



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UNA APLICACIÓN MÓVIL DE RECONOCIMIENTO
DE SUELOS USANDO DEEP LEARNING.

ORTIZ TRUJILLO YASSER JAVIER
INGENIERO DE SISTEMAS

MACHALA
2022



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UNA APLICACIÓN MÓVIL DE
RECONOCIMIENTO DE SUELOS USANDO DEEP LEARNING.

ORTIZ TRUJILLO YASSER JAVIER
INGENIERO DE SISTEMAS

MACHALA
2022



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

TRABAJO TITULACIÓN
PROPUESTAS TECNOLÓGICAS

DESARROLLO DE UNA APLICACIÓN MÓVIL DE RECONOCIMIENTO DE SUELOS
USANDO DEEP LEARNING.

ORTIZ TRUJILLO YASSER JAVIER
INGENIERO DE SISTEMAS

RIVAS ASANZA WILMER BRAULIO

MACHALA, 23 DE FEBRERO DE 2022

MACHALA
2022

tesis yasser ortiz

INFORME DE ORIGINALIDAD

7%

INDICE DE SIMILITUD

6%

FUENTES DE INTERNET

1%

PUBLICACIONES

2%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	repositorio.unp.edu.pe Fuente de Internet	1%
2	hdl.handle.net Fuente de Internet	1%
3	dspace.utb.edu.ec Fuente de Internet	1%
4	www.slideshare.net Fuente de Internet	<1%
5	repositorio.utc.edu.ec Fuente de Internet	<1%
6	repositorio.utn.edu.ec Fuente de Internet	<1%
7	Submitted to Universidad Tecnológica Israel Trabajo del estudiante	<1%
8	repositorio.unsa.edu.pe Fuente de Internet	<1%
9	MGTI Luz Maria Hernandez Cruz, M. en C. Guadalupe Manuel Estrada Segovia, Mtra.	<1%

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

El que suscribe, ORTIZ TRUJILLO YASSER JAVIER, en calidad de autor del siguiente trabajo escrito titulado DESARROLLO DE UNA APLICACIÓN MÓVIL DE RECONOCIMIENTO DE SUELOS USANDO DEEP LEARNING., otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

El autor declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 23 de febrero de 2022



ORTIZ TRUJILLO YASSER JAVIER
0705267268



UNIVERSITAS
MAGISTRORUM
ET SCHOLARIUM

Dedicatoria

El presente trabajo lo dedico especialmente a Dios, por darme la inspiración necesaria para continuar en esta etapa de obtener un título profesional.

A mis padres, por su apoyo, trabajo y abnegación en todos estos años, gracias a ustedes he logrado avanzar hasta donde estoy actualmente. Estoy muy agradecido de ustedes por la buena educación que me han dado, son los mejores padres.

A todas las personas que de manera directa o indirecta han hecho que este trabajo se realice con éxito como son todos los docentes tutores del desarrollo de la tesis, así como todas las personas que compartieron su conocimiento en internet.

Agradecimiento

Al culminar este trabajo intentó ofrecer este apartado para agradecer a Dios por todas sus bendiciones y a mis padres que han sabido proporcionar su ejemplo de trabajo e integridad.

También quiero agradecer a la Universidad Técnica de Machala, directivos y educadores por la organización del programa de titulación en la carrera Ingeniería de Sistemas.

Resumen

La inclusión de la inteligencia artificial en proyectos tecnológicos se ha incrementado en los últimos años, esto se debe a que existe un gran potencial en este tipo de tecnología, debido a que con la inteligencia artificial se puede realizar varias actividades como es el clasificar imágenes para detectar objetos diferentes y de esta manera poder identificar qué objeto es el que se muestra en una determinada ventana de una aplicación.

La inteligencia artificial comprende varios aspectos como es el uso de redes neuronales para realizar los procesos de entrenamiento de un modelo que se creará mediante el uso de lenguajes de programación especiales para dicho enfoque, además se necesita tener previamente una data de imágenes a clasificar para dicho entrenamiento.

Actualmente se pueden encontrar en distintos artículos de revistas científicas muchos proyectos que están relacionados con la implementación de aplicaciones móviles que incluyan en sus funciones el uso de inteligencia artificial para realizar ciertos procesos que requieran el análisis de imágenes para detectar por ejemplo caries o placas bacterianas en las fotos de las bocas de un grupo de pacientes seleccionados como data para la red neuronal que se esté utilizando o para la visión artificial que se esté usando.

Conociendo lo anterior, se desarrolló una aplicación móvil de reconocimiento de suelos usando Deep Learning, que mediante un modelo de entrenamiento creado previamente con redes neuronales convolucionales es consumido para realizar las respectivas predicciones de las imágenes capturadas del celular o de fotos de la galería del dispositivo, indicando su probabilidad y nombre de suelo.

El desarrollo de este aplicativo móvil fue realizado con el Framework de Xamarin el cual es un entorno multiplataforma que trabaja con el lenguaje de programación C# y XAML (Lógica y Diseño respectivamente), además se utilizó ciertas librerías para trabajar con TensorFlow, el cual es otro Framework que sirve para crear y usar redes neuronales. Se empleó la metodología XP para la construcción del proyecto debido a su enfoque ágil y por darle más prioridad al tiempo de desarrollo que en la comunicación con los interesados.

Para el uso de la aplicación el usuario deberá elegir entre dos opciones para proceder con el análisis de la imagen o foto. En la ventana inicial saldrá un menú en el que se podrá elegir entre seleccionar una imagen de la galería o tomar una foto, seguido de esto se mostrará una nueva ventana llamada análisis en la cual se podrá observar la imagen obtenida de la opción previamente seleccionada, además saldrá un mensaje indicando el nombre del suelo detectado con su probabilidad de certeza y una lista de los otros suelos con sus respectivas probabilidades de certeza ordenadas de forma descendente.

Para evaluar la aplicación móvil se utilizó el estándar de calidad ISO/IEC 25010, debido que este estándar se enfoca en aplicar varias métricas de evaluación de calidad al producto final, es decir a la aplicación móvil desarrollada, además de estas métricas también se emplean pruebas unitarias que confirman el cumplimiento de las historias de usuarios planteadas en este informe.

Palabras clave: Aplicación móvil, Aprendizaje profundo, Análisis del suelo, Redes neuronales convolucionales

Abstract

The inclusion of artificial intelligence in technological projects has increased in recent years, this is because there is great potential in this type of technology, because with artificial intelligence several activities can be carried out, such as classifying images for detect different objects and in this way be able to identify which object is the one that is shown in a certain window of an application.

Artificial intelligence comprises several aspects such as the use of neural networks to carry out the training processes of a model that will be created through the use of special programming languages for said approach, in addition, it is necessary to previously have image data to classify for said training.

Currently, you can find in different articles in scientific journals many projects that are related to the implementation of mobile applications that include in their functions the use of artificial intelligence to carry out certain processes that require image analysis to detect, for example, caries or bacterial plaque in the photos of the mouths of a group of patients selected as data for the neural network that is being used or for the artificial vision that is being used.

Knowing the above, a soil recognition mobile application was developed using Deep Learning, which through a training model previously created with convolutional neural networks is consumed to make the respective predictions of the images captured from the cell phone or photos from the device's gallery, indicating its probability and soil name.

The development of this mobile application was carried out with the Xamarin Framework, which is a multiplatform environment that works with the programming language C# and XAML (Logic and Design respectively), in addition certain libraries were used to work with TensorFlow, which is another Framework used to create and use neural networks. The XP methodology was used for the construction of the project due to its agile approach and for giving more priority to development time than to communication with stakeholders.

To use the application, the user must choose between two options to proceed with the analysis of the image or photo. In the initial window a menu will appear in which you can choose between selecting an image from the gallery or taking a photo, followed by a new window called analysis in which you can see the

image obtained from the previously selected option, In addition, a message will appear indicating the name of the soil detected with its probability of certainty and a list of the other soils with their respective probabilities of certainty ordered in descending order.

To evaluate the mobile application, the ISO/IEC 25010 quality standard was used, because this standard focuses on applying several quality evaluation metrics to the final product, that is, to the developed mobile application, in addition to these metrics tests are also used. units that confirm compliance with the user stories outlined in this report.

Keywords: Mobile Application, Deep Learning, Soil Analysis, Convolutional Neural Networks

Introducción

La utilización de aplicaciones móviles ha aumentado considerablemente en estos últimos años, debido a los nuevos avances tecnológicos en los dispositivos móviles, como es la incorporación de pantallas táctiles en ellos, así como mejoras en las versiones de los sistemas operativos que estos utilizan.

La creciente innovación tecnológica respecto al uso de la inteligencia artificial es muy diversa, debido a esto existen hoy en día varios proyectos tecnológicos orientados a solucionar problemas, como en la salud, el trabajo y en cualquier otra actividad que las personas realicen.

Considerando lo preliminar, este trabajo muestra el desarrollo de una aplicación móvil de reconocimiento de suelos usando Deep Learning, como alternativa en el proceso de revisión de suelos fértiles previo a la siembra de algún tipo de cultivo.

La aplicación móvil consta de una interfaz gráfica amigable fácil de usar por parte del usuario, además se la desarrolló en un marco de trabajo multiplataforma, igualmente usa librerías especiales para el consumo del modelo de red neuronal.

El informe clasifica sus contenidos en distintos capítulos, de esta manera el primer capítulo detalla el ámbito de la aplicación, los requerimientos que esta requiere en su desarrollo y la justificación del porqué se debe cumplir dichos requerimientos, el segundo capítulo contiene la definición del prototipo, fundamentación teórica, definición de la metodología a usar, objetivos de desarrollo y las fases de dicha metodología como es la planificación, diseño y ejecución, finalmente el tercer capítulo contiene el plan de evaluación y el análisis de los resultados.

Índice de Contenido

DEDICATORIA	6
AGRADECIMIENTO	7
RESUMEN	8
ABSTRACT	10
INTRODUCCIÓN	12
ÍNDICE DE CONTENIDO	13
ÍNDICE DE TABLAS	15
ÍNDICE DE ILUSTRACIONES	16
1. DIAGNOSTICO DE NECESIDADES Y REQUERIMIENTOS	17
1.1. ÁMBITO DE APLICACIÓN: DESCRIPCIÓN DEL CONTEXTO Y HECHOS DE INTERÉS.	17
1.2. ESTABLECIMIENTO DE REQUERIMIENTOS.	17
1.3. JUSTIFICACIÓN DEL REQUERIMIENTO A SATISFACER.	19
2. DESARROLLO DEL PROTOTIPO	20
2.1. DEFINICIÓN DEL PROTOTIPO TECNOLÓGICO.	20
2.1.1. DEFINITION DEL SISTEMA	20
2.1.2. DEFINICIÓN DEL ALCANCE	20
2.1.3. IDENTIFICACIÓN DE RIESGOS	21
2.1.4. REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES	22
2.2. FUNDAMENTACIÓN TEÓRICA DEL PROTOTIPO.	23
2.2.1. ENTORNO DE TRABAJO	23
2.2.2. METODOLOGÍA XP	24
2.2.3. ANÁLISIS DE SUELOS	24
2.2.4. APLICACIONES MÓVILES	28
2.3. METODOLOGÍA	31
2.4. OBJETIVOS DEL PROTOTIPO.	31
2.4.1. OBJETIVO GENERAL	31
2.4.2. OBJETIVOS ESPECÍFICOS	31
2.5. FASE DE PLANIFICACIÓN	32
2.5.1. DEFINICIÓN DE ROLES	32
2.5.2. CRONOGRAMA	34
2.5.3. HISTORIAS DE USUARIO	35
2.5.4. ANÁLISIS DEL SISTEMA	38
2.6. DISEÑO DEL PROTOTIPO.	41
2.6.1. DISEÑO ARQUITECTÓNICO	41

2.6.2. MODELADO ENTIDAD-RELACIÓN Y RELACIONAL	42
2.6.3. DIAGRAMA DE CASOS DE USO	43
2.6.4. DIAGRAMAS DE ACTIVIDAD	45
2.7. EJECUCIÓN Y/O ENSAMBLAJE DEL PROTOTIPO.	49
2.7.1. FRAMEWORK DE DESARROLLO	49
2.7.2. LENGUAJE DE PROGRAMACIÓN	49
2.7.3. MÓDULOS DEL SISTEMA	49
3. EVALUACIÓN DEL PROTOTIPO	60
3.1. PLAN DE EVALUACIÓN.	60
3.1.1. PRUEBAS UNITARIAS	60
3.1.2. EVALUACIÓN DE CALIDAD DEL SOFTWARE	65
3.2. RESULTADOS DE LA EVALUACIÓN.	72
4. CONCLUSIONES.	73
5. RECOMENDACIONES.	74
BIBLIOGRAFÍA	75
ANEXOS	79

Índice de Tablas

Tabla 1: <i>Identificación de riesgos</i>	21
Tabla 2: <i>Definición de requerimientos funcionales</i>	22
Tabla 3: <i>Definición de requerimientos no funcionales</i>	22
Tabla 4: <i>Definición de roles</i>	32
Tabla 5: <i>Historia de usuario menú inicial</i>	35
Tabla 6: <i>Historia de usuario seleccionar imagen de galería</i>	35
Tabla 7: <i>Historia de usuario tomar fotografía con cámara del celular</i>	36
Tabla 8: <i>Historia de usuario ventana analizar</i>	36
Tabla 9: <i>Historia de usuario analizar imagen o foto</i>	37
Tabla 10: <i>Historia de usuario navegación entre ventanas</i>	37
Tabla 11: <i>Plan de publicaciones</i>	38
Tabla 12: <i>Plan de duración de iteraciones</i>	39
Tabla 13: <i>Plan de entrega</i>	40
Tabla 14: <i>Velocidad del proyecto</i>	40
Tabla 15: <i>Prueba de aceptación N°1</i>	61
Tabla 16: <i>Prueba de aceptación N°2</i>	61
Tabla 17: <i>Prueba de aceptación N°3</i>	62
Tabla 18: <i>Prueba de aceptación N°4</i>	63
Tabla 19: <i>Prueba de aceptación N°5</i>	64
Tabla 20: <i>Prueba de aceptación N°6</i>	64
Tabla 21: <i>Indicador de evaluación de calidad</i>	65
Tabla 22: <i>Evaluación de calidad según ISO/IEC 25010</i>	66
Tabla 23: <i>Resumen de resultados de evaluación</i>	72

Índice de Ilustraciones

<i>Ilustración 1: Clasificación de suelos por su color</i>	25
<i>Ilustración 2: Comunicación de la aplicación móvil</i>	28
<i>Ilustración 3: Cronograma de actividades</i>	34
<i>Ilustración 4: Diseño arquitectónico del prototipo</i>	42
<i>Ilustración 5: Modelo entidad-relación</i>	43
<i>Ilustración 6: Modelo relacional</i>	43
<i>Ilustración 7: Caso de uso seleccionar imagen de galería</i>	44
<i>Ilustración 8: Caso de uso tomar fotografía con la cámara</i>	44
<i>Ilustración 9: Caso de uso analizar imagen elegida</i>	45
<i>Ilustración 10: Diagrama de actividades seleccionar imagen de galería</i>	45
<i>Ilustración 11: Diagrama de actividades tomar fotografía con la cámara</i>	46
<i>Ilustración 12: Diagrama de actividades analizar imagen elegida</i>	46
<i>Ilustración 13: Diseño de interfaz menú</i>	47
<i>Ilustración 14: Diseño de interfaz analizar</i>	48
<i>Ilustración 15: Estructura del proyecto</i>	50
<i>Ilustración 16: Código de inicio en XAML (Vista)</i>	51
<i>Ilustración 17: Código de inicio en C# (Vista)</i>	52
<i>Ilustración 18: Código de inicio en C# (Controlador)</i>	53
<i>Ilustración 19: Código de análisis en XAML (Vista)</i>	54
<i>Ilustración 20: Código de análisis en C# (Vista)</i>	55
<i>Ilustración 21: Código de análisis en C# (Controlador)</i>	56
<i>Ilustración 22: Código de Prediccion en C#</i>	57
<i>Ilustración 23: Código de IClasificador en C#</i>	58
<i>Ilustración 24: Código de TensorFlowClasifier P1 en C#</i>	58
<i>Ilustración 25: Código de TensorFlowClasifier P2 en C#</i>	59
<i>Ilustración 26: Código de TensorFlowClasifier P3 en C#</i>	59
<i>Ilustración 27: Ventana del menu principal</i>	79
<i>Ilustración 28: Selector del álbum de la galería</i>	80
<i>Ilustración 29: Elegir una imagen del álbum seleccionado</i>	81
<i>Ilustración 30: Ventana del análisis de la imagen</i>	82
<i>Ilustración 31: Análisis de la imagen elegida</i>	83
<i>Ilustración 32: Tomar una foto con la cámara</i>	84
<i>Ilustración 33: Confirmación de la foto tomada</i>	85
<i>Ilustración 34: Análisis de la foto tomada</i>	86

1. DIAGNÓSTICO DE NECESIDADES Y REQUERIMIENTOS

1.1.Ámbito de aplicación: descripción del contexto y hechos de interés.

Dado el incremento del uso de los dispositivos móviles para realizar diversas actividades en diferentes oficios o trabajos, además del apogeo de nuevas tecnologías adaptables a dichos dispositivos, se puede mencionar que el incorporar conceptos como la inteligencia artificial en el sector agrícola es muy interesante y más aún si se puede ayudar a optimizar algún proceso en la labor diaria de las personas que se dedican a esta actividad.

Actualmente existen publicados en revistas científicas de todo el mundo muchos proyectos tecnológicos que involucran la inteligencia artificial en aplicaciones móviles para solucionar problemas o satisfacer las necesidades del público objetivo.

Dichos proyectos utilizan por lo general redes neuronales convolucionales que aprenden información de la data suministrada mediante un proceso de entrenamiento y validación que involucra internamente varias neuronas que forman parte de distintas capas, entre ellas está la capa de convolución, de reducción y la multicapa, además de una función de validación llamada softmax, el resultado obtenido creará un modelo que servirá para clasificar las imágenes del objetivo de investigación propuesto.

El alcance del presente proyecto comprende el desarrollo de una aplicación móvil de reconocimiento de suelos usando Deep Learning, con el uso del Framework Xamarin. Este aplicativo permitirá al usuario navegar en un menú fácil de utilizar y elegir entre dos opciones la una le permitirá tomar una foto del suelo al instante y la segunda le permitirá elegir una imagen de la galería de su celular, seguido de cualquiera de las dos opciones anteriores se abrirá una nueva ventana que le mostrará al usuario la imagen seleccionada o tomada y le permitirá dar la opción de analizar la foto dando como resultado un mensaje con el tipo de suelo que se detectó, la probabilidad de certeza del mismo y una lista de los otros suelos con sus respectivas probabilidades de certeza ordenadas de forma descendente.

1.2. Establecimiento de requerimientos.

Con el propósito de facilitar el manejo de la aplicación móvil se simplificó sus funciones tomando en cuenta lo más importante y necesario para cumplir con el objetivo del proyecto, por lo que los requerimientos generales se detallan a continuación:

- Contiene un menú de agradable apariencia que muestra un texto de bienvenida y un mensaje instructivo de descripción, en esta ventana existen dos opciones la primera que es para obtener una imagen de la galería del celular y la segunda que es para obtener una foto mediante la cámara del dispositivo.
- Existe una ventana de análisis y resultado en la cual se podrá ver la imagen previamente seleccionada o tomada, además de la opción para dar la orden del análisis y un campo de texto en donde se detalla el tipo de suelo que es y su probabilidad de certeza, adicional a esto se mostrará una lista con los demás tipos de suelos indicando su probabilidad de certeza obtenida ordenado de forma descendente.
- La aplicación móvil será implementada en el sistema operativo Android.

1.3. Justificación del requerimiento a satisfacer.

Este proyecto se elabora bajo la línea de investigación de tecnologías para la producción sustentable.

Actualmente el personal encargado de sembrar algún producto agrícola en una hacienda debe realizar con anterioridad un estudio del suelo el cual comprende analizar algunos factores como: fertilidad del suelo, topografía, textura, color del suelo, microorganismos y la profundidad del suelo, lo cual realizar consume tiempo y recursos económicos para el dueño.

Por los motivos anteriores se pretende desarrollar una aplicación móvil de reconocimiento de suelos usando Deep Learning, el cual facilitará y disminuirá el tiempo en que los encargados de sembrar un cultivo realicen sus actividades mencionadas anteriormente.

Este aplicativo permitirá al usuario navegar en un menú de agradable apariencia en donde se encuentre con dos opciones previas al análisis de la imagen del suelo que se tiene como objetivo. Las dos opciones involucran el uso de imágenes, la opción uno permite tomar una foto y la opción dos permite seleccionar una imagen de la galería del dispositivo, seguido de esto se viaja a una segunda ventana en donde se puede ver los resultados del análisis que involucran mostrar la imagen, el tipo de suelo, la probabilidad de certeza y una lista de los suelos restantes con sus indicaciones respectivas.

El proyecto a pesar de estar construido en un Framework multiplataforma se lo enfocó al sistema operativo Android, debido a que es el sistema operativo más común entre los celulares de las personas, además de no contar con un dispositivo o emulador con el sistema operativo IOS.

2. DESARROLLO DEL PROTOTIPO

2.1. Definición del prototipo tecnológico.

2.1.1. Definición del Sistema

Actualmente el uso de las aplicaciones móviles es muy frecuente debido a las mejoras que estos han recibido en sus sistemas operativos y en sus funcionalidades. Una aplicación móvil que facilite el análisis de suelos para el ámbito agrícola muestra una oportunidad de innovación tecnológica que pueda servir para optimizar las actividades diarias de las personas que se dedican a sembrar cultivos y que siempre deben tener en cuenta el tipo de suelo en donde se va a realizar dicha actividad.

La aplicación desarrollada se llevó a cabo en un framework multiplataforma (Xamarin) para que sea más diversa la compatibilidad con los sistemas operativos Android y iOS. Su construcción se enfocó en darle importancia a la apariencia y manejo agradable para que de esta manera el usuario pueda hacer un análisis de suelos sin problemas.

La orientación de este objetivo de estudio se dio por el uso de las redes neuronales convolucionales que son parte de la inteligencia artificial y que en esta aplicación se las usa mediante el consumo de un modelo que clasifica imágenes de distintos tipos de suelos que posteriormente servirán para predecir el suelo que el usuario fotografió o seleccionó de su galería de su dispositivo.

2.1.2. Definición del alcance

Se plantea el desarrollo de una aplicación enfocada en el uso de inteligencia artificial para el análisis de suelos según su color aplicando técnicas de redes neuronales convolucionales. La aplicación se desarrolla para el sistema operativo Android bajo el framework Xamarin aplicando la metodología de desarrollo ágil XP.

2.1.3. Identificación de riesgos

Tabla 1: Identificación de riesgos

N #.	Riesgo	Efecto	Impacto	Prob.	Evaluación de riesgo	Estrategia de mitigación
1	Falta de comunicación sobre el diseño de bocetos en el equipo de trabajo	Desacuerdos en la apariencia final del prototipo.	2	2	Medio	Comunicación periódica y exposición de avances entre el equipo de trabajo.
2	Falta de comunicación sobre el desarrollo del prototipo en el equipo de trabajo	Insatisfacción con el resultado final del prototipo.	3	2	Alto	Comunicación periódica y exposición de avances entre el equipo de trabajo.
3	Problemas de compatibilidad entre librerías internas del prototipo	Lentitud o inoperatividad de las funciones del prototipo.	3	1	Medio	Realizar pruebas para detectar compatibilidad entre librerías.
4	Problemas de compatibilidad entre hardware y software	Apariencia gráfica distinta a la deseada y funcionalidad parcial.	2	1	Bajo	Implementación constante de pruebas durante el desarrollo.
5	Fallos en el consumo del modelo	Resultados nulos en el análisis de suelos	3	1	Medio	Buscar alternativas para el consumo de modelos.

Fuente: Elaboración propia

2.1.4. Requerimientos funcionales y no funcionales

Tabla 2: Definición de requerimientos funcionales

Especificación de requerimientos funcionales	
RF-01	Permite que el usuario tome una fotografía del suelo en analizar desde la cámara de su dispositivo.
RF-02	Permite que el usuario seleccione una imagen del suelo a analizar desde la galería de su dispositivo.
RF-03	Permite cambiar la imagen seleccionada o fotografiada antes de realizar el análisis.
RF-04	Permite ver el tipo de suelo y su porcentaje de certeza como resultado del análisis.
RF-05	Permite verificar el resultado realizando otra vez el proceso de análisis con los datos ya cargados.

Fuente: Elaboración propia

Tabla 3: Definición de requerimientos no funcionales

Especificación de requerimientos no funcionales	
RNF-01	La aplicación será compatible con dispositivos Android
RNF-02	Las ventanas deberán ser fáciles de manejar y usar
RNF-03	El consumo del modelo de la red neuronal convolucional será de manera local.
RNF-04	La aplicación utilizará un diseño de interfaz gráfica responsive para adaptar las ventanas a cualquier tipo de pantalla.
RNF-05	Las tonalidades de color, imágenes y tamaños de letras permitirán que el usuario se familiarice rápidamente con su funcionamiento.
RNF-06	La aplicación consumirá rápidamente el modelo para predecir y mostrar el resultado en un corto tiempo.

Fuente: Elaboración propia

2.2. Fundamentación teórica del prototipo.

2.2.1. Entorno de Trabajo

2.2.1.1. Herramientas de software

2.2.1.1.1. Visual Studio Community 2019

Es un entorno de trabajo para desarrolladores que están familiarizados con el lenguaje C# y .Net, así como lo menciona su página oficial en [1], *“El mejor IDE completo para desarrolladores de .NET y C++ en Windows. Completamente equipado con una buena matriz de herramientas y características para potenciar y mejorar todas las fases del desarrollo de software.”*

2.2.1.1.2. Xamarin

En una herramienta de trabajo instalable en Visual Studio para desarrollar aplicaciones móviles multiplataforma con el lenguaje C# y .Net, así como lo indica su página oficial en [2], *“Xamarin amplía la plataforma de desarrollo .NET con herramientas y bibliotecas específicamente para crear aplicaciones para Android, iOS, tvOS, watchOS, macOS y Windows.”*

2.2.1.2. Librerías

2.2.1.2.1. TensorFlow.Lite

Es una plataforma de código abierto que permite crear redes neuronales convolucionales y muchos otros tipos de técnicas que enfrasca el aprendizaje automático, así como lo muestra su página oficial en [3], *“TensorFlow es una plataforma de código abierto de extremo a extremo para el aprendizaje automático. Cuenta con un ecosistema integral y flexible de herramientas, bibliotecas y recursos de la comunidad que permite que los investigadores innoven con el aprendizaje automático y los desarrolladores creen e implementen aplicaciones con tecnología de AA fácilmente.”*

2.2.1.2.2. Acr.UserDialogs

Es una librería multiplataforma que permite al programador usar diversos tipos de cuadros de diálogo para embellecer el comportamiento de las aplicaciones desarrolladas, así lo detalla el

repositorio nuget en [4], *“Una biblioteca multiplataforma que le permite llamar a cuadros de diálogo de usuario estándar desde una biblioteca compartida/portátil. Compatible con Android, iOS y UWP”*

2.2.1.2.3. Xam.Plugin.Media

Es una librería que permite usar la cámara y la galería del celular en el aplicativo móvil para seleccionar imágenes, tomar fotos o videos, así lo describe el repositorio nuget en [5], *“Tome o elija fotos y videos de una API multiplataforma.”*

2.2.2. Metodología XP

Se explica en [6], *“La programación extrema o Extreme Programming (XP) es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, Extreme Programming Explained: Embrace Change (1999).”* Es una metodología ágil que destaca por dar prioridad al desarrollo de software.

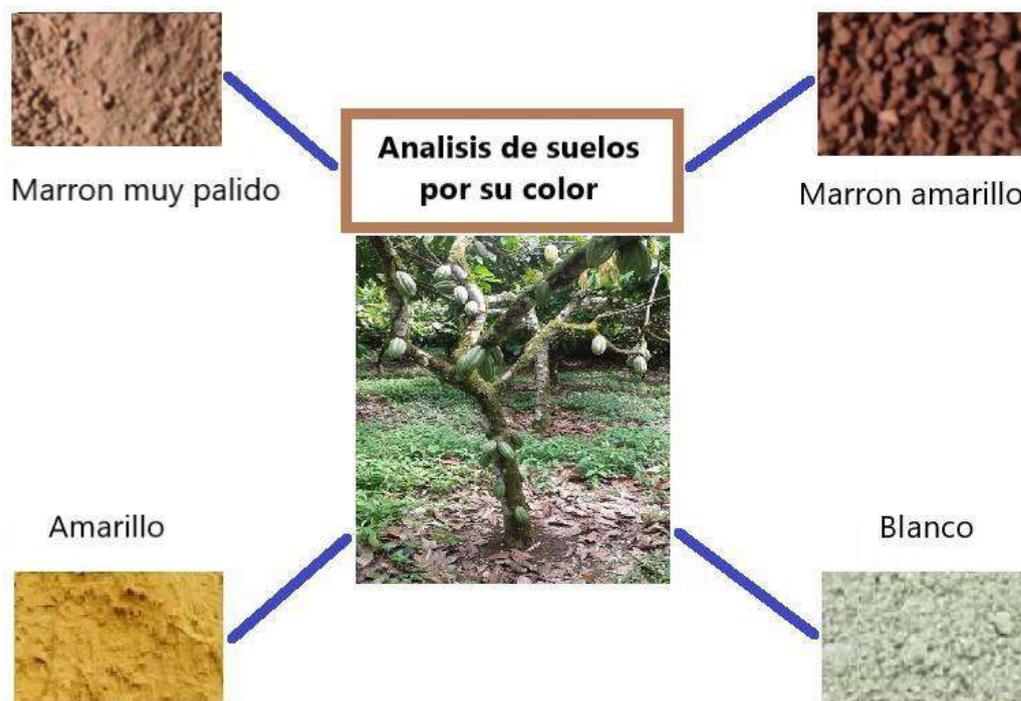
La metodología de programación extrema es muy conocida por sus buenas prácticas en la hora de optimizar el tiempo en el desarrollo del software, así se explica en [7], *“En términos de implementación y pruebas, XP atendió el 89% de las prácticas investigadas contra el 11% de Scrum, confirmando así la falta de técnicas de ingeniería en Scrum.”*

2.2.3. Análisis de suelos

Para el desarrollo de esta propuesta tecnológica se realizó una investigación en diversos artículos de revistas científicas.

Un suelo fértil puede ser asociado con la tasa más alta de descomposición de la materia orgánica del suelo (MOS), que ocasiona un retorno más rápido de los nutrientes orgánicos hacia las plantas, así es explicado en este estudio [8].

Ilustración 1: Clasificación de suelos por su color



Existen diversos tipos de tierras de cultivo, como lo mencionan en [9], tierras de riego y tierras de secano. Las tierras de riego por lo general se utilizan en cultivos de plantas con raíces pequeñas como maíz, cebolla, papa y zanahoria, las mismas que se deben regar periódicamente para evitar que se sequen. Las tierras de secano sirven para sembrar plantas como el trigo, frijoles y mijo al no requerir de tanto riego debido que no necesitan tanta agua para crecer.

En el artículo [10], las causas por las que el suelo agrícola se podría contaminar y no ser apto para sembrar sería por el bajo grado de minerales de magnesita y la abundancia de materiales de desecho que en ciertos casos están vinculados con actividades de minería en la zona. Además, se menciona que en un estudio del suelo las muestras comunes a recolectar son físicas como la textura y la humedad del suelo, así como químicas por ejemplo el PH, el carbono orgánico y el nitrógeno.

En este estudio [11], se menciona que los suelos franco arenosos que por lo general están compuestos de arena con humus tienen una mayor tasa de infiltración que los suelos compuestos de limo y arcilla, esto significa que este último tipo de suelo demora demasiado en absorber

agua mientras que en el primer tipo de suelo el agua ingresa rápidamente permitiendo que la planta mediante sus raíces absorba los nutrientes del suelo sin inconvenientes.

Como se ha mencionado antes el estudio del suelo es fundamental para conocer si el suelo es factible o no para la siembra de plantas, se argumenta [12], que el suelo analizado varía mucho de textura, había un alto nivel de PH y poco de limo, arcilla, arena y franco, por lo que se concluyó que el suelo era salino.

En este estudio de suelo [13], se explica que una tierra puede volverse improductiva debido a la presencia de niveles altos de salinidad y sodicidad, lo cual afectaría la calidad de agua en el riego de las plantas que se pretendan cultivar en la zona.

Se indica en este estudio [14], que los suelos cultivables de campos europeos de Biochar estaban sobrerrepresentados, mientras que los suelos poco profundos y a su vez poco fértiles estaban infrarrepresentados.

Además, se argumenta en este estudio [15], que el fósforo es un tipo de nutriente usado en ciertos fertilizantes de suelos que podría ser más efectivo para que dicho suelo sea más fértil.

En un estudio hecho en una finca de maíz [16], se presenta que la creciente demanda de alimentos y uso de la tierra genera problemas tales como una baja fertilidad del suelo, prácticas agronómicas defectuosas, baja densidad de cultivos y aplicación de fertilizantes.

En este estudio [17], se analiza el nivel de producción del suelo clasificándose en tres categorías que son bajo, medio y alto con respecto al nivel potencial de producción, para su análisis se identificó la calidad y la densidad del suelo agrícola.

La mezcla de tierras para diversos tipos de producción de cultivos ha causado la reducción de los minerales del suelo, por lo que se recomienda una renovación de la calidad del suelo mediante una gestión sostenible de la tierra, así lo menciona este estudio realizado [18].

El presente artículo [19], se describe que la gestión insostenible de los recursos naturales en Benín obstaculiza el crecimiento agrícola, es por ello que se debe dar importancia al análisis en la degradación del suelo y al uso de los fertilizantes en las tierras de cultivo que en el caso del estudio se refiere al algodón.

En un estudio hecho en la República del Congo [20], se presenta que el nivel de fertilidad del suelo no solo depende del rendimiento o productividad del suelo, sino que también depende del riesgo de ataque de plagas y enfermedades a los cultivos, así como las malas prácticas agrícolas y los cambios repentinos en el clima.

Además, en un examen sobre la gestión integrada de la fertilidad del suelo se menciona [21], que dicha integración comprendida por el uso de abono animal y fertilizante NPK mejora la fertilidad del suelo contribuyendo a la sostenibilidad agrícola y al ecosistema.

Se indica en este artículo [22], que la erosión del suelo es una gran amenaza para la sostenibilidad de la agricultura en África y que es suficiente implementar 2 de 3 componentes recomendados por la ONU que son la alteración mínima del suelo y cobertura del suelo.

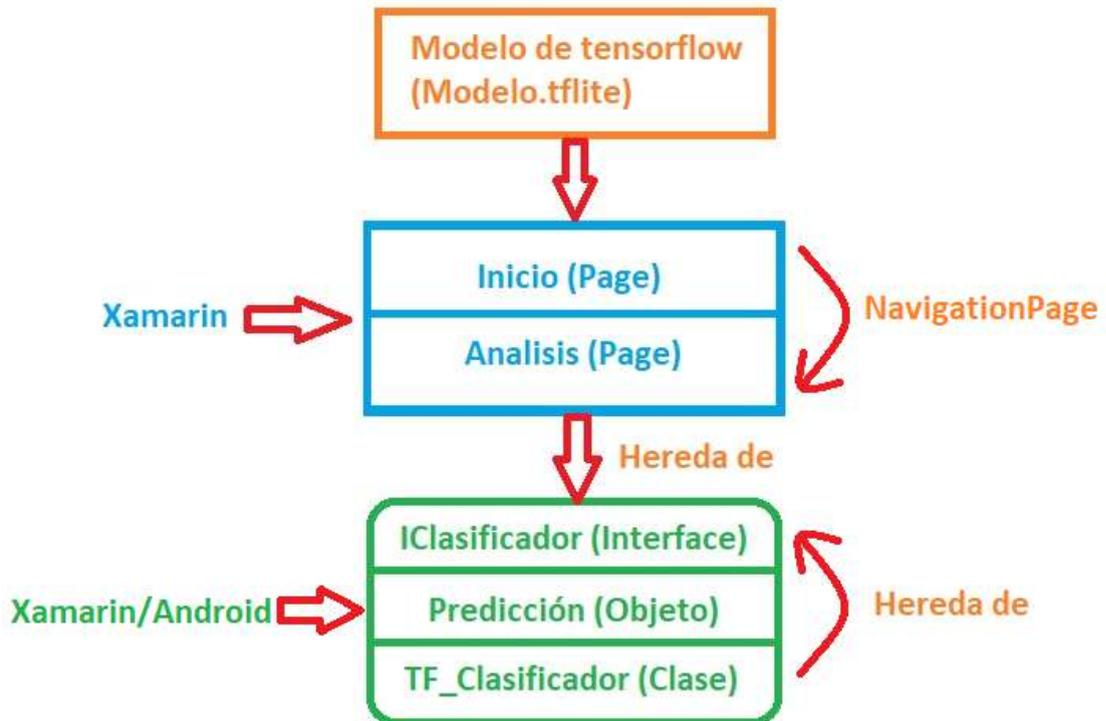
El aumento de tasas de bio-carbón y fertilizantes minerales con estiércol de aves de corral aplicado a un suelo de siembra de cebolla sirvió para evaluar su mayor rendimiento, obteniendo como resultado que en efecto esta combinación de fertilizantes mejora el suelo agrícola, así lo indica este estudio [23].

Se realizó un estudio de los tipos de suelos más fértiles en Aversa en el sur de Italia, dando como resultado que los andosoles son el suelo más extenso y fértil de la zona, seguido de phaeozems y cambisoles, así se detalla en este artículo [24].

En este estudio [25], se explica que el suelo arcilloso de Uzo-Uwani LGA del estado de Enugu es ideal para la siembra de arroz, debido que es un suelo rico en minerales, fértil y buen retenedor de humedad.

2.2.4. Aplicaciones móviles

Ilustración 2: Comunicación de la aplicación móvil



Los avances tecnológicos en los últimos años han sido importantes, en especial las aplicaciones móviles orientadas a solucionar problemas de la salud, las mismas que según los autores del artículo [26], han sido estimadas como un recurso viable para la mejora de calidad y acceso a la asistencia salubre.

Las aplicaciones multiplataforma buscan tener el mismo comportamiento de rendimiento en cualquiera de las plataformas para las cuales está diseñada, a diferencia de las aplicaciones nativas que solo sirven en su mismo entorno de ejecución o sistema operativo nativo. Es por esto que los investigadores [27], contextualizan que el desarrollo de aplicaciones móviles con entornos de trabajo como Xamarin y React Native son muy recomendables para crear aplicaciones multiplataforma, siendo el primero más efectivo que el segundo en la ejecución de aplicaciones de Android a diferencia del segundo marco de trabajo que ofrece mejor rendimiento en aplicaciones de IOS.

Los desarrolladores de aplicaciones móviles [28], crearon una aplicación usando Xamarin con el propósito de recopilar datos de muestras de nacidos digitales para plantas y hongos.

En un segmento del artículo [29], se definen ventajas para los desarrolladores y los usuarios de aplicaciones móviles entre las cuales se identifican las siguientes: Uso de un mismo lenguaje de programación C#, es código multiplataforma, funciona de forma nativa, incluye un gestor para subir la aplicación a las tiendas online.

Según los investigadores del artículo [30], se recomienda utilizar entornos de trabajo multiplataforma como lo es Xamarin para lo que son pequeños equipos de desarrollo, así como para equipos que son proclives a cambios como por ejemplo proyectos de investigación.

Xamarin se puede utilizar para dar solución a distintos problemas como lo es la creación de un agente de supervisión que dentro de su arquitectura básica incluiría módulos como el de recopilación de datos, base de datos local e intercambiador TCP/IP, así lo describen los autores del artículo [31].

Actualmente se han propuesto varios tipos de aplicaciones móviles que utilizan imágenes basadas en teléfonos inteligentes o SBI por sus siglas en inglés. En este artículo [32], se menciona que en la última década se han implementado comúnmente unas SBI para diagnóstico de muestras en vivo para la detección de microscopía y micro fluidos.

Los autores de la investigación [33], explican que la creación de dispositivos de control de la salud, la reducción de tamaño de estos y el incremento de la disponibilidad de redes inalámbricas han dado paso al auge de las aplicaciones móviles orientadas al diagnóstico de varios problemas de la salud.

En una evaluación de efectividad hacia una aplicación móvil orientada a la medición precisa de ingesta dietética, se pudo concluir que los métodos intuitivos y tecnológicos de captura de datos dietéticos se acomodan correctamente a las necesidades del usuario, demostrando así que sus resultados son equivalentes a los que se podrían obtener con métodos tradicionales. Así se concluyó este artículo investigativo [34].

En un estudio observacional documentado en el artículo [35], se argumenta que el uso de una aplicación móvil usada para medir las respuestas a un tratamiento del síndrome del intestino irritable es altamente adherible al paciente, por lo que la aplicación cumple satisfactoriamente con su objetivo.

En este artículo [36], los autores presentan una multiplataforma que han desarrollado tanto en dispositivos móviles, escritorio y en la nube que tiene la función de combinar información sensorial de diversas fuentes para realizar un diagnóstico válido mediante protocolos adaptables mismos que son la ventaja del proyecto.

Las aplicaciones móviles ofrecen muchas ventajas y soluciones a problemas, así como la solución planteada en este artículo [37], que describen que se desarrolló una aplicación móvil basada en la nube para el tratamiento y monitoreo de pacientes que sufren de la parálisis de BELL.

Es interesante saber que las aplicaciones móviles son versátiles a la hora de utilizarlas, así como mencionan los autores del artículo [38], con un ejemplo sobre que cada vez más los estudiante de medicina utilizan aplicaciones móviles para la búsqueda de material de referencia para sus actividades profesionales.

Un ejemplo más de innovación en el uso de aplicaciones móviles con redes neuronales es el de este artículo [39], que mencionan el uso de una app para reconocer si el paciente posee glaucoma y como resultado se tuvo una precisión del 76% y muy pocos falsos positivos con desconocidos.

En la actualidad existen muchos proyectos interesantes como el del artículo de a continuación [40], que consisten en una app para reconocer placas bacterianas en los dientes del usuario mediante el uso de una herramienta de fluorescencia inducida por luz LIF que envía esos datos al celular mediante una API subida a Amazon Web Services.

El uso de una aplicaciones móvil vinculada al sector de la salud es muy útil e importante debido que en ella es posible reportar el estado de salud de cada ciudadano, sus últimos contactos y sus posibles síntomas a fin de identificar los lugares potenciales en donde se encuentra circulando el virus COVID 19, así se explica en el artículo siguiente [41].

Además de la utilización de una aplicación móvil para detectar problemas en pacientes o registrar datos de ellos también se la puede utilizar para agendar recordatorios de citas por especialista demostrando que con el uso de este aplicativo la asistencia de usuarios a las citas médicas es del 91% mientras que con los que no la usan es del 80%, así se concluyó en el artículo [42].

2.3. Metodología

Para la construcción de este proyecto tecnológico se pensó en utilizar la metodología XP porque el trabajo se desarrolla enfocándose principalmente en las buenas prácticas del desarrollo de software, el control de actividades en secuencia y las revisiones periódicas que forman parte esencial de esta metodología en particular.

2.4. Objetivos del prototipo.

2.4.1. Objetivo general

Desarrollar una aplicación móvil de reconocimiento de suelos usando Deep Learning

2.4.2. Objetivos específicos

- Codificar la aplicación para sistemas Android mediante Xamarin.
- Emplear las etapas de la metodología XP durante el proceso de desarrollo.
- Definir los requerimientos funcionales y no funcionales de la aplicación propuesta.
- Diseñar los bocetos, interfaces y la descripción del proceso interno del aplicativo usando herramientas CASE.
- Desarrollar las pruebas unitarias y de evaluación de calidad del software.

2.5. Fase de planificación

Esta fase es muy importante en la metodología XP debido que en dicha fase se presentan los procesos que se involucran en el flujo de trabajo del proyecto. A continuación, se indica la definición de roles, el cronograma de actividades, las historias de usuario y el análisis del sistema.

2.5.1. Definición de roles

Tabla 4: Definición de roles

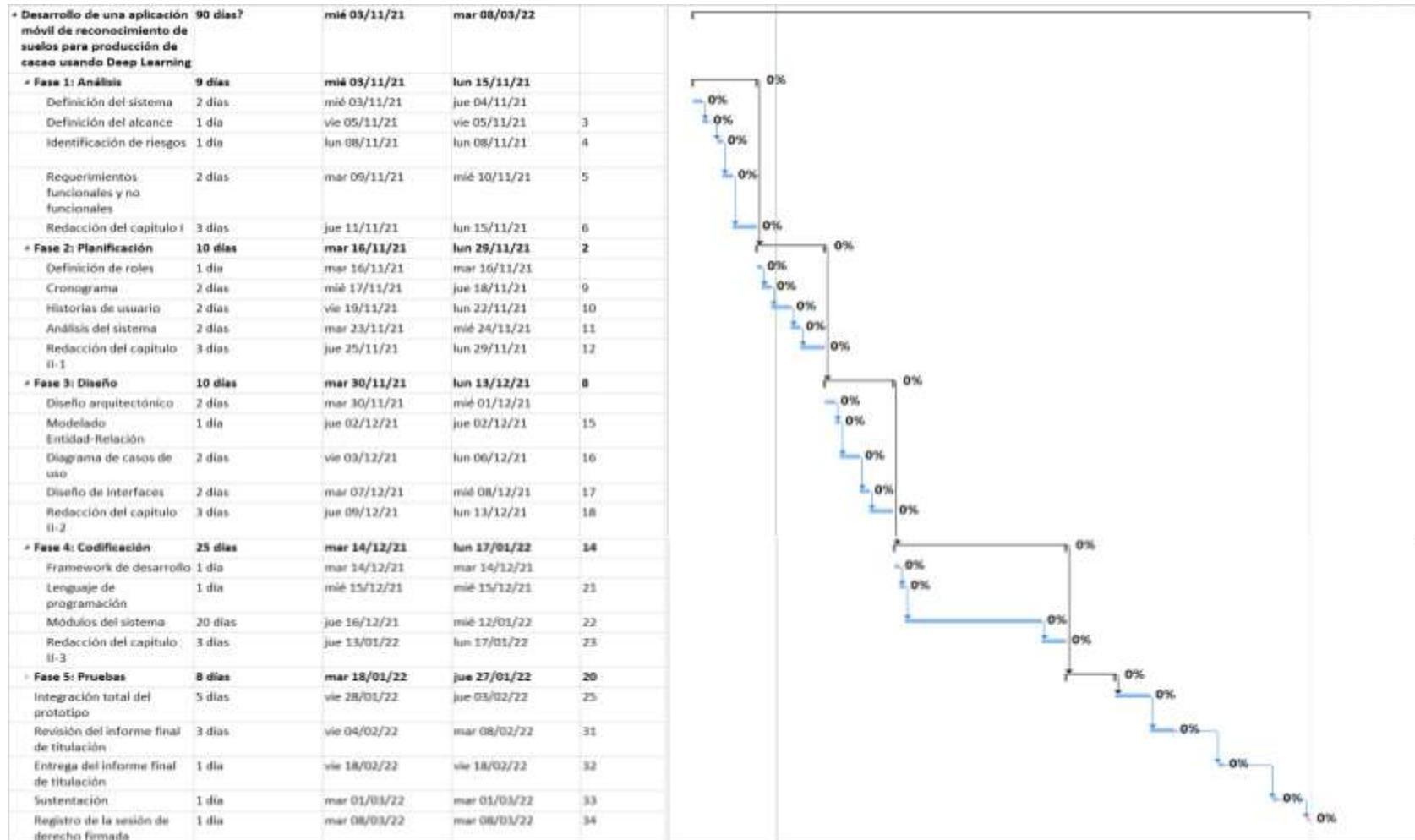
Perfil de Cargo				
Nombre del cargo:		Desarrollador		
Departamento:	Desarrollo	Sección:	---	
Elaborado por:	Yasser Ortiz T.	Ejerce supervisión sobre:	---	
Descripción General:		Personal responsable del desarrollo de la aplicación móvil		
Descripción del cargo				
Funciones/Responsabilidades				
1	Diseño de bocetos de la aplicación móvil			
2	Construcción de las interfaces involucradas en el software			
3	Desarrollo del prototipo propuesto			
4	Evaluación del desempeño en el aplicativo elaborado			
Análisis de competencias necesarias para el puesto				
Grado académico y área de estudio				
Edu. Básica Completa	X	General unificado		
Edu. Media Completa	X	General unificado		
Edu. Técnico Profesional	-	---		
Edu. Universitaria	X	Ingeniería de sistemas		
Experiencia				
	Exp. Laboral	Exp. En el campo	Exp. En la especialidad	Exp. Rubro / negocio
Meses:				
Años:				

Observaciones:						
Idiomas						
	Frecuencia con que lo usa			Nivel de Manejo		
Idioma	Diaria	Semanal	Mensual	Básico	Medio	Alto
Inglés		X		X		
Español	X					X
Programas computacionales						
Visual Studios Community 2019				Enterprise Architect		
Balsamiq Wireframes						

Fuente: Elaboración propia

2.5.2. Cronograma

Ilustración 3: Cronograma de actividades



Fuente: Elaboración propia

2.5.3. Historias de usuario

En este apartado se colocará una lista de fichas en las cuales se indica los requerimientos o funciones nuevas solicitadas por el usuario o el cliente. Estas fichas se las conoce como historias de usuario y se las utiliza comúnmente para describir tareas que la persona solicitante desea que se añada al software en desarrollo.

Tabla 5: Historia de usuario menú inicial

Historia de Usuario	
Número: 1	Usuario: Cliente
Nombre de historia: Opciones del menú inicial	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Días estimados: 2	Iteración asignada: 1
Programador responsable: Yasser Ortiz T.	
Descripción: El usuario puede elegir entre dos opciones para analizar el suelo: La primera opción es para seleccionar una imagen previamente guardada en la galería del celular. La segunda opción es para activar la cámara del celular y permitir tomar fotografías del suelo en ese instante.	
Observaciones: Se puede leer una descripción del funcionamiento del aplicativo en el menú inicial.	

Fuente: Elaboración propia

Tabla 6: Historia de usuario seleccionar imagen de galería

Historia de Usuario	
Número: 2	Usuario: Cliente
Nombre de historia: Seleccionar imagen de galería	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Días estimados: 4	Iteración asignada: 2
Programador responsable: Yasser Ortiz T.	
Descripción: El usuario puede seleccionar una imagen de su galería de fotos para posteriormente analizarla en la ventana que le aparecerá	

automáticamente luego de confirmar dicha imagen, caso contrario puede regresar al menú inicial.

Observaciones:

Se puede abortar la acción regresando al menú inicial.

Fuente: Elaboración propia

Tabla 7: Historia de usuario tomar fotografía con cámara del celular

Historia de Usuario	
Número: 3	Usuario: Cliente
Nombre de historia: Tomar fotografía con cámara del celular	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Días estimados: 4	Iteración asignada: 2
Programador responsable: Yasser Ortiz T.	
Descripción: El usuario puede tomar una fotografía con la cámara de su celular para posteriormente analizarla en la ventana que le aparecerá automáticamente luego de confirmar dicha foto, caso contrario puede regresar al menú inicial.	
Observaciones: Se puede abortar la acción regresando al menú inicial.	

Fuente: Elaboración propia

Tabla 8: Historia de usuario ventana analizar

Historia de Usuario	
Número: 4	Usuario: Cliente
Nombre de historia: Opciones de la ventana analizar	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Días estimados: 2	Iteración asignada: 1
Programador responsable: Yasser Ortiz T.	
Descripción: El usuario puede observar la foto elegida según la opción seleccionada anteriormente, además podrá regresar al menú inicial por si desea cambiar de opción, de igual manera si está seguro de la foto puede analizarla y con ello se mostrará un resultado que indica el tipo de suelo y su porcentaje de certeza.	

Observaciones:

Se puede ver una lista del resto de suelos con el porcentaje de certeza que obtuvieron con el análisis.

Fuente: Elaboración propia

Tabla 9: Historia de usuario analizar imagen o foto

Historia de Usuario	
Número: 5	Usuario: Cliente
Nombre de historia: Analizar imagen o foto	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Días estimados: 6	Iteración asignada: 3
Programador responsable: Yasser Ortiz T.	
Descripción: El usuario puede analizar la imagen y al finalizar el análisis se mostrará en la parte inferior el resultado que indicará el tipo de suelo y el porcentaje de certeza del mismo.	
Observaciones: Se puede abortar la acción regresando al menú inicial.	

Fuente: Elaboración propia

Tabla 10: Historia de usuario navegación entre ventanas

Historia de Usuario	
Número: 6	Usuario: Cliente
Nombre de historia: Navegación entre ventanas	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Días estimados: 2	Iteración asignada: 1
Programador responsable: Yasser Ortiz T.	
Descripción: El usuario puede viajar entre las ventanas del menú inicial y analizar, siempre y cuando se quiera analizar una imagen o cancelar la decisión.	
Observaciones: El botón de regreso se encuentra en la parte superior izquierda de la barra de título de la ventana analizar.	

Fuente: Elaboración propia

2.5.4. Análisis del sistema

A continuación, se presenta el análisis del sistema que está conformado por el plan de publicación, el plan de duración de iteraciones, el plan de entrega y el cálculo de la velocidad del proyecto.

2.5.4.1. Plan de publicaciones

Este plan sirve para organizar las actividades a elaborar tomando en cuenta las prioridades definidas en las historias de usuario, esto se hace así porque el desarrollo de ciertas actividades depende de la existencia previa de otras.

En dicho plan se establecerán algunos parámetros a tomar en cuenta, para ello se realizará una tabla que conste de las siguientes columnas: Número de tarea, nombre de las tareas, nombre de las historias de usuario, la prioridad de desarrollo, el riesgo de desarrollo y el tiempo de duración de cada actividad dada en días.

Tabla 11: Plan de publicaciones

N.º	Tarea	Historia de usuario	Prioridad	Riesgo	Días estimados
1	Ventana principal	Opciones del menú inicial	Alta	Alto	2
		Seleccionar imagen de galería	Alta	Alto	4
		Tomar fotografía con cámara del celular	Alta	Alto	4
2	Navegación entre ventanas	Navegación entre ventanas	Baja	Bajo	2
3	Ventana secundaria	Opciones de la	Alta	Alto	2

		ventana analizar			
		Analizar imagen o foto	Alta	Alto	6

Fuente: Elaboración propia

2.5.4.2. Plan de duración de iteraciones

Una vez definido el plan de publicaciones y las historias de usuario se debe ahora especificar qué actividades se realizarán durante la ejecución de cada iteración asignada previamente, mencionado lo anterior, se establece que en la primera iteración se desarrolla las interfaces gráficas del prototipo, en la segunda iteración se elabora la función de seleccionar una imagen desde la galería del celular, así como la función de obtener una fotografía con la cámara del dispositivo y como tercera iteración se codifica la función de analizar el suelo puesto que esta actividad utiliza librerías externas para su correcto funcionamiento.

Tabla 12: Plan de duración de iteraciones

Iteración	Orden de las historias de usuario	Duración de la iteración
Primera Iteración	Opciones del menú inicial	6 días
	Navegación entre ventanas	
	Opciones de la ventana analizar	
Segunda Iteración	Seleccionar imagen de galería	8 días
	Tomar fotografía con cámara del celular	
Tercera Iteración	Analizar imagen o foto	6 días

Fuente: Elaboración propia

2.5.4.3. Plan de entrega

En este apartado se debe agrupar las historias de usuario con sus respectivas iteraciones, a esto se le suma las fechas respectivas de inicio y fin de cada iteración sin olvidar que estas deben estar en

coordinación con las fechas estipuladas en el cronograma establecido anteriormente.

En la tabla siguiente se describe las fechas de inicio y fin para cada iteración, considerando solo los días laborables por semana:

Tabla 13: Plan de entrega

Iteración	Fecha de entrega	
	Inicio	Fin
Primera Iteración	16/12/2021	23/12/2021
Segunda Iteración	24/12/2021	04/01/2022
Tercera Iteración	05/01/2022	12/01/2022

Fuente: Elaboración propia

2.5.4.4. Velocidad del proyecto

En este apartado se calcula la velocidad del proyecto, el mismo que es un valor aproximado de la cantidad promedio de historias de usuario que se deben realizar por cada iteración propuesta en el proyecto.

Hay que considerar que no todas las actividades duran lo mismo y esto es por las distintas dificultades que existen al realizar cada una de ellas durante cada iteración.

Tabla 14: Velocidad del proyecto

Descripción	Primera Iteración	Segunda Iteración	Tercera Iteración
Historia de usuario	3	2	1
Tiempo (Días)	6	8	6

Fuente: Elaboración propia

La fórmula para realizar el cálculo de la velocidad del proyecto es la siguiente:

$$Velocidad\ del\ proyecto = \frac{Número\ de\ historias\ de\ usuario}{Número\ de\ iteraciones}$$

$$Velocidad\ del\ proyecto = \frac{3+2+1}{3} = \frac{6}{3}$$

$$Velocidad\ del\ proyecto = 2\ historias\ de\ usuario/iteración$$

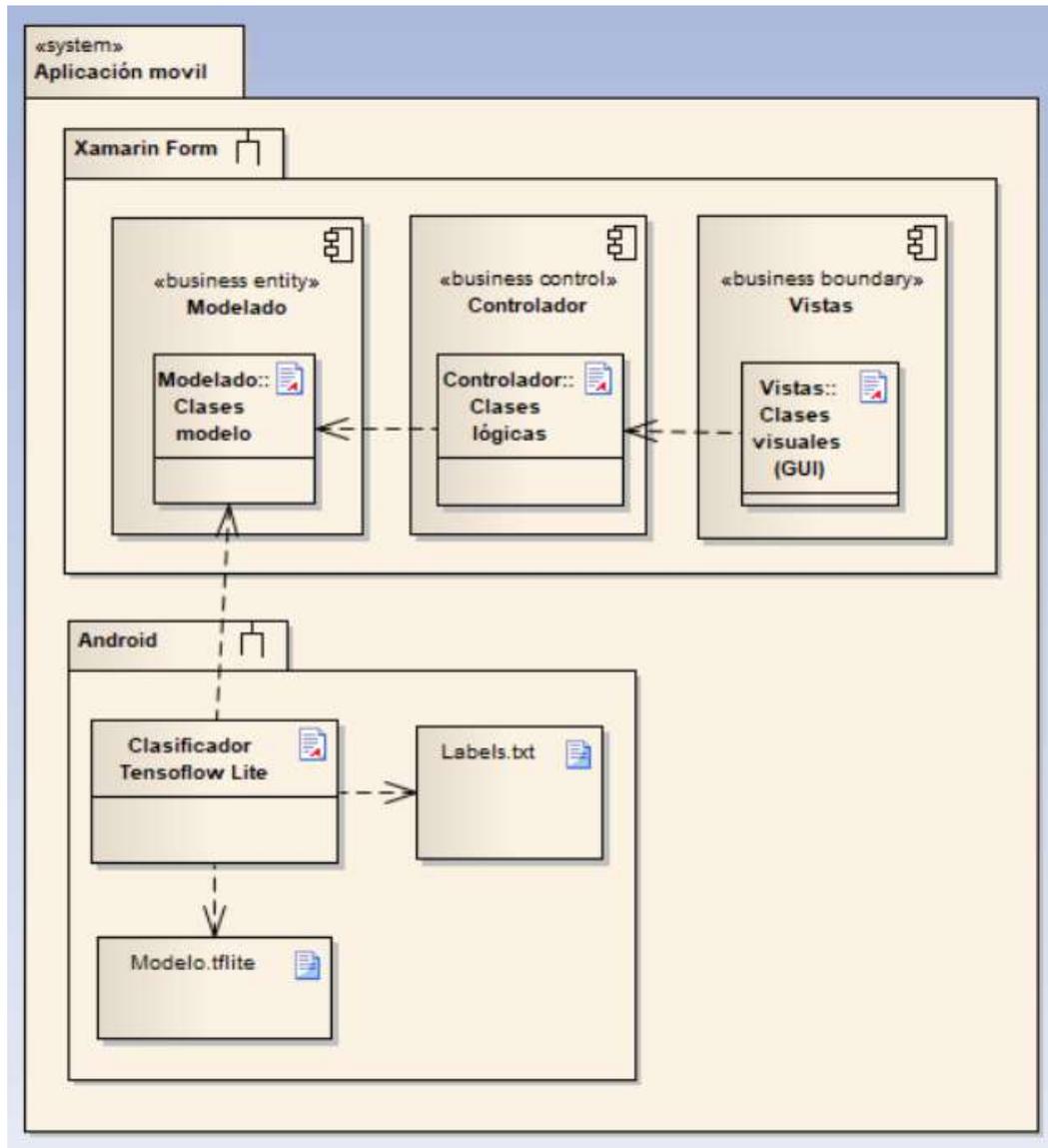
2.6. Diseño del prototipo.

Para conocer el comportamiento y funcionalidad del prototipo se pensó en ilustrar un diseño arquitectónico que facilite la comprensión de los aspectos antes mencionados, además de mostrar la clase encargada de recibir los datos del modelo para ser vistos en la aplicación móvil con un diagrama de entidad-relación y un modelo relacional. De igual manera para ver el comportamiento del usuario con el aplicativo se consideró usar diagramas de caso de uso, así como presentar las interfaces gráficas para ver el resultado de la apariencia visual del proyecto.

2.6.1. Diseño arquitectónico

El aplicativo desarrollado opera entre la comunicación de tres capas, la primera capa es la denominada vista que contiene todas las interfaces gráficas del proyecto y se comunica con la segunda capa conocida como controlador que contiene toda la parte lógica del software, además requiere de la comunicación con la tercera capa llamada modelo que es la encargada de encapsular los datos recibidos para posteriormente ser procesados por la capa controlador que a su vez envía dichos datos a la capa vista para que esta muestre la información resultante de manera clara al usuario final o cliente.

Ilustración 4: Diseño arquitectónico del prototipo

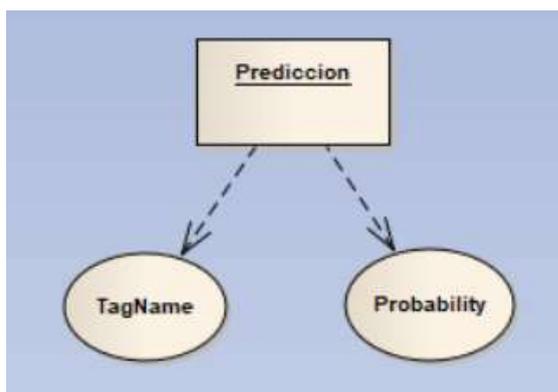


Fuente: Elaboración propia

2.6.2. Modelado entidad-relación y relacional

En la aplicación móvil se utiliza una clase con dos campos o propiedades y un constructor que servirán para modelar los datos tales como la probabilidad de certeza, el nombre del tipo de suelo y la obtención de la información. Cuando una clase se configura de dicha manera se la llama objeto o modelo y para dar mayor entendimiento de su funcionamiento se ilustrarán a continuación los modelados propuestos:

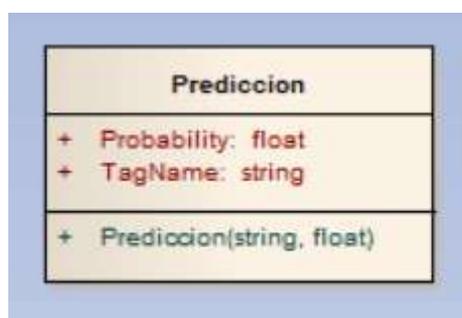
Ilustración 5: Modelo entidad-relación



Fuente: Elaboración propia

Prediccion (Probability, TagName)

Ilustración 6: Modelo relacional



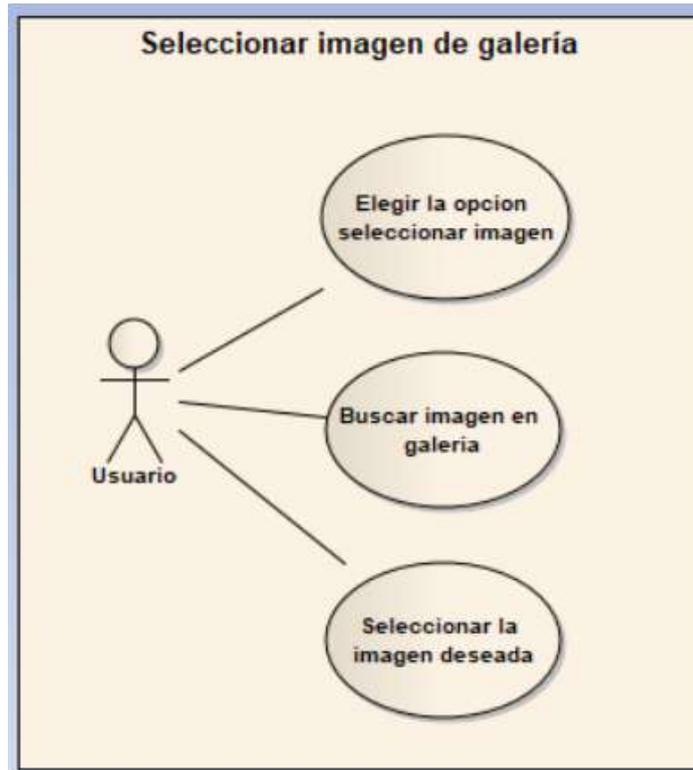
Fuente: Elaboración propia

Es importante mencionar que el diagrama entidad relación y el modelo relacional únicamente se los realizó para identificar la estructura de la clase que recibirá los datos del modelo.tflite a leer en el clasificador TensorFlow, esto debido que en este proyecto no se utiliza una base de datos como tal.

2.6.3. Diagrama de casos de uso

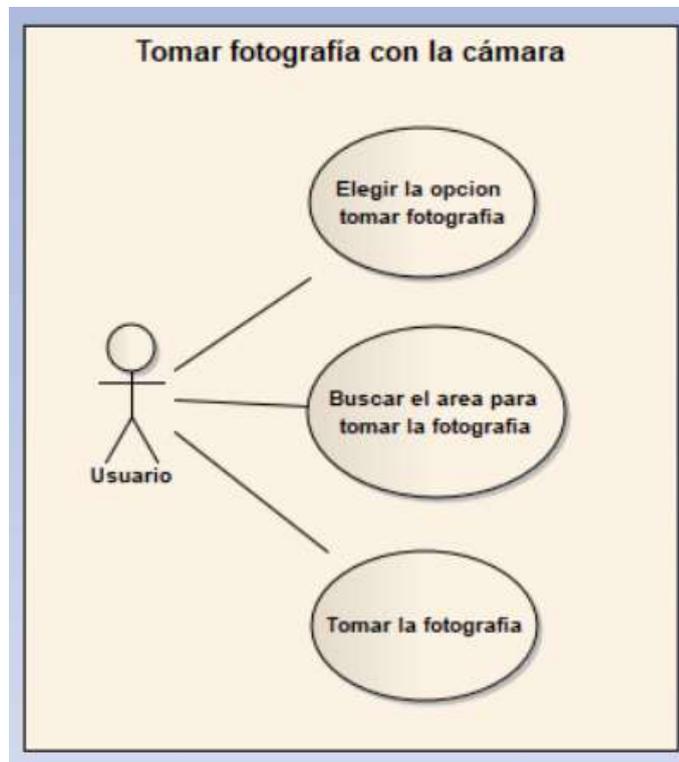
En este apartado se representa el comportamiento que tiene el usuario con el uso de las funciones de la aplicación móvil para ello se diseña el diagrama de casos de uso vinculado a cada proceso que el usuario disponga, este diagrama contiene elementos tales como el actor, el caso de uso y la asociación misma que puede ser de algunos tipos.

Ilustración 7: Caso de uso seleccionar imagen de galería



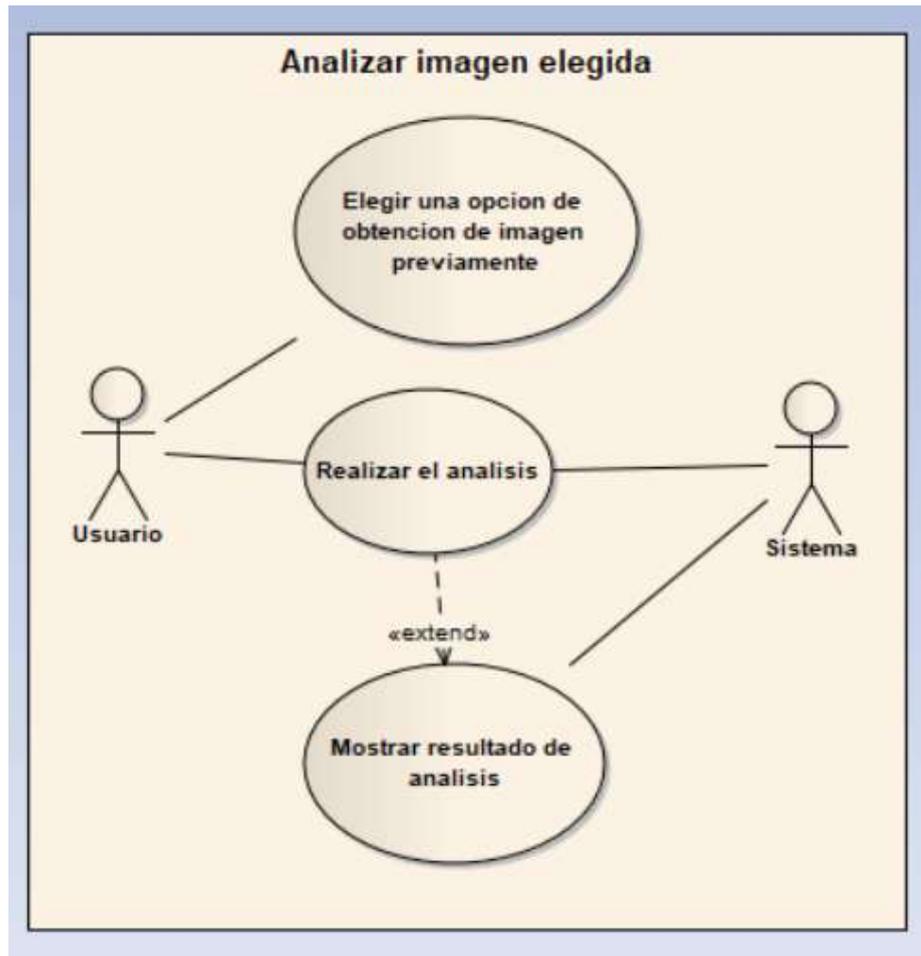
Fuente: Elaboración propia

Ilustración 8: Caso de uso tomar fotografía con la cámara



Fuente: Elaboración propia

Ilustración 9: Caso de uso analizar imagen elegida

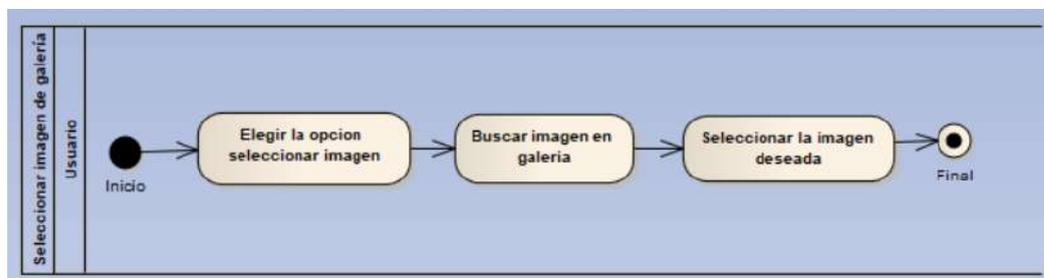


Fuente: Elaboración propia

2.6.4. Diagramas de actividad

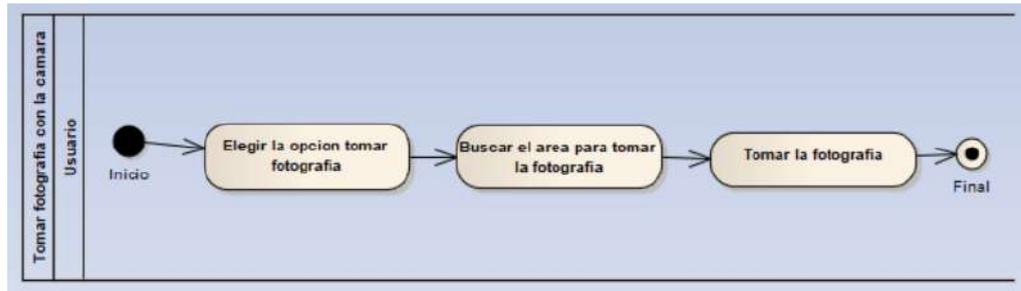
Se establece el diseño del diagrama de actividades de la aplicación móvil que indica el comportamiento lógico que esta tiene.

Ilustración 10: Diagrama de actividades seleccionar imagen de galería



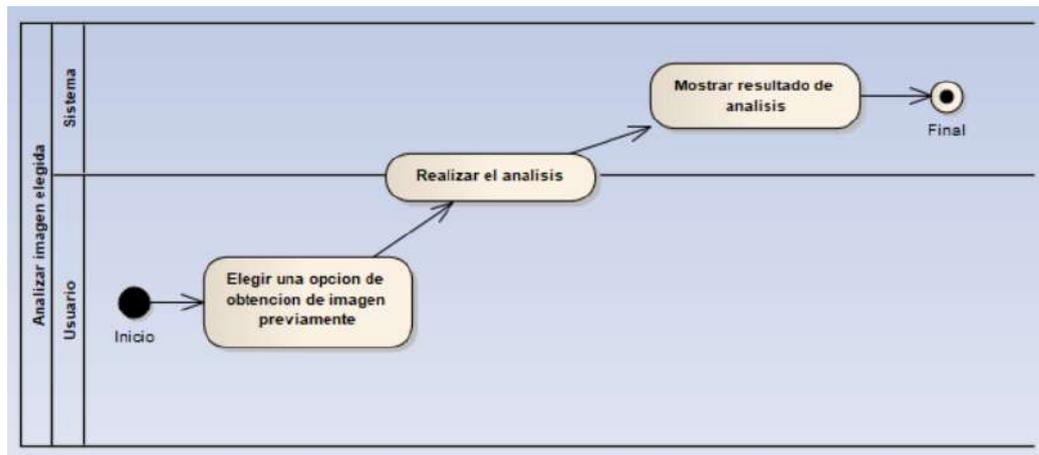
Fuente: Elaboración propia

Ilustración 11: Diagrama de actividades tomar fotografía con la cámara



Fuente: Elaboración propia

Ilustración 12: Diagrama de actividades analizar imagen elegida



Fuente: Elaboración propia

2.6.5. Diseño de interfaces

Para el bosquejado de las interfaces gráficas del aplicativo móvil se consideró usar la herramienta Balsamiq Wireframes. El diseño de las vistas se lo elaboró lo más parecido al resultado final que se pretende tener de las ventanas del prototipo.

A continuación, se muestra los bosquejos de las ventanas involucradas dentro del proyecto y una explicación de su funcionamiento:

- **Ventana menú inicial:** Esta ventana contiene una breve descripción de las funciones que ofrece al usuario, seguido de dos botones que son las opciones para elegir de qué forma obtener la imagen en analizar.

La primera opción consiste en seleccionar una imagen de la galería del dispositivo del celular.

La segunda opción consiste en tomar una fotografía con la cámara del dispositivo del celular.

- **Ventana analizar:** Esta ventana contiene la imagen seleccionada previamente, seguido de un botón para proceder en realizar el análisis, a continuación de lo anterior se muestra un campo de texto que muestra como resultado el tipo de suelo y su probabilidad de certeza, además de una lista de la información del resto de suelos.

Ilustración 13: Diseño de interfaz menú



Fuente: Elaboración propia

Ilustración 14: Diseño de interfaz analizar



Fuente: Elaboración propia

2.7. Ejecución y/o ensamblaje del prototipo.

2.7.1. Framework de desarrollo

El marco de trabajo utilizado para realizar el proyecto es Visual Studios Community 2019, este entorno permite instalar varios módulos para realizar diversas actividades en el desarrollo de software, en este caso se instaló el módulo de Xamarin que ofrece a los programadores que están familiarizados con el lenguaje de programación C# crear aplicaciones móviles de carácter multiplataforma.

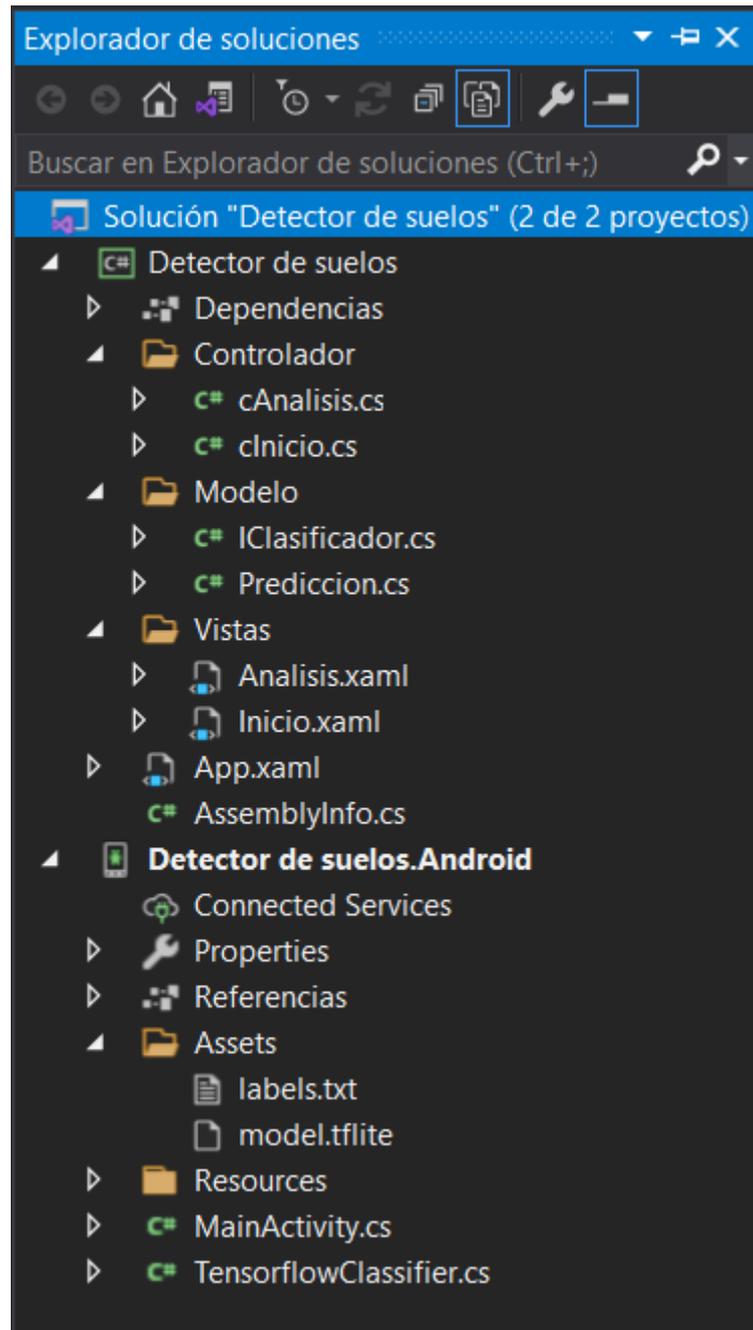
2.7.2. Lenguaje de programación

Como ya se indicó anteriormente el lenguaje de programación utilizado es una combinación de C# para la construcción de la parte lógica y XAML que es un lenguaje de formato que funciona similar a HTML para el diseño de las vistas de la aplicación móvil.

2.7.3. Módulos del sistema

Así como se mencionó en el apartado del diseño arquitectónico la aplicación está construida conforme el patrón de arquitectura MVC (Modelo, Vista Controlador), en donde los objetos de clase usados se almacenan en el directorio Modelo, de igual forma todas las interfaces gráficas involucradas se guardan en el directorio Vistas y por último las clases que contienen todos los métodos y funciones de la lógica de negocios se archivan en el directorio Controlador.

Ilustración 15: Estructura del proyecto

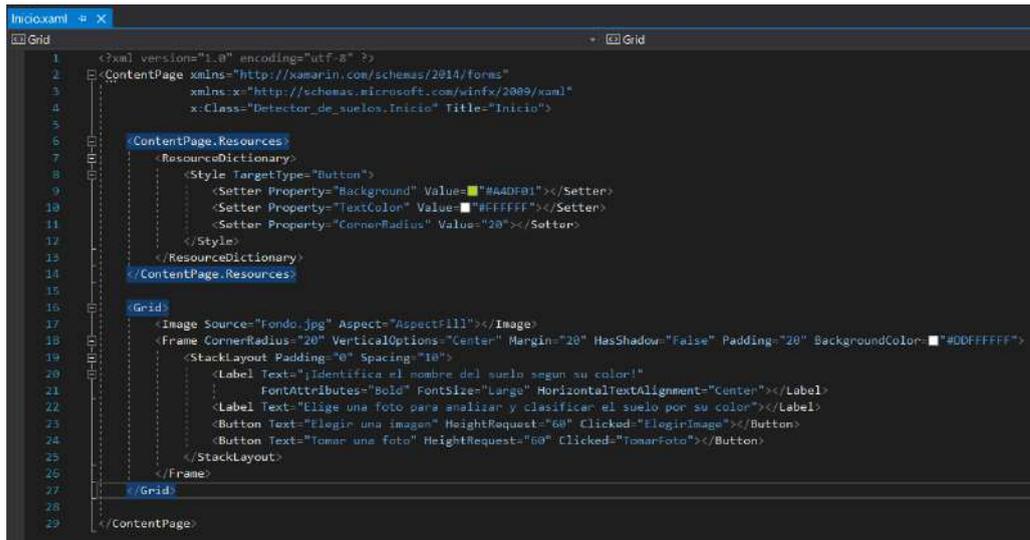


Fuente: Elaboración propia

2.7.3.1. Inicio

Este es el código para la vista de Inicio, en donde se puede ver los componentes usados en ella, está compuesta por un contenedor que abarca una imagen de fondo, así como un Frame transparente que contiene dos etiquetas que sirven como mensajes estáticos y dos botones que son las opciones a elegir.

Ilustración 16: Código de inicio en XAML (Vista)



```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <ContentPage xmlns="http://xamarin.com/schemas/2014/forums"
3   xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4   x:Class="Detector_de_suelos.Inicio" Title="Inicio">
5
6   <ContentPage.Resources>
7     <ResourceDictionary>
8       <Style TargetType="Button">
9         <Setter Property="Background" Value="#A4D0B1"/></Setter>
10        <Setter Property="TextColor" Value="White"/></Setter>
11        <Setter Property="CornerRadius" Value="20"/></Setter>
12      </Style>
13    </ResourceDictionary>
14  </ContentPage.Resources>
15
16  <Grid>
17    <Image Source="Fondo.jpg" Aspect="AspectFill"/></Image>
18    <Frame CornerRadius="20" VerticalOptions="Center" Margin="20" HasShadow="False" Padding="20" BackgroundColor="White">
19      <StackLayout Padding="0" Spacing="10">
20        <Label Text="Identifica el nombre del suelo segun su color!"
21          FontAttributes="Bold" FontSize="Large" HorizontalTextAlignment="Center"/></Label>
22        <Label Text="Elige una foto para analizar y clasificar el suelo por su color"/></Label>
23        <Button Text="Elegir una imagen" HeightRequest="60" Clicked="ElegirImage"/></Button>
24        <Button Text="Tomar una foto" HeightRequest="60" Clicked="TomarFoto"/></Button>
25      </StackLayout>
26    </Frame>
27  </Grid>
28 </ContentPage>
```

Fuente: Elaboración propia

En la clase anterior se vincula el código hecho en XAML hacia la clase siguiente que está en C# que instancia una clase del controlador llamada Inicio.cs que contiene los métodos y funciones necesarios para que sirva esta clase.

Ilustración 17: Código de inicio en C# (Vista)

```
Inicio.xaml.cs
C# Detector de suelos
1 using System;
2 using Xamarin.Forms;
3 using Detector_de_suelos.Controlador;
4 using Xamarin.Essentials;
5
6 namespace Detector_de_suelos
7 {
8     5 referencias
9     public partial class Inicio : ContentPage
10    {
11        public cInicio ob;
12
13        1 referencia
14        public Inicio()
15        {
16            InitializeComponent();
17            ob = new cInicio();
18        }
19
20        0 referencias
21        private async void ElegirImage(object sender, EventArgs e)
22        {
23            try..
24            {
25                PermissionStatus status = await Permissions.CheckStatusAsync<Permissions.Camera>();
26                if (status != PermissionStatus.Granted)
27                {
28                    _ = await Permissions.RequestAsync<Permissions.Camera>();
29                }
30                await ob.ElegirImage(sender, e);
31                await Navigation.PushAsync(new Analisis(ob.foto));
32            } catch (Exception) { }
33        }
34
35        0 referencias
36        private async void TomarFoto(object sender, EventArgs e)
37        {
38            try..
39            {
40                PermissionStatus status = await Permissions.CheckStatusAsync<Permissions.Camera>();
41                if (status != PermissionStatus.Granted)
42                {
43                    _ = await Permissions.RequestAsync<Permissions.Camera>();
44                }
45                await ob.TomarFoto(sender, e);
46                await Navigation.PushAsync(new Analisis(ob.foto));
47            } catch (Exception) { }
48        }
49    }
50 }
```

Fuente: Elaboración propia

A continuación, se muestra el código de la clase controlador de inicio, que contiene los métodos que requiere la clase inicio en C# para hacer funcionar los botones de elegir imagen y tomar foto.

Ilustración 18: Código de inicio en C# (Controlador)

```
clnicio.cs  + X
C# Detector de suelos
1  using System;
2  using Plugin.Media.Abstractions;
3  using Plugin.Media;
4  using System.Threading.Tasks;
5
6  namespace Detector_de_suelos.Controlador
7  {
8      public class cInicio
9      {
10         public MediaFile foto;
11
12         public async Task ElegirImage(object sender, EventArgs e)
13         {
14             try
15             {
16                 _ = await CrossMedia.Current.Initialize();
17                 foto = await CrossMedia.Current.PickPhotoAsync(new PickMediaOptions());
18             }
19             catch (Exception) { }
20         }
21
22         public async Task TomarFoto(object sender, EventArgs e)
23         {
24             try
25             {
26                 _ = await CrossMedia.Current.Initialize();
27                 foto = await CrossMedia.Current.TakePhotoAsync(new StoreCameraMediaOptions()
28                 {
29                     Directory = "Vision",
30                     Name = "Target.jpg"
31                 });
32             }
33             catch (Exception) { }
34         }
35     }
36 }
37
```

Fuente: Elaboración propia

2.7.3.2. Análisis

Este es el código para la vista de Análisis, en donde se puede ver los componentes usados en ella, está compuesta por un contenedor que abarca una imagen de fondo, así como un Frame transparente que contiene la imagen deseada, seguido de un botón para analizar dicha imagen, después hay una etiqueta estática que muestra el resultado y por último una lista con los resultados generales.

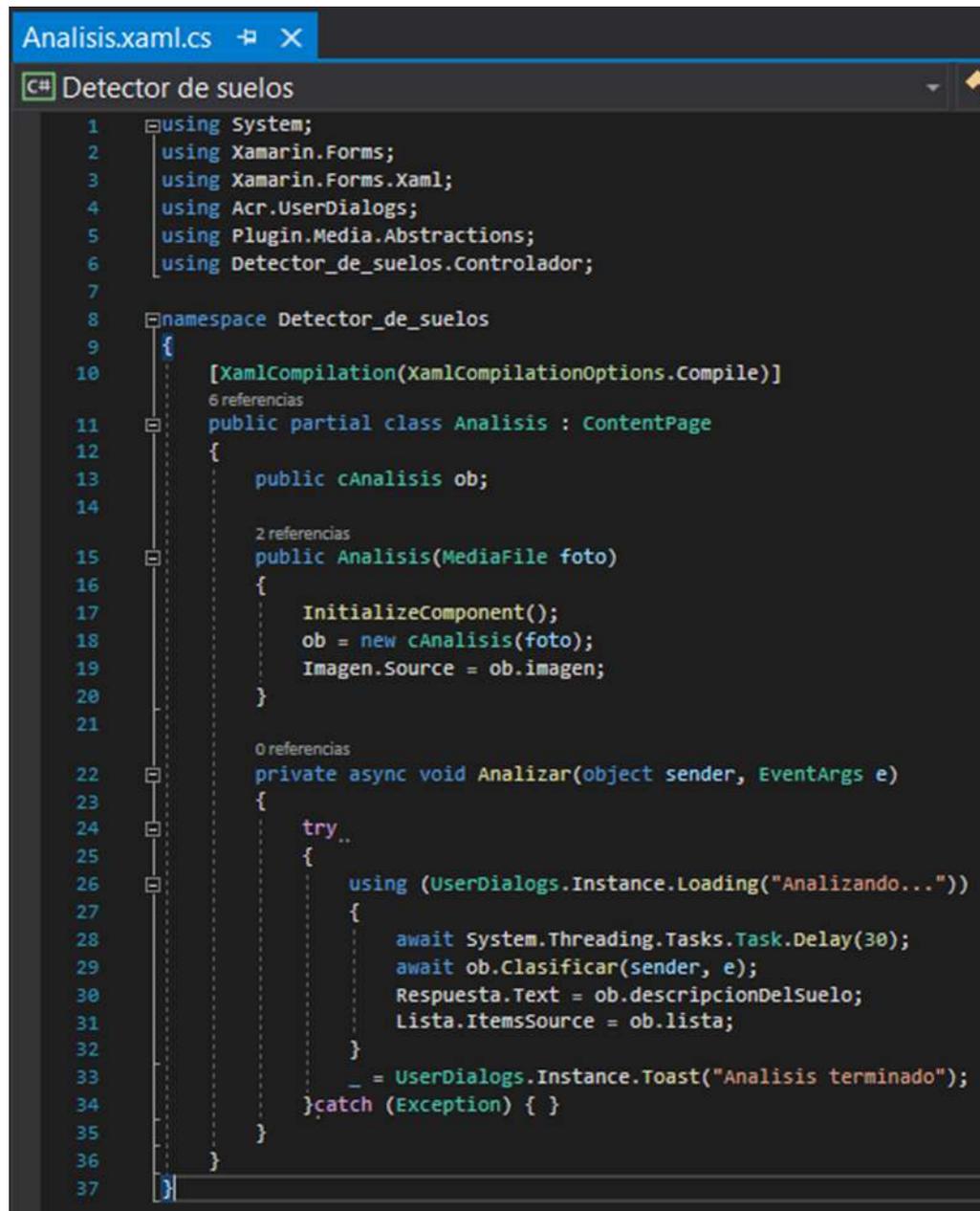
Ilustración 19: Código de análisis en XAML (Vista)

```
Analysis.xaml  X
ContentPage
1 <?xml version="1.0" encoding="utf-8" ?>
2 <ContentPage xmlns="http://xamarin.com/schemas/2014/foras"
3             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4             x:Class="Detector_de_suelos.Analisis" Title="Análisis">
5
6     <ContentPage.Resources>
7         <ResourceDictionary>
8             <Style TargetType="Button">
9                 <Setter Property="Background" Value="#A4DF81"/></Setter>
10                <Setter Property="TextColor" Value="#FFFFFF"/></Setter>
11                <Setter Property="CornerRadius" Value="20"/></Setter>
12            </Style>
13        </ResourceDictionary>
14    </ContentPage.Resources>
15
16    <Grid>
17        <Image Source="Fondo.jpg" Aspect="AspectFill"/></Image>
18        <Frame CornerRadius="20" VerticalOptions="Center" Margin="20" HasShadow="False" Padding="20" BackgroundColor="#D0FFFFFF">
19            <ScrollView Orientation="Vertical">
20                <StackLayout Padding="8" Spacing="10">
21                    <Image x:Name="Imagen" HorizontalOptions="Center"
22                          WidthRequest="200" HeightRequest="200"/></Image>
23                    <Button Text="Analizar" HeightRequest="60" Clicked="Analizar"/></Button>
24                    <Label x:Name="Respuesta" FontSize="20"/></Label>
25                    <ListView x:Name="Lista" Header="">
26                        <ListView.HeaderTemplate>
27                            <DataTemplate>
28                                <Grid>
29                                    <Label Text="LISTA DE PREDICCIONES"
30                                          HorizontalTextAlignment="Center"
31                                          FontAttributes="Bold"
32                                          FontSize="20"
33                                          HorizontalOptions="FillAndExpand"/>
34                                </Grid>
35                            </DataTemplate>
36                        </ListView.HeaderTemplate>
37                        <ListView.ItemTemplate>
38                            <DataTemplate>
39                                <ViewCell>
40                                    <StackLayout Orientation="Horizontal">
41                                        <Label Text="{Binding TagName}"
42                                              HorizontalTextAlignment="Center"
43                                              FontSize="20"
44                                              HorizontalOptions="FillAndExpand"/>
45
46                                        <Label Text=">"
47                                              HorizontalTextAlignment="Center"
48                                              FontSize="20"
49                                              HorizontalOptions="FillAndExpand"/>
50
51                                        <Label Text="{Binding Probability}"
52                                              HorizontalTextAlignment="Center"
53                                              FontSize="20"
54                                              HorizontalOptions="FillAndExpand"/>
55
56                                        <Label Text="%"
57                                              HorizontalTextAlignment="Center"
58                                              FontSize="20"
59                                              HorizontalOptions="FillAndExpand"/>
60                                    </StackLayout>
61                                </ViewCell>
62                            </DataTemplate>
63                        </ListView.ItemTemplate>
64                    </ListView>
65                </StackLayout>
66            </ScrollView>
67        </Frame>
68    </Grid>
ContentPage
```

Fuente: Elaboración propia

En la clase anterior se vincula el código hecho en XAML hacia la clase siguiente que está en C# que instancia una clase del controlador llamada cAnálisis.cs que contiene los métodos y funciones necesarios para que sirva esta clase.

Ilustración 20: Código de análisis en C# (Vista)

The image shows a screenshot of a Visual Studio code editor window titled 'Analisis.xaml.cs'. The code is written in C# and is part of a namespace named 'Detector_de_suelos'. It includes several using statements for System, Xamarin.Forms, Xamarin.Forms.Xaml, Acr.UserDialogs, Plugin.Media.Abstractions, and Detector_de_suelos.Controlador. The main class is 'Analisis', which is a partial class derived from 'ContentPage'. It has a public property 'ob' of type 'cAnálisis'. The constructor 'Analisis(MediaFile foto)' initializes the component, creates a new 'cAnálisis' object, and sets the 'Imagen.Source' property. There is also an asynchronous method 'Analizar(object sender, EventArgs e)' which uses 'UserDialogs.Instance.Loading' to show a progress dialog, delays for 30 units, calls 'ob.Clasificar', updates the 'Respuesta.Text' and 'Lista.ItemsSource', and finally shows a toast message 'Análisis terminado' using 'UserDialogs.Instance.Toast'. The code is wrapped in a try-catch block to handle exceptions.

```
1 using System;
2 using Xamarin.Forms;
3 using Xamarin.Forms.Xaml;
4 using Acr.UserDialogs;
5 using Plugin.Media.Abstractions;
6 using Detector_de_suelos.Controlador;
7
8 namespace Detector_de_suelos
9 {
10     [XamlCompilation(XamlCompilationOptions.Compile)]
11     public partial class Analisis : ContentPage
12     {
13         public cAnálisis ob;
14
15         public Analisis(MediaFile foto)
16         {
17             InitializeComponent();
18             ob = new cAnálisis(foto);
19             Imagen.Source = ob.imagen;
20         }
21
22         private async void Analizar(object sender, EventArgs e)
23         {
24             try..
25             {
26                 using (UserDialogs.Instance.Loading("Analizando..."))
27                 {
28                     await System.Threading.Tasks.Task.Delay(30);
29                     await ob.Clasificar(sender, e);
30                     Respuesta.Text = ob.descripcionDelSuelo;
31                     Lista.ItemsSource = ob.lista;
32                 }
33                 _ = UserDialogs.Instance.Toast("Análisis terminado");
34             } catch (Exception) { }
35         }
36     }
37 }
```

Fuente: Elaboración propia

A continuación, se muestra el código de la clase controlador de análisis, que contiene los métodos que requiere la clase análisis en C# para hacer funcionar el botón de analizar.

Ilustración 21: Código de análisis en C# (Controlador)

```
cAnalysis.cs
C# Detector de suelos
Detector_de_suelos.Cont

1 using System;
2 using System.Linq;
3 using Xamarin.Forms;
4 using Plugin.Media.Abstractions;
5 using System.IO;
6 using System.Collections.Generic;
7 using System.Threading.Tasks;
8
9 namespace Detector_de_suelos.Controlador
10 {
11     3 referencias
12     public class cAnalisis
13     {
14         public MediaFile _foto;
15         public string descripcionDelSuelo;
16         public List<Prediccion> lista;
17         public ImageSource imagen;
18
19         1 referencia
20         public cAnalisis(MediaFile foto)
21         {
22             _foto = foto;
23             imagen = ImageSource.FromFile(_foto.Path);
24         }
25
26         1 referencia
27         public async Task Clasificar(object sender, EventArgs e)
28         {
29             try
30             {
31                 var stream = _foto.GetStream();
32                 var memoryStream = new MemoryStream();
33                 stream.CopyTo(memoryStream);
34                 var bytes = memoryStream.ToArray();
35
36                 DependencyService.Get<IClasificador>().ClassificationCompleted += Classifier_ClassificationCompleted;
37                 await DependencyService.Get<IClasificador>().Classify(bytes);
38             }
39             catch (Exception) { }
40         }
41
42         2 referencias
43         public void Classifier_ClassificationCompleted(object sender, ClassificationEventArgs e)
44         {
45             try
46             {
47                 var sortedList = e.Predictions.OrderByDescending(x => x.Probability);
48                 var arreglo = sortedList.ToList();
49
50                 if (arreglo == null)
51                 {
52                     return;
53                 }
54
55                 descripcionDelSuelo = $"El terreno se considera {arreglo[0].YagName} con una certeza del {arreglo[0].Probability}%";
56                 lista = arreglo;
57
58                 var classifier = (IClasificador)sender;
59                 classifier.ClassificationCompleted -= Classifier_ClassificationCompleted;
60             }
61             catch (Exception) { }
62         }
63     }
64 }
```

Fuente: Elaboración propia

2.7.3.3. Consumo del modelo

Para el consumo del modelo se requirió crear tres clases que sirvieran para interactuar con la clase cAnalisis, la primera es una clase modelo llamada Prediccion que guarda los atributos del modelo consumido.

Ilustración 22: Código de Predicción en C#

```
Prediccion.cs  + X
C# Detector de suelos  Detector_de_suelos.Prediccion
1  using System;
2  using System.Collections.Generic;
3
4  namespace Detector_de_suelos
5  {
6      0 referencias
7      public class PredictionResult
8      {
9          0 referencias
10         public List<Prediccion> Predictions { get; set; }
11     }
12
13     -referencias
14     public class Prediccion
15     {
16         -referencias
17         public float Probability { get; set; }
18
19         -referencias
20         public string TagName { get; set; }
21
22         1 referencia
23         public Prediccion(string tagName, float probability)
24         {
25             TagName = tagName;
26             Probability = (float)Math.Round(probability * 100, 2);
27         }
28     }
29 }
```

Fuente: Elaboración propia

La segunda es una clase de tipo interface llamada IClasificador que servirá para comunicar los métodos de la clase TensorFlowClasifier con la clase cAnálisis, esto se hace así debido que la clase TensorFlowClasifier usa librerías para consumir un modelo entrenado con TensorFlow y posteriormente guardado como modelo TensorFlow Lite, por lo que dichas librerías son compatibles con Android, pero no con Xamarin de manera directa.

Ilustración 23: Código de IClasificador en C#

```
IClasificador.cs
Detector de suelos
1 using System;
2 using System.Collections.Generic;
3 using System.Threading.Tasks;
4
5 namespace Detector_de_suelos
6 {
7     public interface IClasificador
8     {
9         event EventHandler<ClassificationEventArgs> ClassificationCompleted;
10
11         Task Classify(byte[] bytes);
12     }
13
14     public class ClassificationEventArgs : EventArgs
15     {
16         public List<Prediccion> Predictions { get; private set; }
17
18         public ClassificationEventArgs(List<Prediccion> predictions)
19         {
20             Predictions = predictions;
21         }
22     }
23 }
24
```

Fuente: Elaboración propia

La tercera y última clase ya citada anteriormente es la clase llamada TensorFlowClasifier en la cual se crea algunos métodos para consumir el modelo, el primer método es el que consume el modelo junto con su archivo de texto para sus etiquetas, los dos restantes son métodos auxiliares para transformar los pesos del modelo en formato MappedByteBuffer y la imagen en analizar en ByteBuffer.

Ilustración 24: Código de TensorFlowClasifier P1 en C#

```
TensorflowClasifier.cs
Detector de suelos.Android
1 using System.Collections.Generic;
2 using System.IO;
3 using System.Linq;
4 using Android.App;
5 using Android.Graphics;
6 using Java.Nio;
7 using System.Threading.Tasks;
8 using System;
9 using Java.IO;
10 using Java.Nio.Channels;
11
12 [assembly: Xamarin.Forms.Dependency(typeof(Detector_de_suelos.Droid.TensorFlowClasifier))]
13 namespace Detector_de_suelos.Droid
14 {
15     public class TensorFlowClasifier : IClasificador
16     {
17         //FloatSize is a constant with the value of 4 because a float value is 4 bytes
18         const int FloatSize = 4;
19         //PixelFormat is a constant with the value of 3 because a pixel has three color channels: Red Green and Blue
20         const int PixelSize = 3;
21
22         public event EventHandler<ClassificationEventArgs> ClassificationCompleted;
23     }
24 }
```

Fuente: Elaboración propia

Ilustración 25: Código de TensorFlowClasifier P2 en C#

```
2 referencias
public async Task Classify(byte[] bytes)
{
    var mappedByteBuffer = GetModelAsMappedByteBuffer();

    var interpreter = new Xamarin.TensorFlow.Lite.Interpreter(mappedByteBuffer);

    var tensor = interpreter.GetInputTensor(0);

    var shape = tensor.Shape();

    var width = shape[1];
    var height = shape[2];

    var byteBuffer = GetPhotoAsByteBuffer(bytes, width, height);

    var sr = new StreamReader(Application.Context.Assets.Open("labels.txt"));
    var labels = sr.ReadToEnd().Split('\n').Select(s => s.Trim()).Where(s => !string.IsNullOrEmpty(s)).ToList();

    var outputLocations = new float[1][] { new float[labels.Count] };

    var outputs = Java.Lang.Object.FromArray(outputLocations);

    interpreter.Run(byteBuffer, outputs);

    var classificationResult = outputs.ToArray<float[]>();

    var result = new List<Prediccion>();

    for (var i = 0; i < labels.Count; i++)
    {
        var label = labels[i];
        result.Add(new Prediccion(label, classificationResult[0][i]));
    }

    ClassificationCompleted?.Invoke(this, new ClassificationEventArgs(result));
}
```

Fuente: Elaboración propia

Ilustración 26: Código de TensorFlowClasifier P3 en C#

```
1 referencia
private MappedByteBuffer GetModelAsMappedByteBuffer()
{
    var assetDescriptor = Application.Context.Assets.OpenFd("model.tflite");
    var inputStream = new FileInputStream(assetDescriptor.FileDescriptor);

    var mappedByteBuffer = inputStream.Channel.Map(FileChannel.MapMode.ReadOnly, assetDescriptor.StartOffset, assetDescriptor.DeclaredLength);

    return mappedByteBuffer;
}

1 referencia
private ByteBuffer GetPhotoAsByteBuffer(byte[] bytes, int width, int height)
{
    var modelInputSize = FloatSize * height * width * PixelSize;

    var bitmap = BitmapFactory.DecodeByteArray(bytes, 0, bytes.Length);
    var resizedBitmap = Bitmap.CreateScaledBitmap(bitmap, width, height, true);

    var byteBuffer = ByteBuffer.AllocateDirect(modelInputSize);
    byteBuffer.Order(ByteOrder.NativeOrder());

    var pixels = new int[width * height];
    resizedBitmap.GetPixels(pixels, 0, resizedBitmap.Width, 0, resizedBitmap.Width, resizedBitmap.Height);

    var pixel = 0;

    for (var i = 0; i < width; i++)
    {
        for (var j = 0; j < height; j++)
        {
            var pixelVal = pixels[pixel++];

            byteBuffer.PutFloat(pixelVal >> 16 & 0xFF);
            byteBuffer.PutFloat(pixelVal >> 8 & 0xFF);
            byteBuffer.PutFloat(pixelVal & 0xFF);
        }
    }

    bitmap.Recycle();

    return byteBuffer;
}
```

Fuente: Elaboración propia

3. EVALUACIÓN DEL PROTOTIPO

En esta etapa de evaluación se siguen las directrices conforme lo estipulan las fases de la metodología XP, en esta fase se utilizan como herramientas unas matrices para evaluar el cumplimiento de las funciones del prototipo según la perspectiva del usuario y también la calidad del proyecto en general.

3.1. Plan de evaluación.

A continuación, se realizarán dos tipos de pruebas, en primera instancia se aplicará las ya conocidas pruebas unitarias que como ya se indicó anteriormente se las usa tomando en cuenta la perspectiva del usuario mediante la consulta de las historias de usuario elaboradas en el capítulo II, de igual manera se aplicará una prueba de control de calidad del software, mediante el uso del modelo de calidad ISO/IEC 25010 que indica algunos parámetros para evaluar el software.

3.1.1. Pruebas unitarias

Una prueba unitaria es una evaluación que tiene la función de aplicar un caso de prueba a un determinado módulo del software que no haya sufrido ningún cambio durante el transcurso del desarrollo del software.

Este tipo de evaluaciones son ejecutadas en cada función del software por los desarrolladores, con el fin de averiguar si dichas funciones cumplen con su objetivo.

A continuación, se muestra un tipo de prueba unitaria conocida como prueba de aceptación que será aplicado en cada objetivo a satisfacer de las historias de usuario del proyecto.

Tabla 15: Prueba de aceptación N°1

Pruebas de Aceptación	
Clasificación del suelo mediante el color	
Código de prueba	PA001
N.º Historia de usuario que prueba:	1
Nombre de historia de usuario que aprueba:	Opciones del menú inicial
Descripción:	<p>El usuario puede elegir entre dos opciones para analizar el suelo:</p> <p>La primera opción es para seleccionar una imagen previamente guardada en la galería del celular.</p> <p>La segunda opción es para activar la cámara del celular y permitir tomar fotografías del suelo en ese instante.</p>
Pasos de ejecución	<ol style="list-style-type: none"> 1. El usuario puede elegir dos opciones para analizar el suelo 2. El usuario se dirige a la opción deseada
Resultado obtenido: Evaluación:	<p>El usuario logró ir a la opción deseada.</p> <p style="text-align: center;">APROBADO</p>

Fuente: Elaboración propia

Tabla 16: Prueba de aceptación N°2

Pruebas de Aceptación	
Clasificación del suelo mediante el color	
Código de prueba	PA002
N.º Historia de usuario que prueba:	2
Nombre de historia de usuario que aprueba:	Seleccionar imagen de galería

Descripción:	El usuario puede seleccionar una imagen de su galería de fotos para posteriormente analizarla en la ventana que le aparecerá automáticamente luego de confirmar dicha imagen, caso contrario puede regresar al menú inicial.
Pasos de ejecución	<ol style="list-style-type: none"> 1. El usuario puede seleccionar cualquier imagen disponible en su galería 2. El usuario es dirigido a la ventana de análisis 3. El usuario puede regresar a la ventana anterior (opcional)
Resultado obtenido:	El usuario logró navegar por las ventanas con la opción de seleccionar imagen de galería.
Evaluación:	APROBADO

Fuente: Elaboración propia

Tabla 17: Prueba de aceptación N°3

Pruebas de Aceptación	
Clasificación del suelo mediante el color	
Código de prueba	PA003
N.º Historia de usuario que prueba:	3
Nombre de historia de usuario que aprueba:	Tomar fotografía con cámara del celular
Descripción:	El usuario puede tomar una fotografía con la cámara de su celular para posteriormente analizarla en la ventana que le aparecerá automáticamente luego de confirmar dicha foto, caso contrario puede regresar al menú inicial.
Pasos de ejecución	<ol style="list-style-type: none"> 1. El usuario puede tomar cualquier foto de una muestra de suelo estandarizado.

	<p>2. El usuario es dirigido a la ventana de análisis</p> <p>3. El usuario puede regresar a la ventana anterior (opcional)</p>
Resultado obtenido:	El usuario logró navegar por las ventanas con la opción de tomar fotografía con cámara del celular.
Evaluación:	APROBADO

Fuente: Elaboración propia

Tabla 18: Prueba de aceptación N°4

Pruebas de Aceptación	
Clasificación del suelo mediante el color	
Código de prueba	PA004
N.º Historia de usuario que prueba:	4
Nombre de historia de usuario que aprueba:	Opciones de la ventana analizar
Descripción:	El usuario puede observar la foto elegida según la opción seleccionada anteriormente, además podrá regresar al menú inicial por si desea cambiar de opción, de igual manera si está seguro de la foto puede analizarla y con ello se mostrará un resultado que indica el tipo de suelo y su porcentaje de certeza.
Pasos de ejecución	<p>1. El usuario puede analizar la foto</p> <p>2. El usuario puede regresar atrás para elegir otra opción (opcional)</p> <p>3. El usuario puede ver el resultado del análisis</p>
Resultado obtenido: Evaluación:	El usuario logró navegar por las ventanas después de haber elegido la opción analizar. APROBADO

Fuente: Elaboración propia

Tabla 19: Prueba de aceptación N°5

Pruebas de Aceptación	
Clasificación del suelo mediante el color	
Código de prueba	PA005
N.º Historia de usuario que prueba:	5
Nombre de historia de usuario que aprueba:	Analizar imagen o foto
Descripción:	El usuario puede analizar la imagen y al finalizar el análisis se mostrará en la parte inferior el resultado que indicará el tipo de suelo y el porcentaje de certeza del mismo.
Pasos de ejecución	<ol style="list-style-type: none"> 1. El usuario puede dar clic en el botón analizar. 2. El usuario puede ver el resultado del análisis.
Resultado obtenido:	El usuario logró obtener los resultados del análisis de la muestra de suelo.
Evaluación:	APROBADO

Fuente: Elaboración propia

Tabla 20: Prueba de aceptación N°6

Pruebas de Aceptación	
Clasificación del suelo mediante el color	
Código de prueba	PA006
N.º Historia de usuario que prueba:	6
Nombre de historia de usuario que aprueba:	Navegación entre ventanas
Descripción:	El usuario puede viajar entre las ventanas del menú inicial y analizar, siempre y cuando se

	quiera analizar una imagen o cancelar la decisión.
Pasos de ejecución	<ol style="list-style-type: none"> 1. El usuario puede cancelar la selección de una imagen 2. El usuario puede cancelar la toma de una fotografía. 3. El usuario puede regresar a la ventana inicial desde la ventana de análisis
Resultado obtenido:	El usuario logró navegar por las ventanas sin ningún inconveniente o error.
Evaluación:	APROBADO

Fuente: Elaboración propia

3.1.2. Evaluación de calidad del software

La evaluación del control de calidad del software aplicada en este proyecto se consideró según la norma ISO/IEC 25010, debido que esta norma es un estándar internacional que define un modelo de evaluación aplicable a todo tipo de software.

Para calificar y conocer los resultados de las subcategorías o parámetros estipulados por la norma ISO/IEC 25010, se requirió utilizar un indicador de evaluación del tipo escala de Likert.

Tabla 21: Indicador de evaluación de calidad

Indicadores de evaluación	
Valoración	Interpretación de resultados
5	Excelente
4	Muy Bueno
3	Bueno
2	Malo
1	Muy malo

Fuente: Elaboración propia

Tabla 22: Evaluación de calidad según ISO/IEC 25010

Característica	Sub-característica	Objetivo	1	2	3	4	5	Total
Funcionalidad	Compleitud	El sistema cubre todas las funciones para las que fue diseñado					X	5
	Corrección	Las funciones del programa proporcionan el resultado correcto					X	5
	Pertinencia	Las funciones del programa van de acuerdo a sus objetivos.					X	5
Eficiencia	Comportamiento	La aplicación otorga buen tiempo de respuesta al ejecutar funciones					X	5
	Utilización de recursos	La aplicación utiliza los recursos adecuados en el cumplimiento determinado de sus funciones					X	5

	Capacidad	La aplicación cumple con los límites máximos para cada requisito propuesto.					X	5
Compatibilidad	Coexistencia	La aplicación puede desplegarse en un sistema distinto al destinado originalmente				X		4
	Interoperabilidad	La aplicación opera con módulos y API's externos con facilidad				X		4
Usabilidad	Inteligibilidad	La aplicación es de Fácil entendimiento para su usuario					X	5
	Aprendizaje	El usuario puede aprender con facilidad el funcionamiento de la aplicación					X	5
	Operabilidad	La aplicación puede ser manejada por el					X	5

		usuario con facilidad							
	Protección frente a errores de usuario	La aplicación captura y evita saltos de errores					X	5	
	Estética	La interfaz de usuario es agradable a la vista					X	5	
	Accesibilidad	La aplicación es altamente accesible para los usuarios en general					X	5	
Fiabilidad	Madurez	La aplicación satisface las necesidades del usuario en condiciones normales					X	5	
	Disponibilidad	La aplicación permanece operativa en cualquier momento					X	5	
	Tolerancia a fallos	La aplicación opera con normalidad ante una falla de					X	5	

		hardware o software						
	Capacidad de recuperación	La aplicación recupera y restaura datos en caso de ocurrir un error o fallo				X		4
Seguridad	Confidencialidad	La aplicación protege contra el acceso de datos no autorizados						-
	Integridad	La aplicación previene el acceso o modificaciones no autorizados						-
	No repudio	La aplicación guarda acciones que puedan ser deshechas en un futuro						-
	Autenticidad	La aplicación permite el acceso a usuarios autenticados						-
	Responsabilidad	La aplicación rastrea y						-

		guarda las acciones realizadas						
Mantenibilidad	Modularidad	La aplicación trabaja con sus módulos independientes uno de otros					X	5
	Reusabilidad	Los componentes de la aplicación pueden ser usados en más de una vez					X	5
	Analizabilidad	Es apreciable el cambio realizado en un módulo específico					X	5
	Capacidad de ser modificado	La aplicación puede ser modificada de manera efectiva y eficiente.					X	5
	Capacidad de ser aprobado	La aplicación puede ser sometida a pruebas					X	5
Portabilidad	Adaptabilidad	La aplicación se adapta eficientemente					X	5

		a diferentes dispositivos						
	Facilidad de instalación	La aplicación se instala sin presentar problemas					X	5
	Capacidad de ser reemplazado	La aplicación puede ser reemplazada y/o actualizada					X	5

Fuente: Elaboración propia

3.2. Resultados de la evaluación.

En la siguiente tabla se muestran los resultados obtenidos a partir de la aplicación de la prueba de calidad del software en el proyecto según la norma ISO/IEC 25010:

Tabla 23: Resumen de resultados de evaluación

Características	Parámetros	Puntaje	Promedio	Total
Funcionalidad	3	15	5	Excelente
Eficiencia	3	15	5	Excelente
Compatibilidad	2	8	4	Muy Bueno
Usabilidad	6	30	5	Excelente
Fiabilidad	4	19	4,75	Excelente
Seguridad	5	-	-	-
Mantenibilidad	5	25	5	Excelente
Portabilidad	3	15	5	Excelente
			33,75	Muy Bueno

Fuente: Elaboración propia

En la tabla anterior se puede interpretar que el prototipo propuesto tiene como grandes fortalezas las características de: Funcionalidad, eficiencia y usabilidad, de igual manera tiene buen puntaje en la Mantenibilidad debido que se construyó la aplicación en tres capas del tipo Modelo, Vista y Controlador, a su vez el alto puntaje en Portabilidad se debe a que el proyecto está construido en un entorno multiplataforma muy estable y confiable.

El promedio total del aplicativo indicó un resultado del 4,22/5 lo cual significa que se ha cumplido con la gran parte de las directrices que se exigen en la norma ISO/IEC 25010, la cual es una certificación oficial de la calidad de los productos de software.

4. Conclusiones.

Una vez concluida las etapas de desarrollo de la aplicación móvil para el análisis de suelos con Deep Learning se puede indicar que:

- La preferencia de usar Xamarin como framework de desarrollo sirvió para codificar la aplicación de manera que se comparte las interfaces gráficas y gran parte del código en los dispositivos Android, teniendo la posibilidad en un futuro de desplegar el proyecto a dispositivos iOS y Windows Phone.
- Gracias al uso de la metodología XP se logró un desarrollo ágil en el proyecto, debido que en ella se prioriza los tiempos y el manejo del personal en todos los ciclos de desarrollo del prototipo.
- El haber identificado los requerimientos funcionales y no funcionales permitió la optimización del tiempo en distribuir los esfuerzos en programar inicialmente lo más prioritario y dejar para el final las funcionalidades menos esenciales.
- Se diseñaron interfaces gráficas de usuario muy intuitivas fáciles aprender y usar con el fin de que el usuario no tenga inconvenientes con el manejo del aplicativo, además se estableció su comportamiento con la ayuda de las herramientas CASE, tales como el diagrama de actividades y el de casos de uso.
- Se realizó la evaluación de manera satisfactoria del correcto cumplimiento de los objetivos de cada función considerada en la aplicación, mediante las pruebas unitarias sobre las historias de usuario establecidas, además se utilizó las directrices de la norma ISO/IEC 25010 para evaluar y así certificar la calidad del software desarrollado.

5. Recomendaciones.

Tomando en cuenta los imprevistos, las dificultades y la cantidad de tiempo invertido en la investigación de ciertos algoritmos para cumplir con los objetivos del proyecto, se indican las siguientes recomendaciones:

- Establecer el proyecto en tres capas como son el conocido Modelo, Vista Controlador lo cual le permite al desarrollador dar un fácil mantenimiento al software, así como reutilizar código en otras funciones requeridas.
- Considerar aprender y utilizar framework multiplataforma para el desarrollo de aplicaciones móviles con el objetivo de evitar contratiempos en aspectos como el tiempo de desarrollo, los problemas de compatibilidad y las limitaciones en la reutilización de código entre los distintos sistemas operativos de los dispositivos móviles.
- Manejar herramientas de control de versiones para facilitar la comparación y recuperación de código modificado con el fin de reducir tiempo en la etapa de desarrollo del software.
- Aplicar una norma de calidad de software para evaluar el proyecto desarrollado, así como el usar pruebas unitarias para identificar si se cumplen los objetivos del funcionamiento de cada proceso del aplicativo.

Bibliografía

- [1] «Visual Studio: IDE y Editor de código para desarrolladores de software y Teams». <https://visualstudio.microsoft.com/es/> (accedido nov. 18, 2021).
- [2] «Xamarin | Open-source mobile app platform for .NET». <https://dotnet.microsoft.com/apps/xamarin> (accedido nov. 18, 2021).
- [3] «TensorFlow». <https://www.tensorflow.org/> (accedido nov. 18, 2021).
- [4] «NuGet Gallery | Acr.UserDialogs 7.2.0.534». <https://www.nuget.org/packages/Acr.UserDialogs> (accedido nov. 18, 2021).
- [5] «NuGet Gallery | Xam.Plugin.Media 5.0.1». <https://www.nuget.org/packages/Xam.Plugin.Media/> (accedido nov. 18, 2021).
- [6] J. M. Bautista, «UNIVERSIDAD UNION BOLIVARIANA», p. 7.
- [7] O. A. Dada y I. T. Sanusi, «The adoption of Software Engineering practices in a Scrum environment», *Afr. J. Sci. Technol. Innov. Dev.*, vol. 0, n.º 0, pp. 1-18, ago. 2021, doi: 10.1080/20421338.2021.1955431.
- [8] M. Mayer *et al.*, «Soil fertility relates to fungal-mediated decomposition and organic matter turnover in a temperate mountain forest», *New Phytol.*, vol. 231, n.º 2, pp. 777-790, jul. 2021, doi: 10.1111/nph.17421.
- [9] G. Aredehey, G. B. Zenebe, y A. Gebremedhn, «Land use impacts on physicochemical and microbial soil properties across the agricultural landscapes of Debrekidan, EasternTigray, Ethiopia», *Cogent Food Agric.*, vol. 5, n.º 1, p. 1708683, ene. 2019, doi: 10.1080/23311932.2019.1708683.
- [10] C. R. Paramasivam y S. Anbazhagan, «Soil fertility analysis in and around magnesite mines, Salem, India», *Geol. Ecol. Landsc.*, vol. 4, n.º 2, pp. 140-150, abr. 2020, doi: 10.1080/24749508.2019.1608407.
- [11] G. T. Patle, T. T. Sikar, K. S. Rawat, y S. K. Singh, «Estimation of infiltration rate from soil properties using regression model for cultivated land», *Geol. Ecol. Landsc.*, vol. 3, n.º 1, pp. 1-13, ene. 2019, doi: 10.1080/24749508.2018.1481633.
- [12] H. Oumenskou *et al.*, «Multivariate statistical analysis for spatial evaluation of physicochemical properties of agricultural soils from Beni-Amir irrigated perimeter, Tadla plain, Morocco», *Geol. Ecol. Landsc.*, vol. 3, n.º 2, pp. 83-94, abr. 2019, doi: 10.1080/24749508.2018.1504272.
- [13] O. G. Adhanom, «Salinity and sodicity hazard characterization in major irrigated areas and irrigation water sources, Northern Ethiopia», *Cogent Food Agric.*, vol. 5, n.º 1, p. 1673110, ene. 2019, doi: 10.1080/23311932.2019.1673110.
- [14] F. G. A. Verheijen *et al.*, «Representativeness of European biochar research: part I – field experiments», *J. Environ. Eng. Landsc. Manag.*, vol. 25, n.º 2, pp. 140-151, abr. 2017, doi: 10.3846/16486897.2017.1304943.

- [15] B. Shewangizaw, K. Kassie, S. Assefa, y T. Feyisa, «On farm verification of soil test-based phosphorus fertilizer recommendations for bread wheat (*Triticum aestivum* L.) on the vertisols of central highlands of Ethiopia», *Cogent Food Agric.*, vol. 6, n.º 1, p. 1807811, ene. 2020, doi: 10.1080/23311932.2020.1807811.
- [16] N. Abdul Rahman, A. Larbi, A. Opoku, F. M. Tetteh, y I. Hoeschle-Zeledon, «Corralling, planting density, and N fertilizer rate effect on soil properties, weed diversity, and maize yield», *Agroecol. Sustain. Food Syst.*, vol. 43, n.º 3, pp. 243-260, mar. 2019, doi: 10.1080/21683565.2018.1516264.
- [17] P. A. Swanepoel y F. Tshuma, «Soil quality effects on regeneration of annual *Medicago* pastures in the Swartland of South Africa», *Afr. J. Range Forage Sci.*, vol. 34, n.º 4, pp. 201-208, oct. 2017, doi: 10.2989/10220119.2017.1403462.
- [18] Z. Bazie, S. Feyssa, y T. Amare, «Effects of *Acacia decurrens* Willd. tree-based farming system on soil quality in Guder watershed, North Western highlands of Ethiopia», *Cogent Food Agric.*, vol. 6, n.º 1, p. 1743622, ene. 2020, doi: 10.1080/23311932.2020.1743622.
- [19] B. G. Honfoga, «Diagnosing soil degradation and fertilizer use relationship for sustainable cotton production in Benin», *Cogent Environ. Sci.*, vol. 4, n.º 1, p. 1422366, ene. 2018, doi: 10.1080/23311843.2017.1422366.
- [20] M. B. Arsene y J. Nkulu Mwine Fyama, «Potential threats to agricultural food production and farmers' coping strategies in the marshlands of Kabare in the Democratic Republic of Congo», *Cogent Food Agric.*, vol. 7, n.º 1, p. 1933747, ene. 2021, doi: 10.1080/23311932.2021.1933747.
- [21] T. Bayu, «Review on contribution of integrated soil fertility management for climate change mitigation and agricultural sustainability», *Cogent Environ. Sci.*, vol. 6, n.º 1, p. 1823631, ene. 2020, doi: 10.1080/23311843.2020.1823631.
- [22] K. Ikazaki, F. Nagumo, S. Simporé, y A. Barro, «Are all three components of conservation agriculture necessary for soil conservation in the Sudan Savanna?», *Soil Sci. Plant Nutr.*, vol. 64, n.º 2, pp. 230-237, mar. 2018, doi: 10.1080/00380768.2017.1422393.
- [23] F. S. Higashikawa *et al.*, «Effects of the application of biochar on soil fertility status, and nutrition and yield of onion grown in a no-tillage system», *Arch. Agron. Soil Sci.*, vol. 0, n.º 0, pp. 1-16, sep. 2021, doi: 10.1080/03650340.2021.1978073.
- [24] S. Vingiani *et al.*, «Soils of the Aversa plain (southern Italy)», *J. Maps*, vol. 14, n.º 2, pp. 312-320, nov. 2018, doi: 10.1080/17445647.2018.1458338.
- [25] P. C. Onyia, D. C. Ozoko, y S. I. Ifediegwu, «Phytoremediation of arsenic-contaminated soils by arsenic hyperaccumulating plants in selected areas of Enugu State, Southeastern, Nigeria», *Geol. Ecol. Landsc.*, vol. 0, n.º 0, pp. 1-12, ago. 2020, doi: 10.1080/24749508.2020.1809058.

- [26] F. Saigí-Rubió, D. Novillo-Ortiz, y J. D. Piette, «Red CYTED-RITMOS: hacia la búsqueda de soluciones para fomentar la salud móvil en América Latina», *Rev. Panam. Salud Pública*, vol. 41, p. e33, mar. 2017, doi: 10.26633/RPSP.2017.33.
- [27] A. Biørn-Hansen, T.-M. Grønli, y G. Ghinea, «Animations in Cross-Platform Mobile Applications: An Evaluation of Tools, Metrics and Performance», *Sensors*, vol. 19, n.º 9, p. 2081, may 2019, doi: 10.3390/s19092081.
- [28] C. Powell, J. Motley, H. Qin, y J. Shaw, «A born-digital field-to-database solution for collections-based research using collNotes and collBook», *Appl. Plant Sci.*, vol. 7, n.º 8, p. e11284, ago. 2019, doi: 10.1002/aps3.11284.
- [29] D. Astiaso Garcia, D. Groppi, y S. Tavakoli, «Developing and testing a new tool to foster wind energy sector industrial skills», *J. Clean. Prod.*, vol. 282, p. 124549, feb. 2021, doi: 10.1016/j.jclepro.2020.124549.
- [30] R. Kraft *et al.*, «Combining Mobile Crowdsensing and Ecological Momentary Assessments in the Healthcare Domain», *Front. Neurosci.*, vol. 14, p. 164, feb. 2020, doi: 10.3389/fnins.2020.00164.
- [31] P. A. Savenkov y A. N. Ivutin, «Methods of Machine Learning in System Abnormal Behavior Detection», *Adv. Swarm Intell.*, vol. 12145, pp. 495-505, jun. 2020, doi: 10.1007/978-3-030-53956-6_45.
- [32] B. Hunt, A. J. Ruiz, y B. W. Pogue, «Smartphone-based imaging systems for medical applications: a critical review», *J. Biomed. Opt.*, vol. 26, n.º 4, p. 040902, abr. 2021, doi: 10.1117/1.JBO.26.4.040902.
- [33] J. A. S. Aranda, R. S. Bavaresco, J. V. de Carvalho, A. C. Yamin, M. C. Tavares, y J. L. V. Barbosa, «A computational model for adaptive recording of vital signs through context histories», *J. Ambient Intell. Humaniz. Comput.*, pp. 1-15, mar. 2021, doi: 10.1007/s12652-021-03126-8.
- [34] C. M. Blanchard *et al.*, «Evaluation of PIQNIQ, a Novel Mobile Application for Capturing Dietary Intake», *J. Nutr.*, vol. 151, n.º 5, pp. 1347-1356, mar. 2021, doi: 10.1093/jn/nxab012.
- [35] Z. Z. R. M. Weerts *et al.*, «Smart Data Collection for the Assessment of Treatment Effects in Irritable Bowel Syndrome: Observational Study», *JMIR MHealth UHealth*, vol. 8, n.º 11, p. e19696, nov. 2020, doi: 10.2196/19696.
- [36] N. Petrellis, «A COVID-19 Multipurpose Platform», *Digit. Biomark.*, vol. 4, n.º 3, pp. 89-98, oct. 2020, doi: 10.1159/000511704.
- [37] P. Watts, P. Breedon, C. Nduka, C. Neville, V. Venables, y S. Clarke, «Cloud Computing Mobile Application for Remote Monitoring of Bell's Palsy», *J. Med. Syst.*, vol. 44, n.º 9, p. 149, 2020, doi: 10.1007/s10916-020-01605-7.
- [38] M. Rivera-Díaz, N. Varas-Díaz, M. Reyes-Estrada, D. Coriano, N. Ortiz, y M. Padilla, «TECNOLOGÍA MÓVIL COMO HERRAMIENTA POTENCIAL

EN LA INVESTIGACIÓN SOBRE ESTIGMA ASOCIADO AL VIH/SIDA ENTRE ESTUDIANTES DE MEDICINA», *Rev. Puertorriquena Psicol.*, vol. 30, n.º 1, pp. 60-68, 2019.

- [39] F. Guo, W. Li, X. Zhao, J. Qiu, y Y. Mai, «A mobile app for Glaucoma diagnosis and its possible clinical applications», *BMC Med. Inform. Decis. Mak.*, vol. 20, n.º Suppl 3, p. 128, jul. 2020, doi: 10.1186/s12911-020-1123-2.
- [40] J.-M. Kim, W. R. Lee, J.-H. Kim, J.-M. Seo, y C. Im, «Light-Induced Fluorescence-Based Device and Hybrid Mobile App for Oral Hygiene Management at Home: Development and Usability Study», *JMIR MHealth UHealth*, vol. 8, n.º 10, p. e17881, oct. 2020, doi: 10.2196/17881.
- [41] D. Montenegro-López, «Uso de tecnologías en el lugar de atención para el manejo de la pandemia por COVID-19 en Colombia», *Rev. Panam. Salud Pública*, vol. 44, p. e97, ago. 2020, doi: 10.26633/RPSP.2020.97.
- [42] J. J. Cubillas Mercado, M. Quero Haro, M. B. Ramos Rodriguez, M. I. Ramos, W. Lopez, y J. M. Gonzalez Conejo, «Uso de la aplicación móvil de Salud Responde para la optimización de los recursos sanitarios», *Aten. Primaria*, vol. 51, n.º 10, pp. 660-661, dic. 2019, doi: 10.1016/j.aprim.2019.03.003.

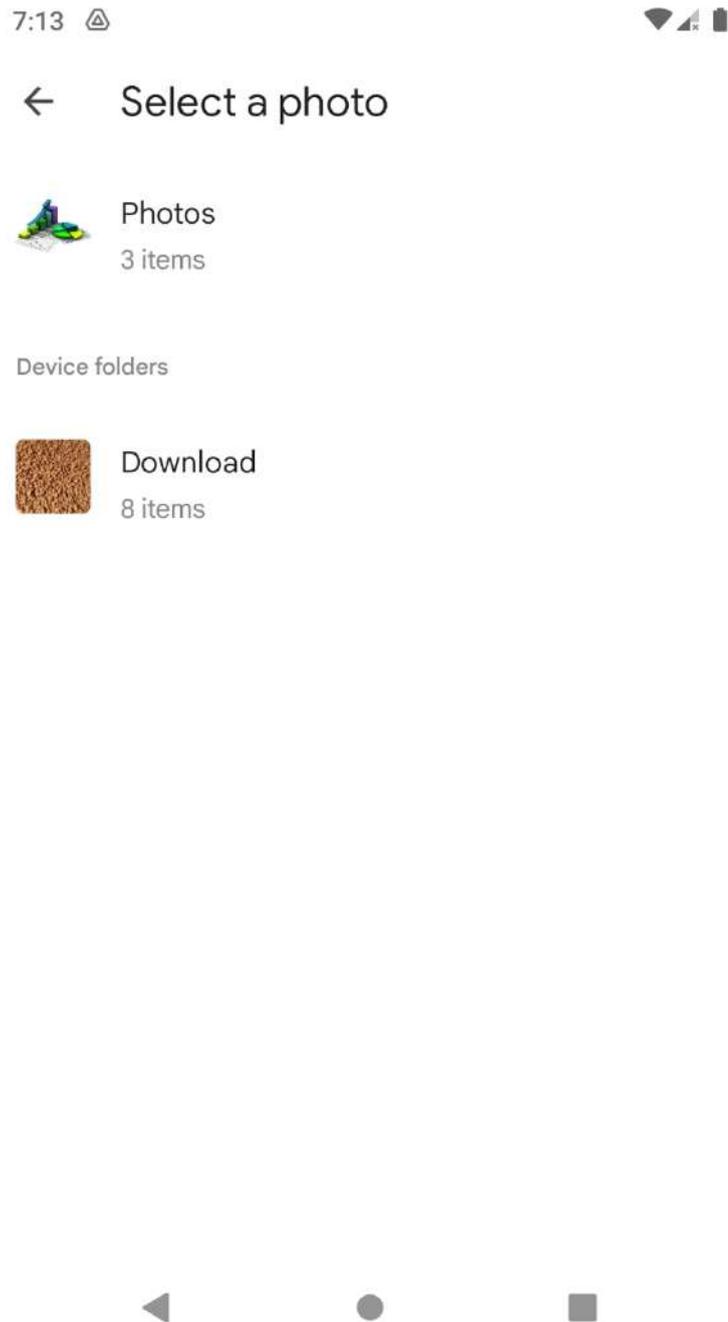
Anexos

Ilustración 27: Ventana del menú principal



Fuente: Elaboración propia

Ilustración 28: Selector del álbum de la galería



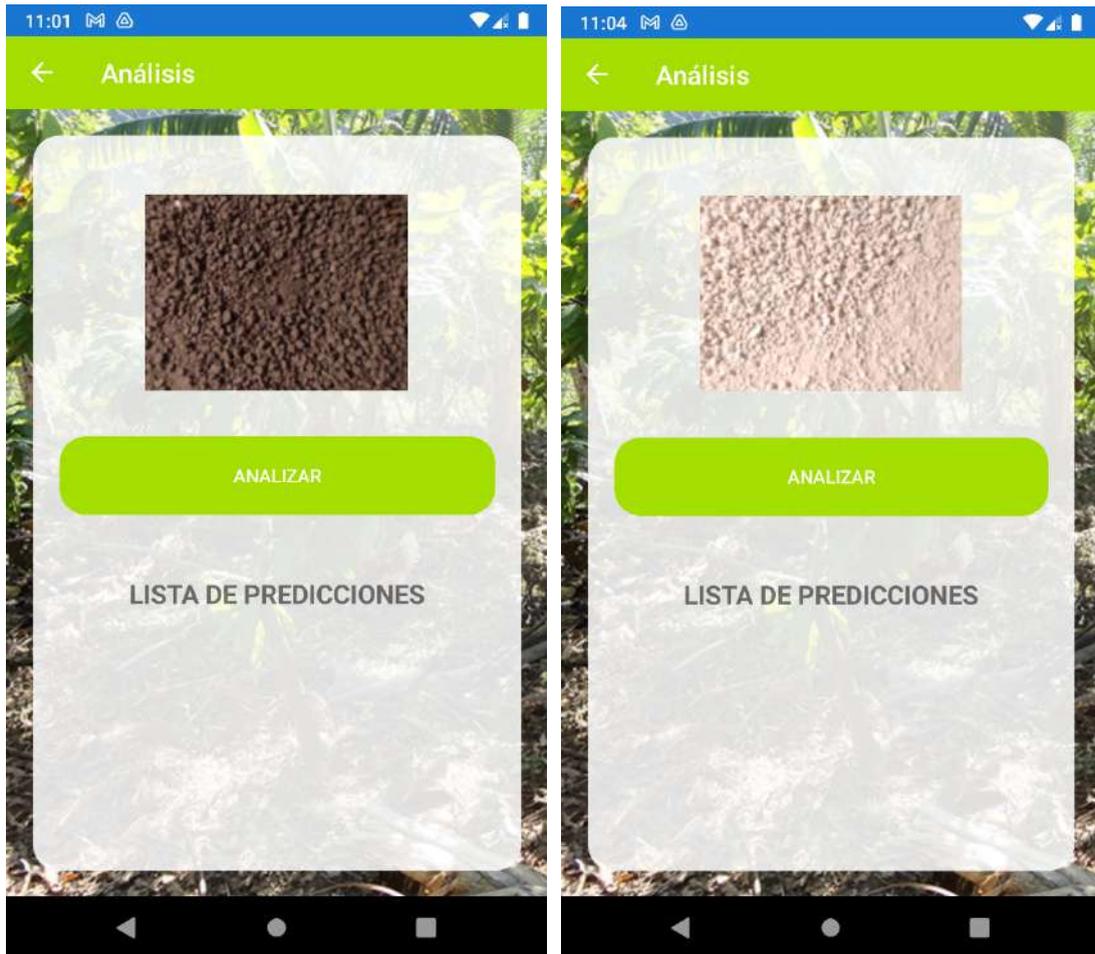
Fuente: Elaboración propia

Ilustración 29: Elegir una imagen del álbum seleccionado



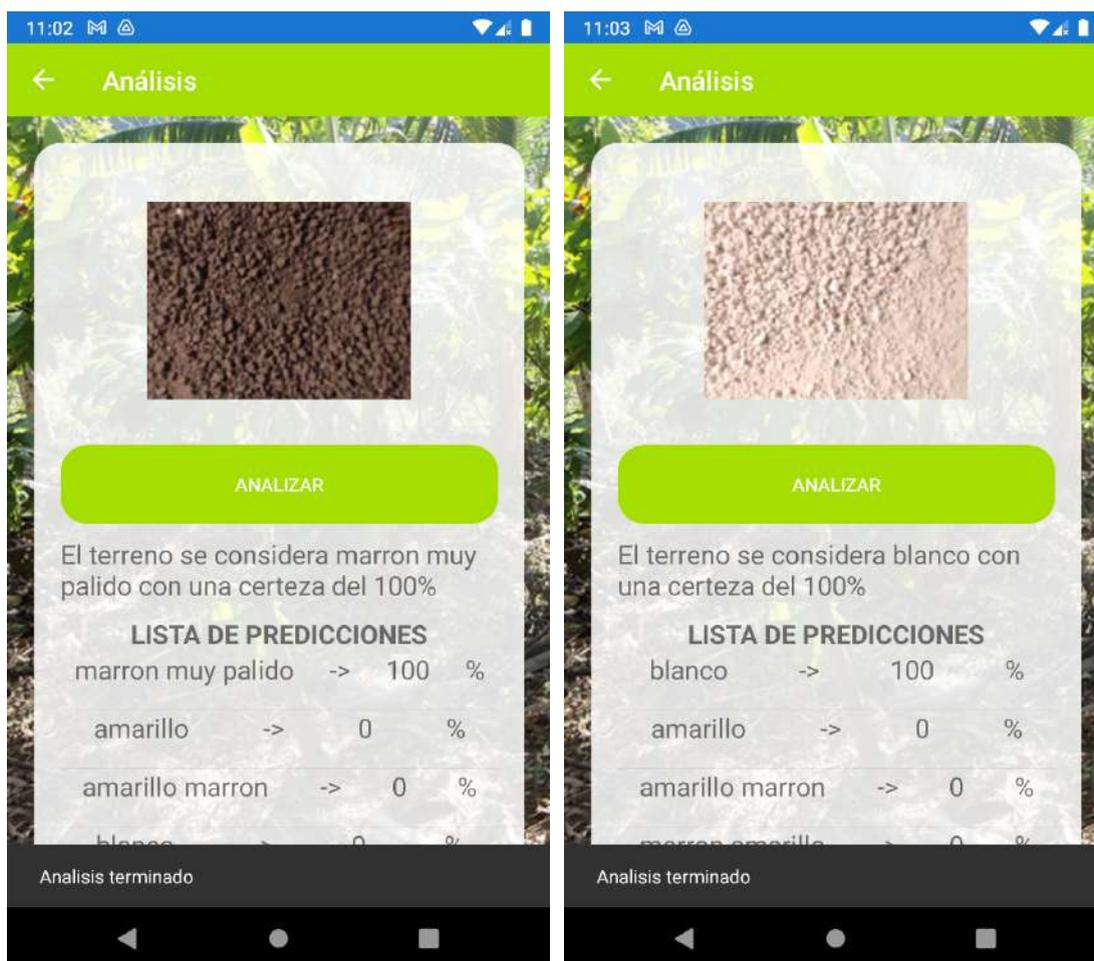
Fuente: Elaboración propia

Ilustración 30: Ventana del análisis de la imagen



Fuente: Elaboración propia

Ilustración 31: Análisis de la imagen elegida



Fuente: Elaboración propia

Ilustración 32: Tomar una foto con la cámara



Fuente: Elaboración propia

Ilustración 33: Confirmación de la foto tomada



Fuente: Elaboración propia

Ilustración 34: Análisis de la foto tomada



Fuente: Elaboración propia