

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas
Carrera de Ingeniería en Sistemas Computacionales

**DESARROLLO DE UNA APLICACIÓN MÓVIL ANDROID, PARA LA DETECCIÓN DE
CONTENIDO RIESGOSO EN TELÉFONOS INTELIGENTES USADOS POR NIÑOS Y
ADOLESCENTES, BASADO EN TÉCNICAS DE VISIÓN ARTIFICIAL.**

Trabajo de grado previo a la obtención del título de Ingeniero en Sistemas
Computacionales

Autor:

Roger Alexander Vaca Ponce

Director:

MSc. Daisy Imbaquingo

Ibarra – Ecuador

Enero 2021



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	100304612-3		
APELLIDOS Y NOMBRES:	VACA PONCE ROGER ALEXANDER		
DIRECCIÓN:	URCUQUÍ - BARRIO LA PLAZA VIEJA, CALLE CARLOS DE LA TORRE.		
EMAIL:	ravacap@utn.edu.ec		
TELÉFONO FIJO:	062-939-918	TELÉFONO MÓVIL:	0959785344

DATOS DE LA OBRA	
TÍTULO:	DESARROLLO DE UNA APLICACIÓN MÓVIL ANDROID, PARA LA DETECCIÓN DE CONTENIDO RIESGOSO EN TELÉFONOS INTELIGENTES USADOS POR NIÑOS Y ADOLESCENTES, BASADO EN TÉCNICAS DE VISIÓN ARTIFICIAL.
AUTOR (ES):	ROGER ALEXANDER VACA PONCE
FECHA: DD/MM/AAAA	21/01/2021
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	INGENIERO EN SISTEMAS COMPUTACIONALES
ASESOR /DIRECTOR:	MSc. DAISY IMBAQUINGO

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 21 días del mes de enero de 2021

EL AUTOR:



.....
Nombre: ROGER ALEXANDER VACA PONCE
100304612-3



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS INGENIERÍA EN SISTEMAS COMPUTACIONALES

CERTIFICACIÓN DEL DIRECTOR

Por medio del presente yo **MSc. DAISY IMBAQUINGO**, certifico que el estudiante: **ROGER ALEXANDER VACA PONCE**, portador de la cédula de identidad Nro. **100304612-3**, ha realizado en su totalidad el desarrollo del proyecto de grado “**DESARROLLO DE UNA APLICACIÓN MÓVIL ANDROID, PARA LA DETECCIÓN DE CONTENIDO RIESGOSO EN TELÉFONOS INTELIGENTES USADOS POR NIÑOS Y ADOLESCENTES, BASADO EN TÉCNICAS DE VISIÓN ARTIFICIAL.**”, previo a la obtención del título de Ingeniería en Sistemas Computacionales y realizando con el interés profesional y responsabilidad, que verifico en honor a la verdad.

Atentamente,

DAISY ELIZABETH
IMBAQUINGO ESPARZA

Firmado digitalmente por DAISY
ELIZABETH IMBAQUINGO ESPARZA
Fecha: 2021.01.20 16:10:09 -05'00'

.....
MSc. Daisy Imbaquingo
DIRECTORA DEL TRABAJO DE GRADO

DEDICATORIA

Este trabajo de grado está dedicado a Dios, a mis queridos padres Luis Vaca y Rosa Ponce, a mis hermanas, Tatiana, Amanda y Ariana, a mis sobrinos Ludoviquito y Samara, a mis abuelitos Luis, María, Luz y Manuel por ser mi apoyo incondicional en todo instante de mi carrera profesional y mi vida en general, sin la ayuda de ellos este logro no sería posible.

Roger Vaca

AGRADECIMIENTOS

A Dios por guiarme en cada momento de mi vida por el camino correcto y enviarme a las personas perfectas para cada instante de mi vida.

A mis padres Luis Vaca y Rosa Ponce, por incentivar me cada día a continuar mi carrera profesional sin importar los obstáculos presentes, por no dejarme solo en ningún momento y ser muy buenos padres conmigo y con mis hermanas, gracias a sus valores a sus palabras a sus gestos de amor y apoyo incondicional he llegado a esta cima tan alta.

A mi novia Anabel Carrera por estar junto a mí, en esta etapa muy importante de mi vida y ser mi compañera del día a día.

A mis hermanas Tatiana, Amanda y Ariana por ser unas personas maravillosas, siempre me han apoyado incondicionalmente, y admiro mucho su humildad.

A mis abuelitos, a mis tíos Jorge, Pablo y Jacqueline que durante mi carrera universitaria ellos me apoyaron cuando más necesitaba, infinitamente gracias a ellos.

A mi querida primita Olguita quien se ha preocupado mucho por mi bienestar y mi formación como profesional, sinceramente que Dios le pague y le colme de bendiciones.

A mi primer tutor de tesis, MSc. Erick Herrera estoy inmensamente agradecido de corazón por ser un profesional que ha estado siempre para compartir sus conocimientos y apoyarme cuando más lo necesité, su ayuda ha sido fundamental para culminar este proyecto exitosamente.

A mi tutora MSc. Daisy Imbaquingo que en todo momento ha sido una ayuda para complementar mi trabajo de grado.

A mis asesores de tesis PhD. Iván García y MSc. Diego Trejo por apoyarme con su valioso conocimiento para mejorar la calidad de mi trabajo de grado.

A mis amigos que en el transcurso de la carrera fueron muchos quienes compartieron momentos conmigo y fueron un apoyo incondicional, especialmente a Mario Palacios y a René Ipiales infinitas gracias por su amistad.

TABLA DE CONTENIDOS

Contenido

INTRODUCCIÓN	XV
1. Antecedentes.....	XV
2. Situación actual	XV
3. Planteamiento del problema	XV
4. Objetivos.....	XVI
5. Objetivo general.....	XVI
6. Objetivos específicos	XVI
7. Alcance.....	XVI
8. Justificación	XIX
9. Contexto	XX
CAPÍTULO 1.....	1
1.1. Menores de edad y la tecnología	1
1.1.1. Ventajas.....	2
1.1.2. Desventajas	2
1.1.3. Usos de Internet	2
1.1.4. Riesgos que asechan a menores en internet.	4
1.2. ¿Control Parental o violación de derechos?	5
1.2.1. Control Parental.....	5
1.2.2. Derechos de los menores de edad	5
1.3. Inteligencia Artificial.....	7
1.3.1. Visión artificial.....	8
1.3.2. Fases de la Visión Artificial	8
1.3.3. Métodos de reconocimiento de objetos.....	9
1.3.4. Métodos de reconocimiento de texto	16
1.4. Teléfono inteligente	17
1.4.1. Características de un teléfono inteligente	18
1.4.2. Sistemas Operativos más usados.....	18
1.4.3. Sistema Operativo Android	19
1.5. Librerías de inteligencia artificial compatibles con Android.	22
1.5.1. OpenCV4Android.....	22
1.5.2. Tesseract OCR	23
1.6. Aplicaciones Móviles de Control Parental (Android)	23

1.6.1.	Universidad de Guayaquil del Ecuador	24
1.6.2.	Spyzie.....	25
1.7.	Proyectos de Visión Artificial.....	27
1.7.1.	Detección de imágenes con contenido explícito usando los modelos de color hsv y ybcr 27	
1.7.2.	Diseño de un sistema de reconocimiento automático de matrículas de vehículos mediante una red neuronal convolucional.....	27
1.7.3.	Desarrollo de un clasificador visual de especies de aves mediante redes neuronales convolucionales.....	27
1.8.	Metodologías de Desarrollo de Software	28
1.8.1.	Marco de trabajo Scrum.....	28
1.9.	Lenguaje de Programación R.....	31
CAPÍTULO 2.....	33
2.1.	Análisis	33
2.1.1.	Equipo Scrum	33
2.1.2.	Requerimientos del Proyecto	34
2.1.3.	Product backlog	38
2.2.	Diseño	39
2.2.1.	Sprint 0	39
2.3.	Desarrollo	42
2.3.1.	Sprint 1	42
2.3.2.	Sprint 2	45
2.3.3.	Sprint 3	57
2.3.4.	Sprint 4	62
2.3.5.	Sprint 5	68
2.3.6.	Sprint 6	74
CAPÍTULO 3.....	80
3.1.	Muestreo	80
3.2.	Matrices de confusión.....	82
3.3.	Métricas de desempeño	82
3.3.1.	Precisión.....	82
3.3.2.	Exactitud.....	83
3.3.3.	Especificidad.....	83
3.3.4.	Exhaustividad.....	83
3.3.5.	Valor F1.....	83

3.4. Resultados	83
3.4.1. Resultados por clase.....	83
3.4.2. Resultados por versión de Android	84
3.4.3. Resumen de Resultados.....	87
3.5. Curva ROC.....	87
CONCLUSIONES.....	89
RECOMENDACIONES.....	90
REFERENCIAS	91
ANEXOS.....	95

ÍNDICE DE FIGURAS

Fig 1. Árbol de problemas	XVI
Fig 2. Arquitectura del proyecto	XVIII
Fig 3. Funcionamiento de la aplicación	XIX
Fig 4. Acceso a los aparatos tecnológicos en América Latina	2
Fig 5. El sistema de detección de YOLO	12
Fig 6. Red Neuronal Convolutiva de YOLO	13
Fig 7. Red Neuronal Convolutiva Tiny-YOLO	15
Fig 8. Localización de objetos en imágenes con los distintos métodos	15
Fig 9. Estadística SO de smartphones más usados	19
Fig 10. Sistemas de capas de Android	20
Fig 11. OpenCV sobre Android	23
Fig 12. Demo de la aplicación Spyzie	26
Fig 13. Demo de la aplicación Spyzie	26
Fig 14. Marco de Trabajo SCRUM	29
Fig 15. Lenguaje de programación R para estadísticas	32
Fig 16. Ciclo desarrollo del proyecto	33
Fig 17. Arquitectura de la aplicación móvil	41
Fig 18. Asignación de número de clase a cada objeto	48
Fig 19. Archivo de texto para cada imagen pornográfica	48
Fig 20. Etiquetado de imagen pornográfica en labellmg	48
Fig 21. Resultado de etiquetado de imagen pornográfica	49
Fig 22. Archivo de texto para cada imagen sangrienta	49
Fig 23. Etiquetado de imagen sangrienta en labellmg	49
Fig 24. Resultado de etiqueta de imagen sangrienta	50
Fig 25. Distribución de clases para objetos de riesgo	52
Fig 26. Características del Hardware de Notebook Google Colab	54
Fig 27. Configuración de Notebook Google Colab	55
Fig 28. Uso de Red Neuronal en Notebook Google Colab	56
Fig 29. Resultado del entrenamiento del modelo de visión artificial	56
Fig 30. Ilustración Capturas de pantalla en Android	59
Fig 31. Flujograma de procesos para la captura de pantalla	60
Fig 32. Ilustración Clasificador de Imágenes	63
Fig 33. Imagen sangrienta detectada por el clasificador	67
Fig 34. Imagen pornográfica detectada por el clasificador	67
Fig 35. Ilustración de reconocimiento de palabras ofensivas mediante OCR	69
Fig 36. Flujograma reconocimiento de palabras ofensivas mediante OCR	70
Fig 37. Flujograma envío de correo electrónico	75
Fig 38. Correo electrónico con alerta de imagen pornográfica	77
Fig 39. Correo electrónico con alerta de imagen sangrienta	77
Fig 40. Correo electrónico con alerta de palabras ofensivas "torpe"	78
Fig 41. Muestreo ANOVA con GPower	80
Fig 42. Muestreo de regresión múltiple lineal	81
Fig 43. Curva ROC	88

ÍNDICE DE TABLAS

TABLA 1. Contexto del proyecto	XX
TABLA 2. Contenido de riesgo para menores de edad	4
TABLA 3. Comparación de YOLO con otros modelos	14
TABLA 4. Estructura Equipo Scrum	33
TABLA 5. Historia de Usuario Nro. 1	34
TABLA 6. Historia de Usuario Nro. 2	34
TABLA 7. Historia de Usuario Nro. 3	35
TABLA 8. Historia de Usuario Nro. 4	35
TABLA 9. Historia de Usuario Nro. 5	36
TABLA 10. Historia de Usuario Nro. 6	36
TABLA 11. Historia de Usuario Nro. 7	37
TABLA 12. Historia de Usuario Nro. 8	37
TABLA 13. Historia de Usuario Nro. 9	37
TABLA 14. Product Backlog	38
TABLA 15. Sprint 0.....	39
TABLA 16. Planificación Sprint 0	39
TABLA 17. Revisión Sprint 0	40
TABLA 18. Herramientas tecnológicas para el proyecto.....	40
TABLA 19. Retrospectiva Sprint 0	41
TABLA 20. Planificación general de Sprints.....	42
TABLA 21. Planificación Sprint 1	42
TABLA 22 Revisión Sprint 1	43
TABLA 23. Fuentes de contenido pornográfico	43
TABLA 24. Organización dataset imágenes pornográficas	44
TABLA 25. Fuentes de contenido sangriento.....	44
TABLA 26. Organización dataset de imágenes sangrientas	45
TABLA 27. Organización dataset de imágenes de riesgo	45
TABLA 28. Retrospectiva Sprint 1	45
TABLA 29. Planificación Sprint 2	46
TABLA 30. Revisión Sprint 2	46
TABLA 31. Variables en archivo de configuración Tiny-YOLO.....	51
TABLA 32. Organización dataset para entrenamiento.....	51
TABLA 33. Retrospectiva Sprint 2	57
TABLA 34. Planificación Sprint 3	58
TABLA 35. Revisión Sprint 3	58
TABLA 36. Retrospectiva Sprint 3	61
TABLA 37. Planificación Sprint 4	62
TABLA 38. Revisión Sprint 4	62
TABLA 39. Tecnologías para Clasificador de Imágenes.....	63
TABLA 40. Retrospectiva Sprint 4	68
TABLA 41. Planificación Sprint 5	68
TABLA 42. Revisión Sprint 5	69
TABLA 43. Tecnologías para reconocimiento de palabras ofensivas mediante OCR.....	70

TABLA 44. Retrospectiva Sprint 5	73
TABLA 45. Planificación Sprint 6	74
TABLA 46. Revisión Sprint 6	74
TABLA 47. Retrospectiva Sprint 6	78
TABLA 48. Estructura matriz de confusión.....	82
TABLA 49. Resultados por clase	83
TABLA 50. Resultados en Android 6.....	84
TABLA 51. Resultados en Android 7.....	85
TABLA 52. Resultados en Android 8.....	86
TABLA 53. Resultados en Android 9.....	86
TABLA 54. Resumen de Resultados.....	87
TABLA 55. Set de imágenes de prueba	87

RESUMEN

En los últimos años, el uso de teléfonos inteligentes en niños y adolescentes ha incrementado en un considerable número y por dicha razón los peligros que esta población de personas se enfrenta es cada vez mayor, debido a muchos factores sociales tales como la poca supervisión por medio de un adulto o el desconocimiento de algunos de ellos sobre los riesgos a los cuales sus representados se ven amenazados. Además, la falta de conocimiento sobre aplicaciones que permiten disminuir la presencia de estos riesgos ha llevado a la humanidad a no tomar cartas en el asunto en cuanto a proteger la integridad de sus hijos en el mundo tecnológico y del internet se trata.

En la actualidad la tecnología como la inteligencia artificial y los mismos teléfonos inteligentes tienen la capacidad de ser muy útiles en cuanto a control parental se trata.

El presente trabajo de titulación tiene como finalidad desarrollar una a aplicación móvil Android, para la detección de contenido riesgoso en teléfonos inteligentes usados por niños y adolescentes, basado en técnicas de visión artificial que en una implementación futura sea una herramienta que corrobore en la lucha de monitorear el contenido de riesgo presenciado en teléfonos inteligentes usados por niños y adolescentes que para esto se realizó una revisión bibliográfica para conocer el tipo de contenido por analizar.

Además, se utilizó tecnologías como visión por computador, Android Studio, librerías de Inteligencia artificial como redes neuronales. Todo esto siguiendo el marco de trabajo SCRUM y los resultados fueron validados mediante métricas competentes usando el Lenguaje de programación R.

La función que cumple la aplicación móvil es capturar la pantalla automáticamente y analizar su contenido para en caso de reconocer contenido restringido notificar al padre de familia o tutor de quien lo está usando el dispositivo.

Palabras Claves: Visión por computador, contenido riesgoso, Android Studio, Redes Neuronales Convolucionales, teléfonos inteligentes.

ABSTRACT

In recent years, the use of smartphones in children and adolescents has increased in a considerable number and for this reason the dangers that this population of people face is increasing, due to many social factors such as little supervision through of an adult or the ignorance of some of them about the risks to which their clients are threatened. In addition, the lack of knowledge about applications that reduce the presence of these risks has led humanity not to take action on the matter in terms of protecting the integrity of their children in the technological and internet world.

Today technology such as artificial intelligence and smartphones themselves have the ability to be very useful when it comes to parental control.

The purpose of this degree work is to develop an Android mobile application for the detection of risky content on smartphones used by children and adolescents, based on artificial vision techniques that in a future implementation will be a tool that corroborates in the fight of monitoring the risk content witnessed in smartphones used by children and adolescents, for which a bibliographic review was carried out to know the type of content to be analyzed.

In addition, technologies such as computer vision, Android Studio, artificial intelligence libraries such as neural networks were used. All this following the SCRUM framework and the results were validated by competent metrics using the R programming language.

The function of the mobile application is to automatically capture the screen and analyze its content in case it recognizes restricted content, notify the parent or guardian of whoever is using the device.

Keywords: Computer vision, risky content, Android Studio, Convolutional Neural Networks, smartphones.

INTRODUCCIÓN

1. Antecedentes

Según la Unicef, “más de 175.000 niños se conectan todos los días por primera vez a internet, un nuevo niño cada medio segundo, convirtiéndose en los más vulnerables de los riesgos y daños detectados en el ciber espacio como el acceso a contenido perjudicial, explotación y abuso sexual, ciberacoso y uso indebido de su información privada, entre otros. La proporción de niños y adolescentes afectados oscila entre el 5% y el 21%, las niñas parecen tener más probabilidades de sufrir este fenómeno que los niños. Cifra que fue revelada por la Unesco, a través de un estudio realizado en países de altos ingresos. Sin embargo, la realidad en Ecuador no es diferente”.(Martinez, 2019)

Según(Livingstone, 2019), la situación actual de los niños se encuentra más expuestos a los mensajes de odio (del 13% al 20%), sitios pro-anorexia (del 9% al 13%), sitios de autoagresión (del 7% al 11%), y mentira cyber-bullying (del 8% al 12%).

2. Situación actual

Los niños y adolescentes tienen cada vez más acceso a un teléfono inteligente y acceso a internet sin la supervisión de los padres o educadores, situación encontrada en países europeos, pero no varía a nivel mundial y mucho menos en Ecuador.(UNICEF, 2017)

Ante esta preocupación, el papa Francisco, máximo representante de la Iglesia Católica hizo el pedido durante la audiencia que concedió el 14 de noviembre 2019 en el Palacio Apostólico del Vaticano a los participantes en el congreso sobre “Promoción Digital de la Dignidad de la Infancia: del concepto a la acción”.(Aciprensa, 2019) afirma:

En su discurso, el Pontífice recordó que las empresas tecnológicas no deben limitarse a desarrollar nuevas tecnologías y desentenderse de los malos usos que se les pueda dar. Por eso pidió que sean parte de la solución y que participen en la búsqueda de estrategias para proteger a los menores en los entornos digitales (párr. 3).

3. Planteamiento del problema

Un número mayor de pruebas empíricas revelan que los niños acceden a internet en edades cada vez más tempranas. En algunos países, los niños menores de 15 años tienen la misma probabilidad de usar internet que los adultos mayores de 25 años. (UNICEF, 2017)

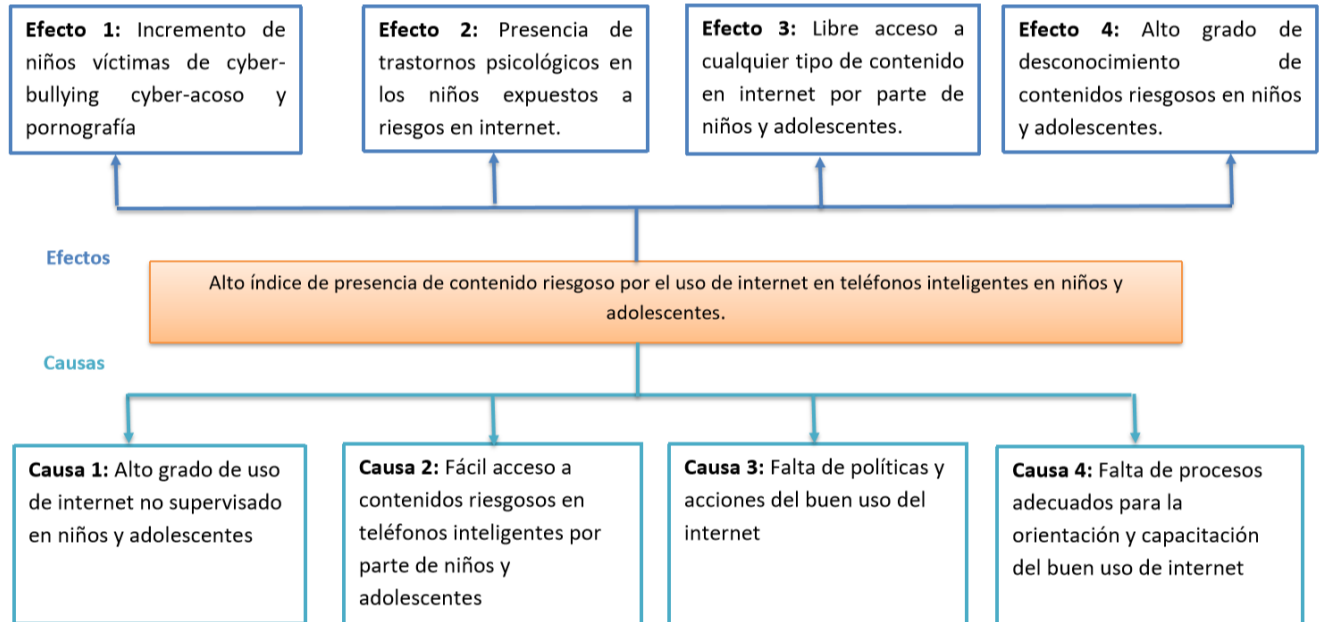


Fig 1. Árbol de problemas
Fuente: Elaboración propia

4. Objetivos

5. Objetivo general

Desarrollar una aplicación móvil Android para la detección de contenido riesgoso en teléfonos inteligentes usados por niños y adolescentes, basado en técnicas de visión artificial.

6. Objetivos específicos

- Elaborar un marco teórico que sustente el contenido del proyecto.
- Construir datasets que contengan palabras e imágenes consideradas de riesgo para niños y adolescentes.
- Desarrollar una aplicación móvil Android, mediante técnicas de visión artificial, aplicando SCRUM como marco de trabajo de desarrollo.
- Validar la precisión y características operativas de la aplicación de visión artificial desarrollado, empleando el lenguaje de programación R.

7. Alcance

Realizar una base teórica que abarquen temas como: contenido considerado de riesgo para niños y adolescentes, algoritmos de visión artificial para reconocimiento de texto y objetos en imágenes digitales que puedan usarse mediante la librería de OpenCV.

Este proyecto tiene como finalidad desarrollar una aplicación móvil en Android Studio e implementarla, que permita detectar y alertar cuando el usuario esté expuesto a contenido considerado riesgoso para niños y adolescentes. Para esto, se recurrirá a investigaciones de instituciones como UNICEF, INFA, y otras instituciones relacionadas al tema mediante internet, en caso de no encontrar información se acudirá personalmente a instituciones de la localidad que brinden este tipo de información.

Con la información obtenida se construirá datasets con imágenes y texto en español que se situarán localmente en el computador de desarrollo.

Se realizará una revisión bibliográfica para seleccionar un algoritmo que permita detectar texto y objetos en imágenes, que haya sido implementado antes en otras investigaciones y que genere los mejores resultados para así usarlo en la presente solución. La implementación futura de esta aplicación permitirá la reducción de exposición de niños a riesgos que implica el uso de un teléfono móvil.

Para la detección de imágenes riesgosas es necesario elaborar el modelo de visión artificial usando el dataset de imágenes previamente construido. En el desarrollo de la aplicación móvil se usará la librería OpenCV y se deberá seleccionar un algoritmo compatible entre sí. De esta forma se procesarán las imágenes para analizar su contenido.

Las imágenes que serán procesadas se obtendrán mediante capturas de pantallas automáticas por parte del Sistema Operativo (OS) Android, y si se logra detectar algún elemento clasificado como riesgo se enviará un mensaje de alerta mediante correo electrónico al tutor o persona, que represente al niño o adolescente.

La validación de los resultados se realizará mediante el lenguaje de programación R para:

- Cálculo de precisión, exactitud, sensibilidad, especificidad y F1 del prototipo de visión artificial
- Características operativas por curva ROC.

Precisión, exactitud, sensibilidad, especificidad, F1 y ROC, son métricas que se utilizan para evaluar aplicaciones o modelos basados en clasificación dentro de inteligencia artificial permitiendo verificar ciertos datos mediante la representación de gráficos estadísticos.

Cabe mencionar que la aplicación funcionará en celulares inteligentes y que cumplan ciertas características de procesamiento.

La aplicación se entregará en formato APK la misma que se no necesitará de internet para clasificar las imágenes entre riesgosas o no, solamente para enviar correos electrónicos de alertas se usará internet.

Para el desarrollo del proyecto se aplicará el marco de trabajo SCRUM.

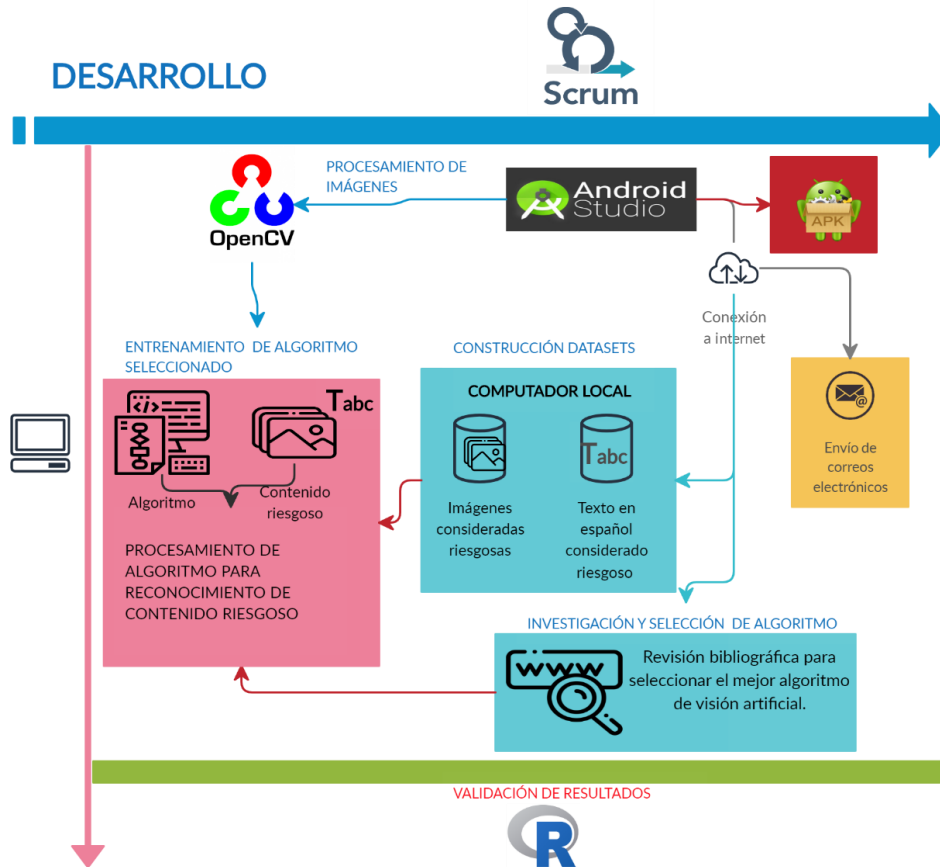


Fig 2. Arquitectura del proyecto
Fuente: Elaboración propia

IMPLEMENTACIÓN

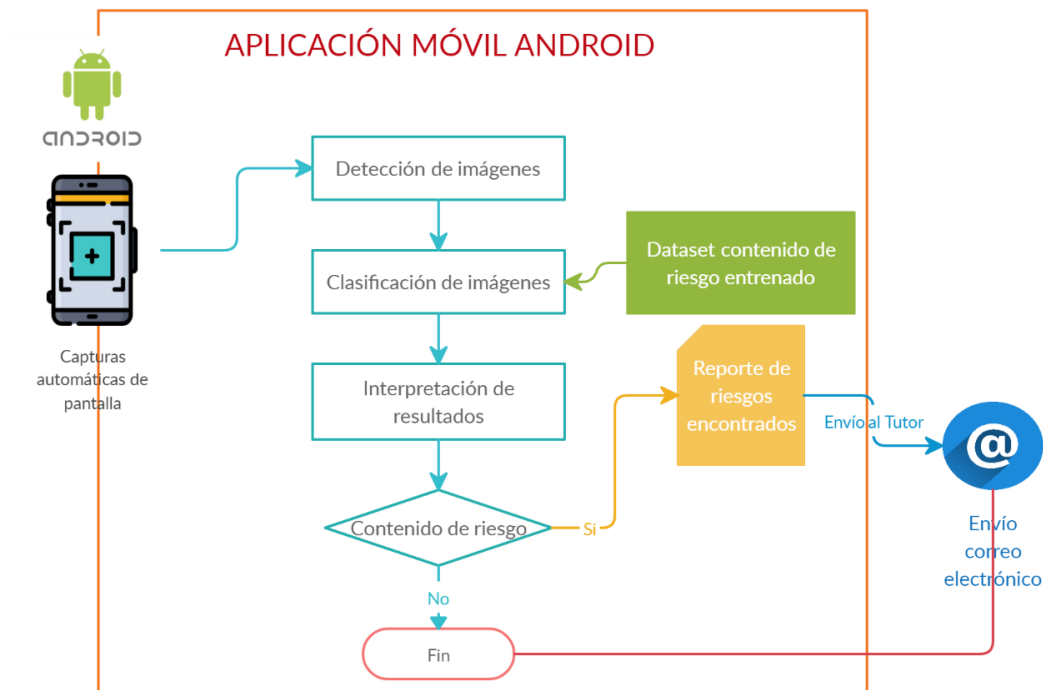


Fig 3. Funcionamiento de la aplicación
Fuente: Elaboración propia

8. Justificación

Este proyecto estará enfocado a los objetivos de desarrollo sostenible, especialmente con el Objetivo 16 que destaca un problema social muy importante a contrarrestar ya que involucra a los niños con violencias, amenazas de homicidio, la trata de personas y violencia sexual (Naciones Unidas, 2015).

Entre las metas importantes que destaca este objetivo se cita en el literal 16.2. en donde (Naciones Unidas, 2015) destaca «Poner fin al maltrato, la explotación, la trata y todas las formas de violencia y tortura contra los niños»(Metas del objetivo 16, párr. 2).

Las empresas de tecnología e internet deberían tomar medidas para evitar que sus redes y servicios sean utilizados por delincuentes para recopilar y distribuir imágenes de abuso sexual infantil o cometer otras violaciones contra los niños. (UNICEF, 2017).

Justificación Social. - La presente solución ayudará a padres de familia de la localidad a monitorear el tipo de contenido que sus hijos se exponen con el acceso a internet en una implementación futura para evitar el incremento de niños y adolescentes con problemas sociales y psicológicos.

Justificación Tecnológica. - Hacer uso de las nuevas tecnologías que permitan corroborar con soluciones a problemas que enfrenta la sociedad con el día a día, existen varias investigaciones sobre aplicaciones que usan técnicas de visión artificial, por ello este proyecto se basará en sus experiencias para realizarse.

Tras revisiones bibliográficas se encontraron investigaciones como: Optical Character Recognition using KNN on Custom Image Dataset (Hazra et al., 2017), Deep Learning for Image-to-Text Generation (He & Deng, 2017) ,A Text Extraction Approach towards Document Image Organisation for the Android Platform (Madhuram & Parameswaran, 2018), You Only Look Once: Unified, Real-Time Object Detection (Redmon, 2016), Parental controls: advice for parents, researchers and industry(Zaman & Nouwen, 2016).

Justificación Económica. - Al existir daños psicológicos en niños y adolescentes por distintos riesgos encontrados en internet, se debe acudir a rehabilitaciones con especialistas generando gastos económicos en la familia del afectado, con la presente solución en una futura implementación se evitarían estos gastos y de esta manera se colabora con la economía de las personas que en caso no lo tienen y deben recurrir a préstamos.

Justificación Metodológica. - El proyecto se basa de una investigación documental recopilando las mejores investigaciones basado en intereses similares, para la mejora continua de proyectos de visión artificial orientado a resolución de problemas sociales más destacadas para la elaboración de la presente solución.

Justificación Ambiental. - La aplicación móvil será amigable con el medio ambiente por no hacer uso constante de conexión a internet para la captura de objetos en imágenes.

9. Contexto

En la actualidad existen varias investigaciones y proyectos que apoyan al control parental, a pesar de eso las técnicas que usan no son tan sofisticadas como la utilización de algoritmos de visión artificial y de igual manera existen investigaciones de visión artificial que no se enfocan en la resolución de este tipo de problemas sociales.

Para dar a conocer los trabajos anteriores se refleja un resumen en la TABLA 1.

TABLA 1. Contexto del proyecto

Tema	Diferencia	Aporte	Cita /autor
------	------------	--------	-------------

Implementación de una aplicación Móvil Android como una herramienta de aprendizaje del Idioma Kichwa Otavalo a partir del inglés.	La aplicación móvil no utiliza técnicas de visión artificial para la solución plantada.	Se utilizará técnicas de visión artificial, tecnología que en la actualidad es eficiente.	Quilumbaqui Santacruz, Carlos Edison. (2018) Universidad Técnica del Norte
Prototipo de un control parental para el internet en el hogar, operado desde un dispositivo Android	En este caso no se realiza técnicas tan avanzadas para detectar contenido riesgos.	Se aplicará técnicas de visión artificial que ayude a tomar decisiones al padre de familia de manera más rápida con una visión clara de la situación que enfrenta el niño o adolescente	Edith Solórzano, Juan Bohórquez (2016) Universidad de Guayaquil
Online Risks, Sexual Behaviors, And Mobile Technology Use In Early Adolescent Children: Parental Awareness, Protective Practices, And Mediation	Investigaciones sobre riesgos en línea hacia adolescentes de 11 a 14 años, orientado a EE. UU y prácticas de protección parental.	Además de la investigación se propone una aplicación que permita detectar riesgos en teléfonos inteligentes y en lenguaje español.	Allison, Kendra (2018) Estados Unidos

CAPÍTULO 1

Marco Teórico

En este capítulo se evidencia la revisión bibliográfica realizada que permita sustentar el presente trabajo. Por ello es importante mencionar los tipos de peligros que los niños y adolescentes se enfrentan al usar teléfonos inteligentes, especialmente por el acceso al internet, ya que éste es el principal problema que ha motivado a la elaboración de este proyecto. Al ser un área tecnológica se denota importante también las aplicaciones móviles Android especialmente las que usan técnicas de visión artificial. A la investigación se suman el marco de desarrollo de trabajo de software SCRUM ya que permite seguir un flujo controlado para el cumplimiento de los objetivos. También se mencionan los tipos de validaciones que se puede realizar a una aplicación de visión artificial como la precisión y características operativas que para esto se utiliza el lenguaje de programación R, obteniendo una validación estadística de lo mencionado. A continuación, se ilustra un gráfico que detalla la estructura de la revisión bibliográfica realizada.

1.1. Menores de edad y la tecnología

Los niños y adolescentes en la actualidad tienen el acceso a un teléfono inteligente y a internet sin la supervisión de los padres o educadores, situación encontrada en países europeos, pero no varía a nivel mundial y mucho menos en Ecuador. La tecnología digital está cambiando el mundo, especialmente si se trata de la infancia de los niños que actualmente tienen la facilidad de obtener contenido de internet. A esta era tecnológica se involucran los niños y adolescentes menores de 18 años, ya que representan uno de cada tres usuarios conectados a nivel mundial. Agregado a esto se hace énfasis en la manipulación de teléfonos inteligentes que están privatizando a los niños de acoplarse a su entorno social. (UNICEF, 2017)

Según (United Nations, 2014) presentó un cuadro con resultados de encuestas a nivel Latinoamérica en donde refleja un alto porcentaje en Ecuador del acceso al computador, internet y teléfono móvil, para más detalles verla Fig 4.

América Latina (12 países): acceso a computador, Internet y teléfono móvil por hogar en niños y niñas de 6 a 12 años y de 13 a 18 años, alrededor de 2012
(En porcentajes)

País	Tramo etario	Acceso a teléfono móvil	Acceso a computador	Acceso a Internet
Bolivia (Estado Plurinacional de), 2011	6 a 12 años	82	18	5
	13 a 18 años	87	24	7
Brasil, 2012	6 a 12 años	91	47	39
	13 a 18 años	92	51	44
Chile, 2011	6 a 12 años	98	61	45
	13 a 18 años	99	66	51
Colombia, 2012	6 a 12 años	93	29	22
	13 a 18 años	94	35	28
Costa Rica, 2012	6 a 12 años	97	49	49
	13 a 18 años	97	55	54
Ecuador, 2012	6 a 12 años	88	31	19
	13 a 18 años	90	36	25
El Salvador, 2012	6 a 12 años	94	18	9
	13 a 18 años	95	22	12
Nicaragua, 2009	6 a 12 años	56	4	1
	13 a 18 años	62	5	1
Paraguay, 2011	6 a 12 años	93	24	19
	13 a 18 años	94	29	22
Perú, 2012	6 a 12 años	82	26	17
	13 a 18 años	85	30	21
Uruguay, 2012	6 a 12 años	98	95	49
	13 a 18 años	98	91	56
Venezuela (República Bolivariana de), 2012	6 a 12 años	43	41	25
	13 a 18 años	44	42	29

Fig 4. Acceso a los aparatos tecnológicos en América Latina
Fuente: (United Nations, 2014)

1.1.1. Ventajas

- a) Según (Ebro, 2019), una de las ventajas importantes que los nativos de la era digital perciben es la familiarización con herramientas de este tipo, ya que les permitieran desenvolverse autónomamente es su futuro indagando y buscando mejoras para la evolución del mismo.

1.1.2. Desventajas

- a) El ciberacoso es uno de los principales peligros que los niños, niñas y adolescentes se enfrentan al usar las redes sociales y el internet ya que les permiten relacionarse con cualquier tipo de persona por eso una buena educación influye bastante para contrarrestar estos ataques, aunque una buena forma de prevenir estas situaciones es el uso de aplicaciones de control parental (Ebro, 2019).
- b) Acceso fácil a contenido no acorde a su edad, por ejemplo, al contenido violento o sexual y todas sus consecuencias.

1.1.3. Usos de Internet

El uso del internet se ha incrementado en los últimos años debido a la adquisición de tecnología para cumplir con ciertas actividades diarias, además que los precios de los aparatos electrónicos han disminuido sus costos. También cuenta la facilidad que existe al compartir contenido con otros usuarios gracias al software como un servicio en la nube, ya permite usar ciertos programas instalados en un servidor remoto sin la necesidad de tenerlo en nuestro propio dispositivo (Serrano-Cobos, 2016).

Uno de los usos que representa el internet es el de “biblioteca”, ya que reenvía grandes cantidades de información que pone al alcance de las personas tales como: diccionarios, revistas, archivos, imágenes, videos, entre otros. Se puede acceder a internet para obtener diferentes materiales de interés orientado a estudiantes (Adell, 2014).

Otro uso es como canal de comunicación de las experiencias de aprendizaje en las que participan personas como docentes y alumnos de varios centros e incluso de diversos países y que usan la Internet para comunicarse entre sí y para intercambiar información. Los proyectos de aprendizaje, en los que estudiantes investigan y buscan respuesta a una serie de cuestiones del currículum utilizando medios, tecnologías y fuentes de información (Adell, 2014).

Entre otras actividades que se pueden realizar con el uso del internet es comprar productos de diverso tipo en las distintas plataformas existentes, como también comunicarse con personas a través de plataformas de comunicación por video y jugar juegos online con personas de cualquier parte del mundo (Raffino, 2019).

a) Redes sociales

Según (United Nations, 2014) algunas de las opiniones de los adolescentes sobre las redes sociales son las siguientes:

- “Facebook representa un lugar en el que me puedo comunicar con mis amigos, subir fotos mías o con ellos. Es mi perfil virtual”. Rocío, 14 años.
- “Si no hubiera, se acabaría el cyberbullying, los acosos y todo eso”. Santiago, 13 años.
- “Lo bueno es que me comunico, lo malo es tener un malentendido porque por mensaje escrito uno no percibe cómo te dice las cosas el otro”. Salvador, 14 años.

- “Lo malo es que hay gente que te agrega sin saber quién eres, para tratarte mal o para acosarte”. Rocío, 12 años.
- “Yo lo uso para poner algunas fotos con mis amigos, lo malo es que todas mis publicaciones están expuestas a personas desconocidas”. Facundo, 17 años.

1.1.4. Riesgos que asechan a menores en internet.

Entre las preocupaciones más tratadas con el uso del internet tiene que ver con el acceso a contenidos inadecuados, analizando en primer lugar los contenidos a los que los menores acceden de forma voluntaria. Un gran porcentaje de adolescentes busca activamente páginas con un contenido sexual explícito, un 5.7% lo hace con mucha frecuencia y el resto, en ocasiones o rara vez. El hombre de entre 15 y 17 años constituye el perfil tipo de los menores que acceden a este tipo de contenidos(Catalina García et al., 2014).

Los riesgos que más comúnmente se enumeran a la hora de hablar de internet relacionado con los menores de edad son el acceso a contenidos inapropiados -violentos o pornográficos-, el contacto con desconocidos o un mal uso de los de los datos personales.(Jiménez Iglesias et al., 2015).

El contacto con desconocidos es un riesgo realizado por los menores que podrían comprometer su seguridad y privacidad online. Una gran cantidad de los adolescentes utiliza Internet con frecuencia con el objetivo de “ligar” y un 20.5% para buscar nuevos amigos. De estos un 7.6% dice hablar con desconocidos de su misma edad y el 3.8% con desconocidos de diferentes edades(Catalina García et al., 2014).

Algo más de un tercio de los más pequeños y casi un 55% de los que tienen entre 15 y 17 años ha identificado a su alrededor la circulación de comentarios o imágenes ofensivas en el ámbito digital. A continuación, aparecen acciones relacionadas con insultos, burlas amenazas, que son percibidas por cuatro de cada diez menores de 15 a 17 años. Estas situaciones de acoso en el entorno adolescente resulta superior entre los colegios e institutos privados que entre los públicos(Catalina García et al., 2014).

En conclusión, de las distintas fuentes se encontró cierto contenido de riesgo para menores de edad que el presente proyecto debe trabajar en ello para disminuir los efectos secundarios de estos peligrosos sucesos, por esto ver la TABLA 2.

TABLA 2. Contenido de riesgo para menores de edad

Riesgo	Tipo de Contenido
Ciberacoso	Palabras sexuales
Ciberbullying	Palabras de burla
Pornografía	Imágenes sexuales
Violento	Imágenes sangrientas

1.2. ¿Control Parental o violación de derechos?

En este apartado se citan dos grandes temas por aclarar debido a que los padres como tutelares de sus hijos menores de edad tienen ciertos derechos que cumplir hacia ellos debido a que estos no son lo suficientemente maduros para tomar sus propias decisiones. Y por otro lado son los mismos derechos que podrían evitar la intervención excesiva de los padres en la vida de sus hijos.

1.2.1. Control Parental

Debido al uso no controlado de los dispositivos móviles en menores de edad, los riesgos que se asocian pueden aparecer en cualquier momento, por esto se menciona una regulación de las actividades por parte de los padres en sus dispositivos para mitigar los riesgos posibles que puedan aparecer (Allison, 2018).

Además, en el mercado existen varios controles de aplicaciones para que los padres sientan la seguridad de proteger a sus hijos de riesgos como contenido inapropiado en línea y ciberacoso entre los principales (Zaman & Nouwen, 2016).

1.2.2. Derechos de los menores de edad

A pesar de la existencia de los derechos de las personas y en especial los menores de edad, también se cuenta con las obligaciones de los tutores/representantes de estos para enseñar, educar y velar por sus derechos, existe una relación muy delicada que cuidar, ya que las primeras personas en violar sus derechos podrían ser quienes están a su cargo.

Para la elaboración de este proyecto se debe tomar muy en cuenta estos aspectos ya que se pretende inmiscuir en las comunicaciones de los menores, especialmente espiando el tipo de contenido digital que ellos reciben o manipulan al momento de usar un teléfono inteligente.

Se podría hablar de una violación a la correspondencia de los pequeños, hay que acatarse a las leyes que nos rigen para actuar con seguridad y responsabilidad.

Por este motivo se muestra a continuación leyes establecidas en las diferentes organizaciones que tratan este tema además de las diferentes opiniones encontradas para llegar a un consenso sobre el este tema.

Según afirma (Unicef, 2006), en el artículo 16 de la Convención Internacional sobre los Derechos del Niño de las Naciones Unidas establece: “Ningún niño será objeto de injerencias arbitrarias o ilegales en su vida privada, su familia, su domicilio o su correspondencia ni de ataques ilegales a su honra y a su reputación”(p.15).

De igual forma en el código de la niñez y adolescencia, correspondiente a la Ley No. 100. en Registro Oficial 737 de 3 de enero del 2003, Según menciona (ACNUR, 2003):

. 60: Derecho a ser consultados. - Los niños, niñas y adolescentes tienen derecho a ser consultados en todos los asuntos que les afecten. Esta opinión se tendrá en cuenta en la medida de su edad y madurez. Ningún niño, niña o adolescente podrá ser obligado o presionado de cualquier forma para expresar su opinión.

Debido a esto, se entendería que ni los mismos padres de familia podrían inferir en su vida privada, pero aún queda pendiente el tema del rol como padre de familia o tutelar hacia el menor de edad, y especialmente si de velar por sus derechos se trata.

Por otro lado, en la misma ley el Art.73 según (ACNUR, 2003) indica que:

Art. 73.- Deber de protección en los casos de maltrato. - Es deber de todas las personas intervenir en el acto para proteger a un niño, niña o adolescente en casos flagrantes de maltrato, abuso sexual, tráfico y explotación sexual y otras violaciones a sus derechos; y requerir la intervención inmediata de la autoridad administrativa, comunitaria o judicial.

Además en relación al Art.74, (ACNUR, 2003) también menciona que: “El Estado adoptará las medidas legislativas, administrativas, sociales, educativas y de otra índole, que sean necesarias para proteger a los niños, niñas y adolescentes”, además que “impulsará programas dirigidos a la prevención e investigación de los casos de maltrato, abuso y explotación sexual, tráfico y pérdida”.

Una vez citado lo anterior (Velasco;Calle, 2015) expone que:

Esto no limita a sus padres, como titulares de la patria potestad del menor, de acceder a las comunicaciones de sus hijos en Internet. Sin embargo, cabe pensar ¿qué tan sano es para un niño, saberse vigilado a cada segundo por sus padres en Internet? (párr.3).

En Colombia la Corte Suprema ha dado su opinión acerca de este tema mencionando que es un control legal ya que no pretende violar la privacidad de los niños, sino que pretende protegerlos, especialmente sus padres deben acatar el deber que los corresponde de velar por la seguridad de sus pequeños y su adecuado desarrollo en el mundo virtual (Velasco;Calle, 2015).

De igual existen opiniones sobre este tema, de si los padres pueden violar realmente la intimidad de los menores en Europa. Especialmente en España también se acatan a las leyes mencionadas y dictadas por (Unicef, 2006), también se menciona sobre el nivel de madurez del menor para evaluar la intervención de sus tutores. Por esto basado en el art. 5 de la Convención de los Derechos del Niño de las Naciones Unidas de 20 de noviembre de 1989 (CDNNU)". (Sepín, 2016) menciona: "su capacidad de decisión no podrá ser sustituida ni anulada por sus progenitores o representantes, quienes no podrán más que complementar u orientar la decisión de estos".

Por lo tratado anteriormente, es importante tener una buena comunicación con los menores, especialmente con aquellos que se encuentran atrapados en el mundo de la tecnología para conocer cuáles son sus intereses acerca de esto. La idea de supervisar no es leer todos sus mensajes de aplicaciones sino establecer ciertas normas como: supervisar el horario de uso , conocer que aplicaciones utilizan constantemente, hablar con ellos sobre los peligros que ellos pueden enfrentarse y entre lo más importante como padre de familia hacer cumplir sus derechos y sus normativas tomando en cuenta que son menores de edad y su madurez aun no es suficiente para permitir que ellos tomen sus propias decisiones en cuanto a que contenido les brindará un buen desarrollo físico y mental (Sepín, 2016).

1.3. Inteligencia Artificial

La inteligencia Artificial (IA) es la capacidad que se le brinda a las máquinas o aparatos electrónicos para usar algoritmos, los cuales les permite tomar decisiones y seguir aprendiendo en base a los datos brindados inicialmente, algo similar de cómo los humanos lo hacen. Las grandes diferencias entre las funciones que realizan los dispositivos basados en IA y las personas es la no necesidad de descanso, ya que puedes trabajar con múltiples tareas y volúmenes extensos de información, de igual forma el margen de error con el que manipula los

datos el aparato es mucho menos a las que un humano puede reflejar en el cumplimiento de las tareas. En la actualidad los sistemas basados en inteligencia artificial pueden realizar muchas tareas y de forma más eficiente que las personas lo realizan, gracias a su capacidad constante de aprender y la facilidad de tomar decisiones. (Rouhiainen, 2018).

1.3.1. Visión artificial

Es una rama de la inteligencia artificial que permite entrenar a las computadoras para entender lo que pasa en el mundo real de una forma visual, para ello utiliza imágenes obtenidas mediante cámaras fotográficas o a partir de videos y además utiliza algoritmos de computación. Los dispositivos entrenados para este tipo de procesamiento pueden hasta clasificar los objetos detectados para posterior a ello tomar decisiones según lo programado.

1.3.2. Fases de la Visión Artificial

Existen varias fases dentro de la visión artificial y la consideración de estas depende del alcance de la aplicación, por ello fueron tomadas algunas de las más importantes, para esto se ha recolectado información de distintos autores y según (Alvear, 2017), (SAS, 2020) y (Minardi, 2016) detallan a continuación las siguientes:

- a) Adquisición de Imágenes:** Es un proceso en donde adquiere una imagen del mundo real para transformar en imagen digital.
- b) Preprocesamiento:** Se somete a un preprocesamiento la imagen digital obtenida en un proceso anterior, para eliminar partes que no serán tomadas en cuenta en los procesos siguientes.
- c) Detección de bordes:** En este apartado se efectúa la separación de los objetos de interés y del fondo de la imagen, para de esta manera resaltar el contenido que se desea analizar.
- d) Segmentación:** En este proceso convierte la imagen compuesta por pixeles a un conjunto de regiones para de esta manera distribuir el análisis y detectar los objetos por determinados colores.
- e) Extracción de Características:** Se extraen las características importantes o patrones sobre un objeto en específico como por ejemplo los puntos del rostro llamados landmarks o de cualquier otro objeto.
- f) Reconocimiento:** En esta fase se determina si la imagen contiene un objeto en específico, como por ejemplo en las cámaras de celulares inteligentes permiten

detectar si un rostro está presente o de igual manera en las fotografías de las redes sociales.

- g) Interpretación:** Haciendo uso de varias imágenes que contengan un objeto determinado es posible construir escenarios que interpreten el contexto de ese contenido.

1.3.3. Métodos de reconocimiento de objetos

Según (Chaves et al., 2018) existen varios algoritmos para esta funcionalidad, alguno de ellos utilizan estrategias distintas para el análisis de la imagen, por ello se distribuirá en distintos grupos:

- a) Estrategias Basadas en Ventana Deslizante:** Para localizar objetos se hace uso de una ventana que puede variar de dimensiones y recorre por toda la imagen en donde evalúa una función para conocer si en dicha región se encuentra o no el objeto de interés, entre estos algoritmos se encuentran.

- **Viola and Jones (2004).** – Hace uso de clasificadores en cascada, los cuales permiten se busque de forma eficiente y rápida. Para buscar objetos se emplea una ventana deslizante de 24 x24 píxeles y está especializado para la búsqueda de rostros en imágenes.
- **HOG+SVM.** – Utiliza un descriptor basado en el gradiente y utiliza una función de corrección de gama para la reducción de efectos de iluminación, es muy robusto con este tipo de problemas. Aun que su principal desventaja es el trabajo con el cambio de tamaño de objetos.
- **DPM.** – Utiliza una búsqueda de ventana deslizante y cada región está formada por conjuntos de filtros de las diferentes partes que se compone el objeto y además permite ubicar al objeto en diferentes poses. La desventaja de este algoritmo se debe al tiempo considerado de procesamiento que necesita para detectar objetos en modelos complejos.
- **ESS.** – Es un algoritmo de ramificación que utiliza un método de búsqueda llamado Búsqueda Eficiente en sub-ventanas que permite la identificación de las regiones en donde puedan existir objetos.

- b) Estrategias Basadas en Conjuntos de Regiones Candidatas:** Debido a los inconvenientes para localizar los objetos con distintos tamaños, se busca analizar un

gran número de regiones además de añadir métodos más sofisticados que mejore la calidad de la detección, por ello se da origen a los métodos de propuesta de regiones candidatas, los cuales generan regiones con altas probabilidades de poseer el objeto deseado, gracias a la teoría de que los objetos compartan características en las imágenes y que a la vez son diferentes a su fondo.

- **Objectness.** – Para determinar si en una región se encuentra un objeto implementa el uso de una métrica en donde datos como color, bordes y los píxeles de la imagen son esenciales elementos.
- **Selective Search.** - Junta super píxeles para formar regiones y así localizar al objeto, por ello deben existir similitudes en color, textura, tamaño y la forma de las regiones, para esto utilizan un algoritmo basado en grafos.
- **Edge Boxes.** – Usa la estrategia de ventana deslizante y la información de los bordes para crear de regiones candidatas.

c) Estrategias Basadas en Aprendizaje Profundo: Una característica importante de estos métodos es su aprendizaje a partir de los datos (píxeles). En este episodio se integra a las Redes Neuronales Convolucionales (CNN) ya que son líderes en aprendizaje profundo especializadas en análisis de imágenes. Un dato importante es el nivel de capas de profundidad existentes en la red, mientras mayor es el número, tiene más posibilidades de aprender estructuras complejas en las imágenes.

Las capas iniciales identifican características como esquinas, y bordes, mientras que las capas finales obtienen características más complejas, como patrones en la imagen o texturas formadas por esquinas y bordes. Se menciona que desde el 2012 ha incrementado el uso de este tipo de estrategias.

- **RCNN.** – En primera instancia utiliza el método Selective Search para obtener regiones candidatas para luego utilizar CNN en su proceso de clasificación y conocer la existencia del objeto de interés. Cabe recalcar que este método tiene buena precisión a cambio de un alto costo computacional debido a que analiza al menos 2000 regiones candidatas por imagen, necesita un promedio de 50 segundos para analizar una imagen y además se debe entrenar tres modelos diferentes para la detección de los objetos.

- **Fast-RCNN.** – Como su nombre lo indica, es una versión de RCNN más rápida y eficiente, esto debido a que el método comparte los cálculos para analizar las regiones candidatas y no lo hace individualmente como el anterior método, todo esto empleando una técnica conocida como Conjunto de Regiones de Interés.
- Además, la red CNN y la regresión de las regiones candidatas se entrenan en un solo modelo de red a diferencia que RCNN debe entrenar tres modelos, en otras palabras, utiliza una sola red neuronal y requiere alrededor de 20 segundos para analizar una imagen.
- **Faster-RCNN.** – Este método utiliza un sistema de propuesta de regiones candidatas (RPN) que reemplaza al antiguo método externo Selective Search que sirve para generar el conjunto de regiones candidatas en un ciclo inicial, para así convertirlo en un método más eficiente y rápido.
 - Este sistema RPN funciona a partir de la primera iteración de la red convolucional para la detección de objetos.
 - De esta forma evita cálculos para la generación de región candidata ya que usa regiones de tamaño prefijado 128×128 , 256×256 y 512×512 píxeles.
 - RPN hace que se necesite 0.12 segundos por imagen para la detección, pero aun así no es muy rápido para implementarlo en aplicaciones de tiempo real como por ejemplo los vehículos autónomos.
- **YOLO.** – Del inglés You Only Look Once, utiliza CNNs en tiempo real y no emplea regiones candidatas, en su reemplazo utiliza una sola CNN para localizar los objetos con regiones rectangulares. Para localizar los objetos se trata como un problema de regresión simple. Para cada imagen se debe aprender las probabilidades de que exista un objeto que pertenezca a una clase predefinida con sus coordenadas de forma rectangular. La CNN solamente se ejecuta una vez por imagen, en una sola iteración haciendo así que se ejecute en tiempo real aun que tiene dificultades con objetos pequeños.
- Los propios creadores de este método en su publicación científica (Redmon, Joseph.; Santosh , Divvala.; Ross, Girshick.; Ali, 2016) mencionan.

Nuestra arquitectura unificada es extremadamente rápida. Nuestro modelo base YOLO procesa imágenes en tiempo real a 45 cuadros por segundo. Una versión más pequeña de la red, Fast YOLO, procesa la asombrosa cantidad de 155 fotogramas por segundo sin dejar de alcanzar el doble de mAP que otros detectores en tiempo real. En comparación con los sistemas de detección de vanguardia, YOLO comete más errores de localización, pero es menos probable que prediga falsos positivos en el fondo. Finalmente, YOLO aprende representaciones muy generales de objetos. Supera a otros métodos de detección, incluidos DPM y R-CNN, al generalizar desde imágenes naturales a otros dominios, como obras de arte (p. 779).

- A continuación, se muestra una figura (Fig. 5) explicando brevemente cuál es el proceso del algoritmo de detección de objetos YOLO.
-

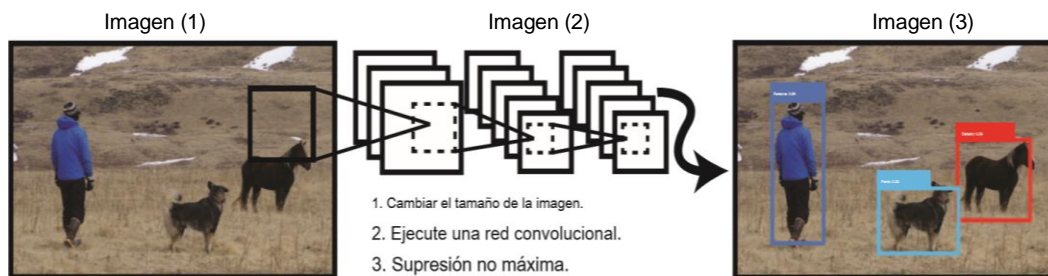


Fig 5. El sistema de detección de YOLO
Fuente: (Redmon, Joseph.; Santosh , Divvala.; Ross, Girshick.; Ali, 2016)

- En la imagen (1) el sistema redimensiona la imagen de entrada a 448×448 , luego, en la Imagen (2) sobre la imagen de entrada se ejecuta una única red convolucional, y finalmente en la Imagen (3) Muestra lo detectado por la confianza del modelo (Redmon, Joseph.; Santosh , Divvala.; Ross, Girshick.; Ali, 2016).
- YOLO es un modelo unificado con el cual se puede entrenar a imágenes completas y optimiza directamente el rendimiento de detección, brindando así mejores beneficios que los métodos tradicionales de detección de objetos (Redmon, Joseph.; Santosh , Divvala.; Ross, Girshick.; Ali, 2016).

- La red base funciona a 45 cuadros por segundo en una GPU Titan X con una velocidad a más de 150 fotogramas por segundo. Por esto se puede procesar videos en tiempo real con menos de 25 milisegundos de latencia, además este método observa la imagen en fase de entrenamiento para posteriormente realizar predicciones sobre la imagen, a diferencia de técnicas basadas en propuestas de ventana deslizante y región (Redmon, Joseph.; Santosh , Divvala.; Ross, Girshick.; Ali, 2016).
- Una dificultad que tiene este modelo es la no fácil detección de objetos pequeños cuando se agrupan, como grupo de aves por ejemplo (Redmon, Joseph.; Santosh , Divvala.; Ross, Girshick.; Ali, 2016).

Diseño de Red

- Este modelo se representa como una red neuronal convolucional en donde sus capas iniciales extraen características de la imagen de entrada y las capas intermedias predicen las coordenadas de salida (Redmon, Joseph.; Santosh , Divvala.; Ross, Girshick.; Ali, 2016).
- La arquitectura de este modelo fue basada a través de sus autores en el modelo GoogLeNet para clasificar imágenes, este modelo tiene 24 capas convolucionales y 2 capas que las conectan (Redmon, Joseph.; Santosh , Divvala.; Ross, Girshick.; Ali, 2016). Para una visión más clara de la arquitectura, se muestra la Fig. 6.

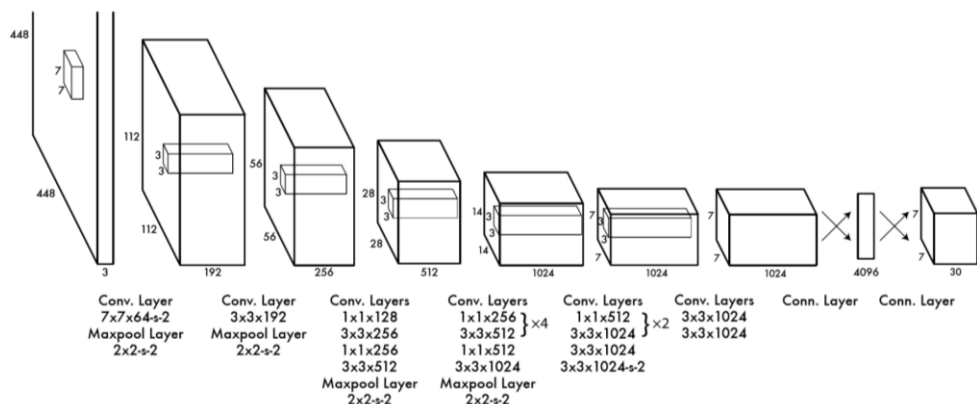


Fig 6. Red Neuronal Convolucional de YOLO
Fuente: (Redmon, Joseph.; Santosh , Divvala.; Ross, Girshick.; Ali, 2016)

- A continuación, se presenta la TABLA 3 en donde se analizan las diferencias entre YOLO y otros modelos de detección de objetos.

TABLA 3. Comparación de YOLO con otros modelos

(DPM) Modelos de piezas deformables	YOLO
Utilizan el enfoque de ventana deslizante para la detección de objetos y canalización disjunta para extraer demás características (Redmon, Joseph.; Santosh , Divvala.; Ross, Girshick.; Ali, 2016).	Reemplaza todos estos elementos con una única red neuronal convolucional (Redmon, Joseph.; Santosh , Divvala.; Ross, Girshick.; Ali, 2016).
R-CNN	YOLO
Para localizar objetos en imágenes, este modelo y sus variantes utilizan propuestas de región, para no usar ventanas deslizantes (Redmon, Joseph.; Santosh , Divvala.; Ross, Girshick.; Ali, 2016).	Propone cuadros delimitadores de alta potencia, algo parecido con R-CNN, aunque sean muchos menos, además impone restricciones espaciales que ayude con las detecciones múltiples del mismo objeto (Redmon, Joseph.; Santosh , Divvala.; Ross, Girshick.; Ali, 2016).

Fuente: Propia

- **Tiny-YOLO.** – Debido a los altos requisitos de computación y memoria que se necesita para la detección de objetos en dispositivos integrados se ha investigado pequeñas arquitecturas de redes neuronales profundas, para ellos existe Yiny-YOLO.
- Los autores además entrenaron una versión rápida de YOLO que amplía límites para localizar los objetos en imágenes. La red neuronal de Tiny-YOLO es minimizada en capas convolucionales, ya que consta de solamente 9 capas y menos filtros. La única diferencia entre YOLO y Tiny- YOLO es el tamaño de la red, ya que los demás parámetros y procesos se conservan (Redmon, Joseph.; Santosh , Divvala.; Ross, Girshick.; Ali, 2016).
- Joseph propuso este modelo de Red Neuronal Convolucional unificado compuesto por 9 capas Convolucionales (CONV) y 3 capas FC, las primeras 6 capas CONV contienen maxpooling lo que convierte a esta red en muy robusta para combatir el ruido y la distorsión, para una clara visualización de la arquitectura de este modelo se muestra a continuación la Fig. 7 (Ma, Jing.;Chen, Li.;Gao, 2018).

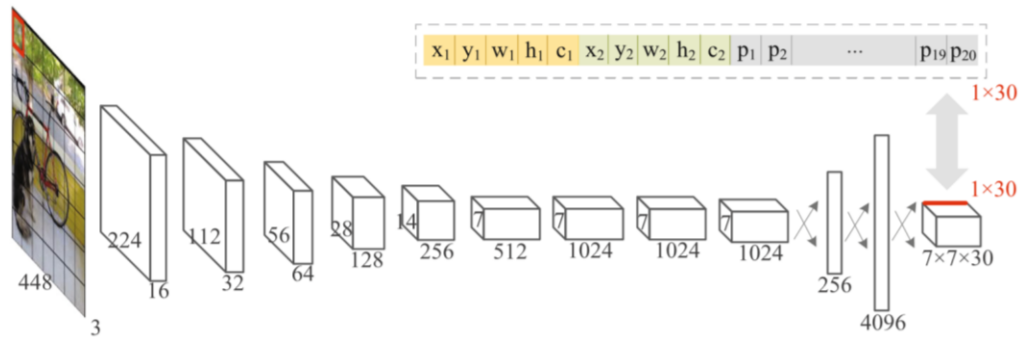


Fig 7. Red Neuronal Convolutcional Tiny-YOLO
Fuente: (Ma, Jing.;Chen, Li.;Gao, 2018)

- **SSD.** - Del inglés Single Shot Detector, genera mapa de características mediante una CNN, para luego predecir los objetos que posiblemente se encuentren en ciertas regiones. Su capa de red predice los objetos posibles en las imágenes a distintas escalas.
- **Mask-RCNN.** – En lugar de identificar regiones, segmenta los objetos en las imágenes que se están analizando todo esto gracias a una máscara conformada por una CNN usando mapa de caracteres en donde genera una matriz binaria indicando, 1 en donde el píxel corresponde a un objeto.

Luego de la explicación de los distintos métodos de localización de objetos, se puede observar en la Fig. 8 una visión general de lo mencionado.

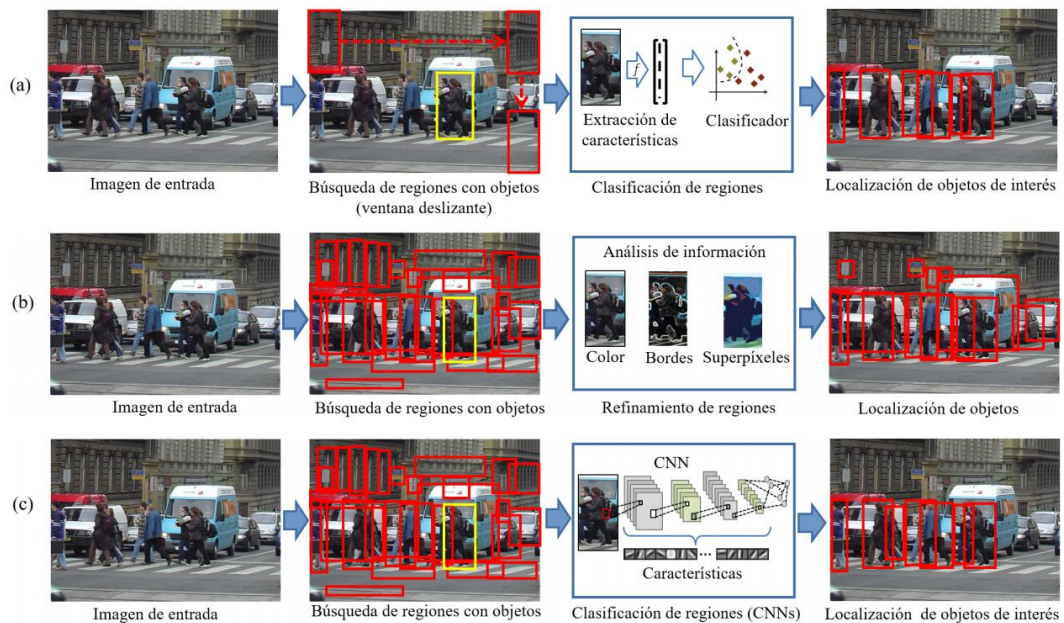


Fig 8. Localización de objetos en imágenes con los distintos métodos

1.3.4. Métodos de reconocimiento de texto

Gracias a la investigación de (Chaudhuri et al., 2017), ha permitido elaborar el siguiente contenido.

Optical character recognition (OCR) o reconocimiento óptico de caracteres, es un proceso que identifica patrones en imágenes digitales para reconocer caracteres alfanuméricos, posee la capacidad también de extraer caracteres de documentos escritos a mano que han sido escaneados, aunque el rendimiento dependa de la calidad del mencionado. A pesar de existir aplicaciones basadas en la tecnología OCR, aún no han logrado superar las funciones que el ser humano realiza (Chaudhuri et al., 2017).

a) Antecedentes

En 1900 se intentó desarrollar tecnología que permita mejorar la calidad de vida de los discapacitados visuales, a manos del científico ruso Tyuring, aunque en 1940 aparecieron los primeros reconocedores de caracteres gracias al avance de las computadoras digitales (Chaudhuri et al., 2017).

Estos reconocedores de caracteres se han fijado en textos impresos a máquina y en símbolos escrito a mano, que para ello utilizaban técnicas de procesamiento de imágenes de bajo nivel en la imagen binaria, posteriormente se implementa algoritmos exitosos pero restringidos para caracteres y números latinos (Chaudhuri et al., 2017).

En la década de los 1950, se comercializa la primera máquina de lectura OCR y se utilizó para transferir a un computador los informes de ventas realizados a máquina(Chaudhuri et al., 2017).

Años más tarde, a principios de los setenta aparecen máquinas que reconocen caracteres impresos a máquina y a mano, concluyendo en un reconocido sistema que fue el IBM 1287 en el año 1965 (Chaudhuri et al., 2017).

a) Técnicas de reconocimiento de caracteres

Para que la máquina reconozca patrones, primero se debe enseñar la clase de patrones con las que se trabajará ya que existen letras, números y algunos símbolos especiales. Para conseguir esto se debe indicar a la máquina el tipo caracteres.

Procedimiento

- Digitalizar un documento analógico con un escáner óptico.
- Buscar regiones que contienen texto para extraer los símbolos mediante un proceso de segmentación.
- Los símbolos que se extraen se pre procesan para eliminando el ruido y así extraer de características de forma más fácil.
- Se compara las características del símbolo extraído con símbolos previamente entrenados.
- Los caracteres obtenidos se usan para contextualizar y formar palabras y números.

b) Tipos de Aplicaciones

Las áreas de aplicación en donde el OCR ha sido exitoso y se ha usado ampliamente se mencionan a continuación.

Ingreso de datos: Se usa para ingresar datos restringidos como de entidades bancarias, estos sistemas leen un conjunto limitado de caracteres impresos, especialmente números y pocos símbolos especiales, como por ejemplo números de cuenta, identificación del cliente, cantidades de dinero entre otros.

Lectura de entrada de texto: Leen páginas en la automatización de oficinas, las restricciones tienen que ver con la fuente y la calidad de impresión. Además, son más útiles con el ingreso de grandes volúmenes de texto.

Automatización de procesos: Su enfoque principal es controlar procesos como la lectura automática de direcciones para la clasificación del correo. Su velocidad de clasificación es de 30 letras por hora.

1.4. Teléfono inteligente

Según (Lozano, 2018), Un teléfono inteligente es un teléfono celular con pantalla táctil y un robusto sistema operativo con el que los usuarios pueden conectarse a internet, instalar aplicaciones y llevar a cabo muchas de las actividades que podrían realizar en una computadora.

(José Miguel Roca Chillida, 2018) afirma:

Un smartphone es un ordenador de bolsillo que ha ido acumulando funciones y posibilidades aparentemente más potentes y versátiles con los procesos e incluyen, como mínimo, un teclado (físico o virtual), una pantalla normalmente táctil, y un importante tamaño de memoria.

1.4.1. Características de un teléfono inteligente

- a) **Sistema Operativo:** El smartphone será controlado por un sistema operativo que permita ejecutar varios tipos de aplicaciones, los sistemas operativos existentes son: BlackBerry OS, WebOS, iOS y Android.
- b) **Aplicaciones:** Los smartphones nos brindan la facilidad de usar ciertas aplicaciones de una manera más adaptada hacia el dispositivo, en lugar de usar una aplicación web en un navegador y que además posea un diseño responsivo, se puede instalar aplicaciones fabricadas especialmente para que se ejecuten sobre un smartphone como, por ejemplo: Facebook, WhatsApp, Instagram, entre otros.
- c) **Navegación Web:** Mediante el uso de un navegador web como Chrome, Firefox, safari o entre otros se puede acceder a toda la información que posee la web, tal y como si el acceso fuese desde un computador con la diferencia de la calidad de Interfaz de usuario debido a las medidas más pequeñas que posee un smartphone.
- d) **Mensajería Multimedia:** Brindan un mejorado y atractivo servicio de envío de mensajes y videos de alta calidad.

1.4.2. Sistemas Operativos más usados

De acuerdo a las estadísticas presentadas en la plataforma web de (NetMarketShare, 2020), se puede observar en la Fig. 9 , los sistemas operativos más usados en el último año, tomado el rango desde el Mes de Junio de 2019 hasta el mes de Junio del año 2020.

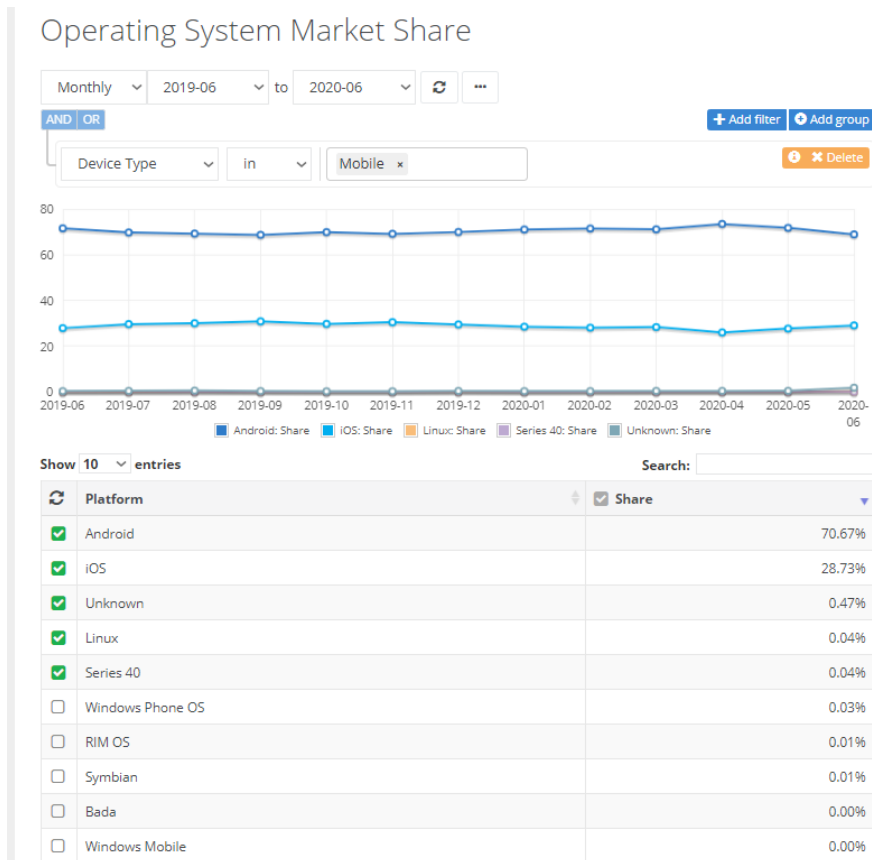


Fig 9. Estadística SO de smartphones más usados
Fuente: (NetMarketShare, 2020)

1.4.3. Sistema Operativo Android

Sistema Operativo basado en Linux para teléfonos móviles. Además, pueden usarlo en tablets, netbooks, reproductores de música e incluso PC's. Android permite programar en un entorno de trabajo (framework) de Java. Además, permite integrar fácilmente nuevas aplicaciones desarrolladas por programadores que puedan hacerlo, o incluso, modificar el propio sistema operativo, dado que Android es de código libre, el único requisito es tener conocimiento de programación en Java. (Báez et al., 2019).

- a) **Arquitectura:** Para observar fácilmente esta arquitectura, se puede observar la Fig. 10, ya se muestran las capas que componen el Sistema Operativo Android.

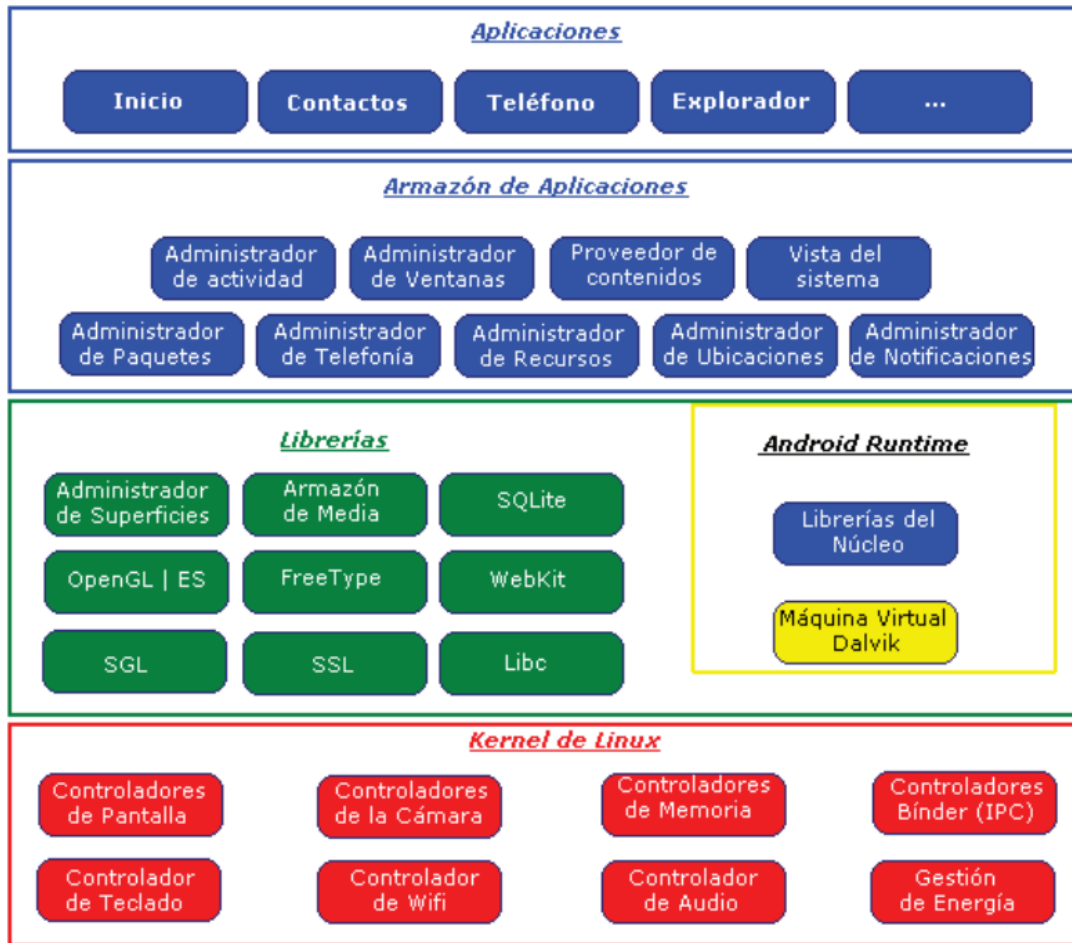


Fig 10. Sistemas de capas de Android
Fuente: (Báez et al., 2019)

Donde al profundizar más los conceptos se cita lo siguiente:

- **Kernel de Linux:** El núcleo del sistema es Linux y actúa como una capa de abstracción entre el hardware del dispositivo y las aplicaciones instaladas. Además, el sistema operativo de Google depende de Linux para otros servicios básicos como la seguridad, gestión de memoria, gestión de procesos, pila de red o controladores (Adeva, 2020).
- **Runtime:** El sistema operativo de Google para dispositivos móviles incluye un conjunto de bibliotecas que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje de programación Java. Cada aplicación Android corre su propio proceso con su instancia a la máquina virtual Dalvik (Adeva, 2020).

- **Librerías:** El sistema operativo Android incluye un conjunto de bibliotecas de C o C++ que son utilizadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de las aplicaciones de Android. Entre estas bibliotecas, caben destacara System C, bibliotecas de medios, de gráficos, 3 D o SQLite, entre otras (Adeva, 2020).
- **Marco del trabajo (Armazón) de aplicaciones:**

El entorno de Google permite que los desarrolladores tengan acceso a las mismas API del entorno de trabajo utilizadas por las aplicaciones base. Además, que la arquitectura de Android está diseñada para simplificar la reutilización de componentes (Adeva, 2020).

- **Aplicaciones:**

Android cuenta con ciertas aplicaciones base que permiten el uso de las funciones básicas de un dispositivo como son, correo electrónico, mensajes de texto SMS, calendario, mapas, navegador, contactos y otros. Aplicaciones desarrolladas en lenguaje Java (Adeva, 2020).

a) Componentes Básicos

Actividades: Por nombre en inglés Activity es el componente principal encargado de mostrar al usuario la interfaz gráfica (Báez et al., 2019).

- **Servicios:** o Services son tareas no visibles que se ejecutan siempre por en segundo plano, mientras alguna otra aplicación se ejecutara en primer plano (Báez et al., 2019).
- **Interfaz de Usuario:** Pantallas que el dispositivo nos muestra para realizar operaciones, además, es importante que el manejo sea intuitivo y sencillo (Báez et al., 2019).
- **Layouts en xml:** Es un recurso en donde defines la interfaz mediante código xml, tomando en cuenta ciertos parámetros en específicos (Báez et al., 2019).
- **Eventos de usuario: Son útiles para intervenir y** capturar en la interacción del usuario con una determinada aplicación. Para ellos se menciona ciertos eventos como: onClick, onKey, onTouch, entre otros (Báez et al., 2019).
- **Menús y barras de acciones:** Los menús permiten visualizar al usuario las acciones disponibles en una aplicación mediante una interfaz, esas acciones pueden afectar la aplicación presente o a otra funcionalidad del propio

dispositivo. Para crear uno, se recomienda definirlo en un archivo XML todo el contenido, este se encuentra en el directorio res/menú del proyecto, pues esta es la mejor opción (Báez et al., 2019).

- **Diálogos y notificaciones:** Los diálogos son avisos o comprobaciones que provienen de una determinada aplicación, en forma de ventana. La principal diferencia entre diálogos y notificaciones es que las notificaciones son solamente informativas, no se ejecutan en primer plano, por lo tanto, no muestra una interfaz para ello, ya que no requieren interacción directa con el usuario (Báez et al., 2019).
- **Estilos y temas:** El estilo permite personalizar el diseño de una vista o una ventana. Un tema es lo mismo que un estilo, pero aplicado a una actividad al completo, no sólo a una vista (Báez et al., 2019).

1.5. Librerías de inteligencia artificial compatibles con Android.

1.5.1. OpenCV4Android

En la página oficial de opencv se encuentra la información completa de esta librería, además se ha tomado textualmente indicaciones relevantes. (*OpenCV4Android SDK — OpenCV 2.4.5.0 documentation*, 2019)

El paquete OpenCV4Android SDK permite el desarrollo de aplicaciones de Android con el uso de la biblioteca OpenCV.

A partir de la versión 2.4.3 *OpenCV4Android SDK* utiliza la API de *OpenCV Manager* para la inicialización de la biblioteca. *OpenCV Manager* es una solución basada en el servicio de Android que proporciona los siguientes beneficios para los desarrolladores de aplicaciones OpenCV:

Tamaño apk compacto, ya que todas las aplicaciones usan los mismos binarios de Manager y no almacenan bibliotecas nativas dentro de sí mismas;

Las optimizaciones específicas de hardware se habilitan automáticamente en todas las plataformas compatibles;

Actualizaciones automáticas y correcciones de errores;

Fuente confiable de la biblioteca OpenCV. Todos los paquetes con OpenCV se publican en Google Play.

En la Fig. 11 se puede apreciar el tipo de teléfono que se necesita en el desarrollo del proyecto.



Fig 11. OpenCV sobre Android
Fuente: (Acodigo, 2018)

1.5.2. Tesseract OCR

Es un software libre, disponible en más de 100 idiomas. En 2011, Robert Theis lanzó una versión llamada "tess-two" de las herramientas de Android Tesseract OCR, y en la actualidad, se mantiene actualizada con nuevas funciones y fácil de usar.(craftcodecrew.com, 2018)

a) Tess-two

Según (github.rmtheis, 2019) Una bifurcación de Tesseract Tools para Android que agrega algunas funciones adicionales. Tesseract Tools para Android es un conjunto de API de Android y archivos de compilación para las bibliotecas de procesamiento de imágenes Tesseract OCR y Leptónica .

El módulo tess-two contiene herramientas para compilar las bibliotecas Tesseract y Leptónica para su uso en la plataforma Android. Proporciona una API Java para acceder a las API Tesseract y Leptónica compiladas de forma nativa(github.rmtheis, 2019).

1.6. Aplicaciones Móviles de Control Parental (Android)

Un número creciente de controles parentales y tecnologías de seguimiento y monitoreo de tecnología están en el mercado. Se lanzan principalmente como herramientas, aplicaciones o servicios en los que los padres pueden confiar en un intento de mantener a sus hijos seguros.

Dichos controles permiten a los padres, por ejemplo, evitar que los niños vean contenido inapropiado en línea, detectar el acoso cibernético en una etapa temprana y limitar el chat o las compras en la aplicación. Aunque los controles parentales a menudo se equiparan con los programas de filtro, el estado del arte en los controles parentales comercializados es más diverso (Zaman & Nouwen, 2016).

1.6.1. Universidad de Guayaquil del Ecuador

En la Universidad de Guayaquil del Ecuador, estudiantes de la Carrera de Ingeniería en Sistemas Computacionales, Edith Solórzano y Juan Bohórquez realizaron su trabajo de grado con un tema titulado “PROTOTIPO DE UN CONTROL PARENTAL PARA EL INTERNET EN EL HOGAR, OPERADO DESDE UN DISPOSITIVO ANDROID” (Solórzano, Edith . Bohórquez, 2016).

El proyecto mencionado trata del desarrollo de una aplicación móvil para controlar los accesos a internet que otros dispositivos realicen dentro del hogar.

Para el funcionamiento del proyecto se necesita un servidor con sistema operativo Linux Raspbian y en donde funcionarán los servicios SQUID3, Apache e Iptables, además utilizará Cron para administrar tareas.

Esta aplicación realizará el control sobre el contenido visualizado en aplicaciones de: Redes sociales, pornografía, salas de chat, etc. Y como resultados menciona la entrega de una solución práctica hacia los padres de familia para que puedan controlar el acceso a sitios peligrosos en el internet de su hogar combinando hardware y software que sean de fácil uso.

La aplicación de Control Parental en este proyecto funcionará siempre y cuando el dispositivo al que se pretende controlar se encuentre conectado en la misma red LAN, para esto el dispositivo móvil debe estar conectado vía Wifi.

Para el correcto funcionamiento del proyecto, el usuario debe poseer un dispositivo móvil que cuente con el sistema operativo Android desde la versión 5.0 en adelante, además de acceso a una red inalámbrica provista por el servidor CP-DROID anteriormente mencionado. El hardware utilizado que hace las veces de servidor debe ser conectado al router suministrado por el proveedor de servicios de Internet (ISP) y debe quedar encendido de forma permanente, además debe quedar conectado a una toma eléctrica regulada, y de preferencia contar con un UPS.

La aplicación orientada a dispositivos móviles con sistemas operativos Android, únicamente puede ser utilizada dentro del hogar.

Observación: Para este proyecto no menciona que la aplicación usará inteligencia artificial, por lo cual no tendría tanto impacto como el proyecto presentado por el autor de este documento, ya que además para el control de contenidos riesgosos debe estar conectado necesariamente el dispositivo a controlar en una misma red LAN en donde se encuentre el dispositivo servidor, a diferencia de la aplicación presentada, podrá funcionar en cualquier lugar que se encuentre, ya que para la localización de contenido riesgosos no necesita de internet, aunque para la entrega de reporte si lo necesite, pero si al proyecto se le siguiera realizando mejoras, podría cambiar esa funcionalidad.

1.6.2. Spyzie

Según (Hardzone.es, 2019), Hay ciertas aplicaciones que integran ajustes y funciones de control parental, pero cada una de ellas nos permitirá tener un control sobre esa aplicación. Sin embargo, hay otras herramientas que nos permiten tener un control total sobre un móvil. En este caso, Spyzie es sin duda una de las mejores apps de control parental para iOS y Android.

En el sitio oficial de la aplicación (spyzie.com, 2020) se puede observar la variedad de funcionalidades que brinda el este software, ya que se ha mencionado anteriormente ser uno de los mejores aplicativos de control parental. Entre las funcionalidades se encuentran: El rastreo de WhatsApp, Capturas de Pantalla, Registro de Llamadas, Mensajes, Exportación de Datos, Alertas, Ubicación, Registrador de Teclas, Messenger, Historial de Navegación, Línea de Tiempo, Fotos, Instagram, Snapchat, Contactos.

A continuación, se muestran las Fig. 12 y Fig. 13 que presenta el demo de la página oficial de esta aplicación.

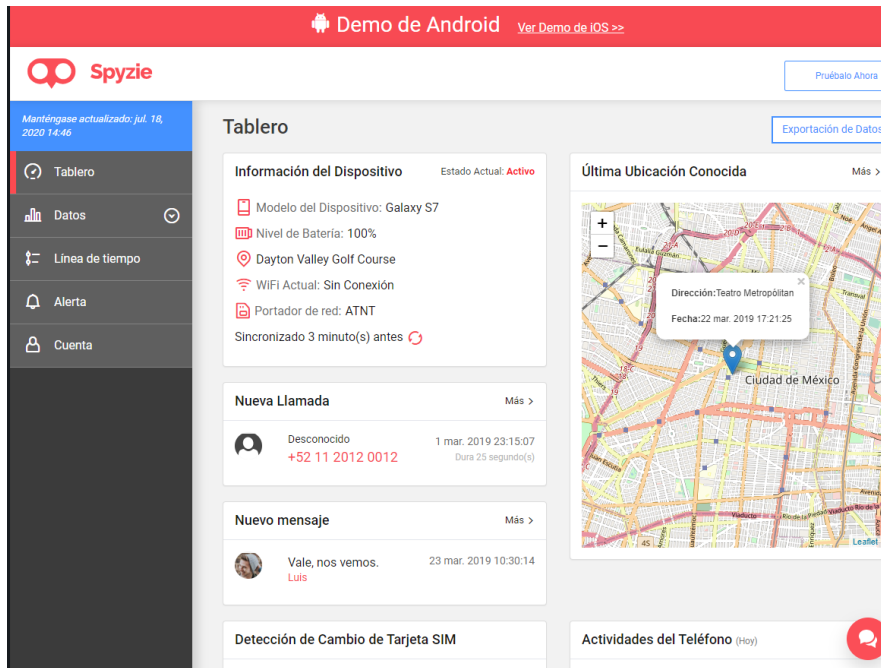


Fig 12. Demo de la aplicación Spyzie
Fuente: (spyzie.com, 2020)

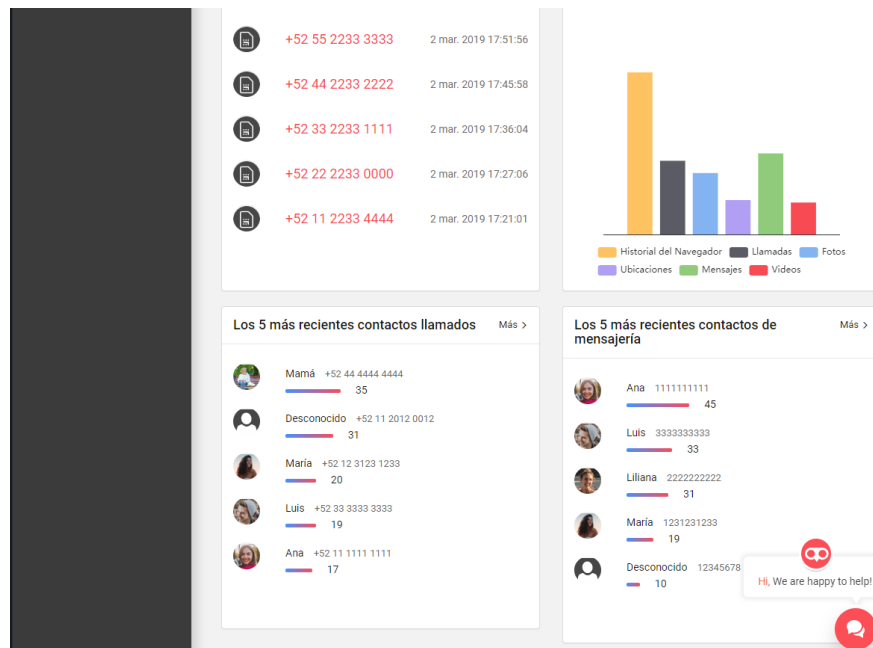


Fig 13. Demo de la aplicación Spyzie
Fuente: (spyzie.com, 2020)

La funcionalidad similar a la aplicación del presente proyecto, es “Capturas de Pantalla” y según su página oficial (spyzie.com, 2020) funciona de la siguiente forma: Si los miembros de tu familia están empleando una gran cantidad de tiempo en su móvil, o se complacen con alguna aplicación en particular durante demasiado tiempo, entonces puedes usar la función de obtención

de capturas de pantalla de Spyzie. Esto te permite tomar una captura de pantalla instantánea de la actividad que se está ejecutando en el dispositivo objetivo para así poder tomar medidas inmediatas que logren enmendar la situación.

Según la explicación que brinda la página oficial de Spyzie, la persona que realiza el control parental deberá capturar la pantalla en el momento que ésta lo decida para visualizar el contenido que se está transmitiendo en la aplicación

Por otro lado, la aplicación presentada en el presente proyecto se basa especialmente en la captura de pantalla, por lo tanto, esta función se ejecutará durante un tiempo sin la intervención de un usuario, por lo tanto, estará capturando y enviando reportes sobre riesgos encontrados de forma periódica que, para ello usará visión por computador sin necesidad de internet para el reconocimiento de objetos y para los reportes únicamente hará uso del internet.

1.7. Proyectos de Visión Artificial

1.7.1. Detección de imágenes con contenido explícito usando los modelos de color hsv y ycbcr

En el Instituto Politécnico Nacional se presentó un Trabajo de Grado de Maestría en seguridad y tecnologías de la información realizada por el Ingeniero Jorge Alberto marcial basilio en México el mes de diciembre del 2011, creando una aplicación de escritorio detectar imágenes de personas desnudas los modelos de color HSV y YCbCr con un conjunto total de 4800 imágenes. (ING. MARCIAL BASILIO & ALBERTO, 2015).

1.7.2. Diseño de un sistema de reconocimiento automático de matrículas de vehículos mediante una red neuronal convolucional.

En la Universidad Oberta de Catalunya se presentó un trabajo del área de Inteligencia Artificial correspondiente a la Titulación de del autor Francisco José Núñez Sánchez-Agustino de Máster Universitario en Ingeniería Informática en donde se presentó un modelo de Visión Artificial para el reconocimiento de imágenes que contengan placas de matriculación vehicular con el uso de 34.584 imágenes. (Nuñez, 2016).

1.7.3. Desarrollo de un clasificador visual de especies de aves mediante redes neuronales convolucionales.

Proyecto desarrollado por Antonio Miguel Pérez Segarra en la Universidad de Málaga en julio de 2019

A continuación se presenta el trabajo elaborado por (Perez, 2019).

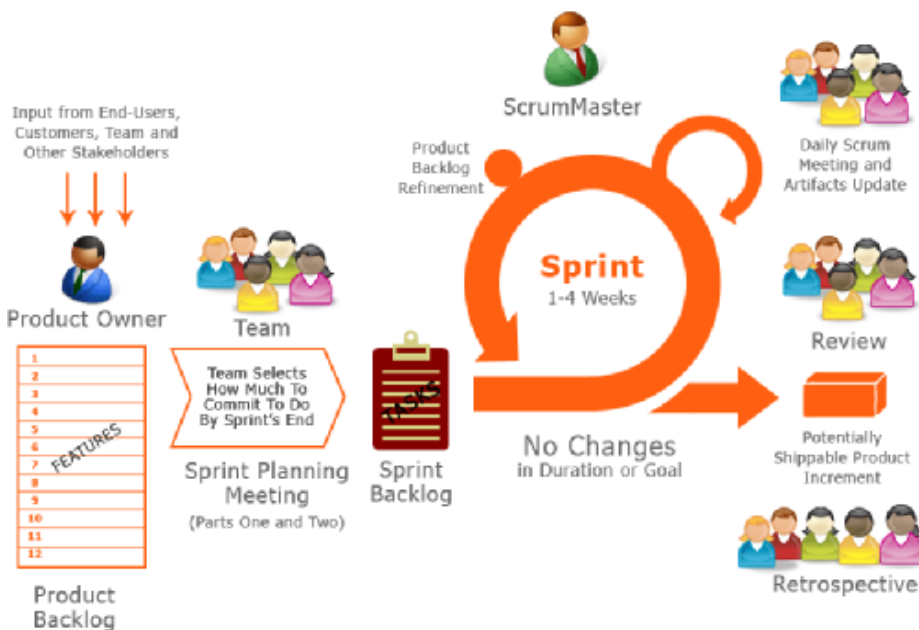
El método CLAHE consiste en sacar varias regiones de la imagen y normalizar los histogramas de esas regiones por separado, lo que hace que aumenten más de contraste las regiones más oscuras de la imagen y viceversa. En el entrenamiento del modelo computacional se fue evaluando mediante gráficas la precisión mediante los conjuntos de datos de entrenamiento y validación dando así una precisión media de 86.74 % (Perez, 2019).

1.8. Metodologías de Desarrollo de Software

Las metodologías de desarrollo de software son procedimientos que se deben seguir con cierta disciplina para lograr la culminación de los objetivos planteados en donde requieran habilidades y conocimientos específicos. Se utilizan en la ejecución de ciertas actividades con el fin de formalizarla y optimizarla. Además permite definir los pasos y la forma de cómo se debe proceder para conseguir finalizar una tarea (Maida & Pacienza, 2015).

1.8.1. Marco de trabajo Scrum

Mediante este marco de trabajo las personas pueden resolver problemas complejos para asegurar la entrega productos con alto valor productivo, además, es liviano, fácil de entender pero difícil de dominarlo (Schwaber & Sutherland, 2017). En la Fig. 14 muestra una visión general de cada uno de los elementos ejecutando sus correspondientes funciones.



a) El Equipo Scrum

Según (Schwaber & Sutherland, 2017) los equipos Scrum son multifuncionales y por ende tienen todas las capacidades necesarias para lograr un trabajo exitoso sin depender de personas externas al equipo, por ello se compone de los diversos roles.

- **Product Owner.** – El dueño del producto es la persona que se encarga de manejar la lista del producto o Product Backlog, puede definir los elementos de la lista del producto y ordenarlos de la manera que este lo considere para alcanzar los objetivos.
- **El Product Owner** deberá ser una sola persona, no un comité y además los demás integrantes de la organización deberán respetar sus decisiones.
- **Development Team (Equipo de desarrollo).** – Esta conformado por el grupo de profesionales que se encargaran de desarrollar y entregar el incremento o entregable, el cuál puede ponerse en producción al finalizar cada sprint. Cada miembro podrá tener habilidades distintas para aportar en el desarrollo, pero todo el grupo es el responsable de la construcción del producto.
- **El Scrum Master.** – Es responsable de apoyar y hacer cumplir lo que define la metodología Scrum, ellos, ayudan a los demás a entender aspectos como la teoría y la práctica de este marco de trabajo. También se dispone al servicio de todo el equipo, ellos verifican que el desarrollo se ejecute de mejor forma y ayuda a solventar inconvenientes en donde se involucren personas extras al equipo scrum.

b) Eventos de Scrum

Según (Schwaber & Sutherland, 2017) los eventos son bloques de tiempo que tienen una duración definida y no puede modificarse. Para que un evento finalice debe asegurarse que los objetivos se hayan cumplido.

Además, (Schwaber & Sutherland, 2017) define ciertos eventos que se citan a continuación.

- **El Sprint.** – Es un período limitado máximo a la duración de un mes, al terminar el sprint se debe entregar un incremento que se considere terminado, aunque

solamente es una parte proyecto final, se lo puede poner en producción. Durante el sprint no debe existir cambios que alteren el objetivo final. Al finalizar un sprint debe comenzar inmediatamente el siguiente sprint.

- Planificación de Sprint (Sprint Planning). – Se planifica el trabajo que se realizará durante el sprint, debe durar máximo 8 horas si el sprint durará un mes. El responsable que este evento suceda exitosamente y los asistentes comprendan el proyecto es el Scrum Master.
- Objetivo del Sprint (Sprint Goal). - Es una meta establecida para el sprint, explicando a los asistentes la razón de la construcción del sprint. Este se construye en la Planificación del Sprint.
- Scrum Diario (Daily Scrum). – Es una reunión diaria que tiene el equipo de desarrollo y dura 15 minutos, este se realiza para planificarse entre ellos las actividades que se realizarán en las próximas 24 horas. De esta manera también se realiza una evaluación diaria del trabajo, por ello se responde a las siguientes preguntas que deberán realizarse entre los asistentes. ¿Que hice ayer ?, ¿Qué haré hoy? ¿Hay algún impedimento para lograrlo?
- Revisión de Sprint (Sprint Review). – Al final del Sprint se realiza una revisión del incremento u objetivo del Sprint. Los interesados realizan comentarios sobre lo que se hizo durante el sprint, agregando además si hubo algún cambio en la Lista del Producto. Es una reunión informal que tiene una duración de cuatro horas con la que se pretende obtener retroalimentación de información y fomentar la colaboración.
- Retrospectiva de Sprint (Sprint Retrospective). – En esta instancia el Equipo Scrum deberá inspeccionarse para considerar mejoras que pueden aplicar en los siguientes Sprints, por tanto, se realizará al finalizar un Sprint y antes de la Planificación para el siguiente. Tiene una duración de tres horas para un Sprint de un mes y el Scrum Master deberá asegurarse que el evento se lleve exitosamente a cabo y todos hayan entendido correctamente lo sucedido.

c) Artefactos

Según (Viewnext.com, 2019), “Los artefactos son todos los elementos que te garantizan la transparencia y el registro de la información fundamental del proceso de Scrum”.

Según (Schwaber & Sutherland, 2017), los artefactos son los siguientes.

- Lista de Producto (Product Backlog). – Es una lista de características que el producto va a constituirse, y el Product Owner es el responsable de su contenido y además mantenerla ordenada. La lista del producto se va llenando conforme se desarrolle el proyecto, pues al inicio se consideran ciertos aspectos y no todos los que debería contener. Se deben ordenar por prioridades para que el equipo de desarrollo lo desarrolle en ese orden.
- Lista de Pendientes del Sprint (Sprint Backlog). – Representa los elementos que se han seleccionado desde la Lista de Producto para realizarlos en el sprint. El equipo de desarrollo modifica esta lista de pendientes indicando si una actividad está “por hacer”, “haciendo” o “echo”.
- Incremento. – Es el producto entregable por cada Sprint, y equivale al cumplimiento de todos los elementos de la Lista de Producto. Este sería entregado al final de cada Sprint, especificando que está terminado.

1.9. Lenguaje de Programación R

Este lenguaje se usa para el análisis estadístico ya que brinda muchas herramientas orientadas a las estadísticas, y de igual manera es muy útil para representar gráficos ya que permite visualizarlos en alta calidad solamente con utilizar funciones de graficación.

Es un lenguaje interpretado y el usuario debe acceder mediante una consola (Goette, 2014).

Según (Unir, 2019) entre las características del lenguaje de programación R que se destacan son:

- Un efectivo manejo y almacenamiento de datos.
- Cuenta con operadores necesarios para cálculos con matrices.
- Fuerte en herramientas para el análisis de datos.
- Permite visualizar los datos mediante gráficas.
- Este lenguaje es bien dotado ya que incluye saltos condicionales, bucles, funciones recursivas, utilidades para la entrada y salida de datos, entre otros.
- Proporciona documentación en formato físico y digital.

En la Fig. 15 se aprecia la interfaz de RStudio, para un uso más amigable con el Lenguaje R.

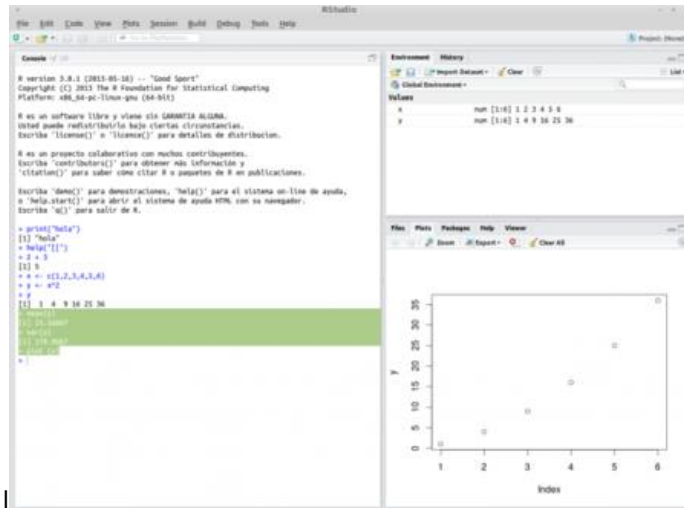


Fig 15. Lenguaje de programación R para estadísticas
Fuente: Propia

CAPÍTULO 2

Desarrollo

El desarrollo del proyecto consta de la construcción de una aplicación móvil Android, por tanto, se usa Android Studio como IDE de desarrollo, que además utiliza técnicas de visión artificial para la detección de contenido de riesgo hacia niños y adolescentes, mediante el análisis de capturas de pantalla del dispositivo móvil. Este proyecto adopta el Marco de Trabajo SCRUM, de las metodologías ágiles de desarrollo de software, en la Fig 16 se refleja el flujo de trabajo.

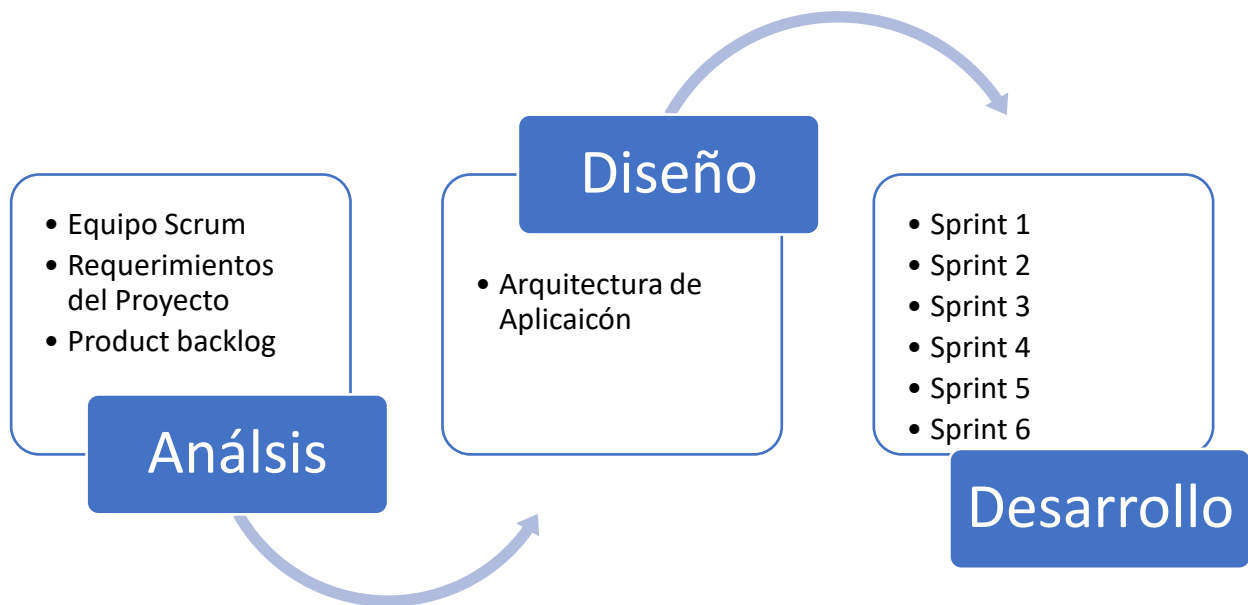


Fig 16. Ciclo desarrollo del proyecto
Fuente: Propia

2.1. Análisis

2.1.1. Equipo Scrum

Para este proyecto se han designado varios roles conforme lo indica el Marco de Trabajo Scrum, a continuación, en la TABLA 4 se muestra la estructura del Equipo Scrum.

TABLA 4. Estructura Equipo Scrum

Miembro	Descripción	Rol
---------	-------------	-----

MSc. Daisy Imbaquingo	Director del presente Trabajo de Grado y Docente de la Carrera de Ingeniería en Sistemas Computacionales de la Universidad Técnica del Norte.	Propietario del Producto (Product Owner).
MSc. Daisy Imbaquingo	Director del presente Trabajo de Grado y Docente de la Carrera de Ingeniería en Sistemas Computacionales de la Universidad Técnica del Norte.	Jefe del Proyecto (Scrum Master).
Sr. Roger Alexander Vaca	Estudiante de la Carrera de Ingeniería en Sistemas Computacionales de la Universidad Técnica del Norte. (Tesista)	Equipo de Desarrollo (Development Team)

2.1.2. Requerimientos del Proyecto

Los requerimientos que debe conformar el proyecto fueron levantados por el propietario del producto (MSc. Daisy Imbaquingo) y se recurre al uso de Historias de usuario mostrados a continuación.

TABLA 5. Historia de Usuario Nro. 1

Historia de Usuario	
Número: 1	Usuario: Cliente
Nombre: Arquitectura tecnológica	
Prioridad: Alta	Riesgo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roger Vaca	
Descripción: Definir las librerías de visión artificial que serán instaladas en Android, además de la técnica que será utilizada para localización de objetos en imágenes y todas las herramientas tecnológicas que permitan llevar a cabo correctamente el proyecto.	
Validación:	

TABLA 6. Historia de Usuario Nro. 2

Historia de Usuario	
Número: 2	Usuario: Cliente
Nombre: Configuración de librerías en Android Studio	
Prioridad: Alta	Riesgo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roger Vaca	
Descripción: Configurar las librerías necesarias que permitan hacer uso de OCR y Tiny-YOLO como técnica de visión artificial en la aplicación móvil Android que será construida.	
Validación:	
<ul style="list-style-type: none"> • Deberá funcionar correctamente la técnica de visión artificial y OCR en un dispositivo Android 	

TABLA 7. Historia de Usuario Nro. 3

Historia de Usuario	
Número: 3	Usuario: Cliente
Nombre: Creación de Dataset de contenido riesgoso	
Prioridad: Alta	Riesgo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roger Vaca	
Descripción: Almacenar imágenes pornográficas y sangrientas.	
Validación:	

TABLA 8. Historia de Usuario Nro. 4

Historia de Usuario	
Número: 4	Usuario: Cliente
Nombre: Preparación de dataset	
Prioridad: Alta	Riesgo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roger Vaca	
Descripción: Usar las imágenes de la Historia de Usuario Nro 3, para crear bounding boxes sobre los objetos que caracterizan a la imagen como pornográfica o sangrienta.	

Validación:

- Por cada imagen se debe generar un archivo de texto con el mismo nombre y su contenido deben ser coordenadas que localicen al objeto dentro de la imagen junto a un número que representa la clase para cada objeto, para esto, seguir la documentación de la metodología de visión artificial que será usada.
-

TABLA 9. Historia de Usuario Nro. 5

Historia de Usuario	
Número: 5	Usuario: Cliente
Nombre: Entrenar modelo de visión artificial	
Prioridad: Alta	Riesgo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roger Vaca	
Descripción: Entrenar un modelo de visión artificial que permita detectar objetos, para esto se usará el dataset de la Historia Usuario Nro 3, el archivo de etiquetas de la Historia de Usuario Nro 4 y la técnica de Tiny-YOLO.	
Validación:	
<ul style="list-style-type: none">• Para la preparación del modelo se deberá seguir la documentación de la red convolucional que se va a usar.	

TABLA 10. Historia de Usuario Nro. 6

Historia de Usuario	
Número: 6	Usuario: Cliente
Nombre: Capturas de pantalla en Android	
Prioridad: Alta	Riesgo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roger Vaca	
Descripción: Crear funcionamiento que permita tomar capturas de pantalla automáticamente mientras el dispositivo Android funciona normalmente con las demás aplicaciones.	
Validación:	
<ul style="list-style-type: none">• Las capturas de pantallas deben almacenarse en el dispositivo móvil.• La aplicación solicitará permiso una sola vez al Sistema Operativo Android para iniciar con las capturas de pantalla.	

TABLA 11. Historia de Usuario Nro. 7

Historia de Usuario	
Número: 7	Usuario: Cliente
Nombre: Detector de imágenes de riesgo en Android	
Prioridad: Alta	Riesgo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roger Vaca	
Descripción:	
<p>Haciendo uso del modelo entrenado en la Historia de Usuario Nro 5, se procede a crear la aplicación móvil utilizando las librerías de visión artificial para construir un detector de imágenes de riesgo a partir de las capturas de pantalla de la Historia de Usuario Nro 6.</p>	
Validación:	
<ul style="list-style-type: none"> • La detección debe realizarse en el mismo dispositivo móvil. 	

TABLA 12. Historia de Usuario Nro. 8

Historia de Usuario	
Número: 8	Usuario: Cliente
Nombre: OCR en Android	
Prioridad: Alta	Riesgo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roger Vaca	
Descripción:	
<p>Haciendo uso de las capturas de pantalla de la Historia de Usuario Nro 6, se agrega la funcionalidad de OCR a la aplicación móvil, para analizar texto considerado de riesgo para niños y adolescentes.</p>	
Validación:	
<ul style="list-style-type: none"> • El texto analizado deberá estar en español. 	

TABLA 13. Historia de Usuario Nro. 9

Historia de Usuario	
Número: 9	Usuario: Cliente
Nombre: Envío de correos electrónicos desde Android	
Prioridad: Alta	Riesgo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roger Vaca	
Descripción: La aplicación deberá enviar correos electrónicos, notificando a un receptor que se encontró cierto contenido de riesgo en el dispositivo móvil presente.	
Validación:	
<ul style="list-style-type: none"> • El envío debe ser automático • La configuración del correo electrónico puede realizarse de forma estática en el código de la aplicación. • En el contenido del mensaje debe incluir la evidencia del contenido riesgoso detectado. 	

2.1.3. Product backlog

En la TABLA 14 se detalla el Product Backlog conforme el Product Owner brinda prioridad a cada una de las historias de usuario.

TABLA 14. Product Backlog

PRODUCT BACKLOG			
ORDEN	ID	NOMBRE HISTORIA	ESTIMACIÓN
1	HU-01	Arquitectura tecnológica	25
2	HU-03	Creación de Dataset de contenido riesgoso	40
3	HU-04	Preparación de Dataset	45
4	HU-05	Entrenar modelo de visión artificial	105
5	HU-02	Configuración de librerías en Android Studio	2
6	HU-06	Capturas de pantalla en Android	38
7	HU-07	Detector de imágenes de riesgo en Android	20
8	HU-08	OCR en Android	25

9	HU-09	Envío de correos electrónicos desde Android	10
---	-------	---	----

2.2. Diseño

Siguiendo el Marco de Trabajo Scrum, para esta fase se consideró desarrollarse en el periodo de tiempo correspondiente al sprint 0, y los detalles se muestran en la TABLA 15.

TABLA 15. Sprint 0

Historia de Usuario	Fase de desarrollo	Tarea	Tiempo estimado (horas)
HU-01	Análisis	Investigación sobre herramientas tecnológicas	20
HU-01	Análisis	Herramientas tecnológicas que se usarán en el proyecto	2
HU-01	Diseño	Arquitectura de la aplicación móvil	3
TOTAL			25

2.2.1. Sprint 0

a) Planificación de Sprint

Fecha: 01/04/2020

Objetivo: Establecer la arquitectura y herramientas tecnológicas para el desarrollo del proyecto.

Miembros: Scrum Master, Product Owner, Development Team.

TABLA 16. Planificación Sprint 0

Sprint	Fecha inicio	Fecha fin	Duración
Sprint 0	01/04/2020	15/04/2020	25

b) Revisión de Sprint

El sprint ha terminado exitosamente, se ha cumplido lo planificado en los tiempos establecidos.

TABLA 17. Revisión Sprint 0



Historia de Usuario	Tarea	Tiempo estimado (horas)	Tiempo utilizado (horas)	Estado
HU-01	Investigación sobre herramientas tecnológicas	20	25	Terminado
HU-01	Herramientas tecnológicas que se usarán en el proyecto	2	2	Terminado
HU-01	Arquitectura de la aplicación móvil	3	3	Terminado
TOTAL		25	30	

Terminado el Sprint 0 se muestra el incremento del producto

- Herramientas tecnológicas

Después de una investigación y análisis profundo se define y detalla las herramientas tecnológicas además de la técnica de visión artificial que se usará para la ejecución del proyecto en la TABLA 18.

TABLA 18. Herramientas tecnológicas para el proyecto

TIPO	IMAGEN	DESCRIPCIÓN
IDE DE DESARROLLO		Android Studio 3.6.2 es el IDE de desarrollo de aplicaciones Android seleccionado para la construcción del dispositivo móvil.
LIBRERÍA DE VISIÓN ARTIFICIAL		OPEN CV 3.4.5 es la librería que procese las imágenes y las clasifique.

LIBRERÍA ANDROID	tess-two	tess-two 6.0.0 es una librería para reconocimiento de texto a partir de imágenes digitales.
NOTEBOOK	colab	GOOGLE COLAB es un notebook online que se usará para entrenar un modelo de visión artificial

- Arquitectura de la aplicación móvil

En la Fig. 17 se visualiza la arquitectura de la aplicación móvil en donde se obtendrán imágenes a partir de capturas de pantallas automáticas para posteriormente usar librerías que permitan detectar contenido considerado de riesgo para niños y adolescentes, enviando así notificaciones mediante correo electrónico a un destinatario determinado en caso de existir una detección de este tipo.

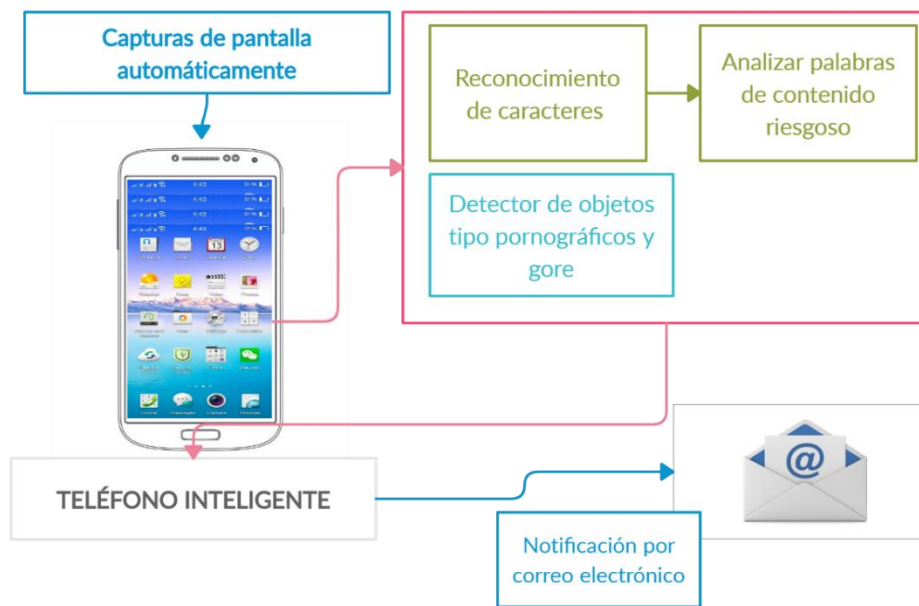


Fig 17. Arquitectura de la aplicación móvil
Fuente: Propia

c) Retrospectiva del Sprint

TABLA 19. Retrospectiva Sprint 0

RETROSPECTIVA

Fecha: 30/04/2020

Asistentes a reunión: Scrum Master, Product Owner y Team development

¿Que salió bien?	Se consiguió el incremento.
¿Que no salió bien?	Toma tiempo en buscar imágenes de buena calidad y especialmente sangrientas.
¿Qué se debe mejorar?	Conocer de mejor forma la calidad de las imágenes para visión artificial.

2.3. Desarrollo

Para el desarrollo de este proyecto se planificaron varios sprints en los cuales al finalizar cada uno de estos, se presentó un incremento, y por tanto fueron distribuido de la siguiente manera, mirar la TABLA 20.

TABLA 20. Planificación general de Sprints

Sprint	Fecha inicio	Fecha fin	Duración
Sprint 1	16/04/2020	30/04/2020	40
Sprint 2	01/05/2020	01/06/2020	150
Sprint 3	02/06/2020	22/06/2020	40
Sprint 4	23/06/2020	30/06/2020	20
Sprint 5	01/07/2020	10/07/2020	25
Sprint 6	11/07/2020	13/07/2020	15

2.3.1. Sprint 1

a) Planificación de Sprint

Fecha de reunión: 16/04/2020

Asistentes a la reunión: Scrum Master, Product Owner, Development Team.

Objetivo: Establecer un dataset que contenga imágenes pornográficas y sangrientas.

TABLA 21. Planificación Sprint 1

Historia de Usuario	Fase de desarrollo	Tarea	Tiempo estimado (horas)
HU-03	Desarrollo	Descargar imágenes pornográficas	20
HU-03	Desarrollo	Descargar imágenes sangrientas	20
TOTAL			40

b) Revisión de Sprint

El sprint 1 ha terminado exitosamente, obteniendo como resultado lo planificado previamente.

TABLA 22 Revisión Sprint 1

Historia de Usuario	Tarea	Tiempo estimado (horas)	Tiempo utilizado (horas)	Estado
HU-03	Descargar imágenes pornográficas	20	20	Terminado
HU-03	Descargar imágenes sangrientas	20	20	Terminado
TOTAL		40	40	

Incremento entregable del producto

- Dataset de imágenes riesgosas

Imágenes pornográficas

Para conseguir este tipo de imágenes se usaron varios sitios web que contiene apropiadamente contenido pornográfico, empleando así una metodología documental de investigación. Por esto se muestra en la TABLA 23 la lista de sitios web que brindan este tipo de contenido.

TABLA 23. Fuentes de contenido pornográfico

Fuente	Descripción
https://www.bestgore.com	Sitio web que proporciona imágenes sangrientas
https://www.youtube.com/	Sitio web que proporciona videos de diferente contenido.

Este dataset se organizó de la siguiente manera, ver TABLA 24

TABLA 24. Organización dataset imágenes pornográficas

Tipo objetos	Cantidad
Ano	2214
Pene	7994
Senos	3478
Trasero	7436
Vagina	1842
Testículos	1912
Variado	6448
TOTAL	31.364

Imágenes sangrientas

De igual para la adquisición de imágenes sangrientas se utilizó una metodología documental, buscando así en sitios web.

TABLA 25. Fuentes de contenido sangriento

Fuente	Descripción
http://pornomass.com/es	Sitio web que proporciona imágenes pornográficas
https://www.xvideos.com/	Sitio web que proporciona videos pornográficos
https://www xnxx.com/	Sitio web que proporciona videos pornográficos

El dataset correspondiente a imágenes sangrientas se organizó de la siguiente manera, ver TABLA 26.

TABLA 26. Organización dataset de imágenes sangrientas

Tipo objetos	Cantidad
Mutilación	1552
Sangriento	3424
TOTAL	4976

Dataset Global

De esta manera se muestra un dataset con imágenes consideradas de riesgo para niños y adolescentes, ver total en la TABLA 27.

TABLA 27. Organización dataset de imágenes de riesgo

Tipo contenido	Cantidad
Pornográfico	31.364
Gore	4.976
TOTAL	36.340

c) Retrospectiva del Sprint

TABLA 28. Retrospectiva Sprint 1

RETROSPECTIVA	
Fecha: 30/04/2020	
Asistentes a reunión: Scrum Master, Product Owner y Team development	
¿Que salió bien?	Se consiguió el incremento.
¿Que no salió bien?	Toma tiempo en buscar imágenes de buena calidad y especialmente d tipo gore.
¿Qué se debe mejorar?	Conocer de mejor forma la calidad de las imágenes para visión artificial.

2.3.2. Sprint 2

a) Planificación de Sprint

Fecha de reunión: 01/05/2020

Asistentes a la reunión: Scrum Master, Product Owner, Development Team.

Objetivo: Crear y entrenar un modelo de visión artificial para detección de imágenes pornográficas y de tipo gore.

TABLA 29. Planificación Sprint 2

Historia de Usuario	Fase de desarrollo	Tarea	Tiempo estimado (horas)
HU-04	Análisis	Investigar creación de un modelo de visión artificial personalizado con Tiny-YOLO	10
HU-04	Desarrollo	Preparación de Dataset	35
HU-05	Desarrollo	Configurar darknet usando Tiny-Yolo como técnica de visión artificial para entrenamiento de modelo personalizado	5
HU-05	Desarrollo	Entrenar el modelo de visión artificial ejecutando darknet sobre google colab	100
TOTAL			150

b) Revisión de Sprint

Una vez terminado el sprint 2 se realiza la revisión de las actividades realizadas.

TABLA 30. Revisión Sprint 2

Historia de Usuario	Tarea	Tiempo estimado (horas)	Tiempo utilizado (horas)	Estado
HU-04	Investigar creación de un modelo de visión artificial personalizado con Tiny-YOLO	10	10	Terminado
HU-04	Preparación de Dataset	35	35	Terminado

HU-05	Configurar darknet usando Tiny-Yolo como técnica de visión artificial para entrenamiento de modelo personalizado	5	5	Terminado
HU-05	Entrenar el modelo de visión artificial ejecutando darknet sobre google colab	100	100	Terminado
TOTAL		150	150	

A continuación, se detalla el resultado de este sprint

- Modelo de visión artificial

Preparación de Dataset. – Una fase importante hacia la creación de un modelo personalizado de visión artificial es la preparación del Dataset que se crea previamente, por ello es importante mencionar.

Los datos requeridos para esto son: imágenes y etiquetas. Para esto se debe crear un archivo de tipo texto asociada a cada imagen y por ende deberá llevar el mismo nombre de la imagen para identificarlo.

En este archivo debe indicar mediante coordenadas, en qué parte de la imagen se encuentra el objeto de interés, además a esto se adjunta un número que representa a la clase que pertenece el objeto, siguiendo la siguiente sintaxis.

<object-class><x_center><y_center><width> <height>

Para realizar este paso de etiquetado puede ayudarse de diferentes herramientas, en este proyecto se usó el software labellmg.

Antes de empezar el etiquetado es recomendable realizar una lista asignando un número de clase para cada objeto tal y como muestra la Fig 18.

```
for_colab_final > obj.names
seno
vagina
pene
testiculos
ano
trasero
mutilacion
sangriento
```

Fig 18. Asignación de número de clase a cada objeto
Fuente: Propia

A continuación, se ilustran ejemplos de este importante paso.



Fig 19. Archivo de texto para cada imagen pornográfica
Fuente: Propia

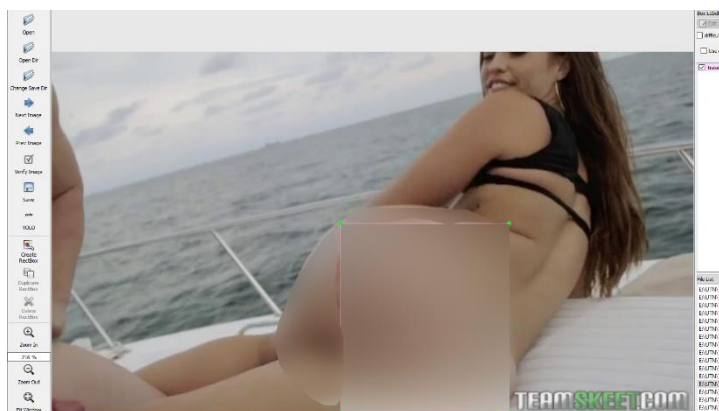


Fig 20. Etiquetado de imagen pornográfica en labelImg
Fuente: Propia

```
sex_pornes.xxx_a-kelsi-monroe-le-gustara-follar-en-el-mar_360p.mp4_20200606C
> UTN > Tesis > datasets > pornography > trasero > sex_pornes.xxx_a-kelsi-monroe-le-gu
1 5 0.581250 0.737500 0.262500 0.525000
2
```

Fig 21. Resultado de etiquetado de imagen pornográfica
Fuente: Propia



Wrong Turn



Wrong Turn

Fig 22. Archivo de texto para cada imagen sangrienta
Fuente: Propia

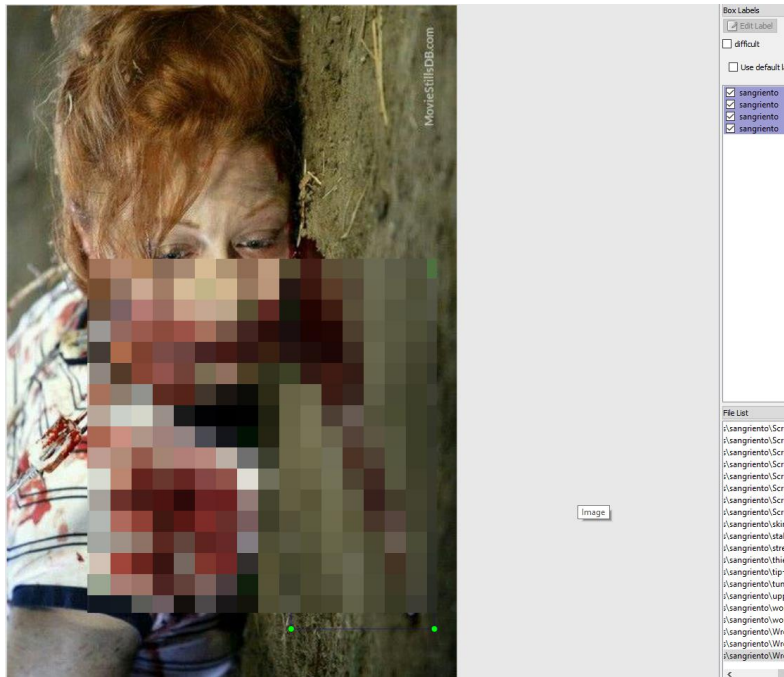


Fig 23. Etiquetado de imagen sangrienta en labelImg
Fuente: Propia

```
> UTN > Tesis > datasets > gores > sangriento > New folder > Wrong Turn.txt
1 16 0.793349 0.663492 0.318290 0.526984
2 16 0.378860 0.758730 0.306413 0.228571
3 16 0.334917 0.487302 0.199525 0.142857
4 16 0.591449 0.492063 0.076010 0.095238
```

Fig 24. Resultado de etiqueta de imagen sangrienta
Fuente: Propia

Preparación de modelo mediante darknet y Tiny-Yolo. – Darknet es un marco de red neuronal de código abierto que se puede encontrar en varios repositorios, para este proyecto se usó una versión situada en esta dirección <https://github.com/AlexeyAB/darknet.git>.

– Archivo Makefile: En la carpeta Darknet se encuentra el archivo “Makefile” que debe ser cambiado ciertos parámetros mediante un editor de texto.

$GPU = 1.$

$CUDNN = 1$

$OPENCV = 1$

Esto se debe a que para el presente proyecto se usó un computador con GPU para mejorar el proceso de entrenamiento de nuestro modelo, además, se trabajó sobre una red neuronal profunda usando el módulo que la librería Open CV dispone.

– Archivo de configuración (final-yolov3-tiny-obj.cfg): En el marco de red neuronal que se trabajó, se dispone de varios archivos de configuración con versiones distintas de los métodos de detección de objetos.

Crear una copia de “yolov3-tiny-obj”, debido a que se usó la técnica de Tiny-YOLO en su versión 3.

Entre los cambios que se realizaron para preparar el modelo es agregar un filtro en donde se aplica una fórmula de acuerdo con el número de nuestras clases.

$filters = (classes + 5) * 3$

Para el presente proyecto se usó 8 clases, por lo tanto, nuestra fórmula será la siguiente.

$$filters = (8 + 5) * 3 = 39$$

Estos valores fueron cambiados en específicas líneas del archivo de configuración de Tiny-Yolo. A continuación, se muestra cómo se distribuye esta configuración en la TABLA 31.

TABLA 31. Variables en archivo de configuración Tiny-YOLO

variable	concepto	valor	líneas
classes	Número de objetos que se pretende clasificar	8	135, 177
filters	Configuración necesaria para darknet.	39	127, 171

- Carpeta data: En esta carpeta se deberán agregar todas las imágenes con su respectivo archivo de etiqueta mencionado anteriormente.
- train.txt y val.txt: Son archivos que contienen datos de entrenamiento y validación, puesto que para lograr una evaluación objetiva se debe dividir el Dataset actual en dos grupos, de entrenamiento y validación.

La cantidad de imágenes para validación se extrae de acuerdo con la cantidad de imágenes en el Dataset, para pequeñas cantidades se usa a partir del 5% y para cantidades grandes hasta el 30%.

Para este proyecto se contó con 1.780 imágenes, por lo tanto, se usó el 10% de cada clase como datos de validación, ver TABLA 32.

TABLA 32. Organización dataset para entrenamiento

Imágenes pornográficas			
Tipo objetos	Cantidad	Entrenamiento	Validación
Ano	2214	1993	221
Pene	7994	7195	799

Senos	3478	3131	347
Trasero	7436	6693	743
Vagina	1842	1658	184
Testículos	1912	1721	191
Mezclados	6488	5840	648
	31.364	28231	3133
Imágenes gore			
Tipo objetos	Cantidad	Entrenamiento	Validación
Mutilación	1552	1397	155
Sangriento	3424	3082	342
	4.976	4479	497

Posterior a esto, se agrega las rutas de dichas imágenes en archivos de texto correspondiente, crear un archivo train.txt y val.txt

- Archivo obj.names: Este archivo debe contener los nombres de las clases en el orden que se usó para el etiquetado de imágenes, ver Fig 25.

```
for_colab_final > obj.names
seno
vagina
pene
testiculos
ano
trasero
mutilacion
sangriento
```

Fig 25. Distribución de clases para objetos de riesgo
Fuente: Propia

- Archivo obj.data: En este archivo se debe colocar datos como: el número de clases, ruta del archivo train.txt, ruta del archivo val.txt y ruta del archivo

obj.names y una ruta en donde se guardará automáticamente el modelo entrenado.

De acuerdo con este proyecto los datos para las variables mencionadas serían:

```
train = /content/darknet/data_for_colab_final/train.txt
```

```
valid = /content/darknet/data_for_colab_final/val.txt
```

```
names = /content/darknet/data_for_colab_final/obj.names
```

```
backup = /content/darknet/backup
```

En donde:

/content/: Corresponde a la ubicación de darknet sobre el notebook Google Colab.

/darknet/: Marco de red neuronal que se usó para el entrenamiento.

/data_for_colab_final/: Carpeta adicional que simplemente contiene los demás archivos necesarios citados en los literales anteriores.

/backup: Carpeta situada en Google drive que servirá para guardar los archivos del modelo entrenado.

Google COLAB

Darknet está escrito en el lenguaje de programación Python y además hace el uso funcionalidades que la librería de Open CV facilita. Estas funciones requieren de alto recurso computacional para completar el entrenamiento de un modelo de visión artificial. Debido a la complejidad de disponer siempre de un equipo computacionalmente fuerte se recurre a las facilidades que brinda Google con el uso de su notebook Google Colab.

Google Colab se traduce a un computador con altas características de hardware para ejecutar líneas de código muy avanzadas, ya que este tiene un sistema operativo Linux además dispone de las siguientes características:

- GPU: 1xTesla K80, cómputo 3.7, que tiene 2496 núcleos CUDA, 12GB GDDR5 VRAM (lo más importante, es compatible con CUDA)

- CPU: Procesadores Xeon hiperprocesados de un solo núcleo a 2.3 Ghz, es decir (1 núcleo, 2 hilos)
- RAM: ~ 12.6 GB disponibles
- Disco: ~ 33 GB disponibles

En la Fig 26 se observa las características del notebook que nos proporciona Google colab, que fácilmente se puede consultar desde el mismo a través del comando “!nvidia-smi”.

```
!nvidia-smi
```

```
Tue Jul 28 20:24:30 2020
```

```
-----
```

NVIDIA-SMI 450.51.05		Driver Version: 418.67		CUDA Version: 10.1			
GPU	Name	Persistence-M	Bus-Id	Disp.A	Memory-Usage	Volatile	Uncorr. ECC
Fan	Temp	Perf	Pwr:Usage/Cap			GPU-Util	Compute M.
							MIG M.
0	Tesla P100-PCIE...	Off	00000000:00:04.0	Off		0%	0
N/A	35C	P0	26W / 250W		0MiB / 16280MiB		Default
							ERR!

```
-----
```

```
Processes:
```

GPU	GI	CI	PID	Type	Process name	GPU Memory
ID	ID	ID				Usage
No running processes found						

```
-----
```

Fig 26. Características del Hardware de Notebook Google Colab
Fuente: Propia

Configurar entorno de Google COLAB

Se necesita tener disponible en el notebook Google Colab todos los archivos preparados anteriormente tales como:

- Darknet: Marco de red neuronal ya configurado.
- Archivo de configuración (final-yolov3-tiny-obj.cfg)
- Carpeta data que contiene las imágenes y etiquetas.
- Archivos train.txt, test.txt, obj.names, obj.data
- Archivo de pesos: Darknet necesita un archivo de pesos para ejecutar el entrenamiento. Inicialmente se usó el archivo “yolov3-tiny.conv.15” para este proyecto, y este es reemplazado por el último archivo de pesos que se genera en la carpeta “backup” anteriormente configurada como producto del entrenamiento. Este se presenta como “final-yolov3-tiny-obj_1000.weights”,

aunque el número varía de acuerdo con el número “1000” de iteraciones que se realicen.

Entrenar modelo

Para iniciar este proceso, se debe crear un nuevo notebook y además asegurarnos que éste utilice la GPU que nos proporciona Google Colab.

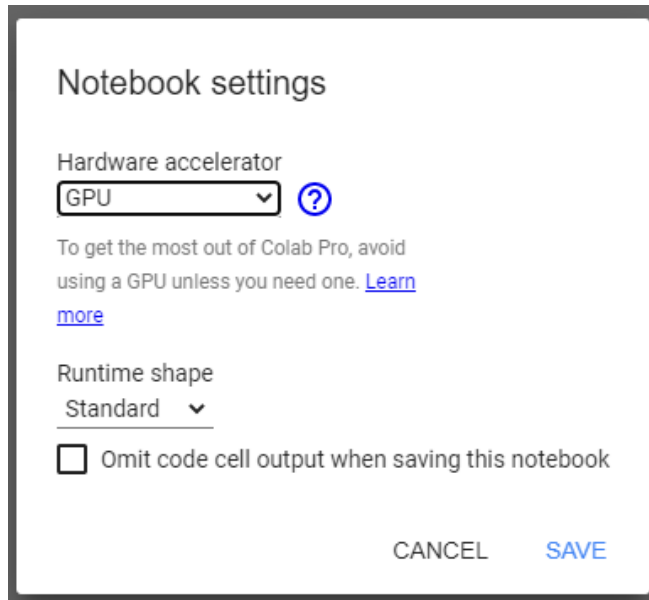


Fig 27. Configuración de Notebook Google Colab
Fuente: Propia

Ahora para ya iniciar el proceso de entrenamiento se necesita seguir varios pasos ordenadamente. Estos pasos fueron extraídos de un repositorio que está accesible a cualquier persona y a continuación se proporciona la dirección de este recurso y en la Fig 28 se observa que el entrenamiento ya está puesto en marcha.

Código de implementación:

https://github.com/quangnhhat185/Machine_learning_projects/blob/master/Gun_detection_Yolov3/Yolov3_Gun_Detection.ipynb

```

[ ] !./darknet
usage: ./darknet <function>

!./darknet detector train data_for_colab_final/obj.data cfg/final-yolov3-tiny-obj.

CUDA-version: 10010 (10010), GPU count: 1
OpenCV version: 3.2.0
final-yolov3-tiny-obj
0 : compute_capability = 600, cudnn_half = 0, GPU: Tesla P100-PCIE-16GB
net.optimized_memory = 0
mini_batch = 8, batch = 64, time_steps = 1, train = 1

```

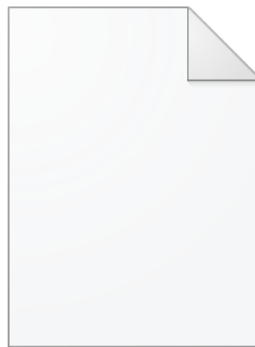
layer	filters	size/strd(dil)	input	output
0 conv	16	3 x 3/ 1	416 x 416 x 3 ->	416 x 416 x 16 0.150 BF
1 max		2x 2/ 2	416 x 416 x 16 ->	208 x 208 x 16 0.003 BF
2 conv	32	3 x 3/ 1	208 x 208 x 16 ->	208 x 208 x 32 0.399 BF
3 max		2x 2/ 2	208 x 208 x 32 ->	104 x 104 x 32 0.001 BF
4 conv	64	3 x 3/ 1	104 x 104 x 32 ->	104 x 104 x 64 0.399 BF
5 max		2x 2/ 2	104 x 104 x 64 ->	52 x 52 x 64 0.001 BF
6 conv	128	3 x 3/ 1	52 x 52 x 64 ->	52 x 52 x 128 0.399 BF
7 max		2x 2/ 2	52 x 52 x 128 ->	26 x 26 x 128 0.000 BF
8 conv	256	3 x 3/ 1	26 x 26 x 128 ->	26 x 26 x 256 0.399 BF
9 max		2x 2/ 2	26 x 26 x 256 ->	13 x 13 x 256 0.000 BF

Fig 28. Uso de Red Neuronal en Notebook Google Colab
Fuente: Propia

Archivo de modelo entrenado

Como resultado del entrenamiento se obtuvo un archivo de pesos con un tamaño de 33.1MB que posteriormente debe ser implementado en el dispositivo móvil Android mediante el uso de la librería OpenCV y sus funcionalidades de visión artificial.

El primer archivo resultante tiene como nombre “final-yolov3-tiny-obj_1000.weights” y de acuerdo con el tiempo de entrenamiento se obtendrán más archivos agregado el último número de la iteración. En la Fig 29 se puede observar el último archivo generado como producto del entrenamiento del modelo de visión artificial para este proyecto.



[final-yolov3-tiny-obj_314000.weights](#)

Fig 29. Resultado del entrenamiento del modelo de visión artificial
Fuente: Propia

c) Retrospectiva del Sprint

TABLA 33. Retrospectiva Sprint 2

RETROSPECTIVA	
Fecha: 01/06/2020	
Asistentes a reunión: Scrum Master, Product Owner y Team development	
¿Que salió bien?	Se consiguió el incremento satisfactoriamente.
¿Que no salió bien?	<ul style="list-style-type: none">- En un principio se tuvo limitaciones al uso de Google Colab, pero gracias a la comunicación entre el Equipo Scrum se resolvió.- Complicaciones con el uso de Google Colab ya que se necesita de conexión a internet constante- Toma de tiempo en buscar la documentación adecuada de darknet debido a la existencia de distintas versiones.
¿Qué se debe mejorar?	<ul style="list-style-type: none">- Tener licencias de paga para herramientas que limitan funcionalidades.

2.3.3. Sprint 3

a) Planificación de Sprint

Fecha de reunión: 02/06/2020

Asistentes a la reunión: Scrum Master, Product Owner, Development Team.

Objetivo: Crear un nuevo proyecto de Android que realice capturas de pantallas automáticas y además tenga configurado librerías de visión artificial.

TABLA 34. Planificación Sprint 3

Historia de Usuario	Fase de desarrollo	Tarea	Tiempo estimado (horas)
HU-02	Desarrollo	Creación de nuevo proyecto Android y configuración de Open CV como librería de visión artificial y tess-two para reconocimiento de caracteres	2
HU-06	Desarrollo	Crear funcionalidad en aplicación Android que permita realizar capturas de pantalla automáticamente en segundo plano.	38
TOTAL			40

b) Revisión de Sprint

Terminado el Sprint 3, se puede visualizar, la creación de una aplicación móvil que adquiere imágenes gracias a la toma de capturas de pantalla automáticas y que posteriormente serán analizadas mediante inteligencia artificial.

TABLA 35. Revisión Sprint 3

Historia de Usuario	Tarea	Tiempo estimado (horas)	Tiempo utilizado (horas)	Estado
HU-02	Creación de nuevo proyecto Android y configuración de Open CV como librería de visión artificial y tess-two para reconocimiento de caracteres	2	2	Terminado
HU-06	Crear funcionalidad en aplicación Android que permita realizar capturas de pantalla automáticamente en un segundo plano.	38	38	Terminado
TOTAL		40	40	

El incremento objetivo de este sprint se muestra a continuación luego de cumplir con las actividades planteadas.

- Creación de aplicación en Android Studio

Para el desarrollo del proyecto se usó Android Studio en su versión 3.6.2, que además la aplicación estará disponible para teléfonos inteligentes Android a partir de la versión 4.0.

Además, se agregó la librería OpenCV 3.4.5 usando un sdk compatible con Android Studio y la librería tess-two en su versión 6.0.0 que posteriormente será de mucha utilidad para el reconocimiento de caracteres.

- Capturas de pantalla en aplicación móvil Android

A continuación en la Fig 30 y Fig 31 se visualiza de manera clara el flujo de funcionamiento y de los procesos realizados para cumplir con lo planificado.



Fig 30. Ilustración Capturas de pantalla en Android
Fuente: Propia

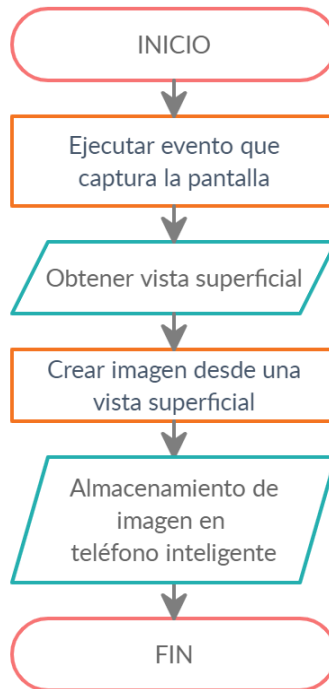


Fig 31. Flujograma de procesos para la captura de pantalla
Fuente: Propia

Ejecutar evento que capture la pantalla

- **MediaProjection:** Una funcionalidad que permite el sistema operativo Android para capturar el contenido de la pantalla y / o grabar el audio del sistema. Esto depende del tipo de MediaProjection que se defina usar.
- **MediaProjectionManager#createScreenCaptureIntent:** Mediante este método el sistema otorga la capacidad de capturar el contenido de la pantalla (screenshot).
- **startActivityForResult:** Método que da inicio de una actividad y fué usado para dar inicio al método de captura de pantalla createScreenCaptureIntent.
- **onActivityResult:** Método útil para obtener el resultado originado por startActivityForResult.

Obtener vista superficial

- **VirtualDisplay:** Permite capturar el contenido de la pantalla, además a este método se debe enviar argumentos como un nombre, ancho, alto, la densidad de la pantalla virtual, flags, la superficie en la que se mostrará (surface), un callback y handler opcional.

- SurfaceView: Esta clase permite crear una superficie o capa sobre la pantalla y servirá para que la aplicación dibuje el contenido capturado de la pantalla.
- ImageReader: Permite el acceso a los datos de imagen que se representan en un Surface, ya que los objetos de la imagen se representan en objetos de la clase Image.
- ImageReader.setOnImageAvailableListener: Es una interfaz que indica cuando una imagen nueva está disponible proveniente de una surface para lograr manipular sus datos y posteriormente poder almacenarlo en memoria.

Crear imagen desde una vista superficial

- Bitmap: Clase que permite crear mapa de bits, en donde podrán ser reflejadas las imágenes.
- Bitmap.createBitmap: Permite crear una imagen digital a partir de las características del SurfaceView que captura el contenido de la pantalla.
- Bitmap.copyPixelsFromBuffer: Copia los datos de los píxeles que se encuentran en el mapa de bits hacia una matriz de píxeles.

Almacenamiento de imagen en teléfono inteligente

- FileOutputStream: Crea un archivo en la dirección indicada.
- Bitmap.compress: Comprime el mapa de bits en el formato de la imagen que se le indique y además lo sobre escribe en un archivo creado con anterioridad para así escribir la imagen en el dispositivo.

c) Retrospectiva del Sprint

TABLA 36. Retrospectiva Sprint 3

RETROSPECTIVA	
Fecha: 22/06/2020	
Asistentes a reunión: Scrum Master, Product Owner y Team development	
¿Que salió bien?	Las capturas de pantalla automáticas funcionan correctamente en segundo plano sobre una aplicación móvil Android.
¿Que no salió bien?	No se pudo usar MediaProjection en un servicio background para capturar el contenido de la pantalla.

¿Qué se debe mejorar?	Mejorar la comunicación para optar por alternativas para resolver los problemas tecnológicos.
-----------------------	---

2.3.4. Sprint 4

a) Planificación de Sprint

Fecha de reunión: 23/06/2020

Asistentes a la reunión: Scrum Master, Product Owner, Development Team.

Objetivo: Desarrollar funcionalidad en la aplicación móvil que permita detectar si las capturas de pantalla realizadas anteriormente son de tipo pornográficas o gore.

TABLA 37. Planificación Sprint 4

Historia de Usuario	Fase de desarrollo	Tarea	Tiempo estimado (horas)
HU-07	Desarrollo	Programar el clasificador de imágenes usando Darknet-Yolo de OpenCV.	20
TOTAL			20

b) Revisión de Sprint

Al final del Sprint 4 se muestran los datos sobre cómo se fue realizando el desarrollo.

TABLA 38. Revisión Sprint 4

Historia de Usuario	Tarea	Tiempo estimado (horas)	Tiempo utilizado (horas)	Estado
HU-07	Programar el clasificador de imágenes usando Darknet-Yolo de OpenCV.	20	20	Terminado
TOTAL		20	20	

Siguiendo la planificación se ha logrado cumplir con el incremento.

- Clasificador de imágenes pornográficas y tipo gore.

En la Fig 32 se ilustra el funcionamiento general del clasificador.



Fig 32. Ilustración Clasificador de Imágenes
Fuente: Propia

De esta forma se pretende transmitir claramente un flujo general para el funcionamiento del clasificador de imágenes.

En la TABLA 39, se presentan las tecnologías que permitieron desarrollar el clasificador.

TABLA 39. Tecnologías para Clasificador de Imágenes.

Java	Lenguaje de programación para desarrollo de aplicaciones móviles en Android Studio.
OpenCV	Librería de código abierto y compatible con Android Studio que permite usar redes neuronales.
Darknet	Marco de trabajo para redes neuronales.

La librería de OpenCv ya ha sido implementada en el IDE de desarrollo Android Studio y ahora se desarrolla el clasificador de imágenes usando dicha librería y para importarla se usa `import org.opencv.*` para usar todas las funcionalidades de OpenCV, sino reemplaza a * por el método que se necesite y además importante inicializar esta librería usando el método `OpenCVLoader.initDebug()`.

En este apartado se presencia el uso de la visión artificial para cumplir el objetivo planteado, por tanto, se menciona a continuación las fases importantes que conforma un detector de imágenes en visión por computador.

Antes de iniciar el proceso de clasificación de imágenes es necesario cargar el archivo de configuración basado en Tiny-Yolo como técnica de visión artificial, además del archivo de pesos generados anteriormente en el entrenamiento del modelo personalizado realizado en Google Colab.

Adquisición de imagen

Se adquiere la imagen a través de la librería OpenCV usando el método Imgcodecs.imread al cual se envía un parámetro tipo String que especifica la ruta de la imagen que se pretende analizar.

Esta ruta corresponde a el almacenamiento interno del dispositivo móvil ya que en el sprint anterior se manifiesta haber almacenado la captura de pantalla en el lugar mencionado, y por ende se debe recuperar y pasar al método de clasificación.

- `Imgcodecs.imread`: es un método que proporciona OpenCV para leer imágenes y además permite agregarla en una matriz.

Preprocesamiento

La imagen adquirida no se puede usarla directamente con la red neuronal, para ello se debe convertir a blob y además redimensionarla.

La redimensión se cambia de acuerdo con los resultados que se obtiene, además es importante certificarse que ese valor sea múltiplo de 32.

Para realizar el preprocesamiento de la imagen se implementó:

```
blob = Dnn.blobFromImage (imagen, scalefactor = 1.0 , tamaño, media, swapRB  
= True )
```

En donde:

Dnn: Red Neuronal Profunda que facilita la librería OpenCV

blobFromImage: Método para crear un blob a partir de una imagen.

imagen: Es la imagen adquirida que servirá para crear un blob a partir de ella.

Scalefactor: Valor de la escala usada en las imágenes, por su defecto es 1.

Tamaño: Tamaño de espacio que la red neuronal espera para tratar la imagen.

Media: Técnica usada para combatir la iluminación en la imagen.

swapRB: Para confirmar que se está usando una imagen RGB, su valor debe ser true.

A esta función se ha remplazado con los valores del presente trabajo y se obtiene lo siguiente.

```
Mat imageBlob = Dnn.blobFromImage(imagen, 0.00392, new Size(640, 640),  
new Scalar(0, 0, 0), /*swapRB*/true, /*crop*/false);
```

Segmentación

La fase de segmentación lo realiza Tiny-YOLO, puesto que se confió en el método que este implementa para completar la detección de objetos.

Este algoritmo cuenta con varias capas profundas de la red convolucional, se enfocó especialmente en dos capas ya que son fuertes en detección con objetos pequeños y estas son: yolo_16 y yolo_23. Para trabajar simplemente con estas capas se implementó el siguiente código.

```
java.util.List<Mat> result = new java.util.ArrayList<Mat>(2);
```

```
List<String> outBlobNames = new java.util.ArrayList<>();  
outBlobNames.add(0, "yolo_16");  
outBlobNames.add(1, "yolo_23");
```

```
tinyYolo.forward(result, outBlobNames);
```

Clasificación

Para esta fase es importante contar con un valor de umbral que servirá para filtrar los resultados que más se acerquen a los esperados.

Se toman los resultados arrojados por la red neuronal y se procede a extraer los datos de cada imagen, específicamente las coordenadas de los objetos que se pretenden localizar si la imagen lo contiene.

A continuación, se cita el código explicado anteriormente

```
//Confidence threshold
float confThreshold = 0.5f;
```

```
//Coordinates data
List<Integer> clsIds = new ArrayList<>();
List<Float> confs = new ArrayList<>();
List<Rect> rects = new ArrayList<>();
```

En este bloque de código se analiza si el objeto localizado es aceptado y en caso de serlo, se obtienen las coordenadas del mismo sobre la imagen.

```
if (confidence > confThreshold) {
    int centerX = (int) (row.get(0, 0)[0] * imagen.cols());
    int centerY = (int) (row.get(0, 1)[0] * imagen.rows());
    int width = (int) (row.get(0, 2)[0] * imagen.cols());
    int height = (int) (row.get(0, 3)[0] * imagen.rows());

    int left = centerX - width / 2;
    int top = centerY - height / 2;
    System.out.println("classIdPoint: " + (int) classIdPoint.x);
    clsIds.add((int) classIdPoint.x);
    confs.add((float) confidence);

    rects.add(new Rect(left, top, width, height));
}
```

De esta forma se dibujan rectángulos sobre la imagen indicando la ubicación de los objetos detectados dentro de ella.

Interpretación

La fase de interpretación en visión por computador permite asignar un significado a los objetos localizados en la imagen. En el presente proyecto se pretende localizar imágenes pornográficas y de tipo sangrientas, por ello se presentará el código que permite agregar nombres a cada rectángulo posicionado en la imagen gracias a la fase de clasificación.

```
// labels names
List<String> labelNames = Arrays.asList("seno", "vagina", "pene",
    "testiculos", "ano", "trasero", "mutilacion", "sangriento", "decapitacion",
    "tortura", "ataque_objetos", "desollamiento");
int intConf = (int) (conf * 100);
Imgproc.putText(imagen, labelNames.get(idGuy) + " " + intConf + "%", box.tl(),
    Core.FONT_HERSHEY_SIMPLEX, 2, new Scalar(255, 255, 0), 2);
```



```
Imgproc.rectangle(imagen, box.tl(), box.br(), new Scalar(255, 0, 0), 2);
```

En este punto ya se han dibujado los rectángulos juntos con su nombre de objeto en la imagen. Ahora esa imagen se almacena en el dispositivo con el uso de este código.

```
//Asignar un nombre a la imagen imagen
```

```
String outImage = "/output-" + outIndex + ".png";
```

```
//Escribir la imagen en la memoria del dispositivo Android
```

```
Imgcodecs.imwrite(getPath(_OUTPUT_FOLDER) + outImage, imagen);
```

En la Fig 33 y Fig 34 se aprecia el resultado del clasificador descrito.



Fig 33. Imagen sangrienta detectada por el clasificador
Fuente: Propia

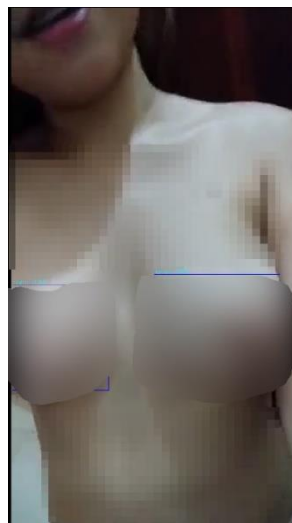


Fig 34. Imagen pornográfica detectada por el clasificador
Fuente: Propia

c) Retrospectiva del Sprint

TABLA 40. Retrospectiva Sprint 4

RETROSPECTIVA	
Fecha: 30/06/2020	
Asistentes a reunión: Scrum Master, Product Owner y Team development	
¿Que salió bien?	Se logró detectar contenido de riesgo con un clasificador basado en Tiny-YOLO.
¿Que no salió bien?	Tomó un poco más de tiempo la codificación.
¿Qué se debe mejorar?	Mejorar habilidad del uso de OpenCV.

2.3.5. Sprint 5

a) Planificación de Sprint

Fecha de reunión: 01/07/2020}

Asistentes a la reunión: Scrum Master, Product Owner, Development Team.

Objetivo: Desarrollar funcionalidad en la aplicación móvil que permita detectar si las capturas de pantalla realizadas anteriormente contienen palabras ofensivas.

TABLA 41. Planificación Sprint 5

Historia de Usuario	Fase de desarrollo	Tarea	Tiempo estimado (horas)
HU-08	Desarrollo	Crear un glosario de palabras que se caractericen como ofensivas	2
HU-08	Desarrollo	Obtener el texto en español que se encuentren en las capturas de pantalla mediante el uso de la librería tess-two (OCR)	20
HU-08	Desarrollo	Detectar si existen palabras ofensivas en el texto extraído de las capturas de pantalla, usando el glosario creado anteriormente	3
TOTAL			25

b) Revisión de Sprint

De igual forma al finalizar el Sprint 5 se muestran la información de cómo se fue realizando el desarrollo.

TABLA 42. Revisión Sprint 5

Historia de Usuario	Tarea	Tiempo estimado (horas)	Tiempo utilizado (horas)	Estado
HU-08	Crear un glosario de palabras que se caractericen como ofensivas	2	2	Terminado
HU-08	Obtener el texto en español que se encuentren en las capturas de pantalla mediante el uso de la librería tess-two (OCR)	20	20	Terminado
HU-08	Detectar si existen palabras ofensivas en el texto extraído de las capturas de pantalla, usando el glosario creado anteriormente	3	3	Terminado
TOTAL		25	25	

El desarrollo de este sprint ayudó a complementar las funcionalidades de la aplicación móvil, y cumpliendo con el objetivo se logra detectar palabras ofensivas de las capturas de pantalla realizadas automáticamente en el teléfono inteligente Android.

- Reconocimiento de palabras ofensivas en capturas de pantalla.

La Fig 35 posee un esquema del funcionamiento de la funcionalidad.



Fig 35. Ilustración de reconocimiento de palabras ofensivas mediante OCR.

Fuente: Propia

Usando el mismo flujo de entrada de imágenes del sprint anterior, se desarrolló un método que permite extraer caracteres a partir de una imagen, para con ello analizar las palabras capturadas y posteriormente compararlas con un glosario de términos considerados ofensivos, para con esto generar una alerta por el contenido encontrado. En la TABLA 43 se citan las tecnologías usadas en este apartado.

TABLA 43. Tecnologías para reconocimiento de palabras ofensivas mediante OCR.

Java	Lenguaje de programación para desarrollo de aplicaciones móviles en Android Studio.
Tess-two	Librería derivada de Tesseract OCR, útil para extracción de caracteres.

Tess-two es la librería que permite el reconocimiento de caracteres y lo puede hacer en diferentes idiomas. Mediante programación es importante agregar un archivo de configuración llamado "spa.traineddata" en donde "spa" es la abreviatura al idioma español y la extensión "traineddata" es propia de la librería para obtener los resultados esperados.

Por lo tanto, en la Fig 36 se muestra el flujo del funcionamiento del método para reconocimiento de palabras ofensivas.

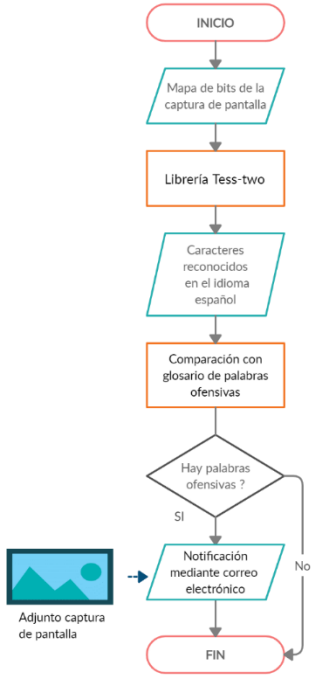


Fig 36. Flujograma reconocimiento de palabras ofensivas mediante OCR
Fuente: Propia

A continuación, se muestra el código fuente usado para el desarrollo de este método, de acuerdo con el flujograma presentado anteriormente.

Entrada de imagen - Mapa de bits de la captura de pantalla

La captura de pantalla actualmente se encontraría almacenada en el almacenamiento interno del teléfono inteligente, por tanto en este paso se procedió a obtener esa imagen para posteriormente enviárselo al método que implementa Tess-two.

```
Bitmap bitmap = BitmapFactory.decodeFile(filePath);  
foundText = this.extractText(bitmap);
```

En donde:

- filePath: Ruta en la memoria interna del teléfono inteligente en donde se encuentra la captura de pantalla.
- BitmapFactory.decodeFile: Método que permite extraer la captura de pantalla y guardarla en un objeto de la clase Bitmap.
- bitmap: Variable que contiene el Mapa de bits que representa la captura de pantalla.
- extractText: Método en permite usar la librería Tess-two.
- foundText: Variable que contienen todos los caracteres extraídos de la captura de pantalla mediante el uso de Tess-two.

Proceso - Librería Tess-two

Para usar esta librería se implementó el siguiente código y a continuación su respectiva explicación.

```
public String extractText(Bitmap bitmap){  
    String extractedText = "";  
    try {  
        String dataPath = getTessDataParentDirectory();  
        TessBaseAPI tessBaseApi = new TessBaseAPI();  
        tessBaseApi.init(dataPath, "spa");  
        tessBaseApi.setImage(bitmap);  
        extractedText = tessBaseApi.getUTF8Text();  
        tessBaseApi.end();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

```
    return extractedText;
}
```

- bitmap: Parámetro que recibe el presente método y contiene el Mapa de bits que representa la captura de pantalla.
- extractedText: Variable que almacenará el texto extraído de la captura de pantalla mediante Tess-two.
- getTessDataParentDirectory: Método creado para obtener la ruta absoluta de los archivos de la aplicación situados en la memoria interna del teléfono inteligente.
- dataPath: Variable que almacena la ruta absoluta de los archivos.
- tessBaseApi: Objeto de la Clase TessBaseAPI, que pertenece a la librería Tess-two.
- tessBaseApi.init(dataPath, "spa"): Inicia la librería argumentando la ruta en donde se encuentra el archivo de configuración de idioma y un identificado para el idioma español.
- tessBaseApi.setImage(bitmap): Método de la librería mencionada que se encargará de procesar el mapa de bits enviado como parámetro y que representa la captura de pantalla.
- tessBaseApi.getUTF8Text(): Obtiene los caracteres reconocidos en el formato UTF8 y los almacena en la variable correspondiente.
- tessBaseApi.end(): Indica a la librería que el proceso ha terminado.
- return extractedText: El método actual devuelve la variable con los caracteres reconocidos, en caso que no exista, conservará su valor inicial.

Salida - Caracteres reconocidos en el idioma español

Los caracteres reconocidos en las capturas de pantalla se almacenan en un variable que retorna el método anteriormente analizado.

```
foundText = this.extractText(bitmap);
```

Proceso - Comparación con glosario de palabras ofensivas

En este proceso, se usó un glosario de términos almacenados en una variable dentro del código de programación y con ello se verificó si existe alguna coincidencia con el texto extraído.

```
public String[] dictionaryWords() {  
    String[] dictionary = {"maldito", "bastardo", "torpe", "idiota", "gilipollas",  
"puta", "estúpido", "tontos", "traidor", "sinvergüenza", "sapos",  
"ridículos", "puercos",  
    "mediocre", "mentiroso", "matar", "majaderos", "ladron", "ladrón",  
"hipócrita", "hipocrita", "ignorante", "imbécil", "imbecil", "delincuente",  
"envidioso",  
    "descarado", "cobarde", "gil", "bruto", "tarado", "huevo",  
"huevo", "folle", "follar", "lameculos", "lame culos", "verga", "marica",  
"marika", "cabron",  
    "cabrón", "mazo", "maso", "bambaro", "mierda", "pelotas",  
"mamaverga", "mama verga"  
    };  
    return dictionary;  
}
```

Para hacer uso de este glosario, se usa el método dictionaryWords.

```
String[] glossary = glossaryWords();
```

Ahora simplemente se compara cada palabra de este glosario con el texto extraído y almacenado en la variable foundText.

Salida – Notificación mediante correo electrónico

Si en el proceso anterior existe al menos una palabra ofensiva en el texto que fue extraído de la captura de pantalla, se procede a enviar una notificación a un correo electrónico. Para esto se usa el siguiente método, en donde se profundiza en el siguiente sprint a igual que su resultado.

```
sendEmail("rogervaca1995@gmail.com",  
    "rogervaca1995@gmail.com",  
    "Contenido Prohibido",  
    "El niño tuvo mensajes sobre " + inputWord,  
    new String[]{filePath});
```

c) Retrospectiva del Sprint

Fecha: 10/07/2020	
Asistentes a reunión: Scrum Master, Product Owner y Team development	
¿Que salió bien?	Se consiguió el incremento.
¿Que no salió bien?	Todo salió según lo planteado
¿Qué se debe mejorar?	No aplica.

2.3.6. Sprint 6

a) Planificación de Sprint

Fecha de reunión: 11/07/2020

Asistentes a la reunión: Scrum Master, Product Owner, Development Team.

Objetivo: Desarrollar funcionalidad en la aplicación móvil que permita enviar correos electrónicos en cuanto detecte contenido de riesgo en las capturas de pantalla.

TABLA 45. Planificación Sprint 6

Historia de Usuario	Fase de desarrollo	Tarea	Tiempo estimado (horas)
HU-09	Desarrollo	Crear y configurar funcionalidad para el envío de correos electrónicos a un destinatario predefinido.	10
TOTAL			10

b) Revisión de Sprint

Finalizado el Sprint 6 se muestran a continuación los resultados y las actividades desarrolladas.

TABLA 46. Revisión Sprint 6

Historia de Usuario	Tarea	Tiempo estimado (horas)	Tiempo utilizado (horas)	Estado
---------------------	-------	-------------------------	--------------------------	--------

HU-09	Crear y configurar funcionalidad para el envío de correos electrónicos a un destinatario predefinido.	10	15	Terminado
TOTAL		10	15	

Al finalizar el sprint se muestra el incremento conseguido

- Envío de correos electrónicos

Para el desarrollo de esta funcionalidad se ha considerado el siguiente flujo, ver Fig 37.

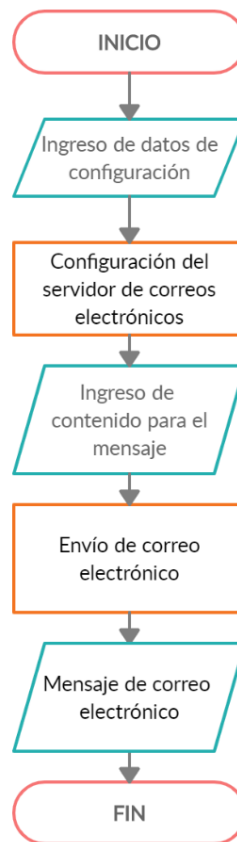


Fig 37. Flujograma envío de correo electrónico
Fuente: Propia

Entrada – Ingreso de datos de configuración

Para ingresar los datos del correo electrónico que emitirá las notificaciones se presentará un formulario con los siguientes campos.

Origen

- email: Correo electrónico del origen de los mensajes.

- password: Contraseña del email de origen.

Destino

- email: Correo electrónico de quien deberá ser notificado en caso de encontrar contenido riesgoso en el dispositivo.

Proceso – Configuración del servidor de correos electrónicos

En este punto se capturan los datos ingresados anteriormente mediante el formulario y se almacenan en la clase “Mail” implementada en el proyecto.

Entrada – Ingreso de contenido para el mensaje

El contenido del mensaje constará de una pequeña descripción indicando si se encontró alguna palabra ofensiva o si se trata de una imagen con objetos clasificados como riesgosos.

Además, a este mensaje se adjunta la imagen que representa el objeto riesgoso o la palabra ofensiva.

Para cumplir estos datos se usa el código siguiente:

```
sendEmail("rogervaca1995@gmail.com",  
          "rogervaca1995@gmail.com",  
          "subject",  
          "message",  
          new String[]{filePath});
```

en donde,

- subject: Encabezado del mensaje de correo electrónico.
- message: Descripción que indica que tipo de contenido se detectó.
- FilePath: Ruta de la captura de pantalla con contenido riesgoso almacenado en la memoria interna del teléfono inteligente.

Proceso – Envío de correo electrónico

Una vez preparado el contenido del mensaje y configurado las cuentas de correo electrónico necesarias, se procedió a enviar los mensajes usando una “AsyncTask” o tarea asíncrona que permite ejecutar esta operación en un segundo plano, y así no interfiere en los procedimientos de primer plano que se ejecutan en la aplicación.

Además, es necesario configurar tres librerías que solamente ayudarán al envío de estos correos, por eso se agregaron al proyecto las siguientes:

- mail.jar
- additionnal.jar
- activation.jar

Salida – Mensaje de correo electrónico.

Para evidenciar el resultado de este proceso y el cumplimiento del incremento, se muestra a continuación varios mensajes enviados al correo electrónico indicando el tipo de contenido detectado.

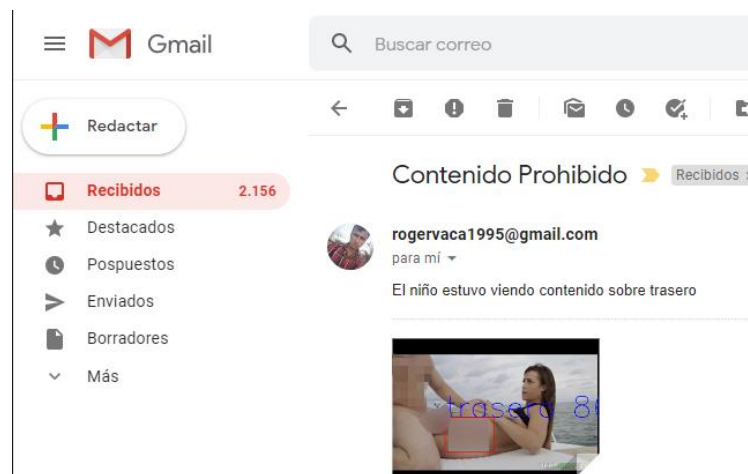


Fig 38. Correo electrónico con alerta de imagen pornográfica
Fuente: Propia

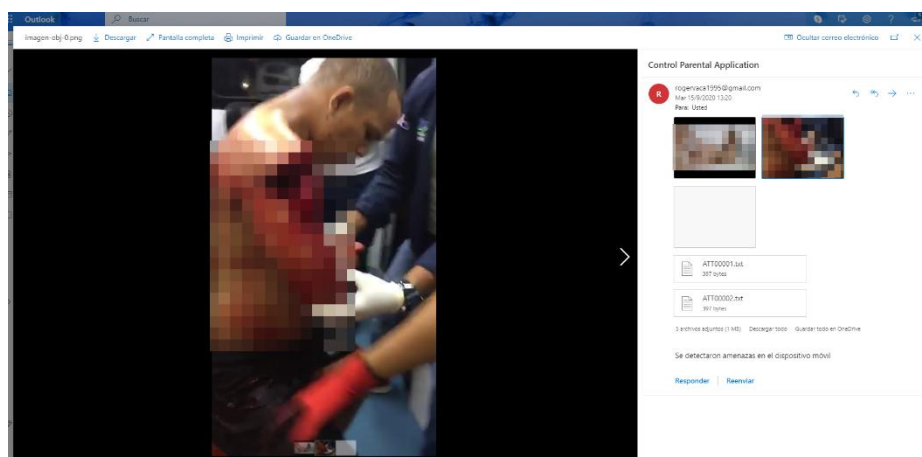


Fig 39. Correo electrónico con alerta de imagen sangrienta
Fuente: Propia

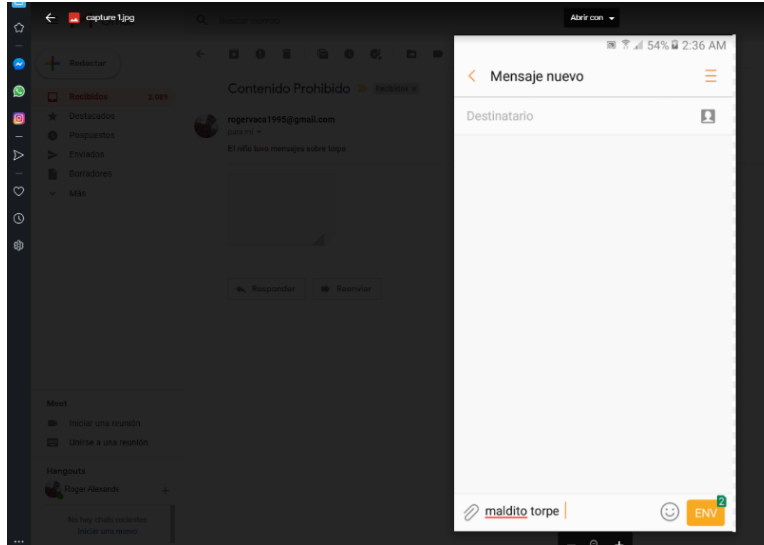


Fig 40. Correo electrónico con alerta de palabras ofensivas “torpe”
Fuente: Propia

c) Retrospectiva del Sprint

TABLA 47. Retrospectiva Sprint 6

RETROSPECTIVA	
Fecha: 13/07/2020	
Asistentes a reunión: Scrum Master, Product Owner y Team development	
¿Que salió bien?	El incremento se cumplió correctamente
¿Que no salió bien?	Tomó un poco más de tiempo de lo planteado
¿Qué se debe mejorar?	No aplica.

CAPÍTULO 3

Validación de Resultados

3.1. Muestreo

Para conocer el número de la muestra a usar, se usó la prueba de Kruskal-Wallis orientada a poblaciones que tengan más de dos categorías, además de tipo cuantitativo no-normal debido a que se consideran varias versiones de sistemas operativos Android. En este caso la prueba estadística aplicada es ANOVA de una vía.

Para el cálculo muestral se utilizó el programa estadístico G*Power de la Universidad de California en los Ángeles y está diseñado para realizar estimaciones de la potencia estadística y del tamaño del efecto. En la Fig 53 se puede observar los datos obtenidos.

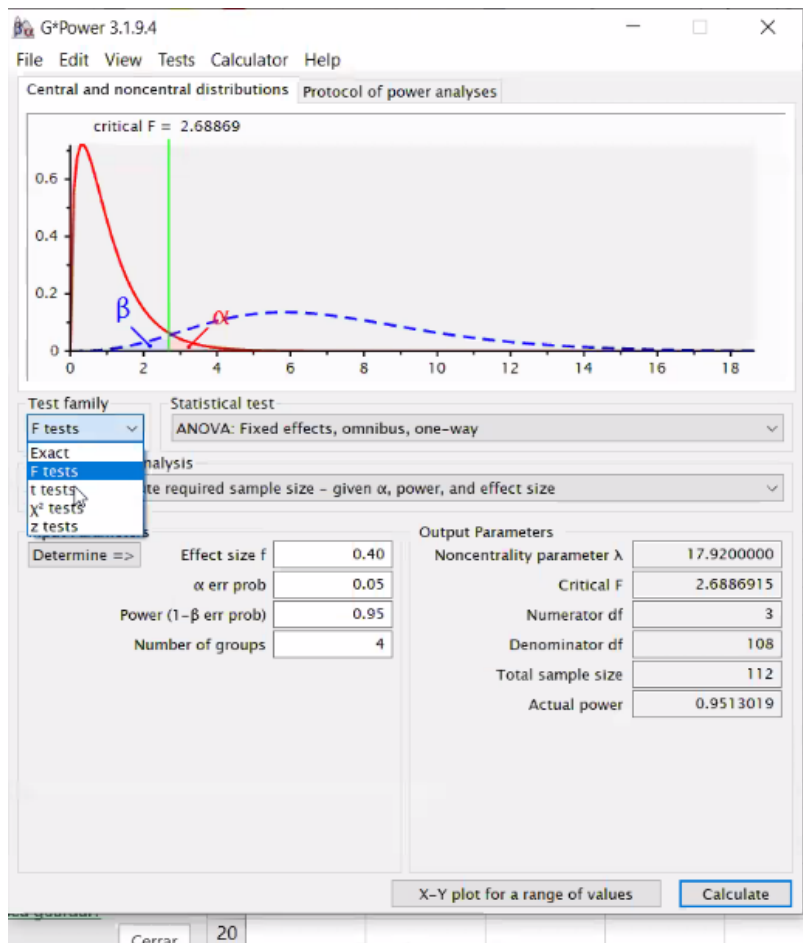


Fig 41. Muestreo ANOVA con GPower
Fuente: propia

La variable “Denominator df” indica la cantidad de individuos a los que se deben tomar para aplicar las respectivas pruebas.

Además, se realizó un muestreo de la familia “Exact” con una prueba de regresión múltiple lineal usando un modelo randómico. En la Fig 54 se muestran los resultados de este cálculo.

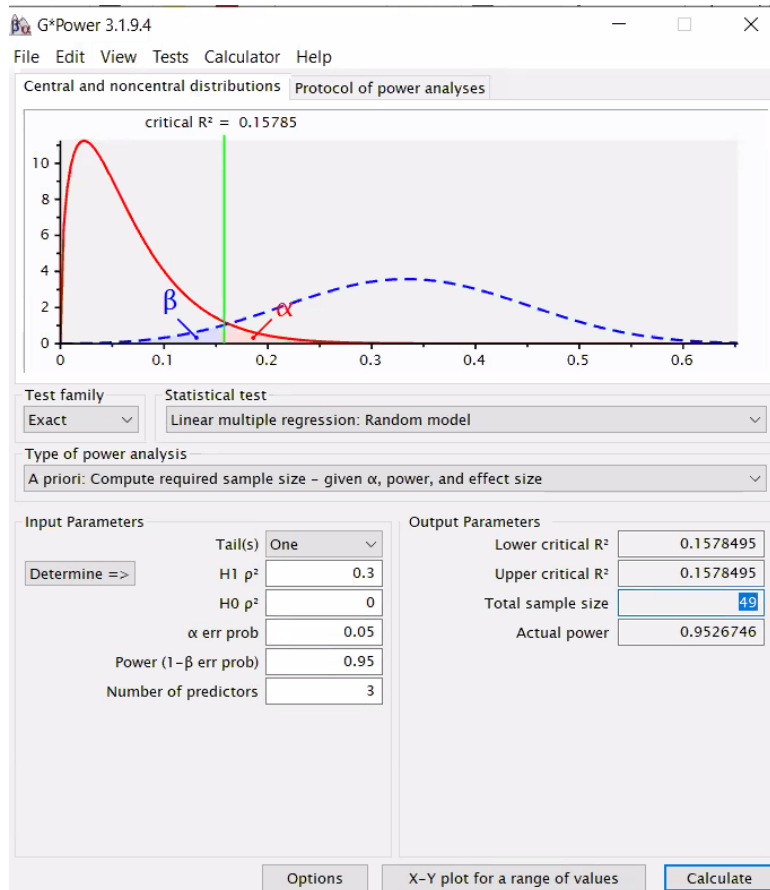


Fig 42. Muestreo de regresión múltiple lineal
Fuente: propia

Entonces las pruebas se aplicaron de la siguiente manera:

- Prueba 1: Se ejecutó la aplicación en un solo teléfono inteligente en donde 25 imágenes contenían objetos de riesgo de la misma clase y las otras 25 no contenía ningún objeto de riesgo. Esta prueba fue realizada por cada clase de objeto.
- Prueba 2: La aplicación fue ejecutada en 108 teléfonos inteligentes divididos en cuatro grupos de 27 teléfonos cada uno, correspondientes a las versiones de Android 6, 7, 8 y 9.

- ✓ En cada teléfono se probó un conjunto de 50 imágenes en donde 25 de ellas contenían objetos riesgosos de clase variada y 25 no contenía ningún objeto de riesgo.

3.2. Matrices de confusión

La matriz de confusión permite conocer el rendimiento del algoritmo utilizado en este proyecto manejando ciertas variables como son: verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. En la Fig 55 se visualiza detalladamente su estructura.

TABLA 48. Estructura matriz de confusión

		PREDICCIÓN DE ALGORITMO	
		Positivos	Negativos
VALOR REAL	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Fuente: propia

En donde la representación se entiende de la siguiente forma:

VP. – El detector clasificó como positivos y en realidad son positivos.

VN. – El detector clasificó como negativos y en realidad son negativos.

FP. – El detector clasificó como positivos, pero en realidad son negativos.

FN. – El detector clasificó como negativos, pero en realidad son positivos.

3.3. Métricas de desempeño

Estas métricas permiten evaluar ciertos comportamientos del modelo de la visión artificial desarrollado y probado en los diferentes dispositivos, entre ellos son: precisión, exactitud, especificidad, exhaustividad, valor F1.

3.3.1. Precisión.

Permite medir la calidad del modelo de visión artificial orientado a la clasificación usando la siguiente fórmula.

$$precision = \frac{VP}{VP + FP}$$

3.3.2. Exactitud.

La exactitud permite medir el porcentaje de los resultados positivos que obtuvo el modelo de visión artificial y se calcula con la siguiente fórmula.

$$accuracy = \frac{VP + VN}{VP + VN + FP + FN}$$

3.3.3. Especificidad.

Informa el número de resultados identificados como negativos mediante la siguiente fórmula.

$$recall = \frac{VN}{VN + FP}$$

3.3.4. Exhaustividad.

Permite conocer la cantidad máxima que el modelo de inteligencia artificial puede identificar y se usa la siguiente fórmula.

$$recall = \frac{VP}{VP + FN}$$

3.3.5. Valor F1.

Combina las métricas de precisión y exhaustividad para facilitar la comparación entre varias soluciones y se emplea la fórmula siguiente.

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

3.4. Resultados

3.4.1. Resultados por clase

TABLA 49. Resultados por clase

Clase	VP	FN	FP	VN	Precisión	Exactitud	Especificidad	Sensibilidad	Valor F1
senos	24	1	1	24	0.96	0.96	0.96	0.96	0.96
vagina	21	4	0	25	1	0.92	1	0.84	0.91

trasero	18	7	0	25	1	0.86	1	0.72	0.83
pene	16	9	0	25	1	0.82	1	0.64	0.78
testículos	15	10	1	24	0.93	0.78	0.96	0.6	0.73
ano	21	4	1	24	0.95	0.9	0.96	0.84	0.89
sangriento	17	8	3	22	0.85	0.78	0.88	0.68	0.75
mutilación	22	3	2	23	0.91	0.9	0.92	0.88	0.89

3.4.2. Resultados por versión de Android

TABLA 50. Resultados en Android 6

Android	VP	FN	FP	VN	Precisión	Exactitud	Especificidad	Sensibilidad	Valor-F1
6.0.1	20	5	0	25	1	0.9	1	0.8	0.888888889
6.0.1	21	4	0	25	1	0.92	1	0.84	0.913043478
6.0.1	18	7	0	25	1	0.86	1	0.72	0.837209302
6.0.1	20	5	0	25	1	0.9	1	0.8	0.888888889
6.0.1	23	2	0	25	1	0.96	1	0.92	0.958333333
6.0.1	21	4	0	25	1	0.92	1	0.84	0.913043478
6.0.1	22	3	1	24	0.956521739	0.92	0.96	0.88	0.916666667
6.0.1	19	6	0	25	1	0.88	1	0.76	0.863636364
6.0.1	24	1	0	25	1	0.98	1	0.96	0.979591837
6.0.1	19	6	0	25	1	0.88	1	0.76	0.863636364
6.0.1	20	5	1	24	0.952380952	0.88	0.96	0.8	0.869565217
6.0.1	18	7	3	22	0.857142857	0.8	0.88	0.72	0.782608696
6.0.1	19	6	2	23	0.904761905	0.84	0.92	0.76	0.826086957
6.0.1	19	6	1	24	0.95	0.86	0.96	0.76	0.844444444
6.0.1	17	8	1	24	0.944444444	0.82	0.96	0.68	0.790697674
6.0.1	22	3	0	25	1	0.94	1	0.88	0.936170213
6.0.1	16	9	1	24	0.941176471	0.8	0.96	0.64	0.761904762
6.0.1	22	3	0	25	1	0.94	1	0.88	0.936170213
6.0.1	19	6	1	24	0.95	0.86	0.96	0.76	0.844444444
6.0.1	23	2	0	25	1	0.96	1	0.92	0.958333333
6.0.1	17	8	1	24	0.944444444	0.82	0.96	0.68	0.790697674
6.0.1	21	4	5	18	0.807692308	0.8125	0.782608696	0.84	0.823529412
6.0.1	18	7	4	21	0.818181818	0.78	0.84	0.72	0.765957447

6.0.1	21	4	0	25	1	0.92	1	0.84	0.913043478
6.0.1	21	4	0	25	1	0.92	1	0.84	0.913043478
6.0.1	22	3	1	24	0.956521739	0.92	0.96	0.88	0.916666667
6.0.1	19	6	0	25	1	0.88	1	0.76	0.863636364

TABLA 51. Resultados en Android 7

Android	VP	FN	FP	VN	Precisión	Exactitud	Especificidad	Sensibilidad	Valor-F1
7.1.1	19	6	1	24	0.95	0.86	0.96	0.76	0.844444444
7.1.1	18	7	0	25	1	0.86	1	0.72	0.837209302
7.1.1	23	2	1	24	0.958333333	0.94	0.96	0.92	0.93877551
7.1.1	16	9	3	22	0.842105263	0.76	0.88	0.64	0.727272727
7.1.1	17	8	2	23	0.894736842	0.8	0.92	0.68	0.772727273
7.1.1	19	6	4	21	0.826086957	0.8	0.84	0.76	0.791666667
7.1.1	15	10	2	23	0.882352941	0.76	0.92	0.6	0.714285714
7.1.1	20	5	2	23	0.909090909	0.86	0.92	0.8	0.85106383
7.1.1	18	7	1	24	0.947368421	0.84	0.96	0.72	0.818181818
7.1.1	16	9	2	23	0.888888889	0.78	0.92	0.64	0.744186047
7.1.1	18	7	1	24	0.947368421	0.84	0.96	0.72	0.818181818
7.1.1	20	5	1	24	0.952380952	0.88	0.96	0.8	0.869565217
7.1.1	17	7	2	23	0.894736842	0.816326531	0.92	0.708333333	0.790697674
7.1.1	22	3	1	24	0.956521739	0.92	0.96	0.88	0.916666667
7.1.1	18	7	1	24	0.947368421	0.84	0.96	0.72	0.818181818
7.1.1	16	9	3	22	0.842105263	0.76	0.88	0.64	0.727272727
7.1.1	17	8	2	23	0.894736842	0.8	0.92	0.68	0.772727273
7.1.1	19	6	1	24	0.95	0.86	0.96	0.76	0.844444444
7.1.1	18	7	2	23	0.9	0.82	0.92	0.72	0.8
7.1.1	20	5	1	24	0.952380952	0.88	0.96	0.8	0.869565217
7.1.1	21	4	0	25	1	0.92	1	0.84	0.913043478
7.1.1	17	8	2	23	0.894736842	0.8	0.92	0.68	0.772727273
7.1.1	16	9	2	23	0.888888889	0.78	0.92	0.64	0.744186047
7.1.1	20	5	1	24	0.952380952	0.88	0.96	0.8	0.869565217
7.1.1	16	9	1	24	0.941176471	0.8	0.96	0.64	0.761904762
7.1.1	16	9	2	23	0.888888889	0.78	0.92	0.64	0.744186047
7.1.1	19	6	4	21	0.826086957	0.8	0.84	0.76	0.791666667

TABLA 52. Resultados en Android 8

Android	VP	FN	FP	VN	Precisión	Exactitud	Especificidad	Sensibilidad	Valor-F1
8.0.0	16	9	1	24	0.941176471	0.8	0.96	0.64	0.761904762
8.0.0	18	7	2	23	0.9	0.82	0.92	0.72	0.8
8.0.0	21	4	1	24	0.954545455	0.9	0.96	0.84	0.893617021
8.0.0	23	2	0	25	1	0.96	1	0.92	0.958333333
8.0.0	22	3	1	24	0.956521739	0.92	0.96	0.88	0.916666667
8.0.0	17	8	2	23	0.894736842	0.8	0.92	0.68	0.772727273
8.0.0	16	9	2	23	0.888888889	0.78	0.92	0.64	0.744186047
8.0.0	17	8	3	22	0.85	0.78	0.88	0.68	0.755555556
8.0.0	16	9	3	22	0.842105263	0.76	0.88	0.64	0.727272727
8.0.0	19	6	2	23	0.904761905	0.84	0.92	0.76	0.826086957
8.0.0	20	5	1	24	0.952380952	0.88	0.96	0.8	0.869565217
8.0.0	17	8	1	24	0.944444444	0.82	0.96	0.68	0.790697674
8.1.0	16	9	2	23	0.888888889	0.78	0.92	0.64	0.744186047
8.1.0	23	2	0	25	1	0.96	1	0.92	0.958333333
8.1.0	20	5	1	24	0.952380952	0.88	0.96	0.8	0.869565217
8.1.0	18	7	2	23	0.9	0.82	0.92	0.72	0.8
8.1.0	16	9	1	24	0.941176471	0.8	0.96	0.64	0.761904762
8.1.0	18	7	1	24	0.947368421	0.84	0.96	0.72	0.818181818
8.1.0	21	4	1	24	0.954545455	0.9	0.96	0.84	0.893617021
8.1.0	19	6	0	25	1	0.88	1	0.76	0.863636364
8.1.0	20	5	2	23	0.909090909	0.86	0.92	0.8	0.85106383
8.0.0	19	6	2	23	0.904761905	0.84	0.92	0.76	0.826086957
8.0.0	16	9	3	22	0.842105263	0.76	0.88	0.64	0.727272727
8.0.0	16	9	1	24	0.941176471	0.8	0.96	0.64	0.761904762
8.0.0	19	6	2	23	0.904761905	0.84	0.92	0.76	0.826086957
8.0.0	17	8	1	24	0.944444444	0.82	0.96	0.68	0.790697674
8.0.0	18	7	2	23	0.9	0.82	0.92	0.72	0.8

TABLA 53. Resultados en Android 9

Android	VP	FN	FP	VN	Precisión	Exactitud	Especificidad	Sensibilidad	Valor-F1
9	20	5	0	25	1	0.9	1	0.8	0.888888889
9	15	10	3	22	0.833333333	0.74	0.88	0.6	0.697674419
9	16	9	1	24	0.941176471	0.8	0.96	0.64	0.761904762
9	21	4	1	24	0.954545455	0.9	0.96	0.84	0.893617021
9	18	7	0	25	1	0.86	1	0.72	0.837209302
9	18	7	1	24	0.947368421	0.84	0.96	0.72	0.818181818
9	19	6	1	24	0.95	0.86	0.96	0.76	0.844444444

9	17	8	1	24	0.944444444	0.82	0.96	0.68	0.790697674
9	16	9	2	23	0.888888889	0.78	0.92	0.64	0.744186047
9	18	7	1	24	0.947368421	0.84	0.96	0.72	0.818181818
9	16	9	3	22	0.842105263	0.76	0.88	0.64	0.727272727
9	20	5	1	24	0.952380952	0.88	0.96	0.8	0.869565217
9	19	6	1	24	0.95	0.86	0.96	0.76	0.844444444
9	16	9	1	24	0.941176471	0.8	0.96	0.64	0.761904762
9	20	5	0	25	1	0.9	1	0.8	0.888888889
9	19	6	1	24	0.95	0.86	0.96	0.76	0.844444444
9	22	3	0	25	1	0.94	1	0.88	0.936170213
9	21	4	0	25	1	0.92	1	0.84	0.913043478
9	16	9	2	23	0.888888889	0.78	0.92	0.64	0.744186047
9	23	2	0	25	1	0.96	1	0.92	0.958333333
9	21	4	0	25	1	0.92	1	0.84	0.913043478
9	19	6	1	24	0.95	0.86	0.96	0.76	0.844444444
9	20	5	2	23	0.909090909	0.86	0.92	0.8	0.85106383
9	19	6	3	22	0.863636364	0.82	0.88	0.76	0.808510638
9	18	7	3	22	0.857142857	0.8	0.88	0.72	0.782608696
9	16	9	2	23	0.888888889	0.78	0.92	0.64	0.744186047
9	19	6	1	24	0.95	0.86	0.96	0.76	0.844444444

3.4.3. Resumen de Resultados

TABLA 54. Resumen de Resultados

VP	FN	FP	VN	Precisión	Exactitud	Especificidad	Sensibilidad	Valor-F1
2033	667	139	2561	0.936003683	0.850740741	0.948518519	0.752962963	0.83456486

Para esta fase se organizó el set de imágenes de la siguiente forma, ver Tabla 55.

TABLA 55. Set de imágenes de prueba

	Positivas	Negativas
Imágenes	2700	2700

3.5. Curva ROC

La Curva ROC es una métrica que permite medir el rendimiento en este caso del modelo de visión artificial el cuál se concentra en la sensibilidad y la especificidad más alta para así evaluar la capacidad de identificar entre el objetivo deseado y lo contrario.

Representación de los datos obtenidos en el proceso de validación en gráficos estadísticos de la curva ROC, en la Fig 43 se observa la gráfica referente al resumen de resultados presentado en el punto anterior.

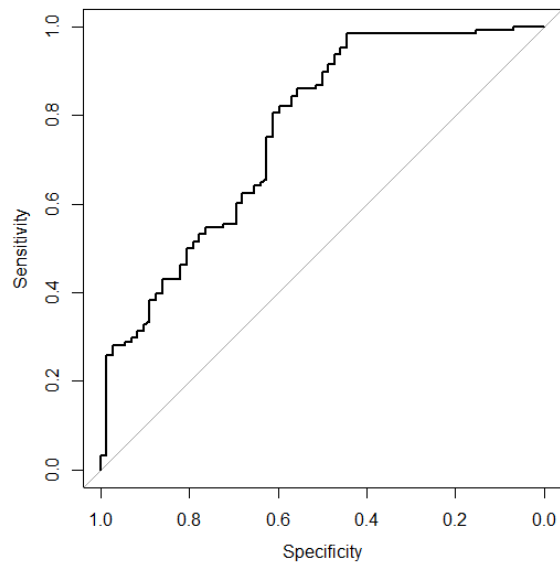


Fig 43. Curva ROC
Fuente: Propia

CONCLUSIONES

- Al desarrollar el capítulo I, sobre la investigación de la base teórica se estableció la técnica de visión artificial a usar (Tiny YOLO), así como también el contenido que se considera riesgoso para niños, niñas y adolescentes que posteriormente será detectado mediante capturas de pantallas en teléfonos inteligentes y además de seleccionar el IDE de desarrollo y la tecnología para construir la aplicación móvil (Android Studio).
- Al conocer el tipo de contenido riesgoso por detectar, se procedió a elaborar un dataset que contengan numerosas cantidades de estas imágenes para luego graficar los cuadros delimitadores que indiquen en la imagen el lugar exacto del objeto de riesgo. Para esto se usó el software labellmg.
- Para entrenar el modelo de visión artificial se utilizó un computador de altas prestaciones en la nube como lo es Google Colab, ya que provee las librerías necesarias para usar el algoritmo seleccionado previamente y ejecutar los procesos de forma exitosa usando su versión gratuita.
- El desarrollo de la aplicación móvil fue sobre Android Studio y se configuró sobre éste, la librería OpenCV, que permitió vincular el modelo de visión artificial anteriormente elaborado, haciendo uso de un paquete de Redes neuronales (DNN).
- Para la evaluación del prototipo de visión artificial se sustrajo la muestra haciendo uso de la regresión lineal multivariada mediante el lenguaje de Programación R (RStudio), para después con estos resultados realizar pruebas necesarias y extraer métricas de rendimiento adecuadas.
- El desarrollo de este proyecto fue monitoreado mediante el Marco de trabajo SCRUM, para cumplir con la planificación planteada.

RECOMENDACIONES

- Revisar la literatura y trabajos anteriores sobre proyectos de visión artificial para elegir el algoritmo o método adecuado tomando en cuenta el dispositivo sobre el cuál funcionaría.
- Seleccionar un IDE de desarrollo que facilite la integración de librerías de visión artificial.
- Al elaborar el dataset de imágenes, considerar tres conjuntos: entrenamiento, pruebas y validación, en donde las imágenes de entrenamiento serán las únicas que contenga los cuadros delimitadores y servirá para construir el prototipo. Las imágenes de validación servirán para evaluar si el modelo de visión artificial necesita ser modificado en ciertos parámetros para mejorar su precisión, y por último las imágenes de prueba servirán para verificar la existencia del contenido deseado en imágenes sumamente desconocidas para el algoritmo.
- Además, para la elaboración del dataset de imágenes es recomendable realizarlo con un número excesivo de imágenes, como de 80, 90 o mas de 100 mil ya que mientras mas se le entrene al modelo mas precisión adquiere. Para esto se puede aplicar la técnica de data augmentation en donde a partir de una imagen se generan muchas más, ya que en cada una se aplicará una técnica diferente que permita crear una imagen distinta de la otra.
- Usar un marco de trabajo que permita agilizar el cumplimiento de la planificación del proyecto, para esto están las metodologías ágiles, como lo es SCRUM.
- Usar Herramientas de estadística para validar los resultados y así mejorar la calidad del producto.

REFERENCIAS

- Aciprensa. (2019). *El Papa Francisco pide mayor protección a los menores en internet*.
<https://www.aciprensa.com/noticias/el-papa-pide-medidas-para-evitar-los-abusos-a-menores-en-internet-96850>
- ACNUR, A. de la O. para los R. (2003). *Código de la niñez y adolescencia*. 100.
- Acodigo. (2018). *Compilar el SDK OpenCV para Android*.
<http://acodigo.blogspot.com/2018/02/compilar-el-sdk-opencv-para-android.html>
- Adell, J. (2014). *Internet en educación*. March.
- Adeva, R. (2020). *Android: qué es, versiones, aplicaciones y cómo saber la versión instalada*.
<https://www.adslzone.net/reportajes/software/que-es-android/>
- Allison, K. (2018). Online Risks, Sexual Behaviors, and Mobile Technology Use in Early Adolescent Children: Parental Awareness, Protective Practices, and Mediation. *ProQuest Dissertations and Theses*, 130.
https://search.proquest.com/docview/2185936326?accountid=13042%0Ahttp://oxfordfx.ho.sted.exlibrisgroup.com/oxford?url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:dissertation&genre=dissertations+%26+theses&sid=ProQ:ProQuest+Dissertations+%26+Theses+G
- Alvear, V. (2017). Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura (Internet of Things and Artificial Vision, Performance and Applications: Literature Review). *Enfoque UTE*, 8(1), 244–256.
<http://ingenieria.ute.edu.ec/enfoqueute/>
- Báez, M., Borrego, Á., Cordero, J., Cruz, L., González, M., Hernández, F., Palomero, D., Llera, J. R. De, & Sanz, D. (2019). *Introducción a Android*.
- Catalina García, B., López de Ayala López, M. C., & García Jiménez, A. (2014). Los riesgos de los adolescentes en Internet: Los menores como actores y víctimas de los peligros de Internet. *Revista Latina de Comunicación Social*, 69, 462–485.
<https://doi.org/10.4185/RLCS-2014-1020>
- Chaudhuri, A., Mandaviya, K., Badelia, P., & K Ghosh, S. (2017). Optical Character Recognition Systems for Different Languages with Soft Computing. In *Springer* (Vol. 352, Issue December). <https://doi.org/10.1007/978-3-319-50252-6>
- Chaves, D., Saikia, S., Fernández-Robles, L., Alegre, E., & Trujillo, M. (2018). Una Revisión Sistemática de Métodos para Localizar Automáticamente Objetos en Imágenes. *Revista Iberoamericana de Automática e Informática Industrial*, 15(3), 231.
<https://doi.org/10.4995/riai.2018.10229>
- craftcodecrew.com. (2018). *OpenCV & Tesseract: Android Computer Vision for Dummies – CraftCode Crew*. <https://craftcodecrew.com/opencv-tesseract-android-computer-vision-for-dummies/>
- Ebro, B. (2019). *Niños y tecnología. Ventajas e inconvenientes de los dispositivos móviles*. - *Sentirse bien se nota*. <https://www.sentirsebiennota.com/2019/05/17/ninos-y-tecnologia-ventajas-e-inconvenientes-de-los-dispositivos-moviles/>
- Europeanscrum. (2020). *Formacion Scrum - EuropeanScrum.org - The official organization of*

- Scrum in Europe for any sector. <https://www.europeanscrum.org/formacion-scrum.html>
- github.rmtheis. (2019). *GitHub - rmtheis/tess-two: Fork of Tesseract Tools for Android*. <https://github.com/rmtheis/tess-two>
- Goette, P. E. (2014). R, un lenguaje y entorno de programación para análisis estadístico. *Genbeta*. <https://www.genbeta.com/desarrollo/r-un-lenguaje-y-entorno-de-programacion-para-analisis-estadistico>
- Hardzone.es. (2019). *Spyzie: la mejor aplicación de control parental para iOS y Android*. <https://hardzone.es/2019/04/11/spyzie-mejor-aplicacion-control-parental-ios-android/>
- Hazra, T. K., Singh, D. P., & Daga, N. (2017). Optical character recognition using KNN on custom image dataset. *2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON)*, 110–114. <https://doi.org/10.1109/IEMECON.2017.8079572>
- He, X., & Deng, L. (2017). Deep learning for image-to-text generation: A technical overview. *IEEE Signal Processing Magazine*, 34(6), 109–116. <https://doi.org/10.1109/MSP.2017.2741510>
- ING. MARCIAL BASILIO, J. A., & ALBERTO, J. (2015). *DETECCIÓN DE IMÁGENES CON CONTENIDO EXPLÍCITO USANDO LOS MODELOS DE COLOR HSV y YCbCr*. <http://www.repositoriodigital.ipn.mx/handle/123456789/12664>
- Jiménez Iglesias, E., Garmendia Larrañaga, M., & Casado del Río, M. (2015). Percepción de los y las menores de la mediación parental respecto a los riesgos en internet. *Revista Latina de Comunicación Social*, 70, 49–68. <https://doi.org/10.4185/RLCS-2015-1034>
- José Miguel Roca Chillida. (2018). *¿Qué es un smartphone?* <http://www.informeticplus.com/que-es-un-smartphone>
- Livingstone, S. (2019). EU Kids Online . *The International Encyclopedia of Media Literacy*, 1–17. <https://doi.org/10.1002/9781118978238.ieml0065>
- Lozano, M. (2018). *¿Qué es un Smartphone? Conoce todos los detalles | WhistleOut*. <https://www.whistleout.com.mx/CellPhones/Guides/que-es-un-smartphone>
- Ma, Jing.;Chen, Li.;Gao, Z. . (2018). *Hardware Implementation and Optimization of Tiny-YOLO Network*. 815, 193–202. <https://doi.org/10.1007/978-981-10-8108-8>
- Madhuram, M., & Parameswaran, A. (2018). *A Text Extraction Approach towards Document Image Organisation for the Android Platform*. 1560–1565.
- Maida, E., & Pacienza, J. (2015). *Metodologías de desarrollo de software*. 12–13. <http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf>
- Martinez, A. (2019). *La realidad del ciberacoso en Ecuador y cómo denunciarlo | Metro Ecuador*. <https://www.metroecuador.com.ec/ec/estilo-vida/2019/05/10/telefonica-movistar-la-realidad-del-ciberacoso-en-ecuador-y-como-denunciarlo.html>
- Minardi, F. (2016). *Vision Artificial*.
- Naciones Unidas. (2015). *Paz y justicia - Desarrollo Sostenible*. <https://www.un.org/suNaciones Unidas>. (2015). Paz y justicia - Desarrollo Sostenible. <https://www.un.org/sustainabledevelopment/es/peace->

justice/sustainabledevelopment/es/peace-justice/

- NetMarketShare. (2020). *Operating system market share*. WebArticle. <https://www.netmarketshare.com/operating-system-market-share.aspx?options=%7B%22filter%22%3A%7B%22%24and%22%3A%5B%7B%22deviceType%22%3A%7B%22%24in%22%3A%5B%22Mobile%22%5D%7D%7D%5D%7D%2C%22dateLabel%22%3A%22Custom%22%2C%22attributes%22%3A%22share%22%2C%22g>
- Nuñez, F. J. (2016). Diseño de un sistema de reconocimiento automático de matrículas de vehículos mediante una red neuronal convolucional. *Universitat Oberta de Catalunya*, 53. <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/52222/6/fnunezsTFM0616memoria.pdf>
- OpenCV4Android SDK — OpenCV 2.4.5.0 documentation. (2019). https://docs.opencv.org/2.4/doc/tutorials/introduction/android_binary_package/O4A_SDK.html#general-info
- Perez, A. (2019). DESARROLLO DE UN CLASIFICADOR VISUAL DE ESPECIES DE AVES MEDIANTE REDES NEURONALES CONVOLUCIONALES. *Journal of Chemical Information and Modeling*, 110(9), 1689–1699.
- Raffino, M. (2019). *Internet: Concepto, Tipos, Servicios, Usos y Navegadores*. 1 de Marzo de 2019. <https://concepto.de/internet/>
- Redmon, Joseph.; Santosh , Divvala.; Ross, Girshick.; Ali, F. (2016). (YOLO) You Only Look Once. *Cvpr*. <https://doi.org/10.1109/CVPR.2016.91>
- Redmon, J. S. D. R. G. A. F. (2016). (YOLO) You Only Look Once. *Cvpr*. <https://doi.org/10.1109/CVPR.2016.91>
- Rouhiainen, L. (2018). *Inteligencia artificial 101*.
- SAS. (2020). *Computer Vision: What it is and why it matters | SAS*. SASinsights. https://www.sas.com/en_us/insights/analytics/computer-vision.html
- Schwaber, K., & Sutherland, J. (2017). *La Guía de Scrum. La Guía Definitiva de Scrum: Las Reglas del Juego*. 22. <http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf#zoom=100>
- Sepín. (2016). *¿Pueden los padres violar realmente la intimidad de los menores?* <https://blog.sepin.es/2016/09/intimidad-menores-control-parental/>
- Serrano-Cobos, J. (2016). Internet technology trends: Towards a paradigm shift. *Profesional de La Información*, 25(6), 843–850. <https://doi.org/10.3145/epi.2016.nov.01>
- Solórzano, Edith . Bohórquez, J. (2016). *PROTOTIPO DE UN CONTROL PARENTAL PARA EL INTERNET EN EL HOGAR, OPERADO DESDE UN DISPOSITIVO ANDROI* (Vol. 2002, Issue 1). <https://doi.org/10.1109/ciced.2018.8592188>
- spyzie.com. (2020). *Conoce Instantáneamente la Actividad en el Dispositivo Objetivo con la Obtención de Capturas de Pant*. <https://www.spyzie.com/es/screenshots-capture.html>
- Unicef. (2006). *Convención sobre los derechos del niño*. 75–96. <https://doi.org/10.18356/51f8034c-es>

- UNICEF, F. de las N. U. para la I. (2017). *Estado Mundial de la Infancia 2017. Niños en un mundo digital*. www.unicef.org/SOWC2017
- Unir. (2019). *Lenguaje R, ¿qué es y por qué es tan usado en Big Data?*
<https://www.unir.net/ingenieria/revista/noticias/lenguaje-r-big-data/549204698515/>
- United Nations. (2014). Derechos de la infancia en la era digital. *Desafíos: Boletín de La Infancia y Adolescencia Sobre El Avance de Los Objetivos de Desarrollo Del Milenio. Derechos de La Infancia En La Era Digital*, 18, 1–12.
<http://repositorio.cepal.org/handle/11362/37139>
- Velasco;Calle, D. (2015). *Control Parental vs Derecho a la privacidad*.
<https://www.velascocalle.co/2015/09/01/control-parental-vs-derecho-a-la-privacidad/>
- Viewnext.com. (2019). *Artefactos Scrum ¿Qué son y para qué sirven?* | Viewnext.
<https://www.viewnext.com/artefactos-scrum/>
- Zaman, B., & Nouwen, M. (2016). Parental controls : advice for parents , researchers and industry. *Eukidsonline*, February, 1–9. <https://doi.org/10.1053/j.jfas.2015.12.002>

ANEXOS

ANEXO 1: Política de Privacidad para el uso de la Aplicación Móvil

¿Qué información obtiene el aplicativo y como es utilizada?

La aplicación capturará el contenido de la pantalla del dispositivo móvil.

El aplicativo sirve para el Control Parental que analiza el tipo de imágenes y de texto que se visualicen en la pantalla del teléfono inteligente sin la necesidad de migrar a un dispositivo externo mediante la red u otros medios.

Al instalar la aplicación, se solicitará al usuario mediante una ventana emergente que brinde los permisos necesarios para proceder a las capturas de pantallas de forma automática por una sola vez para que dure por un tiempo prolongado.

En caso de que el aplicativo detecte contenido riesgoso en las capturas de pantalla, se proceden a guardar estas imágenes en la memoria interna del mismo teléfono inteligente y además se enviará un mensaje de correo electrónico hacia el tutor o padre de familia del niño o adolescente que lo esté usando, adjuntando como respaldo el contenido hallado.

¿Pueden terceros obtener acceso a la información obtenida por el aplicativo?

Únicamente cuando sea requerido por la ley. Cuando creamos que es necesario para proteger nuestros derechos, proteger su seguridad o la seguridad de otros, investigar fraude o en respuesta a una solicitud del gobierno.

Política de retención y manejo de información

Se conservarán únicamente las últimas 30 imágenes detectadas como riesgosas, en la memoria del teléfono inteligente ya que se necesita el espacio en el dispositivo móvil para el uso correcto del aplicativo. Además, las imágenes que fueron enviadas al correo electrónico dependerán de la administración del tutor o padre de familia mismo.

Seguridad

Nos preocupamos por salvaguardar la confidencialidad de su información, por lo tanto, se menciona que la única persona a quien se facilitará el contenido de riesgo hallado en el dispositivo será el tutor o padre de familia de la persona quien lo usa. Tenga en cuenta que, aunque nos esforzamos por proveer un nivel de seguridad razonable para manejar y procesar la información, ningún sistema de seguridad puede prevenir todas las brechas de seguridad potenciales.

Cambios

Esta Política de Privacidad puede ser actualizada de vez en cuando por cualquier razón. Le notificaremos de cualquier cambio en nuestra política de privacidad mediante la publicación de la nueva política de privacidad.

Su Consentimiento

Al aceptar el monitoreo del contenido visualizado en el dispositivo, usted está consintiendo que el contenido encontrado en el mismo sea mostrado al representante de quien lo usa.