/{idProducto}
- DELETE** /api/Movimientos/{idMovimiento}

Productos ▼

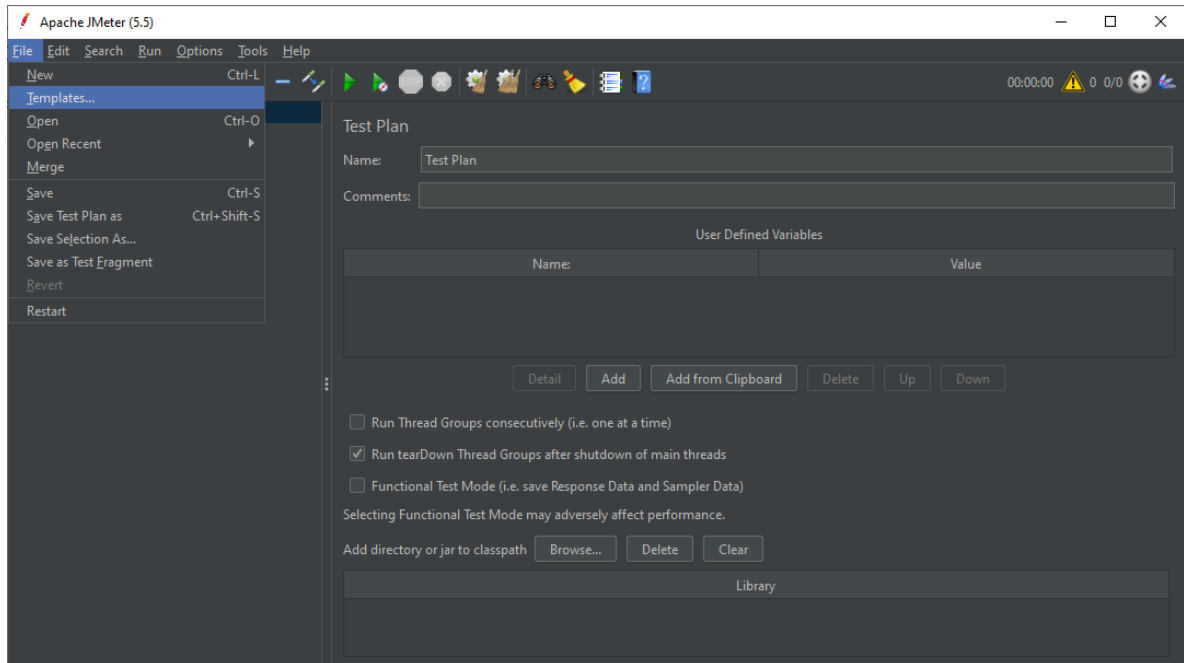
- GET** /api/Productos
- POST** /api/Productos
- GET** /api/Productos/SinStock

Activar Windows
ve a configuración para activar Windo

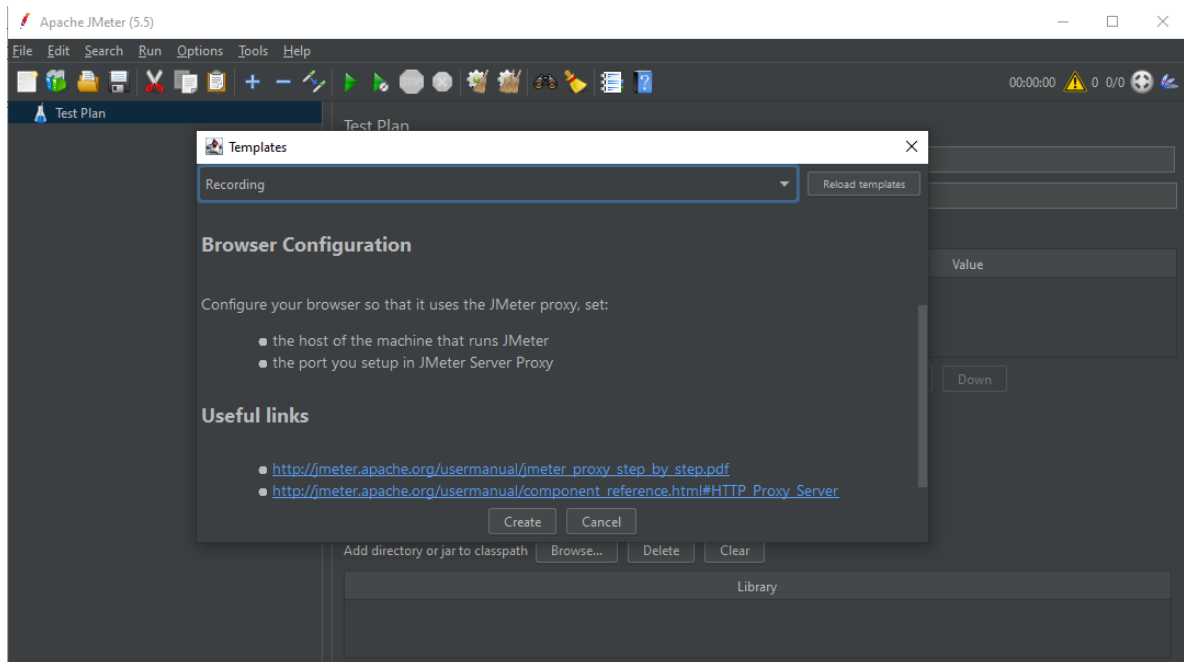
Anexo 4 Configuración JMeter en los dispositivos móviles

Configuración JMeter

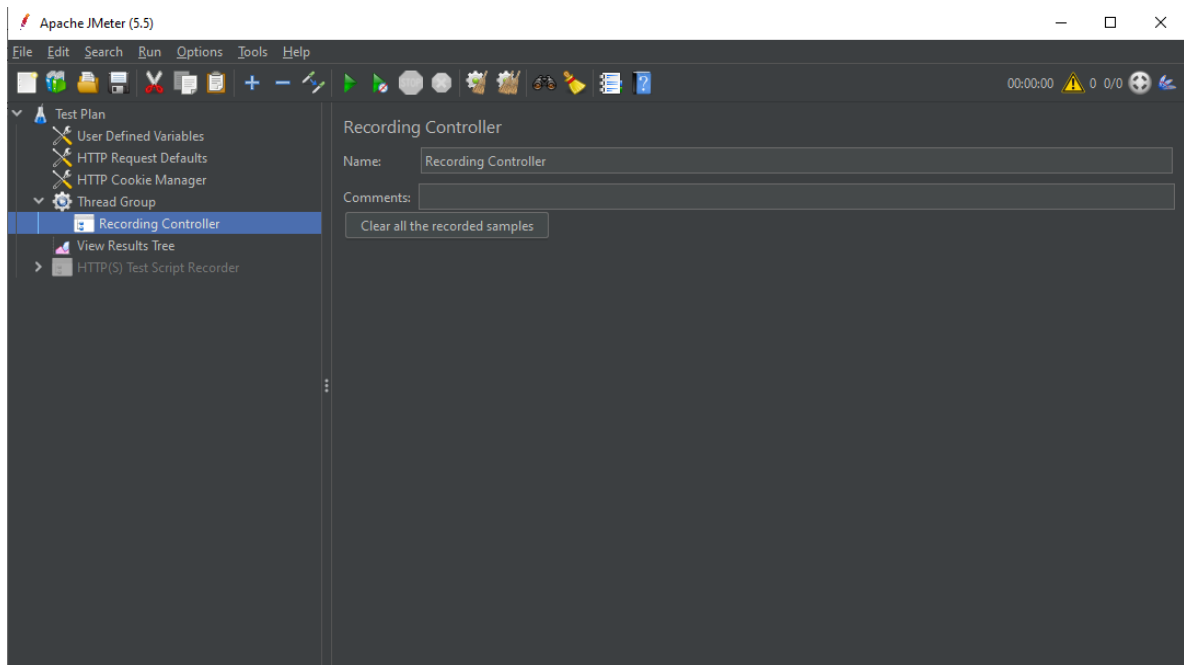
Se ejecuta el software JMeter y se presiona en file, seguido de eso Template.



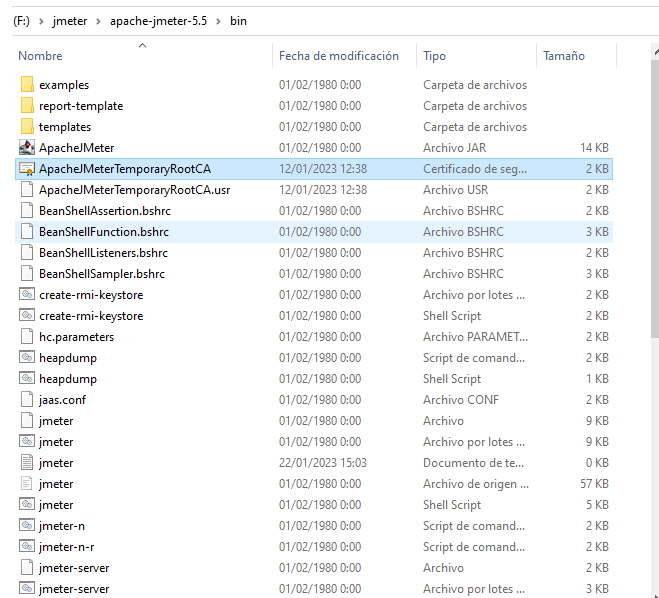
Una vez se selecciona templates se desplegará una nueva ventana en la cual se presiona el template Recording y en Create.



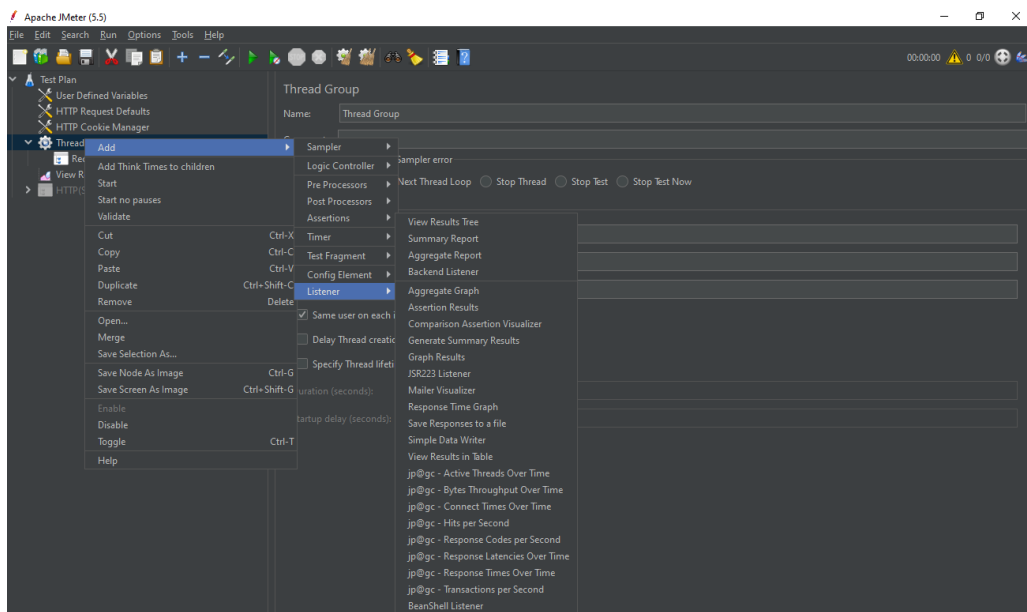
Nos dirigimos a la opción Thread Group y se observa que se agregó un controlador de grabación.



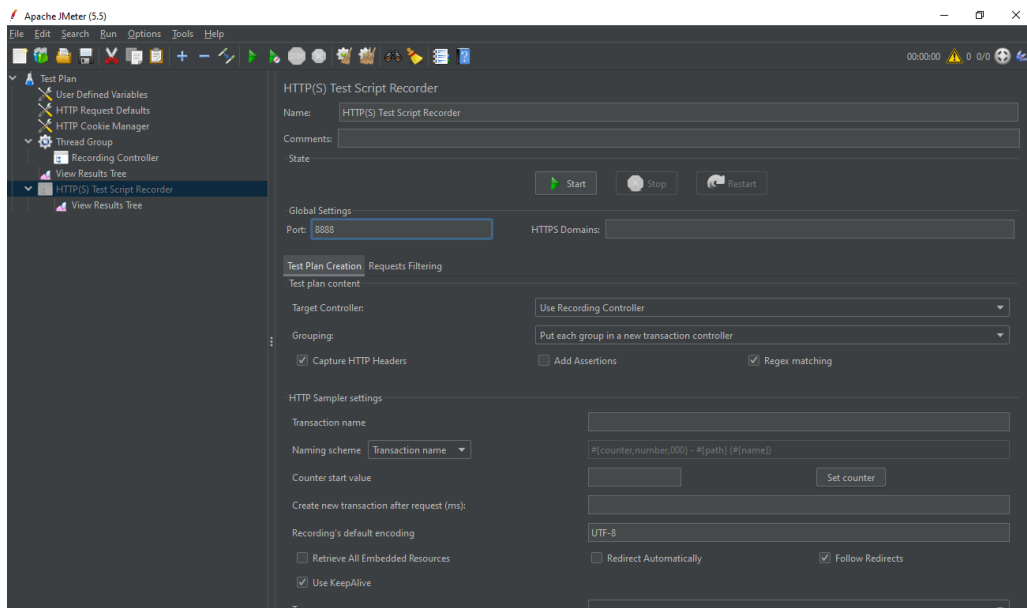
Nos dirigimos a la carpeta en la que se instaló JMeter y dentro de la carpeta bin se encuentra un certificado de seguridad que deberá ser instalado en el dispositivo móvil para poder tener conexión y posteriormente realizar la simulación.



Regresamos a JMeter y damos click derecho en Thread Group, posteriormente en Add, a continuación, presionamos Listener y finalmente agregamos el Listener que vamos a utilizar, por los general los más utilizados son: View Results Tree, Summary Report, Aggregate Report y Graph Results.



Posteriormente se selecciona en Test Script Recorder que se encuentra en la parte izquierda y dentro de esta ventana se observa la opción “Port”, aquí escribimos el puerto 8888 que posteriormente servirá para la conexión del dispositivo móvil con JMeter.

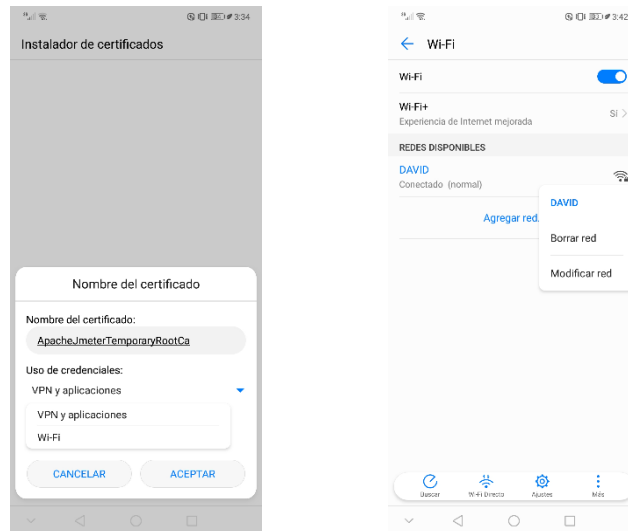


Instalación certificado de seguridad y configuración del dispositivo móvil (Android)

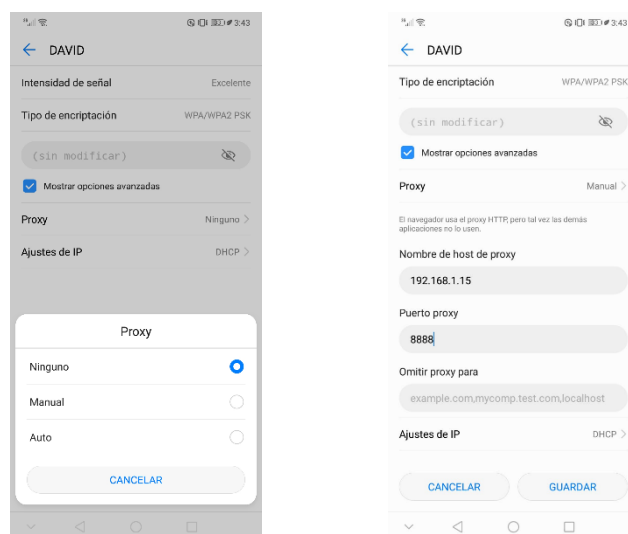
Se copia el archivo que dice certificado de seguridad y se envía a nuestro dispositivo.

ApacheJMeter	01/02/1980 0:00	Archivo JAR	14 KB
ApacheJMeterTemporaryRootCA	12/01/2023 12:38	Certificado de seg...	2 KB
ApacheJMeterTemporaryRootCA.usr	12/01/2023 12:38	Archivo USR	2 KB

Dentro de nuestro dispositivo móvil se busca el archivo ApacheJmeterTemporaryRootCA y se instala. Le damos un nombre y en el apartado “Uso de Credenciales” se escoge la opción Wi-Fi, finalmente se acepta y el certificado se instaló correctamente. Una vez instalado el certificado nos dirigimos al Wi-Fi de nuestro dispositivo y se selecciona la red a la cual estamos conectados y se selecciona la opción modifica red.



Dentro de la opción modificar red se selecciona Proxy y la opción Manual, finalmente en la opción host de proxy escribimos la direccion IPv4 de nuestra PC en la cual esta iniciado el JMeter y el puerto Proxy que se configuro en JMeter. Una vez hecho todos estos pasos el dispositivo móvil estará conectado con JMeter y se podrá realizar la simulación.

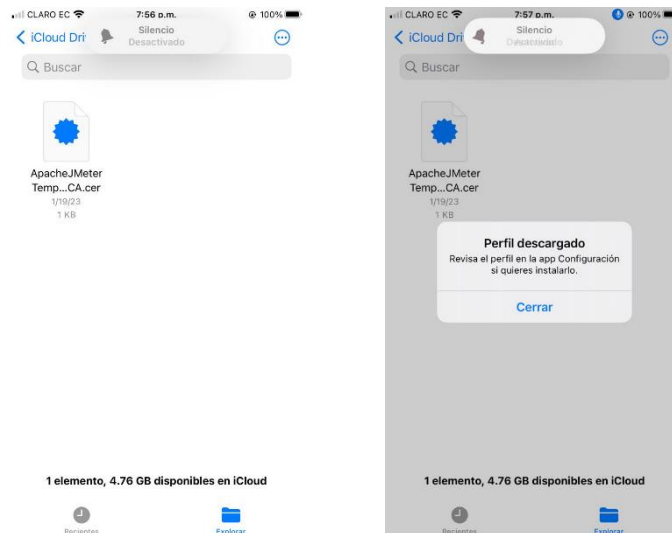


Instalación Certificado de Seguridad en el dispositivo móvil (iOS)

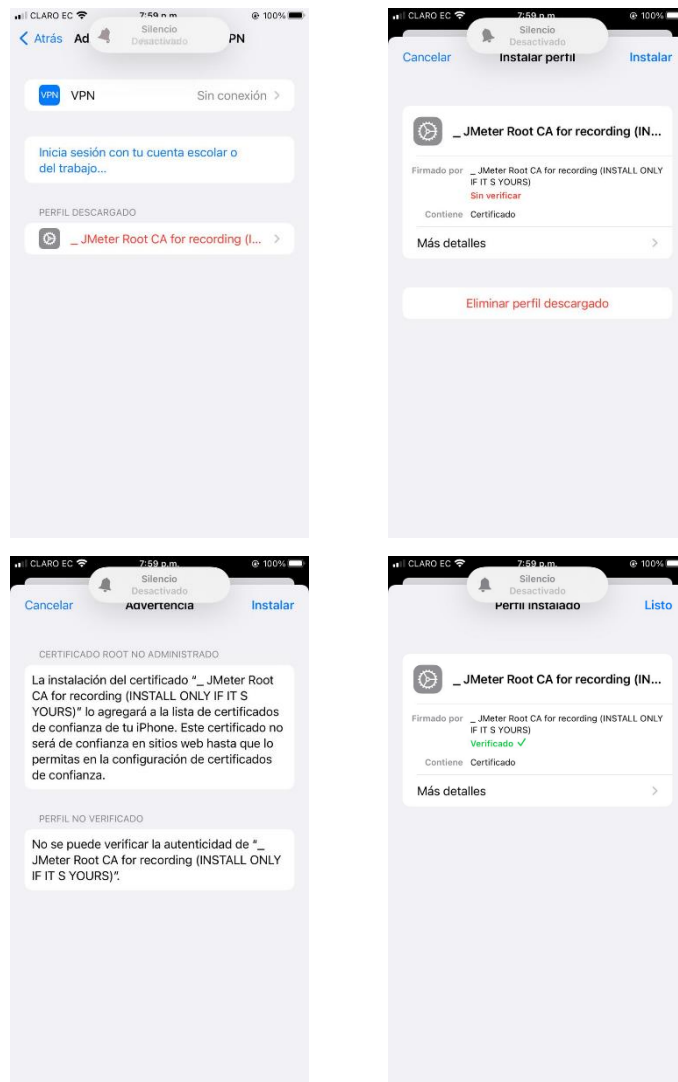
Se copia el archivo que dice certificado de seguridad y se envía a nuestro dispositivo.

ApacheJMeter	01/02/1980 0:00	Archivo JAR	14 KB
ApacheJMeterTemporaryRootCA	12/01/2023 12:38	Certificado de seg...	2 KB
ApacheJMeterTemporaryRootCA.usr	12/01/2023 12:38	Archivo USR	2 KB

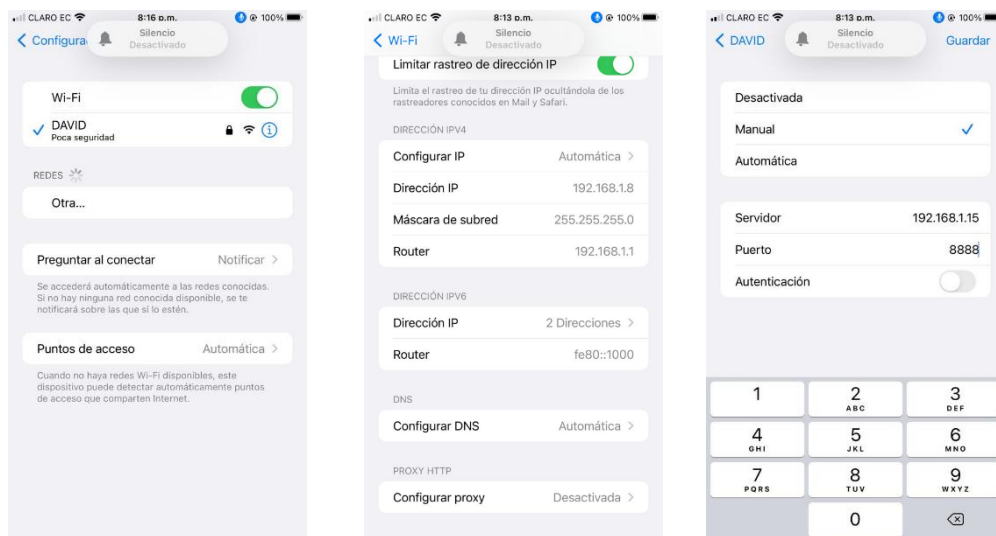
Dentro de nuestro dispositivo móvil se busca el archivo ApacheJmeterTemporaryRootCA se presiona y nos mostrara un mensaje.



A continuación, dentro de nuestro dispositivo nos dirigimos a configuración, dentro de configuración se presiona en general y finalmente se selecciona en Admin. de Dispositivos y VPN. Y el certificado de seguridad se encuentra listo para ser instalado. Para continuar se selecciona el certificado y nos mostrará una nueva ventana en la cual se nos muestra la opción instalar y nos aparecerá una ventana de advertencia, se acepta la advertencia y el certificado se instala.

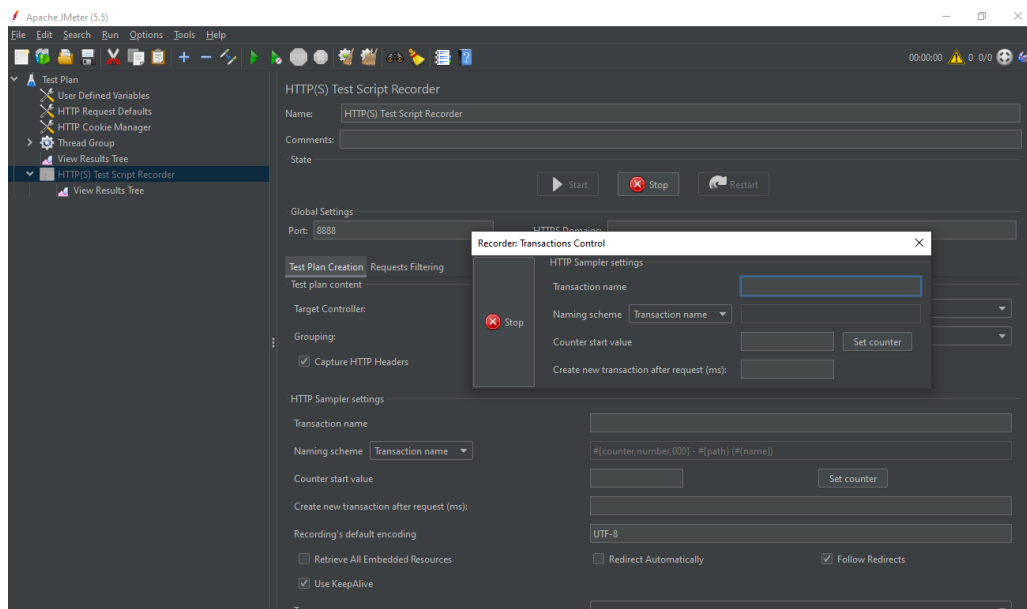


Para continuar nos dirigimos al Wi-Fi de nuestro dispositivo y se selecciona la red a la cual estamos conectados se presiona y nos mostrara las opciones disponibles de la red para configurar, nos dirigimos a la opción configurar proxy y se selecciona la opción manual y escribimos en la opción servidor la direccion IPv4 de la PC en la cual esta iniciado el JMeter y el puerto proxy que se configuro en JMeter. Una vez se realizó todo este proceso el dispositivo móvil tendrá conexión con JMeter para poder realizar la simulación.

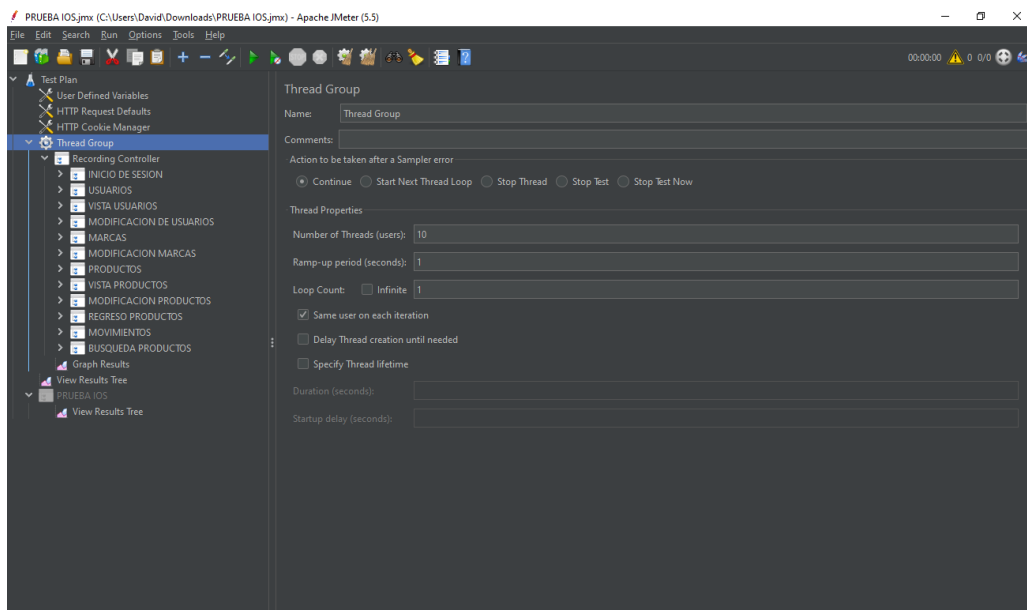


Simulación JMeter

Para empezar con la simulación nos dirigimos a la opción Test Script Recorder y se presiona en la opción Star y empezara a realizar una grabación de las interacciones del dispositivo móvil que anteriormente fue conectado con el certificado de seguridad.



Se aprecia en la ventana izquierda en la opción Thread Group que la grabación de las interacciones se realizó con éxito y a continuación se configura el número de usuarios que realizaran la simulación en el apartado Number of Threads.



Para poder observar los resultados de la simulación se selecciona en el listener View Results in Tree y a continuación se selecciona el botón de play que se encuentra en la barra de navegación superior. Finalmente se observa los resultados de la simulación y que esta se realizó con éxito.

