

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI



FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

Tema: “Agenda digital personalizada para la organización de actividades académicas”

Trabajo de titulación previa la obtención del
título de Ingeniero en Ciencias de la Computación

AUTORES: Anrango Oyagata Malkik Salvador

Rosero Acosta Noreen Odalys

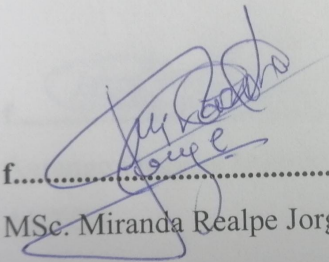
TUTOR: Msc. Miranda Realpe Jorge Humberto

Tulcán, 2022

CERTIFICADO JURADO EXAMINADOR

Certificamos que los estudiantes Anrango Oyagata Malkik Salvador con el número de cédula 1004642805 y Rosero Acosta Noreen Odalys con el número de cédula 1726169335 han elaborado el trabajo de titulación: “Agenda digital personalizada para la organización de actividades académicas”

Este trabajo se sujeta a las normas y metodología dispuesta en el Reglamento de Titulación, Sustentación e Incorporación de la UPEC, por lo tanto, autorizamos la presentación de la sustentación para la calificación respectiva.



f.....
MSc. Miranda Realpe Jorge Humberto

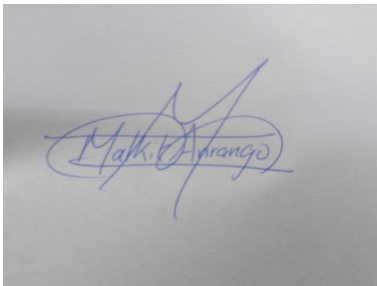
TUTOR

Tulcán, abril de 2022

AUTORÍA DE TRABAJO

El presente trabajo de titulación constituye requisito previo para la obtención del título de **Ingeniero** en la Carrera de computación de la Facultad de Industrias Agropecuarias y Ciencias Ambientales

Yo, Anrango Oyagata Malkik Salvador con cédula de identidad número 1004642805 declaro: que la investigación es absolutamente original, auténtica, personal y los resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.

A square image showing a handwritten signature in blue ink. The signature is stylized and appears to read 'Malkik Anrango'.

f.....

Anrango Oyagata Malkik Salvador

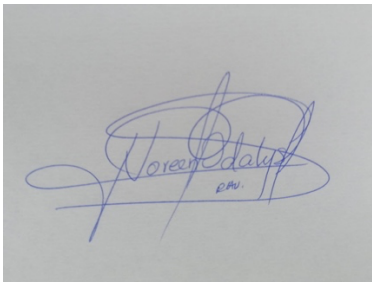
AUTOR

Tulcán, abril de 2022

AUTORÍA DE TRABAJO

El presente trabajo de titulación constituye requisito previo para la obtención del título de **Ingeniero** en la Carrera de computación de la Facultad de Industrias Agropecuarias y Ciencias Ambientales

Yo, Rosero Acosta Noreen Odalys con cédula de identidad número 1726169335 declaro: que la investigación es absolutamente original, auténtica, personal y los resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.



f.....

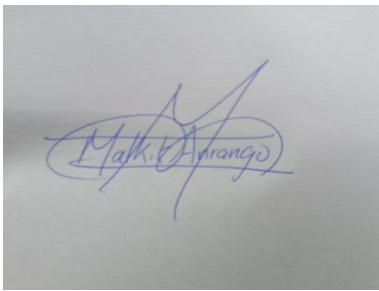
Rosero Acosta Noreen Odalys

AUTORA

Tulcán, abril de 2022

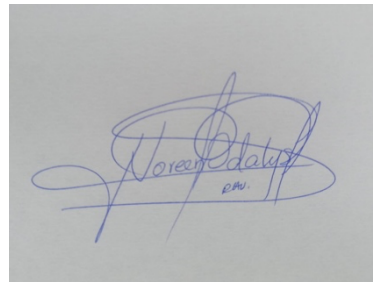
ACTA DE CESIÓN DE DERECHOS DEL TRABAJO DE TITULACIÓN

Nosotros, Anrango Oyagata Malkik Salvador y Rosero Acosta Noreen Odalys declaramos ser autores de los criterios emitidos en el trabajo de investigación: “Agenda digital personalizada para la organización de actividades académicas” y eximimos expresamente a la Universidad Politécnica Estatal del Carchi y a sus representantes legales de posibles reclamos o acciones legales.



f.....

Anrango Oyagata Malkik Salvador
AUTOR



f.....

Rosero Acosta Noreen Odalys
AUTORA

Tulcán, abril de 2022

DEDICATORIA

Este trabajo de investigación está dedicado a la Pachamama, Pachakamak, al tayta, Imbabura, a mi padre Umasapak Kishpichik, a mi madre Urpiku Oyagata quienes plantaron y cuidaron las semillas de la sabiduría, a los que murieron y a los que nacieron, a los que se rindieron y a los que se levantaron, a los que vencieron el miedo y se hicieron eternos, a los que se sacrificaron y a los que hicieron historia.

Shuk shunkulla - shuk maquilla - shuk yuyaila - shuk shimilla - kaypimi kanchik - kaypimi kashun - kaypimi kakrinchik runakunaka.

Un solo sentimiento, una sola mano de fuerza, un solo pensamiento, un solo lenguaje, aquí estamos, aquí estaremos y aquí volveremos a estar los Runa-s.

Malkik Anrango

Dedico este trabajo a mi madre, Susana, que con su infinito amor y esfuerzo impulsó cada uno de mis sueños, a mi tío Gustavo por enseñarme que todo puede ser posible incluso cuando no sepas como lograrlo, a mi padre y hermanos por su confianza, a mi "Mimita" por su dedicación y entrega, a los seres amados que fueron separados de mi por la muerte, pero el amor y su recuerdo nos mantienen unidos, a él, por compartir su tiempo, amor, risas y metas, a todos quienes en este camino ayudaron a revelar a esta mujer responsable, atenta y valiente.

Noreen Rosero

AGRADECIMIENTO

Agradezco a la Universidad Politécnica Estatal del Carchi por darme la oportunidad de superarme académica y profesionalmente, también a la vida por haber puesto siempre a las personas indicadas, y amables que a más de haberme ofrecido su amistad me motivaron a seguir esforzándome. A mi padre, a mi madre, a mis hermanos, hermana y primos quienes me ayudaron a volar por el cosmos

Tukuylla mashikunata ñuka muskuypi paktalla purishkamanta. Y a todos los que se sumaron y me ayudaron a seguir caminando con fuerza.

Yupaychani.

Malkik Anrango

Agradezco a la Universidad Politécnica Estatal del Carchi que me abrió las puertas a un mundo completamente nuevo, por enseñarme de lo que soy capaz y permitirme conocer amigos, compañeros y docentes que marcaron mi vida. A mi familia por sus innumerables muestras de apoyo incondicional durante todo este proceso, a mi abuelito, que en algún lugar de la eternidad está celebrando este logro del que soñaba ser parte. Mi gratitud es todo lo que tengo para devolver por los momentos compartidos.

Gracias vida, por lo que pones, por lo que quitas y por lo que acomodas.

Noreen Rosero

ÍNDICE DE CONTENIDOS

RESUMEN	17
INTRODUCCIÓN	19
I. PROBLEMA	20
1.1. Planteamiento del problema	20
1.2. Formulación del problema	23
1.3. Justificación	23
1.4. Objetivos y preguntas de investigación	25
1.4.1. Objetivo General	25
1.4.2. Objetivos Específicos	25
1.4.3. Preguntas de Investigación	25
II. FUNDAMENTACIÓN TEÓRICA	26
2.1. Antecedentes investigativos	26
2.2. Marco teórico	28
2.2.1 Aplicaciones Informáticas	28
2.2.1.1 Agendas digitales	28
2.2.1.2 Software genérico	30
2.2.1.3 Software a medida	30
2.2.1.4 Entornos virtuales de aprendizaje (EVA)	31
2.2.1.5 Aplicaciones relacionadas con la educación	32
2.2.1.6 Aplicación informática para actividades académicas	36
2.2.2. Actividades Estudiantiles	37
2.2.2.1. Recursos digitales enfocados al alumnado	38
2.2.3. Metodologías de desarrollo de software	39
2.2.3.1 Metodologías tradicionales	40
2.2.3.2 Metodologías ágiles	41
2.2.3.3. Comparación entre metodologías tradicionales y ágiles.	41

2.2.3.4. Valores y Objetivos de la metodología XP.	43
2.2.3.5. Fases de la metodología XP	43
2.2.3.6. Herramientas de la metodología XP	45
2.2.3.7. Roles XP	48
2.2.4. Herramientas tecnológicas de desarrollo	49
2.2.4.1. Entornos de desarrollo	49
2.2.4.2. Lenguajes de programación	51
2.2.4.3 Flutter	53
2.2.4.3. Framework	54
2.2.4.4. Librerías	54
2.2.4.5. Tendencias Tecnológicas	55
2.2.5. Inteligencia Artificial	57
2.2.5.1 Algoritmos de Inteligencia Artificial	58
2.2.5.2 Herramientas de desarrollo IA	63
2.2.6 Bases de datos	65
2.2.6.1 Sistema Gestor de Base de Datos (SGBD)	65
2.2.6.2 SGBD Relacionales	66
2.2.6.3 SGBD No Relacionales	67
2.2.6.4. Firebase	68
2.2.7 Visión artificial	69
2.2.8 OCR	70
2.2.9 API	71
2.2.10. Heroku	71
III. METODOLOGÍA	72
3.1. Enfoque metodológico	72
3.1.1. Enfoque	72
3.1.1.1. Enfoque Cualitativo	72

3.1.1.2. Enfoque Cuantitativo	72
3.1.2. Tipo de Investigación	73
3.1.2.1. Investigación Documental	73
3.1.2.2. Investigación descriptiva	73
3.2. Idea a defender	73
3.3. Definición y Operacionalización de variables	74
3.3.1. Definición de Variables	74
3.3.2. Operacionalización de variables	74
3.4. Métodos utilizados	76
3.4.1 Métodos	76
3.4.1.1 Método Analítico – Sintético	76
3.4.1.2 Método Deductivo – Inductivo	76
3.5. Análisis Estadístico	77
IV. RESULTADOS Y DISCUSIÓN	79
4.1 Resultados	79
4.1.1. Resultados de la encuesta	79
4.2 PROPUESTA	89
4.2.1. Estudio de Factibilidad	90
4.2.1.1. Factibilidad Organizacional	90
4.2.1.2. Factibilidad Técnica	90
4.2.1.3. Factibilidad Económica	91
4.2.1.4. Factibilidad Operativa	92
4.2.2. Metodología XP	93
4.2.2.1. Fase de planificación	93
4.2.2.2. Fase de diseño	116
4.2.2.3. Fase de codificación	125
4.2.2.4. Fase de pruebas	152

4.2.2.5. Resumen de iteraciones	160
4.3 DISCUSIÓN	160
V. CONCLUSIONES Y RECOMENDACIONES	163
5.1. Conclusiones	163
5.2 Recomendaciones	164
VI. REFERENCIAS BIBLIOGRÁFICAS	166
VII. ANEXOS	171

ÍNDICE DE FIGURAS

Figura 1 Estadísticas de hogares que poseen computador de escritorio y portátiles entre 2012 y 2018	34
Figura 2 Estadística de las personas que tienen Smartphone por edades	34
Figura 3 Estadística de personas que han usado internet en los últimos 12 meses.	34
Figura 4 Estadística de razones de uso de internet por áreas urbana, rural y nacional.....	35
Figura 5 Evolución del porcentaje de aulas virtuales (Aplicación informática) abiertas en el Grado en Matemáticas de la UPV/EH	36
Figura 6 Metodología XP o Programación Extrema	44
Figura 7 IDE NetBeans, ventana principal.....	50
Figura 8 Lista de IDE's	50
Figura 9 Los lenguajes de programación más famosos en el mundo por TIOBE en julio 2020 y 2019.....	52
Figura 10 Los lenguajes de programación más famosos en el mundo por PYPL en el año 2020	53
Figura 11 Habilitadores Tecnológicos.....	57
Figura 12 Tecnologías Emergentes	57
Figura 13 Estructura de la red neuronal artificial	60
Figura 14 Funciones de activación más utilizadas	60
Figura 15 Estructura, eje y dimensión de Arreglos o Arrays	64
Figura 16 Ejemplo de base de datos relacional	66
Figura 17 Estructura relacional (tablas) y no relacional (documentos).....	68
Figura 18 Diseño de Base de Datos.....	116

Figura 19 Inicio de Sesión	116
Figura 20 Login	117
Figura 21 Registro	117
Figura 22 Menú lateral	118
Figura 23 Resumen	118
Figura 24 Noticias	119
Figura 25 Comunidad	119
Figura 26 Recordatorios	120
Figura 27 Horario de Clases	120
Figura 28 Calendario	121
Figura 29 Herramientas	121
Figura 30 Caso de Uso Login	122
Figura 31 Caso de Uso Resumen	122
Figura 32 Casos de Uso Noticias	123
Figura 33 Caso de Uso Comunidad	123
Figura 34 Caso de Uso Recordatorios	124
Figura 35 Caso de Uso Calendario	124
Figura 36 Caso de Uso Horario de Clases	124
Figura 37 Caso de Uso Menú Lateral	125
Figura 38 Estructura	126
Figura 39 Estructura Interfaz	126
Figura 40 Controlador	126
Figura 41 Elementos Rutas	127
Figura 42 Main	128
Figura 43 Autenticación y Registro del Software	128
Figura 44 Resumen	129
Figura 45 Noticias	129
Figura 46 Comunidad	130
Figura 47 Recordatorios	130
Figura 48 Recordatorios	131
Figura 49 Calendario	131
Figura 50 Cuaderno digital	132
Figura 51 Asignaturas	132
Figura 52 Docentes	133

Figura 53 Modelo Usuario.....	133
Figura 54 Modelo Recordatorio	134
Figura 55 Modelo Privacidad	134
Figura 56 Modelo Post o Publicación.....	135
Figura 57 Modelo Noticia.....	135
Figura 58 Modelo Notas o cuaderno digital	136
Figura 59 Modelo Like Post	136
Figura 60 Modelo Horario	137
Figura 61 Modelo Docente	137
Figura 62 Modelo Clases	138
Figura 63 Modelo Asignatura.....	138
Figura 64 Modelo Apariencia.....	139
Figura 65 Usuario	139
Figura 66 Preprocesamiento de Imagen	141
Figura 67 Detección de bloques	142
Figura 68 Ejemplo de datos de entrenamiento	143
Figura 69 Ejemplo de datos de entrenamiento	145
Figura 70 Función de activación sigmoid.....	149
Figura 71 Modelo de red neuronal Norkik	150
Figura 72 Carga computacional OpenCV	152

ÍNDICE DE TABLAS

Tabla 1 Comparación entre metodologías ágiles y tradicionales	42
Tabla 2 Ficha de historia de usuario	45
Tabla 3 Tarjeta CRC (Clase – Responsabilidades - Colaborador)	46
Tabla 4 Tarea de usuario o Tarea de ingeniería.....	47
Tabla 5 Ficha de prueba de aceptación.....	48
Tabla 6 Librerías en función del lenguaje	55
Tabla 7 Estructura relacional (tablas) y no relacional (documentos)	74
Tabla 8 Recursos Software y Hardware	91
Tabla 9 Factibilidad Económica	92
Tabla 10 Roles del proyecto	93
Tabla 11 Estimación de tiempo	94
Tabla 12 Historia de usuario 1	95

Tabla 13 Historia de usuario 2.....	96
Tabla 14 Historia de usuario 3.....	96
Tabla 15 Historia de usuario 4.....	97
Tabla 16 Historia de usuario 5.....	97
Tabla 17 Historia de usuario 6.....	98
Tabla 18 Historia de usuario 7.....	99
Tabla 19 Historia de usuario 8.....	99
Tabla 20 Historia de usuario 9.....	100
Tabla 21 Historia de usuario 10.....	100
Tabla 22 Historia de usuario 11.....	101
Tabla 23 Historia de usuario 12.....	102
Tabla 24 Historia de usuario 13.....	102
Tabla 25 Historia de usuario 14.....	102
Tabla 26 Historia de usuario 15.....	103
Tabla 27 Historia de usuario 16.....	103
Tabla 28 Tarea de usuario 1	104
Tabla 29 Tarea de usuario 2	104
Tabla 30 Tarea de usuario 3	105
Tabla 31 Tarea de usuario 4	105
Tabla 32 Tarea de usuario 5	105
Tabla 33 Tarea de usuario 6	105
Tabla 34 Tarea de usuario 7	106
Tabla 35 Tarea de usuario 8	106
Tabla 36 Tarea de usuario 9	106
Tabla 37 Tarea de usuario 10	107
Tabla 38 Tarea de usuario 11	107
Tabla 39 Tarea de usuario 12	107
Tabla 40 Tarea de usuario 13	108
Tabla 41 Tarea de usuario 14	108
Tabla 42 Tarea de usuario 15	108
Tabla 43 Tarea de usuario 16	108
Tabla 44 Tarea de usuario 17	109
Tabla 45 Tarea de usuario 18	109
Tabla 46 Tarea de usuario 19	109

Tabla 47 Tarea de usuario 20	110
Tabla 48 Tarea de usuario 21	110
Tabla 49 Tarea de usuario 22	110
Tabla 50 Tarea de usuario 23	111
Tabla 51 Tarea de usuario 24	111
Tabla 52 Tarea de usuario 25	111
Tabla 53 Tarea de usuario 26	111
Tabla 54 Tarea de usuario 27	112
Tabla 55 Tarea de usuario 28	112
Tabla 56 Estimación de tareas de usuario	112
Tabla 57 Plan de entrega del proyecto.....	115
Tabla 58 Carga Computacional OCR.....	140
Tabla 59 Carga Computacional Redes Neuronales	146
Tabla 60 Iteraciones o épocas de entrenamiento	147
Tabla 61 Prueba de Aceptación 1	152
Tabla 62 Prueba de Aceptación 2	153
Tabla 63 Prueba de Aceptación 3	153
Tabla 64 Prueba de Aceptación 4	154
Tabla 65 Prueba de Aceptación 5	154
Tabla 66 Prueba de Aceptación 6	155
Tabla 67 Prueba de Aceptación 7	155
Tabla 68 Prueba de Aceptación 8	156
Tabla 69 Prueba de Aceptación 9	156
Tabla 70 Prueba de Aceptación 10	157
Tabla 71 Prueba de Aceptación 11	157
Tabla 72 Prueba de Aceptación 12	158
Tabla 73 Prueba de Aceptación 13	158
Tabla 74 Prueba de Aceptación 14	159
Tabla 75 Prueba de Aceptación 15	159
Tabla 76 Prueba de Aceptación 16	159
Tabla 77 Resumen de iteraciones	160

ÍNDICE DE ANEXOS

Anexo 1 Acta de predefensa de Malkik Anrango.....	171
Anexo 2 Acta de predefensa de Noreen Rosero	172
Anexo 3 Certificado del abstract por parte de idiomas	173
Anexo 4 Informe de Turnitin.....	175
Anexo 5 Certificado de finalización del proyecto	176
Anexo 6 Encuesta a los estudiantes de Computación.....	177
Anexo 7 Manual de Usuario Norkik.....	181

RESUMEN

La presente investigación denominada “Agenda digital personalizada para la organización de actividades académicas” profundizó en el estudio del uso de tecnologías en la educación. El objetivo principal fue desarrollar una agenda digital personalizada con herramientas de inteligencia artificial, para la organización de actividades académicas en la carrera de Computación de la Universidad Politécnica Estatal del Carchi, para dar cumplimiento a esta meta se planteó un enfoque mixto en conjunto con la investigación documental y descriptiva permitieron recolectar datos a través de la aplicación de una encuesta a los estudiantes y una entrevista a la Directora de la Carrera, dando como resultado información detallada de las herramientas tecnológicas utilizadas en su proceso académico. A partir de los resultados conseguidos se desarrolló un prototipo de aplicación móvil para la organización de actividades académicas, en la realización de la propuesta se empleó la metodología ágil XP donde se definió los instrumentos de más utilidad para el desarrollo del proyecto. Finalmente, en el ámbito técnico se establece un sistema desarrollado con Flutter como Framework, una base de datos NoSQL en Firebase, el uso de estas tecnologías fusionado con la metodología XP permitió que el ciclo de vida del software sea rápido y confiable, y se obtuvo como referencia la documentación generada que puede ser reutilizable en proyectos futuros o en trabajos de implementación.

Palabras clave: Agenda digital, actividades académicas, aplicación móvil

ABSTRACT

The present research work named "Personalized digital agenda for the organization of academic activities" made a deep study on the use of technologies in education. The main goal was to develop a personalized digital agenda with artificial intelligence tools for the organization of academic activities in the Computing career at Universidad Politécnica Estatal del Carchi. To fulfill this aim, a mixed approach was proposed as well as the documentary and descriptive research. They allowed to gather data through the application of a survey addressed to students and an interview applied to the Director of the Career. Therefore, it was obtained detailed information on the technological tools used in the academic process. Basing on the results, a mobile application prototype was developed for the organization of academic activities. On the other hand, in the proposal the agile XP methodology was used where the most useful instruments for the development of the project were defined. Finally, in the technical field, a system developed with Flutter as Framework is established, a NoSQL database in Firebase. The use of these technologies merged with the XP methodology allowed the software life cycle to be fast and reliable. Besides, it was obtained as a reference the generated documentation that can be reused in future projects or in implementation work.

Keywords: Digital agenda, academic activities, mobile application.

INTRODUCCIÓN

Actualmente la tecnología es de vital importancia para la sociedad, debido a que están inmersas en la vida cotidiana de las personas y satisfacen sus necesidades. El impacto de la tecnología en la educación no ha sido menor, durante la pandemia se han superado varios desafíos que impulsaron el uso y desarrollo de nuevas soluciones informáticas que aportan al proceso académico de los estudiantes.

La Carrera de Computación de la Universidad Politécnica Estatal del Carchi se caracteriza por la formación de profesionales capaces de generar soluciones tecnológicas innovadoras que contribuyen al desarrollo de la región, los estudiantes de la comunidad universitaria tienen a su disposición diferentes herramientas tecnológicas para acompañar sus actividades académicas, estas aplicaciones conllevan varias desventajas, pues, al ser genéricas, el estudiante debe adaptarse a las funcionalidades que el software ya tiene implementado, además, no todas las funcionalidades que ofrecen estos programas son adecuados para un estudiante.

Por lo tanto, el objetivo principal de este estudio fue desarrollar una agenda digital personalizada con herramientas de inteligencia artificial, para la organización de actividades académicas en la carrera de Computación de la Universidad Politécnica Estatal del Carchi. La importancia de esta investigación se fundamenta en la adquisición de conocimiento sobre las ventajas que brinda la tecnología en el proceso académico del estudiante.

El enfoque mixto de investigación permitió analizar y dimensionar las variables de estudio sobre las agendas digitales y las actividades académicas. Se estableció una investigación documental y descriptiva que permitieron recolectar datos para el desarrollo de una propuesta enfocada al impulso de la solución tecnológica, la misma que fue guiada por los modelos de metodología de software, dando lugar a una solución tecnológica eficaz.

I. PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

En la última década es evidente que las aplicaciones informáticas gobiernan el diario vivir de las personas, de modo que su utilización se ve reflejada en las computadoras y celulares inteligentes. Dicha tecnología está inmersa en diferentes aspectos, tales como educación, entretenimiento, ocio, profesión, etc. De manera que si se clasifica según su utilización son muchos factores para considerar. En el caso de esta investigación se enfocó al aspecto educativo, es decir, a las aplicaciones informáticas en relación con la organización de actividades académicas, o también denominadas agendas digitales.

En este sentido, es claro que existen diversas aplicaciones de esta índole en los repositorios de App's. Pero, a pesar de que existen, estas no sustentan los requerimientos de los alumnos, ya sea porque no existe alguna investigación o simplemente no se enfoca 100% a la organización de actividades académicas, por lo que en consecuencia las aplicaciones disponibles no tienen las funcionalidades que un estudiante promedio requiere.

Otro de los aspectos a considerar es la relación entre la universidad y las aplicaciones móviles, puesto a que se puede evidenciar la existencia de aplicaciones genéricas en varios repositorios de Apps (Google Play – Apple Store – Ovi Store). De modo que sus respectivas funciones no cumplen con lo requerido por parte de los estudiantes en relación con su respectiva universidad.

Por otra parte, Cisneros (2017) asegura que el software genérico, aunque es útil no siempre es la mejor opción, la relación calidad precio no es buena, un software comercial es bastante bajo en costos con respecto a el software a medida, pero las limitaciones del software encarecen al producto, el usuario debe adaptarse a lo que la aplicación le ofrece, provocando que generalmente se use solo el 15% de las funcionalidades.

El periodista y comunicador social Moncada (2019) en su sitio blog menciona que el Observatorio Periodístico de la Fundación Universitaria Horizonte, UniHorizonte, desarrolló una encuesta situada en Colombia a 335 estudiantes de diferentes

instituciones de educación superior sobre el uso que dan los estudiantes a las aplicaciones que están instaladas en su celular y su relación con su vida personal y académica. En dicha encuesta lista las siguientes aplicaciones móviles (Apps) más usadas: WhatsApp, Instagram, Facebook, y YouTube, y concluye que los teléfonos celulares son utilizados más para entretenimiento y muy poco para realizar y organizar tareas o trabajos referentes a las carreras que se estudian en determinadas universidades.

El hecho de que las aplicaciones de ocio sean más utilizadas puede posiblemente deberse a diversos factores como, la deficiente personalización, insuficientes funcionalidades, o puede considerarse que la aplicación no cumple con las necesidades que el usuario requiere en función de sus actividades a las que dedica tiempo. Lo que a su vez conlleva que los mismos programas sean utilizados para el aspecto académico u organización de actividades.

Además, es evidente que los teléfonos inteligentes son una realidad que actualmente todos pueden sentir, las mismas son usadas para facilitar diversas actividades de ocio, profesionales y hasta educativas. Si bien estas nuevas tecnologías tienen mucho que ofrecer a sus usuarios, no siempre se aprovechan al máximo, llegando al punto en que las aplicaciones móviles sean llamadas “distractores” en algunos casos (Sangacha et al., 2019).

En Ecuador uno de los principales problemas es que los estudiantes ven a los Smartphone como dispositivos de entretenimiento y no como una herramienta para mejorar su aprendizaje (Jiménez, 2016). De acuerdo con Ricaurte (2018) el 15 de enero de 2018, las aplicaciones más descargadas en Ecuador fueron: WhatsApp, Messenger, Instagram, Facebook, Facebook Lite, Wish, Snapchat entre otras.

Aunque varias universidades en Quito, la capital del Ecuador, como la Universidad Central del Ecuador, la Universidad de las Fuerzas Armadas y la Pontificia Universidad Católica del Ecuador se han podido apreciar varios aportes en el área de ingeniería de sistemas, especialmente con aplicaciones móviles para asistir al estudiante universitario, estas se enfocan en una área bastante lejana a la académica, no se puede olvidar que la tecnología avanza de forma rápida y ya existen nuevas

herramientas que pueden ser utilizadas para crear aplicaciones intuitivas que mejoran la experiencia del usuario y pueden aportar de forma invaluable a la vida académica de los universitarios. Además, las necesidades de los estudiantes universitarios al hablar de aplicaciones móviles que aporten a su rendimiento académico se encuentran dispersas en diferentes herramientas.

En la Universidad Politécnica Estatal del Carchi, ubicada en la ciudad de Tulcán, a sus 16 años de vida institucional al igual que muchas universidades del país ha aportado con soluciones tecnológicas para la provincia del Carchi.

“Parte de las soluciones innovadoras que se desarrollan en la universidad, no abastecen el 100% a las necesidades que surgen, de tal modo que parte de estas tecnologías no son destinadas a los estudiantes”. (G. Arcos, comunicación personal, 19 de junio de 2020). De acuerdo con la directora de la carrera de Computación, el departamento encargado de desarrollar las soluciones tecnológicas se enfoca precisamente a las necesidades primordiales que surgen, como, por ejemplo, los requerimientos de conectividad, soporte técnico de los sistemas vigentes, los sistemas de Base de Datos, entre otros. Por lo que existe un desenfoco hacia las necesidades del estudiante. Obligando al mismo a utilizar programas genéricos que de una manera u otra los ayuden en ciertas actividades académicas. Tales como WhatsApp, Messenger, Telegram, etc.

Es decir, a pesar de que el estudiante de la UPEC tiene herramientas que puede manejar y de las que puede sacar provecho para la organización de las actividades durante su vida académica, estas no satisfacen sus necesidades al 100%.

Aun existiendo tecnologías que pueden hacer aportes valiosos al momento de organizar actividades educativas algunos grupos de estudiantes se ven más atraídos por la forma tradicional de organización, es decir escribir sus tareas en un cuaderno o agenda física, en la que su información corre muchos riesgos que pueden ocasionar problemas al estudiante al momento de presentar alguna actividad.

Generalmente las agendas impresas son rígidas y no permiten ser personalizadas, en ocasiones habrá hojas en blanco que se desperdiciarán, además este tipo de agendas no son precisamente económicas, estas pueden llegar a ser costosas especialmente si

de diseños exclusivos se trata. Una vez termine el año calendario no se pueden reutilizar obligando al estudiante de la UPEC a invertir en otra.

Si bien es cierto que existen programas que organicen actividades inclusive utilizando inteligencia artificial, los mismos se los considera genéricos, puesto a que el software va dirigido a un segmento de clientes demasiado amplio. Obligando a ciertos usuarios como los estudiantes de la Computación utilizar varios programas.

Otro aspecto por considerar es el hecho de que en la universidad no existe una investigación que estudie las necesidades del estudiante, enfocadas en la organización de actividades académicas, y tal afirmación se lo puede fundamentar a través de la búsqueda en los repositorios digitales de la institución educativa.

1.2. FORMULACIÓN DEL PROBLEMA

La subutilización de una agenda digital personalizada es generada por el desaprovechamiento de herramientas con inteligencia artificial, afectando a la organización de las actividades académicas, en la carrera de Computación de la Universidad Politécnica Estatal del Carchi, en el año 2021.

1.3. JUSTIFICACIÓN

El uso de la tecnología en la educación es de gran importancia en el desarrollo personal y profesional de una persona, en esta pandemia la tecnología ha desempeñado un papel muy importante al ayudar a superar muchos de los retos que por causa del virus no se han podido cumplir según lo planificado. Esta situación impulsó el uso y creación de nuevas aplicaciones que han sido de utilidad para muchos usuarios, especialmente estudiantes por todo el mundo que se han adaptado a una modalidad virtual, para continuar con sus estudios.

Actualmente las tiendas de aplicaciones tienen una gran cantidad de ofertas en cuanto a aplicaciones educativas y de organización (como agendas) se refiere, pero, estas aplicaciones conllevan varias desventajas, pues, al ser genéricas, el usuario debe adaptarse a las funcionalidades y metodologías que el software ya viene implementado, además, no todas las funcionalidades que ofrecen estos programas son

adecuadas para una persona, aunque el costo es bajo la relación costo - calidad no es buena.

El presente proyecto se realiza con el fin de explotar las ventajas que tienen los softwares a medida en beneficio del usuario adaptándose a sus necesidades, esto hace que la aplicación evolucione conjuntamente con el usuario ganando eficiencia, Es así como conjuntamente, la tecnología y la educación podrán optimizar el tiempo del usuario, promoviendo a la institución y al estudiante a estar en mejora continua.

En cuanto a los beneficiarios directos serán los estudiantes de la carrera de Computación de la Universidad Politécnica Estatal del Carchi, puesto a que poseerán una herramienta informática que permita acompañar en todo el transcurso de la vida académica dentro de la universidad. En este contexto, el usuario podrá acceder a diversas funcionalidades que la solución pueda brindar, tales como: horarios de clase y tutorías de docentes, noticias actualizadas de la institución, cartelera cultural y deportiva, recordatorios, etc. Todo esto contenido en una agenda digital intuitiva y personalizada que, cumpla con las necesidades del estudiante, a través del estudio de sus requerimientos.

Mientras que los beneficiarios indirectos, corresponden a la misma institución, así como los administrativos, y docentes, las cuales, al existir una herramienta informática, que cubra las necesidades del beneficiario, mantendrán informado de manera más directa a sus estudiantes, posibilitando a que la información generada, llegue de manera oportuna, cómoda y eficaz. Además de que el proyecto de investigación pretende influir de manera indirecta en el desempeño del estudiante con la institución de educación superior.

Otro de los aspectos a considerar, es la factibilidad, puesto a que el desarrollo del proyecto es posible debido a que existe accesibilidad a recursos humanos (los investigadores), acceso a la información, resaltando la innovadora solución tecnológica de estudiantes para estudiantes.

1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN

1.4.1. Objetivo General

Desarrollar una agenda digital personalizada con herramientas de inteligencia artificial, para la organización de actividades académicas en la carrera de Computación de la Universidad Politécnica Estatal del Carchi.

1.4.2. Objetivos Específicos

- Fundamentar bibliográficamente las agendas digitales y la organización de actividades académicas, utilizando fuentes de información científicas, para la sustentación de la investigación.
- Determinar la metodología de desarrollo de software para la creación de una agenda digital personalizada.
- Proponer una agenda digital personalizada para la gestión de apuntes, tareas y recordatorios mediante el uso de herramientas de inteligencia artificial.

1.4.3. Preguntas de Investigación

¿Cómo se define las agendas digitales personalizadas en relación con la organización de las actividades académicas?

¿Cuáles son las principales actividades académicas del estudiante que deben considerarse para determinar una metodología de desarrollo?

¿Qué tipo de agenda digital personalizada permitiría la organización de las actividades académicas?

¿Qué algoritmos de Inteligencia Artificial se adecuarían al comportamiento del usuario en relación con la organización de las actividades académicas, de manera que aumenten la personalización de la agenda digital?

II. FUNDAMENTACIÓN TEÓRICA

2.1. ANTECEDENTES INVESTIGATIVOS

Castro (2018) estudiante de la Universidad Central del Ecuador ubicada en la ciudad de Quito en su trabajo de grado titulado “PROPUESTA DE UN SISTEMA DE ACOMPAÑAMIENTO A ESTUDIANTES UNIVERSITARIOS BASADO EN TECNOLOGIAS DE INFORMACIÓN Y COMUNICACIÓN (TIC’S)”, en el que tiene como objetivos proponer un modelo de acompañamiento que sea de utilidad para el departamento de bienestar universitario, que permita conocer las deficiencias que presente el estudiante, y canalizar su ayuda, analizar los sistemas de acompañamiento y seguimiento a estudiantes universitarios a nivel nacional e internacional y finalmente adelantar programas y acciones de prevención ante los riesgos implícitos en la vida universitaria que puedan afectar el desempeño académico de los estudiantes. En este proyecto investigativo se quiere comprobar que un modelo de acompañamiento al estudiante universitario basado en las TIC’s, mejorará el desempeño académico del estudiante universitario en las Instituciones de Educación Superior del Ecuador concluyendo que la mayoría de las universidades no tiene sistemas de acompañamiento a estudiantes universitarios, determinando una causa muy significativa en el bajo nivel de educación.

Zurita (2020) en su investigación “Sistema web para la gestión académica y administrativa de Empresa de Capacitación Profesional Dienav”, se plantearon como objetivos analizar las actividades y gestión académica en conjunto con sus problemas, diseñar un proceso de control con la incorporación de los requerimientos necesarios, además de codificar e implementar los procesos, del Sistema de Gestión Académica y Administrativa de manera modular y por iteraciones a fin de propiciar entregables bajo funcionalidades ya predefinidas por las partes. Al final de la investigación concluyeron que se cumplió con el análisis de los procesos que se interrelacionan entre sí para proveer servicio a la comunidad educativa, cumpliendo con las expectativas de los usuarios finales. De la misma manera la investigación se basó en una metodología de desarrollo ágil denominada XP, lo que permitió el diseño de la solución web con la entrega de soluciones funcionales en los procesos de manera incremental por iteraciones, satisfaciendo los requerimientos más urgentes del cliente.

El trabajo citado aporta a la investigación debido a que da a contextualizar las actividades, gestión y problemas académicos que se puede presentar en la parte estudiantil. Por otra parte, la metodología de desarrollo seleccionada por el investigador hace referencia a su efectividad, en cuanto a cumplir los requerimientos del cliente o usuario final.

Moran et al. (2019) publicaron un artículo científico denominado “Implementación del Sistema de Gestión Académica (SGA) en el Instituto Tecnológico Superior Juan Bautista Aguirre”, con su objetivo principal como describir la implementación del sistema de gestión académico en el Instituto Tecnológico Superior Juan Bautista, en donde el desarrollo del software partió por la aplicación de la metodología de desarrollo SCRUM, la cual establecen roles, tiempos de entrega al iniciar el proyecto. Por otra parte, en el levantamiento de información los autores recolectaron datos antes y después del desarrollo, utilizando técnicas de investigación, tales como observaciones, y entrevistas tanto a docentes, administrativos y estudiantes, lo que a su vez permitieron establecer los requerimientos del software y los tiempos de entrega. La investigación concluye en 4 factores muy importantes, las cuales parten desde el conocimiento de los procesos académicos y administrativos de la institución educativa, el ciclo de vida del sistema informático, identificación de la metodología de desarrollo hasta lo que respecta en la dominación de los paradigmas y lenguajes de programación en conjunto con una base de datos. De acuerdo con los autores citados la aplicación y conocimiento de las metodologías de desarrollo, así como también el conocimiento de paradigmas y lenguajes de programación, son factores clave para un proyecto de desarrollo. Tomando en cuenta los beneficiarios directos e indirectos. Dando como resultado una propuesta integral, dinámica y eficiente. En la investigación los puntos importantes como la metodología de desarrollo, gestión estudiantil, desarrollo de una propuesta, son puntos clave que se consideraron de suma importancia en la parte teórica y práctica.

2.2. MARCO TEÓRICO

2.2.1 Aplicaciones Informáticas

El surgimiento de las aplicaciones informáticas ha cambiado el mundo marcando una revolución en la calidad de vida de las personas. Este tipo de herramientas se las puede evidenciar, desde el simple hecho de manejar los dispositivos móviles, de escritorio y de trabajo.

El concepto de aplicaciones informáticas se la presenta como un software o programa informático que facilita trabajos o tareas que tienen cierto grado de complejidad. Es decir, que las herramientas informáticas son programas que de un cierto modo son usadas para realizar tareas o actividades de manera sencilla (Yadira, 2019).

Tomando en consideración la definición de aplicaciones informáticas, estas tienen ciertas características o clasificaciones, esto dependiendo de la orientación del desarrollo y usabilidad. Existen programas que son generalmente usados por todos los usuarios orientados a una temática o área haciendo que estas herramientas tengan una cantidad considerable de funcionalidades. Este tipo de software se lo identifica como genérico o comercial. A diferencia de los programas informáticos a medida en la que solo está enfocado a cumplir actividades o tareas necesarias y son desarrolladas para usuarios determinados u organizaciones específicas (SVDigital, 2020).

En lo que respecta a la investigación, se plantea el estudio de aplicaciones informáticas disponibles en el repositorio de programas, las cuales permitirán extraer información relevante en el comportamiento del usuario relacionándolo con el software, de manera que los mismos ayuden a llevar un registro detallado sobre las funcionalidades más relevantes, así como también las que requieren. De modo que el resultado sea la sinterización de las partes detalladas.

2.2.1.1 Agendas digitales

Según la Real Academia de la Lengua Española una agenda es un libro o dispositivo que permite mantener un orden de diferentes actividades, oficios o quehaceres de una persona durante un período de tiempo. Partiendo de este concepto la evolución de las

Tecnologías de la Información y Comunicación (TIC) han permitido ir más allá de lo físico generando ventajas para el usuario al momento de gestionar su tiempo con agendas digitales.

El Ministerio de Educación (2017) define a la agenda digital como un software que provee diferentes funciones orientadas a la gestión de tiempo, organización de tareas y actividades considerando a las tecnologías de la información y comunicación una oportunidad para la innovación y mejora de diferentes procesos.

La agenda digital es el punto clave de esta investigación, puesto a que el enfoque se realiza a estos programas informáticos, independientemente si es web, móvil o de escritorio. El resultado de esta investigación es la incorporación de los comportamientos del usuario en relación con la inteligencia artificial, requerimientos, funcionalidades, en la agenda digital, de modo que permita adquirir nueva información relevante con respecto a la aplicación ya sintetizada de los anteriores puntos mencionados.

Si bien es cierto que el término personalización en esta investigación es un aspecto clave, pues, este término en informática hace referencia al desarrollo de aplicaciones que estén acorde a las especificaciones del cliente, en base única al usuario. De modo que una agenda digital personalizada se la conceptualiza como una aplicación orientada a la organización de las actividades académicas, en el cual, se vea reflejado en resolver sus necesidades básicas como estudiante por medio de las tecnologías anteriormente mencionadas.

Como bien se menciona, la personalización tiene que ver con solventar las necesidades de los usuarios, sin embargo, existe su contra parte las cuales se construyen programas orientadas a un amplio número de usuarios, en donde se trata de resolver varias necesidades de manera genérica. Es el caso de los softwares genéricos y también su opuesto que se lo denomina software a medida.

2.2.1.2 Software genérico

El software genérico es desarrollado para resolver múltiples necesidades. Generalmente, el usuario llega a utilizar en torno al 15% de su funcionalidad, esto siempre depende del tipo de empresa y de la manera de trabajar de dicho usuario (Teknei, 2018).

Este tipo de software es diseñado para usarlo en diferentes entornos organizacionales, pero ante el gran problema de sensibilidad que presenta genera desafíos de usabilidad para el usuario, esto sucede al momento del diseño, pues es necesario enfocarse en un público general y no en casos específicos procurando cumplir con las tareas que el usuario necesita (Li & Nielsen, 2019).

Con esos antecedentes se destaca que el software genérico, aunque es útil no siempre es la mejor opción, la relación calidad precio no es buena, un software comercial es bastante bajo en costos con respecto a el software a medida, pero las limitaciones del software encarecen al producto, el usuario debe adaptarse a lo que la aplicación le ofrece sin adaptarse a sus necesidades.

2.2.1.3 Software a medida

El software a medida es diseñado conjuntamente con el cliente, por lo que el resultado final se adapta a la empresa u organización y su personal ofreciendo un diseño e implementación con un alto nivel de usabilidad (Teknei, 2018).

Este tipo de software tiene un diseño personalizado para una determinada empresa u organización lo que permite ganar eficiencia, tomando en cuenta que la relación calidad precio será excepcional.

De acuerdo con lo referenciado y la investigación, lo que respecta a medida es lo que diferencia de las aplicaciones genéricas ya existentes. Agregando además

funcionalidades que estén acorde a las necesidades o requerimientos del usuario en la carrera de computación de la Universidad Politécnica Estatal del Carchi.

Por otro lado, también se considera mencionar que existen aplicaciones informáticas orientadas a la educación, tal es el caso de los Entornos virtuales de aprendizaje o también denominados EVA.

2.2.1.4 Entornos virtuales de aprendizaje (EVA)

Un entorno virtual de aprendizaje o Virtual Learning Environment en inglés es un determinado lugar en el web estructurado por un conjunto de material informático que hace posible la interacción. Entre sus características está primero estar desarrollado con tecnología digital, ser asequible por medio del uso de Internet haciendo que sus contenidos sean de fácil acceso y ser un apoyo en el proceso de aprendizaje de cualquier persona (Rodríguez y Castillo, 2019)

Sin duda el uso de estas herramientas incentiva a la interacción en el proceso enseñanza-aprendizaje haciendo que el estudiante tome el papel protagónico e invitándolo a la participación, lo que genera que tanto docentes como estudiantes participen activamente de las diferentes actividades de aprendizaje, al constar con varios recursos que asisten y facilitan el proceso de aprendizaje en donde se incluye diferentes módulos para la gestión de actividades académicas, calendarios, evaluaciones, entre otros.

Rodríguez y Castillo (2019) en su libro “Entornos virtuales de aprendizaje posibilidades y retos en el ámbito universitario” destacan las siguientes acciones que se pueden llevar a cabo para la gestión y planificación en el proceso de aprendizaje:

- Permite realizar tareas de administración y gestión de nuestra asignatura
- Podemos publicar materiales y actividades
- Fomenta la comunicación multidireccional entre profesor-alumnos, alumnos-profesor y alumnos entre sí
- Permite la participación y la cooperación activa de todos los miembros del grupo para realizar tareas comunes

- Podemos realizar actividades interactivas
- Permite realizar una evaluación y seguimiento de los estudiantes.

Un EVA no solamente ayudan a la organización de actividades académicas, su uso va mucho más lejos promoviendo procesos de comunicación docente-estudiante, estudiante-docente y estudiante-estudiante creando un ambiente de trabajo óptimo y colaborativo en el que todos son partícipes rompiendo de forma impresionante las barreras de tiempo y espacio.

Esta herramienta es vista como una oportunidad que al ser combinada con la modalidad de enseñanza ofrece la optimización de la formación de los estudiantes, involucrándolos en su proceso de aprendizaje, mejorando su rendimiento académico y desarrollando habilidades para su vida profesional utilizando el acceso a esta plataforma tecnológica para fortalecer su proceso de formación (Piña y Torrealba, 2020).

2.2.1.5 Aplicaciones relacionadas con la educación

Es claro que las aplicaciones informáticas han impulsado en los últimos años la calidad de vida de los estudiantes, haciendo que estas herramientas sean integradas en los procesos de enseñanza y aprendizaje en los diferentes niveles de educación (Martins et al., 2019). Sin embargo, la efectividad de estos programas informáticos se ve reflejado en la utilización tanto por la parte docente como la del estudiante.

Dentro de la temática de aplicaciones informáticas, surge el término web (información en internet), en la que cualquier persona puede acceder a distintos tipos de información. En este sentido, la accesibilidad que tienen los estudiantes hacia la información va acorde con la tecnología, en la que el desarrollo tecnológico va en concordancia con la educación.

Guaña Moya, García Herrera, & Quinatoa Arequipa (2017) mencionan la sociedad, la web (aplicación informática) y la educación, en donde resalta primeramente los paradigmas sociales, como sociedad 1.0 (establece aprender haciendo, donde el sistema educativo preparaba trabajadores para la industrialización), sociedad 2.0 (gestión del conocimiento) y sociedad 3.0 (en donde alude al futuro inmediato). De la

misma manera se hace referencia a la web 1.0 (en la que la información era estática), web 2.0 (se incluye información dinámica como redes sociales, wikis, etc.) y web 3.0 (que alude a la inteligencia artificial, tecnología 3D, etc).

En este contexto, el autor enlaza los paradigmas de la sociedad con la web, de modo que como resultado surge los paradigmas de la educación: educación 1.0 donde los estudiantes aprenden unidireccionalmente del profesor, educación 2.0, en la que los estudiantes aprenden en comunidades colaborativas de aprendizaje y la educación 3.0, que alude a que los estudiantes no solo aprenden de las comunidades, sino también de la utilización de la web semántica y búsqueda de información inteligente, en la que se relaciona con la inteligencia artificial.

La educación 3.0, como bien lo menciona Guaña Moya et al. (2017), se caracteriza por sus componentes como la universidad 3.0, como lugar físico o virtual que genera, comparte y gestiona información en base a búsquedas inteligentes, los estudiantes 3.0, que hace referencia a individuos que transcribe, producen y adquiere información ya sea presencial o virtual, docentes 3.0, alude a individuos naturales o artificiales que guía o gestiona la información, utilizando herramientas tecnológicas, y entornos de prácticas y aprendizaje 3.0, que relacionan a campus presénciales o virtuales, con el apoyo de inteligencia artificial y visión 3D.

Es claro que la tecnología y la educación van estrechamente ligados, sin embargo, en el aspecto territorial entre países, existe variantes en el acceso a tecnologías como aplicaciones informáticas, Para este caso se toma como aspecto general a Ecuador, de acuerdo a las estadísticas generadas por la INEC (2018) , a continuación, se muestra lo siguiente:

Los hogares que tienen una computadora de escritorio supera el promedio a comparación de los que poseen computadoras portátiles, es claro que, que si se refiera al año 2020 el porcentaje aumenta de los hogares que poseen computadoras portátiles.

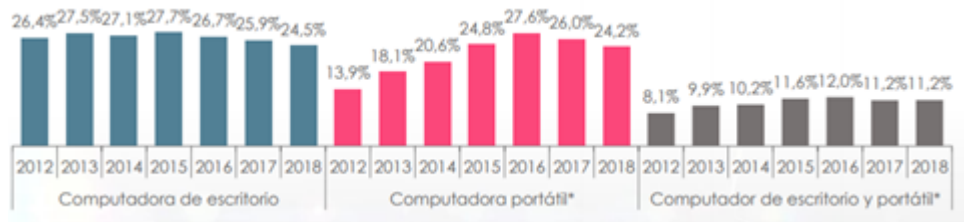


Figura 1. Estadísticas de hogares que poseen computador de escritorio y portátiles entre 2012 y 2018

Fuente: INEC (2018)

En cuanto a personas que poseen un teléfono inteligente con respecto a edades existe un mayor incremento entre el intervalo de edades de 16 – 24 y 25 – 34, es claro que esto se da debido a que es para el entretenimiento o educación.

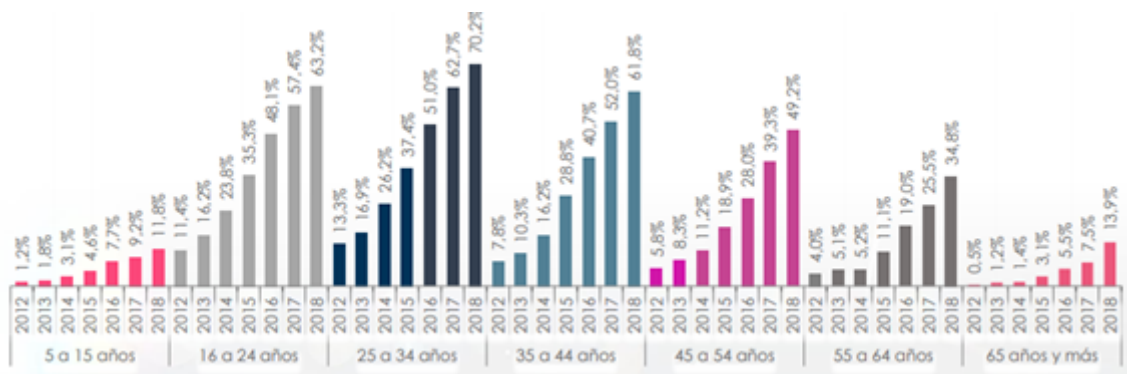


Figura 2. Estadística de las personas que tienen Smartphone por edades
Fuente: INEC (2018)

En cuanto, a personas que han utilizado internet en los últimos 12 meses podemos encontrar un alto porcentaje superior a la mitad en el año 2020, es evidente que el porcentaje es superior en lo urbano que, en lo rural, sin embargo, cabe resaltar que el porcentaje es considerable.

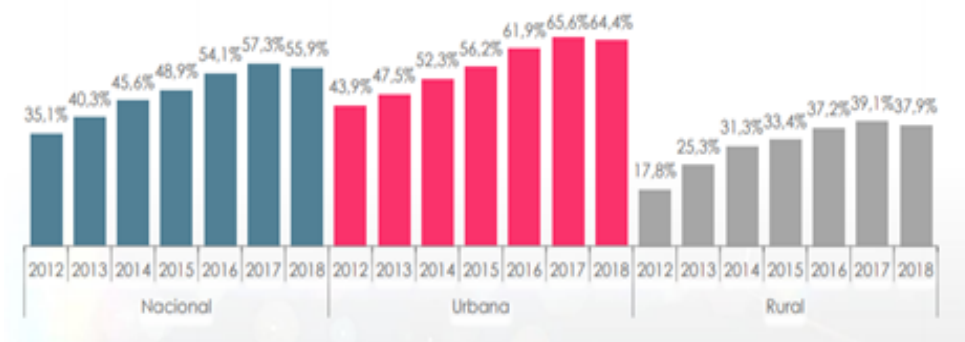


Figura 3. Estadística de personas que han usado internet en los últimos 12 meses.
Fuente: INEC (2018)

El motivo de las razones por la que las personas utilizan el internet se ve reflejado en la Figura 4, donde obtener información es liderada como motivos de uso de internet, continuando por comunicación general, posteriormente por educación y aprendizaje, razones de trabajo y otros. Dicha estadística fue realizada hasta el año 2016.

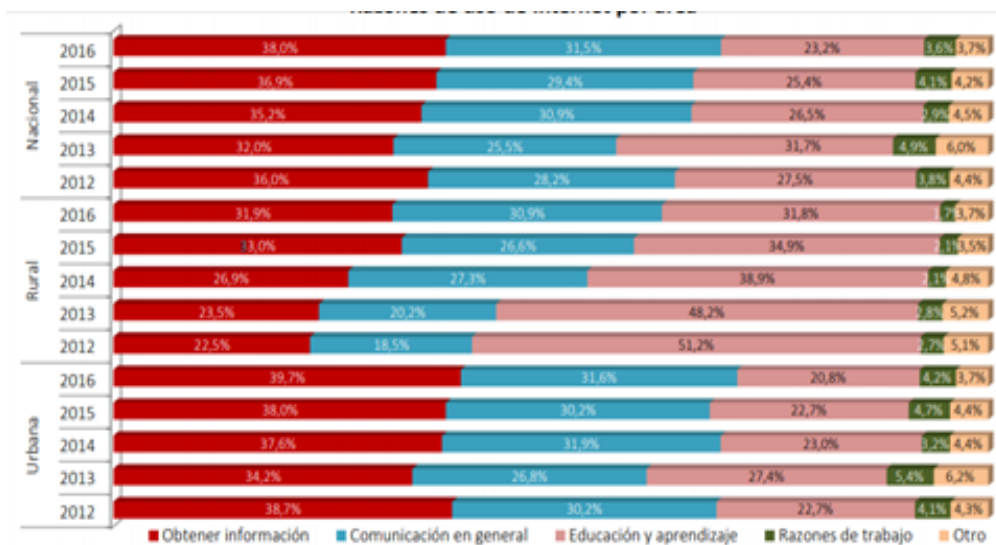


Figura 4. Estadística de razones de uso de internet por áreas urbana, rural y nacional
Fuente: INEC (2016)

Los resultados de las estadísticas anteriormente mencionadas dejan claro que la tecnología en el Ecuador está tomando mayor realce, en gran parte a los jóvenes. De modo que la utilización y accesibilidad de estas herramientas tecnológicas es fundamentalmente importante en la educación. Guaña et al., (2017) menciona que Ecuador en los últimos años ha experimentado un incremento considerable en la utilización de la tecnología, lo que en relación con la educación está en un cambio

continuo, debido a que todas las universidades se han centrado en la acreditación y calidad educativa lo que en consecuencia hace que sus estudiantes adquieran habilidades para ser partícipes del mundo tecnológico.

2.2.1.6 Aplicación informática para actividades académicas

Las actividades de aprendizaje son, en primer lugar, acciones. Sin embargo, también son recursos que permiten conseguir el aprendizaje y no solo medios para comprobarlo. Dentro del aspecto de aplicaciones para actividades académicas, se encuentran las denominadas aulas virtuales, que facilitan tanto a docentes como estudiantes visualizar, y gestionar actividades que conducen al aprendizaje. Existen dos tipos de aulas virtuales (aplicación informática), la primera que solo se enfocan en la educación online y las que facilitan el aspecto docente presencial (Fernández, M., Sánchez, A., y Heras, D., 2020). Ambas herramientas se diferencian por las funcionalidades que se orientan en la enseñanza y aprendizaje online y presencial.

En el mismo sentido Fernández, M., Sánchez, A., y Heras, D. (2020) mencionan que dentro del aspecto docente es claro que estas herramientas informáticas complementan la educación presencial, a través de apuntes, entrega de tareas, distribución de información académica, etc. Como un claro ejemplo establece a Moodle (Aula virtual) o aplicaciones similares. En definitiva, este tipo de programas facilitan el anexo entre estudiante-docente, permitiendo eficazmente evaluar las tareas o actividades que son entregadas por parte del alumnado.

En el siguiente gráfico se visualiza el uso de una aplicación informática para gestión de actividades (Aulas virtuales), en el grado de Matemáticas de la País Vasco/Euskal Herriko Unibertsitatea (UPV/EHU), en la cual se puede evidenciar el porcentaje aumenta considerablemente en comparación a años anteriores.

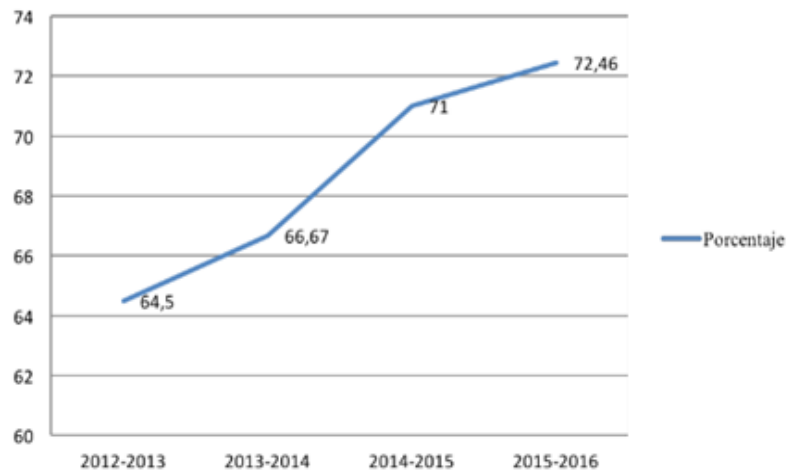


Figura 5. Evolución del porcentaje de aulas virtuales (Aplicación informática) abiertas en el Grado en Matemáticas de la UPV/EH

Fuente: Camacho & De Santurtzi (2016)

Toda la información anteriormente referenciada indica que existe un realce en la incorporación y utilización de herramientas digitales en el aspecto educativo (lo que incluye docentes, estudiantes, administrativos, etc.) Justificando la factibilidad de tecnologías de la información en cualquier área o temática que tenga relación con el proceso académico de las personas.

Por último, si consideramos el tiempo actual, es evidente que la utilización de las herramientas digitales en el aspecto de la educación están en su punto máximo, esto debido a los eventos suscitados en los últimos meses en el contexto de la pandemia. Pues, el aspecto virtual toma bastante protagonismo.

2.2.2. Actividades Estudiantiles

Una actividad de enseñanza-aprendizaje es un procedimiento que se realiza en un aula de clase para facilitar el conocimiento en los estudiantes, y se eligen con el propósito de motivar la participación en el proceso de enseñanza y aprendizaje presencial (Fernández, M., Sánchez, A., y Heras, D., 2020). Las actividades estudiantiles son todas aquellas operaciones o trabajos que se realizan dentro del proceso de aprendizaje, las mismas tiene como objetivos fortalecer los conocimientos adquiridos en el aula, intensificar las habilidades necesarias para adaptarse a su campo de trabajo, debido a esto se puede argumentar que las actividades académicas no solamente son necesarias, sino que son fundamentales en el proceso de aprendizaje

de un estudiante por su aporte facilitando el desarrollo de habilidades que permitirán un desempeño óptimo en la vida laboral de los estudiantes.

Las actividades académicas pueden ser: tareas, trabajos autónomos, trabajos en grupo, consulta, prácticas, exposiciones, foros, cursos, conferencias, visitas a lugares específicos, entre otros, todos con el fin de aportar conocimientos académicos para favorecer el desarrollo académico del estudiante.

Y son los mismos puntos citados, los que se toma en cuenta en la investigación, además, de las diferentes actividades que se consideren relevantes, las cuales surgen a través del estudio y aplicación de los instrumentos de investigación, tales como entrevistas o encuestas. Por lo que se define en la operacionalización de las variables. Es decir que, en función de las actividades y los datos obtenidos de los instrumentos de investigación se define la metodología de desarrollo lo que a su vez surge el software como propuesta a desarrollar

2.2.2.1. Recursos digitales enfocados al alumnado

Las TIC han pasado a ser parte de la cotidianidad de los sujetos. Ante esta situación, la educación debe considerar la introducción de determinados recursos tecnológicos con el objetivo de adaptarse a las necesidades sociales del momento. La evolución de estos recursos aumenta considerablemente el abanico de posibilidades y oportunidades pedagógicas, proporcionando una gran utilidad educativa (Botía y Marín, 2019).

Según Botía y Marín (2019), los recursos académicos digitales como materiales utilizados para la enseñanza, aprendizaje e investigación. Dichos recursos pueden ser usados alrededor de todo el mundo, al ser recursos digitales facilitan la accesibilidad a los mismos por medio de Internet, por lo que se los considera como un gran incentivo para la nueva sociedad del conocimiento provocando un gran impacto en el proceso de aprendizaje de los estudiantes.

Los recursos digitales en la educación tienen como objetivo facilitar el desarrollo de actividades de aprendizaje tanto para el docente como para el estudiante y ofrece

varias ventajas que los recursos tradicionales no pueden favorecer al desarrollo del proceso de aprendizaje.

Hasta el momento, mucho se ha mencionado de las actividades de aprendizaje, agendas digitales, software a medida, etc. Sin embargo, quedan otros puntos clave por mencionar como son, las metodologías de desarrollo, sus distintas herramientas, etc.

2.2.3. Metodologías de desarrollo de software

Las metodologías de desarrollo de software han pasado por un proceso de evolución que inicia en los años 40 cuando aparecen las primeras computadoras , en esa época no tenía ningún tipo de estándar ni parámetros , lo que llevó a que gran parte de proyectos fracasaran al no cumplir con las expectativas de los usuarios sin respetar ninguna fecha límite de entrega ni presupuesto desembocando en la llamada “Crisis del Software” en donde la única opción vista para superarla fue adoptando metodologías clásicas que progresivamente fueron incorporando estándares, controles y reglas al desarrollo del software (Zumba, 2018).

La industria del Hardware seguía creciendo, con procesos estandarizados, divididos en tareas u operaciones, documentos, manuales y certificaciones de calidad; el software parecía no alienarse a esa rigurosidad, en puertal de los años 70, los métodos de desarrollo aún eran informales. En 1972 Edsger Dijkstra, presenta su trabajo titulado “The Humble Programmer” y sienta las bases para la creación de las metodologías tradicionales conocidas y aún usadas hasta hoy (Zumba, 2018).

En lo que respecta a esta temática es evidente el transcurso de cambios que ha afectado a las metodologías, permitiendo el desarrollo de propuestas de software. Los mismos cambios han permitido el nacimiento de nuevos métodos. Muchos enfocadas a los usuarios, al software, accesibilidad, usabilidad, etc. La investigación se basó en la empatía con el usuario, en donde se recolecta información necesaria y relevante, para permitir diseñar una propuesta que este acorde a los requerimientos.

En los siguientes puntos se detallan las principales metodologías de desarrollo que se tomaron en cuenta en esta investigación, de manera que, entre las que se mencionan se hace la elección de una metodología que este acorde a esta investigación. Es decir,

que la metodología elegida se relacionará con las principales actividades del estudiante. Haciendo empatía con el usuario.

2.2.3.1 Metodologías tradicionales

Las metodologías de desarrollo tradicionales o clásicas fueron abordadas para ordenar el caótico proceso que conllevaba el desarrollo de software, definiendo así un proceso secuencial en el que se avanza de forma unidireccional y sin opción a retroceder, este proceso es detallado de forma irreversible y no hay cabida para cambios, una vez que los requerimientos son definidos para el funcionamiento del proyecto, se establece plazos y tiempo de comunicación con el cliente (Montero et al., 2018).

- Cascada

El modelo en cascada es una metodología de desarrollo lineal, el desarrollo del proyecto debe ser dividido en fases de forma que se cree un proceso secuencial, en el cual cada fase debe completarse para poder continuar con la siguiente sin ninguna opción a retroceso, siendo así esta metodología la base para otras metodologías tradicionales (Marquez y Fernandez, 2019).

- Prototipado

Es un modelo de desarrollo repetitivo en donde se plantea una maqueta o prototipo del producto. Este prototipo que será desarrollado por el equipo de trabajo conjuntamente con las ideas del cliente permite derivar a requerimientos que ayudarán a la construcción de la maqueta o prototipo (Marquez y Fernandez, 2019).

- DRA

Desarrollo Rápido de Aplicaciones método, también conocido como RAD (en inglés Rapid Application Development), implica un proceso de desarrollo lineal secuencial que se trabaja en paralelo con dos equipos de trabajo cada equipo estará encargado de una parte funcional del proyecto con el objetivo de establecer un espacio corto de tiempo para terminar un producto reutilizando componentes (Marquez & Fernandez, 2019).

Es claro que varias de las metodologías ya mencionadas tienen sus ventajas y desventajas, ya sea en tiempo, recolección de la información, etc. Sin embargo, en

esta investigación se descarta las metodologías tradicionales debido a su secuencia lineal de trabajo.

2.2.3.2 Metodologías ágiles

Las metodologías ágiles para el desarrollo de software aparecen como alternativas a las metodologías clásicas que en su momento generaron resultados ineficientes, es así como nacen nuevas alternativas que se enfocan en el software sin dejar de lado a la arquitectura o documentación, permite que los requerimientos sean ajustables y hace participe al cliente desde etapas tempranas (Zumba, 2018).

- Scrum

Scrum es un marco de trabajo diseñado de tal forma que logra la colaboración eficaz del equipo de trabajo, emplea un conjunto de reglas y se definen roles para generar una estructura de correcto funcionamiento. Scrum define tres roles, los cuales son: El Scrum master, el dueño del producto o Product owner y el equipo de desarrollo o team. El scrum master es la persona que lidera el equipo asegurándose que el equipo cumpla las reglas y procesos de la metodología. El dueño del producto es el representante de los accionistas y clientes que usan el software. El equipo de desarrollo es el grupo de profesionales encargados de convertir la lista de requerimientos o también llamado Product Backlog en funcionalidades del software (Montero et al., 2018).

- Extreme Programming (XP)

Es concebida para proyecto de pequeño y medio tamaño, prevé participación activa y frecuente con el cliente. Esta metodología está designada a un campo más práctico y accesible, numérico para el manejo de softwares a desarrollar, el trabajo se facilita al desempeñar funciones que no requieren de componentes complejos (Marquez & Fernandez, 2019).

2.2.3.3. Comparación entre metodologías tradicionales y ágiles.

Las metodologías ágiles, al igual que las tradicionales ayudan al desarrollo de propuestas informáticas. Sin embargo, lo que diferencia es la capacidad de iterar, al contrario de las tradicionales que eran secuenciales lineales. Si se refiere a SCRUM,

es evidente las ventajas que trae consigo, sin embargo, debido a que requiere mayoritariamente personal con roles asignados, no se considera factible su aplicabilidad en la investigación por el hecho de que no se cuenta con la cantidad de personal correspondiente a lo que requiere SCRUM. En cuanto a XP, es la que se eligió para esta investigación, debido a que acepta proyectos de pequeño y mediano tamaño, así como también el requerimiento de la actividad permanente del cliente durante el tiempo del desarrollo, además de la programación en pares. En donde el mismo se ajusta al recurso humano disponible.

De las ya mencionadas metodologías, existen otras que se enfocan en la investigación, empatía con el usuario, o stakeholders (afectados con el resultado del proyecto), como pueden ser el Desing Thinking, si bien es cierto, que no nace del desarrollo sino de la filosofía de los diseñadores gráficos, son puntos que se consideró en la investigación.

En la siguiente tabla se visualiza la comparación entre las metodologías ágiles y tradicionales. Afirmando que las m. ágiles se consideran en el desarrollo del software.

Tabla 1.
Comparación entre metodologías ágiles y tradicionales

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Preparados para el cambio durante el proyecto.	Cierta resistencia a los cambios.
Reglas de trabajo impuestas internamente (Por equipos).	Reglas de trabajo impuestas externamente.
Proceso menos controlado, con pocos principios.	Procesos mucho más controlados con numerosas políticas/ Normas.
El Cliente es parte del desarrollo.	El Cliente interactúa con el equipo de desarrollo mediante reuniones en determinadas etapas del proceso.

Pocos Artefactos	Más Artefactos.
Pocos Roles	Más Roles.

Fuente: Meléndez Valladarez, S. M., Pérez Reyes, N. N., & Gaitan, E. M. (2017). *Metodología ágil de desarrollo de software programación extrema (tesis pregrado)*. Universidad Nacional Autónoma de Nicaragua, Managua

2.2.3.4. Valores y Objetivos de la metodología XP.

Hasta el momento se sabe que esta metodología es ágil, eficiente y su riesgo no es alto, pues es predecible y flexible. De acuerdo con Reyes y Marin (2021) mencionan que esta metodología se basa en los valores de comunicación (parte fundamental de la metodología), simplicidad(mantener la simplicidad en la codificación), retroalimentación(el cliente es parte del desarrollo, viendo que se cumplan sus requerimientos), coraje(mantener el coraje ante situaciones de cambio) y respeto (el valor que debe prevalecer entre los miembros del equipo).

En cuanto a los objetivos que menciona el mismo autor citado anteriormente son: la satisfacción del cliente, la potenciación del trabajo en equipo, minimización al riesgo a través del costo, tiempo, calidad y alcance. Pues, estos factores determinan a XP como una metodología que se basa en prueba y error, pues el cliente participa activamente en cuanto al desarrollo del software se trata. Tal y como se mencionó anteriormente.

2.2.3.5. Fases de la metodología XP

Las fases de la metodología XP, está dada por 4 fases, que tiene relación con los valores y objetivos mencionados anteriormente. Según Cardenas y Quimbita (2017) mencionan que cada fase tiene un clico y en función a los requerimientos que se tenga en cada fase, se detalla lo siguiente:

- **Planificación:** Es una de las fases más importantes de esta metodología, pues define la estructura, requerimientos, alcance mediante las historias de usuario,

pues se describe las funcionalidades y características del proyecto de desarrollo.

- **Diseño:** El diseño debe ser simple y sencillo sin perder el principio de funcionalidad con el objetivo de cumplir el tiempo de entrega y facilitar el desarrollo en las siguientes etapas del proyecto. Es recomendable tener un glosario de términos, al igual que mantener organizada las clases y métodos correctamente.
- **Codificación:** Una vez planteada el diseño del proyecto de desarrollo, se procede a aplicar los diseños a través de diferentes herramientas de desarrollo al igual que las herramientas de esta metodología como lo son las historias de usuario. Pues cada una pasara por un proceso de pruebas correspondientes a cada responsable. Generalmente el trabajo de los desarrolladores debe ser en parejas.
- **Pruebas:** En lo que respecta a las pruebas, es aquí donde se aplica las pruebas unitarias, con el objetivo de cumplir las metas planteadas. Estos están en el contexto de un marco de trabajo que automatiza los procesos de validación de datos, funcionalidades, pruebas de aceptación, etc.

En relación con lo mencionado, se resume en la siguiente figura:

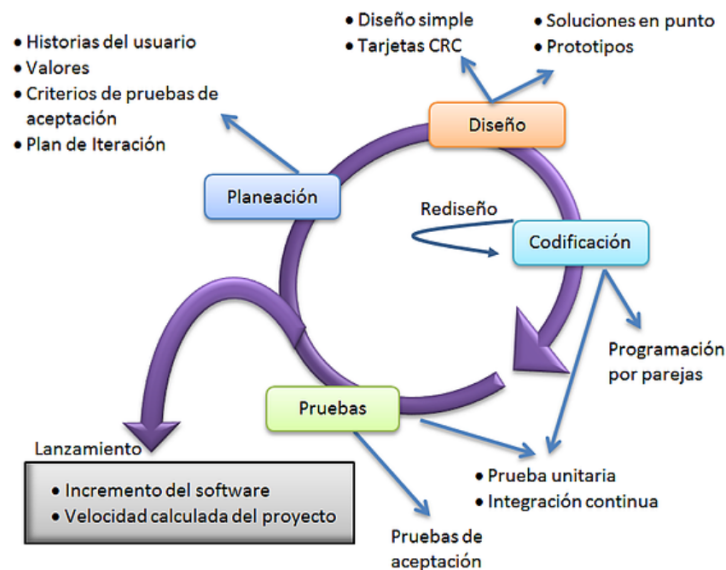


Figura 6. Metodología XP o Programación Extrema

Fuente: Metodología XP o Programación Extrema de software 2020 por Pinto, E.

2.2.3.6. Herramientas de la metodología XP

La denominada metodología XP, como cualquier otra metodología de desarrollo, contiene diferentes elementos o herramientas que ayudan al trabajo en equipo, las mismas corresponden a las fases de la metodología (levantamiento de requerimientos, diseño del software, pruebas, etc.) De acuerdo con Valladarez et al. (2016) los instrumentos se definen a continuación:

- **Historias de usuario**

Las historias de usuario corresponden a la información de las necesidades del cliente con respecto al programa informático. Las mismas son usadas para cumplir las estimaciones de tiempo. Cada historia debe ser comprensible para el programador para que se proceda a implementarlas en un tiempo determinado. En la siguiente tabla se visualiza la estructura de esta herramienta.

Tabla 2.

Ficha de historia de usuario

HISTORIA DE USUARIO	
Número: Se especifica el número de identificación de la historia	Usuario: La persona que usará la funcionalidad detallada en esta historia de usuario
Nombre Historia: El título de la historia.	
Prioridad de Negocio: Nivel o grado de importancia de la historia de usuario para el cliente	Riesgo en Desarrollo: Medida de complejidad de la historia para el equipo de desarrollo.
Puntos Estimados: Cantidad de tiempo (en semanas) para desarrollar la funcionalidad especificada en la historia	Iteración Asignada: Valor máximo de iteraciones para la implementación de la historia
Programador Responsable: Persona encargada de desarrollar la historia de usuario.	

Descripción: Información detallada de la historia de usuario

Observaciones: Opcional, pues se utiliza para aclarar cualquier aspecto de la historia, si es necesario.

Fuente: Meléndez Valladarez, S. M., Pérez Reyes, N. N., & Gaitan, E. M. (2017). *Metodología ágil de desarrollo de software programación extrema (tesis pregrado)*. Universidad Nacional Autónoma de Nicaragua, Managua.

- **Tarjetas CRC (Clase Responsabilidades Colaborador)**

Estas tarjetas denominadas CRC, permiten conocer la arquitectura del software por medio de clases que interactúan entre sí. Cada una se compone de atributos, responsabilidades y colaboradores. De la misma manera se definen las dependencias que puede existir con la clase principal.

Tabla 3.

Tarjeta CRC (Clase – Responsabilidades - Colaborador)

TARJETAS CRC	
Nombre de la clase: Tal como se detalla se especifica el nombre de la clase	
Responsabilidades: Atributos y propiedades de la clase	Colaboradores: Las respectivas clases que se relacionan

Fuente: Meléndez Valladarez, S. M., Pérez Reyes, N. N., & Gaitan, E. M. (2017). *Metodología ágil de desarrollo de software programación extrema (tesis pregrado)*. Universidad Nacional Autónoma de Nicaragua, Managua

- **Tarea de usuario**

Las tareas de usuario o también denominadas tareas de ingenierías (Task Card) vienen compuestas por las historias de usuario. Es decir, que cada historia se puede descomponer en tareas de usuario. Esto con el objetivo de ser más específicos al

momento del desarrollo. Además, las tareas de usuario permiten estimar el tiempo que se va a tardar la implementación de cada historia de usuario. Cabe resaltar que por cada historia puede existir más de una tarea.

Tabla 4.

Tarea de usuario o Tarea de ingeniería

TAREA DE USUARIO		
Número de tarea: Número de identificación de la tarea	Numero de Corresponde asignado de la correspondiente	Historia: al número de la historia
Nombre Tarea: El título de la tarea.		
Tipo de Tarea: Pueden ser: instalación, desarrollo, diseño o implementación.	Puntos Estimados: Número de días para desarrollar esta tarea.	
Fecha Inicio: Fecha inicial del desarrollo de la tarea	Fecha Fin: Fecha final para cerrar la tarea	
Programador Responsable: Persona encargada de desarrollar la tarea de usuario.		
Descripción: Información detallada de la tarea.		

Fuente: Meléndez Valladarez, S. M., Pérez Reyes, N. N., & Gaitan, E. M. (2017). *Metodología ágil de desarrollo de software programación extrema (tesis pregrado)*. Universidad Nacional Autónoma de Nicaragua, Managua

- **Prueba de aceptación**

En lo que respecta a las pruebas de aceptación, son consideradas de vital importancia, pues permiten conocer el éxito de una iteración y el comienzo del siguiente. Además, permite generar una retroalimentación para futuras historias de usuario a ser entregadas.

Tabla 5.

Ficha de prueba de aceptación

PRUEBAS DE ACEPTACIÓN	
Código: El valor es único, pues permite identificar la prueba de aceptación.	N° Historia de Usuario: Corresponde al número de la historia de usuario.
Historia de Usuario: Título o nombre de la historia de usuario	
Descripción: Información sobre la acción que se ejecuta.	
Condiciones de Ejecución: Se refiere a las condiciones que se necesitan para cumplirse la prueba de aceptación.	
Entrada/Pasos de Ejecución: Las respectivas acciones o interacciones que debe realizar el usuario para evaluar la funcionalidad	
Resultado Esperado: Respuesta esperada por el usuario en función de las acciones realizadas.	
Evaluación de la Prueba: Nivel de aceptación por parte del usuario sobre la respuesta del sistema. Los valores pueden ser Aprobada y No Aprobada	

Fuente: Meléndez Valladarez, S. M., Pérez Reyes, N. N., & Gaitan, E. M. (2017). *Metodología ágil de desarrollo de software programación extrema (tesis pregrado)*. Universidad Nacional Autónoma de Nicaragua, Managua

En consideración del análisis de las diferentes herramientas de la metodología XP, se planteó su aplicabilidad. Lo que permitió que los riesgos se minimicen, y el trabajo en equipo sea llevado a cabo de manera efectiva. Estos elementos lograron que la interacción del equipo de desarrollo y el usuario sean adecuados.

2.2.3.7. Roles XP

Los roles son una parte importante para la comprensión de la metodología XP. Por lo que en este apartado se detallan los roles más relevantes. En este mismo sentido, cabe resaltar que los mismos miembros del equipo pueden optar diferentes posturas de la metodología. Para la siguiente descripción de los roles se cita a Pinto (2020), lo cual menciona las siguientes posturas:

- **Cliente:** Se considera como la pieza básica en la metodología, puesto a que define especificaciones, las cuales se ven reflejados en las historias de usuario.

De la misma manera, son los responsables de validar las pruebas de aceptación.

- **Programador:** Son los encargados de la implementación de las historias de usuario generadas por el cliente. De la misma manera calcula un tiempo estimado para el desarrollo de las diferentes historias de usuario que se generen. Finalmente, también son responsables de las pruebas unitarias, corrección de errores en el código, etc.
- **Encargado de pruebas:** Se podría decir que apoya al cliente en la preparación de las pruebas funcionales. También, difunde resultados dentro del equipo de trabajo, a consecuencia de la aplicación de las herramientas de soporte para pruebas.
- **Consultor:** Apoya al equipo de trabajo en asuntos muy puntuales. Al considerarse como un miembro externo guía al equipo en la resolución de problemas específicos.
- **Jefe de proyecto:** Se encarga de la comunicación (factor muy importante en la metodología), pues favorece la relación entre los miembros del equipo de trabajo. Finalmente se asegura de alcanzar los objetivos.

2.2.4. Herramientas tecnológicas de desarrollo

2.2.4.1. Entornos de desarrollo

Por otra parte, los entornos de desarrollo o también denominados IDE (Entorno de Desarrollo Integrado), facilitan la creación de software a través de distintas subherramientas que estas poseen. Moreno (2018) menciona que un IDE consta de 4 herramientas: Editor, que se encarga de colorear la sintaxis del programa, compilador, la cual depende del tipo de lenguaje que se encarga de generar los ejecutables, el depurador, que inspecciona el valor de las variables y finalmente el Constructor de interfaces, en donde se puede crear botones, ventanas, etc.

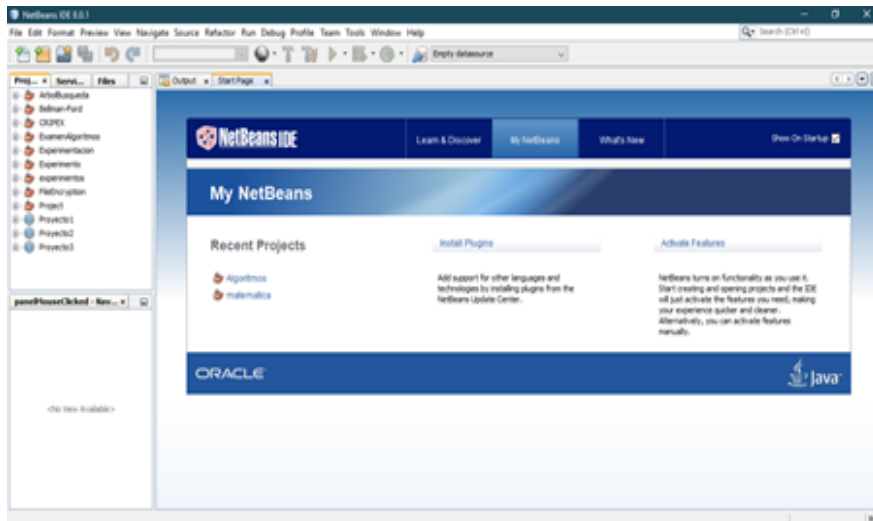


Figura 7. IDE NetBeans, ventana principal

En base, a las características de un IDE, actualmente existe distintas opciones para elegir, esto dependiendo del lenguaje de programación en el que se desea desarrollar, de modo que se menciona en una tabla las más populares, clasificándose por la licencia, el lenguaje y el Sistema Operativo:

	IDE	Licencia	Windows	Linux	Mac OS X
Lenguaje Java	Eclipse	EPL	Sí	Sí	Sí
	NetBeans	CDDL/GPL2	Sí	Sí	Sí
	Visual Studio	Propietario	Sí	No	No
	JDeveloper	Propietario	Sí	Sí	Sí
Lenguaje JavaScript	Eclipse	EPL	Sí	Sí	Sí
	NetBeans	CDDL/GPL2	Sí	Sí	Sí
	Geany	GPL	Sí	Sí	Sí
	KDevelop	GPL	No	Sí	No
	JBuilder	Propietario	Sí	Sí	Sí
	JCreator	Propietario	Sí	No	No
	JDeveloper	Propietario	Sí	Sí	Sí

Figura 8. Lista de IDE's

Fuente: Moreno (2018)

De los entornos de desarrollo anterior mente mencionados, tal como Visual Studio o JDeveloper aceptan más lenguajes de programación. De manera que sean multilenguaje. Si bien es cierto que esta investigación se orientó al desarrollo móvil

también existe sus diversos Entornos de desarrollo, tal es el caso de Android Studio. En la página oficial denominada “developers android” define como un entorno de desarrollo integrado (IDE) para la creación de programas orientadas a Android.

Al igual que cualquier otro IDE, soporta varios lenguajes de programación como también paquetes de desarrollo de software. En esta investigación se utilizó Android Studio para el desarrollo de la propuesta, la cual se redacta en los temas posteriores.

2.2.4.2. Lenguajes de programación

Los denominados lenguajes de programación han sido fundamental para el avance de la tecnología, pues es con estas herramientas lo que ha permitido dar vida al hardware (ordenadores o computadoras físicos) de máquinas físicas. Telefónica (2019) menciona que programar es una actividad que día a día los ingenieros en software luchan en desarrollar programas a prueba de idiotas, de modo que el concepto de un lenguaje de programación va de la mano con el lenguaje, siendo más específicos, es un lenguaje especial, no natural que ha sido creado con reglas, sintaxis, vocabulario, etc. Con el objetivo de comunicación con las computadoras, a fin de que entienda las instrucciones que codifica el ser humano a los ordenadores.

Existen diversidad de lenguajes de programación, esto en función a que es lo que se desea crear. Si el objetivo es desarrollar un programa para la web y para dispositivos móviles, los lenguajes que van relacionados serían JavaScript (El lenguaje de la web), Typescript (Derivado de JavaScript), Java, Python, C++, PHP, etc. La popularidad de aplicaciones web y móviles han marcado tendencia en los últimos años, de modo que los lenguajes de programación irían ligados con esta tendencia.

Según con el Índice de la Comunidad de Programación TIOBE (2020) el cual mide los lenguajes de programación más usados en el mundo, en su sitio web nombra al lenguaje C, Java, Python, C++ y C# como los más populares, en el año 2020.

Jul 2020	Jul 2019	Change	Programming Language	Ratings	Change
1	2	▲	C	16.45%	+2.24%
2	1	▼	Java	15.10%	+0.04%
3	3		Python	9.09%	-0.17%
4	4		C++	6.21%	-0.49%
5	5		C#	5.25%	+0.88%
6	6		Visual Basic	5.23%	+1.03%
7	7		JavaScript	2.48%	+0.18%
8	20	▲	R	2.41%	+1.57%
9	8	▼	PHP	1.90%	-0.27%
10	13	▲	Swift	1.43%	+0.31%
11	9	▼	SQL	1.40%	-0.58%
12	16	▲	Go	1.21%	+0.19%

Figura 9. Los lenguajes de programación más famosos en el mundo por TIOBE en julio 2020 y 2019.

Fuente: TIOBE (2020)

La clasificación de TIOBE se basa en la cantidad de ingenieros certificados a nivel mundial, cursos y motores de búsqueda más populares como Google, Wikipedia, YouTube, etc.

Por otra parte Popularity of Programming Language Index (PYPL, 2020) de la misma manera lanza su lista de lenguajes más populares en el mundo, su clasificación a diferencia de TIOBE, se basa en la cantidad de búsquedas de tutoriales de los diferentes lenguajes. A continuación, se visualiza la respectiva clasificación del presente año 2020.

Rank	Change	Language	Share	Trend
1		Python	31.73 %	+3.9 %
2		Java	17.13 %	-2.7 %
3		Javascript	7.98 %	-0.3 %
4		C#	6.67 %	-0.6 %
5	↑	C/C++	5.93 %	+0.1 %
6	↓	PHP	5.64 %	-1.1 %
7		R	4.14 %	+0.3 %
8		Objective-C	2.61 %	-0.1 %
9		Swift	2.29 %	-0.1 %
10	↑	TypeScript	1.91 %	+0.2 %
11	↓	Matlab	1.74 %	-0.1 %
12		Kotlin	1.62 %	+0.2 %

Figura 10. Los lenguajes de programación más famosos en el mundo por PYPL en el año 2020

De los lenguajes ya mencionados, se hizo la elección de Python en cuanto al desarrollo de la parte Backend (funciones no visibles para el usuario) del proyecto, debido a su popularidad y facilidad de creación en cuanto a Inteligencia artificial se refiere.

En el mismo sentido, también existe otro lenguaje el cual ha tomado popularidad denominado Dart (Creado por Google), pues debido a la aparición de Flutter ha tomado realce en la comunidad de desarrolladores móviles.

2.2.4.3 Flutter

Flutter es un SDK, es decir un kit de desarrollo de software de interfaz gráfica, considerado Open Source, el mismo que fue creado por Google. Flutter permite desarrollar aplicaciones para Android, IOS, Windows, Mac, Web a partir de un único código base (Menacho, 2020).

En esta investigación se resaltó la importancia de Flutter en cuanto al proyecto, debido a que se hace la utilización del mismo por sus grandes ventajas frente a otros

competidores. Una ventaja importante es el hecho de que las aplicaciones creadas con esta herramienta ofrecen mejor velocidad en cuanto al rendimiento se refiere.

Se considera que este framework se divide en 3 componentes los cuales son: Plataforma Dart (Permite que Flutter tenga un rendimiento elevado), Motor de Flutter (Interactúa con el SDK específico de cada plataforma para implementar gráficos), y Widgets para cada plataforma.

Anteriormente se mencionó el término Framework, pues a continuación se define a detalle este término clave en esta investigación, de manera que ayude a comprender los siguientes apartados.

2.2.4.3. Framework

Es un grupo de herramientas que permiten trabajar de forma sencilla y reutilizar proyectos anteriores, se trata de un esquema que permite desarrollar de forma más exacta el código ayudando a eliminar líneas de código innecesarias, representando una ventaja para los programadores, este marco de referencia ofrece una forma de trabajo estándar en un lenguaje de programación determinado (Valerio, 2018).

El uso de framework dan la oportunidad a un equipo de trabajo de entregar un producto en menor tiempo y con menos código haciendo que la aplicación o producto final cumpla con los requerimientos definidos (Muenta, 2020)

Un punto importante que considerar es que los Frameworks imponen el lenguaje para su utilización, al igual que sus normas, herramientas, librerías, etc. Los cuales facilitan el desarrollo de la propuesta. La utilización varía en función de lo que se requiere realizar o diseñar.

2.2.4.4. Librerías

Es claro que el desarrollo de aplicaciones informáticas es una temática muy amplia. De modo, que existen diferentes ramas, tendencias, términos, en este caso se

describirá de las denominadas librerías. Pedro (2018) define a la librería como fragmentos de código que ayuda o facilitan a los desarrolladores a crear aplicaciones.

Es necesario resaltar que la existencia de las librerías depende del lenguaje de programación en el que se esté trabajando. Muchos lenguajes ya vienen incorporadas ciertas librerías, a las cuales se denomina como internas, de la misma manera existe las externas, que son creadas por diferentes programadores y son difundidas en la comunidad del desarrollo.

En función de los lenguajes más utilizados; java, JavaScript y Python, se menciona en la siguiente figura las distintas librerías más utilizadas que posee cada lenguaje:

Tabla 6.
Librerías en función del lenguaje

Lenguaje de programación	Librería	Descripción
Java	mockito-all,	Usado en pruebas del software
	mockito-core	
	guava	Desarrollado por Google
	commons-io	Desarrollado por Apache
Javascript	Jquery	Más popular en la web
	Modernizr	Detecta funcionalidades HTML5 y CSS3
	Ajax	Utilizado por muchos sitios web
Python	Requests	Realiza peticiones HTTP
	Tqdm	Barras de progreso
	NumPy	Usado para computación científica

2.2.4.5. Tendencias Tecnológicas

Es evidente que las tecnologías proporcionan ciertas ventajas para los usuarios ya sea para el desarrollo de Apps o su uso directo. De acuerdo con Martínez y Cruz-

Gavilánez (2018) menciona que Internet de las Cosas (IoT) es considerado una de las ya mencionadas tendencias, las cuales emergen como parte de un estándar en el mundo de la computación.

La mencionada IoT, funciona de la interconexión de dispositivos u objetos a través del internet, en donde el objetivo primordial de esta tendencia tecnológica es facilitar la vida cotidiana de las personas, en donde enfocan a aspectos alimenticios, entretenimiento, salud, etc.

En función del autor citado, se menciona otra tendencia denominada computación en la nube, pues es un factor muy importante al progreso de las Tecnologías de la información. Posteriormente se menciona las criptomonedas, que si bien es cierto no tiene que ver con la investigación, es necesario resaltar como una de las tendencias tecnológicas que rigen en la actualidad. Finalmente menciona la seguridad cognitiva, en donde incluyen diversas disciplinas como aprendizaje por máquina, procesamiento del lenguaje natural, interacción hombre-máquina, todos los campos mencionados correspondientes o derivados de la inteligencia artificial, considerándolos como otra tendencia que es muy utilizada hoy en día.

En el mismo sentido, si nos enfocamos a Ecuador podemos observar que también consideran tecnologías relevantes como el desarrollo de aplicaciones, infraestructura, etc. Citando a León y Martínez (2020) en su encuesta realizada a 180 líderes de negocio y tecnología para la adopción de las mismas empresas ecuatorianas. En la pregunta No. 8 hace referencia a las habilidades tecnológicas que se ha considerado utilizar en la implementación de la respectiva estrategia digital. En donde un 74% utiliza servicios web para facilitar las iniciativas de transformación digital. Mientras que el 53% es al desarrollo y uso de APIs independientes. Se resalta que el 37% incluye a las metodologías Ágiles.

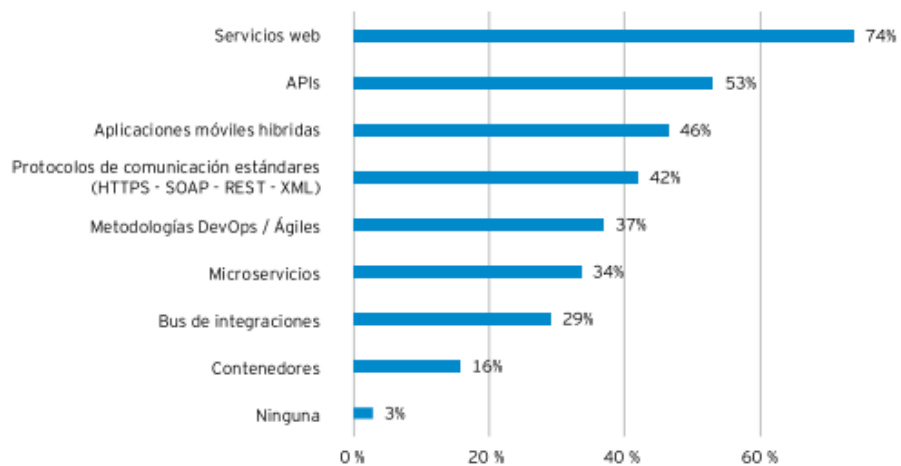


Figura 11. Habilitadores Tecnológicos

Posteriormente, se menciona a las tendencias tecnológicas emergentes que posibilitan en la implementación en las respectivas organizaciones, la mayor parte considera que el Big Data/Data Analytics consideran como factibles para su respectiva organización, mientras que Cloud con un 69% para su utilización en las organizaciones, seguido por Inteligencia Artificial, posteriormente por Procesos Robóticos de Automatización. Para más detalles se muestra en la siguiente figura.

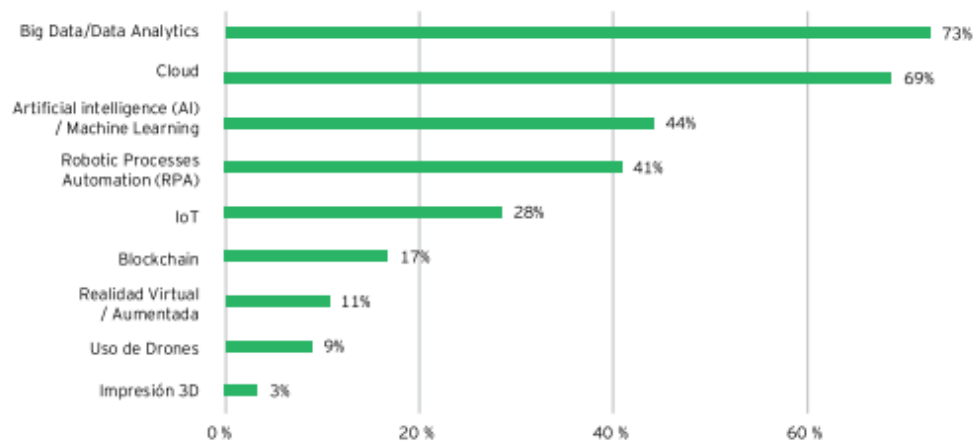


Figura 12. Tecnologías Emergentes

2.2.5. Inteligencia Artificial

En la actualidad, entre las tendencias más populares se encuentran la Inteligencia Artificial (IA). Para definir esta disciplina Astobiza (2021) menciona la

IA como la tecnología que permite a los ordenadores u otros dispositivos simular la inteligencia humana.

En función de la conceptualización descrita, es notorio que expande infinidad de posibilidades en el desarrollo de programas. Actualmente, estas tecnologías son enfocadas más al aspecto educativo. Revista Robots, (2020) menciona varios proyectos en las cuales están enfocadas al aspecto educativo, un ejemplo se muestra en el siguiente párrafo:

- Una revolución educativa: Software desarrollado por Instituto Politécnico Rensselaer que permite el aprendizaje en un entorno virtual a través de la interacción con otras personas de forma amena y atractiva.

El desarrollo de aplicaciones informáticas con utilización de diferentes herramientas permite realizar diferentes tareas en diferentes áreas. Sin embargo, surge la siguiente pregunta ¿Cómo se desarrolla estas aplicaciones?, la respuesta es muy compleja, debido a que existe multitud de herramientas que permiten el desarrollo, en orientación a una o varias disciplinas. De la misma manera, la complejidad del proyecto y de la temática es uno de los factores que es necesario resaltar.

2.2.5.1 Algoritmos de Inteligencia Artificial

Inicialmente se considera la existencia de diferentes algoritmos (Conjunto de operaciones sistemáticas), de Inteligencia Artificial para diversas temáticas en función de lo que se requiere realizar. Si se parte de lo que son los sistemas expertos (Programas que tienen conocimiento en un área determinada), se sabe que para alimentar de conocimiento a los mismos se requiere de terceras personas que realicen el trabajo. Es ahí donde existe una desventaja al momento de la creación y funcionamiento de estos sistemas. En este contexto, es cuando surge el término Aprendizaje Automático o Machine Learning en el cual consiste en que las máquinas aprendan por sí mismas.

De acuerdo con Carbo (2020) menciona 3 grupos de algoritmos principales en relación con el contexto anteriormente mencionado:

- Aprendizaje por refuerzo

Básicamente consta de la iteración prueba y error, es decir que intervienen diferentes elementos o términos, como puede ser la recompensa (podría considerarse el objetivo de lo que se desea alcanzar). El principal concepto que entra en esta temática es la experiencia previa, que es generado por estos algoritmos para obtener conclusiones, correlaciones, patrones, etc.

- Aprendizaje supervisado

Considera a modelos predictivos en el cual usan datos previos de entrenamiento, para posteriormente sean capaces de realizar diferentes salidas en función de los nuevos datos de entrada, con el objetivo de que el modelo sea entrenado hasta lograr resultados coherentes. Aquí se encuentran algoritmos tales como:

- Árboles de decisión
- Clasificaciones Naive Bayes
- Regresión ordinaria por mínimos cuadrados
- Regresión logística
- Support Vector Machines (SVM)

De todos los algoritmos mencionados, el que más relevancia tiene en la actualidad son las redes neuronales con sus respectivas conceptualizaciones. De acuerdo Rivas et al. (2018) menciona que las redes neuronales forman parte de lo que es la inteligencia artificial, por lo que imitan el funcionamiento de las neuronas biológicas de manera que procese la información y permita resolver funciones no lineales.

Entre las ventajas de la red neuronal se describe lo siguiente: Aprendizaje adaptativo, autoorganización, operación en tiempo real y tolerancia a fallos. Mientras que sus respectivas desventajas serian su complejidad para tareas grandes, alto tiempo de

aprendizaje en función de la cantidad y organización de los datos de entrenamiento y no se tiene la capacidad de interpretar los resultados.

En la siguiente imagen o figura se visualizará la estructura de la red neuronal la cual se compone de entradas, pesos en las entradas, función de activación (Sigmoid, TanH, Relu, etc.) y salida:

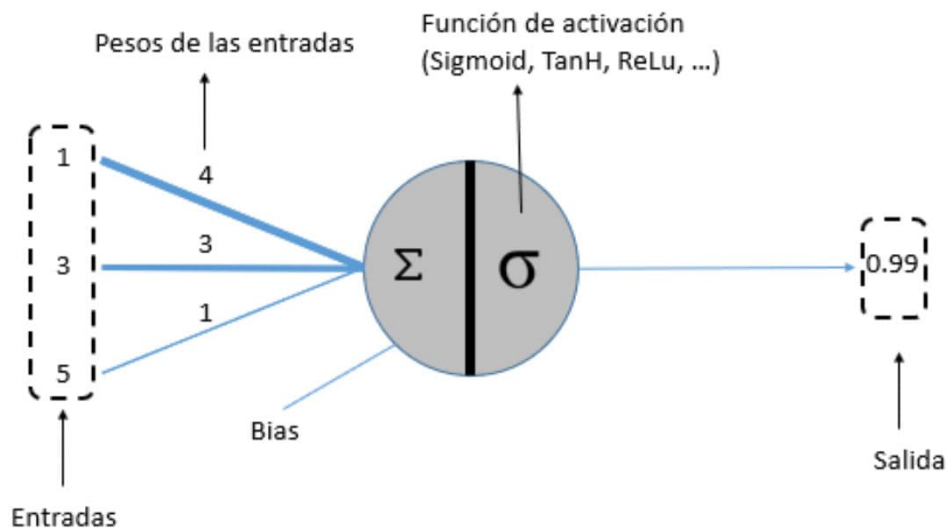


Figura 13. Estructura de la red neuronal artificial
Fuente: (García, 2019)

En resumen, la sumatoria de las entradas multiplicadas por sus pesos se determina el valor posible que surge de lo que son las neuronas. A ese mismo valor se aplica una función de activación que correspondería al valor de salida. En la siguiente gráfica se puede visualizar las respectivas funciones de activación más utilizadas:

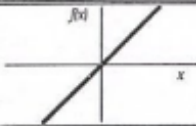
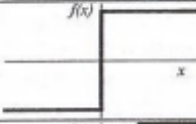
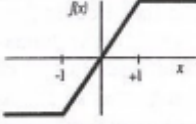
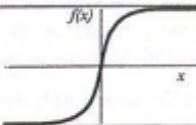
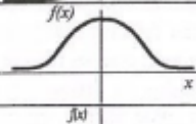
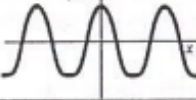
	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal a tramos	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq +l \\ +1, & \text{si } x > +l \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0, +1]$	
Sinusoidal	$y = A \text{sen}(\omega x + \varphi)$	$[-1, +1]$	

Figura 14. Funciones de activación más utilizadas

Fuente: (Serna, 2017)

En este contexto de redes neuronales, es evidente que un modelo de red neuronal está compuesto por más neuronas conectadas entre sí de modo que, en esta conexión surgen diferentes niveles denominadas capas. En donde un modelo contiene una o más capas, compuestas por capa de entrada, capa oculta y capa de salida. Y de esta estructura surge el siguiente concepto de Deep Learning o aprendizaje profundo que nace al usar un gran número de capas en un modelo de red neuronal.

En cuanto a la división o clasificación de las redes neuronales se basan en diferentes factores como la cantidad de capas (monocapa, multicapa), el número de conexiones (recurrentes y no recurrentes), y según su grado de conexión (total mente conectadas, parcialmente conectadas, etc.) (Rivas et al. 2018).

Otro de los conceptos que se relacionan con las redes neuronales son los optimizadores que básicamente ajustan los pesos de modelo de red para que las salidas tengan efectividad en función de los datos de entrenamiento ingresados al

modelo. Existen diferentes optimizadores entre los cuales son: Adgrad, RMSProp, Nadam, Adadelta, Adam (La más usada) (Martinez, 2020).

Finalmente queda por mencionar a las funciones de pérdida, pues las mismas evalúan la desviación que puede existir entre las predicciones realizadas y las respuestas etiquetadas. De modo que mientras menor es el valor, más eficiente es la red neuronal.

- Aprendizaje no supervisado

Tiene relación con el aprendizaje supervisado, sin embargo, este conjunto de algoritmos básicamente realiza un auto entrenamiento sin indicaciones o datos externos. Los principales algoritmos que forman parte de este conjunto son:

- Algoritmos de agrupamiento (clustering)
- Análisis de componentes principales (PCA)
- Singular Value Decomposition (SVD)
- análisis de componentes independientes (ICA)

Si bien es cierto, existe diferentes conjuntos de algoritmos que forman parte del término Inteligencia Artificial, en la actualidad existe diferentes plataformas que ofrecen servicios de desarrollo, tales como: Microsoft Cognitive Services, Google Cloud AI, Watson AI, entre otros.

En lo que respecta al proyecto de investigación se planteó usar librerías, Entornos de desarrollo hasta inclusive las plataformas existentes actuales para la aplicabilidad en la agenda digital, ya sea a través del estudio de los diferentes movimientos de información que existieron, o el estudio de los requerimientos de usuario acoplado al conocimiento adquirido para la propuesta que surgió como resultado de esta investigación.

2.2.5.2 Herramientas de desarrollo IA

López (2018) describe varias herramientas informáticas para el desarrollo de Inteligencia Artificial, en la cual ayuda a incorporar a la rama de la programación:

- TensorFlow: Proyecto de Google, el cual se hizo público en el 2017, en el cual incorpora una rama de la IA denominado Aprendizaje profundo a través de un sistema de código abierto. Su utilización se destina al entrenamiento de redes neuronales, descifrar patrones;
- Keras: Keras (2022) conceptualiza a su herramienta como una API (Interfaz de Programación de Aplicaciones) de aprendizaje profundo en donde se ejecuta en la herramienta Tensor Flow para lo que se refiere al aprendizaje automático. Su característica principal es que se puede realizar una experimentación rápida debido a su desarrollo. Es decir que se pueda pasar de una idea a un resultado.
- Keras y Tensor Flow combina 4 habilidades, la primera es que su ejecución es eficiente en CPU y GPU, la segunda es su capacidad en el escalamiento del cómputo a diversos dispositivos, entre otras. Además, Keras permite el modelado, entrenamiento, predicción, de las redes neuronales, haciendo posible aplicar optimizadores, funciones de pérdida, funciones de activación, etc.
- Numpy: Esta librería perteneciente a Python permite el cálculo, manejo, y análisis de grandes cantidades de datos. Pues, se basa en presentar arreglos o también denominadas Arrays. Esta estructura de datos representa colecciones de un mismo tipo en diferentes dimensiones con funciones eficientes de manipulación (Sanchez, 2020).
- Es decir que, la herramienta tecnológica permite manipular grandes cantidades de datos por medio de Arrays, las cuales se organizan en forma de tablas o cuadrículas en diferentes dimensiones. En la siguiente gráfica se

visualiza la estructura de un array, así como también sus diferentes dimensiones partidos por sus ejes que pueden ser (x, y o z);

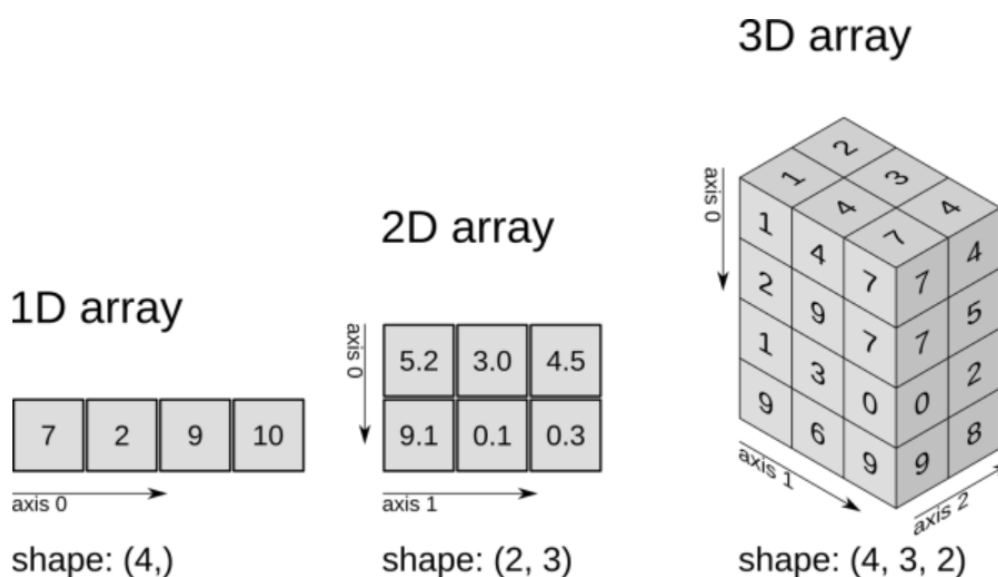


Figura 15. Estructura, eje y dimensión de Arreglos o Arrays
Fuente: (Sanchez, 2020).

- Caffe: Creada por BAIR en el año 2014 se enmarca en el aprendizaje profundo, y es utilizado para la investigación. Facebook lanzó Caffe 2 como sucesor a Caffe;
- ONNX: Esta herramienta se realizó en conjunto con Microsoft y Facebook. Esta tecnología es un formato que permite intercambiar modelos de aprendizaje profundo.

Además de las herramientas anteriormente mencionadas, también existe librerías de Inteligencia Artificial, esto en función del lenguaje de programación. Python y R son dos de los tantos lenguajes que se ha enfocado a la IA, con la incorporación de sus librerías, facilitando el desarrollo de programas. Cabe resaltar que Python se puede enlazar con TensorFlow de manera que permite el desarrollo de IA de manera eficaz y sencilla. En este sentido esta herramienta es la que se utiliza para el desarrollo de la propuesta.

La inteligencia artificial conlleva una gran amplitud de temáticas, ramas o aspectos que cabe considerar, sin embargo, el enfoque de esta investigación se basará por el

estudio del comportamiento de los usuarios con respecto al software, es decir, permitir estudiar el comportamiento o los movimientos de información por medio de la Inteligencia Artificial, para ofrecer una personalización más directa hacia el usuario. La característica clave con lo que respecta a este punto importante (personalización) es la implementación de algoritmos de Inteligencia Artificial, que permitan recomendar, ofrecer, solventar, las necesidades que el usuario pueda presentar en el diario vivir de su vida académica.

2.2.6 Bases de datos

2.2.6.1 Sistema Gestor de Base de Datos (SGBD)

Hasta el momento mucho se ha hablado sobre lenguajes, librerías, aplicaciones etc. De manera que todos estos términos, herramientas no son complementados en un programa sin la existencia de un espacio de almacenamiento de datos, a lo que en informática se denomina como Base de datos. Una Base de datos o BD es un conjunto de datos organizados y almacenados en un sistema para su posterior utilización. Al igual que otras tecnologías, tiene sus clasificaciones, en esta investigación se mencionarán las bases de datos no relaciones y relaciones, al igual que sus SGBD. Al conceptualizar la base de datos, surge un tercer término denominado SGBD o Sistemas Gestores de Base de datos, el cual su función es gestionar las Base de Datos creadas por los diferentes desarrolladores.

Cabe mencionar que la sociedad en la que nos encontramos obliga a generar soluciones con grandes capacidades de almacenamiento, esto debido a que los conjuntos de personas viven un entorno en la cual la información se mueve constantemente, lo que lleva a la obligación de la utilización de Sistemas de almacenamiento como puede ser los SGBD.

En concordancia con los autores Suárez et al. (2018) mencionan que las bases de datos relacionales se han usado desde ya hace mucho tiempo, por su contraparte los BD relacionales tienen la desventaja del rendimiento de ejecución de consultas de datos, debido al existir millones de datos, estos saturan el servidor, disminuyendo su rendimiento. Un ejemplo claro es Facebook, Twitter que, al manejar gran cantidad de

datos, en consecuencia, trae dificultades de controlar la base de datos. En función de los diversos problemas que traía este tipo de BD las bases de datos no relacionales o NOSQL, en donde proporcionan mejor rendimiento de lectura y escritura.

2.2.6.2 SGBD Relacionales

Es la más utilizada en la actualidad. Almacena datos, filas, registros, tuplas, etc. Las cuales están conectadas entre sí (Purizaga, 2019). En otras palabras, este tipo de herramientas almacena y organiza los datos de manera relacional, a través de las denominadas tablas, las cuales componen de campos, filas y columnas, esto en función del nombre de la tabla, y su relación con las demás.

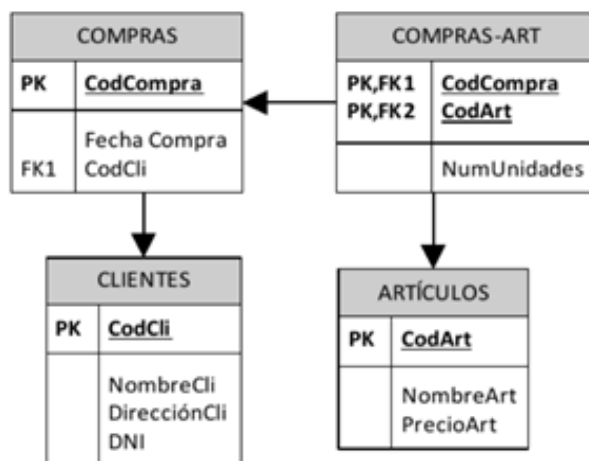


Figura 16. Ejemplo de base de datos relacional
Fuente: Guevara (2019)

Las ventajas que ofrecen son: Menor redundancia (sin repetición de datos), menor espacio de almacenamiento, datos mejor documentados, independencia de los programas, integridad y seguridad. Al igual que sus ventajas también es necesario mencionar sus inconvenientes: la instalación es costosa, requiere un personal cualificado y su implantación es costosa. (Guevara, 2019). Algunos de los ejemplos son: MySQL, OracleDB, MariaDB, PostgreSQL, entre otros.

2.2.6.3 SGBD No Relacionales

Como bien se lo explicó, al inicio de abordar esta temática, la base de datos relacionales trae cierta ventaja en su rendimiento ante gran cantidad de datos. Tan evidente es su rendimiento, que en la actualidad utilizan Facebook, Amazon. Este tipo de tecnología evita el lenguaje estructurado y se basa en colecciones; es decir no se utiliza tablas, columnas o filas, como lo son las relacionales. En reemplazo, se presentan modelos en base clave – valor, lo que, en consecuencia, trae mejor rendimiento (Purizaga, 2020).

Las características de las SGBD no relacionales se muestran a continuación:

- Escalabilidad horizontal: Es decir, agregar más cantidad de clúster;
- Replicación y distribución en diferentes servidores.
- Apropiado uso de índices distribuidos y memoria RAM;
- Posibilidad de agregar dinámicamente, nuevos atributos a los registros guardados.

Para tener más claro el concepto de las SGBD no relacionales, Lafuente (2018) lo conceptualiza como una estructura de información a través de documentos y su utilidad se muestra cuando el desarrollador no tiene una idea clara de lo que se desea almacenar. Algunos ejemplos son: MongoDB, Redis, Elasticsearch y Cassandra.

Es necesario resaltar que las tablas (estructura de los BD relacionales) también pueden ser transformados a documentos, en donde cada no es formato por cada fila de la tabla. En la siguiente imagen, se visualiza de mejor manera, lo anteriormente expresado.

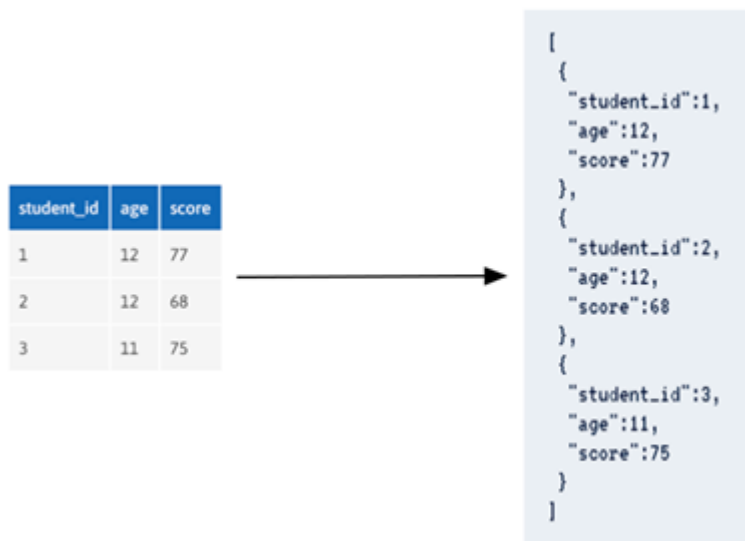


Figura 17. Estructura relacional (tablas) y no relacional (documentos)
Fuente: Lafuente (2018)

Actualmente, una aplicación que no disponga de base de datos, se la considera estática, por lo que no sería interactiva con el usuario. Es por esta razón en lo que respecta a la investigación se consideró la obligación de implementar una Base de datos, de modo que la misma permita el movimiento de la información por parte del usuario, complementando el concepto de personalización. Además, el tipo de Base de datos a considerar es la no relación, por los atributos y ventajas que posee trabajar con este tipo de herramientas, como puede ser el manejo de grandes cantidades de información, y su escalabilidad horizontal.

De manera que la herramienta que fue elegida a consideración del equipo de investigación fue MongoDB, debido a su popularidad y gran cantidad de información para su respectiva utilización.

2.2.6.4. Firebase

Firebase se considera una plataforma de aplicaciones web. Ayuda a los desarrolladores a crear aplicaciones de alta calidad. Almacena los datos en formato de notación de objetos JavaScript (JSON) que no utiliza consultas para insertar, actualizar, eliminar o agregar datos. Es el backend de un sistema que se utiliza como base de datos (Khawas & Shah, 2018).

Firebase está basado en NoSQL y brinda diferentes servicios, tales como:

- **Firestore Analytics:** Proporciona información sobre el uso de la aplicación.
- **Firestore Cloud Messaging:** Anteriormente conocido como Google Clouds Messaging (GCM), FCM es un servicio pago que es una solución multiplataforma para mensajes y notificaciones para Android, aplicaciones web e IOS.
- **Firestore Auth:** Es compatible con proveedores de inicio de sesión social como Facebook, Google GitHub y Twitter
- **Firestore Real-time Database:** Firestore proporciona servicios como una base de datos y backend en tiempo real.
- **Firestore Storage:** Facilita la transferencia de archivos fácil y segura independientemente de la calidad de la red para las aplicaciones de Firestore.
- **Firestore Test Lab for Android:** Proporciona una infraestructura basada en la nube para probar aplicaciones de Android.
- **Firestore Crash Reporting:** Los informes detallados de los errores se crean en la aplicación.
- **Firestore Notifications:** Permite notificaciones de usuarios específicas para desarrolladores de aplicaciones móviles y los servicios están disponibles de forma gratuita.

2.2.7 Visión artificial

La visión artificial es una disciplina de lo que corresponde a la Inteligencia Artificial en el que su objetivo es procesar automáticamente las imágenes para obtener información simbólica de las mismas. Haciendo posible que la máquina pueda interpretar los resultados. En este sentido existen diferentes etapas, las cuales se redactan a continuación (Acuña, 2018).

- **Adquisición de la imagen:** La primera etapa consiste en la obtención de la imagen del mundo físico. Poniendo en consideración factores como la iluminación, sensor óptico y el digitalizador.

- Preprocesamiento: Consiste en reducir el ruido que pueda producir al momento de adquirir la imagen. Esta reducción se basa en aplicar filtros, bordes, etc. Haciendo posible mejorar la imagen.
- Segmentación: Posterior al preprocesamiento sigue la segmentación de la imagen, es decir que se selecciona las características relevantes encontradas en la imagen. Pues, se divide la imagen en objetos diferenciados por el fondo.
- Representación y descripción: Aquí se identifica los objetos que se asocian a la imagen. Lo que se obtiene datos o descriptores de esos objetos.
- Reconocimiento e Interpretación: En función de la anterior etapa se basa en asociar significados a un conjunto de objetos reconocidos

2.2.8 OCR

OCR o también denominado Reconocimiento Óptico de Caracteres, es un sistema que permite reconocer o identificar caracteres alfanuméricos de una imagen. Es decir que permite obtener texto para posteriormente convertirlo en un archivo editable (Ruiz, 2019).

Existen algunas bibliotecas de código abierto para lo que es OCR, como Tesseract, Ocrad, JavaOCR, y Gocr. Inclusive Tensor Flow posee su respectiva herramienta. Sin embargo, la más conocido es Tesseract.

En esta investigación se optó por Tesseract por el simple hecho de su simplicidad en su utilización y aplicabilidad de la herramienta.

- Tesseract – pytesseract. - Según Ruiz (2019) Tesseract es una herramienta de OCR con código abierto creado por HP. Esta herramienta combina lo que es el procesamiento de imágenes de Leptoni y cuenta con un buen apoyo de su comunidad, generando Wrappers (Programas que rodean otros componentes de programa para mejorar la interoperabilidad o compatibilidad). Tal es el caso de pytesseract que se basa en el lenguaje Python.
- Opencv – Opencv Flutter. - Esta herramienta de código abierto contenido por diversos algoritmos que permiten orientarse a la visión artificial y el aprendizaje automático. (Rodriguez, 2021). Esta herramienta es utilizada para

diversos indoles como detección de objetos o rostros, seguridad, marketing, procesamiento de imágenes y fotografía.

- Por otra parte, la comunidad del SDK de Flutter ha creado un plugin que permite la incorporación de Opencv en aplicaciones creadas con este Framework. Con la desventaja que posee las funcionalidades más básicas de procesamiento de imágenes. En esta investigación el plugin abasteció las necesidades para su respectiva utilización.

2.2.9 API

El término API hace referencia a una abreviatura que significa Interfaz de Programación de Aplicaciones. Pues se trata de protocolos que permiten relacionar software de aplicaciones. Es decir que permite la comunicación de dos aplicaciones de software (Fernández, 2019). Las APIS permiten consumir sus servicios sin la necesidad de conocer cómo están implementados. De esto surge lo que es las API remotas, que se diseñan para interactuar por diversos medios de comunicación. Un ejemplo claro son las interfaces de programación web, en donde la comunicación se realiza vía http. Este tipo de tecnología puede ser creada y albergada en un servidor remoto. De manera que pueda ser personalizada en función de las necesidades que un proyecto o desarrollador requiera y disponga.

2.2.10. Heroku

Generalmente las API se albergan en servidores o hosting disponibles en la nube. Tal es el caso de Heroku que permite albergar aplicaciones desarrolladas en diferentes lenguajes. Entre los lenguajes que soporta son: Python, Node JS, Ruby, Java, PHP, Go, Scala, y Cloiure. (Barchino, 2020)

El proyecto de investigación utilizó este hosting por el soporte al lenguaje Python con respectivas librerías al igual que bibliotecas. Además de su servicio gratuito en la memoria RAM, ya que ofrece un máximo de 500 MB de RAM. Lo que hace factible su utilización.

III. METODOLOGÍA

3.1. ENFOQUE METODOLÓGICO

3.1.1. Enfoque

La presente investigación aplicó un enfoque mixto ya que se hizo uso de los enfoques cualitativo y cuantitativo.

3.1.1.1. Enfoque Cualitativo

Tal como menciona Hernández et al. (2017) “La metodología cualitativa, como indica su propia denominación, tiene como objetivo la descripción de las cualidades de un fenómeno”

Se empleó el enfoque cualitativo debido a que se asume una realidad dinámica, es decir, los motivos de utilización de las distintas agendas digitales enfocadas a la educación difieren en función de su utilización y situación. En el mismo sentido la investigación se basó en una lógica y un proceso inductivo, es decir, explorar, describir y generar perspectivas propias de nuestra realidad. De modo que el fenómeno, situación u objeto de estudio es la descripción de las cualidades de una aplicación informática enfocada a la realidad de un estudiante en función de la información que realiza en conjunto con sus respectivas actividades.

3.1.1.2. Enfoque Cuantitativo

Para entender este enfoque Hernández et al. (2017) definen el enfoque cuantitativo como un conjunto de procesos secuenciales y probatorios en donde se determina la hipótesis y se miden las variables en función de la realidad o fenómeno, de manera que al obtener resultados de dicha medición surgen conclusiones.

En el sentido de los autores citados, se aplicó el enfoque cuantitativo en esta investigación debido a la descripción de las aplicaciones informáticas orientadas a la realidad de los estudiantes universitarios en función a sus actividades académicas mediante una encuesta. Es decir, que la información académica que maneja el estudiante fue cuantificada a través de la cantidad de funciones del aplicativo

informático que el alumno utiliza en su vida académica, de forma que los resultados determinaron la validación de la idea que se pretende defender.

3.1.2. Tipo de Investigación

3.1.2.1. Investigación Documental

El presente proyecto investigativo aplicó una investigación documental considerando la revisión de antecedentes investigación donde se utilizó documentos científicos relacionados con agendas digitales para la organización de actividades académicas, así como la revisión bibliográfica realizada sobre el objeto de estudio para el desarrollo del proyecto.

3.1.2.2. Investigación descriptiva

El presente proyecto utilizó una investigación descriptiva porque se describe las agendas digitales para la organización de actividades académicas planteando lo más relevante de este tema, con la finalidad de recabar información, utilizando documentación relacionada con el objeto de estudio, ya sean estos publicados por medios online, libros, y registros de conferencias sobre el tema planteado.

3.2. IDEA A DEFENDER

El desarrollo de una agenda digital personalizada permitirá la organización de actividades académicas de los estudiantes en la Carrera de Computación de la Universidad Politécnica Estatal del Carchi,

3.3. DEFINICIÓN Y OPERACIONALIZACIÓN DE VARIABLES

3.3.1. Definición de Variables

3.3.1.1. Definición Variables Dependiente

Las actividades académicas son todas aquellas operaciones o trabajos que se realizan dentro del proceso de aprendizaje, las mismas tiene como objetivos reforzar los conocimientos adquiridos en el aula, desarrollar habilidades y vincular al estudiante con su campo de trabajo.

3.3.1.1. Definición Variables Independiente

Una agenda digital es un software que provee diferentes funciones orientadas a la gestión de tiempo, organización de tareas y actividades.

3.3.2. Operacionalización de variables

Tabla 7.

Estructura relacional (tablas) y no relacional (documentos)

Variable	Definición	Dimensión	Indicadores	Técnica	Instrumento
Independiente: Agenda digital	Una agenda digital es un software que provee diferentes funciones	- Software - Gestión de tiempo - Organización de	- Número de Programas informáticos. - Número de tareas y	Encuesta	Preguntas cerradas y simples al

	orientadas a la gestión de tiempo, organización de tareas y actividades.	tareas y actividades	actividades.		estudiante.
Dependiente: Actividades académicas	Las actividades académicas son todas aquellas operaciones o trabajos que se realizan dentro del proceso de aprendizaje, las mismas tiene como objetivos reforzar los conocimientos adquiridos en el aula, desarrollar habilidades y vincular al estudiante con su campo de trabajo.	<ul style="list-style-type: none"> - Operaciones o trabajos académicos - Proceso de aprendizaje - Adquisición de conocimiento - Habilidades estudiantiles - Instrumentos tecnológicos en el campo de trabajo 	<ul style="list-style-type: none"> - Cantidad de trabajos incluye recursos (digital). - Recursos en el proceso de aprendizaje - Cantidad de maneras de adquirir conocimiento con medios digitales - Número de habilidades - Cantidad de instrumentos tecnológicos disponibles en el estudiante 	Observación Encuesta	Ficha de observación en formato CheckList Preguntas cerradas y simples al estudiante.

3.4. MÉTODOS UTILIZADOS

3.4.1 MÉTODOS

3.4.1.1 Método Analítico – Sintético

Rodríguez y Pérez (2017) mencionan que, este método refiere al análisis y la síntesis, es decir, el análisis permite la separación de en todas sus partes o en los elementos constitutivos de un todo y la síntesis consiste en unir elementos que forman un todo. En el presente proyecto se realizará la descomposición y determinación de los elementos de un todo para analizar ordenadamente cada uno de ellos.

Se aplicó este método ya que en el presente proyecto se realiza el análisis del uso de agendas digitales, mediante los programas informáticos genéricos existentes, las cuales ayudaran a determinar o separar las características más relevantes de los mismos, de manera que estos sean sintetizados en una aplicación que conlleve las necesidades de los estudiantes, es decir, la separación de un todo en sus elementos constitutivos como; los entornos virtuales de aprendizaje, aulas virtuales, gestores académicos, Repositorios académicos digitales, la ingeniería de software, la metodología de desarrollo, en otros aspectos a considerar.

3.4.1.2 Método Deductivo – Inductivo

Rodríguez y Pérez (2017), afirman que, el método inductivo-deductivo está conformado por dos procedimientos inversos: inducción y deducción. La inducción es una forma de razonamiento en la que se pasa del conocimiento de casos particulares a un conocimiento más general, que refleja lo que hay de común en los fenómenos individuales.

Este método se aplicó en el presente proyecto ya que refiere en establecer algunos aspectos a partir del conocimiento tecnológico adquirido, se puede también tener una referencia general a las agendas digitales o aplicaciones genéricas, y a partir de ello dirigirse a casos particulares como la informática virtual, aulas virtuales, aplicaciones móviles, aplicaciones web, requerimientos tecnológicos de estudiantes, etc. La cual es la base fundamental para el estudio de estas temáticas de la investigación como tal.

3.5. ANÁLISIS ESTADÍSTICO

El manejo de datos de esta investigación se realizó con el objetivo de obtener resultados cuantificables con respecto a la organización del estudiante y las agendas digitales, a la vez también obtener información en cuanto al desarrollo de una propuesta se refiere.

- Población y muestra

En esta investigación se utilizó un tipo de muestreo probabilístico estratificado debido a que la población de estudio es constituida los estudiantes de la carrera de Computación de la Universidad Politécnica Estatal del Carchi con un total de 270 estudiantes de acuerdo con secretaria de la carrera, que en parte se divide en niveles de semestre. Y si bien es cierto cada alumno tiene las mismas probabilidades de ser elegido.

En el mismo sentido, para el cálculo de la muestra se utilizó la fórmula que detalla Arias (2016), con un grado de zeta crítico a 1,69 (con 0,91 grados), considerando de la misma manera el margen de error con 0,07, pues el nivel de confianza es afectado debido a la pandemia suscitada en el proceso de esta investigación, por lo que se permitió optar un error muestral en función de la población. Como resultado de la aplicación de la formula se obtuvo un tamaño muestral:

$$n = \frac{N * Z^2 c * p * q}{(N - 1) * e^2 + Z^2 * c * p * q}$$
$$n = \frac{270 * 1.69^2 c * 0.5 * 0.5}{(270 - 1) * 0.07^2 + 1,69^2 * 0.5 * 0.5}$$
$$n = 94,86 = 95$$

En donde **n** es el tamaño de la muestra, **N** el total de los elementos que integran la población, **Z² * c** es el nivel de confianza elevado al cuadrado, **p** y **q** la probabilidad de éxito y fracaso respectivamente, y **e** equivale al error muestral.

- Técnicas de investigación

Las técnicas de investigación que se utilizaron son documentales pues se obtiene información para este estudio y de la misma manera se aplica una encuesta semiestructura a

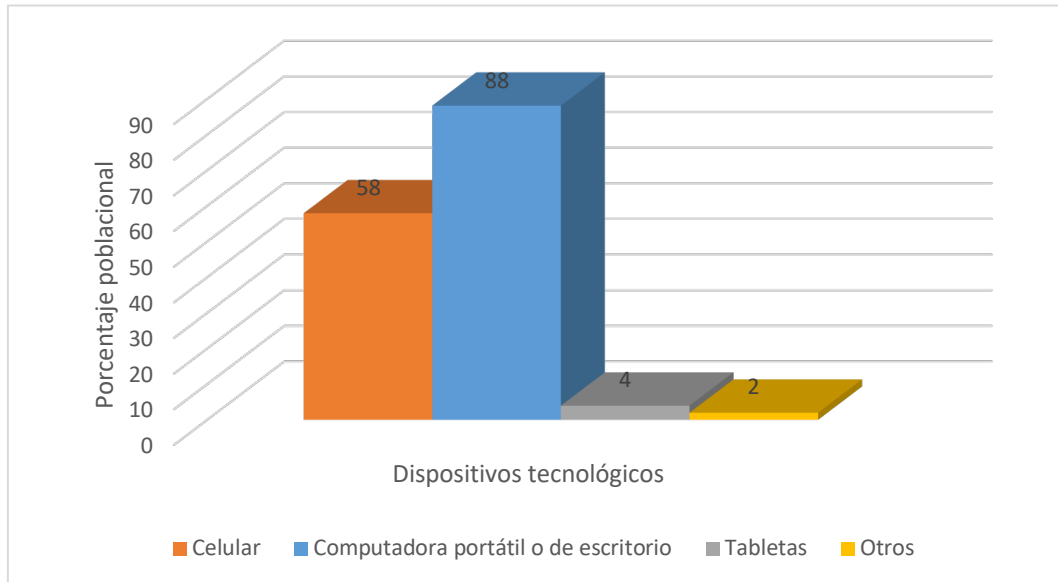
los estudiantes de la carrera de Computación de la Universidad Politécnica Estatal de Carchi para identificar la organización de las actividades académicas por parte del alumnado, así como también los programas que maneja con relación a esta misma variable.

IV. RESULTADOS Y DISCUSIÓN

4.1 RESULTADOS

4.1.1. Resultados de la encuesta

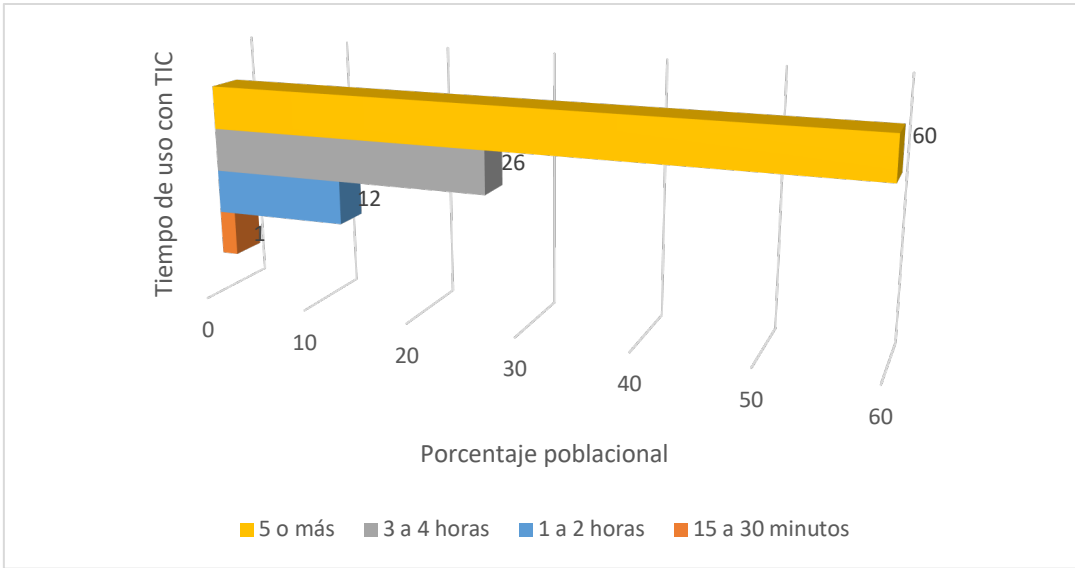
1) ¿Con que medios tecnológicos se apoya su proceso de aprendizaje?



Análisis

En lo que respecta a la primera pregunta cabe resaltar que fue de elección múltiple, de modo que el porcentaje varía en función del 100 por ciento. En este sentido los resultados que se pudo observar es que la mayor parte de estudiantes de la universidad (muestra) consideran la importancia de las computadoras portátiles y de escritorio para el proceso de aprendizaje. Mientras que un poco más de la mitad de la muestra, usan el celular para lo que respecta su aprendizaje. Dando a entender que, entre las opciones presentadas, el celular es la segunda herramienta utilizada por los estudiantes de la carrera de computación en lo que respecta a lo académico.

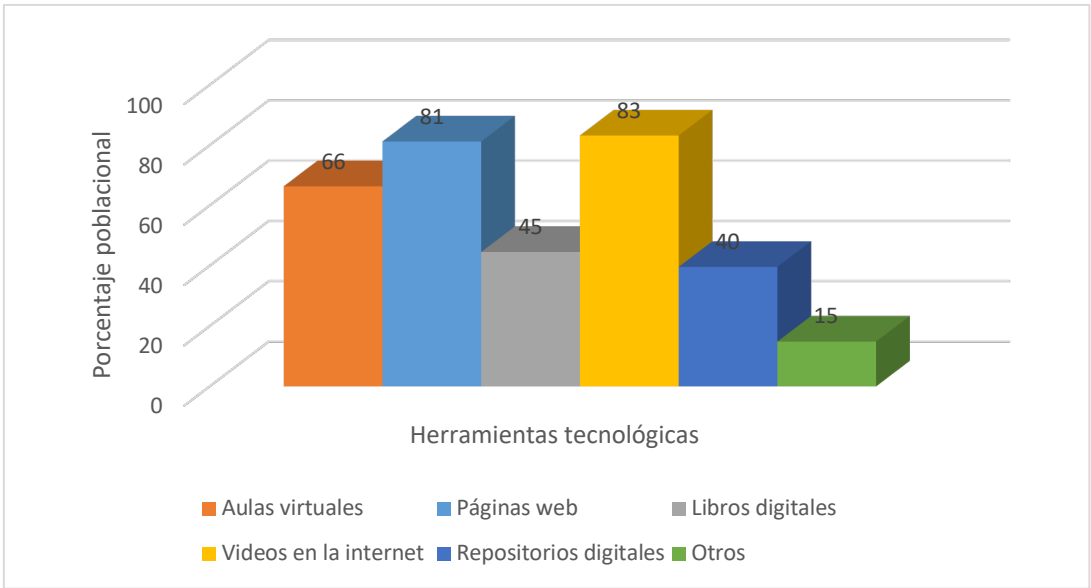
2) ¿Cuánto tiempo destina en el uso de los instrumentos tecnológicos anteriormente mencionados para su proceso de aprendizaje?



Análisis

En la figura se muestran diferentes resultados en el cual la opción “5 o más” tiene un porcentaje mayor a la mitad, seguido por “3 a 4 horas” con una cuarta parte. Esto lleva a pensar que los encuestados utilizan más de 5 horas en el uso de herramientas tecnológicas como el celular o la computadora. Dando como una gran posibilidad la creación de programas informáticos orientados a las plataformas móviles y computadoras. De modo que hace factible la creación de una agenda digital para los estudiantes.

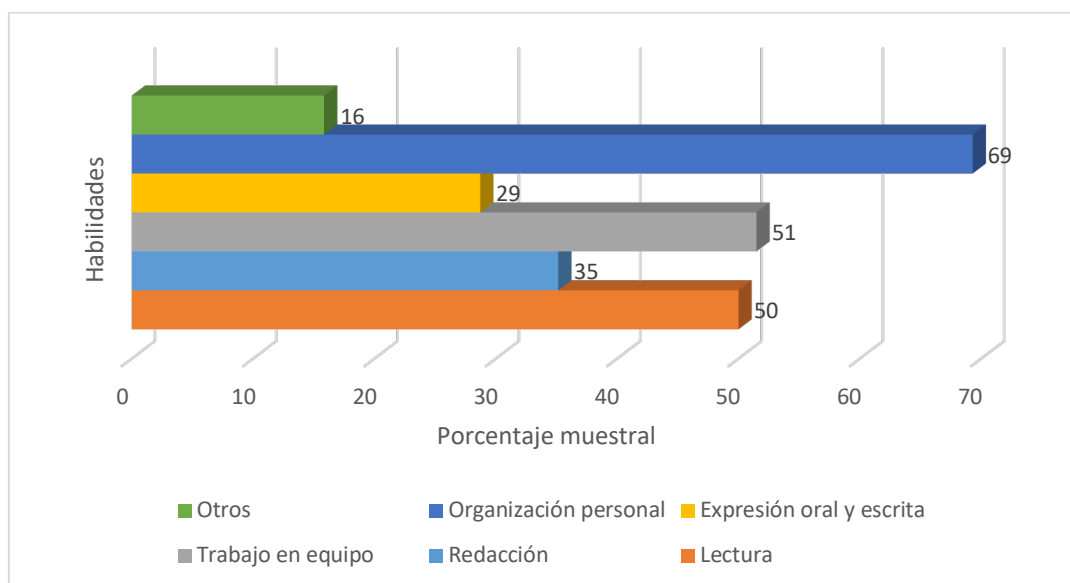
3) ¿A través de que herramientas tecnológicas adquiere conocimiento en su vida académica?



Análisis

Dado la figura, se da a entender que los estudiantes prefieren adquirir conocimiento a través de videos en la internet con un porcentaje considerablemente mayor, seguido por las páginas web con casi la misma cantidad de aprobación que los videos de internet y de la misma manera las aulas virtuales con una cantidad que sobrepasa la mitad de la muestra. Dejando como conclusión que el estudiante prefiere los videos en la red al igual que los sitios y páginas web. De modo que las aplicaciones como navegadores, buscadores, aplicaciones de video en la red están instaladas en los distintos dispositivos del alumnado. Demostrando que la adquisición del conocimiento no solo depende del docente, sino también de terceras aplicaciones que facilitan la información a los usuarios. Cabe mencionar que esta pregunta es de tipo de elección múltiple por lo que los porcentajes varían en valores altos. Independientemente de cumplir la sumatoria del 100 por ciento.

4) ¿Qué habilidades ha obtenido durante su proceso de aprendizaje utilizando medios digitales?

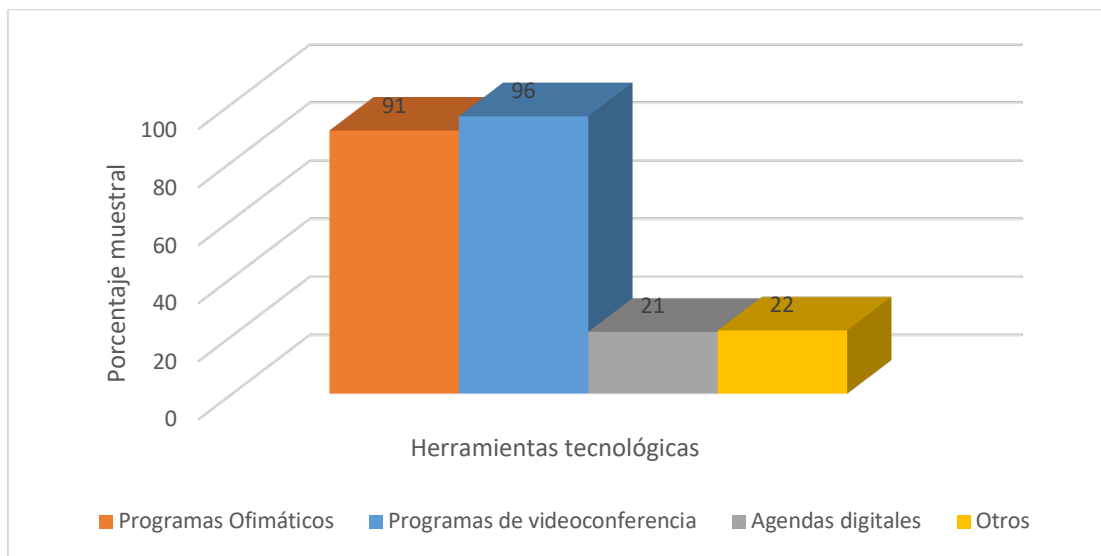


Análisis

Los resultados muestran que más de la mitad de la muestra ha adquirido la habilidad de organización personal, mientras que la otra mitad da a entender que el trabajo en equipo fue una habilidad que desarrollo con la utilización de medios digitales, seguido estrechamente con la lectura con un porcentaje similar al anterior resultado.

Se deduce que las herramientas tecnológicas son claves para la organización personal por parte de los estudiantes. Pues, da a entender que los usuarios tienen aplicaciones ya sea móvil, escritorio o web, que permitan la organización de actividades, recursos etc. Esto independientemente si las aplicaciones que tienen instaladas van orientadas específicamente a la organización.

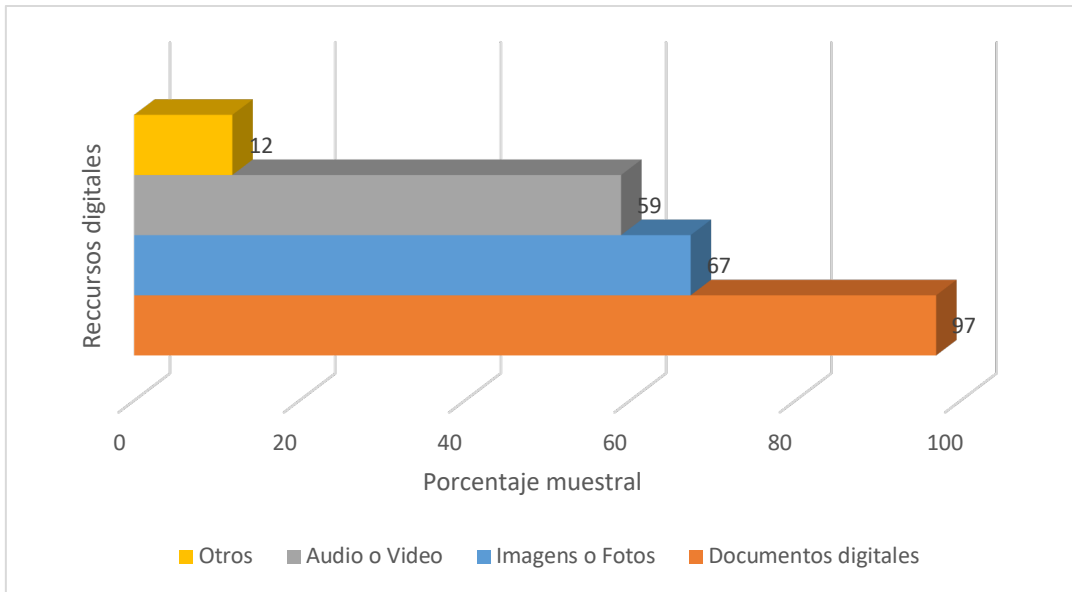
5) ¿Qué programas informáticos utiliza en su proceso de aprendizaje?



Análisis

Dado los resultados, casi la totalidad de la muestra resalta que utilizan programas de videoconferencia, y de la misma manera usan programas ofimáticos. Por cuarto lugar se encuentra el uso de agendas digitales, lo que da a considerar que posiblemente las aplicaciones ofimáticas (Word, Excel, PowerPoint, OneNote, etc.) y los programas de videoconferencia (Microsoft Teams, Google Meet, Zoom, etc.) son destinadas para trabajos estudiantiles, reuniones, clases e inclusive organización personal, debido al alto porcentaje que tuvo la anterior pregunta en lo que respecta a la misma.

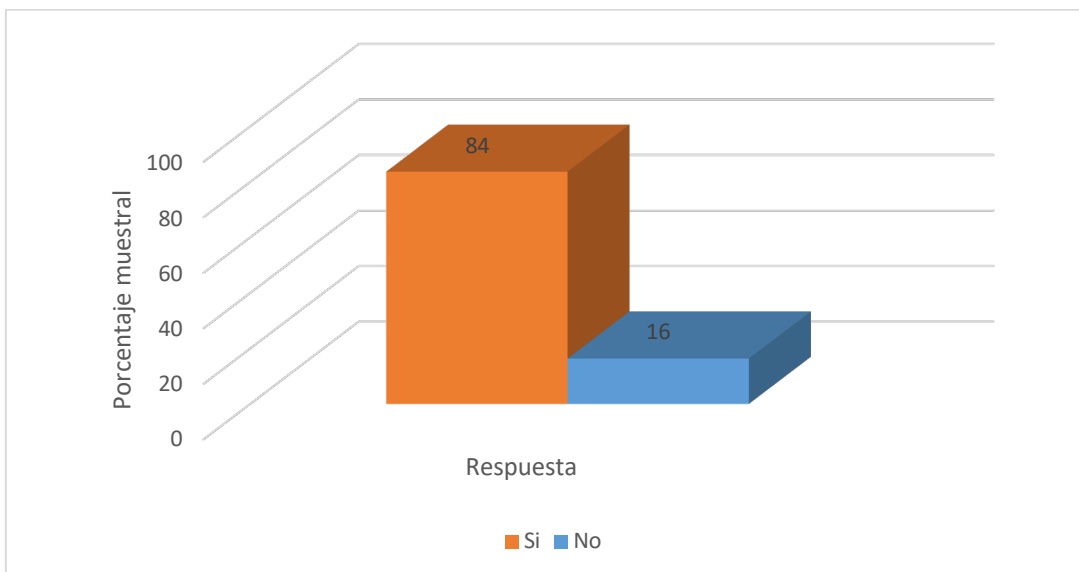
6) ¿Qué recursos digitales utiliza en un periodo académico ordinario?



Análisis

El gráfico hace énfasis a que la mayor parte, (casi la totalidad) de los encuestados utilizan documentos digitales como recursos. Y es evidente este resultado, debido al contexto de la pandemia que cada estudiante se encuentra hoy en día. Haciendo realce en la utilización de recursos digitales tales como fotos, audio, video, y documentos. De todo esto se da a entender que el estudiante maneja casi el 100% de sus recursos en documentos digitales, por lo que da la necesidad de poder organizarlos a través de una herramienta tecnológica.

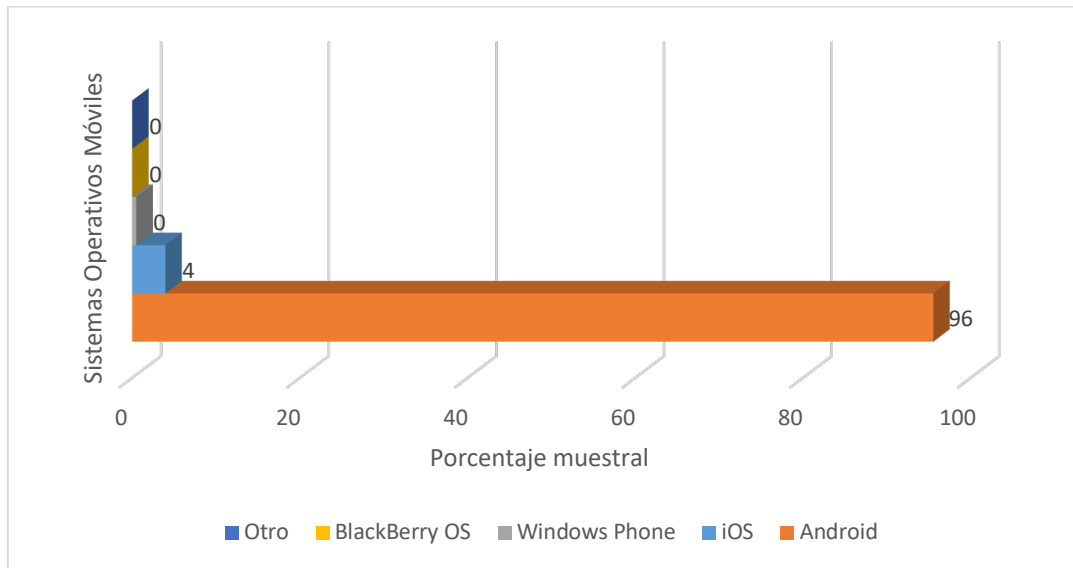
8) ¿Actualmente cuenta con un teléfono que soporta instalación de una APP?



Análisis

Dada la figura se establece que la mayor parte de los encuestados confirman que poseen un dispositivo que soporte la instalación de aplicaciones móviles. Deduciendo la factibilidad de una propuesta orientada a los dispositivos móviles.

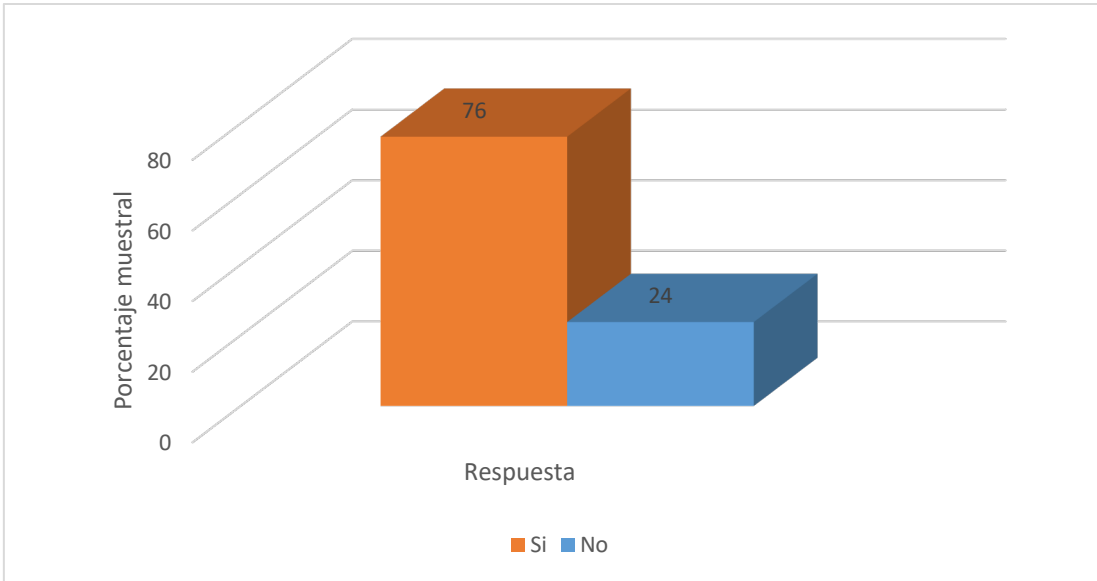
9) Escoja entre las opciones, el sistema operativo que tiene su celular actualmente



Análisis

Con los resultados obtenidos se puede afirmar que la mayor parte de los encuestados soportan la instalación de aplicaciones móviles en el sistema operativo Android. Mientras que la mínima parte soporta para aplicaciones orientadas al sistema iOS.

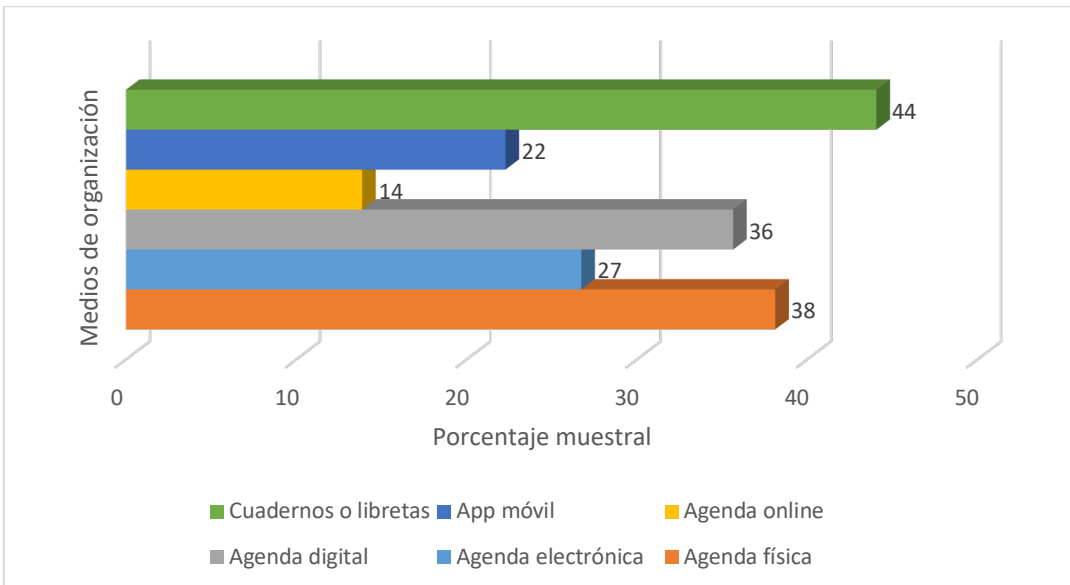
10) ¿Ha utilizado usted, algún un medio digital para la organización de sus actividades académicas y personales?



Análisis

Dada la figura se presenta los siguientes resultados. Con un porcentaje relativamente mayor se confirma por parte del alumnado de que utilizan medios digitales para la organización de sus respectivas actividades académicas y personales. Es decir que la mayor parte los estudiantes consideran importante la organización utilizando herramientas digitales, independientemente si son o no agendas digitales.

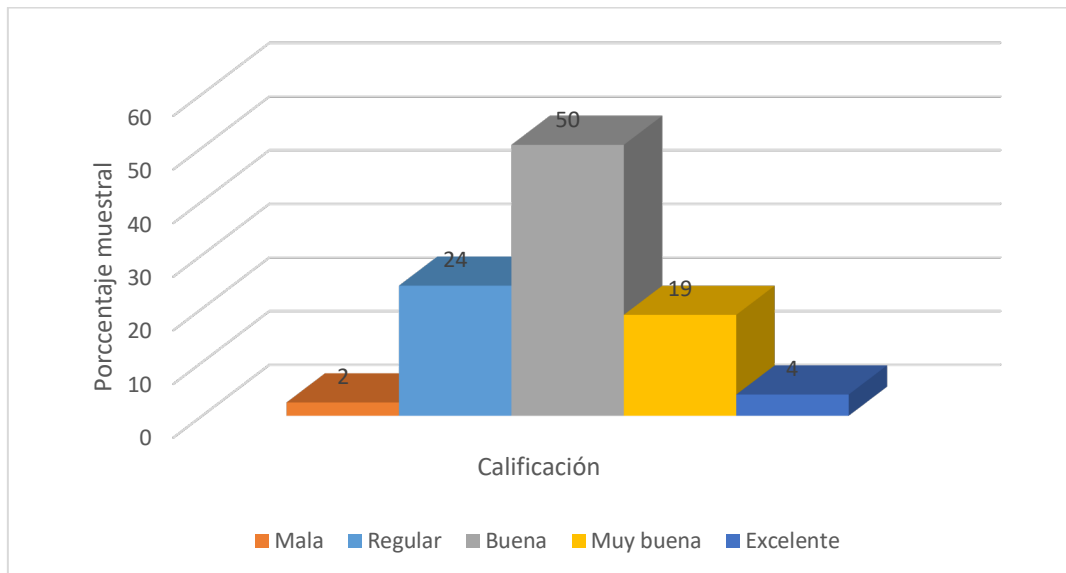
11) Escoja entre las opciones disponibles el tipo de organización que ha usado



Análisis

En función de los resultados presentados en la figura, un porcentaje menor a la mitad de los encuestados afirma que para su organización utilizan cuadernos o libretas, y de la misma manera se organizan por medio de agendas físicas, mientras que cerca de la mitad de la muestra usan agendas digitales. Estos porcentajes revela que aún se sigue utilizando medios físicos para su organización con un porcentaje relativamente considerable. Lo que a su vez podría traer problemas por las desventajas que trae consigo los medios físicos. Tales como perdida de información o recurso, ralentizando su organización personal.

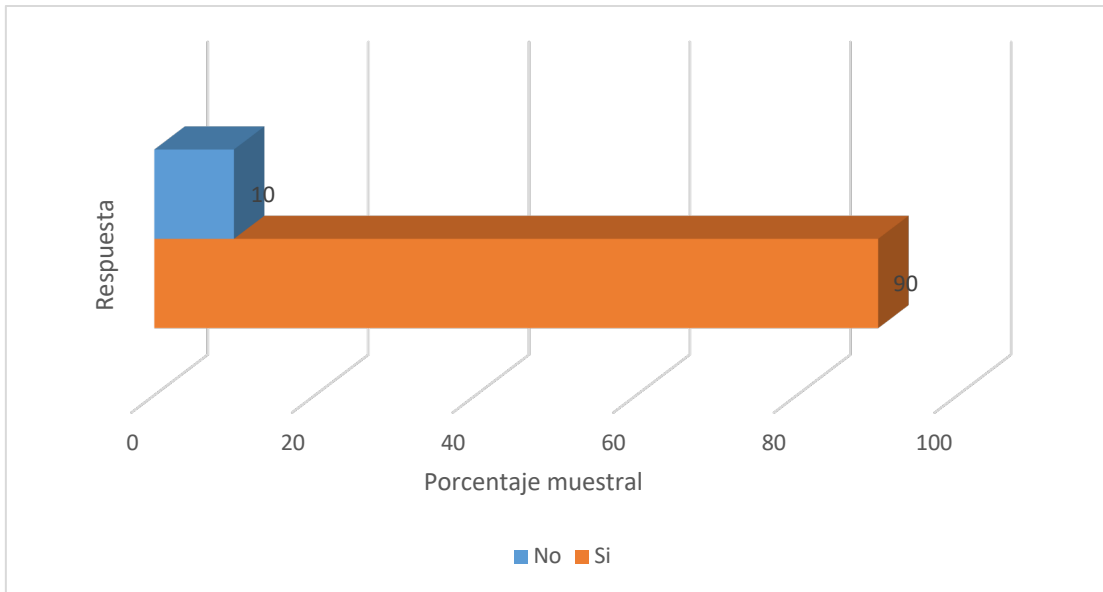
12) ¿Como califica el apoyo que le brinda la herramienta de organización personal y/o académico?



Análisis

El gráfico muestra que exactamente la mitad de los encuestados califican su tipo de organización como buena, mientras la mínima parte como regular. Estos porcentajes demuestran que la calificación a los medios de organización por parte del alumnado no sobrepasa la calificación de buena. Demostrando que las mismas herramientas no satisfacen todas sus necesidades pues no son calificados como excelente por su mayoría.

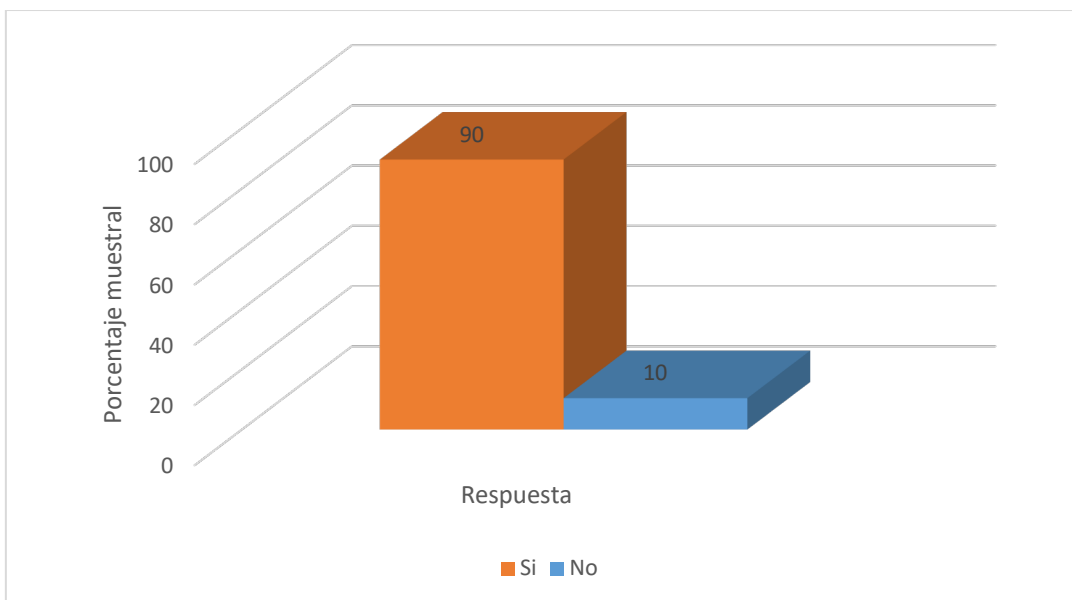
13) ¿Considera importante la habilidad de organización personal por medio de una agenda digital en su celular?



Análisis

Dado los resultados, se deduce que casi la totalidad de los encuestados resaltan la organización personal utilizando una agenda digital en su dispositivo móvil. Dando factible la posibilidad de desarrollo a una propuesta de orientada a la plataforma de tipo móvil.

14) ¿Estaría dispuesto a instalar una agenda digital gratuita en su celular?

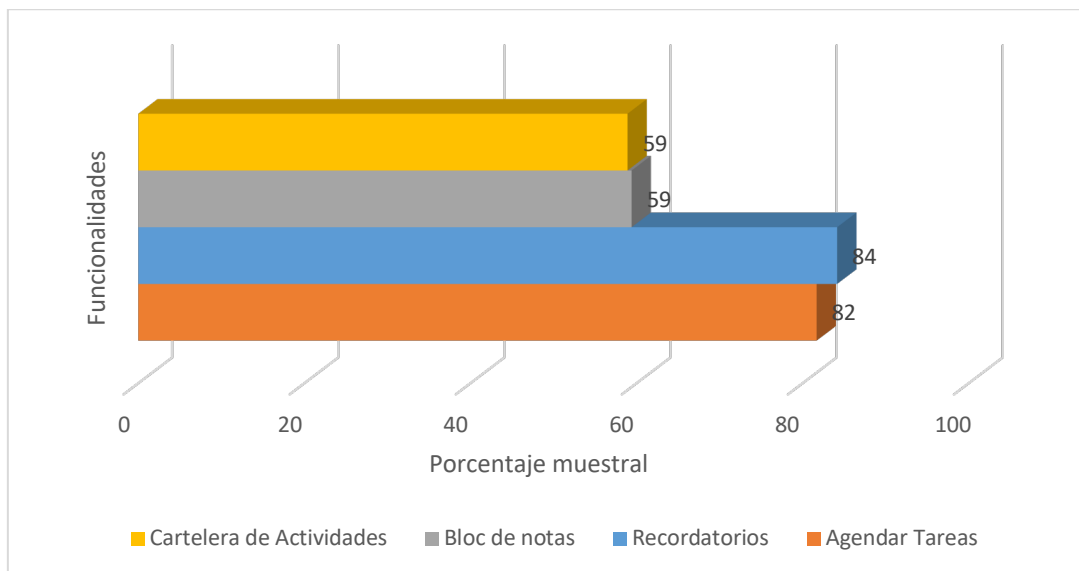


Análisis

Dado los resultados se afirma que el estudiante está dispuesto a instalar una agenda digital en su dispositivo móvil. Haciendo posible el desarrollo de una propuesta que satisfagan sus necesidades en cuanto a la organización personal. Cabe mencionar que

el término gratuito fue el punto clave para visualizar estos resultados presentados. Pues caso contrario podría haber reflejado cambios en los resultados. Al punto de que la respuesta negativa se considere mayoritaria.

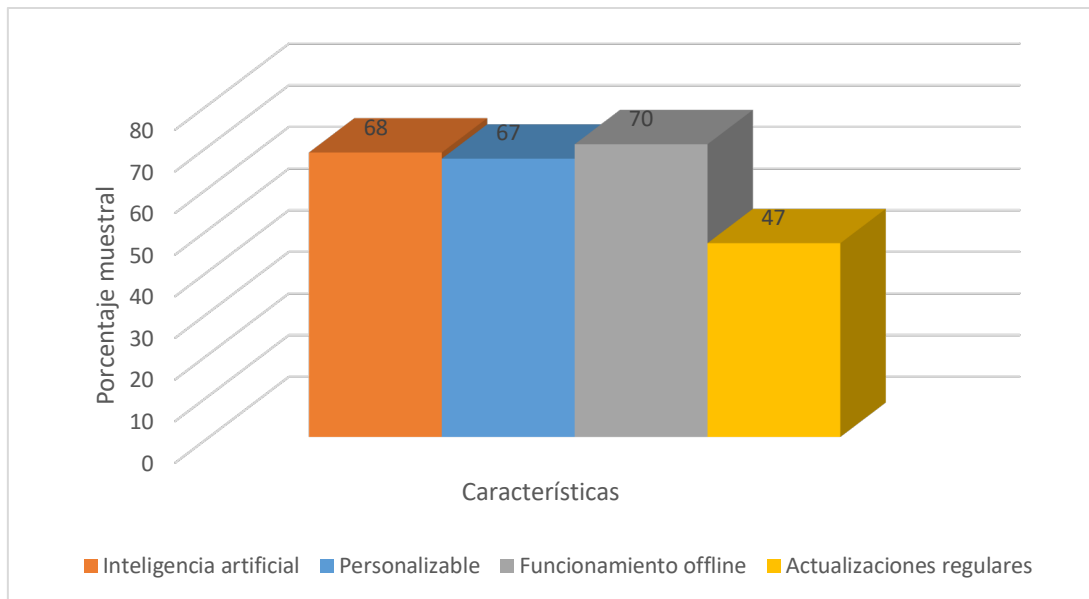
15) ¿Qué funcionalidades básicas consideraría para su agenda que satisfagan sus necesidades?



Análisis

En relación con la figura presentada, se resalta de que los recordatorios y las tareas agendadas son funcionalidades que se deben resaltar en cuanto a una agenda móvil. Si bien es cierto que la cartelera de actividades y el bloc de notas (registro de texto por parte del usuario) son funcionalidades que se debe considerar también, pues también juegan roles importantes para su utilización.

16) Que otras características en función de la tecnología actual consideraría implementar en una agenda digital.



Análisis.

Dado los resultados, otra de las características generales que se resalta es que una agenda móvil debe contener Inteligencia Artificial (Trae consigo funcionalidades como recomendación de libros digitales), Personalización (Que el usuario pueda decidir temas de diseño) y funcionamiento fuera de línea, pues no muchos poseen datos móviles para su funcionamiento online. Si bien es cierto que en ocasiones el dispositivo si accede a la red por diferentes medios. Por otra parte, es necesario resaltar que las actualizaciones regulares no son tan aceptadas por los estudiantes, pues esto puede deberse a problemas de estabilidad, errores, etc. En cuanto al software.

4.2 PROPUESTA

La presente propuesta se elaboró partiendo desde el análisis de los resultados obtenidos en la encuesta realizada a los estudiantes de la Carrera de Computación de la Universidad Politécnica Estatal del Carchi, en la que se determinó diferentes funcionalidades para una agenda académica. Inicialmente se realizó un estudio de factibilidad para demostrar que el diseño de esta solución tecnológica es posible, además se estableció la metodología de desarrollo de software XP (Programación Extrema), lo que ayudó a reducir la dificultad, organizar tareas, simplificar procesos y mejorar el resultado final de la aplicación.

4.2.1. ESTUDIO DE FACTIBILIDAD

4.2.1.1. Factibilidad Organizacional

- Aspectos generales de la organización.

- **Institución:** Universidad Politécnica Estatal del Carchi

- **Ubicación geográfica:** Av. Universitaria, Tulcán

- **Área:** Carrea de Computación

- **Sistema:** Agenda digital personalizada para la organización de actividades académicas

- **Objeto social:** Servicio Público

- **Misión:**

“Formar Ingenieros en Ciencias de la Computación capaces de proponer y generar soluciones tecnológicas innovadoras que contribuyan al desarrollo de la región, con sólidas bases académicas e investigativas, enmarcados en los aspectos ético, crítico y compromiso social.”

-**Visión:**

“Ser un referente regional en la formación de profesionales de Ingeniería en Ciencias de la Computación.”

4.2.1.2. Factibilidad Técnica

Para desarrollar el proyecto se elaboró una lista de recursos a utilizar, como hardware y software.

La agenda inteligente se desarrolló con Python, Flutter, Firebase, estos recursos open source no generan ningún costo en la investigación, y además contamos con los conocimientos necesarios para su desarrollo.

Tabla 8.
Recursos Software y Hardware

SOFTWARE			
Recurso	Versión	Descripción	Cantidad
Phyton	3.9.9	Lenguaje de Programación	1
Flutter	2.10.3	Framework Backend	1
Visual Studio Code		Editor de código	1
Firebase		Plataforma para el desarrollo de aplicaciones web y aplicaciones móviles	1
Git & Github	2.32.0	Sistema de control de versiones	1
TensorFlow (Keras)	2.8.0	API para redes neuronales	1
Pytesseract	0.3.9	OCR	1
Heroku	7.59.4	Servicio de Computación en la Nube	1
Google Maps Plattafom		API de Google Maps	1
Office	2019	Herramienta ofimática	2

HARDWARE		
Recurso	Descripción	Cantidad
Equipo de computación	HP EliteBook 820	2
	Dell Inspiron 13	
Celulares	Samsung A20S	2
	Huawei Y9 Prime	

Los investigadores cuentan con el equipamiento necesario para el desarrollo del proyecto, además de la conexión a internet que facilita la comunicación entre el equipo de trabajo, la organización cuenta actualmente con un servidor y la infraestructura de red necesaria, y la conclusión es que los recursos son adecuados y técnicamente factibles.

4.2.1.3. Factibilidad Económica

Para el presupuesto del presente proyecto se tomó en cuenta todos los recursos utilizados, tales como: software, hardware, recursos humanos y materiales de oficina.

Tabla 9.
Factibilidad Económica

Descripción	Cantidad	Costo Real	Costo referencial
Costos Software			
Phyton	1	00,00	00,00
Flutter	1	00,00	00,00
Visual Studio Code	1	00,00	00,00
Firebase	1	00,00	00,00
Git & Github	1	00,00	00,00
TensorFlow (Keras)	1	00,00	00,00
Pytesseract	1	00,00	00,00
Heroku	1	00,00	00,00
Google Maps Plataforma	1	00,00	00,00
Office	2	00,00	00,00
Total de Software	11	00,00	00,00
Costos Hardware			
Equipos de computación	2	00,00	1380,00
Celulares	2	00,00	500,00
Total de Hardware	4	00,00	1880,00
Talento Humano			
Programadores	2	00,00	00,00
Total de Talento Humano	2	00,00	00,00
Materiales de Oficina			
Internet		164,00	164,00
Materiales de Oficina		30,00	30,00
Varios		20,00	20,00
Total de Materiales de Oficina		214,00	214,00
Subtotal		214,00	2094,00
10% Imprevistos		21,40	209,40
Total		235,40	2303,40

4.2.1.4. Factibilidad Operativa

- **Situación actual**

En la Carrera de Computación de la Universidad Politécnica Estatal del Carchi, los estudiantes utilizan cuadernos y agendas físicas para llevar sus registros de sus actividades académicas, lo que genera retrasos en el cumplimiento de las mismas al no tener un registro y acompañamiento confiable en su proceso educativo.

- **Situación ideal**

La agenda digital con herramientas de inteligencia artificial permitirá al estudiante estar informado de sus actividades académicas de forma oportuna, además de contar con una herramienta personalizada que le brindará información de la Carrera y la Universidad.

4.2.2. Metodología XP

4.2.2.1. Fase de planificación

En esta fase inicial correspondiente a la metodología XP se estableció los roles que desempeñan los miembros de este proyecto, así como también los tiempos equivalentes a las herramientas utilizadas. Además, se detallan los diferentes módulos en el cual se compone la aplicación y finalmente las respectivas historias de usuario en conjunto con las tareas de usuario que marcan los responsables de los requerimientos funcionales establecidos por el cliente, que en este caso son los estudiantes. Cabe resaltar que los investigadores al ser considerados estudiantes de la carrera de computación, también se toma en cuenta sus requerimientos, así como los de terceras personas correspondientes a la carrera.

- **Roles.**

Tabla 10.
Roles del proyecto

Nombre	Descripción	Rol XP
Msc. Jorge Miranda	Docente tutor	Consultor
Malkik Anrango	Investigador	Programador
Noreen Rosero	Investigador	Programador
Estudiantes de computación	Usuario Final	Cliente

- **Estimación de tiempo.**

Tabla 11.
Estimación de tiempo

Estimación	Días	Horas
0,2 semana	1	4
0,4 semana	2	8
1 semana	5	20
1,6 semana	8	32
2 semanas	10	40
2,4 semanas	12	48
2,6 semanas	13	52
3 semanas	15	60
4 semanas	20	80

- **Módulos del sistema.**

1. Modulo Usuario

- a. Submódulo Resumen

- i. Reporte semanal

- b. Submódulo Noticias

- i. Sección Noticias

- ii. Sección Favoritos

- c. Submódulo Notas

- i. Control de notas por asignatura (incluye texto, audio y fotos)

- d. Submódulo Tareas

- i. Control de tareas por asignatura

- ii. Notificaciones

- e. Submódulo Recordatorio

- i. Controles recordatorios

- ii. Notificaciones

- f. Submódulo Horario de clases

- i. Asignaturas
 - ii. Docentes
- g. Submódulo Calendario
 - i. Registro y visualización de tarea, calendario o recordatorio
- h. Submódulo Comunidad
 - i. Publicaciones
 - ii. Creación publicaciones
- i. Submódulo Configuración
 - i. Privacidad
 - ii. Apariencia
 - iii. Cuenta o perfil
 - iv. Información
 - v. Gestión de Horario
 - vi. General
- j. Submódulo Herramientas
 - i. Scanner
 - ii. OCR (Optical Character Recognition)
 - iii. Lector QR

- **Historias de usuario.**

Tabla 12.

Historia de usuario 1

HISTORIA DE USUARIO	
Número: 1	Usuario: Estudiantes
Nombre Historia: Login	
Prioridad: Alto	Riesgo: Alto
Puntos Estimados: 0,4	Iteración: 1
Programador Responsable: Malkik Anrango	
Descripción: El usuario final o estudiante deberá acceder a la agenda por medio de una cuenta de usuario, el mismo que puede ser creado con Gmail. El campo de inicio de sesión deberá contener lo siguiente.	
<ul style="list-style-type: none"> • Un formulario con un campo de correo y otro para ingresar la contraseña. • Deberá contener 2 botones, el primero hará referencia a ingresar por medio de las credenciales ingresadas por el usuario, el segundo y el tercer botón permitirá iniciar sesión 	

por Gmail. Cabe mencionar que existirá un texto con su respectivo hipervínculo que permita crear una cuenta, compuestos con los campos, nombre, usuario, correo, contraseña.

Observaciones: Ninguna

Tabla 13.

Historia de usuario 2

HISTORIA DE USUARIO	
Número: 2	Usuario: Estudiantes
Nombre Historia: Resumen	
Prioridad: Media	Riesgo: Alto
Puntos Estimados: 1	Iteración: 2
Programador Responsable: Noreen Rosero	
Descripción: El usuario final o estudiante una vez accedido a la agenda con las respectivas credenciales, accederá a la pantalla de resumen, las cuales visualizarán los eventos, recordatorios, o tareas pendientes para el día presente, así como también para los días siguientes, en función de la semana que se encuentre. Cada día estará compuesto por tarjetas que listará los eventos para la fecha correspondiente de la misma tarjeta. Es decir que cada tarjeta estará compuesta de lo siguiente.	
<ul style="list-style-type: none">• Un título que detalle el día que corresponde, seguido a la esquina por la fecha (día, mes y año). En el contenido de la tarjeta se enlistará los eventos compuestos por el nombre del evento y una pequeña descripción.	
Observaciones: Cada tarjeta será en función de la semana que se encuentre el estudiante.	

Tabla 14.

Historia de usuario 3

HISTORIA DE USUARIO	
Número: 3	Usuario: Estudiantes
Nombre Historia: Sección Noticias	
Prioridad: Media	Riesgo: Alto
Puntos Estimados: 1	Iteración: 1
Programador Responsable: Malkik Anrango	
Descripción: El estudiante también podrá acceder a una sección de noticias, en el cual se visualizarán las novedades o noticias generadas por la universidad. Cada noticia se conformará por una tarjeta. Por lo que cada tarjeta se compone de lo siguiente:	
<ul style="list-style-type: none">• Un título que permita describir de manera general la noticia, así como también un foto o imagen, la fecha, y una breve descripción de esta.• Deberá existir un campo de texto, al igual que un botón que	

permita la búsqueda de la noticia en cuestión.

Observaciones: Se considera la implementación de un enlace para dirigir a la fuente original, que correspondería a la página principal de la universidad. Es opcional de los programadores si implementar los “me gusta” por cada noticia.

Tabla 15.

Historia de usuario 4

HISTORIA DE USUARIO	
Número: 4	Usuario: Estudiantes
Nombre Historia: Control Notas o apuntes	
Prioridad: Alta	Riesgo: Alto
Puntos Estimados: 1	Iteración: 1
Programador Responsable: Noreen Rosero	
Descripción: El estudiante podrá acceder a una sección de notas, en la que estará organizado por las asignaturas correspondientes al horario de clases. Correspondería a un cuaderno digital. Cabe mencionar que existirá aparte una clasificación denominada otros, pues no correspondería a las materias del estudiante. Esta sección estará compuesta de la siguiente manera	
<ul style="list-style-type: none">• Pestañas correspondientes al nombre de la asignatura u otra clasificación, de modo que al presionar se enlisten las respectivas notas creadas, con los campos correspondientes a título de la nota, su clasificación (asignatura), una breve descripción y la fecha de la nota.• De la misma manera deberá permitir la eliminación, edición o creación de la nota.• Debe contener un campo de texto y un botón para la funcionalidad de búsqueda.• Cada nota debe permitir el ingreso de texto, audio o foto en caso de que sea necesario para el estudiante.	
Observaciones: Cada nota podría componerse por tarjetas.	

Tabla 16.

Historia de usuario 5

HISTORIA DE USUARIO	
Número: 5	Usuario: Estudiantes
Nombre Historia: Control Tareas	
Prioridad: Alta	Riesgo: Alto
Puntos Estimados: 4	Iteración: 1
Programador Responsable: Malkik Anrango	
Descripción: Deberá existir la sección de tareas que permita la organización de estas en función a la asignatura correspondiente. Permitiendo el ingreso de texto o imagen. Su estructura será similar a la sección de notas. Pues se dividirá por pestañas correspondientes al nombre de la asignatura, de modo que se	

enlistarán en tarjetas, detallando, el nombre de la tarea, su asignatura, su fecha y breve descripción. Permitiendo acciones como edición o eliminación de estas. Esta sección tendrá las siguientes características y condiciones

- No deberá permitir crear la tarea si no se ha seleccionado una asignatura, en caso de no existir, se deberá crearla.
- Deberá crear notificaciones en función de la fecha de entrega de la tarea.
- Los campos que se visualizan en la tarjeta de tarea son los mismos para la creación.

Observaciones: Cada tarea podría componerse por tarjetas o listas. Además, cada nota debe permitir o no la opción de notificar al estudiante sobre alguna tarea que aproxime la fecha de entrega

Tabla 17.
Historia de usuario 6

HISTORIA DE USUARIO	
Número: 6	Usuario: Estudiantes
Nombre Historia: Recordatorios	
Prioridad: Alta	Riesgo: Alto
Puntos Estimados: 1	Iteración: 1
Programador Responsable: Noreen Rosero	
Descripción: Deberá existir la sección de recordatorio en donde su similitud es parecida a la sección de tareas. Con la diferencia que no viene clasificado por asignaturas. A continuación, se describe lo que compone esta sección:	
<ul style="list-style-type: none">• Se deberá visualizar los distintos recordatorios en forma de lista, organizado por la fecha desde la que más se aproxima a la fecha actual hasta las posteriores fechas que se aproximen.• Se debe detallar en la lista la fecha, el nombre del recordatorio y la descripción.• En la lista deben estar 2 botones que permitan la edición y eliminación.• Deberá existir un botón flotante para la creación, en donde se compondrá de los mismos campos que se visualizan en la lista.• Deberá crear notificaciones en función de la fecha de recordatorio	
Observaciones: Cada tarea debe permitir la opción de notificar al estudiante sobre alguna tarea que aproxime la fecha de entrega en función de la hora que se haya establecido.	

Tabla 18.
Historia de usuario 7

HISTORIA DE USUARIO	
Número: 7	Usuario: Estudiantes
Nombre Historia: Horario de clases	
Prioridad: Alta	Riesgo: Alto
Puntos Estimados: 1	Iteración: 2
Programador Responsable: Malkik Anrango	
<p>Descripción: En otro apartado deberá existir un horario de clases en forma estructural de tabla, de manera que los días laborales de la semana correspondan a las columnas (puede incluir sábados) y las filas sean las horas o el orden de las filas. Además, deberá contener las siguientes características</p> <ul style="list-style-type: none"> • Poder crear varios horarios de clase identificado por un nombre asignado por el usuario. • Debe existir un botón en la parte esquina superior que permita cambiar de horario de clases, entre los ya creados por el estudiante.} • Al momento de seleccionar una celda del horario de clases, se deberá ingresar una asignatura, El formulario de creación de la asignatura se lo detalla en la historia de usuario 12. 	
<p>Observaciones: Al momento de crear las asignaturas, deben diferenciarse por colores personalizados por el usuario</p>	

Tabla 19.
Historia de usuario 8

HISTORIA DE USUARIO	
Número: 8	Usuario: Estudiantes
Nombre Historia: Control Asignaturas	
Prioridad: Alta	Riesgo: Alto
Puntos Estimados: 1	Iteración: 2
Programador Responsable: Noreen Rosero	
<p>Descripción: En el mismo contexto del horario de clases (Historia de usuario 11). El control de la asignatura estará compuesto de lo siguiente:</p> <ul style="list-style-type: none"> • Al momento de seleccionar la celda se deberá desplegar un formulario con los siguientes campos: el nombre de la asignatura, el docente, el salón de clases, el o los días de clases, así como también el inicio y fin de esta y finalmente el color que diferenciará a la materia. • Adicionalmente debe existir el botón de añadir y otro de cancelar la creación de la asignatura. • Deberá permitir la creación o la selección de un docente (Detallado en la historia de usuario 13). • En otro apartado es considerable la implementación de una vista que visualice la lista de las asignaturas ya creadas, de 	

modo que, al seleccionar, debe visualizarse el resumen del horario semanal, eventos, tareas, notas, etc. Deberá existir en esta sección un botón flotante que permita la creación de esta.

Observaciones: Al momento de crear las asignaturas, deben diferenciarse por colores personalizados por el usuario

Tabla 20.
Historia de usuario 9

HISTORIA DE USUARIO	
Número: 9	Usuario: Estudiantes
Nombre Historia: Control Docentes	
Prioridad: Alta	Riesgo: Alto
Puntos Estimados: 1	Iteración: 2
Programador Responsable: Malkik Anrango	
Descripción: El control de los registros de docente estará compuesto de lo siguiente:	
<ul style="list-style-type: none">• El formulario de creación docente deberá contener lo siguiente: Nombre, Apellido, Teléfono, Correo electrónico, y foto, cada campo tendrá su icono de identificación y campo de texto para el ingreso de los datos• En otro apartado se enlistarán los registros de los docentes existentes, las cuales al seleccionar debe visualizar a detalle los datos correspondientes al profesor, así como también existir dos botones, uno para la edición y eliminación• La lista visualizada de docentes deberá visualizar el nombre y la foto del docente, en caso de que no exista, establecer uno por defecto.• Finalmente debe existir un botón flotante para la creación de registro de docente.	
Observaciones: Es opcional la función de favoritos en docentes.	

Tabla 21.
Historia de usuario 10

HISTORIA DE USUARIO	
Número: 10	Usuario: Estudiantes
Nombre Historia: Calendario	
Prioridad: Alta	Riesgo: Alto
Puntos Estimados: 1	Iteración: 2
Programador Responsable: Noreen Rosero	
Descripción: Este apartado iniciaría como título con la fecha correspondiente al día, mes y año. De modo que se compondrá de lo siguiente:	
<ul style="list-style-type: none">• Una tabla similar o igual a los calendarios convencionales que muestran los días de la semana y los números	

correspondientes a los mismos.

- La celda debe seleccionarse automáticamente al día presente.
- En relación con la celda seleccionada del día del mes, se deberá detallar en forma de listas, las tareas, recordatorios e inclusive el horario de clases.
- Cada elemento de la lista debe contener un botón que permita elegir entre editar, eliminar o cambiar a estado de terminado el evento.
- Se debe permitir de manera fácil la navegación del calendario.
- Debe existir un botón que sitúe inmediatamente al día presente.

Observaciones: Ninguna

Tabla 22.

Historia de usuario 11

HISTORIA DE USUARIO

Número: 11

Usuario: Estudiantes

Nombre Historia: Comunidad o cartelera

Prioridad: Alta

Riesgo: Alto

Puntos Estimados: 3

Iteración: 2

Programador Responsable: Malkik Anrango

Descripción: En lo que respecta a la comunidad, esta sección permitirá al estudiante generar publicaciones y visualizarlas. Esto con el objetivo de crear una cartelera digital dedicada al estudiante.

- Inicialmente debe existir un botón flotante que permita crear una publicación que contendrá lo siguiente: El título, contenido de la publicación, categoría de publicación (estará preestablecido), el número de contacto y una opción de añadir foto. En donde cada campo tendrá su respectiva casilla de ingreso de datos. Adicionalmente dos botones que permita aceptar o cancelar el proceso.
- Las publicaciones visualizarán en estructura de tarjetas por cada una. Detallando el usuario, la fecha de subida, categoría, la foto y/o descripción de la publicación, así como también un botón o hipervínculo que dirección al contacto de WhatsApp (Esto queda opcional del usuario al momento de crear la publicación).
- Se considera la implementación de conteo de “me gustas” por publicación.

Observaciones: Ninguna

Tabla 23.
Historia de usuario 12

HISTORIA DE USUARIO	
Número: 12	Usuario: Estudiantes
Nombre Historia: Privacidad	
Prioridad: Media	Riesgo: Media
Puntos Estimados: 1	Iteración: 2
Programador Responsable: Noreen Rosero	
Descripción: Dentro del apartado de configuración existirá una sección denominada privacidad, en donde se establecerá que datos serán visibles hacia la comunidad, pues cada usuario del software podrá acceder al perfil público. De modo que se compondrá de lo siguiente. Una lista, compuesta de checklist, de manera que el visto permita seleccionar la visibilidad, caso contrario debería estar la casilla en blanco.	
Observaciones: Ninguna	

Tabla 24.
Historia de usuario 13

HISTORIA DE USUARIO	
Número: 13	Usuario: Estudiantes
Nombre Historia: Apariencia	
Prioridad: Alta	Riesgo: Alto
Puntos Estimados: 1	Iteración: 2
Programador Responsable: Malkik Anrango	
Descripción: La apariencia contendrá el tema personalizable de la agenda como tal. Pues se deberá conceder una gama de colores, de modo que, el usuario pueda elegir el que más le agrada. La opción de los colores vendrá representada en una lista, de modo que cada ítem de la lista contendrá un icono del color al que pertenece con su respectivo nombre. Cada ítem será seleccionable. Agregando, además, un botón de aceptar o cancelar de acuerdo con la selección.	
Observaciones: Ninguna	

Tabla 25.
Historia de usuario 14

HISTORIA DE USUARIO	
Número: 14	Usuario: Estudiantes
Nombre Historia: Cuenta o perfil	
Prioridad: Alta	Riesgo: Alto
Puntos Estimados: 1	Iteración: 2
Programador Responsable: Noreen Rosero	
Descripción: En lo que respecta a la cuenta o perfil, deberá	

permitir el cambio de foto, nombre y contraseña, con las respectivas medidas de seguridad. De manera que se representará por lo siguiente.

- Se visualizará en la ventana, la foto, el nombre de usuario, correo, la carrera al que pertenece y la contraseña (el texto de la contraseña será visualizada de manera censurada). Cada campo anteriormente mencionado debe permitir la edición de esta, ya sea por selección o un botón.
- Al momento de la edición. Se deberá desplegar una ventana emergente que permita realizar el proceso de edición. Conteniendo dos botones, el de aceptar y cancelar.

Observaciones: Ninguna

Tabla 26.

Historia de usuario 15

HISTORIA DE USUARIO	
Número: 15	Usuario: Estudiantes
Nombre Historia: Información	
Prioridad: Baja	Riesgo: Baja
Puntos Estimados: 0,4	Iteración: 2
Programador Responsable: Malkik Anrango	
Descripción: La información será el apartado de visualizar datos relativamente importantes como la versión de la agenda y acerca de los desarrolladores de esta.	
<ul style="list-style-type: none">• Estos datos se visualizarán en un párrafo, encabezado por el logo y versión del software.	
Observaciones: Ninguna	

Tabla 27.

Historia de usuario 16

HISTORIA DE USUARIO	
Número: 16	Usuario: Estudiantes
Nombre Historia: Herramientas	
Prioridad: Alta	Riesgo: Alta
Puntos Estimados: 4	Iteración: 2
Programador Responsable: Noreen Rosero	
Descripción: El apartado herramientas darán la posibilidad de tener un conversor de imagen a texto además de un escáner de modo que permita al estudiante facilitar su proceso académico digital.	
<ul style="list-style-type: none">• Se visualizará en una lista las herramientas disponibles al usuario, seguido por el nombre y un icono de simbología a la herramienta.• Cada ítem de la lista será seleccionable, por lo que inmediatamente deberá abrirse la funcionalidad de esta.	

- En cuanto al escaneo, deberá abrirse la cámara del dispositivo de modo que, al capturar la imagen, esta se arregle automáticamente, de manera que la foto sea comprensible y no distorsionada. El usuario poseerá dos botones, el de aceptar o el de cancelar. Dado el caso que se acepte, deberá permitir la conversión en PDF y posteriormente compartir hacia otras aplicaciones que pueden ser mensajería instantánea o correo electrónico.
- En cuanto al OCR, deberá al igual que la anterior, abrir la cámara permitiendo así capturar la imagen, existiendo dos botones (aceptar y cancelar). En caso de ser presionado cancelar. El mismo deberá mostrar en una ventana desplegable el texto ya convertido, permitiendo así copiar, crear notas, recordatorios, tareas o inclusive publicaciones con el texto convertido.

Observaciones: Ninguna

- **Tareas de ingeniería.**

Tabla 28.

Tarea de usuario 1

TAREA DE USUARIO	
Número de tarea: 1	Numero de Historia: 1
Nombre Tarea: Programar la vista de inicio de sesión y validación	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 19/12/2021	Fecha Fin: 21/12/2021
Programador Responsable: Malkik Anrango	
Descripción: Programar la vista de inicio de sesión, así como la respectiva validación de los campos.	

Tabla 29.

Tarea de usuario 2

TAREA DE USUARIO	
Número de tarea: 2	Numero de Historia: 1
Nombre Tarea: Programar la vista de registro de usuario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 25/12/2021	Fecha Fin: 29/12/2021
Programador Responsable: Noreen Rosero	
Descripción: Programar la vista de registrarse y de la misma manera la respectiva validación y anexo con la Base de datos en Firebase.	

Tabla 30.
Tarea de usuario 3

TAREA DE USUARIO	
Número de tarea: 3	Numero de Historia: 2
Nombre Tarea: Programar la vista resumen	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 28/2/2021	Fecha Fin: 29/12/2021
Programador Responsable: Malkik Anrango	
Descripción: Programar la vista de Resumen. Esto se procede luego de crear los demás modelos correspondientes a las respectivas tareas y/o historias de usuarios	

Tabla 31.
Tarea de usuario 4

TAREA DE USUARIO	
Número de tarea: 4	Numero de Historia: 3
Nombre Tarea: Programar la vista Noticias	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 30/12/2021	Fecha Fin: 31/12/2021
Programador Responsable: Noreen Rosero	
Descripción: Programar la vista de Noticias de modo que se permita la renderización de los datos hacia la misma. Adicionando las acciones de visualización y búsqueda	

Tabla 32.
Tarea de usuario 5

TAREA DE USUARIO	
Número de tarea: 5	Numero de Historia: 4
Nombre Tarea: Crear Modelos de datos de Notas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 03/01/2022	Fecha Fin: 05/01/2022
Programador Responsable: Malkik Anrango	
Descripción: Programar los modelos de Notas en Python. Haciendo posible el consumo de servicios CRUD.	

Tabla 33.
Tarea de usuario 6

TAREA DE USUARIO	
Número de tarea: 6	Numero de Historia: 4
Nombre Tarea: Programar la vista de Notas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 06/01/2022	Fecha Fin: 07/01/2022
Programador Responsable: Malkik Anrango	

Descripción: Programar la vista de Notas que permitan de modo que permitan la renderización a la misma. De la misma manera adicionar las acciones de visualización, eliminación, edición, creación y búsqueda de la misma. Implementando una lógica de interacción.

Tabla 34.

Tarea de usuario 7

TAREA DE USUARIO	
Número de tarea: 7	Numero de Historia: 5
Nombre Tarea: Crear Modelos de datos de Tarea	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 10/01/2022	Fecha Fin: 21/01/2022
Programador Responsable: Noreen Rosero	
Descripción: Programar los modelos de Tareas en Python. Haciendo posible el consumo de servicios CRUD.	

Tabla 35.

Tarea de usuario 8

TAREA DE USUARIO	
Número de tarea: 8	Numero de Historia: 5
Nombre Tarea: Programar la vista de Tareas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 22/01/2022	Fecha Fin: 31/01/2022
Programador Responsable: Malkik Anrango	
Descripción: Programar la vista de Tareas que permitan de modo que permitan la renderización a la misma. De la misma manera adicionar las acciones de visualización, eliminación, edición, creación y búsqueda de esta. Seguirá una lógica de organización correspondiente a las asignaturas existentes.	

Tabla 36.

Tarea de usuario 9

TAREA DE USUARIO	
Número de tarea: 9	Numero de Historia: 6
Nombre Tarea: Programar la vista de Recordatorios	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 01/02/2022	Fecha Fin: 02/02/2022
Programador Responsable: Noreen Rosero	
Descripción: Programar la vista de Tareas que permitan de modo que permitan la renderización a la misma. De la misma manera adicionar las acciones de visualización, eliminación, edición, creación y búsqueda de esta. Cabe mencionar que no existirá anexo a la BD. Si no que se usará el almacenamiento del	

dispositivo.

Tabla 37.

Tarea de usuario 10

TAREA DE USUARIO	
Número de tarea: 10	Numero de Historia: 7
Nombre Tarea: Programar la vista de Horario de clases	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 03/02/2022	Fecha Fin: 05/02/2022
Programador Responsable: Malkik Anrango	
Descripción: Programar la vista de Horario de clases (tabla), agregado la lógica de agregar asignaturas correspondientes a cada celda de la tabla. Además, de la incorporación de crear varios horarios de clases.	

Tabla 38.

Tarea de usuario 11

TAREA DE USUARIO	
Número de tarea: 11	Numero de Historia: 8
Nombre Tarea: Crear Modelo de datos asignatura	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 07/02/2022	Fecha Fin: 08/02/2022
Programador Responsable: Noreen Rosero	
Descripción: Crear modelo de datos en Flutter de asignatura	

Tabla 39.

Tarea de usuario 12

TAREA DE USUARIO	
Número de tarea: 12	Numero de Historia: 8
Nombre Tarea: Codificar la vista asignaturas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 09/02/2022	Fecha Fin: 10/02/2022
Programador Responsable: Malkik Anrango	
Descripción: Programar la vista de asignaturas, que permitan generar acciones de listar, crear, editar, eliminar	

Tabla 40.
Tarea de usuario 13

TAREA DE USUARIO	
Número de tarea: 13	Numero de Historia: 8
Nombre Tarea: Codificar vistas adicionales de asignaturas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 11/02/2022	Fecha Fin: 12/02/2022
Programador Responsable: Noreen Rosero	
Descripción: Programar las vistas adicionales de asignaturas, que complementen las acciones anteriormente mencionadas en la tarea de usuario 21.	

Tabla 41.
Tarea de usuario 14

TAREA DE USUARIO	
Número de tarea: 14	Numero de Historia: 9
Nombre Tarea: Crear Modelo de datos docentes	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 14/02/2022	Fecha Fin: 15/02/2022
Programador Responsable: Malkik Anrango	
Descripción: Crear modelo de datos en Flutter de docentes.	

Tabla 42.
Tarea de usuario 15

TAREA DE USUARIO	
Número de tarea: 15	Numero de Historia: 9
Nombre Tarea: Codificar la vista docente	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 15/02/2022	Fecha Fin: 16/02/2022
Programador Responsable: Noreen Rosero	
Descripción: Programar la vista de docentes, que permitan generar acciones de listar, crear, editar, eliminar	

Tabla 43.
Tarea de usuario 16

TAREA DE USUARIO	
Número de tarea: 16	Numero de Historia: 9
Nombre Tarea: Codificar vistas adicionales de docentes	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 17/02/2022	Fecha Fin: 18/02/2022
Programador Responsable: Malkik Anrango	
Descripción: Programar las vistas adicionales de asignaturas, que complementen las acciones anteriormente mencionadas en la tarea de usuario 24.	

Tabla 44.*Tarea de usuario 17*

TAREA DE USUARIO	
Número de tarea: 17	Numero de Historia: 10
Nombre Tarea: Codificar vista Calendario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 21/02/2022	Fecha Fin: 23/02/2022
Programador Responsable: Noreen Rosero	
Descripción: Programar la vista de calendario, agregando la lógica organizativa de los eventos, tareas o recordatorios establecidos para cada fecha del día.	

Tabla 45.*Tarea de usuario 18*

TAREA DE USUARIO	
Número de tarea: 18	Numero de Historia: 11
Nombre Tarea: Crear modelo de datos Publicaciones y usuario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 24/02/2022	Fecha Fin: 26/02/2022
Programador Responsable: Malkik Anrango	
Descripción: Programar el modelo de datos en Flutter, además de su correspondiente migración hacia Firebase.	

Tabla 46.*Tarea de usuario 19*

TAREA DE USUARIO	
Número de tarea: 19	Numero de Historia: 11
Nombre Tarea: Codificar la vista Publicaciones y Comunidad o Cartelera	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 28/02/2022	Fecha Fin: 06/03/2022
Programador Responsable: Noreen Rosero	
Descripción: Programar la vista Publicaciones y Cartelera, que permita crear, editar, visualizar y eliminar las respectivas publicaciones de la cartelera. Es decir, se agregarán operaciones CRUD.	

Tabla 47.*Tarea de usuario 20*

TAREA DE USUARIO	
Número de tarea: 20	Numero de Historia: 11
Nombre Tarea: Codificar la vista perfil de usuario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 07/03/2022	Fecha Fin: 13/03/2022
Programador Responsable: Malkik Anrango	
Descripción: Programar la vista de perfil de usuario, agregando la visualización de las publicaciones creadas por cada usuario.	

Tabla 48.*Tarea de usuario 21*

TAREA DE USUARIO	
Número de tarea: 21	Numero de Historia: 12
Nombre Tarea: Codificar la vista Privacidad	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 14/03/2022	Fecha Fin: 16/03/2022
Programador Responsable: Noreen Rosero	
Descripción: Programar la vista de Privacidad, agregando la visualización y cambio de estado de los datos de usuario que se podrán visualizar a la comunidad.	

Tabla 49.*Tarea de usuario 22*

TAREA DE USUARIO	
Número de tarea: 22	Numero de Historia: 13
Nombre Tarea: Codificar la vista Apariencia	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 16/03/2022	Fecha Fin: 18/03/2022
Programador Responsable: Malkik Anrango	
Descripción: Programar la vista de Apariencia, agregando la visualización y cambio de temas estableciendo una gama alta de colores. Los cambios se establecen, con un botón creado para guardar y aplicar.	

Tabla 50.
Tarea de usuario 23

TAREA DE USUARIO	
Número de tarea: 23	Numero de Historia: 14
Nombre Tarea: Codificar la vista Perfil o cuenta y validación	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 18/03/2022	Fecha Fin: 19/03/2022
Programador Responsable: Noreen Rosero	
Descripción: Programar la vista de Cuenta o perfil, agregando las acciones de cambio de nombre, contraseña, correo y foto. Y su respectiva validación de los campos	

Tabla 51.
Tarea de usuario 24

TAREA DE USUARIO	
Número de tarea: 24	Numero de Historia: 15
Nombre Tarea: Codificar la vista Información	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 19/03/2022	Fecha Fin: 20/03/2022
Programador Responsable: Malkik Anrango	
Descripción: Programar la vista de Información. Agregando datos relevantes a la versión del software y los creadores de esta.	

Tabla 52.
Tarea de usuario 25

TAREA DE USUARIO	
Número de tarea: 25	Numero de Historia: 16
Nombre Tarea: Codificar el backend para OCR	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 21/03/2022	Fecha Fin: 23/03/2022
Programador Responsable: Noreen Rosero	
Descripción: Aplicar algoritmos de inteligencia artificial utilizando TensorFlow para OCR	

Tabla 53.
Tarea de usuario 26

TAREA DE USUARIO	
Número de tarea: 26	Numero de Historia: 16
Nombre Tarea: Codificar la vista OCR	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 24/03/2022	Fecha Fin: 25/12/2021
Programador Responsable: Malkik Anrango	
Descripción: Programar las respectivas vistas del conversor de	

imagen a texto. Consumiendo la API creada.

Tabla 54.

Tarea de usuario 27

TAREA DE USUARIO	
Número de tarea: 27	Numero de Historia: 16
Nombre Tarea: Codificar el backend para el scanner	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 26/03/2022	Fecha Fin: 27/03/2022
Programador Responsable: Noreen Rosero	
Descripción: Programar el backend del scanner haciendo uso de algoritmos de visión artificial.	

Tabla 55.

Tarea de usuario 28

TAREA DE USUARIO	
Número de tarea: 28	Numero de Historia: 16
Nombre Tarea: Codificar la vista scanner	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,4
Fecha Inicio: 28/03/2022	Fecha Fin: 01/03/2022
Programador Responsable: Malkik Anrango	
Descripción: Programar las respectivas vistas del scanner, consumiendo la API, anteriormente creada.	

- **Estimación de tareas de usuario.**

Tabla 56.

Estimación de tareas de usuario

Nombre Historia	N.º Tarea	Tarea	Tiempo estimado		
			Semanas	Días	Horas
Control de usuarios	1	Codificar la vista control de usuario	1	5	20
	2	Renderizar datos de usuario	0,4	2	8
Control de noticias	3	Codificar la vista control de noticias	1	5	20
		Renderizar datos de noticia	0,4	2	8
Control Cartelera	4	Codificar la vista control de Cartelera	1	5	20
	5	Renderizar datos de Cartelera o Comunidad	0,4	2	8

Login	6	Programar la vista de inicio de sesión y validación	0,4	2	8
	7	Programar la vista de registro de usuario	0,4	2	8
Resumen	8	Programar la vista resumen	0,4	2	8
Sección Noticias	9	Programar la vista noticias	0,4	2	8
Control Notas o apuntes	10	Crear Modelos de datos de Notas	0,4	2	8
	11	Programar la vista de Notas	0,4	2	8
Control Tareas	12	Crear Modelos de datos de Tarea	0,4	2	8
	13	Programar la vista de Tareas	0,4	2	8
Control Recordatorios	14	Programar la vista de Recordatorios	0,4	2	8
Horario de clases	15	Programar la vista de Horario de clases	0,4	2	8
Control Asignaturas	16	Crear Modelo de datos asignatura	0,4	2	8
	17	Codificar la vista asignaturas	1	5	20
	18	Codificar vistas adicionales de asignaturas	0,4	2	8
Control Docentes	19	Crear Modelo de datos docentes	0,4	2	8
	20	Codificar la vista docente	1	5	20
	21	Codificar vistas adicionales de docentes	0,4	2	8
Calendario	22	Codificar vista Calendario	0,4	2	8
Comunidad o Cartelera	23	Crear modelo de datos Publicaciones y usuario	0,4	2	8
	24	Codificar la vista Publicaciones y Comunidad o Cartelera	1	5	20
	25	Codificar la vista perfil de usuario	0,4	2	8
Privacidad	26	Codificar la vista	0,4	2	8

Privacidad					
Apariencia	27	Codificar la vista Apariencia	0,4	2	8
Cuenta o perfil	28	Codificar la vista Perfil o cuenta y validación	0,4	2	8
Información	29	Codificar la vista Información	0,4	2	8
Herramientas	30	Codificar el backend para OCR	0,4	2	8
	31	Codificar la vista OCR	0,4	2	8
	32	Codificar el backend para el scanner	0,4	2	8
	33	Codificar la vista scanner	0,4	2	8
Total, de tiempo estimado			16,8	84	344

- **Plan de entrega del proyecto.**

Tabla 57.

Plan de entrega del proyecto

Módulo	Submódulo	Nro.	Nombre de la historia	Calendario estimado			Iteración asignada		Entrega asignada	
				Semanas estimadas	Días estimados	Horas estimadas	1	2	1	2
Usuario o estudiante	Resumen	1	Login	0,4	2	8	X		X	
		2	Resumen	1	5	20		X		X
	Noticias	3	Sección Noticias	1	5	20	X		X	
		4	Control Notas o apuntes	1	5	20	X		X	
	Tareas	5	Control Tareas	4	20	80	X		X	
	Recordatorios	6	Control Recordatorios	1	5	20	X		X	
	Horario de clases	7	Horario de clases	1	5	20		X		X
		8	Control Asignaturas	1	5	20		X		X
		9	Control Docentes	1	5	20		X		X
	Calendario	10	Calendario	1	5	20		X		X
	Comunidad	11	Comunidad o Cartelera	3	15	60		X		X
	Configuración	12	Privacidad	1	5	20		X		X
		13	Apariencia	1	5	20		X		X
		14	Cuenta o perfil	1	5	20		X		X
	Herramientas	15	Información	0,4	2	8		X		X
		16	Herramientas	4	20	80		X		X
Total, de semanas				22,8	114	456				

4.2.2.2. Fase de diseño

- Diseño Base de Datos

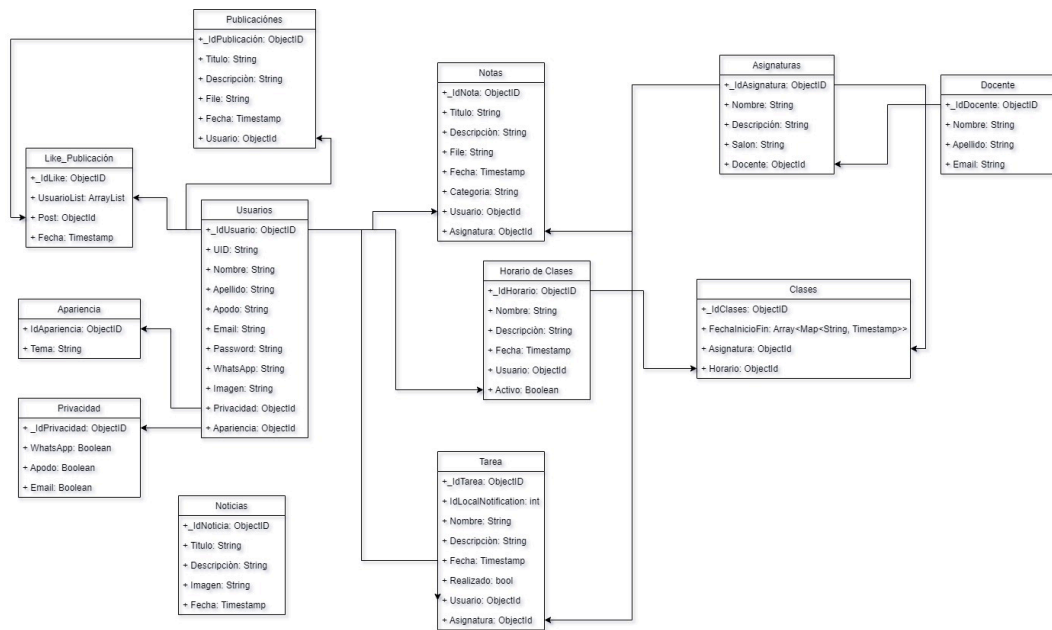


Figura 18. Diseño de Base de Datos



Figura 19. Inicio de Sesión

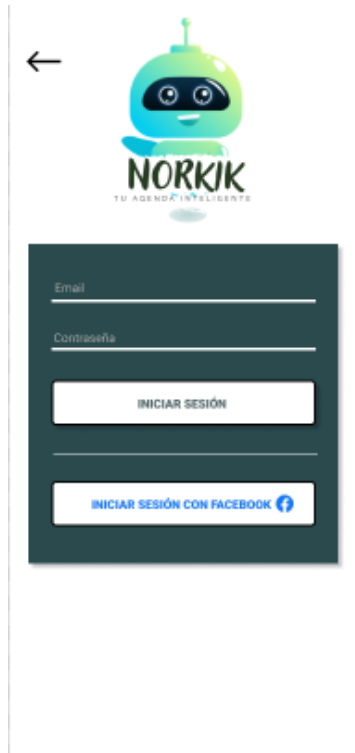


Figura 20. Login

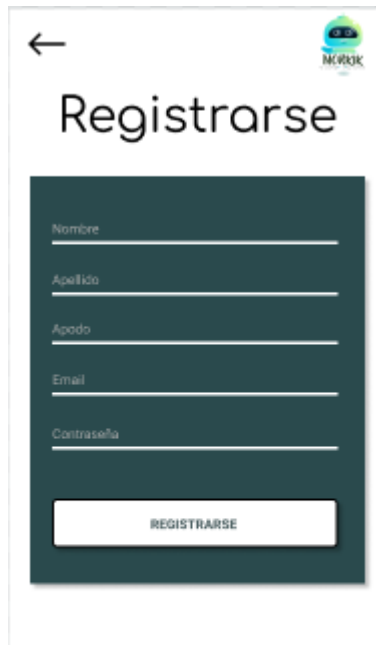


Figura 21. Registro



Figura 22. Menú late



Figura 23. Resumen



Figura 24. Noticias

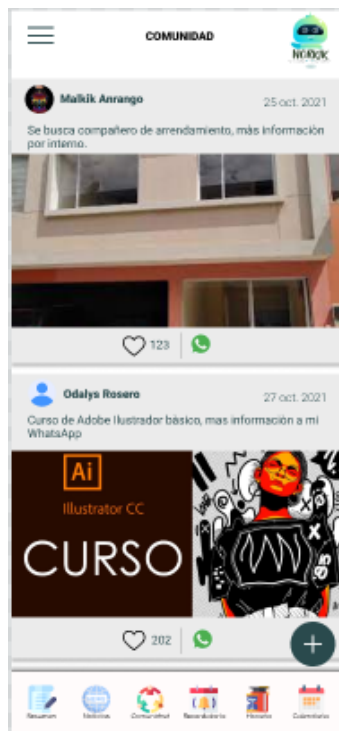


Figura 25. Comunidad

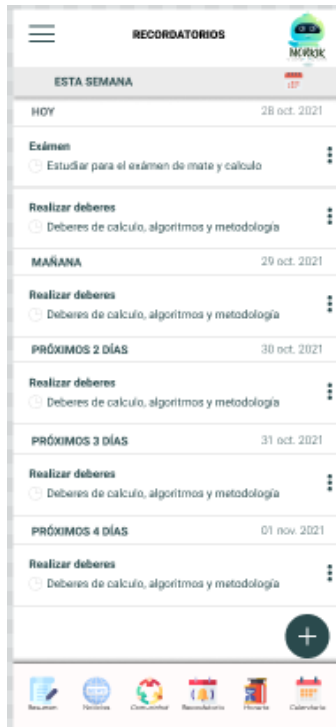


Figura 26. Recordatorios

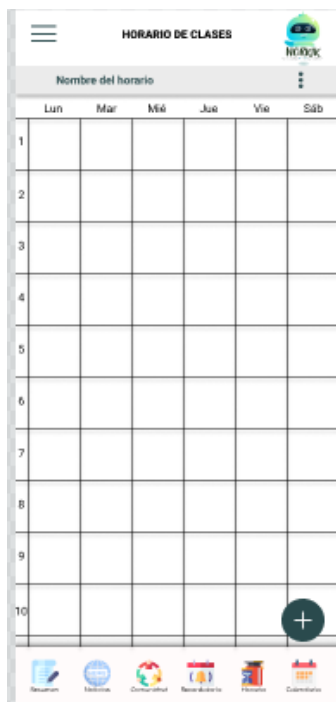


Figura 27. Horario de Clases

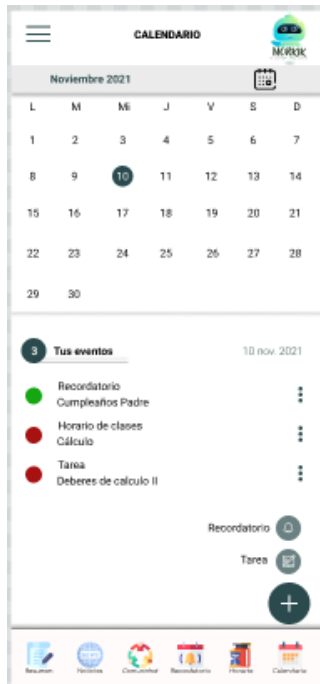


Figura 28. Calendario



Figura 29. Herramientas

- **Diagramas de Casos de Uso**

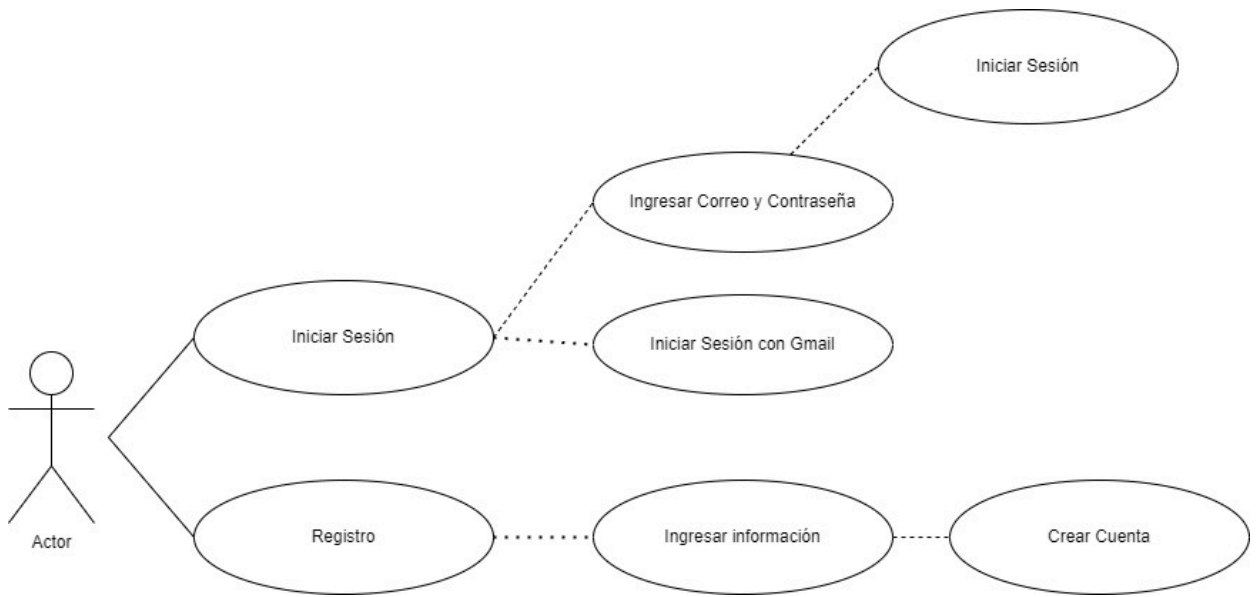


Figura 30. Caso de Uso Login

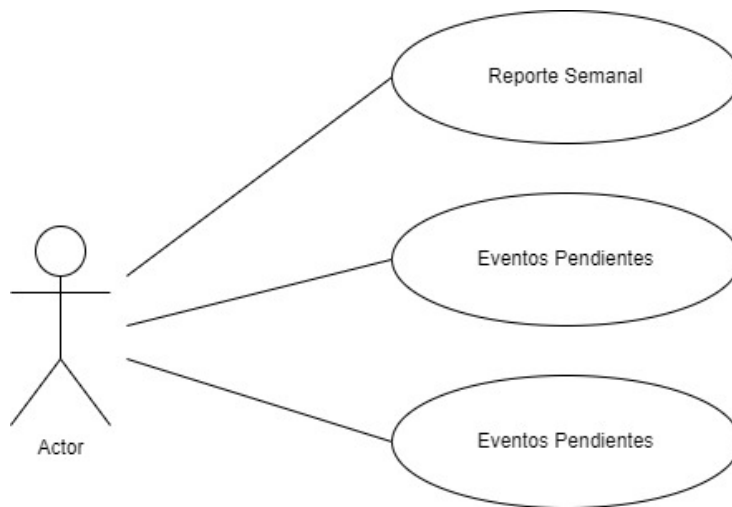


Figura 31. Caso de Uso Resumen

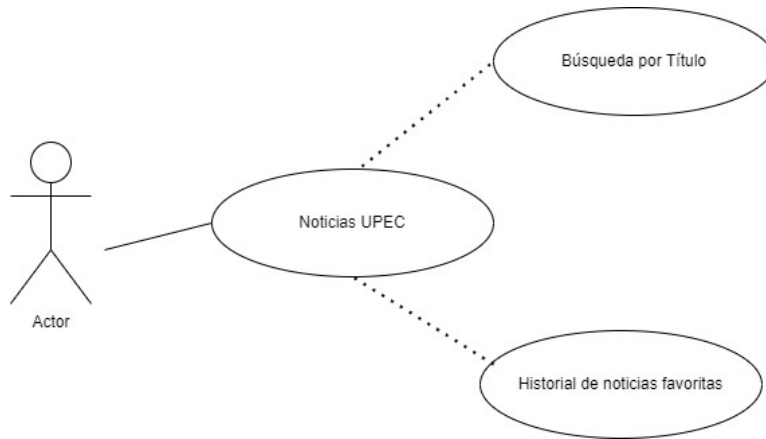


Figura 32. Casos de Uso Noticias

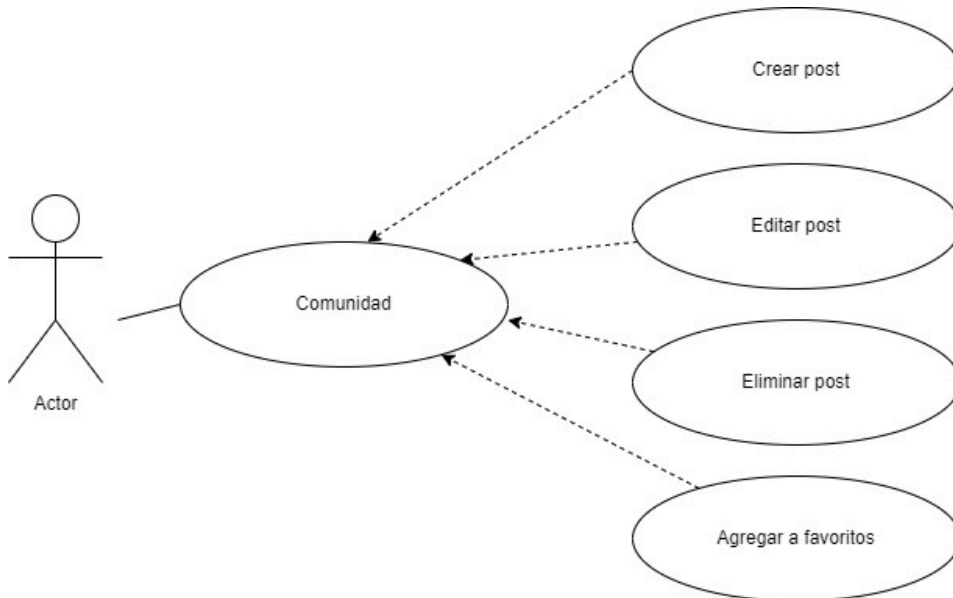


Figura 33. Caso de Uso Comunidad

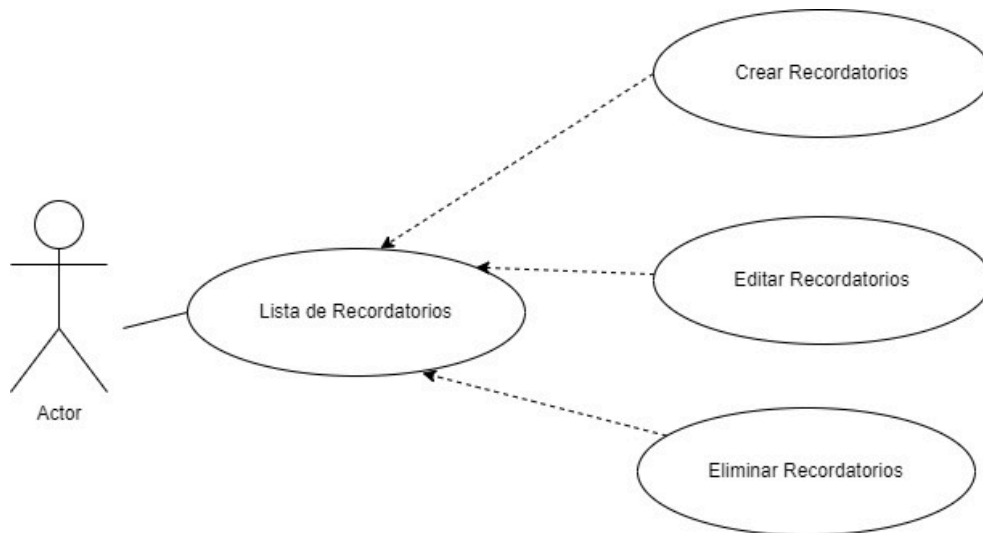


Figura 34. Caso de Uso Recordatorios

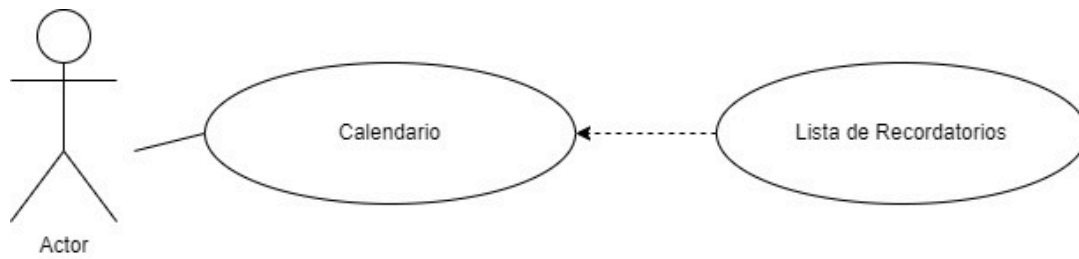


Figura 35. Caso de Uso Calendario

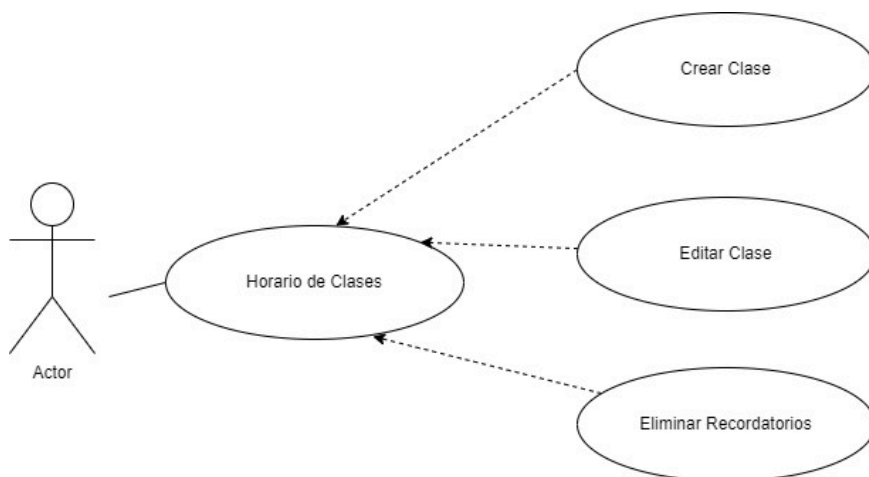


Figura 36. Caso de Uso Horario de Clases

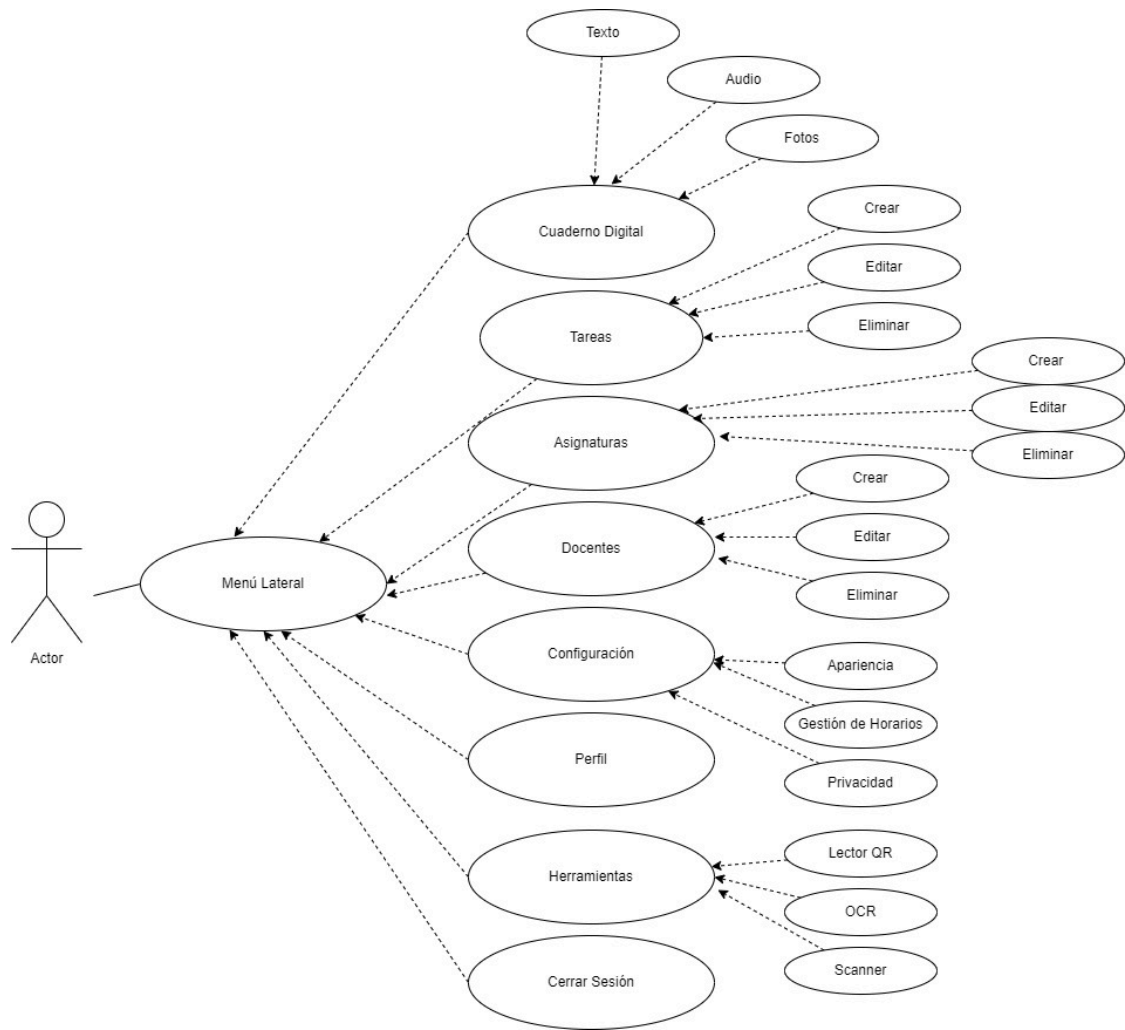


Figura 37. Caso de Uso Menú Lateral

4.2.2.3. Fase de codificación

El desarrollo de la aplicación móvil se partió por las buenas prácticas de desarrollo. Por una parte, se implementó el MVC (Modelo Vista Controlador), de modo que las siguientes gráficas, visualizan la estructura de archivos compuesta por este modelo que equivale a la estructura de las colecciones de la base de datos.

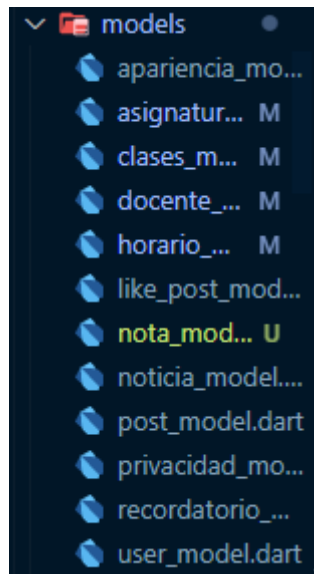


Figura 38. Estructura

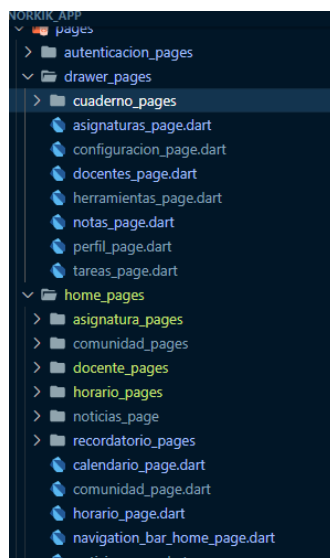


Figura 39. Estructura Interfaz

Mientras que el controlador implementa la conexión entre la vista y los modelos, específicamente con la base de datos, implementando los distintos controladores para que la conexión sea exitosa.

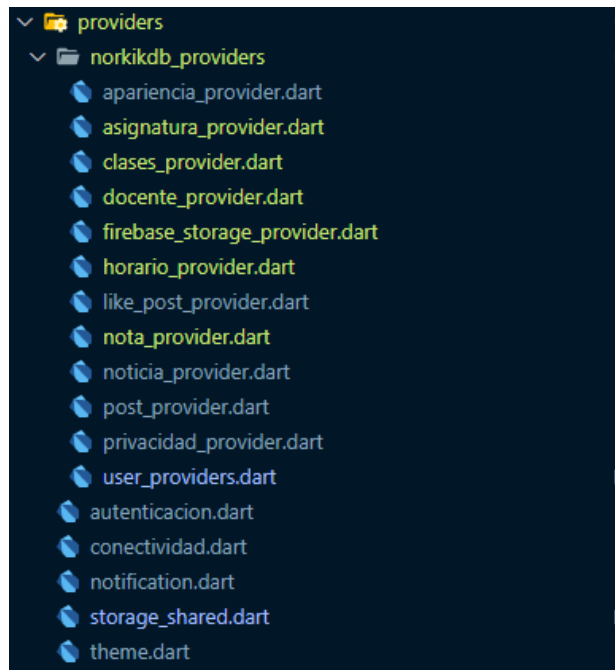


Figura 40. Controlador

También existe otros elementos como las rutas, que establecen la navegación en las vistas, al igual que los 'utils' que permiten implementar herramientas o widgets pequeños en el desarrollo:



Figura 41. Elementos Rutas

Inicialmente el software se compone de absolutamente de widgets, las cuales permiten estructurar la aplicación, de manera que se compone en un árbol de widgets. El widget principal parte del archivo main, en donde el mismo inicia la ejecución del programa.

```

Future<void> main() async {
  // WidgetsFlutterBinding.ensureInitialized();
  // await Firebase.initializeApp();

  return runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    tz.initializeTimeZones();
    final _initApp = Firebase.initializeApp();
    return FutureBuilder(
      future: _initApp,
      builder: (context, snapshot) {
        if (snapshot.hasError) {
          return ErrorWidget(Norkik());
        } else if (snapshot.hasData) {
          return MultiProvider(
            providers: [
              ChangeNotifierProvider(create: (_) => ConnectionStatusModel()),
              ChangeNotifierProvider(create: (_) => NotificationProvider()),
              ChangeNotifierProvider(
                create: (_) => ThemeChanger(getNorkikTheme()),
              ),
              ChangeNotifierProvider<AuthProvider>.value(
                value: AuthProvider(),
              ),
              StreamProvider.value(
                value: AuthProvider().user, initialData: null,
              ),
              ChangeNotifierProvider(
                create: (_) => UserProvider(UserModel.userModelNoData()),
              ),
            ],
            child: NorkikApp(),
          );
        }
      },
    );
  }
}

```

Figura 42. Main

Esto seguido por la estructura del software. A continuación, se visualiza el código de la autenticación y registro del software:

```

@override
State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  TextEditingController _emailController = TextEditingController();
  TextEditingController _passwordController = TextEditingController();
  final _formKey = GlobalKey<FormState>();
  @override
  void dispose() {
    _emailController.dispose();
    _passwordController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    final loginProvider = Provider.of<AuthProvider>(context);
    final usuarioProvider = Provider.of<UserProvider>(context);
    return Scaffold(
      body: SafeArea(
        child: SingleChildScrollView(
          child: Padding(
            padding: EdgeInsets.all(20),
            child: Form(
              key: _formKey,
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  IconButton(
                    onPressed: () {
                      Navigator.pop(context);
                    },
                    icon: Icon(Icons.arrow_back_ios), // IconButton
                    getLogoNorkikWithTheme(context, 200),
                  ),
                  Padding(
                    padding: EdgeInsets.symmetric(vertical: 5),

```

Figura 43. Autenticación y Registro del Software

Las vistas principales del software parten por el menú inferior, en donde se encuentra el resumen, noticias, comunidad, recordatorios, horario de clases y calendario: En donde cada parte del código se especifica a continuación:

```
Widget _cardReporteChart(context) {
  return Card(
    margin: EdgeInsets.symmetric(vertical: 10),
    elevation: 5,
    shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(5.0)),
    child: Column(
      children: [
        ListTile(
          horizontalTitleGap: 5,
          minLeadingWidth: 5,
          minVerticalPadding: 18,
          leading: Icon(
            Icons.bar_chart_rounded,
          ), // Icon
          title: Text('Reporte semanal'),
        ), // ListTile
        ChartsLineal(),
        SizedBox(
          height: 0.15,
          child: Container(
            color: Theme.of(context)
              .textTheme
              .bodyText2!
              .color!
              .withOpacity(0.5),
          ), // Container // SizedBox
        ListTile(
          horizontalTitleGap: 11,
          minLeadingWidth: 0,
          visualDensity: VisualDensity(horizontal: 0, vertical: -4),
          title: Text('Mostrar más'),
          leading: Icon(
            Icons.arrow_forward_rounded,
          ), // Icon
        ), // ListTile
      ],
    ),
  );
}
```

Figura 44. Resumen

```
import 'package:norkik_app/providers/norkikdb_providers/noticia_provider.dart';
import 'package:intl/intl.dart';
import 'package:norkik_app/providers/storage_shared.dart';

class NoticiasPage extends StatefulWidget {
  const NoticiasPage({Key? key}) : super(key: key);

  @override
  State<NoticiasPage> createState() => _NoticiasPageState();
}

class _NoticiasPageState extends State<NoticiasPage>
  with TickerProviderStateMixin {
  List<NoticiaModel> listNoticias = [];
  NoticiaProvider noticiaProvider = NoticiaProvider();
  final _scrollController = ScrollController();
  bool isLoading = false;
  bool isLoadingBottom = false;
  int _currIndexIcon = 0;

  final TextEditingController _textEditingController = TextEditingController();
  AnimationController? _animationController;

  StorageShared storageShared = StorageShared();
  List<String> listFavoritos = [];

  @override
  void initState() {
    super.initState();
    if (mounted) {
      _cargarListFavoritos();
      _agregarNoticias();
    }
  }

  _scrollController.addListener(() {
    if (_scrollController.position.pixels ==
        _scrollController.position.maxScrollExtent) {
      if (_textEditingController.text == '') {
        _agregarMasNoticias();
      }
    }
  });
}
```

Figura 45. Noticias

```

import 'package:norkik_app/providers/storage_shared.dart';
import 'package:flutter_speed_dial/flutter_speed_dial.dart';

class ComunidadPage extends StatefulWidget {
  const ComunidadPage({Key? key}) : super(key: key);

  @override
  State<ComunidadPage> createState() => _ComunidadPageState();
}

class _ComunidadPageState extends State<ComunidadPage>
  with TickerProviderStateMixin {
  bool selectedIconButtonLike = true;

  List<LikePostModel> listLikePost = [];
  List<bool> listIsFavoriteIcon = [];

  LikePostProvider likePostProvider = LikePostProvider();
  final _scrollController = ScrollController();
  bool isLoading = false;
  bool isLoadingBottom = false;
  bool isLoadingFavorite = false;
  int _currIndexIcon = 0;
  bool _isWidgetSearch = false;

  final TextEditingController _textEditingController = TextEditingController();
  AnimationController? _animationController;
  GlobalKey _scaffoldKey = GlobalKey();

  @override
  void initState() {
    super.initState();
    if (mounted) {
      _agregarLikePost();
    }
    _scrollController.addListener(() {
      if (_scrollController.position.pixels ==
          _scrollController.position.maxScrollExtent) {

```

Figura 46. Comunidad

```

import 'package:norkik_app/providers/storage_shared.dart';
import 'package:norkik_app/utils/alert_temp.dart';
import 'package:provider/provider.dart';
import 'package:intl/intl.dart';

class RecordatorioPage extends StatefulWidget {
  const RecordatorioPage({Key? key}) : super(key: key);

  @override
  State<RecordatorioPage> createState() => _RecordatorioPageState();
}

class _RecordatorioPageState extends State<RecordatorioPage> {
  String _opcionSeleccionada = 'Esta semana';
  List<String> _opcionesTiempo = [
    'Ayer',
    'Hoy',
    'Esta semana',
    'Próxima semana',
    'Este mes',
    'Próximo mes',
    'Este año',
    'Todo'
  ];

  StorageShared storageShared = StorageShared();
  List<RecordatorioModel> listRecordatorioModel = [];
  List<RecordatorioModel> listRecordatorioModelAux = [];
  bool isLoading = false;

  @override
  void initState() {
    super.initState();
    if (mounted) {
      _getListRecordatorios();
    }
  }
}

```

Figura 47. Recordatorios

```

class _HorarioPageState extends State<HorarioPage> {
  List<List<bool>> matrizBool =
    List.generate(23, (index) => List.generate(7, (index) => false));

  final _scrollController = ScrollController(initialScrollOffset: 360);
  int x = 0;
  int y = 0;
  HorarioProvider horarioProvider = HorarioProvider();
  ClasesProvider clasesProvider = ClasesProvider();
  StorageShared storageShared = StorageShared();
  bool isLoading = false;

  String nombreHorario = '';

  DocumentReference? referenceHorario;

  List<ClasesModel> listClasesModel = [];
  Map<String, dynamic> mapColors = {};
  @override
  void dispose() {
    // TODO: implement dispose
    super.dispose();
    _scrollController.dispose();
  }

  @override
  void didChangeDependencies() {
    // TODO: implement didChangeDependencies
    if (mounted) {
      _getHorarioDefault();
    }
  }

  @override
  Widget build(BuildContext context) {
    final UserProvider userProvider = Provider.of<UserProvider>(context);

```

Figura 48. Recordatorios

```

import 'package:flutter/material.dart';
import 'package:norkik_app/providers/storage_shared.dart';

class CalendarioPage extends StatelessWidget {
  const CalendarioPage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    // _print();
    return Scaffold(
      body: Text('Calendario'),
    ); // Scaffold
  }

  // _print() async {
  //   StorageShared storageShared = StorageShared();
  //   print((await storageShared.obtenerColoresAsignaturaList()));
  // }

  // _metodo() async {
  //   StorageShared storageShared = StorageShared();
  //   Map<String, dynamic> mapColors =
  //     await storageShared.obtenerColoresAsignaturaList();
  //   mapColors['fpf5T6rkzoIAFU1FMsBi'] = 4280723098;
  //   storageShared.agregarColoresAsignaturaList(mapColors);
  // }
}

```

Figura 49. Calendario

Las siguientes pantallas parten del menú lateral, el cual se compone del cuaderno digital, docentes, asignaturas, herramientas, perfil, y tareas, las mismas que se visualizan a continuación respectivamente:

```
import 'package:norkik_app/pages/drawer_pages/cuaderno_pages/texto_page.dart';
import 'package:norkik_app/providers/norkikdb_providers/asignatura_provider.dart';
import 'package:norkik_app/providers/storage_shared.dart';

class NotasPage extends StatefulWidget {
  const NotasPage({Key? key}) : super(key: key);

  @override
  State<NotasPage> createState() => _NotasPageState();
}

class _NotasPageState extends State<NotasPage>
  with SingleTickerProviderStateMixin {
  StorageShared storageShared = StorageShared();
  AsignaturaProvider asignaturaProvider = AsignaturaProvider();

  Map<String, dynamic> mapColors = {};
  List<AsignaturaModel> listAsignatura = [];
  int selectedPage = 0;
  TabController? tabController;
  AsignaturaModel? asignaturaSelected;

  bool isLoading = false;

  @override
  void initState() {
    super.initState();
    tabController =
      TabController(length: 3, initialIndex: selectedPage, vsync: this);
    tabController!.addListener(() {
      setState(() {});
    });
    if (mounted) {
      _getListAsignaturas();
    }
  }
}
```

Figura 50. Cuaderno digital

```
List<AsignaturaModel> listAsignatura = [];
List<AsignaturaModel> listAsignaturaRespaldo = [];
Map<String, dynamic> mapColors = {};

@override
void didChangeDependencies() {
  // TODO: Implement didChangeDependencies
  super.didChangeDependencies();
  _getListAsignatura();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: isSecondAppBarSelection
      ? AppBar(
          leading: IconButton(
            icon: Icon(Icons.close),
            onPressed: () {
              setState(() {
                isSecondAppBarSelection = false;
                listAsignatura = listAsignaturaRespaldo;
              });
            },
          ), // IconButton
          title: TextField(
            autofocus: true,
            decoration: InputDecoration(
              enabledBorder: UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white), // UnderlineInput
                hintText: 'Buscar',
                hintStyle: TextStyle(
                  color: Theme.of(context).appBarTheme.foregroundColor), // Text
                style: TextStyle(
                  color: Theme.of(context).appBarTheme.foregroundColor), // TextStyle
                onChanged: (text) {
                  _searchAsignaturas(text);
                }
              ),
            ),
          ),
        ),
  );
}
```

Figura 51. Asignaturas

```

import 'package:flutter/material.dart';
import 'package:flutter_speed_dial/flutter_speed_dial.dart';
import 'package:norkik_app/models/docente_model.dart';
import 'package:norkik_app/providers/norkikdb_providers/docente_provider.dart';

class DocentesPage extends StatefulWidget {
  DocentesPage({Key? key}) : super(key: key);

  @override
  State<DocentesPage> createState() => _DocentesPageState();
}

class _DocentesPageState extends State<DocentesPage> {
  DocenteProvider docenteProvider = DocenteProvider();

  bool isLoading = false;
  List<DocenteModel> listDocentes = [];

  @override
  void didChangeDependencies() {
    // TODO: implement didChangeDependencies
    super.didChangeDependencies();
    _getListDocentes();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Docentes'),
      ), // AppBar
      body: isLoading
        ? Center(
            child: CircularProgressIndicator(),
          ) // Center
        : listDocentes.isEmpty
        ? Center(

```

Figura 52. Docentes

En cuanto a los modelos su estructura parte de los atributos, métodos y su respectivo constructor:

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:norkik_app/models/privacidad_model.dart';

import 'apariencia_model.dart';

class UserModel {
  String idUsuario;
  String nombre;
  String apellido;
  String apodo;
  String whatsapp;
  String email;
  String password;
  String imgUrl;

  PrivacidadModel privacidad;
  AparienciaModel apariencia;

  static const String collectionId = 'Usuario';
  UserModel(
    {required this.idUsuario,
    required this.nombre,
    required this.apellido,
    required this.apodo,
    required this.whatsapp,
    required this.email,
    required this.password,
    required this.imgUrl,
    required this.apariencia,
    required this.privacidad});

  factory UserModel.fromFirestore(Map<String, dynamic> usuario) {
    return UserModel(
      idUsuario: usuario['UID'],
      nombre: usuario['Nombre'],
      apellido: usuario['Apellido'],
      apodo: usuario['Apodo'],

```

Figura 53. Modelo Usuario

```

class RecordatorioModel {
  String id;
  String nombre;
  String descripcion;
  DateTime fecha;

  RecordatorioModel(
    {required this.id,
     required this.nombre,
     required this.descripcion,
     required this.fecha});
  factory RecordatorioModel.fromStorage(Map<String, dynamic> mapRecordatorio) {
    return RecordatorioModel(
      id: mapRecordatorio['id'],
      nombre: mapRecordatorio['nombre'],
      descripcion: mapRecordatorio['descripcion'],
      fecha: mapRecordatorio['fecha']);
  }

  Map<String, dynamic> toMap() => {
    'id': id,
    'nombre': nombre,
    'descripcion': descripcion,
    'fecha': fecha.toString();
  };

  RecordatorioModel get getRecordatorioModel => RecordatorioModel(
    id: id, nombre: nombre, descripcion: descripcion, fecha: fecha);

  @override
  String toString() {
    return 'RecordatorioModel{id: $id, nombre: $nombre, descripción: $descripcion, fecha: $fecha}';
  }
}

```

Figura 54. Modelo Recordatorio

```

class PrivacidadModel {
  String? idPrivacidad;
  bool whatsapp;
  bool apodo;
  bool email;
  static const String collectionId = 'Privacidad';
  PrivacidadModel(
    {this.idPrivacidad,
     required this.whatsapp,
     required this.apodo,
     required this.email});
  factory PrivacidadModel.fromFirestore(Map<String, dynamic> privacidad) {
    return PrivacidadModel(
      idPrivacidad: privacidad['idPrivacidad'],
      whatsapp: privacidad['WhatsApp'],
      apodo: privacidad['Email'],
      email: privacidad['Apodo']);
  }

  Map<String, dynamic> toMap() => {
    'Apodo': apodo,
    'Email': email,
    'WhatsApp': whatsapp,
  };

  factory PrivacidadModel.privacidadNoData() {
    return PrivacidadModel(
      idPrivacidad: 'no-id', whatsapp: true, apodo: true, email: true);
  }
}

```

Figura 55. Modelo Privacidad

```

import 'package:cloud_firestore/cloud_firestore.dart';

import 'package:norkik_app/models/user_model.dart';

class PostModel {
  String idPost;
  String titulo;
  String descripcion;
  String imagen;
  DateTime fecha;
  UserModel usuario;
  static const String collectionId = 'Publicacion';

  PostModel(
    {required this.idPost,
    required this.titulo,
    required this.descripcion,
    required this.imagen,
    required this.fecha,
    required this.usuario});

  factory PostModel.fromFirestore(
    Map<String, dynamic> post, UserModel currentUser) {
    return PostModel(
      idPost: post['idPost'],
      titulo: post['Titulo'],
      descripcion: post['Descripcion'],
      imagen: post['File'],
      fecha: post['Fecha'].toDate(),
      usuario: currentUser);
  }

  Map<String, dynamic> toMap(
    DocumentReference<Map<String, dynamic>> usuarioRef =>
    {
      'Titulo': this.titulo,
      'Descripcion': this.descripcion,
    }
  );
}

```

Figura 56. Modelo Post o Publicación

```

class NoticiaModel {
  String idNoticia;
  String titulo;
  String descripcion;
  String imagen;
  DateTime fecha;
  bool isFavorite = false;
  static const String collectionId = 'Noticia';

  NoticiaModel(
    {required this.idNoticia,
    required this.titulo,
    required this.descripcion,
    required this.imagen,
    required this.fecha});

  factory NoticiaModel.fromFirestore(Map<String, dynamic> noticia) {
    return NoticiaModel(
      idNoticia: noticia['idNoticia'],
      titulo: noticia['Titulo'],
      descripcion: noticia['Descripcion'],
      imagen: noticia['Imagen'],
      fecha: noticia['Fecha'].toDate());
  }

  Map<String, dynamic> toMap() => {
    'Titulo': this.titulo,
    'Descripcion': this.descripcion,
    'Imagen': this.imagen,
    'Fecha': this.fecha
  };

  factory NoticiaModel.noticiaNoData() {
    DateTime now = DateTime.now();
    return NoticiaModel(
      idNoticia: 'no-id',
      titulo: 'no-titulo',
    );
  }
}

```

Figura 57. Modelo Noticia

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:norkik_app/models/asignatura_model.dart';
import 'package:norkik_app/models/user_model.dart';

class NotaModel {
  String idNota;
  String titulo;
  String descripcion;
  String file;
  DateTime fecha;
  String categoria;
  AsignaturaModel asignatura;
  UserModel usuario;

  static const String collectionId = 'Notas';

  NotaModel(
    {required this.idNota,
    required this.titulo,
    required this.descripcion,
    required this.file,
    required this.fecha,
    required this.categoria,
    required this.asignatura,
    required this.usuario});

  factory NotaModel.fromFirestore(Map<String, dynamic> mapNota) {
    return NotaModel(
      idNota: mapNota['idNota'],
      titulo: mapNota['Titulo'],
      descripcion: mapNota['Descripcion'],
      file: mapNota['File'],
      fecha: mapNota['Fecha'].toDate(),
      categoria: mapNota['Categoria'],
      asignatura: AsignaturaModel.fromFirestore(mapNota['Asignatura']),
      usuario: UserModel.fromFirestore(mapNota['Usuario']); // NotaModel
    );
  }
}

```

Figura 58. Modelo Notas o cuaderno digital

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:norkik_app/models/post_model.dart';

class LikePostModel {
  String idLikePost;
  PostModel post;
  List<dynamic> userList;
  DateTime fecha;

  static const String collectionId = 'LikePost';

  LikePostModel(
    {required this.post,
    required this.userList,
    required this.fecha,
    required this.idLikePost});

  factory LikePostModel.fromFirestore(
    Map<String, dynamic> post,
  ) {
    return LikePostModel(
      idLikePost: post['idLikePost'],
      post: post['Post'],
      userList: post['UserList'],
      fecha: post['Fecha'].toDate());
  }

  factory LikePostModel.likePostNoData() {
    return LikePostModel(
      post: PostModel.postNoData(),
      userList: [],
      fecha: DateTime.now(),
      idLikePost: 'no-id'); // LikePostModel
  }

  Map<String, dynamic> toMap(DocumentReference<Map<String, dynamic>> postRef) =>
    {'Post': postRef, 'UserList': userList, 'Fecha': fecha};
}

```

Figura 59. Modelo Like Post

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:norkik_app/models/user_model.dart';

class HorarioModel {
  String idHorario;
  String nombre;
  String descripcion;
  DateTime fecha;
  UserModel usuario;
  bool activo;

  static const String collectionId = 'Horarios';

  HorarioModel(
    {required this.idHorario,
    required this.nombre,
    required this.descripcion,
    required this.fecha,
    required this.usuario,
    required this.activo});

  factory HorarioModel.fromFirestore(Map<String, dynamic> mapHorario) {
    return HorarioModel(
      idHorario: mapHorario['idHorario'],
      nombre: mapHorario['Nombre'],
      descripcion: mapHorario['Descripcion'],
      fecha: mapHorario['Fecha'].toDate(),
      activo: mapHorario['activo'],
      usuario: UserModel.fromFirestore(mapHorario['Usuario']),
    ); // HorarioModel
  }

  Map<String, dynamic> toMap(
    DocumentReference<Map<String, dynamic>> documentReferenceUser =>
    {
      'Nombre': nombre,
      'Descripcion': descripcion,
    }
  );
}

```

Figura 60. Modelo Horario

```

class DocenteModel {
  String idDocente;
  String nombre;
  String apellido;
  String email;

  static const String collectionId = 'Docentes';

  DocenteModel({
    required this.idDocente,
    required this.nombre,
    required this.apellido,
    required this.email,
  });

  factory DocenteModel.fromFirestore(Map<String, dynamic> mapDocente) {
    return DocenteModel(
      idDocente: mapDocente['idDocente'],
      nombre: mapDocente['Nombre'],
      apellido: mapDocente['Apellido'],
      email: mapDocente['Email'],
    );
  }

  Map<String, dynamic> toMap() =>
    {'Nombre': nombre, 'Apellido': apellido, 'Email': email};

  factory DocenteModel.docenteModelNoData() {
    return DocenteModel(
      idDocente: 'no-id',
      nombre: 'no-nombre',
      apellido: 'no-apellido',
      email: 'no-email',
    );
  }
}

```

Figura 61. Modelo Docente

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:norkik_app/models/asignatura_model.dart';
import 'package:norkik_app/models/horario_model.dart';

class ClasesModel {
  String idClase;
  // List<Map<String, dynamic>> fechaInicioFin;
  List<dynamic> fechaInicioFin;
  HorarioModel horario;
  AsignaturaModel asignatura;

  static const String collectionId = 'Clases';

  ClasesModel(
    {required this.idClase,
    required this.fechaInicioFin,
    required this.horario,
    required this.asignatura});

  factory ClasesModel.fromFirestore(Map<String, dynamic> mapClase) {
    return ClasesModel(
      idClase: mapClase['idClase'],
      fechaInicioFin: mapClase['FechaInicioFin'],
      asignatura: AsignaturaModel.fromFirestore(mapClase['Asignatura']),
      horario: HorarioModel.fromFirestore(mapClase['Horario'])); // ClasesModel
  }

  Map<String, dynamic> toMap(
    DocumentReference<Map<String, dynamic>> docRefAsignatura,
    DocumentReference<Map<String, dynamic>> docRefHorario) =>
    {
      'FechaInicioFin': fechaInicioFin,
      'Asignatura': docRefAsignatura,
      'Horario': docRefHorario
    };
  }

  factory ClasesModel.cClasesModelNoData() {
    return ClasesModel(

```

Figura 62. Modelo Clases

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:norkik_app/models/docente_model.dart';

class AsignaturaModel {
  String idAsignatura;
  String nombre;
  String descripcion;
  String salon;

  DocenteModel docente;

  static const String collectionId = 'Asignaturas';

  AsignaturaModel(
    {required this.idAsignatura,
    required this.nombre,
    required this.descripcion,
    required this.salon,
    required this.docente});

  factory AsignaturaModel.fromFirestore(Map<String, dynamic> mapAsignatura) {
    return AsignaturaModel(
      idAsignatura: mapAsignatura['idAsignatura'],
      nombre: mapAsignatura['Nombre'],
      descripcion: mapAsignatura['Descripcion'],
      salon: mapAsignatura['Salon'],
      docente: DocenteModel.fromFirestore(mapAsignatura['Docente'])); // AsignaturaModel
  }

  Map<String, dynamic> toMap(
    DocumentReference<Map<String, dynamic>> documentReferenceDocente) =>
    {
      'Nombre': nombre,
      'Descripcion': descripcion,
      'Salon': salon,
      'Docente': documentReferenceDocente
    };
  }

```

Figura 63. Modelo Asignatura

```

class AparienciaModel {
  String idApariencia;
  String tema;
  int tamañoFuente;
  static const String collectionId = 'Apariencia';
  AparienciaModel(
    {required this.idApariencia,
     required this.tema,
     required this.tamañoFuente});

  factory AparienciaModel.fromFirestore(Map<String, dynamic> apariencia) {
    return AparienciaModel(
      idApariencia: apariencia['idApariencia'],
      tema: apariencia['Tema'],
      tamañoFuente: apariencia['TamañoFuente']);
  }

  Map<String, dynamic> toMap() => {
    'TamañoFuente': tamañoFuente,
    'Tema': tema,
  };
}

factory AparienciaModel.aparienciaNoData() {
  return AparienciaModel(
    idApariencia: 'no-id', tema: 'NorkikTheme', tamañoFuente: 14);
}
}

```

Figura 64. Modelo Apariencia

En cuanto los controladores tienen conexión con los modelos, las cuales sus estructuras están compuestas por las operaciones CRUD, es decir, lectura, escritura, actualización, y eliminación. A continuación, se visualiza un ejemplo del Usuario Provider, es decir el controlador de la sección Usuario:

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/cupertino.dart';
import 'package:norkik_app/models/user_model.dart';
import 'package:norkik_app/providers/norkikdb_providers/apariencia_provider.dart';
import 'package:norkik_app/providers/norkikdb_providers/privacidad_provider.dart';

class UserProvider with ChangeNotifier {
  CollectionReference usuariosRef =
    FirebaseFirestore.instance.collection(UserModel.collectionId);

  bool _isLoading = false;
  bool get isLoading => _isLoading;
  void setLoading(bool loading) {
    _isLoading = loading;
    notifyListeners();
  }

  UserModel _userGlobal;
  UserModel get userGlobal => _userGlobal;
  UserProvider(this._userGlobal);
  void setUserGlobal(UserModel user) {
    _userGlobal = user;
    notifyListeners();
  }

  void setUserGlobalWithoutNotify(UserModel user) {
    _userGlobal = user;
  }

  Future<void> createUser(UserModel usuario) async {
    setLoading(true);
    PrivacidadProvider privacidadProvider = PrivacidadProvider();
    AparienciaProvider aparienciaProvider = AparienciaProvider();

    DocumentReference<Map<String, dynamic>> privacidadRef =
      await privacidadProvider.createPrivacidad(usuario.privacidad)
        as DocumentReference<Map<String, dynamic>>;
  }
}

```

Figura 65. Usuario

Cabe mencionar que cada proveedor corresponde a cada Modelo de los anteriormente mencionados, además de proveedores adicionales en cuanto a la autenticación, notificaciones, almacenamiento local, etc.

OCR

Para determinar la mejor manera de implementar una API que ayude al reconocimiento óptico de caracteres se midió la carga computacional de dos herramientas OCR. La primera herramienta hace referencia al uso de Tesseract(encargado de convertir la imagen a texto) en conjunto con Opencv (para el preprocesamiento de la imagen). Y la segunda herramienta es Tensor Flow, específicamente Keras. De manera que los métodos de aplicación del OCR se realizó su respectiva medición de la carga computacional en un servicio web de Google denominado Colab, el cual consta de 12,68GB de RAM y el disco de 107,72GB de RAM. Lo que se obtuvo fue lo siguiente:

Los factores que se tomaron en cuenta fueron el consumo de la memoria RAM durante la ejecución del método OCR, más el tiempo de ejecución de esta. De manera que los resultados fueron los siguientes:

Tabla 58.
Carga Computacional OCR

Métodos de aplicar OCR	de	Parámetros		Imagen de 10 a 50 palabras	Imagen de 50 a 100 palabras	Imagen de 100 o más palabras
PyTesseract Opencv	+	Consumo RAM (Porcentaje)	de	10%	10.2%	9.7%
		Consumo CPU (Porcentaje)	de	2.5%	3.1%	5.1%
		Tiempo ejecución (Segundos)	de	0.58s	1.03s	3.081s
Tensor Flow (Keras-ocr)		Consumo RAM (Porcentaje)	de	10.2%	9.7%	9.5%
		Consumo	de	36.8%	67.2%	62.3%

CPU (Porcentaje)			
Tiempo ejecución (Segundos)	de 50,56s	58.9s	72.30s

De acuerdo con los resultados obtenidos en función de los parámetros, detallados en la anterior tabla. Se puede concluir que la aplicación de OCR con TensorFlow consume más carga computacional a diferencia del otro método. De manera que la herramienta ideal para la implementación de una API de OCR, se lo realizó con Tesseract(pytesseract) y Opencv.

A pesar de los estudios de rendimiento obtenidos, se pudo observar que el primer método (Tensor Flow) tiene más eficiencia en cuanto a la detección de los caracteres, en consecuencia, el tiempo de respuesta es significativo. Mientras que el segundo método su eficiencia disminuye, pero el tiempo de respuesta es casi inmediato.

- **Adquisición de la imagen**

La primera fase de la visión artificial menciona la adquisición de la imagen por lo que la misma acción se la realiza por medio de la cámara que posee cada teléfono celular. La misma que da como resultado una imagen a color.

- **Preprocesamiento**

La segunda fase se la realiza con la herramienta Opencv aplicando diversos filtros como la escala de grises, detección de bordes, eliminación de ruido y posteriormente la imagen a blanco y negro:



Figura 66. Preprocesamiento de Imagen

- **Representación, descripción, reconocimiento e interpretación**

Las siguientes fases correspondientes a la visión artificial se encargó la librería que seleccionamos anteriormente la cual aplica su propia base del conocimiento. De modo que la primera etapa en lo que corresponde al OCR es la detección de bloques a través del análisis de la imagen. De modo que en la siguiente gráfica se visualiza su aplicación:



Figura 67. Detección de bloques

Seguido del anterior método, se procedió con la búsqueda de palabras y líneas. Es decir, detectar unidades de texto, partiendo desde líneas a palabras. Posteriormente se reconoce las palabras para luego clasificar los caracteres, haciendo posible detectar letra a letra para posteriormente tratar de unirlos en función de su contexto, la cual se basa en un análisis lingüístico para tratar de mejorar la eficiencia.

Redes neuronales

En cuanto a las redes neuronales tal como se menciona en el marco teórico se utilizó Tensor Flow (Keras), los motivos de su uso se debieron a su popularidad, aprendizaje y que es considerado de código abierto. Para determinar un modelo eficiente se realizaron diversas pruebas, por lo que se detallan a continuación:

Primeramente, el modelo de la red neuronal se decidió realizarlo por 3 capas. Es decir, una capa de entrada, una capa oculta, y una capa de salida. De manera que la siguiente

fase de la estructuración del modelo fue decidir cuantas neuronas debería tener cada capa.

- **Capa de entrada**

Dado que la capa de entrada tiene que ver en la forma que ingresarían los datos de entrenamiento y predicción se optó por manejar con arreglos de la librería Numpy. Ahora, la cantidad de neuronas que debería tener la capa de entrada se relacionó con el tipo de datos que deseamos entrenar, así como también predecir. En consecuencia, se consideró que los datos que más interacción tiene el estudiante son las notas, las tareas y recordatorios. De manera que estos se rigen por los días semanales en una notación del 1 al 7. Donde lunes equivaldría a 1, martes al número 2, y así sucesivamente hasta llegar al Domingo. Por otra parte, los datos generados por el estudiante tomarían valores desde el 1 al 3. De manera que 1 sería las notas, 2 las tareas y 3 los recordatorios.

Una vez convertido los datos disponibles que genera un usuario a datos numéricos se procedió a combinar estos datos con sus respectivas etiquetas de respuesta al igual que los días de la semana en números y los datos del usuario enumerados respectivamente. En cuanto a la respuesta 1 correspondería a verdadero y cero a falso. Dando como ejemplo unitario del arreglo de la siguiente manera: [1,5,0], donde 1 es el dato nota, 5 el día de la semana y cero el valor de respuesta de si se ha creado un evento o no, en este caso haber creado una nota. Ahora bien, si partimos de la obtención de los datos de entrenamiento a partir de lo que genera el usuario en un intervalo de tiempo quedaría de la siguiente manera:

```
array([[1, 1, 1],
       [1, 2, 0],
       [1, 3, 0],
       [1, 4, 0],
       [1, 5, 0],
       [1, 6, 0],
       [1, 7, 1],
       [2, 1, 1],
       [2, 2, 0],
       [2, 3, 0],
       [2, 4, 0],
       [2, 5, 0],
       [2, 6, 0],
       [2, 7, 1],
       [3, 1, 1],
       [3, 2, 0],
       [3, 3, 0],
       [3, 4, 0],
       [3, 5, 0],
       [3, 6, 0],
       [3, 7, 1]])
```

Figura 68. Ejemplo de datos de entrenamiento

Del arreglo anteriormente visualizado se tuvo que separar las respuestas (1 o 0), de los otros datos como [1,2], obteniendo como resultado datos de entrenamiento etiquetados en función de los índices de cada arreglo respectivamente en base a dos arreglos (el primero corresponde a los datos y el segundo a sus respectivas respuestas).

En consecuencia, al primer arreglo obtenido tendría una dimensión de una fila y dos columnas. Concluyendo que la capa de entrada fue de dos neuronas respectivamente.

En cuanto a su función de activación se decidió por ReLU (Rectified Lineal Unit), debido a que transforma los valores que ingresan. Es decir que anula valores negativos y deja a los positivos. La fórmula y gráfica de esta función se visualiza de la siguiente manera:

$$f(x) = \max(0, x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

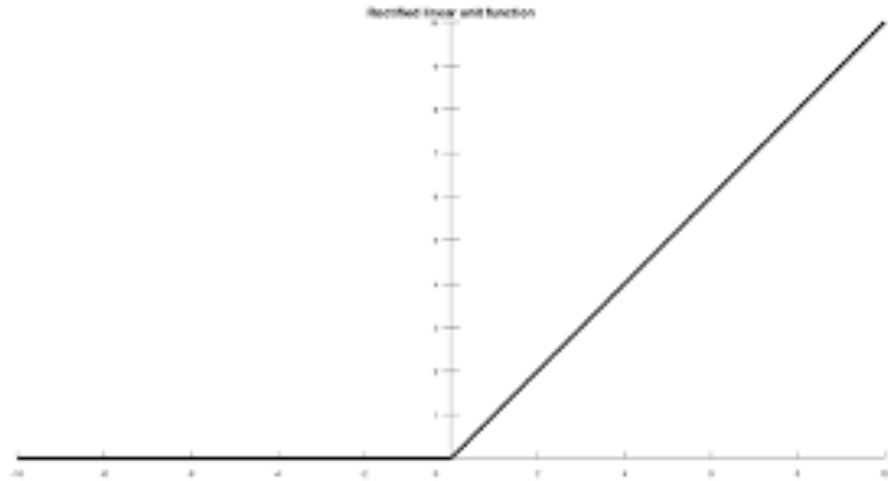


Figura 69. Ejemplo de datos de entrenamiento

- **Capa intermedia**

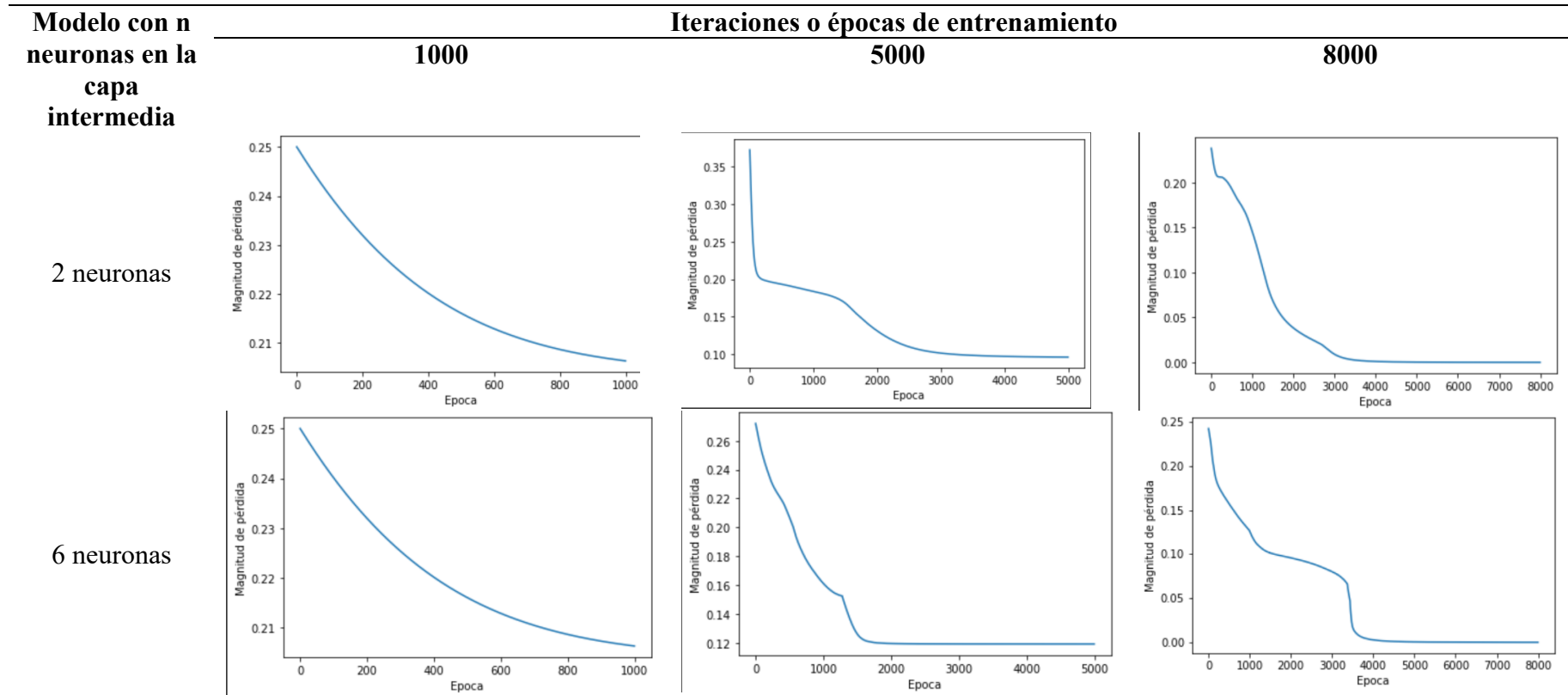
Para determinar el número de neuronas de esta capa, se tuvo que realizar experimentación con el modelo de la red. De manera que la eficiencia de esta dependería de cuan bajo sea el valor de la magnitud de pérdida al igual de las iteraciones o épocas que se deberían realizar con los datos de entrenamiento. En la siguiente tabla se puede observar a detalle la eficiencia del modelo a partir del número de neuronas intermedias con su respectiva gráfica y las veces que se realiza las iteraciones, así como también su carga computacional.

Tabla 59.
Carga Computacional Redes Neuronales

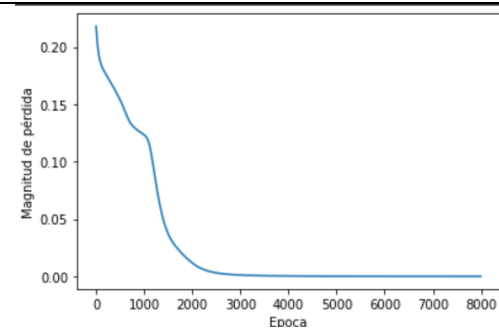
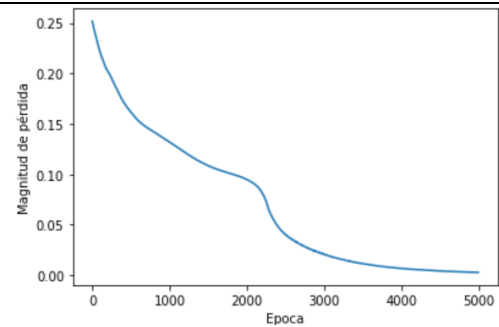
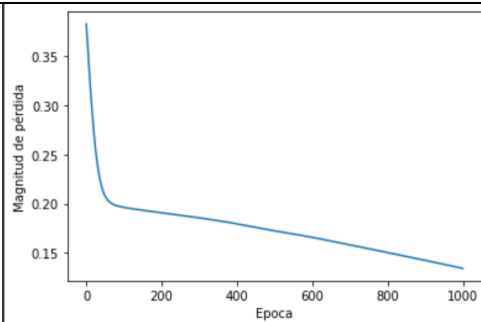
Modelo con n neuronas en la capa intermedia	Parámetros	Iteraciones o épocas de entrenamiento		
		1000	5000	8000
2 neuronas	Exactitud (Accuracy)	0.8571	0.8571	1.0000
	Magnitud de pérdida (Loss)	0.1538	0.1200	2.7826e-05
	Consumo de RAM (Porcentaje)	9.9%	10.1%	10.2%
	Consumo de CPU (Porcentaje)	2.4%	29.3%	21%
	Tiempo de ejecución (Segundos)	2.55s	10.58s	20.94s
6 neuronas	Exactitud (Accuracy)	0.7143	0.8571	1.0000
	Magnitud de pérdida (Loss)	0.2064	0.1191	4.5307e-05
	Consumo de RAM (Porcentaje)	10.3%	10.1%	10.2%
	Consumo de CPU (Porcentaje)	16%	14.7%	11.9%
	Tiempo de ejecución (Segundos)	2.48s	10.71s	20.94s
12 neuronas	Exactitud (Accuracy)	0.8571	1.0000	1.0000
	Magnitud de pérdida (Loss)	0.1341	0.0027	6.0911e-04
	Consumo de RAM (Porcentaje)	10.1%	10.1%	10.3%
	Consumo de CPU (Porcentaje)	6.0%	12.1%	26.5%
	Tiempo de ejecución (Segundos)	2.56s	10.70s	20.93s
24 neuronas	Exactitud (Accuracy)	0.9524	1.0000	1.0000
	Magnitud de pérdida (Loss)	0.0872	0.0012	2.9086e-05
	Consumo de RAM (Porcentaje)	10.3%	10.3%	10.3%
	Consumo de CPU (Porcentaje)	2.7%	9.5%	11.1%
	Tiempo de ejecución (Segundos)	3.41s	10.70s	16.75s

Tabla 60.

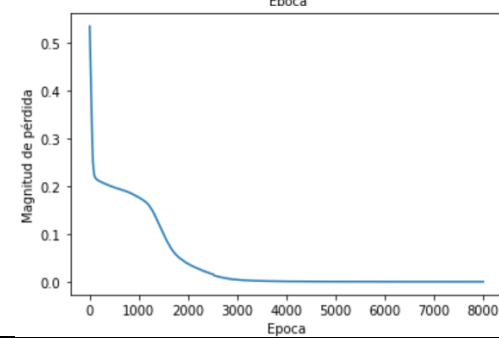
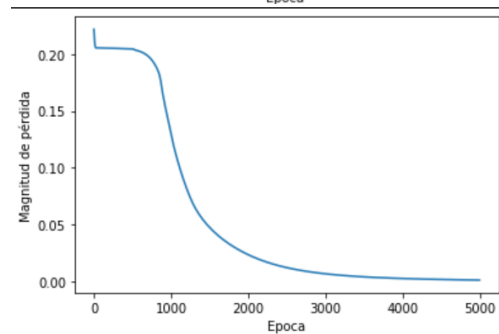
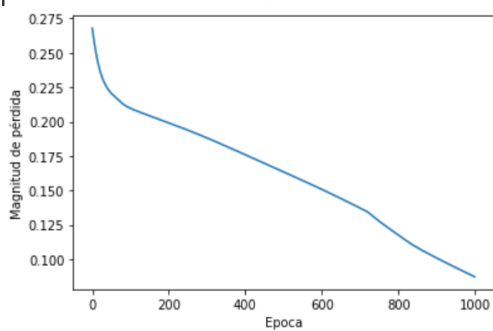
Iteraciones o épocas de entrenamiento



12 neuronas



24 neuronas



De la anterior tabla se pudo concluir que el tiempo de ejecución del modelo va reduciendo en función del número de neuronas, así como también las cantidades de iteraciones en el entrenamiento. Mientras más épocas de entrenamiento se asigna al modelo, mas es su tiempo de ejecución y recursos computacionales, sin embargo, su eficiencia y reducción de error va teniendo mejores resultados.

En cuanto a la construcción del modelo de red neuronal se optó por implementar 24 neuronas en la capa intermedia, de manera que los recursos computacionales y su tiempo de ejecución sean considerablemente normales, al igual que su eficiencia con su magnitud de pérdida relativamente menor.

La función de activación para esta capa correspondería a la misma que se aplicó en la capa de entrada. Con el objetivo de evitar valores negativos en el modelo.

Capa de salida

Finalmente, la capa de salida vendría a ser conformada por una neurona, puesto a que se desea que los valores sean desde 0 hasta 1. De modo que para obtener este comportamiento se aplicó la función de activación Sigmoide. En la siguiente gráfica se puede visualizar el comportamiento de esta función.

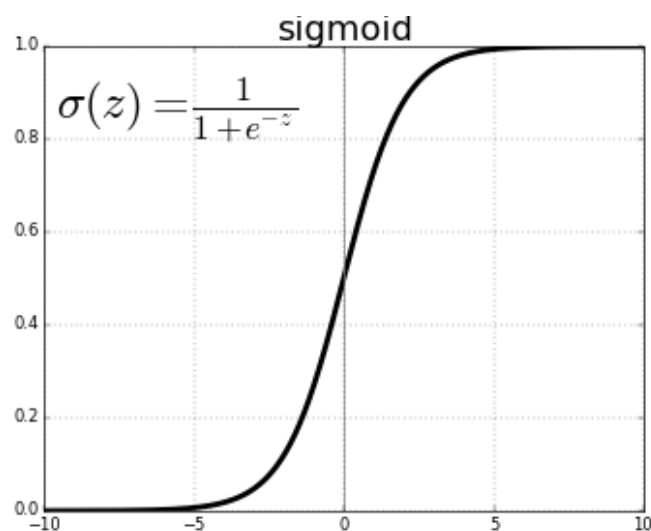


Figura 70. Función de activación sigmoide

Modelo final

Como se mencionó anteriormente, el modelo final de red neuronal quedó conformada por una capa de entrada con dos neuronas, 1 capa oculta o intermedia con 24 neuronas y 1 capa de salida con 1 neurona. Cada capa con su respectiva función de activación. Dando como resultado la siguiente gráfica:

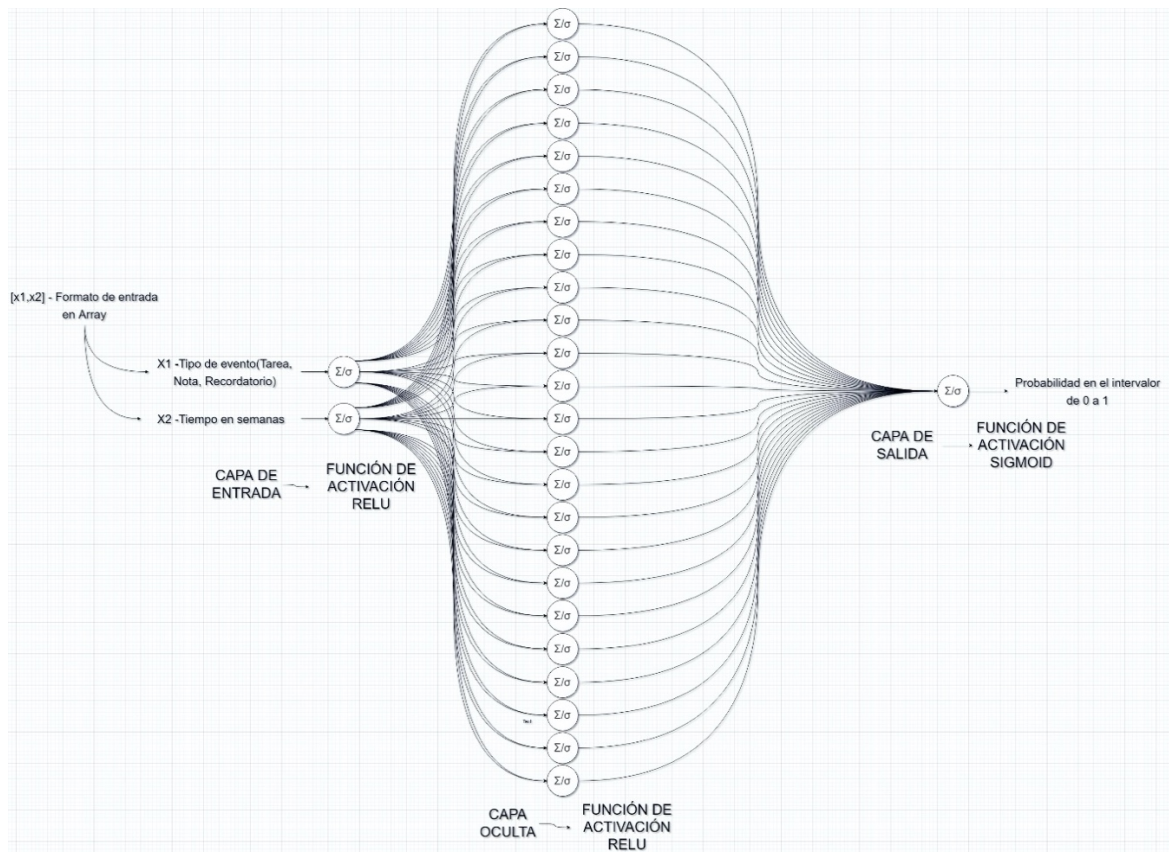


Figura 71. Modelo de red neuronal Norkik

Escáner con Flutter OpenCV

Por cuestión de tiempo, se decidió implementar un plugin Opencv en el lado del usuario, denominado como opencv_4 1.0.0 Flutter con su versión 1.0.0. El proceso se dividió en 3 fases las cuales tienen un poco de relación con el procesamiento de imágenes, estas se mencionan a continuación:

- **Adquisición de la imagen:** Esta fase es la misma que se aplicó para las etapas de visión artificial. Pues, se obtiene la imagen del mundo real por medio de sensores, iluminación y la digitalización. Este factor depende de la calidad del dispositivo móvil que el usuario posea. Las imágenes que se adquieren son almacenadas en una memoria caché, en donde se obtiene la dirección del fichero.
- **Procesamiento:** La segunda fase consta de obtener la imagen en mapas de bits a través de la dirección de la foto, en el caso de Flutter viene representada por el tipo de dato 'Uint8List' el cual permitió aplicar diferentes filtros con Opencv. Estos filtros se aplican al mismo tiempo, para proveer al usuario diferentes opciones de calidad de la imagen. Finalmente, los filtros aplicados se truncan en la misma dirección del fichero, generando así optimización en esta parte del submódulo escáner
- **Construcción del documento PDF:** En esta parte se obtienen diferentes imágenes ya procesadas, organizadas por listas en la cual, el usuario define el orden de estas y la cantidad que desea. Posteriormente se crea un fichero vacío en la dirección raíz de la aplicación. Luego, se pega cada mapa de bit (imágenes) en el fichero creado, separado por páginas. De manera que, el documento creado se pudo visualizar obteniendo la dirección del fichero.

Por otra parte, el proceso del escáner define la aplicación de filtros, sin embargo, en esta investigación se tuvo que elegir algunas cantidades de filtros para disminuir la carga computación en el lado del cliente. En la siguiente gráfica se puede visualizar, lo que es

la carga computacional que generó esta herramienta. Todo esto en un dispositivo con memoria RAM de 3 GB, procesador de Ocho núcleos a 1,8GHz, cámara de 13 megapíxeles y su sistema operativo Android 10.

Tabla 60.

Carga computacional OpenCV

Dispositivo de prueba	Parámetros	4 filtros	6 filtros	8 filtros	10 filtros
Samsung A20S	Consumo de RAM (MB)	414.13MB	414.81MB	415.51MB	448.86MB
	CPU (Latencia)	56.12ms	61.05ms	100.86ms	195ms
	Tiempo de ejecución (Segundos)	10.25s	10.33s	10.95s	11.29s
	GPU (Latencia)	51ms	73.5ms	134.4ms	148.1ms

De acuerdo con la tabla anterior, la carga computacional se ve afectada por la cantidad de filtros que se apliquen al mismo tiempo, afectando considerablemente el rendimiento de la aplicación. Debido a esto que se decidió utilizar 4 filtros aplicables en el submódulo escáner, ayudando así a minimizar el trabajo del lado del cliente, y en consecuencia la aplicación móvil tenga fluidez. Entre los filtros que se consideraron implementar en el submódulo escáner fueron: Bilateral (Borde por defecto), Whiteboard (o blanco y negro), escala de grises y el filtro a color de la imagen.

4.2.2.4. Fase de pruebas

Tabla 61.

Prueba de Aceptación 1

PRUEBAS DE ACEPTACIÓN	
Código: 1	Nº Historia de Usuario: 1
Historia de Usuario: Login	
Descripción: El usuario final o estudiante deberá acceder a la agenda por medio de una cuenta de usuario, el mismo que puede ser creado con Gmail.	
Condiciones de Ejecución: El usuario podrá iniciar sesión y crear una nueva cuenta	

Entrada/Pasos de Ejecución:

1. Dar clic Registro
2. Ingresar Información
3. Dar clic en crear cuenta
4. Iniciar sesión con credenciales
5. Iniciar sesión con Gmail

Resultado Esperado: Una vez validado el inicio de sesión aparece la sección Resumen

Evaluación de la Prueba: Aceptado

Tabla 62.

Prueba de Aceptación 2

PRUEBAS DE ACEPTACIÓN

Código: 2

Nº Historia de Usuario: 2

Historia de Usuario: Resumen

Descripción: En la pantalla de resumen, las cuales visualizarán los eventos, recordatorios, o tareas pendientes para el día presente, así como también para los días siguientes, en función de la semana que se encuentre.

Condiciones de Ejecución: El usuario podrá visualizar los eventos, recordatorios, o tareas pendientes del día y la semana.

Entrada/Pasos de Ejecución:

1. Deslizar la información del reporte semanal

Resultado Esperado: Una vez revisado el resumen pasa a sección noticias.

Evaluación de la Prueba: Aceptado

Tabla 63.

Prueba de Aceptación 3

PRUEBAS DE ACEPTACIÓN

Código: 3

Nº Historia de Usuario: 3

Historia de Usuario: Sección Noticias

Descripción: El estudiante también podrá acceder a una sección de noticias, en el cual se visualizarán las novedades o noticias generadas por la universidad.

Condiciones de Ejecución:

Entrada/Pasos de Ejecución:

1. Visualización de noticias
2. Botón me gusta
3. Las publicaciones con like se guardan en una lista

Resultado Esperado: El botón de like cambie de color al seleccionar, ampliación de la noticia al dar clic

Evaluación de la Prueba: Aceptado

Tabla 64.
Prueba de Aceptación 4

PRUEBAS DE ACEPTACIÓN	
Código: 4	Nº Historia de Usuario: 4
Historia de Usuario: Control Notas o apuntes	
Descripción: El estudiante podrá acceder a una sección de notas, en la que estará organizado por las asignaturas correspondientes al horario de clases. Correspondería a un cuaderno digital.	
Condiciones de Ejecución: El estudiante podrá agregar, editar o eliminar las notas.	
Entrada/Pasos de Ejecución:	
1. Pestañas correspondientes al nombre de la asignatura u otra clasificación, de modo que al presionar se enlisten las respectivas notas creadas, con los campos correspondientes a título de la nota, su clasificación (asignatura), una breve descripción y la fecha de la nota.	
2. De la misma manera deberá permitir la eliminación, edición o creación de la nota.	
3. Debe contener un campo de texto y un botón para la funcionalidad de búsqueda.	
4. Cada nota debe permitir el ingreso de texto, audio o foto en caso de que sea necesario para el estudiante.	
Resultado Esperado: El estudiante podrá agregar, editar o eliminar las notas.	
Evaluación de la Prueba: Aceptado	

Tabla 65.
Prueba de Aceptación 5

PRUEBAS DE ACEPTACIÓN	
Código: 5	Nº Historia de Usuario: 5
Historia de Usuario: Control Tareas	
Descripción: La sección de tareas que permita la organización de estas en función a la asignatura correspondiente.	
Condiciones de Ejecución: Se dividirá por pestañas correspondientes al nombre de la asignatura, de modo que se enlistarán en tarjetas, detallando, el nombre de la tarea, su asignatura, su fecha y breve descripción. Permitiendo acciones como edición o eliminación de estas.	
Entrada/Pasos de Ejecución:	
1. Agregar, editar o eliminar tareas.	
2. Notificaciones de tareas pendientes.	
Resultado Esperado: No deberá permitir crear la tarea si no se ha seleccionado una asignatura, en caso de no existir, se deberá crearla.	
Deberá crear notificaciones en función de la fecha de entrega de la tarea.	
Los campos que se visualizan en la tarjeta de tarea son los mismos para la creación.	
Evaluación de la Prueba: Aceptado	

Tabla 66.
Prueba de Aceptación 6

PRUEBAS DE ACEPTACIÓN	
Código: 6	Nº Historia de Usuario: 6
Historia de Usuario: Control Recordatorios	
Descripción: Existir la sección de recordatorio en donde su similitud es parecida a la sección de tareas. Con la diferencia que no viene clasificado por asignaturas.	
Condiciones de Ejecución: Cada tarea debe permitir la opción de notificar al estudiante sobre alguna tarea que aproxime la fecha de entrega en función de la hora que se haya establecido.	
Entrada/Pasos de Ejecución:	
<ol style="list-style-type: none">1. Agregar, editar o eliminar recordatorios.2. Notificaciones de tareas pendientes.	
Resultado Esperado:	
<ul style="list-style-type: none">• Se deberá visualizar los distintos recordatorios en forma de lista, organizado por la fecha desde la que más se aproxima a la fecha actual hasta las posteriores fechas que se aproximen.• Se debe detallar en la lista la fecha, el nombre del recordatorio y la descripción.• En la lista deben estar 2 botones que permitan la edición y eliminación.• Deberá existir un botón flotante para la creación, en donde se compondrá de los mismos campos que se visualizan en la lista.	
Deberá crear notificaciones en función de la fecha de recordatorio	
Evaluación de la Prueba: Aceptado	

Tabla 67.
Prueba de Aceptación 7

PRUEBAS DE ACEPTACIÓN	
Código: 7	Nº Historia de Usuario: 7
Historia de Usuario: Horario de clases	
Descripción: Un horario de clases en forma estructural de tabla, de manera que los días laborales de la semana correspondan a las columnas (puede incluir sábados) y las filas sean las horas o el orden de las filas.	
Condiciones de Ejecución: Al momento de crear las asignaturas, deben diferenciarse por colores personalizados por el usuario	
Entrada/Pasos de Ejecución:	
<ol style="list-style-type: none">1. Agregar, editar o eliminar clases.	
Resultado Esperado:	
<ul style="list-style-type: none">• Poder crear varios horarios de clase identificado por un nombre asignado por el usuario.• Debe existir un botón en la parte esquina superior que permita cambiar de horario de clases, entre los ya creados por el estudiante.• Al momento de seleccionar una celda del horario de clases, se deberá ingresar una asignatura, El formulario de creación de la asignatura se lo detalla en la historia de usuario 12.	

Evaluación de la Prueba: Aceptado

Tabla 68.

Prueba de Aceptación 8

PRUEBAS DE ACEPTACIÓN

Código: 8**Nº Historia de Usuario: 8****Historia de Usuario:** Control Asignaturas**Descripción:** Al momento de crear las asignaturas, deben diferenciarse por colores personalizados por el usuario**Condiciones de Ejecución:** Al momento de crear las asignaturas, deben diferenciarse por colores personalizados por el usuario**Entrada/Pasos de Ejecución:**

1. Agregar, editar o eliminar asignaturas.

Resultado Esperado:

- Al momento de seleccionar la celda se deberá desplegar un formulario con los siguientes campos: el nombre de la asignatura, el docente, el salón de clases, el o los días de clases, así como también el inicio y fin de esta y finalmente el color que diferenciará a la materia.
- Adicionalmente debe existir el botón de añadir y otro de cancelar la creación de la asignatura.
- Deberá permitir la creación o la selección de un docente
- En otro apartado es considerable la implementación de una vista que visualice la lista de las asignaturas ya creadas, de modo que, al seleccionar, debe visualizarse el resumen del horario semanal, eventos, tareas, notas, etc. Deberá existir en esta sección un botón flotante que permita la creación de esta.

Evaluación de la Prueba: Aceptado

Tabla 69.

Prueba de Aceptación 9

PRUEBAS DE ACEPTACIÓN

Código: 9**Nº Historia de Usuario: 9****Historia de Usuario:** Control Docentes**Descripción:** Permite el control de los registros de docente**Condiciones de Ejecución:****Entrada/Pasos de Ejecución:**

1. Agregar, editar o eliminar docentes.

Descripción: El control de los registros de docente estará compuesto de lo siguiente:

- El formulario de creación docente deberá contener lo siguiente: Nombre, Apellido, Teléfono, Correo electrónico, y foto, cada campo tendrá su icono de identificación y campo de texto para el ingreso de los datos
 - En otro apartado se enlistarán los registros de los docentes existentes, las cuales al seleccionar debe visualizar a detalle los datos correspondientes al profesor, así como también existir dos botones, uno para la edición y eliminación
-

-
- La lista visualizada de docentes deberá visualizar el nombre y la foto del docente, en caso de que no exista, establecer uno por defecto.

Finalmente debe existir un botón flotante para la creación de registro de docente.

Evaluación de la Prueba: Aceptado

Tabla 70.

Prueba de Aceptación 10

PRUEBAS DE ACEPTACIÓN

Código: 10

Nº Historia de Usuario: 10

Historia de Usuario: Calendario

Descripción: Este apartado iniciaría como título con la fecha correspondiente al día, mes y año.

Condiciones de Ejecución:

Entrada/Pasos de Ejecución:

1. Visualización de Calendario con puntos que representan las tareas

Resultado Esperado:

- Una tabla similar o igual a los calendarios convencionales que muestran los días de la semana y los números correspondientes a los mismos.
- La celda debe seleccionarse automáticamente al día presente.
- En relación con la celda seleccionada del día del mes, se deberá detallar en forma de listas, las tareas, recordatorios e inclusive el horario de clases.
- Cada elemento de la lista debe contener un botón que permita elegir entre editar, eliminar o cambiar a estado de terminado el evento.
- Se debe permitir de manera fácil la navegación del calendario.
- Debe existir un botón que sitúe inmediatamente al día presente.

Evaluación de la Prueba: Aceptado

Tabla 71.

Prueba de Aceptación 11

PRUEBAS DE ACEPTACIÓN

Código: 11

Nº Historia de Usuario: 11

Historia de Usuario: Comunidad o cartelera

Descripción: En lo que respecta a la comunidad, esta sección permitirá al estudiante generar publicaciones y visualizarlas. Esto con el objetivo de crear una cartelera digital dedicada al estudiante.

Condiciones de Ejecución: Las publicaciones visualizarán en estructura de tarjetas por cada una. Detallando el usuario, la fecha de subida, categoría, la foto y/o descripción de la publicación

Entrada/Pasos de Ejecución:

1. Inicialmente debe existir un botón que permita crear una publicación que contendrá lo siguiente: El título, contenido de la publicación, categoría de publicación (estará preestablecido), el número de contacto y una opción de añadir foto. En donde cada campo tendrá su respectiva casilla de ingreso de datos.
-

Adicionalmente dos botones que permita aceptar o cancelar el proceso.

2. Agregar, editar y eliminar publicación.
3. Historial de publicaciones con “me gusta”

Resultado Esperado: Las publicaciones visualizarán en estructura de tarjetas por cada una. Detallando el usuario, la fecha de subida, categoría, la foto y/o descripción de la publicación, así como también un botón o hipervínculo que dirección al contacto de WhatsApp.

Evaluación de la Prueba: Aceptado

Tabla 72.

Prueba de Aceptación 12

PRUEBAS DE ACEPTACIÓN

Código: 12

Nº Historia de Usuario: 12

Historia de Usuario: Privacidad

Descripción: Dentro del apartado de configuración existirá una sección denominada privacidad, en donde se establecerá que datos serán visibles hacia la comunidad, pues cada usuario del software podrá acceder al perfil público

Condiciones de Ejecución:

Entrada/Pasos de Ejecución:

1. Visualizar Apodo
2. Visualizar E-mail
3. Visualizar Whatsapp

Resultado Esperado: Una lista, compuesta de checklist, de manera que el visto permita seleccionar la visibilidad, caso contrario debería estar la casilla en blanco.

Evaluación de la Prueba: Aceptado

Tabla 73.

Prueba de Aceptación 13

PRUEBAS DE ACEPTACIÓN

Código: 13

Nº Historia de Usuario: 13

Historia de Usuario: Apariencia

Descripción: La apariencia contendrá el tema personalizable de la agenda como tal.

Condiciones de Ejecución:

Entrada/Pasos de Ejecución:

1. La apariencia contendrá el tema personalizable de la agenda como tal. Pues se deberá conceder una gama de colores, de modo que, el usuario pueda elegir el que más le agrade.
2. La opción de los colores vendrá representada en una lista, de modo que cada ítem de la lista contendrá un icono del color al que pertenece con su respectivo nombre.
3. Cada ítem será seleccionable.
4. Agregando, además, un botón de aceptar o cancelar de acuerdo con la selección.

Resultado Esperado: Cambio de apariencia.

Evaluación de la Prueba: Aceptado

Tabla 74.

Prueba de Aceptación 14

PRUEBAS DE ACEPTACIÓN	
Código: 14	Nº Historia de Usuario: 14
Historia de Usuario: Cuenta o perfil	
Descripción: En lo que respecta a la cuenta o perfil, deberá permitir el cambio de foto, nombre y contraseña, con las respectivas medidas de seguridad.	
Condiciones de Ejecución:	
Entrada/Pasos de Ejecución:	
1. Se visualizará en la ventana, la foto, el nombre de usuario, correo, la carrera al que pertenece y la contraseña (el texto de la contraseña será visualizada de manera censurada). Cada campo anteriormente mencionado debe permitir la edición de esta, ya sea por selección o un botón.	
2. Al momento de la edición. Se deberá desplegar una ventana emergente que permita realizar el proceso de edición. Conteniendo dos botones, el de aceptar y cancelar.	
Resultado Esperado: Visualizar la información del usuario y editar.	
Evaluación de la Prueba: Aceptado	

Tabla 75.

Prueba de Aceptación 15

PRUEBAS DE ACEPTACIÓN	
Código: 15	Nº Historia de Usuario: 15
Historia de Usuario: Información	
Descripción: La información será el apartado de visualizar datos relativamente importantes como la versión de la agenda y acerca de los desarrolladores de esta.	
Condiciones de Ejecución:	
Entrada/Pasos de Ejecución:	
1. Estos datos se visualizarán en un párrafo, encabezado por el logo y versión del software.	
Resultado Esperado: Visualizar información	
Evaluación de la Prueba: Aceptado	

Tabla 76.

Prueba de Aceptación 16

PRUEBAS DE ACEPTACIÓN	
Código: 16	Nº Historia de Usuario: 16
Historia de Usuario: Herramientas	
Descripción: Herramientas digitales para el proceso académico	
Condiciones de Ejecución: El apartado herramientas darán la posibilidad de tener un conversor de imagen a texto además de un scanner de modo que permita al estudiante facilitar su proceso académico digital.	
Entrada/Pasos de Ejecución:	

-
1. OCR Conversión de imagen a texto
 2. Lector QR
 3. Scanner

Resultado Esperado: Funcionamiento de las herramientas

Evaluación de la Prueba: Aceptado

4.2.2.5. Resumen de iteraciones

Tabla 77.

Resumen de iteraciones

Iteración	Historia de Usuario	Fecha de Inicio	Fecha Fin	Estado
1	Login	19/12/2021	21/12/2021	Aprobado
2	Resumen	25/12/2021	29/12/2021	Aprobado
1	Sección Noticias	30/12/2021	31/12/2021	Aprobado
1	Control Notas o apuntes	03/01/2022	07/01/2022	Aprobado
1	Control Tareas	10/01/2022	31/01/2022	Aprobado
1	Control Recordatorios	01/02/2022	02/02/2022	Aprobado
2	Horario de clases	03/02/2022	05/02/2022	Aprobado
2	Control Asignaturas	07/02/2022	12/02/2022	Aprobado
2	Control Docentes	14/02/2022	18/02/2022	Aprobado
2	Calendario	21/02/2022	23/02/2022	Aprobado
2	Comunidad o Cartelera	24/02/2022	13/03/2022	Aprobado
2	Privacidad	14/03/2022	16/03/2022	Aprobado
2	Apariencia	16/03/2022	18/03/2022	Aprobado
2	Cuenta o perfil	18/03/2022	19/03/2022	Aprobado
2	Información	19/03/2022	20/03/2022	Aprobado
2	Herramientas	21/03/2022	01/04/2022	Aprobado

4.3 DISCUSIÓN

Para la discusión se tomó en cuenta el objetivo planteado que es desarrollar una agenda digital personalizada con herramientas de inteligencia artificial, para la organización de actividades académicas en la carrera de Computación de la Universidad Politécnica Estatal del Carchi por lo que se partió con la recolección de

información revisando diversas fuentes bibliográficas que ayudaron a construir un marco teórico y metodológico , mismo que sirvió como base para el diseño y desarrollo de una solución tecnológica que permita la organización de actividades académicas.

Aplicar un método mixto en la investigación permitió realizar una entrevista con la directora de la Carrera de Computación en la Universidad Politécnica Estatal del Carchi y una encuesta a los sujetos que conforman la población y muestra, con la aplicación de estos instrumentos de investigación, se identificó las características principales que debe tener la app de agenda digital como recordatorios, tareas, cartelera de actividades y el bloc de notas.

Para alcanzar el objetivo de esta investigación, se aplicó una metodología de desarrollo ágil, XP, por lo que, a través de historias de usuario, se continuo con la etapa de diseño, en donde se realizó prototipos o bosquejos de la aplicación móvil, una vez realizado este proceso se empezó la codificación de las funcionalidades especificadas en la fase de requerimientos, finalizando con la fase de pruebas, en la que se probó el código desarrollado. Poner en práctica todo este proceso dio como resultado una propuesta tecnológica que acompaña al estudiante en su proceso académico, logrando una mejor organización de sus actividades.

En la Universidad Central del Ecuador se desarrolló un sistema de acompañamiento para los estudiantes usando las TIC, cuyo objetivo fue conjuntamente con bienestar universitario detectar las deficiencias que presente el estudiante, y canalizar su ayuda, Castro (2018) asegura que acompañamiento al estudiante universitario basado en las TIC, mejora el desempeño académico del estudiante e invita a las Instituciones de Educación Superior del Ecuador a que se sumen a esta propuesta de acompañamiento al estudiante.

En comparativa, la propuesta de la Universidad Central se centra en mejorar el desempeño académico al igual que la presente solución tecnológica, la diferencia radica en la orientación de cada proyecto, la propuesta presentada por este equipo de investigadores se centra en una aplicación móvil tomando en cuenta que en los

resultados obtenidos la mayoría de los estudiantes poseen un teléfono, y esta herramienta es idónea para el uso frecuente.

Universidad Nacional del Noroeste de la Provincia de Buenos Aires, se generó una APP universitaria en donde los alumnos a través de este software puedan acceder a su información, tales como horarios de clase, lugar de cursos, exámenes, novedades de cursos, entre otros. Y de la misma manera los docentes puedan acceder a su respectiva información. Agregando que otra de las funcionalidades que implementa es la información de la infraestructura universitaria. En cuanto a resultados, los autores plantean que los beneficiarios cuentan con la información en cualquier lugar y tiempo posible, alegando que también este tipo de aplicativos aumentan un valor agregado a las universidades, ofreciendo un mejor servicio a sus usuarios o lo que quiere decir a su comunidad universitaria.

Este enfoque trasladado al presente proyecto destaca que las aplicaciones personalizadas para las Instituciones de Educación Superior dan “valor agregado” y un servicio de excelencia a sus estudiantes al momento de acceder a información y desarrollar sus actividades académicas.

Con estos resultados, se deja una referencia para futuros proyectos que pueden tomar como base el desarrollo de aplicaciones móviles para la organización de actividades académicas y analizar su impacto en el desempeño académico de la comunidad universitaria.

V. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- La información recolectada a través de la fundamentación teórica permitió construir una referencia sólida acerca de las agendas digitales para la organización de actividades académicas utilizando fuentes de información científica, estableciendo la relación entre la información existente y la investigación realizada.
- La metodología de desarrollo XP empleada en la elaboración de la propuesta tecnológica permitió a los investigadores crear un buen ambiente de trabajo en equipo además de una planificación, organización y ejecución adecuada por medio de cada una de las tareas e historias de usuario, cumpliendo con el proceso de desarrollo de software de forma ordenada finalizando con la etapa de pruebas para depurar errores.
- La elaboración de una agenda digital personalizada para la gestión de notas, tareas, recordatorios fue posible con la implementación de plugins obtenidos del administrador de paquetes Pub.dev que complementaron y ampliaron las funciones de la aplicación para la organización de actividades académicas del estudiante.
- Para la creación de un modelo de red neuronal de 3 capas y 24 neuronas se utilizó la biblioteca de código abierto para aprendizaje automático TensorFlow (Keras) y el lenguaje de programación Python3, todo esto contenido en una Interfaz de Programación de Aplicaciones (API) lanzadas a una plataforma de servicio de computación en la Nube, Heroku, para que la agenda digital consuma los servicios.
- El reconocimiento óptico de caracteres (OCR) necesitó de una biblioteca libre de visión artificial, como lo es OpenCV para el preprocesamiento de las imágenes adquiridas, la representación, descripción, reconocimiento e interpretación de la imagen se hace a través de un motor OCR de código abierto, Pytesseract, donde se detecta unidades de texto, partiendo desde líneas a palabras y posteriormente tratar de unirlos en función su contexto.

- La evaluación de los parámetros en la carga computacional en lo que se refiere a Inteligencia Artificial en un entorno de aprendizaje supervisado permitió identificar a la memoria RAM como un recurso que no se ve considerablemente afectado, al contrario de la CPU, donde cada valor inicial se diferenciaba al doble del resultado final. Dando como conclusión que un servidor con Memoria RAM de 4GB y un CPU de dos núcleos es aceptable para un prototipo de desarrollo, siempre y cuando el modelo de Inteligencia Artificial no sea relativamente grande, y los datos de entrenamiento no sean demasiado altos.

5.2 RECOMENDACIONES

- Se recomienda extender el proceso investigativo tomando como referencia la documentación generada en el presente proyecto, estudiar la posible implementación de la agenda digital personalizada para la organización de actividades académicas y medir el impacto que puede causar la solución tecnológica en el rendimiento académico de la comunidad universitaria.
- Se sugiere implementar metodologías de desarrollo de software ágiles debido a su alta flexibilidad y agilidad lo que permite resultados mucho más productivos y eficientes, además, admite adaptar el software a las necesidades que surjan en el camino facilitando construir aplicaciones que cumplen con los requerimientos del cliente.
- La agenda digital personalizada para la organización de actividades académicas puede funcionar en cualquier teléfono inteligente Android, sin embargo, es recomendable que el dispositivo tenga como mínimo un sistema operativo Android 9 y 4GB de memoria RAM para asegurar el correcto funcionamiento de sus módulos y obtener el máximo provecho a la aplicación móvil.
- Para la creación de un modelo de red neuronal se sugiere experimentar con su estructura de capas partiendo desde el número de neuronas contenidas, así como también su función de activación, todo esto antes de implementar el modelo en un proyecto de desarrollo para aumentar la exactitud de las predicciones y disminuir su magnitud de pérdida.

- En el reconocimiento óptico de caracteres (OCR) necesita de una imagen procesada correctamente en donde se hayan detectado bordes, eliminado ruido y aplicado filtros por lo que se recomienda establecer correctamente el valor del umbral para mejorar la eficiencia del texto obtenido.

VI. REFERENCIAS BIBLIOGRÁFICAS

- Acuña, D. (2018). *VISIÓN ARTIFICIAL APLICADA A LA DETECCIÓN E IDENTIFICACIÓN DE PERSONAS EN TIEMPO REAL* [Escuela Politécnica Nacional]. <http://bibdigital.epn.edu.ec/handle/15000/20098>
- Arias G, F. (2016). *El Proyecto de Investigación Introducción a la metodología científico*. file:///C:/Users/Jim/Downloads/ARIAS FIDIAS El proyecto de investigación - 7ma edición.pdf
- Arcos, G. (2020). *Entrevista, aplicaciones móviles para estudiantes*. Tulcán.
- Astobiza, A. (2021). Inteligencia Artificial para el bien común (AI4SG): IA y los Objetivos de Desarrollo Sostenible. *Arbor*, 197(802), a629–a629. <https://doi.org/10.3989/ARBOR.2021.802007>
- Barchino, A. (2020). *Qué es Heroku y principales características | IfgeekthenNTTdata*. <https://ifgeekthen.nttdata.com/es/Que es Heroku y principales características>
- Botía, M., & Marín, A. (2019). La contribución de los recursos audiovisuales a la educación . *Pedagogías Emergentes En La Sociedad Digital*, May, 91–102. <https://www.researchgate.net/publication/333093001>
- Carbo, M. (2020). *Tipos de algoritmos de Inteligencia Artificial y Machine Learning*.
- Cardenas Tutillo, J. C., & Quimbita Quingaluisa, F. E. (2017). ANÁLISIS, DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE UNA RED SOCIAL ORIENTADA A LA SEGURIDAD PARA LA EMPRESA CEFOSEG [Universidad Politécnica Salesiana - Quito]. In *Tesis*. <http://dspace.ups.edu.ec/bitstream/123456789/5081/1/UPS-CYT00109.pdf>
- Castro, A. (2018). *Propuesta de un sistema de acompañamiento a estudiantes universitarios basado en tecnologías de información y comunicación (TIC'S)*. 149. <http://www.dspace.uce.edu.ec/handle/25000/5483>
- Fernández, Y. (2019). *API: qué es y para qué sirve*. <https://www.xataka.com/basics/api-que-sirve>
- García, O. (2019). *Redes Neuronales artificiales: Qué son y cómo se entrenan | [site:name]*. <https://www.xeridia.com/blog/redes-neuronales-artificiales-que-son-y-como-se-entrenan-parte-i>
- Guaña, E., García, D., & Quinatoa, E. (2016). Utopía o realidad de aplicaciones

- informáticas en la educación. Caso Universidad Ecuatoriana. *Revista Publicando*, 3(9), 119–137.
- Hernández, R., Fernández, C., & Baptista, P. (2017). *Metodología de la investigación* (J. Mares (ed.); Sexta edición).
- INEC. (2016). *Contenido Ficha técnica Equipamiento del hogar*.
- INEC. (2018). *Tecnologías de la Información y Comunicación Encuesta Multipropósito-TIC 2018*. https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2018/201812_Principales_resultados_TIC_Multi_proposito.pdf
- Jiménez, D. (2016). *Implementación de apps educativas para dispositivos móviles orientado al aprendizaje del idioma inglés en los estudiantes del primer semestre de la carrera inglés, facultad de filosofía, letras y ciencias de la educación de la Universidad Central del Ecu.* 165. <http://www.dspace.uce.edu.ec/handle/25000/6615>
- Keras. (2022). *Acerca de Keras*. <https://keras.io/about/>
- Khawas, C., & Shah, P. (2018). Application of Firebase in Android App Development-A Study. *Article in International Journal of Computer Applications*, 179(46), 975–8887. <https://doi.org/10.5120/ijca2018917200>
- Lafuente, A. (2018). *Bases de datos relacionales vs. no relacionales: ¿qué es mejor? - Aukera*. <https://aukera.es/blog/bases-de-datos-relacionales-vs-no-relacionales/>
- León, D., & Martínez, J. (2020). *Tendencias Tecnológicas*.
- Li, M., & Nielsen, P. (2019). *Making Usable Generic Software - The Platform Appliances Approach*. May, 15. <https://doi.org/10.13140/RG.2.2.11381.83687>
- López, M. (2018). *Cuatro herramientas para hacer Inteligencia Artificial en el 2018*. <https://www.unocero.com/ciencia/cuatro-herramientas-para-hacer-inteligencia-artificial-en-el-2018/>
- Marquez, J., & Fernandez, A. D. R. (2019). *ANÁLISIS DE LAS METODOLOGÍAS ÁGILES DE DESARROLLO DE SOFTWARE*. UNIVERSIDAD TÉCNICA DE BABAHOYO.
- Martínez, C., & Cruz-Gavilánez, Y. de la N. (2018). Tendencias tecnológicas y desafíos de la seguridad informática. *Polo Del Conocimiento*, 3(5), 260. <https://doi.org/10.23857/pc.v3i5.640>
- Martinez, J. (2020). *¿Qué es un Optimizador y Para Qué Se Usa en Deep Learning? -*

- DataSmarts Español*. <https://datasmarts.net/es/que-es-un-optimizador-y-para-que-se-usa-en-deep-learning/>
- Martins, A., Fracchia, C., Allan, C., Parra, S., Baeza, N., Celeste, C., Mamani, N., Pascual, K., De Armiño, A., & Laurent, R. (2019). Desarrollo de Herramientas Informáticas y sus Aplicaciones en el Ámbito Educativo. *In XXI Workshop de Investigadores En Ciencias de La Computación (WICC 2019, Universidad Nacional de San Juan)*, 14.
- Meléndez Valladarez, S. M., Pérez Reyes, N. N., & Gaitan, E. M. (2016). *METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACION EXTREMA*. [UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA, MANAGUA]. <http://repositorio.unan.edu.ni/1365/1/62161.pdf>
- Menacho, Á. (2020). *EventYou: Aplicación móvil con Flutter y FlutterFire*. <https://idus.us.es/bitstream/handle/11441/102882/TFG-3024-MENACHO RODRIGUEZ.pdf?sequence=1&isAllowed=y>
- Ministerio de Educación. (2017). *Agenda Educativa Digital – Ministerio de Educación*.
- Montero, B., Cevallos, H., & Dávila, J. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espirales Revista Multidisciplinaria de Investigación ISSN: 2550-6862*, 2(17), 114–121. <https://doi.org/https://doi.org/10.1007/BF03194495>
- Moran, M., Ramos, F., Mantuano, M., & Litardo, J. (2019). IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN ACADÉMICA (SGA) EN EL INSTITUTO TECNOLÓGICO SUPERIOR JUAN BAUTISTA AGUIRRE. *Revista Científica Aristas*, 1(69), 106–121. <https://doi.org/2600-5662>
- Moreno, J. (2018). *Entornos de desarrollo*. Editorial Síntesis. www.sintesis.com
- Muente, G. (2020). *Framework: ¿qué es y cuál es su función en Internet?* <https://rockcontent.com/es/blog/framework/>
- Pedro, H. (2018). *La relación entre Librería, API, Framework y SDK en el desarrollo de Apple*. | by Pedro Hernández | Medium. <https://medium.com/@pedrohr99/la-relacion-entre-libreria-api-framework-y-sdk-en-el-desarrollo-de-apple-b8c971cf06e3>
- Piña, E., & Torrealba, L. (2020). VIRTUAL LEARNING ENVIRONMENTS (EVA

-). A TEACHING STRATEGY FOR THE ADMINISTRATION OF UNIVERSITY CURRICULAR UNITS . Universidad Pedagógica Experimental Libertador (UPEL). Venezuela Este escrito describe algunos aspectos medulares de un estudio de campo d. *REDIUNP*, 1(c), 18–27.
- Pinto, E. (2020). *【 Metodología XP 】 O PROGRAMACION EXTREMA DE SOFTWARE*. <https://eluniversodelmarketing.com/c-metodologias/metodologia-xp/>
- Purizaga, J. (2019). *Gestión de Bases de Datos* (Primera Ed).
- Purizaga, J. (2020). Modelos Del Rendimiento En Gestores De Base De Datos NoSQL. In *Universidad Católica Los Ángeles de Chimbote*. Universidad Católica Los Ángeles de Chimbote.
- PYPL. (2020). *PYPL Popularity of Programming Language index*. <http://pypl.github.io/PYPL.html>
- Revista Robots. (2020). *Inteligencia Artificial y Realidad Virtual. Una combinación letal*. <https://revistaderobots.com/noticias/inteligencia-artificial-y-realidad-virtual-una-combinacion-letal/>
- Reyes Cuba, P. P., & Marin Apaza, R. P. (2021). APLICACIÓN WEB EMPLEANDO LA METODOLOGÍA XP PARA LA GESTIÓN ACADÉMICA DEL INSTITUTO DE INFORMÁTICA DE LA UNIVERSIDAD NACIONAL DEL ALTIPLANO PUNO - 2019 [UNIVERSIDAD NACIONAL DEL ALTIPLANO]. In *2021*. <http://tesis.unap.edu.pe/handle/UNAP/16057>
- Rivas, W., Mazon, B., & Mejía, E. (2018). Generalidades de las redes neuronales artificiales. *Redes Neuronales Artificiales Aplicadas Al Reconocimiento de Patrones*, 11–35. <http://repositorio.utmachala.edu.ec/handle/48000/12499>
- Rodríguez, A., & Pérez, A. (2017). Métodos científicos de indagación y de construcción del conocimiento. *Revista EAN*, 82, 179–200.
- Rodríguez, B., & Castillo, C. (2019). Entornos virtuales de aprendizaje: posibilidades y retos en el ámbito universitario. In *Entornos virtuales de aprendizaje: posibilidades y retos en el ámbito universitario* (Vol. 14). Ediciones de la Universidad de Castilla-La Mancha. <https://doi.org/10.18239/atena.14.2019>
- Rodriguez, H. (2021). *¿Qué es OpenCV y para qué sirve?* [2021]. <https://www.crehana.com/pe/blog/desarrollo-web/que-es-opencv/>
- Ruiz, K. (2019). *Pyesseract: Comience con OCR - Swapps*.

- <https://swapps.com/es/blog/pytesseract-comience-con-ocr/>
- Sanchez, A. (2020). *La librería Numpy | Aprende con Alf*.
<https://aprendeconalf.es/docencia/python/manual/numpy/>
- Sangacha, Lady, Varela, E., Ortiz, J., Ortiz, I., Masabanda, L., & Ferruzola, W. (2019). Aplicación móvil en los ambientes educativos en Ecuador: “monitoreando tareas y asistencia en tiempo real.” *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, 2019(E22), 29–40.
- Serna, E. (2017). *DESARROLLO E INNOVACIÓN EN INGENIERÍA* (Segunda ed).
https://www.researchgate.net/profile/Jhon_Fredy_Narvaez/publication/320170890_Desarrollos_de_la_Ingenieria_ambiental_en_la_evaluacion_de_la_calidad_de_los_recurso_naturales_y_la_salud_ambiental/links/59d26bfca6fdcc181ad611ce/Desarrollos-de-la-Ingenieria-
- Suárez, A., Patricio, C., & Barco, M. (2016). *Análisis de bases de datos y tendencias tecnológicas*. Universidad de Alicante.
- Teknei. (2018). *El Software Personalizado: cómo y cuándo aplicarlo en las empresas – Teknei*. <https://www.teknei.com/2022/02/01/el-software-personalizado-como-y-cuando-aplicarlo-en-las-empresas/>
- Telefónica, F. (2019). *Programación, hablando el idioma de las máquinas*. 23.
www.fundaciontelefonica.com/publicaciones
- Valerio, J. (2018). *Que es un framework y para que se utiliza en programación*.
<https://comenzandodecero.com/que-es-un-framework-y-para-que-se-utiliza-en-programacion/>
- Zumba, J. (2018). Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software. *INNOVA Research Journal*, 3(10), 20–33.
<https://doi.org/10.33890/innova.v3.n10.2018.651>
- Zurita, B. N. (2020). Sistema Web Para La Gestión Académica Y Administrativa De Empresa De Capacitación Profesional Dienav. *Universidad Tecnológica Israel*.

VII. ANEXOS

Anexo 1 Acta de predefensa de Malkik Anrango

	UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES CARRERA DE COMPUTACION	
ACTA		
DE LA SUSTENTACIÓN DE PREDEFENSA DEL DEL TRABAJO DE INTEGRACIÓN CURRICULAR:		
NOMBRE: Malkik Salvador Anrango Oyagata	CÉDULA DE IDENTIDAD: 1004642805	
NIVEL/PARALELO: 0	PERIODO ACADÉMICO: PAO 2021B	
TEMA DEL TIC:	Agenda digital personalizada para la organización de actividades académicas	
Tribunal designado por la dirección de esta Carrera, conformado por:		
PRESIDENTE:	MSC. CARLITOS ALBERTO GUANO CÁRDENAS	
DOCENTE TUTOR:	MSC. JORGE HUMBERTO MIRANDA REALPE	
DOCENTE:	MSC. GEORGINA GUADALUPE ARCOS PONCE	
De acuerdo al artículo 32: Una vez entregados los documentos; y, cumplidos los requisitos para la realización de la pre-defensa el Director/a de Carrera designará el Tribunal, fijando lugar, fecha y hora para la realización de este acto:		
EDIFICIO DE AULAS 4	AULA:	209
FECHA:	martes, 12 de abril de 2022	
HORA:	08H15	
Obteniendo las siguientes notas:		
1) Sustentación de la predefensa:		5,60
2) Trabajo escrito		2,33
Nota final de PRE DEFENSA		7,93
Por lo tanto:	APRUEBA CON OBSERVACIONES	; debiendo acatar el siguiente artículo:
Art. 36.- De los estudiantes que aprueban el informe final del TIC con observaciones.- Los estudiantes tendrán el plazo de 10 días para proceder a corregir su informe final del TIC de conformidad a las observaciones y recomendaciones realizadas por los miembros del Tribunal de sustentación de la pre-defensa.		
Para constancia del presente, firman en la ciudad de Tulcán el martes, 12 de abril de 2022		
 MSC. CARLITOS ALBERTO GUANO CÁRDENAS PRESIDENTE		
 MSC. JORGE HUMBERTO MIRANDA REALPE DOCENTE TUTOR		
 MSC. GEORGINA GUADALUPE ARCOS PONCE DOCENTE		
Adj.: Observaciones y recomendaciones		

Anexo 2 Acta de predefensa de Noreen Rosero



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES
CARRERA DE COMPUTACION



ACTA

DE LA SUSTENTACIÓN DE PREDEFENSA DEL TRABAJO DE INTEGRACIÓN CURRICULAR:

NOMBRE Noreen Odalys Rosero Acosta
NIVEL/PARALELO: 0

CÉDULA DE IDENTIDAD 1726169335
PERIODO ACADÉMICO PAO 2021B

TEMA DEL TIC: Agenda digital personalizada para la organización de actividades académicas

Tribunal designado por la dirección de esta Carrera, conformado por:
PRESIDENTE: MSC. CARLITOS ALBERTO GUANO CÁRDENAS
DOCENTE TUTOR: MSC. JORGE HUMBERTO MIRANDA REALPE
DOCENTE: MSC. GEORGINA GUADALUPE ARCOS PONCE

De acuerdo al artículo 32: Una vez entregados los documentos; y, cumplidos los requisitos para la realización de la pre-defensa el Director/a de Carrera designará el Tribunal, fijando lugar, fecha y hora para la realización de este acto:

EDIFICIO DE AULAS 4 **AULA:** 209
FECHA: martes, 12 de abril de 2022
HORA: 08H15

Obteniendo las siguientes notas:
1) Sustentación de la predefensa: 5,60
2) Trabajo escrito 2,33
Nota final de PRE DEFENSA 7,93
Por lo tanto: **APRUEBA CON OBSERVACIONES** ; debiendo acatar el siguiente artículo:

Art. 36.- De los estudiantes que aprueban el informe final del TIC con observaciones.- Los estudiantes tendrán el plazo de 10 días para proceder a corregir su informe final del TIC de conformidad a las observaciones y recomendaciones realizadas por los miembros del Tribunal de sustentación de la pre-defensa.

Para constancia del presente, firman en la ciudad de Tulcán el martes, 12 de abril de 2022


MSC. CARLITOS ALBERTO GUANO CÁRDENAS
PRESIDENTE


MSC. JORGE HUMBERTO MIRANDA REALPE
DOCENTE TUTOR


MSC. GEORGINA GUADALUPE ARCOS PONCE
DOCENTE

Adj.: Observaciones y recomendaciones

Anexo 3 Certificado del abstract por parte de idiomas



**UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
FOREIGN AND NATIVE LANGUAGE CENTER**

ABSTRACT- EVALUATION SHEET				
NAME: Anrango Oyagata Malkik Salvador y Rosero Acosta Noreen Odalys				
DATE: 20 de abril de 2022				
TOPIC: "Agenda digital personalizada para la organización de actividades académicas ."				
MARKS AWARDED		QUANTITATIVE AND QUALITATIVE		
VOCABULARY AND WORD USE	Use new learnt vocabulary and precise words related to the topic	Use a little new vocabulary and some appropriate words related to the topic	Use basic vocabulary and simplistic words related to the topic	Limited vocabulary and inadequate words related to the topic
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
WRITING COHESION	Clear and logical progression of ideas and supporting paragraphs.	Adequate progression of ideas and supporting paragraphs.	Some progression of ideas and supporting paragraphs.	Inadequate ideas and supporting paragraphs.
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
ARGUMENT	The message has been communicated very well and identify the type of text	The message has been communicated appropriately and identify the type of text	Some of the message has been communicated and the type of text is little confusing	The message hasn't been communicated and the type of text is inadequate
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
CREATIVITY	Outstanding flow of ideas and events	Good flow of ideas and events	Average flow of ideas and events	Poor flow of ideas and events
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input checked="" type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
SCIENTIFIC SUSTAINABILITY	Reasonable, specific and supportable opinion or thesis statement	Minor errors when supporting the thesis statement	Some errors when supporting the thesis statement	Lots of errors when supporting the thesis statement
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input checked="" type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
TOTAL/AVERAGE	9 - 10: EXCELLENT 7 - 8,9: GOOD 5 - 6,9: AVERAGE 0 - 4,9: LIMITED		TOTAL 9	



**UNIVERSIDAD POLITÉCNICA ESTATAL DEL
CARCHI FOREIGN AND NATIVE LANGUAGE
CENTER**

Informe sobre el Abstract de Artículo Científico o Investigación.

Autor: Anrango Oyagata Malkik Salvador y Rosero Acosta Noreen Odalys

Fecha de recepción del abstract: 20 de abril de 2022

Fecha de entrega del informe: 20 de abril de 2022

El presente informe validará la traducción del idioma español al inglés si alcanza un porcentaje de: 9 – 10 Excelente.

Si la traducción no está dentro de los parámetros de 9 – 10, el autor deberá realizar las observaciones presentadas en el ABSTRACT, para su posterior presentación y aprobación.

Observaciones:

Después de realizar la revisión del presente abstract, éste presenta una apropiada traducción sobre el tema planteado en el idioma Inglés. Según los rubrics de evaluación de la traducción en Inglés, ésta alcanza un valor de 9, por lo cual se valida dicho trabajo.

Atentamente




Ing. Edison Peñafiel Arcos MSc
Coordinador del CIDEN

Anexo 4 Informe de Turnitin

14/2/22, 08:55

Turnitin Informe de Originalidad

 **Informe de Originalidad Turnitin**

Proyecto Investigación por Anrango
Rosero
Desde proyectos (Titulación)

Índice de similitud 9%	Similitud según fuente	
	Internet Sources:	8%
	Publicaciones: Trabajo del estudiante:	1% 6%

Procesado el 14-feb.-2022 08:35 -05
Identificador: 1762118121
Número de palabras: 23385

fuentes:

- 1 4% match ()
[Meléndez Valladares, Sinthya Milena, Gaitán, María Elizabeth, Pérez Reyes, Neldin Noel, "Sistema WEB de evaluación al desempeño Docente UNAN-Managua, empleando la metodología Agil Programación Extrema, en el II Semestre del 2015", 2016](#)
- 2 1% match ()
[López Reyes, Alberto José, Hernández Aguirre, Leonel Antonio, "Aplicación Web para gestión y control de compra-venta de medicamentos de farmacia Tania, Managua, segundo Semestre 2017", 2018](#)
- 3 1% match ()
[Castillo Barillas, Diana del Carmen, Navarrete Vásquez, Henry Javier, "Sistema web para el control del concurso al mejor educador de la Universidad Nacional Autónoma de Nicaragua, Managua, UNAN-Managua, aplicando la metodología ágil XP en el II semestre del año 2015", 2016](#)
- 4 1% match (Internet desde 07-ago.-2020)
<http://docplayer.es/80214920-Departamento-de-ciencias-de-la-computacion.html>
- 5 < 1% match (Internet desde 30-jun.-2015)
<http://repositorio.ucsg.edu.ec/bitstream/123456789/1342/1/T-UCSG-PRE-ING-CIS-81.pdf>
- 6 < 1% match (Internet desde 11-dic.-2018)
[http://repositorio.ucv.edu.pe/bitstream/handle/UCV/23724/a%C3%B1azgo_rj.pdf?sequence="](http://repositorio.ucv.edu.pe/bitstream/handle/UCV/23724/a%C3%B1azgo_rj.pdf?sequence=)
- 7 < 1% match (publicaciones)
[\(Carilinda Leite, and Miguel Zabalza\), "Ensino superior: inovação e qualidade na docência", Repositório Aberto da Universidade do Porto, 2012](#)
- 8 < 1% match (trabajos de los estudiantes desde 02-jul.-2020)
[Submitted to Universidad Cesar Vallejo on 2020-07-02](#)
- 9 < 1% match (trabajos de los estudiantes desde 24-ago.-2021)
[Submitted to Pontificia Universidad Catolica del Ecuador - PUCE on 2021-08-24](#)
- 10 < 1% match (Internet desde 12-oct.-2020)
<http://repositorio.upec.edu.ec/bitstream/123456789/889/1/464%20Aplicaci%C3%B3n%20de%20los%20reg%C3%ADmenes%20aduaner>
- 11 < 1% match (Internet desde 26-oct.-2020)
<https://feriacientificaupe.wixsite.com/feriacientificapec/post/ruta-de-las-orqu%C3%ADdeas>
- 12 < 1% match (Internet desde 11-dic.-2020)
https://repositorio.uta.edu.ec/bitstream/123456789/29604/1/Tesis_t1561msi.pdf
- 13 < 1% match ()
[Alvarez Morales, Luis Alberto, "Sistema basado en la norma ISO 15489 para la generación automática de documentos legales en la Notaría del Villar - El Agustino, 2018", "Universidad Cesar Vallejo", 2018](#)
- 14 < 1% match (Internet desde 02-ene.-2021)
<http://repositorio.uo.edu.ec/bitstream/123456789/45271/1/B-CISC-PTG-1686%20Guam%3AIn%20Yumaglla%20-%20Castro%20L%3Apez%20Jordy%20Andr%3Aa9s.pdf>

texto del trabajo:

I. PROBLEMA 1.1. PLANTEAMIENTO DEL PROBLEMA En la última década es evidente que las aplicaciones informáticas gobiernan el diario vivir de las personas, de modo que su utilización se ve reflejado en las computadoras y celulares inteligentes. Dicha tecnología está inmersa en diferentes aspectos, tales como educación, entretenimiento, ocio, profesión, etc. De manera que si se clasifica según su utilización son muchos factores para considerar. En el caso de esta investigación se enfocará a lo

https://www.turnitin.com/newreport_printview.asp?eq=1&eb=1&esm=30&oid=1762118121&sid=0&n=0&m=0&svr=25&r=25.946775206362727&l... 1/30

Anexo 5 Certificado de finalización del proyecto



Tulcán, 04 abril de 2022

CERTIFICACIÓN

Por medio del presente y en mi calidad de DIRECTORA DE LA CARRERA DE COMPUTACIÓN me permito certificar la culminación del proyecto de investigación presentado por la Srta. ROSERO ACOSTA NOREEN ODALYS y el Sr. MALKIK SALVADOR ANRANGO OYAGATA egresados de la Carrera de Computación denominado AGENDA DIGITAL PERSONALIZA PARA LA ORGANIZACIÓN DE ACTIVIDADES ACADÉMICAS, mismo que se ha realizado con todo lo solicitado por la Carrera.

Es todo lo que puedo certificar en honor a la verdad.

Atentamente,



MSc. Georgina Arcos P.
DIRECTORA CARRERA DE COMPUTACIÓN
"EDUCAMOS PARA TRANSFORMAR EL MUNDO"

Anexo 6 Encuesta a los estudiantes de Computación



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS
AMBIENTALES
CARRERA DE COMPUTACIÓN

I. INTRODUCCIÓN

Buen día, somos estudiantes de la Facultad de Industrias Agropecuarias y Ciencias Ambientales y con el fin de determinar el desarrollo de una agenda digital personalizada para el estudiante de computación de la Universidad Politécnica Estatal del Carchi, solicitamos su colaboración llenando esta encuesta, encerrando en círculo la respuesta de su elección. De antemano, muchas gracias.

II. DATOS DE CLASIFICACIÓN

Género: Masculino Femenino LGBT+

Edad: 17- 20 años 21-23 años 24-26 años 27- en adelante

III. CUESTIONARIO

1. ¿Con que medios tecnológicos se apoya su proceso de aprendizaje?

- a. Celular
- b. Computadora Portátil o de Escritorio
- c. Tabletas
- d. Otros.

2. ¿Cuánto tiempo destina en el uso de los instrumentos tecnológicos anteriormente mencionados para su proceso de aprendizaje?

- a. 15 a 30 minutos
- b. 1 a 2 horas
- c. 3 a 4 horas
- d. 5 o más

3. ¿A través de que herramientas tecnológicas adquiere conocimiento en su vida académica?

- a. Aulas virtuales
- b. Páginas web
- c. Libros digitales
- d. Videos en la internet
- e. Repositorios digitales
- f. Otros

- 4. ¿Qué habilidades ha obtenido durante su proceso de aprendizaje utilizando medios digitales?**
- a. Lectura
 - b. Redacción
 - c. Trabajo en equipo
 - d. Expresión oral y escrita
 - e. Organización personal
 - f. Otros
- 5. ¿Qué programas informáticos utiliza en su proceso de aprendizaje?**
- a. Programas Ofimáticos (Word, Excel, Power Point)
 - b. Programas de videoconferencia (Microsoft Teams, Zoom, Google Meet, etc.)
 - c. Agendas digitales (Google Calendar, Evernote, Do it Tomorrow, etc.)
 - d. Otros.
- 6. ¿Qué recursos digitales utiliza en un periodo académico ordinario?**
- a. Documentos Digitales (PDF, Docx, epub, etc)
 - b. Imágenes o Fotos
 - c. Audio
 - d. Otros
- 7. ¿Qué tipos de trabajos digitales o entregables realiza en un semestre?**
- e. Documentos Ofimáticos
 - f. Imágenes o Fotos propias
 - g. Audio
 - h. Otros
- 8. ¿Actualmente cuenta con un teléfono que soporta instalación de APP, como una agenda digital?**
- a. Si
 - b. No
- 9. Escoja entre las opciones, el sistema operativo que tiene su celular actualmente**
- a. Android
 - b. IOS
- 10. ¿Ha utilizado usted, algún un medio digital para la organización de sus actividades académicas y personales?**
- a. Si

b. No

Por qué:

11. Escoja entre las opciones disponibles el tipo de organización que ha usado

- a. Agenda Física (De papel)
- b. Agenda Electrónica (Dispositivo electrónico)
- c. Agenda Digital (Aplicación instalada en su celular)
- d. Agenda Online (Aplicación web)
- e. App Móvil (Aplicación móvil diferente a una agenda)
- f. Cuaderno físico o libretas

12. ¿Como califica el apoyo que le brinda la herramienta de organización personal y/o académico?

- a. Mala
- b. Regular
- c. Buena
- d. Muy Buena
- e. Excelente

13. ¿Considera importante la habilidad de organización personal por medio de una agenda digital en su celular?

- a. Si
- b. No

Por qué:

14. ¿Estaría dispuesto a instalar una agenda digital gratuita en su celular?

- a. Si
- b. No

Por qué:

15. ¿Qué funcionalidades básicas consideraría para su agenda que satisfagan sus necesidades?

- a. Agendar Tareas
- b. Recordatorios
- c. Bloc de notas
- d. Cartelera de Actividades

e. Otros (Cuáles):

16. Que otras características en función de la tecnología actual consideraría implementar en una agenda digital.

- a. Inteligencia artificial (Aprendizaje Automático)
- b. Personalizable (El usuario podrá decidir en temas de diseño)
- c. Funcionamiento fuera de línea (No es necesaria la conexión a Internet)
- d. Actualizaciones Regulares (Mejoras constantes a la aplicación)

Aprobado por Msc. Jorge Miranda (Tutor)



110040_4340246410000004_0001
JORGE HUMBERTO
MIRANDA REALPE

Firma



MANUAL DE USUARIO

Norkik App es una agenda educativa que fue creada para ayudarte a organizar tu vida académica y tener todo bajo control.

Una solución tecnológica de estudiantes para estudiantes.

Introducción

Norkik App es una práctica aplicación para los estudiantes de la Carrera de Computación que fue creada para ayudarlos a organizar su vida académica y tener todo bajo control. La aplicación permitirá anotar tareas, exámenes y recordatorios de forma muy rápida y las notificaciones ayudarán a no olvidar nada. El calendario integrado está optimizado para las necesidades de los estudiantes y permite gestionar eventos y actividades con mayor facilidad.

Será posible también crear horarios de clases y planificar los estudios en consecuencia. Personalizar el horario se podrá asignar diferentes colores a cada asignatura y ver los eventos relacionados en el calendario.

Requisitos del hardware

Se han establecido requisitos mínimos de hardware para el correcto funcionamiento, como:

- Sistema Operativo Android 9 o mayor
- RAM 4 GB
- Conexión a Internet

Requisitos de software

Para asegurar un adecuado funcionamiento de la aplicación móvil se han establecido los siguientes requisitos mínimos de software:

- Teléfono móvil inteligente.
- 165 MB de espacio de almacenamiento.
- Procesador quad-core de 1.2 GHz, o uno más rápido.

INCIO DE SESIÓN

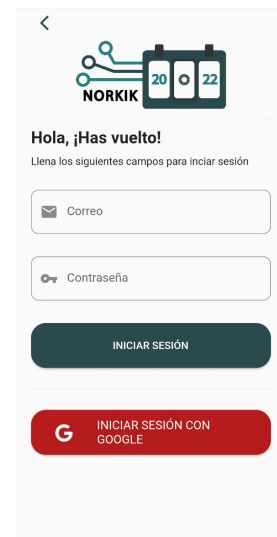


Para acceder al uso de la aplicación móvil se debe de tener instalada la aplicación en el teléfono móvil, en donde al ejecutarla se le presentará la pantalla principal de Norkik App.

1. Botón Iniciar Sesión
2. Botón Registro

La pantalla de inicio de sesión tiene dos opciones de ingreso.

1. Ingresar el correo electrónico y la contraseña.
2. Iniciar Sesión con Google



Si no dispone de una cuenta de usuario en esta pantalla puede crear una nueva para poder acceder a la aplicación móvil. Debe de llenar todos los campos con los datos para su perfil de usuario.

RESUMEN

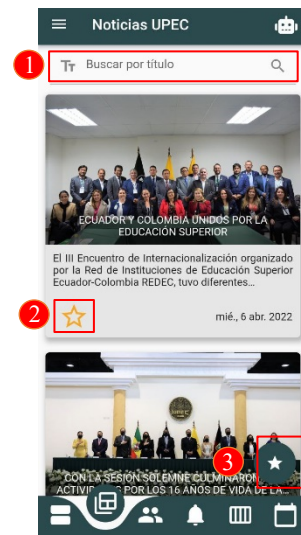


Esta es la primera pantalla que se muestra al iniciar sesión. En esta sección se muestra un reporte semanal, detallando las actividades y eventos próximos.

NOTICIAS UPEC

En Noticias UPEC se despliegan las últimas publicaciones y noticias sobre la Carrera y la Universidad.

- 1. Barra de Búsqueda.** - Permitirá buscar noticias por su título
- 2. Favorito.** - Botón que permitirá seleccionar la noticia como favorita y almacenarla
- 3. Historial de favoritos.** - Lista de noticias seleccionadas como favoritas.



COMUNIDAD



En Comunidad se despliegan las publicaciones de los usuarios que son de interés.

- 1. Favorito.** - Botón que permitirá seleccionar la noticia como favorita
- 2. Opciones.** - Lista de opciones para comunidad.
 - **Buscar.**- Permite buscar publicaciones
 - **Publicar.**- Permite Agregar publicaciones a la sección Comunidad

RECORDATORIOS



En la sección Recordatorios se despliega una lista ordenada de los eventos próximos, con la posibilidad de editarlos o eliminarlos:

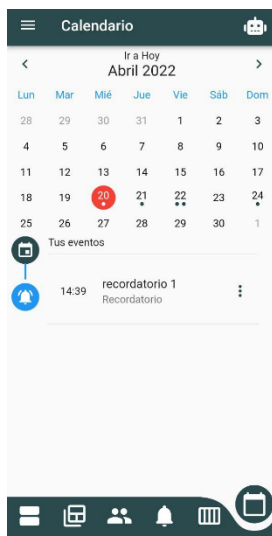
1. **Línea de tiempo.** - Filtro para recordatorios por fechas.
2. **Crear.** - Botón para crear nuevo recordatorio.

HORARIO DE CLASES



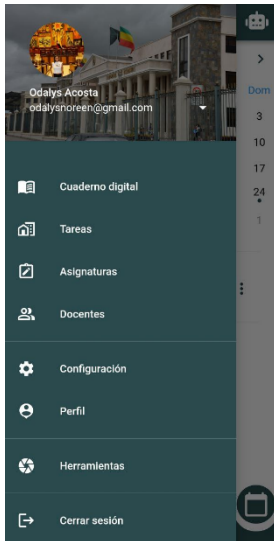
La sección horario permite ver de forma personalizada las clases de la semana, diferenciando a cada materia con colores.

COMUNIDAD



En la sección Calendario se encontrarán las actividades pendientes por día, con la opción de navegar a través del calendario a otros meses y años.

MENÚ LATERAL



Esta es la primera pantalla que se muestra al iniciar sesión. En esta sección se muestra un reporte semanal, detallando las actividades y eventos próximos.

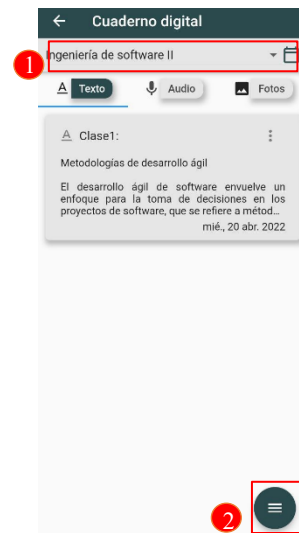
CUADERNO DIGITAL

El cuaderno digital permite guardar de forma ordenada texto, audio e imágenes sobre una materia.

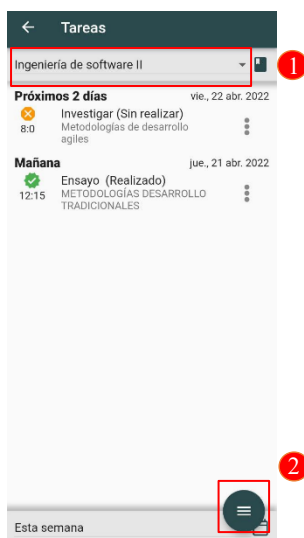
1. Barra de Selección. – Permite elegir la asignatura

2. Opciones. - Lista de opciones para comunidad.

- **Buscar.**- Permite buscar publicaciones
- **Publicar.**- Permite agregar texto, audio o fotos al Cuaderno Digital



TAREAS



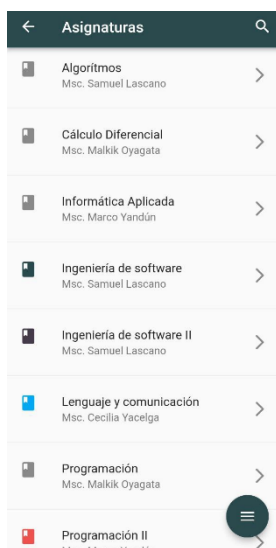
En la sección Tareas se visualizan las tareas pendientes, diferenciadas por colores.

1. Filtro. – Lista de asignaturas

2. Opciones. - Lista de opciones para comunidad.

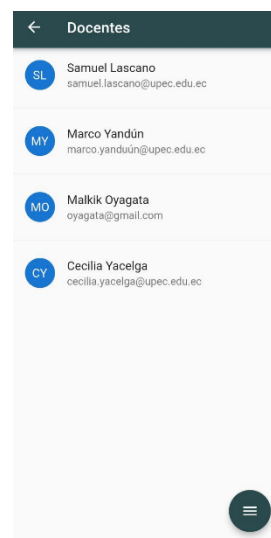
- **Buscar.**- Permite buscar publicaciones
- **Crear.**- Permite crear nuevas tareas

ASIGNATURAS



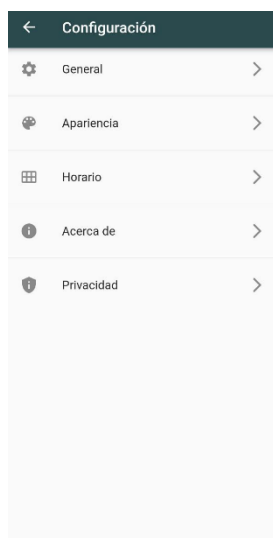
El apartado asignaturas muestra una lista detallada de todas las materias del Pensum y la opción de agregar nuevas asignaturas a la lista.

DOCENTES



El apartado docentes muestra una lista detallada de todos los docentes de la Carrera de Computación y la opción de agregar nuevos docentes a la lista

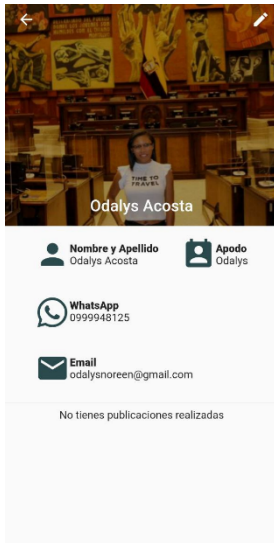
CONFIGURACIÓN



En la sección Configuración se podrá encontrar:

- **General:** Se elegirá la pantalla principal de la aplicación.
- **Apariencia:** Permite cambiar el color y fuente de la app.
- **Horario:** Gestiona los horarios existentes.
- **Acerca de:** Muestra la información de la app.
- **Privacidad:** Permite habilitar la información privada que se muestra en la aplicación.

PERFIL



En el perfil del usuario encontrará la información que ingresó en su registro y podrá editarla.

HERRAMIENTAS

En las herramientas digitales se encuentra:

- **OCR:** Una herramienta para convertir imágenes a texto.
- **Escáner:** Una herramienta que permite escanear documentos desde el celular y generar un archivo pdf.
- **Lector QR:** Lector de códigos QR (Quick Response) son códigos de barras, capaces de almacenar determinado tipo de información, como una URL, SMS, email, Texto, etc.

