



UNIVERSIDAD TÉCNICA DE COTOPAXI
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS
CARRERA DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS
COMPUTACIONALES

PROYECTO DE INVESTIGACIÓN

**“APLICACIÓN DE LA INTELIGENCIA ARTIFICIAL EN EL DESARROLLO
DE SOFTWARE”**

Proyecto de Investigación presentado previo a la obtención del Título de Ingenieros en
Informática y Sistemas Computacionales

AUTORES:

Sarco Calapiña Wendy Magaly

Toaquiza Pacheco Darwin Vladimir

TUTORA:

Dra. Mayra Susana Albán Taipe

LATACUNGA – ECUADOR

Marzo 2022



DECLARACIÓN DE AUTORÍA

Nosotros: **Sarco Calapiña Wendy Magaly** con C.I. **180537353-5** y **Toaquiiza Pacheco Darwin Vladimir** con C.I. **171937989-1** declaramos ser autores del presente proyecto de investigación: “**APLICACIÓN DE LA INTELIGENCIA ARTIFICIAL EN EL DESARROLLO DE SOFTWARE**”, siendo la **Dra. Mayra Susana Albán Taipe** tutora del presente trabajo; y eximimos expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además, certificamos que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de nuestra exclusiva responsabilidad.

Sarco Calapiña Wendy Magaly
C.I.: 180537353-5

Toaquiiza Pacheco Darwin Vladimir
C.I.: 171937989-1



AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN

En calidad de Tutor del Trabajo de Investigación sobre el título: **“APLICACIÓN DE LA INTELIGENCIA ARTIFICIAL EN EL DESARROLLO DE SOFTWARE**, de los estudiantes **Sarco Calapiña Wendy Magaly** con C.I. **180537353-5** y **Toaquiza Pacheco Darwin Vladimir** con C.I. **171937989-1**, de la carrera de Ingeniería en Informática y Sistemas Computacionales, considero que dicho Informe Investigativo cumple con los requerimientos metodológicos y aportes científico-técnicos suficientes para ser sometidos a la evaluación del Tribunal de Validación de Proyecto que el Consejo Directivo de la Facultad de Ciencias de la Ingeniería y Aplicadas, de la Universidad Técnica de Cotopaxi designe, para su correspondiente estudio y calificación.

Latacunga, Marzo 2022

Dra. Albán Taipe Mayra Susana

C.I: 050231198-8




APROBACIÓN DEL TRIBUNAL DE TITULACIÓN


En calidad de Tribunal de Lectores, aprueban el presente Informe de Investigación de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y por la Facultad de Ciencias de la Ingeniería y Aplicadas; por cuanto, los postulantes: **Sarco Calapiña Wendy Magaly** y **Toaquiza Pacheco Darwin Vladimir** con el título de Proyecto de Investigación: “**APLICACIÓN DE LA INTELIGENCIA ARTIFICIAL EN EL DESARROLLO DE SOFTWARE**”, han considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de Sustentación de Proyecto.

Por lo antes expuesto, se autoriza realizar los empastados correspondientes, según la normativa institucional.

Latacunga, Marzo 2022

Para constancia firman:


.....
Lector 1 (Presidente)
MsC. Manuel Villa Quishpe
CC:180338695-0


.....
Lector 2
MsC. Maira Martínez
CC: 171250776-1


.....
Lector 3
MsC. Karla Susana Cantuña Flores
CC: 050230511-3



AGRADECIMIENTO

A Dios por haberme permitido llegar a este punto brindándome con su infinita bondad y amor salud y vida para cumplir con mis objetivos.

A la Universidad Técnica de Cotopaxi por permitirme ser parte de esta institución y haberme brindado la oportunidad de estudiar y formarme como profesional.

A mis docentes por haberme guiado y compartido sus conocimientos y experiencias a lo largo de mi formación profesional.

A mis padres y hermano por su apoyo incondicional en todos los sentidos durante este trayecto. Y a mis padrinos Rosa y Fernando quienes también me han brindado su apoyo para lograr cumplir con mi objetivo.

Magaly



DEDICATORIA

El presente trabajo de investigación lo dedico a:

Mis padres Lucrecia y Raúl quienes me han apoyado y motivado en el transcurso de mis estudios universitarios. A mi madre que con sus consejos, sacrificio, paciencia y amor me ha enseñado a ser una persona de bien. A mi hermano Alexander por su apoyo, por los buenos y malos momentos compartidos.

A toda mi familia por su apoyo y consejos en los momentos más difíciles, ellos me han enseñado que no debo rendirme y que para conseguir mis objetivos debo ser perseverante.

A todos ellos mi amor y agradecimiento eterno.

Magaly



AGRADECIMIENTO

Mis más sinceros agradecimientos a la Universidad Técnica de Cotopaxi por permitirme ser parte de esta ilustre Alma Mater y permitirme formarme en cada uno de los ciclos académicos hasta consagrarme como profesional.

A los docentes en general que fueron parte de mi proceso de formación desde los inicios de la carrera hasta la culminación de la misma, que con su infinita sabiduría supieron impartir parte de su conocimiento y moldear cada aprendizaje para poder defenderme en el ámbito profesional.

Darwin



DEDICATORIA

El presente trabajo de investigación lo dedico:

A mis abuelos Ángel y Leonor por ser quienes me brindaron todo su apoyo en todo sentido, desde los inicios de mis estudios Universitarios hasta este punto culminante de formación, que con sus sabios consejos me guiaron para que siga avanzando y no me detenga cuando se me presenten adversidades en el transcurso de este largo camino.

A mi Dios quien fue mi guía y fuerza para no desmayar en todo este camino de formación que no ha sido para nada fácil.

Darwin



ÍNDICE DE CONTENIDO

PORTADA.....	i
DECLARACIÓN DE AUTORÍA	ii
AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN	iii
APROBACIÓN DEL TRIBUNAL DE TITULACIÓN.....	iv
AGRADECIMIENTO	v
DEDICATORIA.....	vi
ÍNDICE DE CONTENIDO	ix
ÍNDICE DE TABLAS.....	xiii
ÍNDICE DE FIGURAS	xv
RESUMEN	xvi
ABSTRACT	xvii
AVAL DE TRADUCCIÓN.....	xviii
1. INFORMACIÓN GENERAL.....	1
2. INTRODUCCIÓN	3
2.1. EL PROBLEMA.....	4
2.1.1. Situación Problemática.....	4
2.1.2. Formulación del Problema.....	6
2.2. OBJETO Y CAMPO DE ACCIÓN.....	6
2.3. BENEFICIARIOS.....	6
2.3.1. Beneficiarios directos	6
2.3.2. Beneficiarios indirectos	7
2.4. JUSTIFICACIÓN	7
2.5. HIPÓTESIS.....	9
2.6. OBJETIVOS	9
2.6.1. General.....	9
2.6.2. Específicos.....	9



2.7. SISTEMA DE TAREAS.....	9
3. MATERIALES Y MÉTODOS	10
3.1. TIPO DE INVESTIGACIÓN	10
3.1.1. Investigación Mixta	10
3.1.2. Investigación Cualitativa	10
3.1.3. Investigación Cuantitativa	11
3.2. MÉTODOS DE INVESTIGACIÓN.....	11
3.2.1. Investigación Bibliográfica.....	11
3.2.2. Metodología de Desarrollo de Software.....	12
3.2.3. Metodología de Inteligencia Artificial	12
CAPÍTULO I.....	14
1. FUNDAMENTACIÓN TEÓRICA.....	14
1.1. ANTECEDENTES	14
1.2. INGENIERÍA DE SOFTWARE.....	16
1.2.1. Definición	16
1.3. CICLO DE VIDA DEL SOFTWARE.....	17
1.3.1. Fases de Desarrollo de Software	18
1.4. METODOLOGÍAS Y MODELOS UTILIZADOS EN EL DESARROLLO DE SOFTWARE.....	19
1.4.1. Modelos Tradicionales de Desarrollo de Software.....	19
1.4.2. Metodologías Tradicionales de Desarrollo de Software.....	24
1.4.3. Metodologías Ágiles de Desarrollo de Software.....	25
1.5. LENGUAJES DE PROGRAMACIÓN	28
1.5.1. Python.....	28
1.5.2. PostgreSQL.....	28
1.5.3. Framework Django	29



1.6. INTELIGENCIA ARTIFICIAL	29
1.6.1. Aprendizaje Automático.....	30
1.6.2. Minería de Datos	40
CAPÍTULO II.....	45
2. INTEGRACIÓN DE LA INTELIGENCIA ARTIFICIAL EN EL DESARROLLO SOFTWARE: CASO DE ESTUDIO ECUCIENCIA.....	45
2.1. ANTECEDENTES	45
2.1.1. Metodologías y Modelos de Ingeniería de Software aplicadas en el Proyecto EcuCiencia.	49
2.1.2. Aplicación de la Inteligencia Artificial en el Desarrollo del Proyecto EcuCiencia.	51
2.1.3. Resultados de la Integración de la Ingeniería de Software y la Inteligencia Artificial en el Proyecto EcuCiencia.....	53
CAPÍTULO III	59
3. MODELO PARA DETERMINACIÓN DE FACTORES DE ÉXITO EN LA INTEGRACIÓN DE LA INTELIGENCIA ARTIFICIAL CON EL DESARROLLO DE SOFTWARE.....	59
3.1. ANTECEDENTES	59
3.1.1. Antecedentes sobre Fases de Ingeniería de Software con Inteligencia Artificial 59	
3.1.2. Antecedentes del uso de Metodologías Ágiles (Scrum, Xp Y Kanban).....	64
3.2. MODELO CONCEPTUAL DE INTEGRACIÓN DE INGENIERÍA DE SOFTWARE CON INTELIGENCIA ARTIFICIAL.	70
3.3. METODOLOGÍA	75
3.3.1. Muestreo	75
3.3.2. Fiabilidad de los datos	76
3.3.3. Análisis e interpretación de la data.....	76
3.3.4. Juicio de expertos	76



4. CONCLUSIONES Y RECOMENDACIONES.....	82
5. BIBLIOGRAFÍA.....	83
6. ANEXOS.....	104



ÍNDICE DE TABLAS

Tabla 1. Sistema de tareas en relación a los objetivos propuestos.	9
Tabla 2. Oportunidades para la Transformación digital de la cadena de suministro con IA ...	14
Tabla 4. Fases de Desarrollo de Software.	18
Tabla 5. Fases de Desarrollo de Software según el punto de vista de diferentes autores.	18
Tabla 6. Principales modelos tradicionales entre 1979 y 1 990.	20
Tabla 7. Explicación de las fases de desarrollo del modelo en cascada.	21
Tabla 8. Ventajas y desventajas del modelo en Cascada.	21
Tabla 9. Principales metodologías tradicionales entre 1979 y 1990.	24
Tabla 10. Etapas de la Metodología RAD.	25
Tabla 11. Metodologías Ágiles.	26
Tabla 12. Roles de Scrum.	27
Tabla 13. Artefactos de Scrum.	27
14. Algoritmos del Aprendizaje Supervisado.	35
Tabla 15. Algoritmos del Aprendizaje no supervisado.	38
Tabla 17. Principales Algoritmos de Aprendizaje por Refuerzo.	39
Tabla 18. Técnicas de minería de datos.	41
Tabla 19. Detalle de Tesis de la Plataforma EcuCiencia.	47
Tabla 21. Metodologías de Desarrollo de Software aplicadas en EcuCiencia.	49
Tabla 22. Modelo de Ingeniería de Software aplicadas en EcuCiencia.	49
Tabla 23. Etapas del Desarrollo de Software aplicadas en EcuCiencia.	50
Tabla 24. Actividades de las Fases de Ingeniería de Software aplicadas en EcuCiencia.	50
Tabla 26. Metodología de Minería de Datos aplicada en EcuCiencia.	51
Tabla 27. Metodologías de Inteligencia Artificial aplicada en EcuCiencia.	52
Tabla 28. Algoritmos de Inteligencia Artificial aplicadas en EcuCiencia.	52
Tabla 29. Métricas para evaluar algoritmos de Inteligencia Artificial aplicadas en EcuCiencia.	53
Tabla 30. Proyectos de desarrollo de software con implementación de inteligencia artificial.	62
Tabla 32. Metodología Ágil Scrum.	64
Tabla 34. Metodología Ágil Extreme Programming (XP).	66
Tabla 37. Metodología Ágil KANBAN.	69



Tabla 38. Codificación de las fases de desarrollo de Software de la Metodología XP (Extreme Programming).....	70
Tabla 39. Codificación de las fases de desarrollo de Software de la Metodología KANBAN.	71
Tabla 40. Fases de desarrollo de Software de la Metodología XP (Extreme Programming)..	71
Tabla 41. Fases de desarrollo de Software de la Metodología KANBAN.	71
Tabla 42. Determinación de las fases para el diseño del Modelo de Integración.....	72
Tabla 43. Fases del Modelo de Integración de Inteligencia Artificial con el Desarrollo de Software.....	74
Tabla 44. Estadística de Fiabilidad del instrumento.....	76
Tabla 45. Análisis de datos por Género.....	76
Tabla 46. Análisis de datos por Años de experiencia.....	76
Tabla 47. Análisis de datos por Área de especialización o profesionalización.....	77
Tabla 48. Análisis estadísticos de la encuesta.....	78



ÍNDICE DE FIGURAS

Figura 1. Comparación de ejecución con distintos algoritmos (J30, J60 y J120).	15
Figura 2. Comparación de resultados de actividades y tareas de un proyecto.	16
Figura 3. Ciclo de vida del software.....	17
Figura 4. Fases de desarrollo de software de acuerdo al Modelo en Cascada.....	20
Figura 5. Fases del Modelo en V.....	22
Figura 6. Fases del Modelo de Desarrollo Incremental.....	22
Figura 7. Fases del Desarrollo del Modelo de Desarrollo Evolutivo (Espiral).	23
Figura 8. Modelo de prototipo.....	23
Figura 9. Esquema de la Metodología Scrum	27
Figura 10. Aprendizaje supervisado.	31
Figura 11. Modelo de una red neuronal artificial	33
Figura 12. Árboles de decisión.....	34
Figura 13. Algoritmo de clasificación Naïve Bayes.....	35
Figura 14. Modelo de Aprendizaje no Supervisado.	36
Figura 15. Taxonomía de los algoritmos de aprendizaje por refuerzo.	40
Figura 16. Proceso KDD.	42
Figura 17. Fases de la Metodología CRISP-DM.....	43
Figura 18. Metodología SEMMA.....	44
Figura 19. Estadísticas de producción científica por año.	53
Figura 20. Clasificación de Líneas de Investigación mediante Máquinas de Soporte Vectorial.	54
Figura 21. Aplicación de algoritmos de agrupación para determinación de Similaridad.	55
Figura 22. Red de colaboración mediante medidas de similitud.....	56
Figura 23. Aplicación del módulo de Georreferenciación.	56
Figura 24. Procesamiento de datos.....	57
Figura 25. Ventana de solicitudes de recomendación.	58
Figura 26. Modelo Conceptual de integración de la Ingeniería de Software con la Inteligencia Artificial.....	73



UNIVERSIDAD TÉCNICA DE COTOPAXI
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

TÍTULO: “APLICACIÓN DE LA INTELIGENCIA ARTIFICIAL EN EL DESARROLLO DE SOFTWARE”

AUTORES:

Sarco Calapiña Wendy Magaly

Toaquiza Pacheco Darwin Vladimir

RESUMEN

El desarrollo de software (DS) es un área de las ciencias informáticas encargada del procesamiento de la información a través de la creación, diseño y despliegue de productos de software que requieren de un ciclo de vida y de la implementación de un modelo para determinar un producto de calidad. Los procesos de desarrollo de software generan también una alta tasa de fallo debido a la especificación de requerimientos, comunicación limitada entre el equipo de desarrollo y el cliente, una aplicación inadecuada de las metodologías y estimaciones de tiempo que deben ser analizadas a profundidad. Con los cambios tecnológicos establecidos en esta última década y el avance de la Inteligencia Artificial (IA) se considera importante la integración de estas dos aristas de la ciencia que permitan el desarrollo de un producto de software capaz de satisfacer las necesidades de los clientes con base a la integración de tecnologías de última generación. Para el desarrollo del proyecto de investigación se establece una revisión sistemática de literatura basado en bases de datos científicas de información, en sus segunda fase se establece un caso de estudio de la plataforma científica EcuCiencia para determinar modelos, fases y actividades establecidas en el desarrollo de software y la aplicación de algoritmos de IA y finalmente se diseña un modelo para determinar factores de éxito que permitan la integración de la Inteligencia Artificial y la Ingeniería de Software para desarrollar soluciones completas de programación a través de la aplicación de modelos y algoritmos de última generación.

Palabras clave:

Inteligencia Artificial, desarrollo de software, aprendizaje automático, EcuCiencia.



COTOPAXI TECHNICAL UNIVERSITY
ENGINEERING SCIENCES AND APPLIED FACULTY

TOPIC: “ARTIFICIAL INTELLIGENCE APPLICATION IN THE SOFTWARE DEVELOPMENT”

AUTHORS:

Sarco Calapiña Wendy Magaly

Toaquiza Pacheco Darwin Vladimir

ABSTRACT

Software development (SD) is a computer science area managed information processing, through the software products creation, design and deployment, which require a life cycle and the model implementation to determine a quality product. Also, software development processes generate a high failure rate, due to requirement specification, limited communication between the development team and the client, an inadequate application of methodologies and estimated time, what must be analyzed depth. With technological changes established this last decade and the Artificial Intelligence (AI) advance, it is considered important to integrate these science two edges, what allow the a capable software product development of satisfying the customers needs, based onto art state technologies integration. For research project development, it was set a systematic literature review, based onto scientific information databases, its second phase is established a Ecuciencia scientific platform case study to determine models, phases and set activities into AI algorithms software and the application development, and finally, it is designed a model to determine success factors, what allow the Artificial Intelligence and Software Engineering integration for developing complete programming solutions, through the next-generation models and algorithms application.

Keywords:

Artificial Intelligence, software development, machine learning, EcuCiencia.



AVAL DE TRADUCCIÓN

En calidad de Docente del Idioma Inglés del Centro de Idiomas de la Universidad Técnica de Cotopaxi; en forma legal **CERTIFICO** que:

La traducción del resumen al idioma Inglés del trabajo de titulación cuyo título versa: **“APLICACIÓN DE LA INTELIGENCIA ARTIFICIAL EN EL DESARROLLO DE SOFTWARE.”** presentado por: **Sarco Calapiña Wendy Magaly** y **Toaquiza Pacheco Darwin Vladimir**, estudiantes de la Carrera de **Ingeniería en Informática y Sistemas Computacionales** perteneciente a la **Facultad de Ciencias de la Ingeniería y Aplicadas** lo realizaron bajo mi supervisión y cumple con una correcta estructura gramatical del Idioma.

Es todo cuanto puedo certificar en honor a la verdad y autorizo a los peticionarios hacer uso del presente aval para los fines académicos legales.

Latacunga, 18 de marzo del 2022

Atentamente,



CENTRO
DE IDIOMAS

Mg. Marco Paúl Beltrán Semblantes

DOCENTE CENTRO DE IDIOMAS-UTC
CI: 0502666514

1. INFORMACIÓN GENERAL

Título: Aplicación de la Inteligencia Artificial en el Desarrollo de Software.

Fecha de inicio:

Octubre 2021

Fecha de finalización:

Marzo 2022

Lugar de ejecución: Cotopaxi/Latacunga/Av. Simón Rodríguez s/n Barrio El Ejido Sector San Felipe.

Facultad que auspicia: Ciencias de la Ingeniería y Aplicadas

Carrera que auspicia: Ingeniería Informática y Sistemas Computacionales

Proyecto de investigación vinculado: Modelamiento de algoritmos para sistemas de información, Grupo de Robótica e Inteligencia Artificial.

Equipo de Trabajo:

Datos Personales del Tutor:

Apellidos: Albán Taípe

Nombres: Mayra Susana

Cédula: 050231198-8

Teléfono/Celular: 0987773341

E-mail Institucional: mayra.alban@utc.edu.ec

Datos Personales del Estudiante Investigador 1

Apellidos: Sarco Calapiña

Nombres: Wendy Magaly

Cédula: 180537353-5

Teléfono/Celular: 0995453280

E-mail Institucional: wendy.sarco3535@utc.edu.ec

Datos Personales del Estudiante Investigador 2

Apellidos: Toaquiza Pacheco

Nombres: Darwin Vladimir

Cédula: 171937989-1

Teléfono/Celular: 0987593221

E-mail Institucional: darwin.toaquiza9891@utc.edu.ec

Área de Conocimiento: 06 Información y Comunicación (TIC) / 061 Información y Comunicación (TIC) / 0613 Software y desarrollo y análisis de aplicativos.

Línea de investigación: Tecnologías de la Información y Comunicación (TICS).

Sub líneas de investigación de la Carrera: Inteligencia Artificial e Inteligencia de Negocios.

2. INTRODUCCIÓN

Las problemáticas dentro del ámbito del desarrollo de software son múltiples por mencionar algunas tenemos, limitaciones en la especificación de requisitos, deficiencias en la planificación de cronogramas, metodologías que no están acorde al proyecto para su fase desarrollo entre otras. Del mismo modo, podemos señalar a la Inteligencia Artificial (IA) como un factor de importancia debido a que en este campo del conocimiento existe un extenso número de técnicas y algoritmos, que no permiten establecer, de manera óptima cuál de estos algoritmos aportará en mayor grado para complementar al desarrollo de software, debido a que una mala decisión de implementación puede ocasionar falencias en el proyecto, ocasionando pérdidas de tiempo, económicas, entre otros.

Existe un número amplio de técnicas y algoritmos de la Inteligencia Artificial que pueden ser integradas en el desarrollo de software, de las cuales se puede mencionar redes bayesianas, redes neuronales, algoritmos genéticos, entre otros. También, se puede señalar una amplia variedad de áreas donde estas técnicas de desarrollo puedan ser establecidas como por ejemplo las cadenas de suministros, dando como resultado la toma de mejores decisiones y favoreciendo a los resultados requeridos, las técnicas que aportan este gran valor son algoritmos genéticos, agentes inteligentes, redes neuronales, entre otros. Otra de las aportaciones de la IA está relacionada con la deficiencia en la construcción de cronogramas que también es una de las actividades más complejas en multiproyectos de desarrollo de software, con el objetivo de disminuir aquellas deficiencias, la IA nos muestra un apartado del cual se puede sacar provecho como el uso de algoritmos genéticos que son óptimos para este tipo de casos.

El presente trabajo de investigación tiene como objetivo realizar una revisión sistemática de literatura que permita sustentar teóricamente la pregunta de investigación y la hipótesis a fin de construir un marco teórico acorde a dos aristas importantes de investigación que son el desarrollo de software y la inteligencia artificial, en su segunda etapa del proyecto de investigación se establece un caso de estudio en la plataforma científica de EcuCiencia que permitirá la obtención del análisis de los modelos establecidos, las fases, actividades y procedimientos ejecutados en cada uno de los módulos que conforman esta plataforma así como también el análisis de los algoritmos implementados para determinar su funcionalidad y uso dentro de la plataforma. Finalmente se diseña un modelo conceptual para la determinación de factores de éxito que permitan la integración de la Inteligencia Artificial con la Ingeniería de Software. Como resultado se pretende obtener una herramienta de apoyo para la toma de

decisiones que sirva a los programadores una herramienta útil de integración óptima y el desarrollo de productos de software de alta calidad basada en tecnologías de última generación.

2.1. EL PROBLEMA

2.1.1. Situación Problemática

Según Ortiz & Iglesias [1] en su análisis sobre el trabajo de Tim Berners-Lee, expresa su preocupación sobre la creciente pérdida de control de los datos personales y el papel que los algoritmos y la inteligencia artificial están desempeñando en la sociedad, es decir, como el uso de datos personales, algoritmos e inteligencia artificial crean riesgos y oportunidades para los países de ingresos medios y bajos. Además, señala que la implementación de lo ya mencionado había despertado gran interés por parte de otros gobiernos y críticas activistas donde consideran que violan la privacidad y no atiende las causas del propio problema.

El desarrollo de software en el ámbito de las pequeñas y grandes empresas (PyMEs) nos mencionan Villavicencio, Peña, Burneo & Pérez [2] que existe una especial atracción hacia la introducción de mejores prácticas, pero su contraparte los jefes de proyectos y empresarios que están involucrados mantienen carencias de formación y conocimientos sobre los estándares o herramientas para trabajar con una buena gestión. En un diagnóstico donde los involucrados son los profesionales, jefes de proyecto y empresarios se ha identificado como las principales causas de fracasos en proyectos, incluido también limitaciones en la implementación de tecnologías adecuadas, errores en la definición del alcance del proyecto, deficiencias en la comunicación con los clientes, entre otros.

En las empresas de software según Rojas & Araujo [3] se realizan varios proyectos utilizando los mismos recursos humanos, por diferentes motivos entre los cuales se mencionan cambios tecnológicos acelerados, presión económica, trabajo en equipos multidisciplinarios, recursos y tiempo limitado. Se puede inferir que debido al problema de la planificación de cronogramas en proyectos de software se le da un enfoque principal de importancia en la actualidad.

Para Morales & Pardo [4] en la industria del desarrollo de software se sigue trabajando de forma manual o con metodologías robustas, que como consecuencia llegan a ser pesadas o ineficientes. De acuerdo con Molina, Vite & Dávila [5] manifiestan que el principal problema en la variedad de las metodologías existentes para el desarrollo de software es la limitación en la identificación de las mismas, con el fin de lograr la construcción de un producto de calidad. El autor sugiere que independiente de las distintas metodologías que existen y se pretenden

usar, se debe tener en cuenta que el producto final debe de ser un software de calidad, indiscutiblemente.

En el siguiente informe del CHAOS Report [6] nos menciona que la tasa de éxito de los proyectos se basa en el proceso ágil, los proyectos pequeños según las métricas modernas tienen una tasa de falla del 4%, del mismo modo, menciona que el uso de metodologías tradicionales genera una tasa de fracaso de casi el doble. Los proyectos que están dentro del costo tienen un porcentaje del 44% y dentro del plazo de entrega un 40%, evidenciando que más de la mitad de los proyectos sobrepasan el costo y tiempo de entrega. Muchos proyectos requieren mejoras pequeñas para los sistemas e incluso hay casos de reingeniería.

Todo proyecto debe ser llevado y dirigido mediante cronogramas, con el fin de subdividir una tarea compleja en subtareas, pero no todo se maneja de esta forma, según el Project Management Institute GLOBALSTANDARD [7] todos los proyectos que son dirigidos de forma deficiente o con la ausencia de la dirección de proyectos, conlleva a problemáticas de incumplimiento de plazos, sobrecostos, retrabajo, calidad deficiente, la expansión no controlada del proyecto y el incumplimiento de los objetivos propuestos. De modo similar, señalan la razón del problema ocasionado en la planificación de cronogramas en proyectos de software, se relaciona con la estructuración y control de proyectos individuales, como consecuencia no se refleja la realidad durante la ejecución.

Se puede señalar también lo expuesto por Varón [8] quien menciona que los errores se perciben más que todo en la fase inicial del proyecto, es decir, en la fase de análisis, los cuales están relacionados con estimación correcta de su alcance, inadecuada especificación de los requisitos o requerimientos, se ha dado casos en los que a mitad del proyecto se deben incluir nuevas condiciones lo cual genera cambios en el proceso de desarrollo, entrega del software y los costos que se habían planificado.

En la fase de pruebas, si no se emplea el tiempo necesario para realizar las pruebas de software puede generar que la calidad del mismo no sea segura. De modo similar Mascheroni & Irrazábal [9] indican que, si se quiere presentar un software adecuado, es indispensable que se realicen pruebas que ayuden a definir su cumplimiento con las expectativas del cliente, el problema se presentaría si se consume mucho tiempo. El tiempo que se emplee en la fase de pruebas debe ser un tiempo óptimo o equilibrado donde la cantidad de tiempo invertida sea la adecuada, no menor al tiempo estimado de pruebas del proyecto, ni mayor a ello, que como consecuencia sobrepase los límites de la planificación.

Según Madruga & Castillo [10] señalan que gran porcentaje de los proyectos de software tienen una tasa alta de fracasos hoy en día, esto puede deberse a la dificultad que tienen varios programadores al momento de levantar los requerimientos en la etapa de diseño del software. Sin duda alguna, cuando se desarrolla un proyecto de software es posible que se deje a un lado las buenas prácticas, incurriendo nuevamente en procesos que están fuera de los estándares establecidos en la ingeniería de software como, por ejemplo, tener una visión propia del producto sin considerar los requerimientos del cliente, creer a manera personal como se plasman los procesos de negocio del software, entre otros.

La utilización de Inteligencia Artificial con la comprensión del lenguaje natural en Ecuador ha conllevado en la atención al cliente, según Tapia [11] nos da a conocer como la IA están incluidas en chatbots para la atención de clientes, con agentes virtuales que se llevan a cabo en Bancos Ecuatorianos como: Banco de Guayaquil con Ada y Banco del Pacifico con Sophi. Pero su contraparte el mismo autor explica las deficiencias en las expectativas o necesidades del cliente, es decir, si la pregunta no logra ser respondida o ayuda a solucionar cierta inquietud, hace que la aplicación sea ineficiente ocasionando que el usuario no vuelva a usar dicha herramienta tecnológica. Su desarrollo debe basarse en características sumamente importantes ya que la aplicación debería entender las palabras que están siendo ingresadas por parte del usuario para ello se debe aplicar la metodología, el modelo o la técnica correcta caso contrario no se podrá obtener una respuesta por parte de la aplicación.

2.1.2. Formulación del Problema

¿Se puede integrar Inteligencia Artificial con el Desarrollo de Software?

2.2. OBJETO Y CAMPO DE ACCIÓN

1203.04 Inteligencia Artificial – Ingeniería de Software

2.3. BENEFICIARIOS

2.3.1. Beneficiarios directos

- Industrias de Desarrollo de Software del Ecuador.
- Desarrolladores de software.
- Programadores.
- Programadores de IA.
- Estudiantes y docentes de la carrera de sistemas de información.

2.3.2. Beneficiarios indirectos

- Usuarios finales del software desarrollado.

2.4. JUSTIFICACIÓN

La importancia de la Ingeniería de Software para Ruiz & González [12] es relevante hoy en día, debido al incremento exponencial de la industria del software en varios países a nivel mundial, dado que en el ámbito de la ingeniería de software es un campo donde se generan metodologías para el desarrollo del ciclo de vida del software, con un mismo propósito de cumplir cada fase o etapa por medio de entregables o documentación que sustenten el correcto funcionamiento del proyecto. De modo similar Chanatasig [13] nos menciona que la ingeniería de software se le considera como ingeniería aplicada al software, aportando relevancia en empresas internacionales y la alta demanda de aprendizaje hacia estudiantes que se forman para dar múltiples soluciones a distintos problemas que la ingeniería de software acarrea consigo.

Los autores González, Calero & Loaiza [14] manifiestan que el desarrollo de software es una labor creativa que depende mucho de las personas que estén dentro del equipo de desarrollo, de las habilidades y los conocimientos que cada uno de ellos posea e incluso de las metodologías que se utilicen, en otras palabras, el equipo de desarrollo debe estar compuesto por personas que tenga conocimientos en las distintas áreas y que exista una buena comunicación para que se pueda desarrollar el software de la mejor manera.

La importancia de la aplicación de IA como nos menciona Rouhiainen [15] abarca tecnologías que están siendo usadas para aportar ayuda hacia los humanos en casi todos los ámbitos de la vida cotidiana, donde su impacto es aplicable en distintas áreas, específicamente en el desarrollo de software. El crecimiento en los procesamientos que realizan los ordenadores, mejorando los procesos de distintos algoritmos complejos, en esta investigación se darán a conocer algunos de ellos. De modo similar Corvalan [16] en su investigación explica que tanto para el uso y mejoramiento de las tecnologías, las estimaciones sobre el futuro hasta el año 2022 la demanda de Inteligencia Artificial (IA) abarca roles en disciplinas como: analista de datos, desarrolladores de software, expertos en automatización de procesos e ingenieros en aprendizaje automático.

Los desarrolladores de software deben comprender el impacto de estas tecnologías, es decir, la IA en el software y como poder aplicarlas, cabe destacar lo mencionado por Galván [17] donde explica que se pueden aplicar tanto en el ciclo de vida de desarrollo como en las aplicaciones

de las mismas. Dentro de las posibilidades el mismo autor nos da a conocer más aplicaciones y beneficios para el proceso del software como enriquecer los modelos de los requerimientos, así como también los casos de prueba. Al mismo tiempo se habla de un mejoramiento a la precisión de estimaciones en los proyectos de software, tanto en tiempo y esfuerzos que esto implica. Como último punto se habla de automatizar decisiones a construir y probar, en otras palabras, la IA aportaría un análisis de patrones de una aplicación en producción y con ello poder decidir los requerimientos según su grado de prioridad.

Se puede señalar que las técnicas que se establecen en la IA son de tal importancia en el desarrollo de software, como manifiesta Quesada[18] la aplicación de IA en cuanto a técnicas se refiere en el proceso de desarrollo de software no se fija a un único tipo de técnica. De la misma forma Ferreira, Gálvez, Quintero & Antón [19] manifiestan que resulta de interés la aplicación de técnicas de IA en problemas que puedan acarrear la Ingeniería de Software, dichos modelos donde se incluyen las diferentes técnicas de IA que existen, como el aprendizaje automático que han servido para mejorar la precisión de las estimaciones.

La Inteligencia Artificial y la Ingeniería de Software brindan grandes beneficios por un lado la Ingeniería de Software permite analizar, diseñar y crear aplicaciones web, escritorio y móviles haciendo uso de metodologías, herramientas y técnicas que permitan su organización y desarrollo. Por su parte, la Inteligencia Artificial permite la resolución de problemas logrando minimizar los esfuerzos humanos. Es decir, lograr que un software pueda procesar datos y mostrar el resultado de lo que se produce mediante el aprendizaje continuo, de acuerdo con las investigaciones realizadas podemos expresar sus ventajas las cuales son la automatización de procesos, minimización de esfuerzos humanos y ayuda a la toma de decisiones. Por lo que, una combinación entre estos dos campos de la ciencia genera mayores posibilidades de brindar un software de calidad que cumpla con las necesidades de los clientes, que ayuden en la toma de decisiones y también a realizar proyecciones a futuro.

Esta investigación pretende realizar un modelo conceptual para determinar factores de éxito para la integración de la Inteligencia Artificial con la Ingeniería de Software basados en una revisión sistemática de literatura, el análisis del caso de estudio de la plataforma científica EcuCiencia y el análisis de los modelos encontrados en la literatura relacionadas con prácticas ágiles de desarrollo.

2.5. HIPÓTESIS

El diseño de un modelo conceptual para determinar factores de éxito en la integración Inteligencia Artificial y desarrollo de Software podría servir como una herramienta de ayuda para la construcción de productos de software basados en la implementación de algoritmos de inteligencia artificial de última generación.

2.6. OBJETIVOS

2.6.1. General

- Desarrollar un modelo de factores de éxito para la integración de la Inteligencia Artificial con el Desarrollo de Software, que permita el desarrollo de productos de alta calidad mediante la implementación de tecnologías de última generación.

2.6.2. Específicos

- Sustentar teóricamente el problema de investigación a través del uso de bases de datos científicas para la identificación de documentos primarios.
- Analizar la información: Caso de uso Proyecto Red de Estudios Cienciométricos o plataforma EcuCiencia para la determinación de factores que permitan el diseño del modelo de integración de inteligencia artificial con el desarrollo de software.
- Diseñar el modelo teórico de integración de inteligencia artificial con la Ingeniería de Software a través del uso de modelos encontrados en la literatura.

2.7. SISTEMA DE TAREAS

Tabla 1. Sistema de tareas en relación a los objetivos propuestos.

OBJETIVO ESPECÍFICO	ACTIVIDAD	RESULTADO	TÉCNICAS E INSTRUMENTOS
Sustentar teóricamente el problema de investigación a través del uso de bases de datos científicas para la identificación de documentos primarios.	Planificación de revisión de literatura	Marco Teórico	Fichas Bibliográficas
	Determinación de documento primarios de investigación		
	Análisis del contenido de los documentos primarios		
Analizar la información: Caso de uso Proyecto Red de Estudios Cienciométricos o plataforma EcuCiencia para la determinación de factores que permitan el	Determinación de Modelos y Metodologías de desarrollo de software	Clasificación de modelos y metodologías de software	Modelos de Desarrollo de software
	Determinación de metodologías de IA	Clasificación de metodologías de IA	Metodologías de IA

diseño del modelo de integración de inteligencia artificial con el desarrollo de software.	Determinación de algoritmos de IA	Clasificación de algoritmos de IA Aplicados en EcuCiencia	Algoritmos de IA
Diseñar el modelo teórico de integración de inteligencia artificial con la Ingeniería de Software a través del uso de modelos encontrados en la literatura.	Determinación de factores de éxito para la integración de la Inteligencia Artificial y el Desarrollo de Software	Modelo de integración de la IA con la IS	Validación de expertos
	Diseño del modelo conceptual		Validación por procedimiento experimental
	Validación del modelo conceptual		

3. MATERIALES Y MÉTODOS

3.1. TIPO DE INVESTIGACIÓN

3.1.1. Investigación Mixta

Según Ugalde & Balbastre [20] esta investigación es un conjunto entre investigación cualitativa e investigación cuantitativa, la correlación que tienen entre ambas permite que los hallazgos sean más completos, generando más confianza, mayor validación y mejor entendimiento de los resultados. Las ventajas de esta metodología es la facilidad para poder generar y corroborar teorías en un mismo estudio, obtención de inferencias más fuertes y compensación tanto de las desventajas que existe entre cualitativas y cuantitativas cuando se hace uso de manera individual.

Para la realización del presente proyecto investigativo se pretende recurrir a distintos tipos de investigaciones para completar su desarrollo. Se ha realizado un análisis para poder determinar qué tipo de investigación es el más óptimo para el proyecto a realizar, donde a continuación se describen cada una de ellas.

3.1.2. Investigación Cualitativa

También conocida como Metodología Cualitativa, según Guerrero [21] es uno de los tipos de investigación más usados y tiene como objetivo interpretar y profundizar los fenómenos que son analizados desde el punto de vista de los participantes tanto en su ambiente y con lo que le rodea. Este tipo de investigación es útil cuando se busca comprender cierta perspectiva de individuos o grupos de personas que van a ser investigadas. Dentro de sus fases tenemos la definición del problema, diseño de trabajo, recogida de datos, análisis de datos, información y

validación de la información. Las técnicas de la investigación cualitativa son la observación, observación participante, entrevista, entrevista grupal, cuestionario y grupo de discusión.

El presente trabajo de investigación se realizó mediante una investigación bibliográfica bajo el método cualitativo que comprende la recopilación, síntesis y análisis de información tomada de bases de datos de alto factor de impacto obteniendo así múltiples documentos que sirvan de guía para el proceso de desarrollo de esta investigación.

3.1.3. Investigación Cuantitativa

Este tipo de investigación para Rus [22], hace hincapié al análisis de un volumen bastante amplio de datos, realiza estudios sobre las variables cuantitativas, en otras palabras, numéricas a diferencia de las cualitativas, por lo tanto, pueden ser medibles. La investigación cuantitativa puede generar conclusiones extrapolables a un grupo bastante amplio que el investigado. Dentro de las características principales de este tipo de investigación están que el proceso de adquisición de datos es rápido, predicción de ciertos comportamientos de una población en específica estudiada, los resultados que son obtenidos ayudan a la comprensión de la situación de la población, entre otros.

Para el desarrollo de este trabajo de investigación se pretende realizar la recolección de datos mediante una encuesta para lo cual se debe analizar las variables para definir el comportamiento de la población y como resultado se muestra datos numéricos con lo cual se puede tomar decisiones o realizar comparaciones para un determinado caso.

3.2. MÉTODOS DE INVESTIGACIÓN

3.2.1. Investigación Bibliográfica

También conocida como investigación académica, Álvarez [23] describe de forma cuidadosa y detallada los antecedentes, en otras palabras, es la recopilación de resultados de otras investigaciones que tienen relación con un tema específico a ser investigado; determina lo que recientemente se haya hecho sobre algún tema en específico, con sustentación y fundamentación mediante las citas bibliográficas pertinentes. Para el cumplimiento de la investigación se abordan pasos o fases como observación, indagación, interpretación, reflexión, y análisis para obtener bases necesarias para el desarrollo de un estudio en específico.

Al ser un proyecto de investigación es netamente necesario la investigación de múltiples documentos relacionados con el tema propuesto y sus derivados. Los documentos deben ser

fiables para la investigación, por tal motivo, existe aceptación como fuentes de consultas en bases de datos con alto factor de impacto (Scopus, Dialnet, ScienceDirect, entre otros).

3.2.2. Metodología de Desarrollo de Software

Las metodologías de desarrollo de software (MDS), según Rivas, Corona, Gutiérrez & Hernández [24], nos menciona que forma parte primordial de la ingeniería de software (IS), donde las metodologías son un punto importante para la creación y actualización del software de calidad, cumpliendo con los requisitos que el usuario haya establecido. El usar una metodología inapropiada conlleva a que los objetivos no logren ser alcanzados por los inconvenientes que va a ir surgiendo. La variedad de metodologías y el desconocimiento de ellas es un factor negativo para el desarrollo, y en el peor de los casos no se emplea ninguna metodología.

Como ya se ha mencionado anteriormente de forma más detallada, las metodologías ágiles son factibles ante los cambios que se puedan producir durante el proyecto, el cliente forma parte del equipo, es esencial para grupos pequeños, también se pueden adaptar a las necesidades de los usuarios de esta forma se tiene flexibilidad para realizar cambios sin la necesidad de alterar el proyecto, permite el desarrollo rápido, reduce los costos, entre otros. Es así como punto principal se hizo énfasis e interés por la aplicación de estas metodologías porque cumple con la mayoría de expectativas que se pretende implementar en el modelo de desarrollo.

Dentro de las metodologías ágiles se encuentran SCRUM, XP, Kanban, entre otros; en base al estudio bibliográfico realizado, se optó por trabajar con XP y Kanban porque tienen relación entre sus fases de tal modo que se buscó la manera de relacionar esas fases y determinar las fases óptimas para el desarrollo del modelo. El cual será validado por expertos en los campos de inteligencia artificial, sistemas e informática y desarrollo de software para lo cual se ha generado una encuesta y la misma fue aplicada a docentes de la Universidad Técnica de Cotopaxi, Facultad de Ciencias de la Ingeniería y Aplicadas, carrera de Ingeniería en Informática y Sistemas Computacionales.

3.2.3. Metodología de Inteligencia Artificial

Para Alfaro [25] las metodologías de la inteligencia artificial vienen siendo alternativas objetivas para solventar de forma efectiva múltiples problemas dependiendo el caso a solventar, estas metodologías son usadas en grandes empresas y corporaciones con el objetivo de gestionar sus datos y ser eficientes en cada una de sus actividades. Las metodologías hacen uso de

múltiples técnicas para solventar un problema como: sistemas expertos, redes neuronales artificiales, algoritmos genéticos, entre otros.

Para el desarrollo del modelo de integración se ha determinado el uso de minería de datos como parte de la Inteligencia Artificial que cuenta con las siguientes fases: selección de variables, preprocesamiento de la data, modelado de algoritmos y la validación de los algoritmos. Por medio de estas fases se puede hacer un análisis y posteriormente hacer uso de los mismos para darle más relevancia a la funcionalidad que se está desarrollando.

CAPÍTULO I

1. FUNDAMENTACIÓN TEÓRICA

1.1. ANTECEDENTES

Las aportaciones de la Inteligencia Artificial en el desarrollo de software, para asegurar la calidad del software en una revisión sistemática, manifiestan que se ha podido corroborar que en los proyectos donde se aplicaron metodologías de desarrollo de software como son en cascada, ágil, simbolismo, prototipos y en espiral, da paso y acogida a la Inteligencia Artificial, dando buenos resultados, donde sus criterios de calidad está la funcionalidad con un 34%, usabilidad con un 32%, procesamiento con un 27% y sin definir con un 7%. Como consecuencia los algoritmos de IA que se han aplicado en el aseguramiento de la calidad del software son red bayesiana, redes neuronales, algoritmos genéticos, regresión logística, random forest y rough set [26].

El uso de la IA en el desarrollo de software ha generado grandes beneficios en distintos campos, el trabajo desarrollado por Arango [27] se menciona que el uso de técnicas de IA generó la optimización en la gestión de la cadena de suministros. Para ello, se utilizan algunas de las técnicas de IA como se mencionan a continuación y como consecuencia la IA permite que se tomen mejores decisiones y favorecen en los resultados de la empresa.

Tabla 2. Oportunidades para la Transformación digital de la cadena de suministro con IA

PROCESO	TÉCNICAS DE IA	OPORTUNIDAD
Gestión de Almacenamiento	Algoritmos Genéticos	Optimización de la capacidad de búsqueda en el almacenamiento
		Optimización en la selección y la asignación de los operadores de despacho en ambientes de incertidumbre
	Agentes inteligentes	Reducción de costos
		Mejoramiento del promedio de nivel de servicio
Gestión de inventarios	Algoritmos Genéticos	Control para disponibilidad
		Ubicación eficiente
		Modelado en la cadena de suministro
	Redes Neuronales	Integración de inventarios
		Rastreo de la variabilidad del proceso
Demanda /pronósticos	Agentes inteligentes	Predicción de la demanda.
		Predicción de la demanda para reducir el efecto látigo
Gestión de Distribución	Algoritmos Genéticos	Mejorar la estabilidad de distribución mediante modelo de alerta temprana de riesgo
	Agentes Inteligentes	Optimización de costos mediante la colaboración, cooperación y negociación de agentes
		Minimizar los tiempos de entrega y costos de producción.
	Lógica difusa	Mejora el nivel de servicio del cliente mediante modelo basado en lógica difusa para el razonamiento

Fuente: Tomado de: Arango [27]

Tabla 3. Oportunidades para la Transformación digital de la cadena de suministro con IA.
(Continuación)

PROCESO	TÉCNICAS DE IA	OPORTUNIDAD
Gestión de la cadena de Suministro	Aprendizaje de máquina	Mejora proceso de toma de decisiones
		Análisis de datos basados en el contexto
		Predicción de la demanda a partir de datos de clasificación y agrupamiento
	Redes Neuronales	Análisis de factores de riesgo
	Agentes inteligentes	Integración de las cadenas de suministro.
		Uso de vocabularios compartidos
		Minimización de los tiempos de entrega y costos de producción.
		Programación de producción.
Seguimiento y suministro de información		
	Negociación colaborativa	
Gestión de Proveedores	Lógica Difusa	Mejoramiento del uso de recursos dentro de la cadena de suministro.
		Diseño de redes en el contexto de ambientes complejos y con incertidumbre.
		Mejoramiento en la negociación y la colaboración
	Redes Neuronales	Reducción de la subjetividad en el proceso de selección de proveedores
		Mejora en la eficiencia del trabajo
	Agentes Inteligentes	Mejora en el desempeño de proveedores para el retail
Algoritmos Genéticos	Optimización de los costos en la selección y evaluación de proveedores	

Fuente: Tomado de: Arango [27]

En la investigación realizada por García, Pérez, Villavicencio, Piñero & Beovides [28] nos manifiesta la problemática sobre la planificación de multi proyectos de desarrollo de software, que pretende ser mermada con la implementación de Inteligencia Artificial, especialmente con el uso de algoritmos genéticos; estos algoritmos son altamente empleados en soluciones a problemas de planificación de proyectos especialmente RCPSP (Resource-Constrained Multi-Project Scheduling Problem), por su traducción al español (Problema de programación de múltiples proyectos con recursos limitados). Para la validación de la investigación se hizo uso de bases de datos, como PSPLib, donde su contenido está asociado a RCPSP. La comparación de ejecución de resultados se ejecutan distintas corridas de algoritmos. La experimentación con PSPLib se hizo uso de pruebas J30 (30 tareas y 41 recursos), J60 (60 tareas y 49 recursos), J120 (120 tareas y 48 recursos), como resultado la instancia de J30 1-1 la convergencia del algoritmo es de 45 unidades de tiempo, con un valor cercano a 43 propuesto por PSPLib como mejor solución al caso. Para J60 1-1 la convergencia del algoritmo es de 77 unidades de tiempo, que como mejor solución inicial es de 80 o más individuos a partir de 25 generaciones. Para J120 1-1 la convergencia del algoritmo es de 109 unidades de tiempo, que es cercano a 105 propuesto por PSPLib, como mejor solución.

Figura 1. Comparación de ejecución con distintos algoritmos (J30, J60 y J120).

Generaciones/tamaño población	Pruebas J30					Pruebas j60					Pruebas J120						
	20	40	60	80	100	20	40	60	80	100	20	40	60	80	100	120	140
25	55	50	53	52	46	83	83	94	92	78	146	141	144	143	136	134	126
50	55	47	51	49	45	90	83	82	83	77	143	138	143	141	139	136	118
75	54	50	47	53	47	89	86	83	80	78	149	140	145	138	131	126	109
100	53	53	45	49	45	89	87	90	78	77	140	144	141	137	134	115	114

Fuente: Tomado de García, Pérez, Villavicencio, Piñero & Beovides [28]

La siguiente base de datos es GESPRO_Scheduling como parte de la investigación y resultado donde se realizan pruebas considerando 5 proyectos aleatorios con 502 actividades y 100 tareas promedio cada uno, con 21 recursos agrupados en 4 tipos principales que están orientadas al desarrollo de software. Como resultado se emplearon 8 corridas de 200 generaciones y poblaciones con 80 individuos, obteniendo 3 mejores resultados en cada corrida. La comparación se realiza en tiempo de duración en días y el máximo de días que dura la ejecución de los 5 proyectos.

Figura 2. Comparación de resultados de actividades y tareas de un proyecto.

Algoritmos y variables	Min	Median	Mean	Max
X1Hor.Res.MediaT	157.0	161.0	161.6	197.0
X1Hor.ResMaxT	189.0	199.5	200.2	215.0
X2Hor.Hol.MediaT	157.0	159.0	160.6	196.0
X2Hor.HolMaxT	190.0	195.5	196.0	204.0
X3Ver.Res.MediaT	144.0	149.0	150.6	189.0
X3Ver.ResMaxT	175.0	187.0	187.2	201.0
X4Ver.Hol.MediaT	134.0	136.0	137.0	154.0
X4Ver.HolMaxT	147.0	155.0	155.6	167.0
X5Heu.Res.MediaT	135.0	137.0	138.2	159.0
X5Heu.ResMaxT	155.0	161.0	160.8	169.0
X6Heu.Hol.MediaT	104.0	107.0	107.3	121.0
X6Heu.HolMaxT	115.0	124.0	124.0	133.0

Fuente: Tomado de García, Pérez, Villavicencio, Piñero & Beovides [28]

Los grupos de algoritmos ordenados por su calidad de resultados: Grupo1 = {X6Heu.Hol}, Grupo2 = {X4Ver.Hol}, Grupo3 = {X5Heu.Res}, Grupo4 = {X3Ver.Res} y Grupo5 = {X2Hor.Hol, X1Hor.Res}, dando como mejor resultado X6Heu.Hol.

1.2. INGENIERÍA DE SOFTWARE

1.2.1. Definición

Los autores Ramos, Noriega, Laínez & Durango [29] la ingeniería de software es la disciplina que se encarga del desarrollo de software lo cual incluye la ingeniería de requisitos, modelos de procesos y técnicas de estimación. Permite ampliar la posibilidad de que los objetivos de negocio se cumplan de acuerdo al tiempo, calidad y funcionalidad. El desarrollo de software se da a través de un proceso, no se crea a partir de materia prima o piezas pequeñas, además comprende las etapas iniciales desde la especificación hasta el mantenimiento.

De acuerdo a Pantaleo & Rinaudo [30] se le denomina ingeniería de software, este término se usó por primera vez del 7 al 11 de octubre de 1968, cuando se realizó una conferencia en el Comité de la Organización del Tratado del Atlántico Norte (OTAN). Hacía referencia a englobar disciplinas que se relacionan con elaboración de productos software, basándose en la arquitectura y estructura de las aplicaciones. “En el año de 1990 IEEE publicó el estándar que

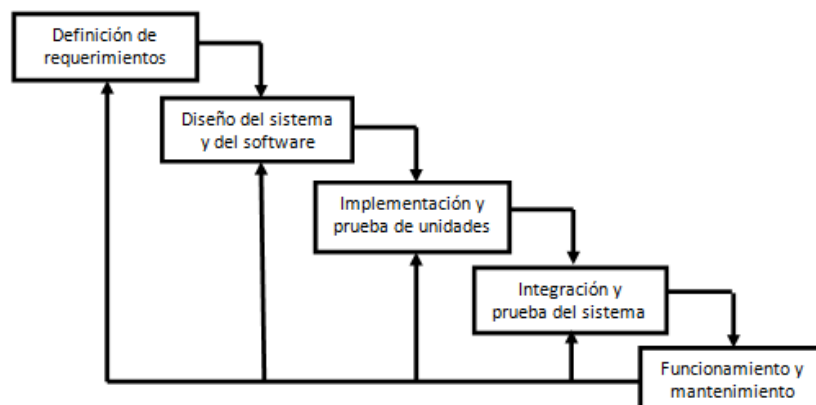
establece el glosario de términos para Ingeniería de Software en donde se define como **la aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento del software.**”

Por su parte Aguilar & Díaz [31] mencionan que la ingeniería de software como disciplina profesional no tiene muchas décadas, y el término Ingeniería de Software se propuso hace menos de medio siglo por Friedrich L. Bauner en una reunión del Comité de la Organización del Tratado del Atlántico Norte (OTAN). La ingeniería de software produce un artefacto intangible, es decir que no se puede tocar, pero es de gran valor.

1.3. CICLO DE VIDA DEL SOFTWARE

El ciclo de vida de un software consta de una etapa denominada planificación, en donde se analizan los requerimientos del cliente, la etapa de diseño en donde se realiza la estructura del sistema y las funciones que se debe cumplir, en la etapa de implementación esta estructura se transforma a código, como etapa siguiente están las pruebas que se realizan para determinar los posibles errores, por último, se realiza el mantenimiento cuando ya se ha instalado el software [32].

Figura 3. Ciclo de vida del software.



Fuente: Internet.

Como menciona Gómez & Moraleda [33] el ciclo de vida del software está compuesto por el proceso de desarrollo y mantenimiento necesario. Cada fase contiene tareas que deben cumplirse lo cual al final genera documentos con la descripción del trabajo realizado, se puede trabajar y completar las actividades de cada fase por grupos de trabajo independientes que lo realizan de forma secuencial o simultánea. El producto de un grupo de trabajo servirá para el siguiente grupo, lo importante de generar documentos es que los mismos servirán para que las partes interesadas puedan validar sus compromisos.

1.3.1. Fases de Desarrollo de Software

Tabla 4. Fases de Desarrollo de Software.

FASE	DESCRIPCIÓN
Análisis	Se analizan las necesidades de los usuarios del sistema. El cliente plantea sus necesidades, requisitos que el sistema debe cumplir. De acuerdo con la información la organización prepara una especificación precisa del sistema que se va a desarrollar.
Diseño	Se encarga del diseño donde se toma en cuenta los recursos necesarios para que el producto funcione de acuerdo con lo detallado en la etapa de análisis. Se debe establecer el orden del sistema para su creación. Un correcto diseño permite la optimización de los recursos.
Codificación	Se realiza lo que va hacer funcionar el sistema, se construirá por separado cada uno de los elementos que se definieron en la fase de diseño, para lo cual se debe utilizar las herramientas adecuadas. Para ello se puede usar lenguajes de programación, sistemas de bases de datos, sistemas de información, entre otros.
Integración	Se trata de unir todo lo anterior para así obtener el sistema completo. Además, en esta fase se debe realizar las pruebas necesarias para garantizar que funcione correctamente.
Explotación	Hace referencia al tiempo en el que está en funcionamiento el sistema.
Mantenimiento	Después de la fase de explotación se debe realizar el mantenimiento, lo cual ayuda a corregir errores o mejorar el sistema.

Fuente: Adaptado de Gómez & Moraleda [33]

Las fases de desarrollo de software varían según el punto de vista de diferentes autores, por lo cual a continuación la siguiente información.

Tabla 5. Fases de Desarrollo de Software según el punto de vista de diferentes autores.

FASES DE DESARROLLO DE SOFTWARE	AUTORES								
	Pressman [34]	Sommerville [35]	Vera [36]	Ruiz, Gómez, Gilbert, Soca & Rodríguez [32]	Delgado & Reyes [37]	Gómez, Marciallo & Ramirez [38]	Gómez & Moraleda [33]	Romero [39]	Aguirre & Aguirre [40]
Documentación					✓				
Inicio/Comunicación	✓								✓
Especificación del software		✓							
Planificación/ Planeación	✓			✓					✓
Requisitos								✓	
Análisis / Análisis de requerimientos			✓	✓	✓	✓	✓		
Ejecución / Desarrollo / Construcción	✓		✓						✓
Diseño / Diseño y Arquitectura / Modelado	✓	✓	✓	✓	✓	✓	✓	✓	
Codificación/ Programación					✓	✓	✓		
Control									✓
Integración						✓	✓		
Implementación		✓	✓	✓				✓	
Pruebas/ Validación		✓	✓	✓	✓				
Cierre									✓
Despliegue	✓								
Explotación							✓		
Verificación								✓	
Mantenimiento/Evolución		✓		✓	✓	✓	✓		

Fuente: Elaboración propia.

1.4. METODOLOGÍAS Y MODELOS UTILIZADOS EN EL DESARROLLO DE SOFTWARE

Según Zumba & León [41] con el paso de los años se ha visto como las metodologías y modelos de desarrollo de software han ido evolucionando. Su inicio fue en los años 40 cuando aparecieron los primeros ordenadores, en ese entonces no se tenía parámetros ni estándares. Por lo cual muchos de los proyectos presentaban fallas y no cumplían con las necesidades del cliente y en muchos de los casos sobrepasaba el tiempo de entrega y los costos establecidos. Al pasar el tiempo fueron apareciendo metodologías y modelos que integraban estándares, controles y formalidades que permitían corregir ciertos problemas como: fallas, incumplimiento de las necesidades del cliente, sobrecostos y entregas tardías. Con la llegada del internet se permitía realizar cambios en el proceso de desarrollo de software y entregar a tiempo, estos modelos y metodologías se denominan ágiles los mismos que permiten ir diseñando y probando el producto con el fin de que cuando se requiera la entrega este sea de calidad.

Según Ruiz, Gómez, Gilbert, Soca & Rodríguez [32] el término Metodología en desarrollo de software se define como un enfoque estructurado y su propósito es agilizar la producción de un software de calidad a un costo razonable. Hay que realizar una diferencia entre las tradicionales, formales o primeros enfoques estructurados y las ágiles. Los enfoques tradicionales se utilizan desde los años 80, de forma paralela con muchos de los modelos ya expuestos más adelante.

1.4.1. Modelos Tradicionales de Desarrollo de Software

Un modelo de desarrollo de software viene siendo una representación abstracta del proceso, dicho modelo va determinando el orden que va a ir las actividades del proceso de desarrollo del software [42]. Un modelo tradicional se lo considera de esta forma porque fue el primero en incluirse al desarrollar el software, estos modelos consideran de manera puntual una disciplina de trabajo sobre el desarrollo de software, logrando obtener resultados eficientes y predecibles. Antes de su desarrollo es necesario un enfoque en la planificación total del todo el trabajo que se vaya a realizar y que todo esté detallado. Dentro de los modelos tenemos: cascada, V, espiral, entre otros [43].

Tabla 6. Principales modelos tradicionales entre 1979 y 1990.

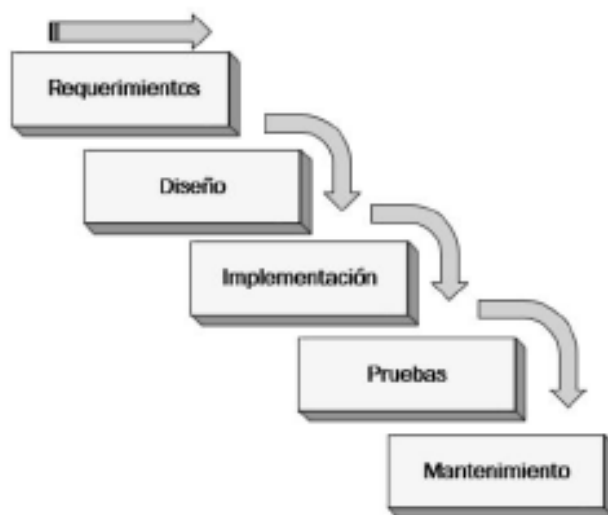
MODELOS TRADICIONALES	
1	Modelo de Cascada
2	Modelo Cascada en “V”
3	Modelo de Desarrollo Evolutivo (Espiral)
4	Modelo de Desarrollo Evolutivo por Prototipos
5	Desarrollo Evolutivo por etapas o Incremental
6	Desarrollo Evolutivo Iterativo
7	Modelo Basado en Componentes

Fuente: Tomado de Zumba & León [41].

1.4.1.1. Modelo en Cascada

El modelo en cascada como nos menciona Aguirre & Aguirre [40] es una metodología del marco tradicional conocida también como Waterfall o Predictiva implementada por Winston W. Royce en 1970 y ha sido utilizada durante muchos años para la gestión de procesos de desarrollo de software. Cuando se introdujo por primera vez se adaptó a varias industrias debido a su secuencia lógica y facilidad de implementación.

Figura 4. Fases de desarrollo de software de acuerdo al Modelo en Cascada.



Fuente: Tomado de internet.

El flujo de las actividades es secuencial y se agrupan en cinco fases. Una fase se considera terminada cuando los productos de trabajo de cada fase son revisados y validados, estos se consideran como punto de referencia para iniciar la siguiente fase. Un punto importante a tener en cuenta es la documentación [44].

Tabla 7. Explicación de las fases de desarrollo del modelo en cascada.

FASES	DESCRIPCIÓN
Requerimientos	Definición del plan y asentamiento del inicio.
Diseño	Desarrollo de los planes de gestión.
Implementación	Ejecución de las actividades establecidas en el punto anterior para terminar los entregables.
Pruebas	Supervisión de las actividades ejecutadas y comparación con la planificación u organización. En este punto se ejercen medidas de corrección en caso de desviaciones.
Mantenimiento	Finalización del plan. Se presentan 3 escenarios los cuales son: cuando se consigue la aprobación, cuando no es posible ejecutar alguna tarea planificada o cuando el proyecto pierde viabilidad.

Fuente: Adaptado de Aguirre & Aguirre [40].

Tabla 8. Ventajas y desventajas del modelo en Cascada.

VENTAJAS	Es uno de los modelos más fáciles de administrar ya que cada fase tiene entregables específicos y cuenta con un proceso sencillo de revisión.
	Es más fácil realizar la planificación y programación ya que los requisitos del proyecto se acuerdan en la primera fase, útil para proyectos de menor tamaño, permitiendo una fácil medición del progreso. (costo, recurso y plazos)
	No se retrasa la ejecución del proyecto ya que los clientes no pueden agregar constantemente nuevos requisitos
	El proceso y los resultados estarán documentados
DESVENTAJAS	No es un modelo apto para proyectos de gran tamaño, es poco efectivo si los requisitos no son establecidos claramente desde el inicio.
	Es una metodología lineal y rígida, carece de flexibilidad para adaptarse a sucesos inesperados.
	El proceso de prueba comienza una vez finaliza el desarrollo. Por lo tanto, hay muchas posibilidades de encontrar errores que pueden ser caros de solucionar.

Fuente: Adaptado de Patino [45].

1.4.1.2. Modelo de Cascada en “V”

Este modelo fue propuesto por Alan Davis a principios de los 90 de acuerdo a Quesada [44] menciona que el modelo en V se deriva del modelo en cascada, con la innovación de trabajar con actividades más efectivas y productivas, este modelo se esfuerza por asegurar la calidad del software. Por lo mismo se ejecutan distintas pruebas en medida que se avanza el proyecto como por ejemplo pruebas unitarias, de integración, del sistema y de aceptación, las mismas se deben desarrollar en paralelo con las actividades de desarrollo para continuar con las siguientes fases.

Figura 5. Fases del Modelo en V.

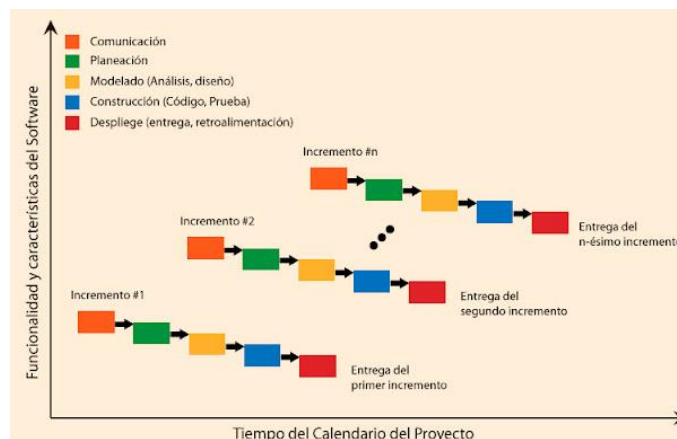


Fuente: Tomado de Pressman [34].

1.4.1.3. Modelo de Desarrollo Incremental

Creado por Harlan Mills en el año 1980 como menciona Zumba & León [41], el modelo incremental se desarrolla cuando el incremento de una actividad funcione correctamente, con la diferencia de que se puede ir hacia adelante o hacia atrás. El primer incremento se basa en los requisitos básicos y cada incremento es una entrega, cuando se termina el usuario puede dar su opinión, evaluar y validar cada una de las funcionalidades de tal modo que puede manifestar su inconformidad cuando no se cumpla con alguno de los requisitos que se acordaron. Incluso el usuario puede pedir cambios o agregar nuevas funcionalidades.

Figura 6. Fases del Modelo de Desarrollo Incremental.



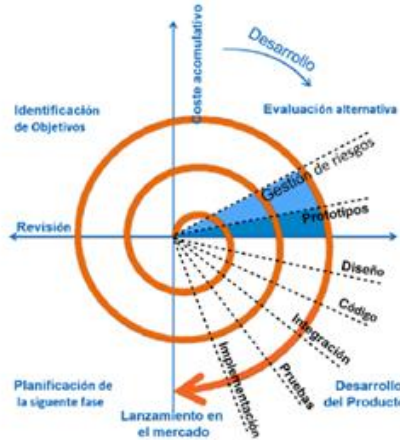
Fuente: Tomado de Ortiz [46].

1.4.1.4. Modelo de desarrollo evolutivo (espiral)

Postulado por Barry Boehm en 1986, en su artículo “A Spiral Model of Software Development and Enhancement”. Es un modelo evolutivo, se puede cambiar los requerimientos es cualquier momento de desarrollo. Los ciclos se organizan de forma espiral, los ciclos internos indican el

análisis y construcción del prototipo y los externos son el ciclo de vida clásico. Su dimensión radial muestra el costo de desarrollo y la angular su progreso. Este modelo involucra mucho tiempo, cada ciclo repetitivo va ganando madurez el producto final [47].

Figura 7. Fases del Desarrollo del Modelo de Desarrollo Evolutivo (Espiral).



Fuente: Tomado de Internet.

1.4.1.5. Modelo evolutivo por prototipos

Este modelo de acuerdo a Delgado & Díaz [48] se basa en analizar lo requisitos de los usuarios, inicia con la recolección de información es decir los requerimientos del cliente y a partir de ellos se determina los objetivos y en base a ello se genera el prototipo que el cliente debe evaluar y de esta manera aclarar los nuevos requisitos del sistema, este proceso se repetirá hasta que el cliente esté de acuerdo con el desarrollo del software.

Figura 8. Modelo de prototipo.



Fuente: Tomado Delgado & Díaz [48].

1.4.1.6. Modelo de desarrollo basado en Componentes

Propuesta por Fred Brooks en 1987 bajo la filosofía, “compre y no construya.” Está compuesta por varias características del modelo en espiral, tienen un enfoque evolutivo e iterativo, se construye un sistema a partir de productos pre-fabricados que se pueden reutilizar, con ello se presentan grandes beneficios para los desarrolladores de software. Comprar componentes de terceros en lugar de desarrollarlos tiene sus ventajas como: ciclos de desarrollo más cortos y una funcionalidad mejorada [49]. De igual manera Mármol & Perez [50] menciona que su desarrollo consiste a través de un proceso de unión de partes independientes y que se puede reutilizar.

1.4.2. Metodologías Tradicionales de Desarrollo de Software

De acuerdo a Molina, Vite & Dávila [5] las metodologías tradicionales se plantearon para organizar y disminuir los problemas del desarrollo de software. Estas metodologías se presentaron en la década de los 60, en este tipo de metodologías como los modelos tradicionales se basa en el proceso secuencial en una sola dirección.

Tabla 9. Principales metodologías tradicionales entre 1979 y 1990.

METODOLOGÍAS TRADICIONALES	
1	RAD (Rapid Application Development)
2	RUP (Rational Unified Process)
3	MSF (Microsoft Solution Framework)
4	Win-Win Spiral Model
5	Iconix
6	Desarrollo de sistemas de Jackson (JSD).
7	Ingeniería de la información.
8	Structured System Analysis and Design Method
9	(SSADM)

Fuente: Tomado de Zumba & León [41].

1.4.2.1. Rapid Application Development (RAD)

Desarrollo Rápido de Aplicaciones, presentado por James Martin desde 1980, es un modelo que dura entre 60 y 90 días, comprende un desarrollo interactivo, permite la elaboración de prototipos mediante software asistido por computadora o CASE (Computer Aided Software Engineering). Es una metodología que fue diseñada para entregar un producto mucho más rápido permitiendo ser implementado en un tiempo más corto. Por lo antes mencionado se puede decir que permite obtener resultados en menor tiempo y de acuerdo a ello satisfacer necesidades de los usuarios lo más pronto posible [51].

Tabla 10. Etapas de la Metodología RAD.

ETAPAS	DESCRIPCIÓN
Modelado de gestión	Se origina información que conlleva el proceso de gestión, identificando a dónde se dirige la información y quién la analiza y procesa.
Modelado de datos	Se determinan los datos y las relaciones que existen entre sí.
Modelado del proceso	Se utiliza para añadir, modificar, suprimir o recuperar un objeto de datos.
Generación de aplicaciones	Se hace uso de herramienta de cuarta generación que permite crear el software y facilitar el desarrollo del programa.
Pruebas y entrega	El proceso de desarrollo finaliza con pruebas de calidad y después se realiza la implementación.

Fuente: Adaptado de Delgado & Díaz [48].

1.4.2.2. Rational Unified Process (RUP)

Propuesta en 1998 por Ivar Jacobson, Grady Booch y James Rumbaugh. Desarrollada por Rational Corporation (Compañía que pertenece a IBM). Su objetivo es garantizar la calidad del producto cumpliendo con las necesidades del cliente dentro del tiempo y costo previsto [37].

1.4.2.3. Microsoft Solution Framework (MSF)

De acuerdo a Microsoft [52] Microsoft Solution Framework (MSF) tiene un enfoque práctico que ofrece soluciones técnicas más rápido con menos personal y riesgos, permite obtener resultados de mayor calidad y ayuda a solucionar problemas comunes que provocan el fracaso de los proyectos aumentando la tasa de éxito, calidad e impacto comercial. Por su parte Cañas [53] manifiesta que es un modelo que combina dos modelos que son comunes y son el modelo en cascada y espiral, además de que cuenta con 5 etapas las cuales son: Visión (Visualizar), planificar, desarrollar, estabilizar e implementación.

1.4.3. Metodologías Ágiles de Desarrollo de Software

Las metodologías ágiles eran para gestionar la creación y desarrollo del software, surge para cubrir los problemas que se presentaban con las metodologías tradicionales y permitir trabajar en el desarrollo de software a grandes pasos. Las organizaciones han utilizado las metodologías ágiles de distintas formas pues se pueden adaptar a las necesidades de los usuarios. Aportan flexibilidad es decir que permite realizar cambios sin alterar las condiciones del proyecto, obteniendo una respuesta rápida y reduciendo costos. Se caracteriza por sus equipos de trabajo pues se organizan de acuerdo a las necesidades del proyecto, en cada etapa trabaja un grupo diferente por ello la comunicación entre ellos es fundamental [54].

Tabla 11. Metodologías Ágiles

METODOLOGÍAS ÁGILES			
1	Adaptative Software Development	11	Internet Speed Development
2	Agile Modeling	12	Lean development
3	Agile Model Driven Development	13	Mobile-D
4	Agile Project Management	14	Open Unified Process
5	Agile Unified Process	15	Pragmatic programming
6	Crystal Methods	16	Scrum
7	Dynamic Systems development methods	17	Story cards driven development
8	Evolutionary Project Management	18	Test Driven Development
9	Extreme Programming	19	Win Win Spiral
10	Feature Driven Development	20	X-Breed

Fuente: Tomado de Zumba & León [41].

1.4.3.1. Extreme Programming

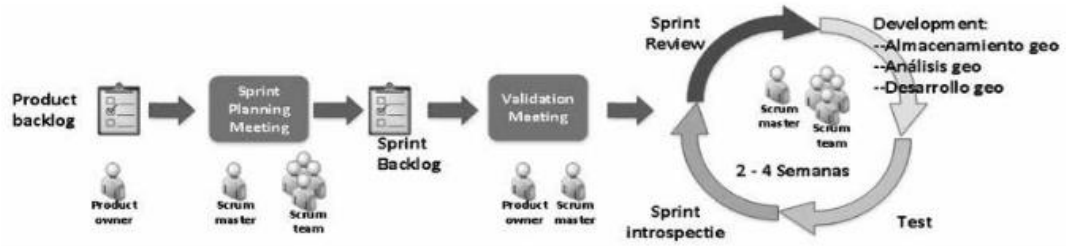
Propuesta por Kent Beck, buscando guiar a equipos de desarrollo de software pequeños. Es una metodología de desarrollo de software que se basa en la simplicidad, comunicación y retroalimentación. Cada participante es importante, así como la comunicación con el cliente, esta debe ser permanente. Utilizan una manera sencilla de planificación y seguimiento con lo cual se puede decidir lo que se debe hacer después y de esta manera saber cuándo finalizará el proyecto. El equipo desarrolla una serie de entregables que se deben aprobar con el cliente, los programadores trabajan en pares mejorando el diseño continuamente [55]. Además Ramírez, Branch & Jiménez [56] menciona que es un proceso ágil que se basa en buenas prácticas de codificación, comunicación y trabajo en equipo, posee una serie de reglas que se pueden dividir en planeación y gestión, diseño, codificación y pruebas.

1.4.3.2. Scrum

Los autores Gómez, Marcillo & Ramírez [38] señalan en su investigación que Scrum es un proceso ágil para desarrollar software el cual se aplicó por primera vez por el ingeniero Ken Schwaber y el médico Jeff Sutherland en 1995, los cuales aseguran que el ciclo de vida se incrementa, se repite y se caracteriza por ser muy adaptable.

De igual manera Ramírez, Salgado, Ramírez M., Manrique, Osuna & Rosales [57] manifiesta que es una metodología que está compuesta por un conjunto de buenas prácticas y trabajo en equipo lo cual permite obtener mejores resultados y genera entregas parciales del proyecto las mismas que son validadas por el usuario final. Su uso se recomienda en proyectos complejos, cuando se tiende a cambiar los requisitos y es importante obtener resultados rápidos. Ofrece ventajas de adaptabilidad, retroalimentación continua, se divide en entregables lo cual permite determinar y crear soluciones de una forma rápida.

Figura 9. Esquema de la Metodología Scrum



Fuente: Tomado de Arias & Durango [58].

1.4.3.2.1. Roles de Scrum

En esta metodología los equipos de trabajo deben tomar sus propias decisiones, multifuncionales, es decir deben auto-organizarse.

Tabla 12. Roles de Scrum.

ROLES DE SCRUM	
Nombre	Descripción
ProductOwner (Dueño del producto)	Responsable de que el equipo de trabajo entienda cuales son los requisitos del proyecto, definir el alcance y las prioridades.
DevelopmentTeam (Equipo de desarrollo)	Son los profesionales quienes pueden ejecutar su trabajo para obtener los resultados de calidad con eficiencia y eficacia.
Scrum Máster	Responsable de que el proceso de entienda, se aplique y se apliquen las reglas de Scrum. Líder del equipo que debe priorizar los requisitos, facilitar las reuniones de forma productiva, enseñar al equipo a autogestionarse, evitar los problemas y aislar al equipo de las interrupciones externas.

Fuente: Adaptado de Romero [39].

1.4.3.2.2. Artefactos de Scrum

Los artefactos de scrum tienen como objetivo disminuir las actividades de documentación y centrarse en el desarrollo para que la información sea clara para todos los miembros del equipo [58].

Tabla 13. Artefactos de Scrum.

ARTEFACTOS DE SCRUM	
Nombre	Descripción
ProductBacklog	Lista de todos los requerimientos del producto que necesariamente debe ir cambiando para que se adecuado y debe siempre estar visible
Burndown	Seguimiento del avance, se trata de un gráfico del trabajo pendiente y los avances que se están realizando.
Sprint Backlog	Tablero físico llamado Taskboard donde el equipo de trabajo hace pública sus actividades dividiéndolas en ítem, que también va cambiando durante la ejecución del Sprint por ello es importante que el equipo de trabajo actualice constantemente la lista.

Fuente: Adaptado de Romero [39].

1.5. LENGUAJES DE PROGRAMACIÓN

El término lenguaje de programación trata de un grupo de actividades entendibles y ejecutables que se pueden realizar por medio de un computador, con sintaxis propia (en español), entorno y reglas de desarrollo. Las reglas necesarias para que el ordenador por medio de compilador entienda las instrucciones y luego las ejecute. Un compilador es un programa que comprueba que se cumplan las reglas sintácticas de un lenguaje de programación [59].

1.5.1. Python

Guido Van Rossum creó Python en diciembre de 1998, el cual se basó en el lenguaje de programación ABC el cual tiene como objetivo principal la facilidad de aprendizaje. Python es un lenguaje de programación general y ha obtenido popularidad en el desarrollo de aplicaciones web, administración de sistemas, ciencia de datos, computación, inteligencia artificial, entre otros [60]. Python es un lenguaje de programación de alto nivel, su característica principal es su simplicidad es decir fácil de leer, escribir, limpiar y depurar. Sarasa [61]. De la misma forma Walker [62] menciona que dispone de la funcionalidad de usarse en cualquier plataforma a la vez que su código es fácil de entender e intuitivo. Su sintaxis y comandos es distinto a otros lenguajes de programación que se hayan usado, siendo así más sencillo e incluso minimalista.

Para Troyano, Cruz, González, Vallejo & Toro [63] Python es un lenguaje de programación donde lo esencial es la legibilidad del código, en la comunidad de desarrolladores se denomina *pitónico* por la claridad y elegancia del lenguaje, además es multiparadigma los mismos que están separados lo que permite que se pueda ingresar de forma progresiva, cuenta con una amplia biblioteca con herramientas necesarias para resolver múltiples tareas.

1.5.2. PostgreSQL

Es un sistema de base de datos objeto-relacional, el cual se distribuye bajo la licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado. Es compatible con una gran parte del estándar SQL. (Sobre PostgreSQL) [64]. Creado por el departamento de Ciencias de la computación de Berkeley de la Universidad de California. Su arquitectura posee una gran reputación gracias a su fiabilidad, integridad de los datos y su buen funcionamiento. Además, puede instalarse en un gran número de sistemas operativos, incluyendo Linux, Unix y Windows [65].

“Los datos son almacenados en tablas de columnas y renglones. Con el uso de llaves, esas tablas se pueden relacionar unas con otras.” [66].

1.5.3. Framework Django

Es un marco web Python de alto nivel que fomenta el desarrollo ágil y diseño limpio y pragmático, fue creado por desarrolladores experimentados, es gratis y de código abierto. La seguridad es lo más importante y por ello se lo toma muy en serio evitando cometer errores comunes [67].

De acuerdo a Gómez [68] Django es un framework web de alto nivel escrito en Python el cual permite desarrollar sitios web de forma rápida, segura y mantenibles. Tiene comunidades de colaboración que se extienden por todo el mundo, los mismos se encargan de actualizar día a día el framework y la documentación. Se basa en el principio *Don't Repeat Yourself* es decir que lo que ya se ha realizado no se tiene que volver hacer. Está compuesta por métodos para registrarse, iniciar y cerrar sesión, panel de administración cuenta además con formularios, plantillas, reportes, entre otros. Django utiliza el Modelo Vista Controlador (MVC).

1.6. INTELIGENCIA ARTIFICIAL

Según NOTA-INCyTU [69] la IA no es fácil, ya que el concepto no es del todo preciso. En términos coloquiales, IA se usa cuando una máquina es capaz de imitar las funciones cognitivas propias de la mente humana, como: creatividad, sensibilidad, aprendizaje, entendimiento, percepción del ambiente y uso del lenguaje. Un subcampo de la IA que ha ganado auge en años recientes es el aprendizaje computacional (machine learning), donde un sistema aprende a ejecutar tareas, ya sea a partir de ejemplos o mediante prueba y error.

La definición de IA como mencionan los autores no es fácil de definirla debido a que el tema es complejo por lo mismo tiene diferentes definiciones las cuales concuerdan en que la IA es la capacidad que tienen las máquinas para practicar actividades que las realizan los humanos, es decir, imitar funciones que realizan pero con la diferencia de que estas no necesitan descansar, es por ello que se ha venido ganando más espacio, y se ha aplicado en varios campos en los cuales se permiten realizar los trabajos de una mejor manera, más sencilla y más rápida.

Desde tiempos inmemoriales, el hombre ha buscado la materialización del deseo de crear seres semejantes a él, pasando por la creación de artefactos con aspecto, movimientos y hasta comportamiento similar al que presentamos los seres humanos. El ruso Isaac Asimov en el año de 1920-1992, escritor e historiador, narra sobre objetos y situaciones que en su tiempo eran ciencia-ficción; sin embargo, con el paso del tiempo, muchas de ellas se han ido volviendo realidad. Asimismo, en su libro *Runaround* describió lo que el día de hoy son las tres leyes de

la robótica. Su obra literaria serviría como motivación para que los científicos e ingenieros tratarán de hacerla realidad.

En los años 50 cuando se logra realizar un sistema que tuvo cierto éxito, se llamó el Perceptrón de Rosenblatt. Este era un sistema visual de reconocimiento de patrones en el cual se aunaron esfuerzos para que se pudieran resolver una gama amplia de problemas, pero estas energías se diluyeron enseguida. Aproximadamente en ese tiempo, el matemático inglés Alan Turing en el año 1912-1954 propuso una prueba con la finalidad de demostrar la existencia de “inteligencia” en un dispositivo no biológico. Esta prueba conocida como “test de Turing” se fundamenta en la hipótesis de que, si una máquina se comporta en todos aspectos como inteligente, entonces debe ser inteligente [70].

1.6.1. Aprendizaje Automático

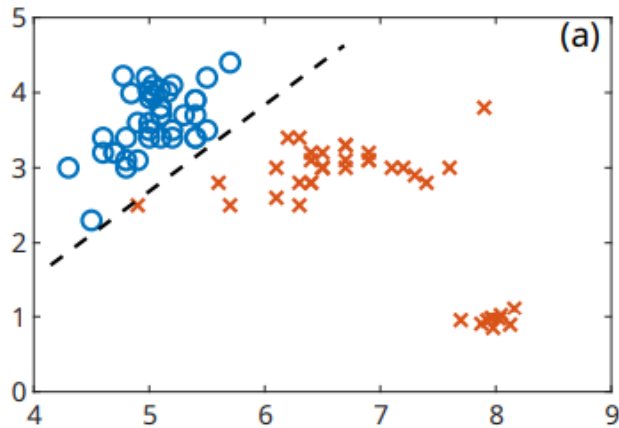
Según Norman [71] el aprendizaje automático es un subconjunto de la IA y se encarga de enseñar a las computadoras a realizar predicciones basadas en datos, donde implica dar a una computadora una serie de datos y realizar una petición de predicciones, en un inicio, la computadora obtendrá demasiadas predicciones erróneas, que, por otro lado, en el transcurso de entrenamiento con miles de predicciones, la computadora actualiza su algoritmo para una mejor predicción. De igual modo, el aprendizaje automático usado en empresas mejora los procesos, proporciona eficiencia productiva, aumenta los ingresos y reduce los costos, por lo tanto, este tipo de aprendizaje día con día obtiene mejoras y no es necesario la presencia humana para funcionar y ser eficiente. El aprendizaje automático puede incorporarse en distintas áreas como economía, salud, seguros, telecomunicaciones, entre otros [72].

1.6.1.1. Aprendizaje supervisado

También conocido como aprendizaje con clase, Sandoval [73] explica que pasa por un proceso de aprendizaje o entrenamiento un algoritmo de Machine Learning, es decir, aprende un patrón a partir de un conjunto de datos donde es usado para su entrenamiento dándole preguntas (características) y respuestas (etiquetas), de tal modo que el algoritmo realice predicciones conociendo las características. Dentro del análisis del aprendizaje supervisado explica Simeone [74] que el conjunto de entrenamiento dispone de pares de entrada y salida, que como objetivo debe aprender un mapeo entre los espacios tanto de entrada como de salida, donde en las entradas se encuentran puntos en un plano bidimensional, en cambio en las salidas son etiquetas que se les asigna en cada entrada (círculos o cruces) y como punto culminante es aprender un clasificador binario. Como consecuencia para Recuero de los Santos [75] las aplicaciones del

aprendizaje supervisado son usados en los problemas de clasificación y dentro de ella están por ejemplo la identificación de los dígitos, diagnósticos de fraude de identidad. Del mismo modo están los problemas de regresión y dentro de ella están las predicciones meteorológicas, predicciones de expectativa de vida, predicciones de crecimiento, entre otros.

Figura 10. Aprendizaje supervisado.



Fuente: Tomado de Simeone [74].

1.6.1.1.1. Máquina de Vectores de Soporte (SVM)

De acuerdo con Betancourt [76] la teoría de Máquinas de Soporte Vectorial (SVM por su nombre en inglés Support Vector Machine) es una nueva técnica de clasificación se basa en la idea de minimización de riesgo estructural. Se ha demostrado tener un gran desempeño más que las máquinas de aprendizaje tradicional como las redes neuronales, aunque han sido introducidas como herramientas poderosas para resolver problemas de clasificación.

Este tipo de algoritmo primero mapea los puntos de entrada a un espacio de características de una dimensión mayor y se encuentra un hiperplano que los separa y maximiza el margen entre las clases, dividiendo la población entre lo usual e inusual, a partir de la población de entrenamiento [77].

Los autores Cano, García, Orts, García, Peña, Pérez & Pérez [78] las máquinas de soporte vectorial son un grupo de métodos de aprendizaje supervisado que se pueden aplicar a problemas de clasificación o regresión. Se introdujeron con el objetivo de clasificar objetos linealmente separables. Representan hiperplanos o fronteras de decisión en términos de pequeños subconjuntos de todos los ejemplos de entrenamiento que maximizan la separación entre conjuntos y por lo tanto la capacidad de clasificación entre clases.

Para Cuevas, Alvarez, Azcona & Rodríguez [79] es una de las técnicas más poderosas del aprendizaje automático a pesar de su sencillez se ha demostrado que es un algoritmo robusto y

generaliza el problema de la vida real. Estudios reportan que las SVM superan en precisión a los modelos autorregresivos, usan modelos lineales para la implementación de clases con separaciones no lineales en el momento de transformar el espacio de entrada en un nuevo espacio. Las SVM están indicadas para ser usadas como clasificadores binarios, la clasificación se divide en dos etapas: la fase de aprendizaje automático y la de reconocimiento. El aprendizaje se logra mediante la búsqueda de alguna dependencia entre el conjunto de vectores con los datos de entrada y salida.

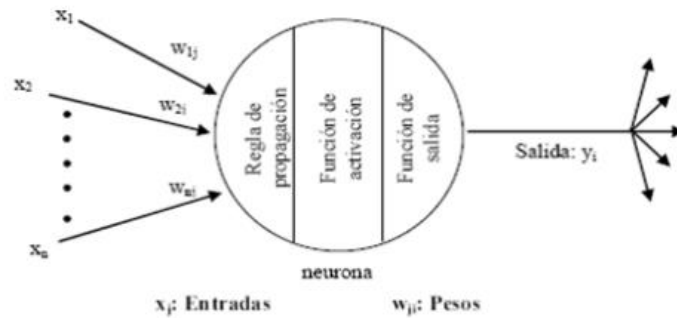
Según Salamanca [80] este algoritmo fue creado por Vladimir Vapnik y su equipo en los años 90. Tuvo un gran alcance por su sólido fundamento teórico, por los excelentes resultados que se obtuvieron en los numerosos problemas de interés. Y en la actualidad se usa con éxito en campos como el reconocimiento de imágenes y textos, recuperación dañada, aplicaciones biomédicas o mejora de motores de búsqueda.

1.6.1.1.2. Redes Neuronales

Según Terán, Martínez, Menchaca, Castán, Treviño, García & Martínez [81] mencionan que son aplicadas en múltiples áreas del conocimiento y son utilizadas comúnmente como clasificadores, para aplicaciones de visión por computadora, como controladores o como mecanismos de pronóstico. Estas redes pretenden simular el comportamiento de neuronas biológicas a base de funciones matemáticas. Una de las características más importantes de las redes neuronales es que son capaces de aprender a través de la experiencia, que también se puede apreciar como aprendizaje a partir de un conjunto de datos de entrenamiento.

Las redes neuronales para Toral [82] vienen siendo modelos matemáticos e intentan imitar el comportamiento del cerebro humano, esta red tiene como fin la construcción de sistemas que puedan presentar un cierto comportamiento inteligente, es decir, tener la capacidad para poder aprender a realizar una tarea en específica. Una red neuronal consta de tres partes principales: entrada, núcleo y salidas. La entrada es quien recibe datos para poder decidir si la neurona estará activa o no, se las representa como: X_1, X_2, \dots, X_n . Los pesos entre la entrada y el núcleo se lo representan como W_1, W_2, \dots, W_n y representa la memoria de la red. El núcleo es quien realiza todas las operaciones para poder determinar su salida de la neurona, los procesos en el núcleo van variando, es decir, depende de la red neuronal que se vaya a trabajar. Las salidas es la respuesta a la neurona, si está activa o no, se las representa como: Y_1, Y_2, \dots, Y_n . En la siguiente gráfica se representa un modelo de red neuronal.

Figura 11. Modelo de una red neuronal artificial



Fuente: Tomado de Toral [82].

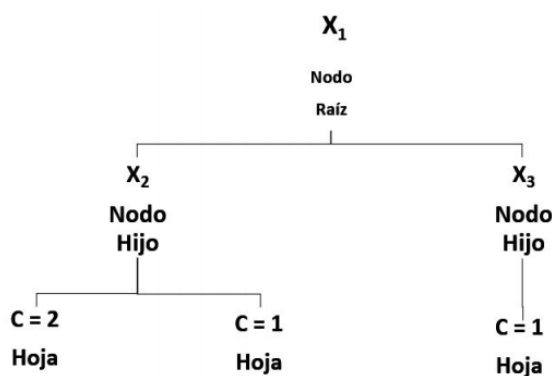
1.6.1.1.3. Árboles de Decisión

Según Dupuoy [83], es una técnica predictiva de clasificación, tiene como objetivo la división jerárquica y secuencial del problema, donde cada una de dichas divisiones o también conocidas como nodos describen de forma gráfica las posibles decisiones y posibles resultados de las diferentes combinaciones de decisiones y evento. Por otro lado, a un evento se le añade probabilidades y a cada rama se le va determinando un resultado. Un árbol de decisión representa reglas que son entendibles por los humanos, el conocimiento lo genera el árbol de decisión y no parte de una suposición de un experto en el tema.

Por otra parte, para Riascos & Molina [84], un árbol de decisión es una herramienta analítica donde ayuda a la selección, estructuración y evaluación de problemas en un ambiente de incertidumbres, adicional permite la evaluación de planes de acción, realizar valoraciones de consecuencias, obtención de cálculos y probabilidades, simulaciones, entre otros.

Un árbol de decisión como nos menciona Contreras, Ferreira & Valle [85], puede ser de regresión o clasificación, para el primero (regresión) la variable que tiene como respuesta es continua, para el segundo caso (clasificación) la variable de respuesta o también conocida como variable de clase es directa. El autor también nos menciona una ventaja del árbol de decisión, ya que nos permite realizar un análisis fácilmente y comprender de mejor forma, por su sencillez, como se da a conocer la Figura 12.

Figura 12. Árboles de decisión



Fuente: Tomado de

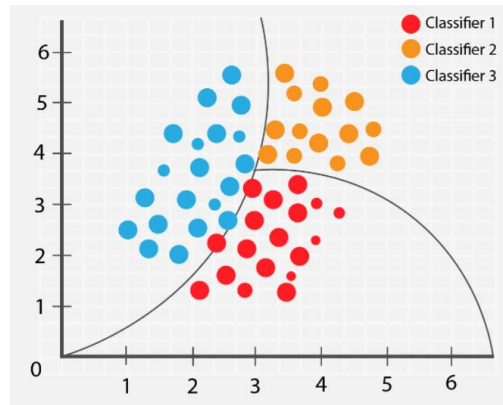
1.6.1.1.4. Naïve Bayes

De acuerdo a González, Torres, Reyes & Villaseñor [86] Naïve Bayes es un clasificador probabilístico, el cual es muy utilizado por ser sencillo y eficiente en la mayoría de los casos, funciona bajo el supuesto de que las características en un conjunto de datos son independientes entre sí. Para clasificar una instancia se determinan las probabilidades de cada valor de los atributos de esa instancia para cada clase disponible, después se elige la clase con el valor de probabilidad más alto.

Para Pereira, López & Quintero [87] el objetivo de la clasificación es entrenar un método determinado a partir de un conjunto de datos para construir un modelo que sea capaz de predecir uno de los valores nominales al que pertenece la clase. La distribución de probabilidades que aprendió se puede ver en forma de árbol. El supuesto de independencia formaría un grafo con arcos que pueden conectar a los nodos, lo cual manifiesta las dependencias entre los atributos. Este tipo de estructura que generaliza a un clasificador Naïve Bayes se le conoce como red bayesiana y es la forma más simple de la misma.

Los autores García, Mata & Zagal [88] hablan acerca del Clasificador de textos Naïve Bayes el cual consiste en asignar a cada documento un colección de etiquetas que denomina la clase a la que este documento pertenece. La decisión se toma a partir de un modelo construido con una parte de la colección que se denomina conjunto de entrenamiento. El objetivo es generar un modelo que pronostique de forma correcta la clase de un conjunto de documentos, cuando se quiere mejorar la clasificación suele usarse dos enfoques los cuales son: modificar el modelo del clasificador o modificar la representación de los datos.

Figura 13. Algoritmo de clasificación Naïve Bayes.



Fuente: Tomado de internet.

Tabla

14. Algoritmos del Aprendizaje Supervisado.

TÉCNICA	ALGORITMOS	DESCRIPCIÓN	REF.
Árbol de Decisión	ID3	Este algoritmo se acopla hacia los datos discretos, específicamente hacia la ganancia de información con el fin de seleccionar características donde se dividen y luego se construye de forma recursiva un árbol de decisión, su complejidad no es alta y los resultados arrojados es fácil de entender.	[89]
	C4.5	Es una mejora del algoritmo ID3, y tiene como fin manejar atributos que son categóricos y continuos en la construcción del árbol de decisión, dividiendo sus valores en dos. Los valores que están encima del umbral se le asigna un hijo y al resto otro hijo. Hace el uso de la ganancia en la selección del atributo para así construir un árbol de decisión.	[90]
	CHAID	Este algoritmo es propuesto por Kass y aplicado en distintos campos de la ciencia, como economía, turismo, educación, entre otros. Es el algoritmo más utilizado y frecuente en los métodos de segmentación, es secuencial y su esquema se divide en 4 fases.	[91]
	MARS	Es un algoritmo usado para solventar dificultades no lineales de regresión, su aplicación puede ser para el pronóstico de series de tiempo, además permite la dependencia de una variable respuesta con respecto a una o más variables explicativas.	[92]
	Árboles de Inferencia Condicional	Algoritmo individual, ayuda a entender a qué grado se puede ir operando una variable en un modelo controlado mediante el efecto del resto, su alto valor está alojado en la interpretabilidad. En tamaños extensos es difícil pensar qué variables y en la forma de aparecer.	[93]
Teorema de Bayes	Naive Bayes	Es un algoritmo clasificador probabilístico simple independiente, proporciona una precisión en los datos en tiempo real que cualquier otro clasificador. Es muy útil cuando se requiere una cantidad pequeña de dato para ser entrenados.	[94]
Relación de variables	Regresión ordinaria por mínimos cuadrados	Se puede manipular las variables controlables que son medibles para la predicción de comportamiento de otras variables, cuando las variables son pocas, no existen problemas de multicolinealidad y una relación precisa entre variables. Si en las tres condiciones falla la regresión no es eficaz.	[95]
Aprendizaje Automático	Regresión logística	Algoritmo más usado en aprendizaje automático, su aplicación a la resolución de problemas de clasificación binaria. Es un algoritmo de fácil interpretación en sus resultados.	[96]
	Métodos de conjunto	Es un algoritmo para combinar predicciones de múltiples estimadores de base que son construidos con un algoritmo de aprendizaje dado, con el fin de mejorar la robustez sobre un solo estimador.	[97]

Fuente: Elaboración propia.

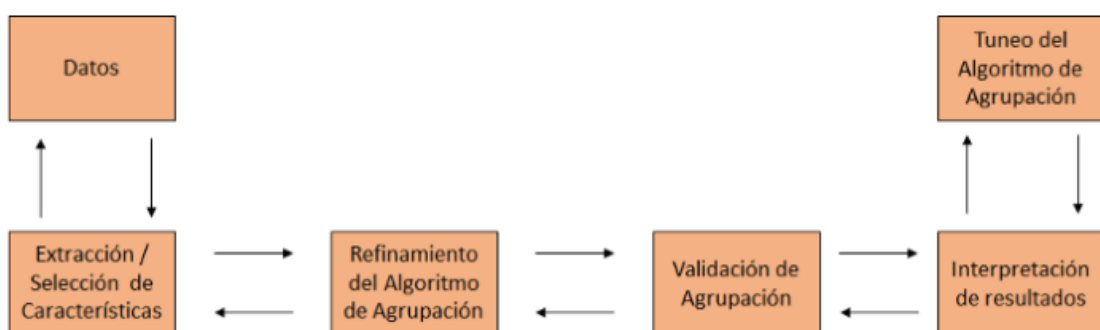
1.6.1.2. Aprendizaje no Supervisado

También conocido como aprendizaje sin clase. Se le asigna las características al algoritmo, más no las etiquetas, en otras palabras, se manejan datos sin etiquetar o datos sin estructura ya que no tenemos información previa que nos aporte de ayuda para identificar patrones o para saber a qué clase pertenecen los datos. Necesitamos que nos agrupe los datos según sus características, el funcionamiento del algoritmo sabe que como comparten ciertas características, llega a asumir que puede pertenecer al mismo grupo [98].

Para Gonzalez [99] el término no supervisado hace referencia a que el algoritmo no tiene que estar guiado como en el caso del algoritmo de Aprendizaje Supervisado; siguiendo con la misma línea del Aprendizaje no Supervisado permite realizar tareas de procesamientos altamente complejas, pero a su vez puede ser impredecible en comparación con los distintos métodos de aprendizaje naturales. Se puede inferir que el Aprendizaje no Supervisado no se le aplica enseñanza alguna, es decir, que va aprendiendo a partir de los datos en específico.

Dentro de las aplicaciones principales del Aprendizaje no Supervisado tenemos, la segmentación de conjuntos en los datos por medio de los atributos compartidos, del mismo modo tenemos la detección de anomalías donde no encajan en ningún grupo y finalmente la simplificación de datasets agregando variables con sus atributos similares, entre otros. En los procesos de análisis se implementa un modelo práctico presentado en la Figura 14.

Figura 14. Modelo de Aprendizaje no Supervisado.



Fuente: Tomado de Roman [100].

1.6.1.2.1. Algoritmos de Agrupamiento (Clustering)

De acuerdo a Vera [101] son técnicas de agrupamiento de aprendizaje automático no supervisado con la característica principal de encontrar patrones que se encuentran dentro de la gran cantidad de datos para producir un conocimiento útil de aporte a la toma de decisiones. El

objetivo es dividir los datos en clustering o grupos y se básicamente se utilizan para organizar, categorizar y analizar datos de gran escala.

1.6.1.2.1.1. K-Means

Para Sanabria [102] este algoritmo fue utilizado por primera vez por James, resuelve el problema de agrupación siguiendo una manera sencilla y fácil de clasificar un conjunto de datos por medio de un cierto número de clusters. Su nombre representa cada uno de los clusters por la media de los centroides y la representación presenta una gran ventaja que tiene un significado gráfico y estadístico inmediato. Es uno de los algoritmos de agrupamiento más utilizados, se traduce al español como K-medias. MacQueen en el año 1967 manifiesta que este algoritmo tiene un propósito de dividir un conjunto de observaciones, donde cada uno es asignada a un grupo de valor medio más cercano de acuerdo al centroide [103].

Con respecto a Mayorga [104] agrupar es dividir un conjunto de datos en un grupo significativo de clusters, se clasifican en dos categorías agrupación de jerarquías y peticiones las cuales se pueden dividir en ascendentes y descendentes. Este tipo de algoritmos son muy utilizados en pronósticos espaciales. Fue propuesto por Cox y Fisher entre 1957 y 1958, desarrollado por Hartigan y MacQueen entre 1975 y 1967. Divide los datos en K grupos donde k que significa el número de grupos el cual se determina de antemano.

El método de agrupamiento de k-means según Espinoza & Alderete [105] es una técnica no jerárquica utilizada para agrupar en k grupos. Cada elemento se asigna a un grupo con el centro más cercano. Actualiza iterativamente los grupos para minimizar la variación de sus elementos, el resultado del método de k-means depende en gran medida del número de agrupaciones que se defina de antemano. De acuerdo a Soni & Patel [106] el valor más alto de la distancia en los grupos se calcula usando una fórmula estándar para medir la distancia que esencialmente muestra la semejanza de los grupos. Es el método de agrupamiento más rápido cuando se utiliza con sus variaciones. La ubicación del centroide tiene mucha importancia debido a que puede generar resultados diferentes cuanto más lejos estén mejor.

1.6.1.2.1.2. Reglas de Asociación

Las reglas de asociación representan una técnica para descubrir asociaciones o correlaciones interesantes a partir de un conjunto de datos, comúnmente son aplicadas a bases de datos transaccionales, las cuales se refieren a una colección de registros de transacciones y cada una está integrada de un conjunto de ítems [107].

Para Lévano [108] en su trabajo de tesis lo explica como un ítem que puede ser un evento, un producto, un síntoma, entre otros. Las reglas de asociación es una técnica muy popular de minería de datos que lo que busca es encontrar tendencias entre los ítems y detectar cuando la ocurrencia de uno de ellos está asociada con otro dentro del mismo conjunto de datos. En las reglas de asociación X es llamado antecedente, mientras que Y es llamado consecuente.

Araujo [109] en su trabajo manifiesta que las reglas de asociación son técnicas que pretenden obtener conclusiones sobre patrones frecuentes, las mismas son encontradas entre la información almacenada de series temporales. Se utilizan principalmente para la toma de decisiones en las organizaciones. Para su desarrollo existen muchos métodos y para el módulo del sistema se utiliza el algoritmo A priori, el cual implica una fase para encontrar patrones llamado conjunto de elementos frecuentes, que son los que aparecen justo en un número de registros en la base de datos que cumplen un umbral específico por el usuario.

Tabla 15. Algoritmos del Aprendizaje no supervisado.

TÉCNICA	ALGORITMOS	DESCRIPCIÓN	REF.
Aprendizaje no Supervisado	PCA	Algoritmo muy utilizado en la identificación de patrones en conjuntos de datos con una cantidad considerable de dimensiones. Este algoritmo matemático permite la reducción de las dimensiones de conjuntos de datos reteniendo gran parte de la variación de ellos en vectores que son llamados componentes principales.	[110]
	SVD	Es un algoritmo de descomposición en valores singulares, es uno de los más conocidos por su perfecta exactitud y ganando notoriedad con el tiempo. SVD aporta a la factorización de matrices permitiendo descomponer una matriz A en otras matrices U, S, V.	[111]
	ICA	El algoritmo de Análisis de Componentes Independientes por su traducción al español, nos ayuda a separar una señal multivariante en una combinación lineal independiente. ICA permite minimizar mayores órdenes de dependencia.	[112]
	Clustering	También conocido como algoritmo de agrupamiento, tiene como fin el agrupamiento de una serie de vectores ya sea en grupos o clusters. El criterio suele ser la similitud, es decir, agrupa vectores similares en grupos. Este algoritmo se considera como no supervisado dentro de la minería de datos.	[113]
	Probabilísticos	Este algoritmo está basado en un resultado que devuelve decisiones aleatorias, y que en promedio obtiene una solución a un problema en específico, con datos establecidos de entrada. Su comportamiento no depende solo con datos de entrada, sino de valores que son producidos por un generador de números aleatorios.	[114]
	Reductores de dimensionalidad	Es un algoritmo donde se mapea un conjunto de datos a subespacios que son derivados del espacio original, de menor dimensión, permitiendo realizar una descripción de datos a un menor costo. Tiene alta relevancia en los procesos asociados al aprendizaje de máquina.	[115]

Fuente: Elaboración propia.

Tabla 16. Algoritmos de Aprendizaje no supervisado (Continuación).

TÉCNICA	ALGORITMOS	DESCRIPCIÓN	REF.
Redes Neuronales	MLP: Multi Layered Perceptron	Es un modelo que se caracteriza por tener entrada y salida, una o más capas ocultas, el entrenamiento de este modelo hace uso de un algoritmo supervisado, que se conoce como backpropagation.	[116]
	Hopfield Network	Esta red es recurrente y completamente interconectada, almacena de forma interna patrones que se pueden presentar de forma incompleta o con ruido, se usa como una herramienta de optimización, del mismo modo es usado en aplicaciones de segmentación y restauración de imágenes.	[117]
Reglas de Asociación	A priori	Permite encontrar un conjunto de elementos frecuentes para generar reglas de asociación desde los registros de una base de datos	[109]

Fuente: Elaboración propia.

1.6.1.3. Aprendizaje por Refuerzo

Este tipo de aprendizaje se da cuando la máquina aprende mediante prueba y error hasta cumplir el objetivo. Analiza entornos y mejora su trabajo de forma que cambia la conducta con la finalidad de lograr su objetivo sin programarlo directamente y se muestra de una forma determinada [118]. Del mismo modo González & Ticona [103] los algoritmos aprenden observando el mundo que les rodea y con el continuo flujo de información en las dos direcciones ejecutando el proceso prueba y error que refuerzan las acciones y recibe una respuesta efectiva en el mundo.

De acuerdo a Redacción España [119] el aprendizaje por refuerzo, Reinforcement Learning o RL. Se basa en un concepto humano como es el "prueba y error". Determina un sistema de recompensas para la máquina cuando acierta. Un ejemplo claro es el ajedrez. Cuando se ingresa las reglas al sistema se enseña a la máquina a jugar una partida, luego, le das una recompensa cada vez que consigue un objetivo (ganar una partida, por ejemplo). De esta manera se puede extraer información útil de su propia experiencia: correlaciones, resultados, conclusiones, entre otros.

Tabla 17. Principales Algoritmos de Aprendizaje por Refuerzo.

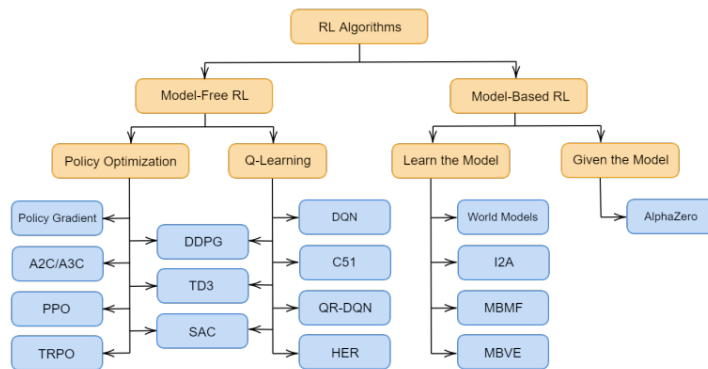
ALGORITMO	DESCRIPCIÓN	REF.
Programación Dinámica	Desarrollada por Richard E. Bellman en la década de los 50, se complementa al ambiente, sirve para dos casos especiales como caso de predicción o control. Considerado como una potente herramienta de optimización resuelve problemas de optimización con restricciones conocidos como problemas de decisión secuencial o multietapa.	[120]
Q-learning	Para estimar la política óptima utiliza una política distinta al contrario que los algoritmos on-policy, que utilizan la misma política que están infiriendo para continuar con la estimación. Esto quiere decir que tendremos que diferenciar entre la política que sigue el agente para aprender y la que está aprendiendo.	[121]
SARSA (State-action-reward-state-action)	Es un método on-policy, es decir, va estimando los valores correspondientes a la política, es decir que la actualización de valores se basa en la política de evaluar y no en una búsqueda directa de la política óptima.	[122]

Fuente: Elaboración propia.

Los algoritmos aprenden de la experiencia es decir que toca darle un refuerzo positivo cuando cumplen con su objetivo. Rouhiainen [15] menciona que la manera en la que los algoritmos aprenden se puede comparar con la de los perros cuando se le da una recompensa cuando aprenden a sentarse por ejemplo.

Uno de los objetivos de la Inteligencia Artificial es el desarrollo de agentes inteligentes que aprendan mediante interacción, basándose en esta idea se puede agregar el aprendizaje por refuerzo como un tercer paradigma de aprendizaje computacional junto con el aprendizaje supervisado y el aprendizaje no supervisado. En RL o aprendizaje por refuerzo un agente interactúa mediante una representación del ambiente, la cual se le denomina estado, el mismo que muestra cómo se encuentra el ambiente para proceder a realizar la acción. El objetivo del agente es maximizar la recompensa obtenida [123].

Figura 15. Taxonomía de los algoritmos de aprendizaje por refuerzo.



Fuente: Tomado de Serrano [124].

1.6.2. Minería de Datos

De acuerdo con Vallejo, Guevara & Medina [125] el importante progreso en la informática y las tecnologías relacionadas junto con la expansión de su uso en diferentes aspectos de la vida, la información se recoge y almacena en grandes bases de datos y su gestión hace posible la integración de diferentes tipos de datos (imagen, video, texto, datos numéricos). La minería de datos nació con la idea de aprovechar dos cosas: la gran cantidad de información (comercio, banca, sanidad) y la potencia de los ordenadores para realizar operaciones de análisis de datos.

La minería de datos trabaja con grandes volúmenes de datos y aplica técnicas adecuadas para el análisis de datos y obtener conocimiento novedoso y útil. A través de diferentes técnicas el cual puede ser expresado a través de reglas, leyes, patrones. Se puede dividir en dos categorías principales: predictivos y descriptivos [126].

Como manifiesta Valero [127] la minería de textos surgió a comienzos de los años 80 con la necesidad del gran esfuerzo humano y los avances tecnológicos ha llevado a creer que la minería de textos tiene un gran valor comercial. Hoy en día la información que nos rodea en la mayoría de los casos es de forma de texto y su crecimiento ha sido mayor y por ello es necesario separar lo esencial y para recuperar esta información se realiza la recuperación de información mediante una serie de técnicas y para ello se necesita una pregunta concreta, pues su objetivo se centra en tomar solamente información útil que satisfaga las necesidades de los usuarios.

Tabla 18. Técnicas de minería de datos.

TÉCNICA	PREDICTIVO		DESCRIPTIVO		
	Clasificación	Regresión	Agrupamiento (Clustering)	Reglas de asociación	Correlaciones / Factorizaciones
Redes Neuronales	✓	✓	✓		
Árboles de Decisión ID3, C4, C5.0	✓				
Árboles de decisión CART	✓	✓			
Otros árboles de decisión	✓	✓	✓	✓	
Redes de Kohoren			✓		
Regresión lineal y logarítmica		✓			✓
Regresión logística	✓			✓	
K-means			✓		
Apriori				✓	
NaïveBayes	✓				
Vecinos más próximos	✓	✓	✓		
Análisis factorial y de comp. ppales.					✓
Twostep, Cobweb			✓		
Algoritmos genéticos y evolutivos	✓	✓	✓	✓	✓
Máquinas de vectores de soporte	✓	✓	✓		
CN2 rules (cobertura)	✓			✓	
Análisis discriminante multivariante	✓				

Fuente: Tomado de Mayorga [104].

Para Marulanda, López & Mejía [128] menciona que se puede definir como el proceso de descubrir conocimiento útil y entendible, desde grandes bases de datos almacenadas en distintos formatos, por medio de modelos inteligentes a partir de los datos. Además, es una parte importante de proceso amplio conocido como descubrimiento de conocimientos en bases de datos KDD y su objetivo principal es la extracción de información oculta de un conjunto de bases de datos por medio de análisis automático o semiautomático.

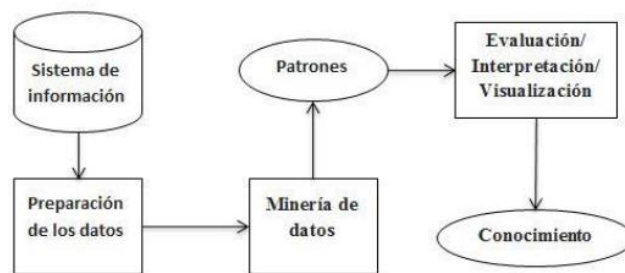
1.6.2.1. Metodologías de la Minería De Datos

1.6.2.1.1. KDD (Knowledge Discovery in Databases)

KDD es un método que empezó a gestarse en 1989 mediante la organización del grupo de trabajo denominado Knowledge Discovery in Databases que hace referencia a descubrimiento de conocimiento en bases de datos durante la Conferencia Internacional de Inteligencia Artificial celebrada en Detroit, organizado por Gregory Piatsky-Shapiro quien decidió utilizar este término para diferenciarlo de los demás métodos [129].

Para Uvidia, Cisneros & Viñan [130] KDD es un análisis automático exploratorio y de modelado de las bases de datos de gran tamaño, su proceso de basa en deducir información validada a partir de los datos. También tiene el significado de un proceso organizado de identificación válida, novedosa y útil que produce patrones comprensibles, además este proceso es iterativo e interactivo.

Figura 16. Proceso KDD.



Fuente: Tomado de Reyes, Flores, Alejo & Rendón [131].

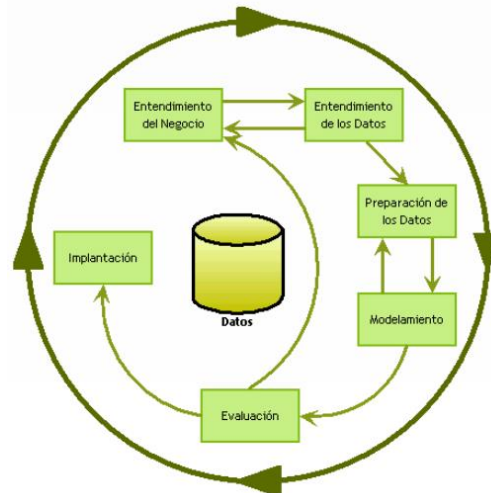
1.6.2.1.2. CRISP-DM (Cross Industry Standard Process for Data Mining)

La metodología CRISP-DM como nos menciona Espinosa [132] es una metodología muy usada en proyectos de minería de datos, para el año de 1997 se empezó su uso con el financiamiento del Programa de Investigación y Desarrollo en Tecnologías de Información de la Unión Europea (ESPRIT) , por sus siglas en inglés, y fue encaminada inicialmente por empresas: SPSS, Teradata, Daimler AG, NCR y Ohra. Su primera versión como metodología se hizo presente en Bruselas en el año de 1999 y para ese mismo año se hizo la publicación como una guía de minería de datos.

En otro sentido Galán [133] nos menciona que para implementar una tecnología ya sea en un negocio tiene que existir la ayuda de una metodología sin duda alguna. CRISP-DM ha tenido bastante apoyo en empresas privadas y organismos públicos en la inclusión de proyectos de

minería de datos. Esta metodología dispone de un modelo o guía basado en 6 fases como son: comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue o implantación. Algunas de estas fases pueden ser bidireccionales, es decir, que de una fase en específica puede volver a una fase anterior para poder ser revisada, la sucesión de fases no tiene que ser netamente ordenada desde la primera hasta la última.

Figura 17. Fases de la Metodología CRISP-DM.



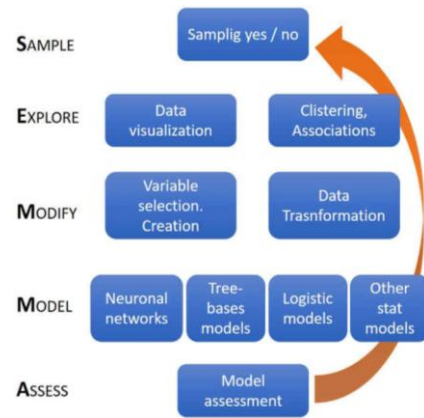
Fuente: Tomado de: Galán [133].

1.6.2.1.3. SEMMA (Sample, Explore, Modify, Model and Assess)

SEMMA es una metodología que están dentro de la minería de datos, como nos menciona Oviedo, Oviedo & Vélez [134] es una propuesta desarrollada por SAS Analytics Solutions para el desarrollo de proyectos de minería de datos. Esta metodología incluye 5 fases como son: muestreo, exploración, modificación, modelado y evaluación. Dentro de sus características principales resalta la inclusión de una fase de muestreo estadístico que no son consideradas en otras metodologías de minería de datos.

Para Moine, Haedo & Gordillo [135] no da una definición de SEMMA, que es el proceso tanto de selección, exploración y modelado de amplios volúmenes de datos con el fin de descubrir patrones de negocio desconocidos. SEMMA fue creada por SAS Institute. Esta metodología esta principalmente enfocada en aspectos técnicos, donde excluye las actividades de análisis y la comprensión del problema en donde se está abordando. Inicialmente fue propuesto para trabajar con el software de minería de dato de la compañía SAS.

Figura 18. Metodología SEMMA.



Fuente: Tomado de Internet.

CAPÍTULO II

2. INTEGRACIÓN DE LA INTELIGENCIA ARTIFICIAL EN EL DESARROLLO SOFTWARE: CASO DE ESTUDIO ECUCIENCIA

2.1. ANTECEDENTES

La Red de Estudios Cienciométricos nace en el año 2018, proyecto perteneciente a la Carrera de Sistemas de Información de la Universidad Técnica de Cotopaxi. En este proyecto se aplican diferentes algoritmos de aprendizaje automático bajo la línea de investigación Tecnologías de Información y Comunicación (TICs) y las sublíneas de investigación Inteligencia Artificial e Inteligencia de Negocios para la Toma de Decisiones y Ciencias informáticas para la modelación de sistemas de información a través del desarrollo de software y el desarrollo de los proyectos de investigación formativa Modelamiento de Algoritmos para Sistemas de Información y Desarrollo Tecnológico de Sistemas de Información Automatizados.

La plataforma de EcuCiencia cuenta con información de las investigaciones registradas por sus usuarios, las cuales corresponden hasta la actualidad a 853 artículos científicos, 272 libros y 866 ponencias. Consta de 3 secciones, la sección uno corresponde al inicio que permite el acceso a los investigadores, la sección dos corresponde a la información del proyecto y la sección tres responde a reportes de los resultados de ciencia métrica como clasificación con Máquinas de Soporte Vectorial (SVM), reportes de estadísticas de artículos, capítulos de libros, libros, y ponencias establecidas por años de publicación, similaridad de investigaciones mediante algoritmos de clasificación como K-means y técnicas de minería de datos.

La red social es otro componente del proyecto de EcuCiencia donde se aplica el Análisis de Redes Sociales (ARS), el cual nos ayuda a identificar las relaciones y colaboraciones científicas que existen entre investigadores. Cuyo fin de aplicación fue el estudio de las relaciones entre artículos, libros y ponencias. Utiliza la metodología Estructural de Wasserman y Faust que permite un balance de toda la información que almacena en la base de datos, fortaleciendo el interés de la identificación de las relaciones y colaboraciones científicas entre los investigadores. Otro de los componentes importantes de esta plataforma es la Georreferenciación, aplicada por ser una técnica de posicionamiento espacial mediante un sistema de coordenadas específicas para identificar la ubicación de los investigadores y los lugares en donde fueron presentadas las investigaciones científicas correspondientes.

Adicionalmente, la plataforma tiene aplicado algoritmos del procesamiento del lenguaje natural que permiten identificar la riqueza léxica de los documentos alojados y la obtención del corpus, así como la generación de nube de palabras. Permite entre otras cosas la visualización de la producción científica y la determinación del estado actual de la ciencia en un determinado grupo de investigación, es decir, artículos científicos, libros y ponencias desarrollados en la región y gracias a ello se puede contar con un inventario real y actualizado de las investigaciones realizadas y publicadas en revistas indexadas a nivel regional y de alto impacto. Además de ello, los autores pueden formar comunidades colectivas de conocimiento para compartir información y formar grupos de trabajo de acuerdo a las líneas de investigación establecidas. Con el desarrollo de proyectos que plantearon el uso de herramientas de inteligencia computacional para la clasificación automática y la generación de representaciones gráficas que sirvieron para realizar estudios de la producción científica mediante una modelación sistémica y la aplicación de metodologías avanzadas de recuperación de información.

Se determinaron las tecnologías de ingeniería de software y de inteligencia artificial más adecuadas para procesar datos y visualizar la información. Python fue el lenguaje de programación utilizado para el diseño y desarrollo de la plataforma EcuCiencia debido a sus funcionalidades de uso, código de fácil entendimiento y accesibilidad para el desarrollo algoritmos de IA y Machine Learning. Por otro lado, se utilizó el gestor PostgreSQL Server como sistema de base de datos, porque ser bastante robusto, por su flexibilidad y compatibilidad con el lenguaje de programación Python. Django también fue aplicado debido a que fomenta un desarrollo rápido, diseño limpio y pragmático ya que se encarga de gran parte del desarrollo web y está orientado al Modelo - Vista - Controlador (MVC) que tiene afinidad con los elementos y sentencias programables de Python y PostgreSQL.

Como resultado de los micro proyectos realizados por estudiantes y docentes de la Universidad Técnica de Cotopaxi de la Carrera de Sistemas de Información se ha logrado plasmarlo en uno solo como es el proyecto Red de Estudios Cienciométricos (REDEC) también conocido como EcuCiencia el mismo se ha logrado con la ayuda de los diferentes grupos de investigación establecidos en la Institución como son Modelamiento de Algoritmos para Sistemas de Información, Red de Estudios Cienciométricos, Desarrollo Tecnológico de Sistemas de Información Automatizados. Para esta fase se han tomado los datos del Repositorio Digital de Tesis de la UTC en los cuales se han identificado 12 tesis de Pregrado, 5 de Posgrado y una de Doctorado que aún se encuentra en estudio.

Tabla 19. Detalle de Tesis de la Plataforma EcuCiencia.

ID	Estudio	Detalle	Año	PRE / POS	Ref.	Estado
TPRE01	Estudio de Algoritmos de Filtrado Basado en Contenidos para Sistemas Recomendadores de Información	Establecer los algoritmos de filtrado basado en contenidos para el diseño de sistemas recomendadores de información.	2018	PRE	[136]	Estudio
TPRE02	Estudio de Algoritmos de Filtrado Colaborativo para Sistemas Recomendadores de Información	Establecer los algoritmos de filtrado colaborativo para el diseño de sistemas recomendadores de información.	2018	PRE	[137]	Estudio
TPRE03	Implementación de una Aplicación Móvil para la Visualización de la Producción Científica de la Universidad Técnica de Cotopaxi	Sistema encargado del inventario de la producción científica como libros, artículos científicos publicados en revistas indexadas y ponencias de los usuarios internos y externos de la plataforma científica.	2018	PRE	[138]	Implementado
TPRE04	Aplicación de Algoritmo de Extracción de Texto en Perfiles de Usuario en Caso de los Investigadores de la Universidad Técnica de Cotopaxi	Recolección de información mediante la implementación de algoritmos de clasificador automática de textos para estructurar datos de relevancia	2019	PRE	[139]	Implementado
TPRE05	Método para la Determinación de Similitud y Distancia entre Investigadores a partir de Algoritmos de Clasificación	Aplicación de algoritmos de clasificación para representar la similitud y distancia de investigadores de acuerdo a su producción científica.	2019	PRE	[140]	Implementado
TPRE06	Implementación de un Algoritmo para la Evaluación de los Documentos Científicos de los Investigadores de la Universidad Técnica de Cotopaxi	Aplicación de un algoritmo para ponderación de documentos científicos e investigadores con el fin de tener un peso de los documentos y por ende a los docentes investigadores.	2019	PRE	[141]	Estudio
TPRE07	Implementación de un Algoritmo de Lógica Difusa en la Identificación de Incentivos y Recomendaciones para los Investigadores de la Universidad Técnica de Cotopaxi	Aplicación de algoritmos de lógica difusa para la identificación de incentivos y recomendaciones a partir de la producción científica de los investigadores	2020	PRE	[142]	Implementado
TPRE08	Un enfoque de Machine Learning en el Desarrollo de Sistemas Recomendadores	Desarrollo de un sistema de recomendación para la conformación de grupos de investigación basado en los perfiles de usuarios, áreas de conocimiento y características de similitud	2020	PRE	[143]	Implementado
TPRE09	Procedimiento Algorítmico Basado en Técnicas del Procesamiento del Lenguaje Natural para el Análisis del Corpus de Artículos Científicos de la Plataforma EcuCiencia.	Aplicación de un procedimiento algorítmico mediante técnicas de procesamiento de lenguaje natural para el análisis del corpus de artículos científicos	2020	PRE	[144]	Implementado

Fuente: Elaboración propia.

Tabla 20. Detalle de Tesis de la Plataforma EcuCiencia (Continuación).

ID	Estudio	Detalle	Año	PRE / POS	Ref.	Estado
TPRE10	Implementación de Metadatos Dublin Core y Protocolos de Interoperabilidad para la Plataforma Científica EcuCiencia de la Universidad Técnica de Cotopaxi.	Implementación de metadatos Dublin Core y protocolo de interoperabilidad para optimización de tiempos de búsqueda de archivos alojados en la plataforma científica.	2020	PRE	[145]	Estudio
TPRE11	Desarrollo de los Módulos de Certificación y Autoevaluación, Investigador, mediante Inferencia de los Datos Extraídos del Proyecto EcuCiencia de la Universidad Técnica de Cotopaxi.	Certificación y autoevaluación de investigadores mediante inferencia de datos extraídos del proyecto EcuCiencia.	2020	PRE	[146]	Implementado
TPRE12	Desarrollo de los Módulos de Gestión de Proyectos y Grupos de Investigación en la Plataforma EcuCiencia.	Administrar la información de grupos y proyectos de investigación de acuerdo con los datos registrados en la plataforma para permite visualizar datos de producción científica mediante una gráfica.	2021	PRE	[147]	Implementado
TPOS01	Método de Análisis de Redes Sociales para Identificar Relaciones y Colaboraciones Científicas entre Investigadores de la Universidad Técnica de Cotopaxi.	Aplicación de la metodología de análisis de redes sociales para la visualización de información y conformación de relaciones entre autores, artículos, libros y ponencias.	2020	POS	[148]	Implementado
TPOS02	Métodos para el análisis de la información en corpus de artículos científicos con algoritmos de clasificación y librerías NLTK en la Plataforma Científica ECUCIENCIA.	Método de análisis de información en corpus de artículos científicos con algoritmos de procesamiento de datos en librerías como NLTK, NUMPY, MATPLOTLIB, PYPD2, SKLEARN y SCIPY de Python.	2020	POS	[149]	Implementado
TPOS03	Sistema de georreferenciación para el análisis de producción científica de los docentes investigadores de la Universidad Técnica de Cotopaxi.	Sistema de georreferenciación para la colaboración de expertos en áreas específicas y visualización de trabajos de investigación en zonas geográficas.	2020	POS	[150]	Implementado
TPOS04	Visualización de Información mediante mapeo auto-organizado en datos de producción científica de la Universidad Técnica de Cotopaxi.	Mapeo auto-organizado para visualización de información. Los datos son entrenados por medio de redes neuronales artificiales de acuerdo a las líneas y sublíneas de investigación mediante mapas de calor distinguiéndose por colores.	2020	POS	[151]	Implementado
TPOS05	Algoritmos de Deep Learning utilizando Tensor Flow para el tratamiento de datos de producción científica en la Universidad Técnica de Cotopaxi	Implementación de algoritmos de aprendizaje profundo apoyándose en Tensor Flow para la toma de decisiones no supervisadas en la plataforma EcuCiencia, los datos que toman relevancia son las Líneas y sublíneas de investigación	2021	POS	[152]	Implementado

Fuente: Elaboración propia.

2.1.1. Metodologías y Modelos de Ingeniería de Software aplicadas en el Proyecto EcuCiencia.

Todo proyecto se desarrolla con base a un ciclo de vida el cual consiste en garantizar que el producto cumpla con los requerimientos de especificación establecidos por el usuario. Permite que se controlen las acciones que se desarrollaran para poder establecer los parámetros más importantes. La Tabla 21 permite señalar las metodologías del Desarrollo del software aplicadas en la plataforma EcuCiencia.

Tabla 21. Metodologías de Desarrollo de Software aplicadas en EcuCiencia.

ID	Metodología	Descripción	Tipo de Aplicación	Referencia
TPRE03	XP	Capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida de un proyecto, es útil para proyectos pequeños donde se garantiza que el software desarrollado funcione de acuerdo a lo requerido, entregando un software de calidad.	Implementado	[138]
	Mobile - D	Utilizada por su simplicidad y por ser una herramienta de apoyo para superar dificultades en el desarrollo de aplicaciones en corto tiempo, su adaptabilidad es óptima en nuevos requerimientos y permite la entrega de productos usables con mayor rapidez con el fin de tener un control del desarrollo y la capacidad de retroalimentar.		
TPRE04, TPRE06, TPRE07, TPRE09, TPRE11, TPOS02, TPOS03	Scrum	Utilizada por que permite tener control y seguimiento de las actividades, permite realizar entregas parciales del proyecto, trabajar colaborativamente. Es utilizado para proyectos con entornos complejos, con requerimientos que se modifican y que es necesario tener resultados rápidos.	Implementado	[139], [141], [142], [144], [146], [149], [150], [57]

Fuente: Elaboración propia.

Los modelos en el desarrollo del software al igual que las metodologías aportan a la correcta utilización de tiempo y recursos, permiten el control de las acciones que se van a desarrollar. La Tabla 22 permite visualizar los modelos de desarrollo de software aplicado en la plataforma EcuCiencia.

Tabla 22. Modelo de Ingeniería de Software aplicadas en EcuCiencia.

ID	Modelo	Descripción	Tipo de Aplicación	Referencia
TPRE05, TPRE10, TPRE12, TPOS01, TPOS04, TPOS05	Iterativo e Incremental	Ayuda a disminuir los riesgos ya que se construye a partir de un diseño preliminar, se puede adaptar a otros entornos y permite obtener resultados funcionales cada periodo de tiempo.	Implementado	[140], [145], [147], [148], [151], [152]
TPRE08	Prototipo	Es construido en poco tiempo y posteriormente evaluado por el cliente para obtener los requisitos que se van a desarrollar.	Implementado	[143]

Fuente: Elaboración propia.

Las etapas de la Ingeniería de software dependen mucho de la metodología o modelo que se utilizó. Por lo cual se considera importante determinar cada una de las etapas identificadas en los proyectos desarrollados en la plataforma EcuCiencia, lo cual se presenta en la Tabla 23.

Tabla 23. Etapas del Desarrollo de Software aplicadas en EcuCiencia.

	FASES/ETAPAS	TP RE 03	TP RE 04	TP RE 05	TP RE 06	TP RE 07	TP RE 08	TP RE 09	TP RE 10	TP RE 11	TP RE 12	TP OS 01	TP OS 02	TP OS 03	TP OS 04	TP OS 05
Móvil	Exploración	✓														
	Inicialización	✓														
	Producción	✓														
	Estabilización	✓														
	Prueba	✓														
Web	Planificación	✓								✓	✓			✓		
	Diseño			✓		✓	✓	✓	✓	✓	✓	✓			✓	✓
	Codificación	✓	✓		✓	✓	✓	✓		✓			✓			
	Verificación/Pruebas	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓	✓
	Priorización y estimación de tiempo		✓		✓						✓		✓			
	Análisis		✓	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓	✓
	Implementación		✓	✓	✓		✓		✓			✓	✓		✓	✓
	Mantenimiento															
	Desarrollo														✓	
	Revisión del sprint														✓	
	Retroalimentación														✓	

Fuente: Elaboración propia.

En cada fase se realizan diferentes actividades las cuales son tareas que se deben cumplir por lo cual también se ha tomado en cuenta mostrar cuales son las que se deben realizar para avanzar de una mejor manera con el desarrollo del software. Se han considerado aquellas que fueron realizadas en la plataforma EcuCiencia lo cual se presenta en la Tabla 24 y 25 respectivamente.

Tabla 24. Actividades de las Fases de Ingeniería de Software aplicadas en EcuCiencia.

ACTIVIDADES	TP RE 03	TP RE 04	TP RE 05	TP RE 06	TP RE 07	TP RE 08	TP RE 09	TP RE 10	TP RE 11	TP RE 12	TP OS 01	TP OS 02	TP OS 03	TP OS 04	TP OS 05
Historias de Usuario	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
Diagramas de caso de uso	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
Diagramas de caso de uso a detalle	✓	✓		✓	✓		✓	✓	✓	✓		✓			
Desarrollo de módulos			✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Establecimiento de los Stakeholder	✓		✓		✓		✓		✓	✓		✓	✓	✓	✓
Definición del alcance del proyecto	✓														
Establecimiento general del proyecto	✓														
Configuración del proyecto	✓														
Planificación inicial	✓														
Día de prueba	✓														

Fuente: Elaboración propia.

Tabla 25. Actividades de las Fases de Ingeniería de Software aplicadas en EcuCiencia (Continuación).

ACTIVIDADES	TP RE 03	TP RE 04	TP RE 05	TP RE 06	TP RE 07	TP RE 08	TP RE 09	TP RE 10	TP RE 11	TP RE 12	TP OS 01	TP OS 02	TP OS 03	TP OS 04	TP OS 05
Día de salida	✓														
Trabajo	✓														
Liberación	✓														
Estabilización	✓														
Pruebas o Casos de prueba	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓		✓	✓	✓
Reparación	✓														
Diagrama de clases		✓		✓	✓				✓	✓	✓			✓	
Diagrama de secuencias		✓	✓	✓			✓	✓	✓	✓	✓			✓	✓
Diagrama de despliegue								✓							
Diagrama de actividades									✓	✓				✓	✓
Diagrama de bloques										✓					
Modelo entidad relación						✓			✓	✓			✓		✓
Diagrama de arquitectura							✓		✓	✓		✓	✓		
Planificación de los sprint's													✓		
Prioridades de las historias de usuario							✓	✓							
Requerimientos de desarrollo	✓		✓		✓										
Análisis de requisitos										✓				✓	
Requerimientos funcionales	✓		✓				✓	✓							✓
Requerimientos no funcionales	✓		✓												✓
Artefactos de Scrum							✓					✓			

Fuente: Elaboración propia.

2.1.2. Aplicación de la Inteligencia Artificial en el Desarrollo del Proyecto EcuCiencia.

Se presentan la metodología de minería de datos, las metodologías, algoritmos de IA y métricas para la evaluación de los modelos aplicados en la plataforma científica. Las cuales se presentan en la Tabla 26, 27, 28 y 29 respectivamente.

Tabla 26. Metodología de Minería de Datos aplicada en EcuCiencia.

ID	Metodología	Descripción	Ref.
TPRE04, TPRE05, TPRE09, TPOS02, TPOS04, TPOS05	KDD	Permite la extracción de conocimiento donde el proceso metodológico puede encontrar un modelo valido, útil y entendible y se describa patrones de acuerdo a la información de grandes bases de datos. Es importante porque combina el descubrimiento y análisis de datos, es decir, extrae patrones en forma de reglas o funciones de datos para el análisis del usuario.	[139], [140], [144], [149], [151], [152]

Fuente: Elaboración propia.

Tabla 27. Metodologías de Inteligencia Artificial aplicada en EcuCiencia.

ID	Metodología	Descripción	Ref.
TPRE07	Lógica Difusa	Ayuda a manejar información de difícil especificación, su aportación como metodología es proporcional de forma simple y elegante de obtener una conclusión a partir de información ambigua, imprecisa, con ruido o incompleta.	[141]
TPRE08	Redes Neuronales	Permite obtener un sistema inteligente para recomendación y conformación de grupos de investigación. Simula el proceso de aprendizaje de una neurona y fue utilizado para el desarrollo del prototipo y las sugerencias de colaboración entre los investigadores.	[142]
TPOS01	Estructural	Ayuda a la extracción del análisis de la información con el fin de estudiar las estructuras científicas de cada uno de los artículos científicos tanto en relaciones internas como externas, es decir, se podrá analizar las redes territoriales como distintas subredes o a su vez como una red en su conjunto.	[148]
	Análisis de Redes Sociales	Ayuda a visualizar información de forma compuesta y como resultado se logra obtener las relaciones entre artículos, libros y ponencias de la plataforma permitiendo mostrar las redes interuniversitarias, es decir, conocer cuando se hacen publicaciones de artículos de coautores de la misma o de diferentes universidades.	
TPOS04	Mapeo Auto-organizado	Permite clasificar y visualizar información con la ayuda de redes neuronales artificiales. Se aplica en el procesamiento de la producción científica de docentes investigadores acorde a las líneas y sublíneas de investigación	[151]

Fuente: Elaboración propia.

Tabla 28. Algoritmos de Inteligencia Artificial aplicadas en EcuCiencia.

ALGORITMOS	TP RE 01	TP RE 02	TP RE 04	TP RE 05	TP RE 06	TP RE 07	TP RE 08	TP RE 09	TP OS 01	TP OS 02	TP OS 04	TP OS 05
Filtrado basado en reglas	✓											
Filtrado colaborativo	✓	✓										
Filtrado basado en enfoques mixtos	✓											
Filtrado basado en contenidos	✓											
Basado en memoria		✓										
Basado en modelos		✓										
Máquinas de soporte vectorial (SVM)			✓									
RandomForest			✓									
Classifier			✓					✓				
LinearSVC			✓									
MultinomialNB			✓									
LogisticRegression			✓									
AgglomerativeClustering				✓								
SpectralClustering				✓								
K-Means	✓			✓								
Naïve Bayes								✓				
De ponderación y evaluación					✓							
Razonamiento difuso						✓						
Redes Neuronales			✓				✓				✓	✓
SkLearn												✓
NLTK									✓			✓
Algoritmos Genéticos											✓	
Deep Learning												✓
MeanShift										✓		✓

Fuente: Elaboración propia.

Tabla 29. Métricas para evaluar algoritmos de Inteligencia Artificial aplicadas en EcuCiencia.

MÉTRICAS	TPRE01	TPPE09	TPOS02	TPOS05
Distancia euclidiana			✓	
Chebyshev		✓	✓	
Índice de Jaccard	✓	✓	✓	
Similitud de Pearson	✓		✓	
Correlación		✓		
Índice Dice		✓		
Cross-Validation				✓

Fuente: Elaboración propia.

De toda la información recabada se puede presenciar los resultados en el proyecto REDEC o también conocido como EcuCiencia. El cual se encuentra en esta dirección <http://ecuciencia.utc.edu.ec/>.

2.1.3. Resultados de la Integración de la Ingeniería de Software y la Inteligencia Artificial en el Proyecto EcuCiencia.

Como resultado de la aplicación de las metodologías, fases y actividades de la ingeniería de software y la inteligencia artificial se obtiene una plataforma inteligente para la administración de la producción científica y la conformación de grupos de investigación. Para el análisis de la producción científica se establece el módulo de Cienciometría que permite medir el impacto de la producción científica, mapeo de campos científicos y producción de indicadores, comparación de productividad institucional, perfiles de autores y rendimiento de las investigaciones. Aplicando Django, Python y PostgreSQL además del modelo Iterativo Incremental se obtiene un sistema de indicadores respecto a la publicación de libros, capítulos de libros, artículos y registro de ponencias nacionales e internacionales, lo cual se presenta en la Figura 19.

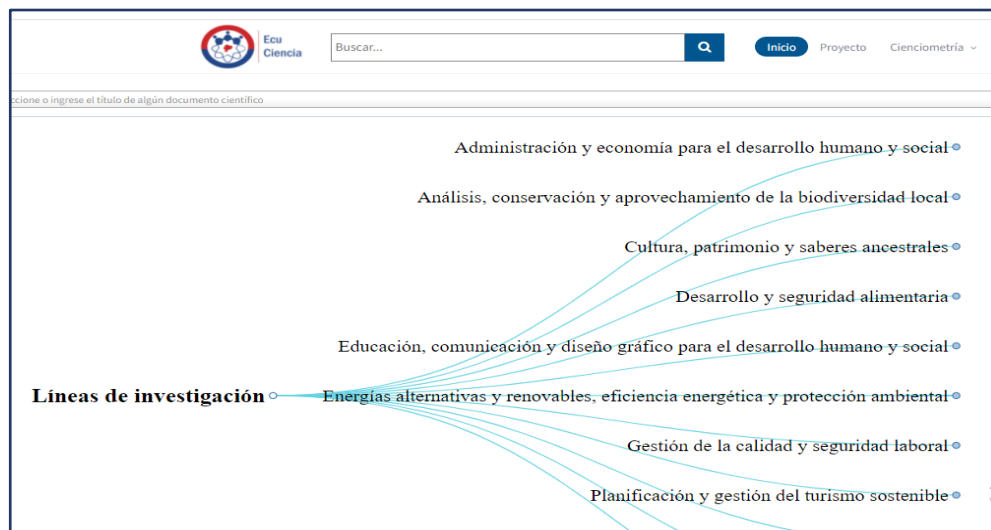
Figura 19. Estadísticas de producción científica por año.



Fuente: Tomado de Plataforma EcuCiencia.

Para la visualización por categorías de la producción científica y su relación con las líneas de investigación registradas en la plataforma se aplicó la metodología Knowledge Discovery in DataBases (KDD), aplicando técnicas de minería de datos se prepara texto con la información suministrada por los investigadores respecto al título, resumen y palabras claves como fuentes de datos. La aplicación de técnicas de machine learning como Máquinas de Soporte Vectorial se puede obtener un modelo capaz de clasificar y predecir las categorías a las que pertenecen los documentos registros mediante un proceso de selección de características, resultados que se presentan en la Figura 20.

Figura 20. Clasificación de Líneas de Investigación mediante Máquinas de Soporte Vectorial.



Fuente: Tomado de Plataforma EcuCiencia.

La aplicación de aprendizaje no supervisado fue utilizado para realizar procesos de clasificación en función de medidas de similitud aplicando técnicas de aprendizaje automático, se establecen como puntos de análisis gráficas de similitud, determinación de distancia entre investigadores, agrupación de resultados, análisis de productividad e indicadores de producción científica de investigadores. Para el desarrollo del sistema se utiliza el modelo interactivo incremental, minería de datos para integración, limpieza y pre procesado de la data, así como también, algoritmos de aprendizaje automático como MeanShift, SpectralClustering, K-means que fue utilizado por ser un método interactivo de clasificación y segmentación de clúster generados a partir palabras claves de los artículos científicos. Se puede señalar también, el uso del algoritmo AgglomerativeClustering utilizado para el desarrollo del dendograma para la clasificación jerárquica y la representación de niveles de distancia y similitud.

En virtud de que una de las finalidades de la plataforma EcuCiencia es la integración de investigaciones de acuerdo a sus áreas de interés científica se consideró importante establecer

la red social de investigación para el fortalecimiento del perfil científico de sus usuarios y crear vínculos que permitan contactar, socializar y conectar con profesionales a fines a áreas del conocimiento y áreas de investigación científica. Para el desarrollo del módulo red social aplicó el modelo interactivo incremental, mediante el uso de la norma IEEE 830-1998 se levantaron los requerimientos del sistema y su desarrollo se basó en el Modelo Vista Controlador utilizando el lenguaje de programación de Python. El IDE PyCharm permitió en análisis de redes sociales (ARS) y el uso de librerías relacionadas con el análisis de datos. Se aplica la librería LNTK utilizada para el procesamiento de lenguaje natural para la obtención de métricas sobre la información y la visualización de patrones de conducta que se relacionan entre sí para la conformación de lazos sociales y procesos de estructuración más amplios.

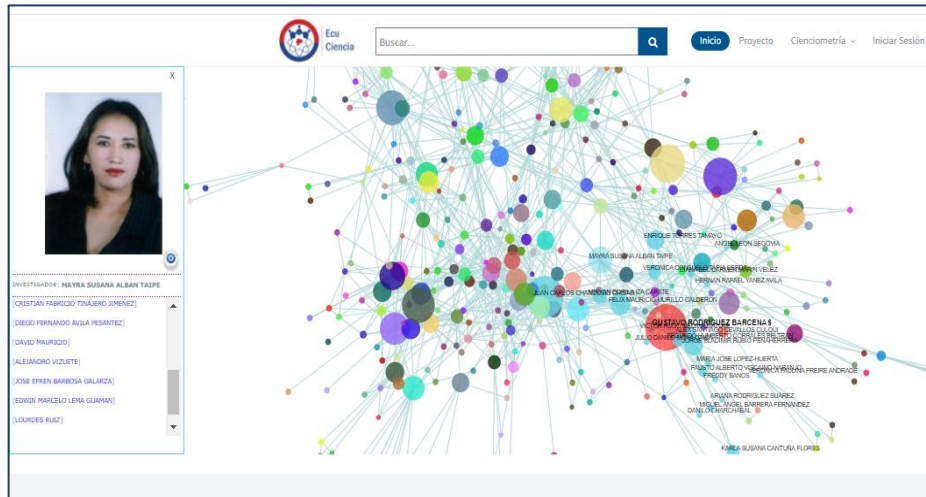
La composición de la red social se encuentra establecida por métodos que permiten encontrar y estudiar subconjuntos de actores homogéneos y los lazos científicos existentes entre ellos, en donde los lazos se convierten en canales que transfieren y fluyen recursos, como resultado de su aplicación se obtienen grafos y subgrafos que se presentan en conexiones claves establecidos entre la producción científica y los investigadores. Los modelos de red aplicados permiten establecer los niveles de integración y la contextualización de las estructuras científicas como patrones de relaciones, el comportamiento de las redes se basan través de las interrelaciones locales que se encuentren en la información procesada en los artículos, libros y ponencias, el resultado de la aplicación de estas tecnologías informáticas se presenta en la Figura 21 y 22.

Figura 21. Aplicación de algoritmos de agrupación para determinación de Similaridad.



Fuente: Tomado de Plataforma EcuCiencia.

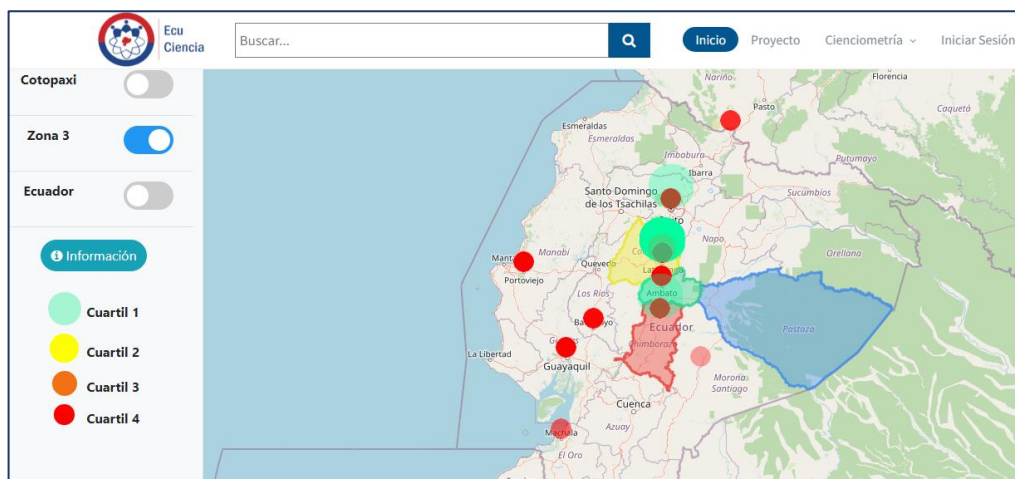
Figura 22. Red de colaboración mediante medidas de similitud.



Fuente: Tomado de Plataforma EcuCiencia.

El desarrollo del módulo de georreferenciación hace uso de tecnologías web con el objetivo de delimitar zonas geográficas de acuerdo con la producción e impacto científico, mediante el uso de herramientas de desarrollo ágil como javascript, css3, jquery, html5, json, entre otros. El estilo de la arquitectura del software aplica la modelo vista controlador para separar las capas de los datos y la lógica de negocio, la arquitectura se divide en tres bloques determinados para describir cómo trabaja la plataforma, establecer los investigadores que acceden al sistema y la ubicación del servidor de datos. Se aplica el método de verificación y validación para cotejar el cumplimiento de la especificación de requerimientos de software en contraste con las implementaciones funcionales. En el entorno de desarrollo se utilizó la herramienta de programación multiplataforma y la interfaz de aplicación de Google Maps para la traficación del mapa que se visualiza en la plataforma, resultados que se presentan en la Figura 23.

Figura 23. Aplicación del módulo de Georreferenciación.



Fuente: Tomado de Plataforma EcuCiencia.

De igual manera, la plataforma cuenta con un módulo de pre procesamiento de datos que trabaja bajo metodologías y técnicas de minería de texto para el análisis del corpus, distancias, similitudes y generación de dominios de conocimiento sobre la base de la riqueza textual. Se aplica un procedimiento algorítmico mediante técnicas del procesamiento del lenguaje natural a través de las librerías NLTK, así como también el uso de medidas de similitud como la distancia Euclidiana, la distancia de Minkowski, el índice de Jaccard, distancia de correlación, escalamiento multidimensional y la construcción de la nube de palabras que se forma de acuerdo a la frecuencia de aparición de las mismas. Este módulo también permite la generación de corpus a partir de determinar la riqueza léxica, frecuencia de palabras, palabras de parada, similitud y distancia de texto, con base a analizar no solo el resumen, título y palabras claves si no también el contenido del documento.

Figura 24. Procesamiento de datos.



Fuente: Tomado de Plataforma EcuCiencia.

Otra de las opciones de uso de la plataforma científica es el sistema de recomendación para grupos de investigación, de manera similar en basado en los perfiles de usuarios, áreas de conocimiento y características de similitud. La arquitectura que se utilizo es Modelo Vista Template escrita en Django, se hizo uso también de una Api Rest para ser intermediario entre la aplicación móvil y la base de datos. Por último, se utilizó el sistema de aprendizaje automático Tensorflow para construir y entrenar la red neuronal que permito dar sugerencias de colaboración entre los investigadores. Como resultado se obtiene un sistema de inteligencia artificial capaz de relacionar datos en red simultáneamente, de fácil uso que está conformada por la sección de grupos en los cuales el investigador está trabajando, se puede visualizar una sección de personas con afinidad al campo profesional con los cuales se puede realizar una amistad y a futuro una colaboración, publicaciones de los diferentes autores a los cuales este siguiendo, y por último la sección de chat que permite establecer y mantener comunicación en

línea para el desarrollo del trabajo, incluye también una lista de sugerencias de personas a las que se les puede agregar como amigos y a su vez se muestra las solicitudes pendiente que pueden ser aceptadas o eliminadas.

Figura 25. Ventana de solicitudes de recomendación.



Fuente: Tomado de Plataforma EcuCiencia.

CAPÍTULO III

3. MODELO PARA DETERMINACIÓN DE FACTORES DE ÉXITO EN LA INTEGRACIÓN DE LA INTELIGENCIA ARTIFICIAL CON EL DESARROLLO DE SOFTWARE.

3.1. ANTECEDENTES

3.1.1. Antecedentes sobre Fases de Ingeniería de Software con Inteligencia Artificial

Para la obtención del modelo conceptual de integración de la Inteligencia Artificial con el Desarrollo de Software se establece como antecedentes algunos trabajos encontrados en la revisión de la literatura que permiten el enlace de estas dos áreas de la ciencia y que han sido utilizadas para determinar productos de desarrollo utilizando la Inteligencia Artificial.

El proyecto de Egas & Roque [153] en donde se establece el diseño de un modelo predictivo basado en técnicas de Machine Learning que permita determinar la temperatura usando los datos de una mini estación meteorológica en la ciudad de Guayaquil. Este proyecto tiene como objetivo observar, obtener y almacenar los distintos datos de temperatura en una mini estación meteorológica con sensores, debido a problemáticas en las variaciones de temperatura. En su desarrollo se emplea el modelo en cascada y sus fases se establece el análisis, diseño, codificación y pruebas. Se utilizó técnicas de Inteligencias Artificial y aprendizaje automático como Machine Learning, mediante la arquitectura de una red neuronal recurrente (LSTM). La validación de resultados permite visualizar un porcentaje del 80% como tasa de precisión del algoritmo que se utilizó para entrenamiento del modelo predictivo, dando como tasa de fallo de LSTM en un 1,43%.

De igual manera Plaza [154] en su trabajo denominado desarrollo de un sistema web para el soporte técnico remoto de primer nivel, orientado a la gestión de incidentes informáticos, basado en Inteligencia Artificial ofrece un producto que contribuya a la resolución de incidentes técnicos, para el desarrollo de este proyecto se utilizó el modelo de desarrollo en cascada, con las siguientes fases: definición de requisitos, diseño del sistema y del software, implementación y pruebas de unidades, integración y prueba del sistema, funcionamiento y mantenimiento. Dentro de la Inteligencia Artificial se usó la API IA de Google, que permite desarrollar tecnologías de interacción hombre-computador, basado en conversaciones de lenguaje natural. El resultado fue la integración de un chatbot el cual está disponible las 24 horas del día, 7 días a la semana recibiendo solicitudes sobre el servicio técnico.

Pacheco & Laguado [155] en su proyecto prototipo de herramienta software para la generación de un pronóstico de inventario mediante algoritmos de Inteligencia Artificial: Caso de estudio producción de cacao, da a conocer los bajos índices de productividad que se producen por la deficiente gestión de los recursos, baja capacitación del personal y débil implementación tecnológica los cuales han sido las causas para la disminución de la productividad empresarial en Colombia. El modelo de desarrollo fue en cascada y sus fases fueron: especificación de requisitos, planeación, modelado, construcción, prueba, apoyo o mantenimiento. Para implementar Inteligencia Artificial se hizo uso de la técnica de Red Neuronal, que se implementó para el pronóstico de demanda, con el uso de la librería Brain.js y el modelo LSTM que se aplica para la predicción de series de tiempo. Como resultado se dio el mejoramiento de la planificación de los procesos de producción, al desarrollar un prototipo para gestionar y optimizar la producción del valor agregado, para evaluar y predecir el comportamiento de la futura demanda.

Se puede señalar a Limaico [156] con su proyecto de desarrollo de un sistema recomendador de ubicaciones para sitios turísticos del centro histórico de Quito basado en Procesamiento de Lenguaje Natural que tiene como fin dar una solución para el mejoramiento de recomendaciones de sistemas, a través del procesamiento de lenguaje natural, logrando entrenar un modelo con el uso de aprendizaje profundo. Dentro de las metodologías se implementó SCRUM, por su enfoque ágil, es beneficioso y sus fases de desarrollo son: recolección de datos, de desarrollo, sprint 1, 2, 3 y resultados. La técnica donde se integra PLN, estas redes neuronales (Deep Learning). Con la aplicación del algoritmo PLN, se hará la construcción de un sistema de recomendaciones de sitios turísticos para distintos usuarios. Como resultado PLN fue entrenado con datos que se obtuvieron en encuestas con 685 registros, logrando obtener un total de 62 categorías o clases.

Por su parte Mora [157] en su proyecto diseño e implementación de un modelo software basado en técnicas de inteligencia artificial, para predecir el índice de radiación solar en Riobamba-Ecuador explica que el modelo pretende servir de predicción local o puede ser utilizado como línea base por el Ministerio de Salud Pública para prevenir el cáncer de piel en la ciudad de Riobamba. Su desarrollo se hizo con la metodología de desarrollo Proceso Unificado de Desarrollo Software (PUD) con las fases de: inicio, elaboración, construcción y transición. Para aplicar inteligencia artificial se usó la técnica para la predicción del índice ultravioleta (IUV) siendo las ANN Backpropagation y las máquinas de vectores de soporte (SVM), las que permitieron construir el modelo. Como resultado se obtuvo una predicción que permitirá tomar

decisiones con el fin de mejorar la calidad de vida de las personas, evitando el cáncer de piel y otras enfermedades.

Finalmente Camacho & Gualotuña [158] con su proyecto desarrollo de una aplicación para evaluar el calibre de arterias y venas a partir de retinografías, aplicando técnicas de inteligencia artificial propone desarrollar una aplicación que permita detección temprana de la Retinopatía Diabética, mediante el análisis y procesamiento de retinografías respecto al calibre de arterias y venas de la retina ocular. Para su desarrollo se hizo uso de la metodología de Prototipado Rápido con las siguientes fases: análisis preliminar, análisis de tareas, desarrollo del prototipo, estudio de casos, implantación y mantenimiento. En lo que respecta a Inteligencia Artificial se aplicaron técnicas de procesamiento de imágenes para el procesamiento de retinografías y para el síndrome metabólico se aplicaron reglas de lógica difusa, también se hizo uso de visión por computador y aprendizaje automático. Los resultados fueron satisfactorios lo cual muestra reportes y recomendaciones junto con los resultados del análisis entre las características de la imagen y el síndrome metabólico.

Por otro lado, se establecen una descripción de proyectos de desarrollo de software y un análisis de la implementación de los algoritmos de Inteligencia Artificial establecida en múltiples propósitos y áreas de la ciencia que han sido diseñados y ejecutados para resolver problemas en la sociedad. Los cuales se presentan en la Tabla 30 y 31 respectivamente.

Tabla 30. Proyectos de desarrollo de software con implementación de inteligencia artificial.

ARTÍCULO	DESCRIPCIÓN	REF.	MODELO	FASES	TÉCNICA	ALGORITMO
Diseño de un modelo predictivo basado en técnicas de machine learning que permita determinar la temperatura usando los datos de una mini estación meteorológica en la ciudad de Guayaquil.	Las variaciones de temperatura han provocado cambios lo que produce pérdidas económicas tanto en agricultura como en otros ámbitos de carácter social. Por lo cual la construcción de este modelo ayudará a determinar la temperatura mediante el uso de datos proporcionados por la entidad en la cual se va a trabajar y de esta manera disminuir los problemas.	[153]	Cascada	Análisis Diseño y Codificación Prueba	Machine Learning	Red Neuronal Recurrente (LSTM)
Desarrollo de un sistema web para el soporte técnico remoto de primer nivel, orientado a la gestión de incidentes informáticos, basado en inteligencia artificial	Su objetivo es diseñar un aplicativo web para gestionar los incidentes informáticos de primer nivel por medio de chatbots, agentes en los cuales ingresa conocimiento para la resolución de algún tipo de actividad disminuyendo el tiempo de respuesta y asegurando la disponibilidad del servicio. Utiliza inteligencia artificial como soporte técnico para gestionar dicho valor aportando valor a la empresa.	[154]	Cascada	Definición de requisitos, Diseño del sistema y del software Implementación y pruebas de unidades Integración y prueba del sistema Funcionamiento y mantenimiento	API IA (Google)	
Prototipo de herramienta software para la generación de un pronóstico de inventario mediante algoritmos de inteligencia artificial: caso de estudio producción de cacao	Con la propuesta se pretende mejorar la planificación de los procesos de producción, al desarrollar un prototipo permitirá a los usuarios gestionar y optimizar la producción del valor agregado, para evaluar y predecir el comportamiento de la futura demanda.	[155]	Cascada	Especificación de los requerimientos Planeación Modelo Construcción Prueba Apoyo o mantenimiento	Red Neuronal	Red Neuronal Recurrente (LSTM)
Desarrollo de un sistema recomendador de ubicaciones para sitios turísticos del centro histórico de Quito basado en Procesamiento de Lenguaje Natural.	Proyecto recomendado de sistemas por medio del procesamiento del lenguaje natural, desarrollado con metodología SCRUM, con la aplicación de IA, mediante el algoritmo PLN, que aportará a la construcción de un sistema recomendador de sitios turísticos.	[156]	SCRUM	Fase de recolección de datos Fase de desarrollo Sprint 1 Sprint 2 Sprint 3 Resultados	Redes Neuronales	Procesamiento de Lenguaje Natural (PLN)

Fuente: Elaboración propia.

Tabla 31. Proyectos de desarrollo de software con implementación de inteligencia artificial (Continuación)

ARTÍCULO	DESCRIPCIÓN	REF.	MODELO	FASES	TÉCNICA	ALGORITMO
Diseño e implementación de un modelo software basado en técnicas de inteligencia artificial, para predecir el índice de radiación solar en Riobamba-Ecuador	Se desarrolló un modelo de software que muestra información en tiempo real de la irradiación solar en la ciudad de Riobamba y las personas que poseen cáncer de piel pueden hacer uso de ello frecuentemente es decir puede ser utilizado como un aporte local o puede ser usado por el Ministerio de Salud para prevenir el cáncer de piel. Permitirá tomar decisiones con el fin de mejorar la calidad de vida de las personas	[157]	Proceso Unificado de Desarrollo Software (PUD)	Inicio Elaboración Construcción Transición	Redes Neuronales Máquinas de Vectores de Soporte (SVM) Perceptron Multicapa ANN	MADALINE NN Backpropagation
Desarrollo de una aplicación para evaluar el calibre de arterias y venas a partir de retinografías, aplicando técnicas de inteligencia artificial	La aplicación realiza análisis y procesamiento de retinografías respecto al calibre de arterias y venas de la retina ocular correlacionado con datos del síndrome metabólico del paciente. Los resultados fueron satisfactorios para los médicos especialistas, los reportes coincidían con los diagnósticos.	[158]	Prototipado Rápido	Análisis preliminar Análisis de tareas Desarrollo del prototipo Estudio de casos de prueba Implantación Mantenimiento	Lógica Difusa Procesamiento de imágenes	Canny
Prototipo de un chatbot para un sistema de servicios vehiculares utilizando Adaptive Cards con Bot Framework	La aplicación tiene por objeto generar un sistema de chatbot el cual inicia un diálogo cuando detecta el ingreso de algún texto por parte del usuario. Esta aplicación también permite ingresar, listar y agregar vehículos existentes, además permite listar los servicios vehiculares, guardar la orden y visualizar la orden de compra de los servicios vehiculares.	[159]	XP (Extreme Programming)	Planificación del proyecto Diseño Codificación Pruebas	Tecnología LUIS (Language Understanding Intelligent Service) de Azure	
Desarrollo de un algoritmo híbrido para recomendaciones de itinerarios turísticos de acuerdo con las preferencias de los usuarios	TTPD por su traducción al español (Problema de diseño de viaje turístico), tiene como fin mostrar al turista un diseño de distintas rutas con puntos que sean de interés y que sea de preferencia al turista en una determinada ciudad que visite. Implementar dos algoritmos de Inteligencia Artificial para el procesamiento de datos. k-Means y algoritmo genético.	[160]	Scrum	Arquitectura Definición de roles Historias épicas Product Backlog Definición de Sprints	Clustering	K-Means Algoritmos genéticos

Fuente: Elaboración propia.

3.1.2. Antecedentes del uso de Metodologías Ágiles (Scrum, Xp Y Kanban)

Tabla 32. Metodología Ágil Scrum.

ARTÍCULO	DESCRIPCIÓN	ROLES	ARTEFACTOS	FASES	ACTIVIDADES	REF.
Scrum: un método ágil para sus proyectos	El año 1995 se puede considerar el año oficial de nacimiento de Scrum. En ese año Ken Schwaber describe los fundamentos del método. Scrum se aplica en empresas y por lo mismo es el método ágil más utilizado en la actualidad. El método Scrum se ilustra con respecto a los valores y principios del manifiesto ágil como la existencia de un equipo auto-organizado, roles definidos, conjunto de ceremonias, constancia del cliente, mecanismos que favorecen las entregas y una aceptación al cambio.	Scrum Master	No describe artefactos	El Product Owner redacta	Las User Stories y las sitúa en el Product Backlog.	[161]
		Product Owner (el cliente)		Product Owner prioriza	Las User Stories y ordena el Product Backlog en consecuencia.	
				El equipo Scrum establece la lista de Users Stories	Lo cuales se tratarán durante el Sprint	
		Equipo de desarrollo		Inicia el Sprint	Se descomponen en tareas por el equipo de desarrollo.	
				Reuniones Diarias	Con una iteración de 2, 3 o 4 semanas.	
				Entrega del Sprint	No describe actividades	
Los Fundamentos de Agile Scrum	Scrum no es una metodología es un marco de trabajo, es decir algo más sofisticado que da aporte a las soluciones complicadas, se debe simplificar si se quiere usar para proyectos más sencillos.	Dueño del producto (Product Owner)	Backlog de producto	Planificación del Sprint	No describe actividades	[162]
		Scrum Master	Sprints	Scrum diario		
		Equipo de desarrollo (Development Team)		Revisión del Sprint		
Otros roles (diseñador, tester, inspector de calidad y jefe de equipo)	Incremento	Retrospectiva del Sprint				

Fuente: Elaboración propia.

Tabla 33. Metodología Ágil Scrum (Continuación).

ARTÍCULO	DESCRIPCIÓN	ROLES	ARTEFACTOS	FASES	ACTIVIDADES	REF.
Propuesta de un método para desarrollar Sistemas de Información Geográfica a partir de la metodología de desarrollo ágil - SCRUM	Esta metodología se planteó a mediados de los 80 y se implementó con éxito en los 90. Este marco de trabajo provee herramientas necesarias para el desarrollo de software de una forma ágil. Su eje central es la satisfacción de los requisitos funcionales y no en el proceso La documentación sobre esta metodología de desarrollo no es muchas, porque se concentra en el desarrollo de un producto funcional y no en el proceso.	Product Owner	Prodcut Backlog	Sprint	Realiza las especificaciones del sistema, y se realiza en un tiempo menor a cuatro semanas	[58]
				Sprint Planning Meeting	Se planifica las actividades del próximo sprint	
		Development Team	Sprint Backlog	Daily Scrum	Reunión máxima de 15 minutos, sincroniza las actividades del equipo de desarrollo	
				Sprint Review	Se revisan las actividades de los sprints	
Scrum Master	Increment	Sprint Retrospective	Reunión máxima de 3 horas, se analiza el comportamiento e identifican los elementos exitosos.			
Resultados preliminares más significativos tras cuatro años de aplicación de la metodología SCRUM en las prácticas de laboratorio	Scrum es una metodología ágil de desarrollo de software iterativa e incremental, el equipo de trabajadores trabaja con una unidad autoorganizada para alcanzar el objetivo el cual es un software diseñado a partir de una serie de características.	No describe roles	Backlog del producto	Creación del equipo		[163]
				Planificación del Sprint	Se define la duración, el tamaño y velocidad ideal del sprint	
			Sprints	Desarrollo del Sprint	Reuniones para el desarrollo del trabajo	
				Fin del Sprint	Se verifica la calidad y valoración del sprint	

Fuente: Elaboración propia.

Tabla 34. Metodología Ágil Extreme Programming (XP).

ARTÍCULO	DESCRIPCIÓN	FASES	ACTIVIDADES	REF.
Metodología ágil de desarrollo de software Programación Extreme (XP).	Metodología que fue basada en prueba y error con el fin de conseguir un software funcional, adicional es ligera para el desarrollo de aplicaciones, se basa en simplicidad, comunicación y realimentación del código desarrollado. Metodología óptima para los cambios de requisitos, la integración de grupo va desde 2 hasta 12 personas. El cliente es quien participa activamente en el proceso de desarrollo.	Planeación	Historias de usuario	[164]
			Plan de Estrategias (Release Plan)	
			Plan de Iteraciones (Iteration Plan)	
			Reuniones Diarias de Seguimiento (Stand Up Meeting)	
		Diseño	Diagrama de Clases	
			Bosquejo	
		Codificación	Disponibilidad del cliente	
			Uso de estándares	
			Programación por pares	
			Propiedad Colectiva del Código	
Ritmo Sostenido				
Pruebas	Pruebas Unitarias			
	Detección y Corrección de Errores			
	Pruebas de Aceptación			
Metodología de Desarrollo Ágil en Programación Extrema / Scrum versus XP: similitudes y diferencias	Es una metodología ágil y su enfoque está en potenciar el trabajo en equipo, dando énfasis en el aprendizaje de los desarrolladores y proporcionando un buen ambiente laboral. Dicha metodología es importante en los requisitos imprecisos y cambiantes. Otros puntos a considerar en la implementación de esta metodología son: la comunicación frecuente entre cliente y programadores, se realizan entregas pequeñas, el sistema es definido mediante historias entre cliente y desarrolladores donde describe cómo debe funcionar el sistema, debe de tener un diseño muy simple, las pruebas unitarias se deben ejecutar constantemente ante una modificación del sistema, la programación es en parejas y cualquier programador puede realizar cambios en ciertas partes del código en cualquier momento, el trabajo a realizar debe ser un máximo de 40 horas y el cliente debe permanecer presente y disponible todo el tiempo con el equipo.	Exploración	Se define la especificación general del producto con funcionalidades y prioridades.	[165]
		Planificación de la entrega (release)	Establecimiento de fechas, finalización y entrega.	
		Iteraciones	División de iteraciones aproximada de 3 semanas y seleccionar historias de usuario al comienzo definidas en el Release Planning.	
		Producción	Codificación de las historias de usuarios desarrolladas anteriormente, la programación se realiza bajo estándares para un código consistente, comprensible y escalable. Programación en pareja.	
		Mantenimiento	Uso de test para corroborar el correcto funcionamiento del proyecto. Cliente y desarrolladores realizan una nueva entrega, y se registra en fase de exploración	
		Muerte del Proyecto	Culminación del proyecto.	

Fuente: Elaboración propia.

Tabla 35. Metodología Ágil Extreme Programming (XP) (Continuación).

ARTÍCULO	DESCRIPCIÓN	FASES	ACTIVIDADES	REF.
Caso práctico de la metodología ágil XP al desarrollo de software.	Esta metodología surgió en la década de los 90, por su creador Kent Beck. En el presente trabajo nos manifiesta que XP más que una metodología, se la considera como una disciplina, que contiene valores y principios de metodologías ágiles. Para la implementación de XP se considera los siguientes aspectos: interacción constante con el cliente, que es considerado como parte del equipo de desarrollo, hace uso de recursos para la realización de actividades complejas, cambios constantes en los requerimientos, jornadas de trabajo deben ser al máximo, se realizan entregas pequeñas, grupos de trabajo no superan las 20 personas. Para la elección de la metodología se considera primordialmente en los cambios continuos en el transcurso de un proyecto.	Planeación	Historias de usuario	[166]
			Velocidad del proyecto	
			Iteraciones	
			Entregas Pequeñas	
			Reuniones	
			Roles XP	
			Traslado del personal	
			Ajustar XP	
		Diseño	Simplicidad en el diseño	
			Metáfora del sistema	
			Tarjetas de clase, responsabilidad, colaboración (CRC cards)	
			Soluciones puntuales (Spike Solution)	
		Codificación	Refactorización (Refactoring)	
			Cliente siempre presente	
			Codificar primero la prueba	
Programación en parejas				
Integración secuencial				
Pruebas	Integraciones frecuentes			
	Estándares y propiedad colectiva del código			
Revisión de metodologías ágiles para el desarrollo de software.	Esta metodología pretende dirigir a equipos pequeños o medianos de desarrollo de software, es decir, entre 2 a 10 desarrolladores. Se pone un enfoque a XP en los ambientes de requerimientos que pueden ser imprecisos y cambiantes. XP se guía en valores como la retroalimentación rápida, asumir simplicidad, cambio incremental, aceptación del cambio y trabajo de calidad. Dentro de sus prácticas en esta metodología se encuentran el planning game, pequeñas entregas, diseño sencillo, programación en pareja, pruebas, refactorización, integración continua, propiedad común del código, paso sostenible, cliente en sitio, metáfora y estándares de código.	Exploración	Historias de usuario	[167]
		Planeación	Cliente y programadores definen historias de usuario a implementar y fechas	
		Iteraciones	Calendario acordado con el cliente e iteraciones, y cada una representa HU implementadas y probadas	
		Producción	Afina el funcionamiento del programa y se despliega	
		Mantenimiento	Realiza mejoras y arreglos, o implementando nuevas funcionalidades.	

Fuente: Elaboración propia.

Tabla 36. Metodología Ágil Extreme Programming (XP) (Continuación).

ARTÍCULO	DESCRIPCIÓN	FASES	ACTIVIDADES	REF.
Metodologías ágiles ¿Cómo desarrollo utilizando XP?	Dentro de esta metodología el autor nos manifiesta que el cliente forma parte clave del equipo de desarrollo y esa es la clave del éxito. En concordancia con el resto de autores se hace nuevamente hincapié, que dicha metodología está destinada hacia equipos pequeños o medianos de 2 a 10 miembros y los requerimientos pueden ser cambiantes, dando como ventaja la aplicación de la metodología ya antes mencionada. El proyecto inicia con una fase de exploración, asentando las bases para que el proyecto sea exitoso en su desarrollo, la planificación se involucra al cliente y desarrolladores, las iteraciones no se concluyen hasta que se tiene una primera versión del sistema o primera entrega, seguido a producción y posterior a ello el mantenimiento donde se incurre a nuevas funcionalidades y que se mantenga el sistema funcionando.	Exploración	Cliente y desarrolladores realizan la metáfora.	[168]
			El cliente realiza o redacta las historias de usuario.	
			Los desarrolladores realizan la división de historias de usuario en tareas de programación, calculando los puntos estimados de cada historia.	
			Los desarrolladores analizan las tecnologías a usar.	
			Los desarrolladores construyen uno o varios prototipos del sistema.	
			Los desarrolladores realizan propuesta inicial de la arquitectura del sistema.	
			El cliente realiza la redacción de los casos de prueba ayudado por el encargado de pruebas.	
		Planificación	El cliente prioriza las historias de usuario.	
			Los desarrolladores realizan la estimación del esfuerzo total.	
			Los desarrolladores realizan la estimación en la velocidad del equipo.	
			Cliente y desarrolladores realizan plan de entrega.	
		Iteraciones	Actualizar documentación.	
			Cliente reasigna prioridad a las historias de usuario.	
			Cliente y desarrolladores realizan el plan de iteración.	
			Los desarrolladores definen el diseño preliminar.	
			Cliente y desarrolladores realizan un mejoramiento de los casos de prueba.	
			Implementación por parte de los programadores.	
			Los desarrolladores realizan mejoras al diseño.	
		Cliente y desarrolladores completan y actualizan el plan de iteración.		
		Producción	Actualizar documentación.	
			Completar y actualizar el plan de entrega.	
			Cliente y desarrolladores realizan reuniones diariamente para retroalimentar con las labores que realizan cada uno.	
		Mantenimiento	Cliente y desarrolladores elaboran continuas pruebas al sistema.	
Cliente y desarrolladores realizan una nueva entrega, y se registra en fase de exploración.				
Muerte	Muerte del proyecto			

Fuente: Elaboración propia.

Tabla 37. Metodología Ágil KANBAN.

ARTÍCULO	DESCRIPCIÓN	FASES	ESTADO	ACTIVIDADES	REF.
El Kanban	Es un sistema de gestión en el cual se produce la cantidad de trabajo que el sistema es capaz de asumir. Sirve para asegurar la gestión continua y sin sobrecargas en el equipo. Además, es una aproximación a la introducción de cambios en el ciclo de vida.	Definición	En curso Lista	Definir las tareas que se van a desarrollar	[169]
		Análisis	En curso Lista	Analizar la forma en la que se van a realizar las actividades o tareas.	
		Desarrollo	En curso Lista	Programar la funcionalidad	
		Revisión de calidad	En curso Lista	Testear la funcionalidad es decir realizar las pruebas	
		Puesta en marcha	En curso Lista	El cliente recibe su petición con éxito.	
Identificando el ámbito de aplicación de Lean IT, Scrum y Kanban	Metodología Visual japonesa que se basa en tarjetas visuales derivado del sistema Just-in-Time de Toyota en los años 40. Administra los flujos de trabajo mediante el estado distribución y progreso de las actividades ayudan a evitar los cuellos de botella, repetición de tareas y mejorando los flujos de trabajo.	Visualizar el flujo del trabajo		Determinar lo pendiente	[170]
		Limitar el trabajo en curso	Stop starting Start finishing	Implementar pequeños y continuos cambios incrementales	
		Optimizar		Identificar los indicadores básicos para mejorar los módulos	
		Mejora continua		Seguimiento y control de la funcionalidad del sistema	
Kanban y Scrum – obteniendo lo mejor de ambos	Visualiza el flujo de trabajo, divide el trabajo en bloques, se escribe cada elemento en una tarjeta y se pone en el muro. Se utiliza una columna con nombres para visualizar el flujo de trabajo de cada elemento.	Pila	Pendiente	Límite el WIP. Se asigna los elementos que pueden estar en proceso en cada flujo	[171]
		Seleccionar			
		Desarrollar	En curso	Mide el lead time. Optimiza el proceso para que sea pequeño y predecible.	
		Implementar			
		Producir	Terminado	No se define actividades	
Organización ágil del trabajo: Parte II - Dos dimensiones del trabajo	Kanban es la técnica por excelencia para observar el estado del trabajo y que fluya normalmente. Está técnica de Taiichi Onho, se desarrolló en los años 50's. Kanban es eficaz, debe estar bien diseñado y deben estar claras las reglas del juego.	Introducir		No se define actividades	[172]
		Ordenar			
		Definir	Hacer Hecho		
		Programar	Hacer Hecho		
		Probar	Hacer Hecho		
		Terminar			

Fuente: Elaboración propia.

3.2. MODELO CONCEPTUAL DE INTEGRACIÓN DE INGENIERÍA DE SOFTWARE CON INTELIGENCIA ARTIFICIAL.

Para iniciar con el proceso de desarrollo del modelo para determinar factores de éxito de la Inteligencia Artificial con el Desarrollo de Software se ha tomado en cuenta dos cosas, en primera instancia la información establecida a través del análisis de la plataforma EcuCiencia considerando los modelos del ciclo de vida del software aplicados, las fases de las metodologías de desarrollo tradicionales y ágiles que fueron implementadas en la construcción de los módulos de esta plataforma así como también el análisis de los algoritmos de Inteligencia Artificial que fueron implementados para procesos de clasificación, similaridad entre documentos y autores, red social, georreferenciación y sistemas de recomendación que aportan al diseño del modelo teórico.

Finalmente se realiza un análisis sistemático de literatura sobre las metodologías de desarrollo ágil establecidas en la revisión de la literatura por ser las de mayor uso y que permiten obtener factores de implementación para el desarrollo de software, así como también la determinación de los algoritmos implementados en investigaciones científicas y que han permitido la construcción de software de alta calidad.

Las Tablas 38 y 39 presentan la descripción de las fases implementadas por varios autores en las metodologías de desarrollo ágiles.

Tabla 38. Codificación de las fases de desarrollo de Software de la Metodología XP (Extreme Programming).

METODOLOGÍA	FASES	CÓDIGO
XP (Extreme Programming)	Planeación	XPF01
	Planificación de la entrega (Release)	XPF02
	Diseño	XPF03
	Codificación	XPF04
	Pruebas	XPF05
	Exploración	XPF06
	Iteraciones	XPF07
	Producción	XPF08
	Mantenimiento	XPF09
	Muerte del proyecto	XPF10

Fuente: Elaboración propia

Tabla 39. Codificación de las fases de desarrollo de Software de la Metodología KANBAN.

METODOLOGÍA	FASES	CÓDIGO
KANBAN	Definición	KBF01
	Análisis	KBF02
	Desarrollar / Producir	KBF03
	Revisión de la calidad / Probar	KBF04
	Puesta en marcha / Implementar	KBF05
	Visualizar el flujo de trabajo	KBF06
	Limitar el trabajo en curso	KBF07
	Optimizar	KBF08
	Mejora continua	KBF09
	Pila	KBF10
	Seleccionar	KBF11
	Introducir	KBF12
	Ordenar	KBF13
	Programar	KBF14
	Terminar	KBF15

Fuente: Elaboración propia.

En las Tablas 40 y 41 respectivamente se puede establecer una determinación de las fases y su nivel de frecuencia de uso, establecidas en la revisión sistemática de literatura en donde se puede analizar que la fase de planificación, diseño, codificación, pruebas, producción y mantenimiento del proyecto han sido establecidas como de mayor frecuencia por los desarrolladores al aplicar metodología XP (Extreme Programming).

Tabla 40. Fases de desarrollo de Software de la Metodología XP (Extreme Programming).

FASES	COD.
Planeación	XPF01
Planificación de la entrega (Release)	XPF02
Diseño	XPF03
Codificación	XPF04
Pruebas	XPF05
Exploración	XPF06
Iteraciones	XPF07
Producción	XPF08
Mantenimiento	XPF09
Muerte del proyecto	XPF10

Fuente: Elaboración propia.

Tabla 41. Fases de desarrollo de Software de la Metodología KANBAN.

FASES	COD.
Definición	KBF01
Análisis	KBF02
Desarrollar / Producir	KBF03
Revisión de la calidad / Probar	KBF04
Puesta en marcha / Implementar	KBF05
Visualizar el flujo de trabajo	KBF06
Limitar el trabajo en curso	KBF07
Optimizar	KBF08
Mejora continua	KBF09
Pila	KBF10
Seleccionar	KBF11
Introducir	KBF12
Ordenar	KBF13
Programar	KBF14
Terminar	KBF15

Fuente: Elaboración propia.

En la metodología Kanban de igual manera se puede establecer que las variables definición, análisis, desarrollo, programar, revisión de la calidad, mejora continua y terminar han sido determinadas con mayor uso de frecuencia lo que implica que varios programadores utilizan

estas fases señaladas para la construcción de productos mediante la implementación de esta metodología.

Por otro lado, la Tabla 42 presenta un proceso de integración de las prácticas ágiles determinadas para el diseño de productos de desarrollo de software en donde se puede establecer que las variables planeación, codificación, pruebas, exploración, iteraciones, producción, mantenimiento y muerte del proyecto son determinadas como un mayor número de frecuencias y por lo tanto se consideran de una alta aceptación en el campo del desarrollo, se considera importante el estudio de estas variables y la implementación de las mismas para la construcción del modelo teórico en virtud de determinar los factores de éxito más relevantes para el propósito de la investigación señalada.

Tabla 42. Determinación de las fases para el diseño del Modelo de Integración.

	KANBA	XP	FASES	
			XP	KBF
1	✓	✓	XPF01	KBF01
Planeación / Definición			XPF02	
Planificación de la entrega (Release)			XPF03	
Diseño			XPF04	KBF14
Codificación / Programar	✓		XPF05	KBF04
Pruebas / Revisión de la calidad / Probar	✓		XPF06	KBF02
Exploración / Análisis	✓		XPF07	KBF07
Iteraciones / Limitar el trabajo en curso			XPF08	KBF03
Producción / Desarrollar / Producir	✓		XPF09	KBF09
Mantenimiento / Mejora continua	✓		XPF10	KBF15
Muerte del proyecto / Terminar	✓		KBF05	
Puesta en marcha / Implementar	✓		KBF06	
Visualizar el flujo de trabajo	✓		KBF08	
Optimizar	✓		KBF10	
Pila	✓		KBF11	
Seleccionar	✓		KBF12	
Introducir	✓		KBF13	
Ordenar	✓			

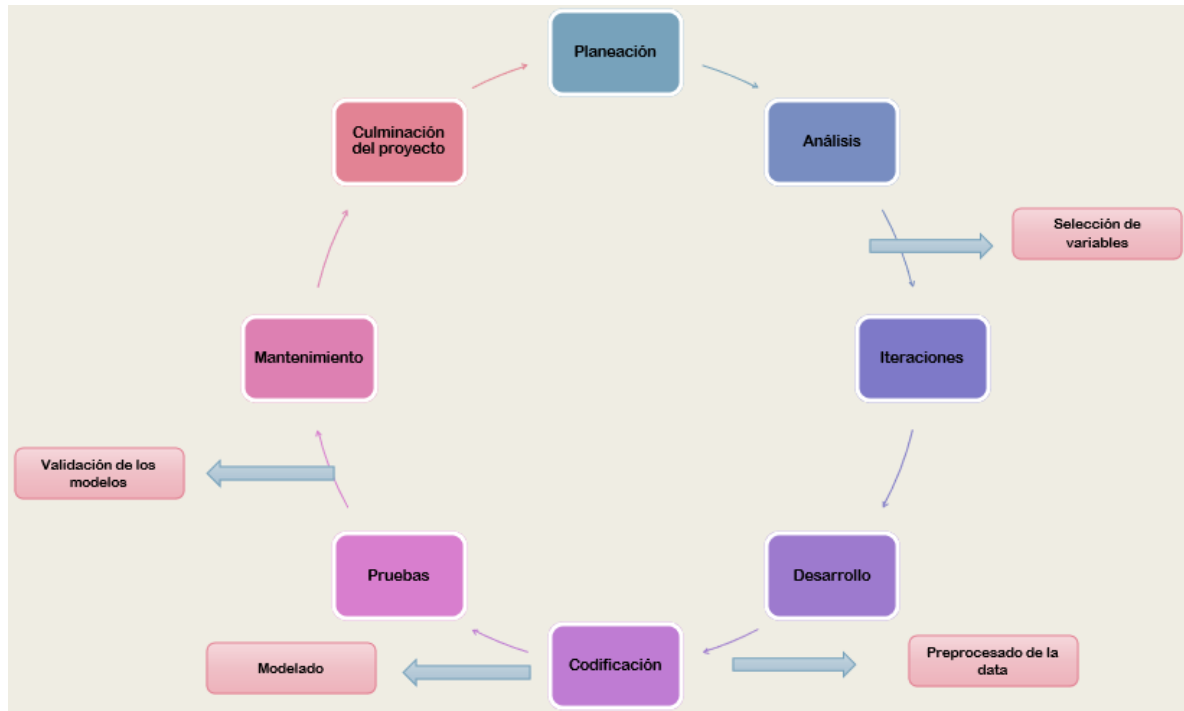
Fuente: Elaboración propia.

De este desglose de variables se establece que la fase de planeación tiene relación con la fase de definición, la fase de codificación con la fase de programación, la fase de pruebas con la fase de revisión de la calidad o probar, la fase de exploración con la fase de análisis, la fase de iteraciones con la fase de limitar el trabajo en curso, la fase de producción con la fase de

desarrollar o producir, la fase de mantenimiento con la fase de mejora continua, la fase de muerte del proyecto con la fase terminar.

Una vez realizada la relación de las nuevas variables del desarrollo de software se establece el análisis de las variables de inteligencia artificial a fin de integrar de manera adecuada los elementos que permitan el diseño del modelo teórico de manera óptima.

Figura 26. Modelo Conceptual de integración de la Ingeniería de Software con la Inteligencia Artificial.



Fuente: Elaboración propia.

Para la validación del modelo conceptual se establece un proceso por juicio de expertos que permiten la sustentación de las variables definidas para el estudio y su grado de confiabilidad con base a lo señalado en la Tabla 43.

Tabla 43. Fases del Modelo de Integración de Inteligencia Artificial con el Desarrollo de Software.

MODELO DE INTEGRACIÓN DE INTELIGENCIA ARTIFICIAL CON EL DESARROLLO DE SOFTWARE		
FASE	DESCRIPCIÓN	REF.
1	Planeación	Se realiza la conformación del equipo de trabajo y establecen las políticas dentro del marco del proyecto, como puede ser: enfatizar la comunicación del equipo. [173]
		Se plantean los objetivos que se desea alcanzar en el proyecto identificando todos sus recursos necesarios para poder cumplirlos. Se consideran tres dimensiones para la planeación como son: calidad, costo y tiempo de duración. [174]
2	Análisis	Se detalla cuál es el problema que se pretende solventar y aquellos factores que lo provocan, para comprender de mejor forma el entorno en donde se realizará la solución. Como resultado se hace una descripción concisa y real del problema y se enlistan las distintas tareas que el sistema tendrá que desempeñar. [175]
		Se designa un analista que es responsable de la identificación de un problema o necesidad que se va a solventar, se obtienen requisitos para dar una solución, estos requisitos son los encargados de definir lo que el sistema debe hacer y satisfacer una necesidad identificada. [176]
	Selección de variables	La selección de variables se encarga de recopilar e integrar las fuentes de datos existentes, identificando y seleccionando las variables más relevantes en los datos para poder aplicar las técnicas de muestreo adecuado. [177]
3	Iteraciones	Dentro del enfoque iterativo ayuda a facilitar la implementación de nuevos cambios, identificando los riesgos de un proyecto, cuando todavía se puede corregirlos a tiempo. La iteración es el resultado de una nueva versión ejecutable a corroborar. [178]
		Cada ciclo se establece en fases y se debe de considerar como mini proyecto, cada fase tiene su núcleo que está hecha por una o más iteraciones que consta de actividades en cualquier desarrollo. [179]
4	Desarrollo	Se hace la construcción del producto o sistema, el punto o línea base de dicha arquitectura del sistema se incrementa hasta llegar a que el sistema esté completo. La culminación de esta fase, no está exento de defectos que se pasaron por alto en el desarrollo. [180]
	Preprocesado de la data	Se corrigen o eliminan los valores faltantes o erróneos y se generan nuevos atributos a partir de los existentes para reducir la complejidad. [181]
5	Codificación	Se realiza la conversión de un diseño a código y es la parte importante de la ingeniería de software, pero no la más complicada. La complejidad de esta fase radica en los lenguajes de programación utilizados y la calidad de diseño realizado previamente. [182]
		En el lenguaje de ingeniería se lo conoce con el término de programación, se realiza la acogida de algoritmos y son trasladados a un lenguaje de programación en específico. [183]
	Modelado	Se constituye una representación abstracta de un cierto aspecto de la realidad, tiene una estructura formada por elementos. Se decide qué técnica, algoritmo y qué patrón se va seguir [184]
6	Pruebas	Se realiza un feedback para corroborar la madurez del sistema desarrollado, es decir, descubrir y documentar defectos del sistema, para poder validar y probar que los requisitos están adecuadamente implementados. [185]
		Fase donde se verifica, ejecuta y se lleva un seguimiento de pruebas, análisis y registro de los errores en plena ejecución del sistema, se realiza un informe de forma detallada con los resultados obtenidos en la ejecución de las pruebas. [186]
	Validación de los modelos	En este punto se analizan y evalúan los patrones obtenidos, y evaluar el modelo comprobando las conclusiones que obtuvieron. En el caso de haber obtenido varios modelos se deben comparar para encontrar el que mejor se ajuste al problema. [187]
7	Mantenimiento	El software que se desarrolló en la fase de desarrollo tiene que ser mantenido, por el mismo hecho de que va a sufrir cambios después de que se entregue al cliente, estos cambios pueden ser por: errores hallados, cambios en el entorno externo, entre otros. [188]
		Una vez que se ha realizado el despliegue del software desarrollado, es necesario empezar a hacer un mantenimiento para dar solución a errores en donde han sido detectados. [189]
8	Culminación del proyecto	Fase final del proyecto, no existen más historias que ser agregadas al sistema desarrollado, logrando satisfacer las necesidades del cliente, siempre y cuando tenga buen rendimiento y confiabilidad. Se procede a la documentación final del sistema. [190]

Fuente: Elaboración propia.

3.3. METODOLOGÍA

3.3.1. Muestreo

La muestra se realizó realizando las siguientes actividades: (i) organización y elaboración de las preguntas que ayuden a determinar los factores de éxito en la integración de la inteligencia artificial con el desarrollo de software. (ii) Envío de correos electrónicos (encuesta en línea) a estudiantes y docentes de la carrera de Ingeniería en Informática y Sistemas Computacionales de la Facultad de Ciencias de la Ingeniería y Aplicadas de la Universidad Técnica de Cotopaxi y también a profesionales que se relacionen con el área de estudio aplicada en esta encuesta.

Se utilizó un cuestionario como instrumento de estudio, para ello, se desarrolló una encuesta impresa y online con Google Forms basado en el modelo propuesto. La encuesta se aplicó a docentes, estudiantes, personal relacionado con las áreas de conocimiento de Inteligencia Artificial, Sistemas e informática y desarrollo de software. El objetivo de esta encuesta fue determinar los factores de éxito que ayudarán a la construcción del modelo de integración.

La encuesta se estructuró en 3 secciones en la primera sección se presenta datos generales de las personas que llenaran la encuesta, en la segunda sección se presenta factores que influyen en la integración del desarrollo de software con inteligencia artificial que se aplicaron con la escala de Likert según 3 niveles (1: No influye, 2: Baja influencia, 3: Alta influencia), y en la tercera sección preguntas que complementan el estudio aplicadas mediante la escala de Likert según una clasificación de 3 niveles (1: Nada importante, 2: Poco importante, 3: Muy importante).

Se realizó una prueba piloto para validar las preguntas. Aplicadas a personas de las áreas mencionadas en la investigación para comprobar si las preguntas permiten la obtención de la información oportuna.

Como resultado se obtiene un total de 28 encuestas para el proceso de validación de expertos que permitan la validación de las variables para la construcción del modelo propuesto.

3.3.2. Fiabilidad de los datos

Para el análisis de la validez del instrumento se utilizó el software SPSS con el cual se determinará el grado de confiabilidad de los datos aplicados en el instrumento.

Tabla 44. Estadística de Fiabilidad del instrumento.

Estadísticas de fiabilidad		
Alfa de Cronbach	Alfa de Cronbach basada en elementos estandarizados	N de elementos
,774	,773	12

Fuente: Elaboración propia.

3.3.3. Análisis e interpretación de la data

Para el análisis e interpretación de los datos, se llevaron a cabo las siguientes actividades: (i) Revisión y depuración de los datos obtenidos con el fin de detectar errores y omisiones, (ii) Análisis descriptivo de la población (análisis de los aspectos de la encuesta: edad, género, área de conocimiento, experiencia del encuestado), (iii) análisis de relación para la integración de Inteligencia Artificial en el proceso de desarrollo de software, la validación de los resultados para verificar las hipótesis.

3.3.4. Juicio de expertos

En las Tablas 45, 46 y 47 respectivamente se muestra la clasificación de las respuestas de los encuestados según el género, área de conocimiento y los años de experiencia correspondientemente. La experiencia se ha clasificado en nivel básico (menos de 2 años de experiencia), intermedio o junior (entre 2 a 6 años), semi senior (entre 7 y 9 años) y senior (más de 10 años de experiencia).

Tabla 45. Análisis de datos por Género.

GÉNERO	PORCENTAJE
Masculino	25%
Femenino	75%
TOTAL	100%

Fuente: Elaboración propia.

Tabla 46. Análisis de datos por Años de experiencia.

EXPERIENCIA	PORCENTAJE
Básico (menos de 2 años)	32,15%
Intermedio junior (2 a 6 años)	10,71%
Semi senior (7 a 9 años)	10,71%
Senior (más de 10 años)	46,43%
TOTAL	100,00%

Fuente: Elaboración propia.

Tabla 47. Análisis de datos por Área de especialización o profesionalización.

ÁREA DE ESPECIALIZACIÓN O PROFESIONALIZACIÓN	PORCENTAJE
Inteligencia Artificial	10,72%
Sistemas e Informática	46,43%
Desarrollo de Software	17,86%
Inteligencia Artificial, Desarrollo de Software	3,57%
Inteligencia Artificial, Sistema e Informática	3,57%
Sistemas e Informática, Desarrollo de Software	10,71%
Inteligencia Artificial, Sistema e Informática, Desarrollo de Software	7,14%
TOTAL	100,00%

Fuente: Elaboración propia.

Como se muestra en la Tabla 45 el 25% de los encuestados corresponde al género femenino, y el 75% del género masculino. En la Tabla 46 se puede visualizar que 32,15 % de los encuestados corresponden al nivel básico en experiencia, el 10,71% se encuentran en el nivel intermedio junior, el 10,71% pertenecen al nivel semi senior y el 46,43% son senior. Las áreas de especialización o profesionalización corresponden al 10,72% tienen conocimiento sobre Inteligencia Artificial, el 46,43% sobre Sistemas e Informática, el 17,86% sobre Desarrollo de Software, el 3,57 en áreas como Inteligencia Artificial y Desarrollo de Software, el otro 3,57% sobre Inteligencia Artificial y Sistemas e Informática, el 10, 71% sobre Sistemas e Informática y Desarrollo de Software y el 7,14% en las tres áreas Inteligencia Artificial, Sistemas e Informática y Desarrollo de Software. Los resultados del análisis estadístico del instrumento aplicado se presentan en la Tabla 48.

Tabla 48. Análisis estadísticos de la encuesta.

		Sección 1: Información General					Sección 2: Factores que influyen en la integración del desarrollo de software con inteligencia artificial							Sección 3: Preguntas que complementan el estudio				
		Identificador	Edad	Género	Área de especialización o	Años de experiencia	Planeación	Análisis	Iteraciones	Desarrollo	Codificación	Pruebas	Mantenimiento	Culminación del Proyecto	Selección de variables	Preprocesado de la Data	Modelado	Validación de los modelos
N	Válido	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28
	Perdidos	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Media		14,50	37,79	1,25	3,04	2,71	2,68	2,82	2,68	2,89	2,79	2,75	2,57	2,36	2,89	2,82	2,86	2,82
Error estándar de la media		1,555	1,698	,083	,343	,256	,090	,074	,090	,060	,094	,111	,108	,128	,060	,074	,067	,090
Mediana		14,50	37,50	1,00	2,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	2,00	3,00	3,00	3,00	3,00
Moda		1 ^a	45	1	2	4	3	3	3	3	3	3	3	3	3	3	3	3
Desviación estándar		8,226	8,987	,441	1,815	1,357	,476	,390	,476	,315	,499	,585	,573	,678	,315	,390	,356	,476
Varianza		67,667	80,767	,194	3,295	1,841	,226	,152	,226	,099	,249	,343	,328	,460	,099	,152	,127	,226
Asimetría		0,000	-,081	1,221	1,143	-,299	-,809	-1,775	-,809	-2,686	-2,383	-2,312	-,936	-,586	-2,686	-1,775	-2,159	-2,806
Error estándar de asimetría		,441	,441	,441	,441	,441	,441	,441	,441	,441	,441	,441	,441	,441	,441	,441	,441	,441
Curtosis		-1,200	-1,235	-,554	,078	-1,808	-1,456	1,234	-1,456	5,614	5,420	4,439	-,038	-,615	5,614	1,234	2,859	7,846
Error estándar de curtosis		,858	,858	,858	,858	,858	,858	,858	,858	,858	,858	,858	,858	,858	,858	,858	,858	,858
Rango		27	31	1	6	3	1	1	1	1	2	2	2	2	1	1	1	2
Mínimo		1	23	1	1	1	2	2	2	2	1	1	1	1	2	2	2	1
Máximo		28	54	2	7	4	3	3	3	3	3	3	3	3	3	3	3	3
Suma		406	1058	35	85	76	75	79	75	81	78	77	72	66	81	79	80	79
Percentiles	25	7,25	29,25	1,00	2,00	1,00	2,00	3,00	2,00	3,00	3,00	3,00	2,00	2,00	3,00	3,00	3,00	3,00
	50	14,50	37,50	1,00	2,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	2,00	3,00	3,00	3,00	3,00
	75	21,75	45,00	1,75	3,75	4,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00

Fuente: Elaboración propia.

Hipótesis H1: ¿En la fase de Análisis se puede determinar un proceso de selección de variables que influyan en el desarrollo del software y la construcción de modelos de Inteligencia Artificial?

La selección de variables se encarga de recopilar e integrar las fuentes de datos existentes, identificando y seleccionando las variables más relevantes en los datos para poder aplicar las técnicas de muestreo adecuado. Se determinan las fases consecutivas que sean capaces de extraer conocimientos válido y útil a partir de la información original. Generalmente se encuentra en bases de datos y otras fuentes tanto internas como externas [177].

Para Troncoso & Fuentes [191] la selección de variables viene siendo la primera de las etapas de la minería de datos, su función es la selección de los datos con los que se va a trabajar, se determinan las fuentes de los datos y el tipo de información a usar. El autor nos sugiere que se debe de conocer en su totalidad las variables que están involucradas e identificar la variable a predecir. De modo idéntico Burgos & González [192] nos menciona que es la primera etapa donde se establece el objetivo de estudio a realizar, al mismo tiempo se realiza la elección adecuada de los datos con su respectiva fuente, para definir una línea meta antes de proceder con el estudio, adicional se hace la elección de variables donde son usadas para un posterior análisis.

Hipótesis H2: ¿En la fase de Desarrollo se pueda establecer una etapa de Preprocesado que permita el análisis de la data y la mejora del rendimiento de los algoritmos de aprendizaje automático aplicados en el desarrollo de software?

En el preprocesamiento de la data corrigen o eliminan los valores faltantes o erróneos y se generan nuevos atributos a partir de los existentes para reducir la complejidad [181]. La cantidad de información que contienen las bases de datos requieren de una buena representación, no solo de reduzcan la dimensionalidad sino también que preserven la información más relevante. Es necesario convertir a categóricos para poder obtener el número de segmentos y rangos en que se puede agrupar [193].

Esta etapa es primordial por la interacción entre el usuario o analista, se hace una verificación sobre la calidad de los datos y se aplican operaciones simples para la eliminación de datos con ruido como: estrategias para el manejo de datos desconocidos, datos nulos, duplicación de los datos y técnicas estadísticas para ser reemplazados. Para el proceso de limpieza de la data estos

valores son omitidos, se remplazan por un valor cercano, es decir, haciendo uso de métricas estadísticas como media, moda, mínimo y máximo para ser reemplazados [194].

El procesamiento de datos es una etapa principal para el proceso de descubrimiento de información o KDD (Knowledge Discovery in Databases, en inglés). García, Ramírez, Luengo & Herrera [195] nos manifiesta que en esta etapa se hace una limpieza de datos, integración y reducción. El uso de los datos normalmente es de baja calidad como consecuencia el proceso de la minería de datos son pobres en resultados, es por ello que se aplican técnicas de preprocesamiento para solventar dicho problema.

Hipótesis H3: ¿En la fase de Codificación respecto al desarrollo de software, se puede establecer el modelado de algoritmos de Inteligencia Artificial?

En el modelo de algoritmos se constituye una representación abstracta de un cierto aspecto de la realidad, tiene una estructura formada por elementos. Se decide qué técnica, algoritmo y qué patrón se va seguir [184]. Se puede definir como el conjunto de tareas que se debe realizar, se caracteriza por ser la parte más relevante del proceso el cual tiene como objetivo identificar los patrones de comportamiento de los datos, los cuales pueden ser descriptivos o predictivos [196].

En la etapa del modelado de algoritmos para Jaramillo & Paz [197] se hace la elección de técnicas que ayuden en la realización del modelado, los cuales deben ser apropiadas para resolver un problema en concreto. En la construcción del modelo con la técnica seleccionada, se integran un conjunto de datos para generar uno o más modelos, se procede a ajustar los parámetros en la técnica que fue seleccionada de forma iterativa con el fin de generar mejores resultados.

En esta fase se seleccionan y se hace uso de distintas técnicas (algoritmos) de modelado, donde se realiza un calibre en sus parámetros para lograr obtener valores óptimos. Por otro lado, en un mismo problema de minería de datos se tiene distintas técnicas susceptibles de ser usadas, cada una de ellas puede tener requisitos diferentes en la forma en que pueden presentarse los datos de entrada, es probable que se realicen ciclos adicionales de preparación de los datos [198].

Hipótesis H4: ¿En la fase de Pruebas respecto al desarrollo de software, se pueda determinar un proceso de validación de los algoritmos de aprendizaje automático estableciendo métricas estadísticas de evaluación?

En la validación de los algoritmos se analizan y evalúan los patrones obtenidos, y evaluar el modelo comprobando las conclusiones que obtuvieron. En el caso de haber obtenido varios modelos se deben comparar para encontrar el que mejor se ajuste al problema [187]. Una vez obtenido el modelo hay que realizar la comprobación y verificar si son válidas y satisfactorias. Si se ha obtenido varios modelos se tiende a comparar en busca de que se ajuste al inconveniente. Si ninguno consigue los resultados esperados, se debe cambiar alguno de los pasos anteriores. [199].

De acuerdo a Güemes [200] se deben evaluar los patrones minados y compararlos con objetivos propuestos, y si fuese necesario se debe volver a las fases anteriores para encontrar una mejor iteración, es decir, que se volverían analizar los datos para detectar sus efectos sobre los datos obtenidos luego del algoritmo utilizado para después poderlo dar a conocer.

4. CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- La revisión bibliográfica permitió el sustento teórico del problema de investigación y la hipótesis para la construcción del marco teórico. Se realiza una revisión sistemática de literatura mediante el análisis de 200 documentos primarios en bases de datos científicas.
- Del caso de estudio del proyecto de EcuCiencia se determina 2 metodologías de desarrollo de software ágil, 2 modelos del ciclo de vida del software para el desarrollo y 11 fases. Así como también, 24 algoritmos de Inteligencia artificial y 1 metodología de minería de datos, que soportan el accionar de la plataforma científica EcuCiencia.
- Se obtiene un modelo conceptual para la determinación de factores de éxito que permitan la integración de la Inteligencia Artificial con el Desarrollo de Software.

RECOMENDACIONES

- Estudio de modelos de desarrollo tradicionales que permitan analizar de manera amplia estos elementos que pueden contribuir en el desarrollo de productos de software con la implementación de algoritmos de Inteligencia Artificial.
- Analizar un mayor número de casos de estudio para poder identificar nuevos factores de éxito para este tipo de procesos de integración.
- Incorporación de metodologías, fases y actividades de minería de datos para diseñar un modelo integral que permita la construcción de productos de desarrollo de alta calidad.

5. BIBLIOGRAFÍA

- [1] J. Ortiz Freuler and C. Iglesias, *Algorithms e Inteligencia Artificial en Latin America. Un Estudio de implementaciones por parte de Gobiernos en Argentina y Uruguay*. World Wide Web Foundation, 2018.
- [2] N. Villavicencio Bermúdez, M. Peña Abreu, S. Burneo Valareso, and I. Pérez Pupo, “Experiencias en la integración de procesos en las organizaciones orientadas a proyectos de software Experiences in processes integration for software projects organizations,” *Rev. Cuba. Ciencias Informáticas*, vol. 10, no. Especial UCIENCIA, pp. 171–185, 2016, [Online]. Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992016000500013
- [3] K. Rojas Escobar and L. A. Araujo Taboada, “Planificación de cronogramas en entornos multi-proyectos de desarrollo de software Schedule planning in multiproject software development environments,” *Ser. Científica la Univ. las Ciencias Informáticas*, vol. 12, no. 10, pp. 24–39, 2019, [Online]. Available: <https://publicaciones.uci.cu/index.php/serie/article/view/484>
- [4] J. J. Morales Arias and C. J. Pardo Calvache, “Revisión sistemática de la integración de modelos de desarrollo de software dirigido por modelos y metodologías ágiles,” *Inf. Técnico*, vol. 80, no. 1, pp. 87–99, 2016, doi: 10.23850/22565035.326.
- [5] B. Molina Montero, H. Vite Cevallos, and J. Dávila Cuesta, “Metologías ágiles frente a las tradicionales en el proceso de desarrollo de software,” *Espirales Rev. Multidiscip. Investig.*, vol. 2, no. 17, pp. 114–121, 2018, [Online]. Available: <https://www.researchgate.net/publication/327537074>
- [6] “Chaos Report 2015,” *The Standish Group International, Inc.*, pp. 1–13, 2015.
- [7] *Guía del PMBOOK*. Pennsylvania: Project Management Institute, Inc., 2017.
- [8] V. Ángel, *Ingeniería de Software I*. Bogotá: Fundación Universitaria del Área Andina, 2018.
- [9] M. A. Mascheroni and E. Irrazábal, “Problemas que afectan a la Calidad de Software en Entrega Continua y Pruebas Continuas,” in *In XXIV Congreso Argentino de Ciencias de la Computación (La Plata, 2018)*, 2018, vol. 3, no. 10, pp. 592–601. [Online]. Available: <http://sedici.unlp.edu.ar/handle/10915/73270>

- [10] D. Madruga Hernández and D. Castillo Domínguez, “Pensando el desarrollo de software desde el usuario,” vol. 10, no. 3, pp. 1–10, 2017.
- [11] E. Tapia, “Ecuador utiliza la inteligencia artificial para atender al cliente.,” *Revista Líderes*, 2018. <https://www.revistalideres.ec/lideres/ecuador-inteligencia-artificial-atencion-banca.html>
- [12] W. Ruiz Martinez and A. A. Gonzalez Gomez, “An approach from the software engineering of an IoT and machine learning technological solution that allows monitoring and controlling environmental variables in a coffee crop,” *Ingeniería*, vol. 26, no. 3, pp. 465–478, 2021, doi: <https://doi.org/10.14483/23448393.18495>.
- [13] H. M. Chanatasig Toapanta, “Enseñanza de la ingeniería de Software,” *Publicando*, vol. 4, no. 12, pp. 1390–1394, 2017, [Online]. Available: <https://revistapublicando.org/revista/index.php/crv/article/view/645>
- [14] F. González González, S. L. Calero Castañeda, and D. F. Loaiza Buitrago, “Comparación de las metodologías cascada y ágil para el aumento de la productividad en el desarrollo de software,” *Repos. Univ. Santiago Cali*, pp. 1–10, 2019, [Online]. Available: <https://repository.usc.edu.co/handle/20.500.12421/1208>
- [15] L. Rouhiainen, *Inteligencia artificial 101 cosas que debes saber sobre nuestro futuro hoy*. Barcelona: Editorial Planeta, S.A., 2018. [Online]. Available: https://planetadelibrosar0.cdnstatics.com/libros_contenido_extra/40/39307_Inteligencia_artificial.pdf
- [16] J. G. Corvalán, “El impacto de la Inteligencia Artificial en el trabajo,” *Rev. Direito Econômico e Socioambiental*, vol. 10, no. 1, p. 35, 2019, doi: <https://doi.org/10.7213/rev.dir.econ.soc.v10i1.25870>.
- [17] P. Galván, “Impacto de la Inteligencia Artificial en el Desarrollo de Software,” *Mayo*, 2018. <https://sg.com.mx/revista/56/inteligencia-artificial-desarrollo-software>
- [18] A. Ramírez Quesada, “Competición Vídeo: Inteligencia artificial aplicada al desarrollo de software,” pp. 1461–1463.
- [19] G. L. Ferreira Lorenzo, D. Gálvez Lío, L. A. Quintero Domingués, and J. Antón Vargas, “Estimación del esfuerzo en proyectos de software utilizando técnicas de inteligencia artificial,” *Rev. Cuba. Ciencias Informáticas*, vol. 8, no. 4, pp. 1–20, 2014, [Online].

Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992014000400001

- [20] N. Ugalde Binda and F. Balbastre Benavent, “Investigación Cuantitativa E Investigación Cualitativa: Buscando Las Ventajas De Las Diferentes Metodologías De Investigación,” *Rev. Ciencias Económicas*, vol. 31, no. 2, pp. 179–187, 2013, [Online]. Available: <http://www.revistas.ucr.ac.cr/index.php/economicas/article/view/12730>
- [21] M. A. Guerrero Berajano, “La investigación cualitativa,” *INNOVA Res. J.*, vol. 1, no. 2, pp. 1–9, 2016, [Online]. Available: <http://biblioteca.udgvirtual.udg.mx/jspui/handle/123456789/1167>
- [22] E. Rus Arias, “Investigación cuantitativa,” *economipedia*, 2021. <https://economipedia.com/definiciones/investigacion-cuantitativa.html>
- [23] A. Álvarez Vega, “Investigación bibliográfica - Una herramienta de evolución,” *Gach Integral Group*, p. 21.
- [24] C. I. Rivas, V. P. Corona, J. F. Gutiérrez, and L. Hernández, “Metodologías actuales de desarrollo de software,” *Rev. Technol. e Innovación*, vol. 2, no. 5, pp. 980–986, 2015.
- [25] M. E. Alfaro Parada, “Utilización de Metodologías de Inteligencia Artificial y sus Aplicaciones en El Salvador.,” *Ing-Novacion*, no. 3, pp. 57–68, 2012.
- [26] L. J. Pin García, M. M. Toala Zambrano, and J. E. Álava Cruzatty, “La Inteligencia Artificial En La Calidad Del Software: Una Revisión Sistemática De La Literatura,” *UNESUM-Ciencias. Rev. Científica Multidiscip.*, vol. 4, no. 1, pp. 75–86, 2020, doi: <https://doi.org/10.47230/unesum-ciencias.v4.n1.2020.208>.
- [27] I. C. Arango Palacio, “Oportunidades para la transformación digital de la cadena de suministro del sector bananero basado en software con inteligencia artificial,” *Rev. Politécnica*, vol. 17, no. 33, pp. 47–63, 2021, doi: <https://doi.org/10.33571/rpolitec.v17n33a4>.
- [28] R. García Vacacela, I. Pérez Pupo, N. Villavicencio, P. Y. Piñero, and S. Beovides, “Experiencias usando algoritmos genéticos en la planificación de proyectos / Experiences by using genetic algorithms in project scheduling,” *Rev. Cuba. Ciencias Informáticas*, vol. 10, no. Especial UCIENCIA, pp. 71–86, 2016, [Online]. Available: <http://scielo.sld.cu/scielo.php?pid=S2227->

18992016000500006&script=sci_arttext&tlng=pt

- [29] D. Ramos, R. Noriega, J. R. Laínez, and A. Durango, *Curso de Ingeniería de Software 2da Edición*. 2017.
- [30] G. Pantaleo and R. Ludmila, *Ingeniería de Software*. Alfaomega Grupo Editor, 2015.
- [31] R. A. Aguilar Vera, J. C. Díaz Mendoza, and Universidad Autónoma de Yucatán, “La Ingeniería de Software en México: hacia la consolidación del primer programa de licenciatura,” *Revista Conaic*, vol. 2, no. 2, 2015.
- [32] A. M. Ruiz Piedra, F. Gómez Martínez, M. del P. Gilbert Lamadrid, E. B. Soca Guevara, and L. Rodríguez Blanco, “Reseña histórica sobre la gestión nacional del desarrollo del software educativo en la Educación Médica Superior en Cuba,” pp. 28–39, 2018.
- [33] S. R. Gómez Palomo and E. A. Moraleda Gil, *Aproximación a la ingeniería del software*. 2020.
- [34] R. S. Pressman, *Ingeniería del Software. Un enfoque práctico.*, 7a Edición. The McGraw-Hill Companies, Inc., 2002.
- [35] I. Sommerville, *Ingeniería de Software*, 9a edición. Pearson Education, Inc., 2011.
- [36] P. M. Vera, “Desarrollo Dirigido por Modelos Basado en Componentes de Interfaz de Usuario,” Universidad Nacional de la Plata, 2015. [Online]. Available: https://www.researchgate.net/publication/302505998_Desarrollo_Dirigido_por_Modelos_Basado_en_Componentes_de_Interfaz_de_Usuario
- [37] A. Delgado Castillo and R. Reyes Pérez, “Propuesta de Guía para el Aseguramiento del Proceso de Calidad de Software para las Empresas Costarricenses que desarrollan Software para Consumo Nacional,” 2019.
- [38] E. Gómez Gutierrez, M. M. Marcillo Guevara, and N. Ramírez López, “Metodologías Ágiles para el Desarrollo de Proyectos,” 2020.
- [39] M. Romero Alencastro and Universidad de Especialidades Espíritu Santo, “Implementación de Metodología Ágil en la Gestión de Proyectos de una Entidad Financiera en Ecuador,” 2020.
- [40] J. Aguirre Barrera and S. Aguirre Barrera, “Metodologías para el desarrollo de Proyectos,” 2020.

- [41] J. P. Zumba Gamboa and C. A. León Arreaga, “Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software.,” *INNOVA*, vol. 3, no. 10, pp. 20–33, 2018.
- [42] J. Cervantes Gómez and M. del C. Gómez Fuentes, “Taxonomía de los modelos y metodologías de desarrollo de software más utilizados,” *Investig. científica y tecnológica para docentes*, vol. 62, no. 52, pp. 37–47, 2012.
- [43] S. M. Velásquez Restrepo, J. D. Vahos Montoya, M. E. Gómez Adasme, A. A. Pino Martínez, E. J. Restrepo Zapata, and S. Londoño Marín, “Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software,” *Rev. CINTEX*, vol. 24, no. 2, pp. 13–23, 2019, doi: 10.33131/24222208.334.
- [44] N. T. Quesada Reyes and Universidad de Sevilla, “Estudio sobre Metodologías Ágiles en los Proyectos Software. Propuesta de Plan de Implantación para PYMES.,” 2020.
- [45] A. Patino, “Metodología Waterfall: definición, pros y contras,” 2020. <https://blog.comparasoftware.com/metodologia-waterfall/>
- [46] M. Ortiz, “Modelo Incremental.” <http://isw-udistrital.blogspot.com/2012/09/ingenieria-de-software-i.html> (accessed Feb. 12, 2022).
- [47] A. M. Hernández Oliva, B. Bron Fonseca, and L. C. Matamoros Benitez, “Modelos de desarrollo de software. ¿Cuál elegir? Software development models. Which to choose?,” *Univ. las Ciencias Informáticas. La Habana, Cuba*, vol. 11, no. 3, pp. 34–55, 2019, [Online]. Available: <https://publicaciones.uci.cu/index.php/serie/article/view/186>
- [48] L. de la C. Delgado Olivera and L. M. Díaz Alonso, “Modelos de Desarrollo de Software | Software Development Models,” *Rev. Cuba. Ciencias Informáticas*, vol. 15, no. 1, pp. 37–51, 2021.
- [49] E. Cusquisiban Villoslada, “Modelos de Desarrollo Basado en Componentes.” <https://es.scribd.com/document/422107259/Modelos-de-Desarrollo-Basado-en-Componentes> (accessed Feb. 12, 2022).
- [50] A. Mármol and J. Perez, “Desarrollo de un marco de trabajo con node.js basado en componentes para el manejo de solicitudes a objetos de negocio embebidos en el backend,” *Rev. Tecnocientífica URU*, no. 19, pp. 21–28, 2020.
- [51] J. Barrera Arotoma and Z. A. Zubieta Campos, “Bots para Automatización de Procesos

- de Registro de Damnificados por Desastres Naturales,” Universidad Tecnológica de Perú, 2019.
- [52] Microsoft, “Microsoft Solutions Framework (MSF) Overview,” 2015.
- [53] F. D. Cañas Amaya, “Desarrollo de un software para la gestión contratación interna y externa de la Fundación Fundagester,” Universidad de Antioquia, 2021.
- [54] A. López Gil, “Estudio comparativo de metodologías tradicionales y ágiles para proyectos de Desarrollo de Software,” Universidad de Valladolid, 2018.
- [55] E. R. Pilataxi Alba, “E-portafolio y su incidencia en los procesos de evaluación de docentes en la Unidad Educativa Ibarra, utilizando la metodología Extreme Programming,” Universidad Técnica del Norte, 2018.
- [56] D. L. Ramírez Bedoya, J. W. Branch Bedoya, and J. A. Jiménez Builes, “Metodología de desarrollo de software para plataformas educativas robóticas usando ROS-XP,” *Rev. Politécnica*, no. 30, pp. 66–89, 2019.
- [57] M. Ramírez Ramírez, M. del C. Salgado Soto, H. B. Ramírez Moreno, E. Manrique Rojas, N. D. C. Osuna Millán, and R. F. Rosales Cisneros, “Metodología SCRUM y desarrollo de Repositorio Digital,” *Rev. Ibérica Sist. e Tecnol. Inf.*, vol. 17, no. 01, pp. 1062–1072, 2019, [Online]. Available: <https://www.proquest.com/docview/2195127128?pq-origsite=gscholar&fromopenview=true>
- [58] J. C. Arias Becerra and C. E. Durango Vanegas, “Propuesta de un método para desarrollar Sistemas de Información Geográfica a partir de la metodología de desarrollo ágil Scrum,” *Cuad. Act.*, vol. 10, no. 1, pp. 29–41, 2018, [Online]. Available: <https://ojs.tdea.edu.co/index.php/cuadernoactiva/article/view/490>
- [59] O. I. Trejos Buriticá, *Programación Imperativa con Lenguaje C*. 2017. [Online]. Available: <https://www.ecoediciones.com/wp-content/uploads/2017/11/Programación-imperativa-con-lenguaje-C.pdf>
- [60] J. C. García Monsálvez, “Python como primer lenguaje de programación textual en la Enseñanza Secundaria,” *Educ. Knowl. Soc.*, vol. 18, no. 2, pp. 147–162, 2017, [Online]. Available: <https://doi.org/10.14201/eks2017182147162>
- [61] A. Sarasa Cabezuelo, *Gestión de la información web usando Python*. Editorial UOC,

2017.

- [62] J. S. Walker, *Python: La Guía Definitiva para Principiantes para Dominar Python*. Babelcube Inc, 2018.
- [63] J. A. Troyano, F. Cruz, M. González, C. G. Vallejo, and M. Toro, “Introducción a la Programación con Python, Computación Interactiva y Aprendizaje Significativo,” in *Actas de las Jenui*, 2018, vol. 3, pp. 223–230. [Online]. Available: <https://www.researchgate.net/publication/328928778>
- [64] PostgreSQL, “41. PostgreSQL: The World’s Most Advanced Open Source Relational Database.” <https://www.postgresql.org/docs/current/> (accessed Nov. 10, 2021).
- [65] A. Escobar Martín, “Replicación multimaestro en bases de datos PostgreSQL,” Universidad de Alcalá Escuela Politécnica Superior, 2017. [Online]. Available: <https://ebuah.uah.es/xmlui/handle/10017/32022>
- [66] M. P. Zea Ordóñez, J. R. Molina Ríos, and F. F. Redrován Castillo, *Administración de Bases de Datos PostgreSQL*. 3Ciencias, 2017.
- [67] Django, “Django makes it easier to build better web apps more quickly and with less code.” <https://www.djangoproject.com/> (accessed Dec. 10, 2021).
- [68] D. E. Gómez García, “Desarrollo del Sistema De Requisiciones para la Empresa Hidroelectrica Abanico S.A. Aplicando el entorno De Programacion Node.Js,” Escuela Superior Politécnica de Chimborazo, 2018.
- [69] “Inteligencia artificial,” *NOTA-INCyTU*, no. 012, pp. 1–6, 2018.
- [70] J. C. Ponce Gallegos *et al.*, *Inteligencia Artificial*. 2014.
- [71] A. T. Norman, “Aprendizaje automático en acción,” Litres, 2021.
- [72] tatic (FROM DATA TO VALUE), “La diferencia entre inteligencia artificial, aprendizaje automático y aprendizaje profundo,” 2021. <https://tatic.net/es/blog/la-diferencia-entre-inteligencia-artificial-aprendizaje-automatico-y-aprendizaje-profundo/>
- [73] L. J. Sandoval, “Algoritmos de aprendizaje automático para análisis y predicción de datos.,” *Rev. Tecnológica*, vol. 11, no. 11, pp. 36–40, 2018, [Online]. Available: <http://hdl.handle.net/10972/3626>

- [74] O. Simeone, "A Very Brief Introduction to Machine Learning with Applications to Communication Systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 648–664, 2018, doi: 10.1109/TCCN.2018.2881442.
- [75] P. Recuero de los Santos, "Tipos de aprendizaje en Machine Learning: supervisado y no supervisado," 2021. <https://empresas.blogthinkbig.com/que-algoritmo-elegir-en-ml-aprendizaje/>
- [76] G. A. Betancourt, "Las Máquinas de Soporte Vectorial (SVMs)," *Sci. Tech.*, vol. XI, no. 27, pp. 67–72, 2005.
- [77] M. E. Gracia Granados, "Máquinas de soporte vectorial y árboles de clasificación para la detección de operaciones sospechosas de lavado de activos," *Lámpsakos*, vol. 21, pp. 26–38, 2019, doi: <https://doi.org/10.21501/21454086.2904>.
- [78] G. Cano *et al.*, "Predicción de solubilidad de fármacos usando máquinas de soporte vectorial sobre unidades de procesamiento gráfico," *Rev. Int. Metod. Numer. para Calc. y Disen. en Ing.*, vol. 33, no. 1–2, pp. 97–102, 2017, doi: <https://doi.org/10.1016/j.rimni.2015.12.001>.
- [79] V. M. Cuevas Soto, S. Alvarez Iriarte, M. Azcona Romera, and I. Rodríguez Rogert, "Capacidad predictiva de las Máquinas de Soporte Vectorial. Una aplicación en la planificación financiera.," *Rev. Cuba. Ciencias Informáticas*, vol. 13, no. 3, pp. 59–75, 2019, [Online]. Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992019000300059
- [80] L. Salamanca Polo, "Generación de conjuntos de máquinas de soporte vectorial mediante técnicas de remuestreo e inyección de ruido en las etiquetas de clase," Universidad Autónoma de Madrid, 2017. [Online]. Available: <http://hdl.handle.net/10486/679328>
- [81] J. D. Terán Villanueva *et al.*, "Estudio de redes neuronales para el pronóstico de la demanda de asignaturas," *Rev. Fac. Ing.*, vol. 28, no. 50, pp. 34–43, 2018, doi: <https://doi.org/10.19053/01211129.v28.n50.2019.8783>.
- [82] J. A. Toral Barrera, "Redes Neuronales," *Univ. Nac. Autónoma México*, 2019, [Online]. Available: http://www.cucei.udg.mx/sites/default/files/pdf/toral_barrera_jamie_areli.pdf
- [83] C. Dupouy Berrios, "Aplicación de árboles de decisión para la estimación del escenario

- económico y la estimación de movimiento la tasa de interés en Chile,” Universidad de Chile, 2014. [Online]. Available: <https://repositorio.uchile.cl/handle/2250/117556>
- [84] J. C. Riascos and J. E. Molina Muñoz, “Breves consideraciones acerca de la importancia de los árboles de decisión en el análisis de carteras,” *Tendencias*, vol. 17, no. 1, pp. 11–33, 2016, [Online]. Available: <https://doi.org/10.22267/rtend.161701.11>
- [85] E. F. Contreras Morales, F. M. Ferreira Correa, and M. Valle, “Diseño de un modelo predictivo de fuga de clientes utilizando árboles de decisión,” *Rev. Ing. Ind.*, vol. 16, no. 1, pp. 7–23, 2017.
- [86] E. F. González Castañeda, A. A. Torres García, C. A. Reyes García, and L. Pineda Villaseñor, “Aplicación de la Sonificación de Señales Cerebrales en Clasificación Automática,” *Rev. Mex. Ing. Biomédica*, vol. 36, no. 3, pp. 235–250, 2015, [Online]. Available: http://www.scielo.org.mx/scielo.php?pid=S0188-95322015000300008&script=sci_abstract&tlng=pt
- [87] A. Pereira Toledo, J. D. López Cabrera, and L. A. Quintero Domínguez, “Estudio experimental para la comparación del desempeño de Naïve Bayes con otros clasificadores bayesianos Experimental Study for the Comparison of Naïve Bayes with other Bayesian Classifiers,” *Rev. Cuba. Ciencias Informáticas*, vol. 11, no. 4, pp. 67–84, 2017, [Online]. Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992017000400006
- [88] L. García García, M. F. Mata Rivera, and R. Zagal Flores, “Clasificador Mejorado de Textos para el de Medio Ambiente usando Naïve Bayes Multinomial en México,” *Tlamati Sabiduría*, vol. 9, no. Especial 4, p. 12, 2018, [Online]. Available: <http://tlamati.uagro.mx/t94e/t94eVIII1.pdf>
- [89] “Algoritmo de árbol de decisión: algoritmo ID3,” *programador clic*, 2021. <https://programmerclick.com/article/46671391368/>
- [90] J. Zárate Valderrama, N. Bedregal Alpaca, and V. Cornejo Aparicio, “Modelos de clasificación para reconocer patrones de deserción en estudiantes universitarios,” *Ingeniare. Rev. Chil. Ing.*, vol. 29, no. 1, pp. 168–177, 2021, doi: <http://dx.doi.org/10.4067/S0718-33052021000100168>.
- [91] S. Ramos Hernández, “TAID versus CHAID Búsqueda de perfiles de mujeres

- trabajadoras en el servicio doméstico,” Universidad de Salamanca, 2015. [Online]. Available:
https://gredos.usal.es/bitstream/handle/10366/128233/TFM_MAADM_Ramos_Hernandez_Sergio.pdf;jsessionid=AAE5660464184ED6C333FB435A4CF87E?sequence=4
- [92] J. D. Velásquez Henao, C. J. Franco Cardona, and P. A. Camacho, “Nonlinear time series forecasting using MARS,” *Dyna*, vol. 81, no. 184, p. 11, 2014, doi: <http://dx.doi.org/10.15446/dyna.v81n184.39699>.
- [93] C. J. Gil Bellosta, “Importancia de variables en árboles,” *Estadística, Análisis de Datos y Más*, 2013. [https://www.datanalytics.com/2013/11/06/importancia-de-variables-en-arboles/#:~:text=Los árboles \(o árboles de,reside fundamentalmente en la interpretabilidad.&text=Y el tamaño dificulta censar qué variables y en qué manera aparecen](https://www.datanalytics.com/2013/11/06/importancia-de-variables-en-arboles/#:~:text=Los árboles (o árboles de,reside fundamentalmente en la interpretabilidad.&text=Y el tamaño dificulta censar qué variables y en qué manera aparecen).
- [94] R. Mosquera, O. D. Castrillón, and L. Parra, “Máquinas de Soporte Vectorial, Clasificador Naïve Bayes y Algoritmos Genéticos para la Predicción de Riesgos Psicosociales en Docentes de Colegios Públicos Colombianos,” *Inf. Tecnol.*, vol. 29, no. 6, pp. 153–162, 2018, doi: <http://dx.doi.org/10.4067/S0718-07642018000600153>.
- [95] C. Gaviria Peña, “Regresión por Mínimos Cuadrados Parciales P LS Aplicada a Datos Variedad Valuados.,” Universidad Nacional de Colombia, 2015. [Online]. Available: <https://repositorio.unal.edu.co/handle/unal/57110>
- [96] D. Rodríguez, “La regresión logística,” *ANALYTICS LANE*, 2018. <https://www.analyticslane.com/2018/07/23/la-regresion-logistica/#:~:text=El algoritmo de regresión logística,obtiene un resultado u otro>.
- [97] Unipython, “ENSEMBLE METHODS O MÉTODOS DE CONJUNTO,” 2022. <https://unipython.com/ensemble-methods-metodos-conjunto/#:~:text=En los algoritmos de conjunto,para formar una predicción final>.
- [98] J. Zamorano Ruíz, “Comparativa y Análisis de Algoritmos de Aprendizaje Automático para la Predicción del Tipo Predominante de Cubierta Arbórea,” Universidad Complutense de Madrid, 2018. [Online]. Available: <https://eprints.ucm.es/id/eprint/48800/>
- [99] L. Gonzalez, “Aprendizaje no Supervisado,” *Aprende IA*, 2020.

<https://aprendeia.com/aprendizaje-no-supervisado-machine-learning/>

- [100] V. Roman, “Aprendizaje No Supervisado en Machine Learning: Agrupación,” *Medium*, 2019. <https://medium.com/datos-y-ciencia/aprendizaje-no-supervisado-en-machine-learning-agrupacion-bb8f25813edc>
- [101] Y. J. Vera Llaque, “Análisis de Trayectorias Vehiculares GPS para Evaluar su Calidad de Agrupamiento utilizando Algoritmos Clustering de Minería de Datos.,” Universidad de Guayaquil, 2018.
- [102] V. A. Sanabria Ruiz, “Aplicación de técnicas de agrupamiento (clustering) para el análisis estadístico de tendencias en twitter basado en el lenguaje de programación R.,” *Networks*, vol. 5, no. 1, p. 96, 2017.
- [103] H. R. Gonzáles Argote and U. A. Ticona Gonzáles, “Clustering, mediterraneidad y comercio internacional: aplicación empírica de los algoritmos Partitioning Around Medoids y K-means,” *Rev. Latinoam. Desarro. Económico*, no. 32, pp. 96–130, 2019.
- [104] B. J. Mayorga Márquez, “Pronóstico Espacial de Demanda Eléctrica Mediante la Técnica de Agrupamiento (Clustering) de Curvas S Históricas - Aplicación a la Empresa Eléctrica Ambato Regional Centro Norte S.A.,” Escuela Politécnica Nacional, 2018.
- [105] J. A. Espinoza Guillen Agramonte and M. B. Alderete Malpartida, “Caracterización de regiones espacialmente homogéneas de monóxido de carbono en Lima Metropolitana mediante el algoritmo de clustering k-means,” *BIOTECH Eng. Untels*, vol. 1, no. 1, pp. 17–28, 2021, doi: <https://doi.org/10.52248/eb.Vol1Iss01.4>.
- [106] K. G. Soni and A. Patel, “Comparative Analysis of K-means and K-medoids Algorithm on IRIS Data,” *Int. J. Comput. Appl.*, vol. 13, no. 5, pp. 899–906, 2017.
- [107] G. Bernal Baró, “Diseño de una Metaheurística para el Minado de Reglas de Asociación en Bases de Datos Transaccionales,” Universidad Autónoma del Estado de México, 2018.
- [108] F. T. Lévano Chiroque, “Identificación de las Reglas de Asociación utilizando los Algoritmos Secuenciales SPADE y GPS,” Universidad Nacional Agraria la Molina, 2018. [Online]. Available: <https://hdl.handle.net/20.500.12996/4021>
- [109] E. A. Araujo Escobar, “Reglas de Asociación y Predicción utilizando series de tiempo,” Universidad Politécnica Salesiana Sede Quito, 2018. [Online]. Available:

<http://dspace.ups.edu.ec/handle/123456789/16017>

- [110] J. Alquicira, “Análisis de componentes principales (PCA),” *conogasi*, 2017. <https://conogasi.org/articulos/analisis-de-componentes-principales-pca/>
- [111] C. A. Ramírez Morales, “Algoritmo SVD aplicado a los sistemas de recomendación en el comercio,” *Tecnol. Investig. y Acad.*, vol. 6, no. 1, pp. 18–27, 2018.
- [112] M. Delbracio and M. Mateu, “Trabajo Final de Reconocimiento de Patrones : Identifiacion utilizando PCA , ICA y LDA .,” *Grup. Trat. señales la Univ. la Republica-Instituto Ing. Electr. Montevideo, Uruguay.*, pp. 1–13, 2006, [Online]. Available: https://www.fing.edu.uy/iie/investigacion/grupos/biometria/proyectos/patrones/RecPat_MM.pdf
- [113] E. J. Blanco and H. Sanz, “Algoritmos de clustering y aprendizaje automático aplicados a Twitter.,” 2016. [Online]. Available: <https://upcommons.upc.edu/bitstream/handle/2117/82434/113257.pdf?sequence=1&isAllowed=y>
- [114] “Algoritmos probabilísticos,” *Ecured*, 2022. https://www.ecured.cu/Algoritmos_probabilísticos
- [115] J. Arroyo Hernández, “Métodos de reducción de dimensionalidad: Análisis comparativo de los métodos APC, ACPP y ACPK,” *Uniciencia*, vol. 30, no. 1, pp. 115–122, 2016, doi: <https://doi.org/10.15359/ru.30-1.7>.
- [116] J. C. Curvelo Santana, S. de Araújo, J. P. M. Biazus, and R. de Souza, “Simulación del proceso de biodegradación de aguas residuales de la industria de carne mediante una red neuronal artificial perceptrón multicapa,” *Rev. Chil. Ing.*, vol. 23, no. 2, pp. 269–275, 2015, [Online]. Available: <https://scielo.conicyt.cl/pdf/ingeniare/v23n2/art11.pdf>
- [117] D. J. Matich, “Redes Neuronales: Conceptos Básicos y Aplicaciones.,” *Univ. Tecnológica Nac.*, vol. 41, pp. 12–16, 2001.
- [118] E. X. Puga Reyes, “Análisis de algoritmos de Clustering basados en particionamiento y basados en Densidad para el procesamiento de trayectorias GPS,” Universidad Guayaquil, 2020.
- [119] Redacción España, “Cómo funcionan los algoritmos de Inteligencia Artificial Máquinas que piensen como seres humanos. Es la aspiración última de la Inteligencia Artificial.

- Pero, ¿cómo se consigue eso? El algoritmo es la clave. Te lo contamos.,” *B12 Tech4Business*, 2020. <https://agenciab12.com/noticia/como-funcionan-algoritmos-inteligencia-artificial> (accessed Apr. 07, 2020).
- [120] M. Silva, “Aprendizaje por Refuerzo: Planificando con Programación Dinámica,” *Medium*, 2019. <https://medium.com/aprendizaje-por-refuerzo-introducción-al-mundo-del/aprendizaje-por-refuerzo-planificando-con-programación-dinámica-200ebd2af48f> (accessed Feb. 14, 2022).
- [121] R. R. Caponera de la Cobellis, “Comparación de algoritmos de aprendizaje por refuerzo basados en Q-Learning Comparison,” Universidad de Málaga, Escuela Técnica Superior de Ingeniería en Informática, 2021.
- [122] A. J. García Carrasco, “Estudio experimental de diversos algoritmos de aprendizaje por refuerzo.,” Universidad de Málaga, Escuela Técnica Superior de Ingeniería en Informática, 2020. [Online]. Available: <https://hdl.handle.net/10630/20718>
- [123] J. García Ramirez, E. Morales, and H. J. Escalante, “Transferencia de Conocimiento Utilizando Múltiples Tareas en Algoritmos de Aprendizaje por Refuerzo Profundo,” Puebla, México, 2019.
- [124] A. M. Serrano, “Aprendizaje por refuerzo: Fundamentos teóricos del algoritmo AlphaZero e implementación,” Universidad Complutense de Madrid, 2021.
- [125] H. F. Vallejo Ballesteros, E. Guevara Iñiguez, and S. R. Medina Velasco, “Minería de Datos,” *Rev. Científica Mundo la Investig. y el Conoc. Recimundo*, vol. 2, no. Esp, pp. 339–349, 2018.
- [126] P. E. Camacho Chacón, A. Zapata González, V. H. Menéndez Domínguez, and P. J. Canto Herrera, “Análisis del desempeño del profesorado universitario en el uso de MOODLE a través de técnicas de minería de datos: propuestas de necesidades formativas,” *Rev. Educ. a Distancia*, no. 58, 2018, doi: <http://dx.doi.org/10.6018/red/58/10>.
- [127] A. I. Valero Moreno, “Técnicas estadísticas en Minería de Textos,” Universidad de Sevilla, 2017. [Online]. Available: <http://hdl.handle.net/11441/63197>
- [128] C. E. Marulanda Echeverry, M. López Trujillo, and M. H. Mejía Salazar, “Minería de datos en gestión del conocimiento de pymes de Colombia,” *Rev. Virtual Univ. Católica del Norte*, no. 50, pp. 224–237, 2017.

- [129] J. C. Mérida Sánchez, “Adaptación De Estándares De Dirección De Proyectos Particularizados Para La Minería De Datos,” Universidad de Oviedo, 2017. [Online]. Available: <http://hdl.handle.net/10651/43633>
- [130] M. I. Uvidia Fassler, A. S. Cisneros Barahona, and J. A. Viñan Carrera, “Minería de datos de la evaluación integral del desempeño académico de la unidad de nivelación,” *Rev. mktDescubre - ESPOCH FADE*, no. 9, pp. 44–54, 2017, [Online]. Available: <https://core.ac.uk/download/pdf/234577131.pdf>
- [131] A. Reyes Nava, A. Flores Fuentes, A. Roberto, and E. Rendón Lara, “Minería de datos aplicada para la identificación de factores de riesgo en alumnos,” *Res. Comput. Sci.*, vol. 139, no. 1, pp. 177–189, 2017.
- [132] J. J. Espinosa Zúñiga, “Aplicación de metodología CRISP-DM para segmentación geográfica de una base de datos pública,” *Ing. Investig. y Tecnol.*, vol. 21, no. 1, pp. 1–17, 2020, doi: 10.22201/fi.25940732e.2020.21n1.008.
- [133] V. Galán Cortina, “Aplicación de la metodología CRISP-DM a un proyecto de minería de datos en el entorno universitario,” Universidad Carlos III de Madrid, 2015. [Online]. Available: <https://e-archivo.uc3m.es/handle/10016/22198>
- [134] E. A. Oviedo Carrascal, A. I. Oviedo Carrascal, and G. L. Vélez Saldarriaga, “Minería de Datos: Aportes y Tendencias en el Servicio de Salud de Ciudades Intelientes,” *Rev. Politécnica*, vol. 11, no. 20, pp. 111–120, 2015, doi: 10.33333/rp.vol49n1.
- [135] J. Mi. Moine, A. S. Haedo, and S. Gordillo, “Estudio comparativo de metodologías para minería de datos,” in *XIII Workshop de Investigadores en Ciencias de la Computación*, 2011, p. 4. [Online]. Available: <http://sedici.unlp.edu.ar/handle/10915/20034>
- [136] H. W. Pacheco Pazmiño, “Estudio de algoritmos de filtrado basado en contenidos para sistemas recomendadores de información,” Universidad Técnica de Cotopaxi, 2018. [Online]. Available: <http://repositorio.utc.edu.ec/handle/27000/5611>
- [137] J. A. Chuquitarco Chasiluisa, “Estudio de algoritmos de filtrado colaborativo para sistemas recomendadores de información,” Universidad Técnica de Cotopaxi, 2018. [Online]. Available: <http://repositorio.utc.edu.ec/handle/27000/5630>
- [138] E. I. Aispur Calvopiña and E. L. Medina Chiluisa, “Implementación de una aplicación móvil para la visualización de la producción científica de la Universidad Técnica de

- Cotopaxi.” Universidad Técnica de Cotopaxi, 2018. [Online]. Available: <http://repositorio.utc.edu.ec/handle/27000/5634>
- [139] J. J. Allauca Chaquina and E. M. Chicaiza Haro, “Aplicación de algoritmo de extracción de textos en los perfiles de usuarios en caso de los investigadores de la Universidad Técnica de Cotopaxi.” Universidad Técnica de Cotopaxi, 2019. [Online]. Available: <http://repositorio.utc.edu.ec/handle/27000/5752>
- [140] D. G. Falconí Punguil and J. N. Gualpa Mendoza, “Método para la determinación de similitud y distancia entre investigadores a partir de algoritmos de clasificación.” Universidad Técnica de Cotopaxi, 2019. [Online]. Available: <http://repositorio.utc.edu.ec/handle/27000/5698>
- [141] E. V. Gamboy Pinales and O. J. Yunda Cando, “Implementación de un algoritmo para la evaluación de los documentos científicos de los investigadores de la universidad técnica de Cotopaxi,” Universidad Técnica de Cotopaxi, 2019. [Online]. Available: <http://repositorio.utc.edu.ec/handle/27000/5335>
- [142] L. G. Ayala Raura and J. D. Chiluisa Gallardo, “Implementación de un algoritmo de lógica difusa en la identificación de incentivos y recomendaciones para los investigadores de la Universidad Técnica de Cotopaxi,” Universidad Técnica de Cotopaxi, 2020. [Online]. Available: <http://repositorio.utc.edu.ec/handle/27000/6704>
- [143] J. D. Moya Caisa and L. C. Tapia Vega, “Un enfoque de machine Learning en el desarrollo de sistemas recomendadores,” Universidad Técnica de Cotopaxi, 2020. [Online]. Available: <http://repositorio.utc.edu.ec/bitstream/27000/6686/1/T-001489.pdf>
- [144] G. A. Chariguaman Morocho and N. L. Quilumbaquin Tutillo, “Procedimiento Algorítmico Basado en Técnicas del Procesamiento de Lenguaje Natural para el Análisis del Corpus de Artículos Científicos de la Plataforma EcuCiencia.” Universidad Técnica de Cotopaxi, 2020.
- [145] A. C. Robles Campoverde and L. R. Velasco Pillajo, “Implementación de Metadatos Dublin Core y Protocolos de Interoperabilidad para la Plataforma Científica EcuCiencia de la Universidad Técnica de Cotopaxi.” Universidad Técnica de Cotopaxi, 2020.
- [146] E. J. Estrella Guerrero and E. S. López Liquinchana, “Desarrollo de los Módulos de Certificación y Autoevaluación, Investigador, mediante Inferencia de los Datos

Extraídos del Proyecto EcuCiencia de la Universidad Técnica de Cotopaxi,” Universidad Técnica de Cotopaxi, 2020.

- [147] G. L. Jaramillo Tenezaca and J. C. Osorio Quispe, “DESARROLLO DE LOS MÓDULOS DE GESTIÓN DE PROYECTOS Y GRUPOS DE INVESTIGACIÓN EN LA PLATAFORMA ECUCIENCIA,” Universidad Técnica de Cotopaxi, 2021. [Online]. Available: <http://repositorio.utc.edu.ec/bitstream/27000/4501/1/PI-000727.pdf>
- [148] E. M. Chango Holguin, “Método de Análisis De Redes Sociales Para Identificar Relaciones y Colaboraciones Científicas Entre Investigadores de la Universidad Técnica de Cotopaxi.,” Universidad Técnica de Cotopaxi, 2020. [Online]. Available: <http://repositorio.utc.edu.ec/handle/27000/6954>
- [149] S. H. Corrales Beltrán, “Métodos para el análisis de la información en corpus de artículos científicos con algoritmos de clasificación y librerías NLTK en la Plataforma Científica ECUCIENCIA,” Universidad Técnica de Cotopaxi, 2020. [Online]. Available: <http://repositorio.utc.edu.ec/handle/27000/7234>
- [150] L. R. Quisaguano Collaguazo, “Sistema de georreferenciación para el análisis de producción científica de los docentes investigadores de la Universidad Técnica de Cotopaxi,” Universidad Técnica de Cotopaxi, 2020. [Online]. Available: <http://repositorio.utc.edu.ec/handle/27000/6955>
- [151] A. G. Rivera Chasiquiza, “Visualización de Información mediante mapeo auto-organizado en datos de producción científica de la Universidad Técnica de Cotopaxi.,” Universidad Técnica de Cotopaxi., 2020. [Online]. Available: <http://repositorio.utc.edu.ec/handle/27000/7233>
- [152] D. G. Falconí Punguil, “Algoritmos de Deep Learning utilizando Tensor Flow para el tratamiento de datos de producción científica en la Universidad Técnica de Cotopaxi,” Universidad Técnica de Cotopaxi, 2021. [Online]. Available: <http://repositorio.utc.edu.ec/handle/27000/7764>
- [153] K. O. Egas Arizala and T. F. Roque Colt, “Diseño de un Modelo Predictivo Basado en Técnicas de Machine Learning que permita determinar la Temperatura usando los datos de una Mini Estación Meteorológica en la Ciudad De Guayaquil,” Universidad de Guayaquil, Facultad de ciencias Matemáticas y Físicas, 2020.

- [154] P. A. Plaza Navas, “Desarrollo de un sistema web para el soporte técnico remoto de primer nivel, orientado a la gestión de incidentes informáticos, basado en inteligencia artificial,” Universidad de Guayaquil, Facultad de Ciencias Matemáticas y Físicas, 2017. [Online]. Available: <http://repositorio.ug.edu.ec/bitstream/redug/24301/1/B-CISC-PTG.1408.PlazaNavasPabloAndres.pdf>
- [155] D. A. Pacheco Jaimes and S. A. Laguado Sequeda, “Prototipo de herramienta Software para la generación de un pronóstico de inventario mediante algoritmos de Inteligencia Artificial: Caso de estudio Producción de Cacao.,” Universidad Autónoma de Bucaramanga, Programa de Ingeniería en Sistemas, Facultad de Ingenierías, 2020. [Online]. Available: https://repository.unab.edu.co/bitstream/handle/20.500.12749/15361/2020_Tesis_Daniel_Andres_Pacheco_Jaimes.pdf?sequence=1&isAllowed=y
- [156] J. F. Limaico Mera, “DESARROLLO DE UN SISTEMA RECOMENDADOR DE UBICACIONES PARA SITIOS TURÍSTICOS DEL CENTRO HISTÓRICO DE QUITO BASADO EN PROCESAMIENTO DE LENGUAJE NATURAL,” Escuela Politécnica Nacional, 2020.
- [157] V. E. Mora Chunllo, “Diseño e implementación de un modelo software basado en técnicas de inteligencia artificial, para predecir el índice de radiación solar en Riobamba-Ecuador,” Universidad de las Fuerzas Armadas, 2015. [Online]. Available: <http://repositorio.espe.edu.ec/handle/21000/12216>
- [158] M. S. Camacho Gavilánez and L. M. Guanotuña Topon, “Desarrollo de una aplicación para evaluar el calibre de arterias y venas a partir de retinografías, aplicando técnicas de inteligencia artificial,” Universidad de las Fuerzas Armadas, 2020. [Online]. Available: <http://repositorio.espe.edu.ec/handle/21000/22144>
- [159] C. H. Llerena Valencia, “Prototipo de un chatbot para un sistema de servicios vehiculares utilizando Adaptive Cards con Bot Framework,” Universidad Técnica de Cotopaxi, Facultad de Ingeniería en Sistemas, Electrónica e Industrial, 2021. [Online]. Available: <https://repositorio.uta.edu.ec/jspui/handle/123456789/33208>
- [160] J. E. Erazo Sánchez, “Desarrollo de un algoritmo híbrido para recomendaciones de itinerarios turísticos de acuerdo con las preferencias de los usuarios,” Escuela Politécnica Nacional, 2019. [Online]. Available: <http://bibdigital.epn.edu.ec/handle/15000/20356>

- [161] J. P. Subra and A. Vannieuwenhuyse, *Scrum: un método ágil para sus proyectos*. Ediciones ENI, 2018. [Online]. Available: https://books.google.es/books?hl=es&lr=&id=TyQuFpGhZ8sC&oi=fnd&pg=PA13&dq=fases+de+scrum&ots=_7eOZ2EgUu&sig=rhqusU8pPde3bgWuwUZBTRzeOCc#v=onepage&q=fases+de+scrum&f=false
- [162] N. K. Rad and F. Turley, *Los Fundamentos de Agile Scrum*. Van Haren, 2019. [Online]. Available: https://scholar.google.com/scholar?hl=es&as_sdt=0%2C5&q=Los+Fundamentos+de+Agile+Scrum&btnG=#d=gs_cit&u=%2Fscholar%3Fq%3Dinfo%3AjN0B119ooHYJ%3Ascholar.google.com%2F%26output%3Dcite%26scirp%3D0%26hl%3Des
- [163] L. Castillo Vidal, “Resultados preliminares más significativos tras cuatro años de aplicación de la metodología SCRUM en las prácticas de laboratorio,” *Rev. Investig. en Docencia Univ. la Informática*, vol. 11, no. 1, p. 8, 2018.
- [164] S. M. Meléndez Valladarez, M. E. Gaitan, and N. N. Pérez Reyes, “METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACION EXTREMA,” Universidad Nacional Autónoma De Nicaragua, Managua UNAN-MANAGUA, 2016.
- [165] J. C. Salazar, Á. Tovar, J. C. Linares, A. Lozano, and L. Valbuena, “Scrum versus XP: similitudes y diferencias,” *Tecnol. Investig. y Acad.*, vol. 6, no. 2, pp. 29–37, 2018, [Online]. Available: <https://revistas.udistrital.edu.co/index.php/tia/article/view/10496>
- [166] L. M. Echeverry Tobón and L. E. Delgado Carmona, “Caso práctico de la metodología ágil XP al desarrollo de software,” Universidad Tecnológica de Pereira, 2007.
- [167] A. Navarro Cadavid, J. D. Fernández Martínez, and J. Morales Vélez, “Revisión de metodologías ágiles para el desarrollo de software,” *redalyc*, vol. 11, no. 2, pp. 30–39, 2013, [Online]. Available: <http://www.redalyc.org/articulo.oa?id=496250736004>
- [168] D. Pérez Ramírez, “Metodologías Ágiles. ¿Cómo desarrollo utilizando XP?,” in *Convención Científica de Ingeniería y Arquitectura*, 2008, pp. 1–6.
- [169] M. Bermejo, *El Kanban*. Barcelona, España: UOC, 2011.
- [170] S. Ruiz Ayuso, “Identificando el ámbito de aplicación de Lean IT , Scrum y Kanban ...,” *Cap. Hum. Rev. para la Integr. y Desarro. los Recur. humanos*, no. 35, p. 12, 2020, [Online]. Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=7607022>

- [171] “Kanban,” *crisp*. <https://www.crisp.se/gratis-material-och-guider/kanban> (accessed Mar. 22, 2022).
- [172] P. Letier, “Organización ágil del trabajo: Parte II - Dos dimensiones del trabajo.” <http://agilismoatwork.blogspot.com/search/label/Workflows> (accessed Mar. 22, 2022).
- [173] A. O. Duarte and M. Rojas, “Las Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo,” *Revista Avances en Sistemas e Informática*, vol. 5, no. 2, pp. 159–171, 2008.
- [174] R. E. López Menéndez de Jiménez, “Metodologías ágiles de desarrollo de software aplicadas a la gestión de proyectos empresariales,” *Rev. tecnológica*, vol. 8, no. Escuela Especializada en Ingeniería ITCA-FEPADE, p. 6, 2015, [Online]. Available: http://fcaenlinea.unam.mx/anexos/1728/Unidad_1/u1_act2_2.pdf
- [175] M. A. Álvarez Báez, “Desarrollo de una mejora en la metodología V en la fase de pruebas de software implementado en el campo de la aeronáutica,” 2017. [Online]. Available: <http://ciateq.repositorioinstitucional.mx/jspui/handle/1020/125>
- [176] D. R. Cardozzo, *Desarrollo de Software: requisitos, estimaciones y análisis*, Segunda Ed. 2016.
- [177] C. Pérez López and D. Santín González, *Minería de datos. Técnicas y herramientas: técnicas y herramientas*. Editorial Paraninfo, 2007.
- [178] H. A. Flórez Fernández, “Procesos De Ingeniería De Software,” *Rev. vínculos*, vol. 6, no. 1, pp. 1–14, 2009.
- [179] A. Martínez and R. Martínez, “Guía a Rational Unified Process,” 2014.
- [180] T. Gustavo, “El Proceso Unificado de Desarrollo de Software,” 2002.
- [181] O. M. Sposito, M. E. Etcheverry, H. L. Ryckeboer, and J. Bossero, “Aplicación de técnicas de minería de datos para la evaluación del rendimiento académico y la deserción estudiantil,” *Novena Conf. Iberoam. en Sist. cibernética e informática*, vol. 29, pp. 06–2, 2010.
- [182] E. G. Maida and J. Pacienza, “Metodologías de desarrollo de software,” Pontificia Universidad Católica Argentina, 2015. [Online]. Available: <https://repositorio.uca.edu.ar/handle/123456789/522>

- [183] M. L. Araque Jiménez, “Gestión De Riesgos En Proyecto De Software a Desarrollar En Empresa Privada,” Universidad Militar Nueva Granada, 2015. [Online]. Available: <http://hdl.handle.net/10654/7671>
- [184] F. Sancho Caparrini, “El modelado de problemas,” 2019. <http://www.cs.us.es/~fsancho/?e=94> (accessed Mar. 06, 2022).
- [185] R. Noriega Martínez, *El Proceso de Desarrollo de Software*. 2015.
- [186] M. S. Chillán Zulca and N. M. Lozano Pushug, “Elaboración de una guía de procedimientos para la fase de pruebas en el desarrollo de software,” Escuela Politécnica Nacional, 2017. [Online]. Available: <https://bibdigital.epn.edu.ec/handle/15000/18767>
- [187] B. Beltrán Martínez, “Minería de datos,” *Cómo hallar una aguja en un pajar. Ing.*, vol. 14, no. 53, pp. 53–66, 2001, [Online]. Available: <https://www.cs.buap.mx/~bbeltran/NotasMD.pdf>
- [188] F. Alonso, M. Loic, and J. Segovia, *Introducción a la Ingeniería de Software: Modelos de desarrollo de programas*. DELTA PUBLICACIONES, 2005.
- [189] J. M. García Rodríguez, “Estudio comparativo entre las metodologías ágiles y las metodologías tradicionales para la gestión de proyectos software,” Universidad de Oviedo, 2015. [Online]. Available: <http://hdl.handle.net/10651/32457>
- [190] L. del C. Ramírez and A. S. Flórez Fuentes, “Desarrollo De Software Good Practice , a Solution for Better Software,” *Rev. Mundo FESC*, vol. 4, no. 8, pp. 37–45, 2014.
- [191] F. H. Troncoso Espinosa and P. G. Fuentes Figueroa, “Predicción De Fraudes En El Consumo De Agua Potable Mediante El Uso De Minería De Datos,” *Univ. Cienc. y Tecnol.*, vol. 24, no. 104, pp. 58–66, 2020, doi: 10.47460/uct.v24i104.366.
- [192] J. S. Burgos Vargas and K. L. González Cubero, “IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN BASADO EN LA METODOLOGÍA KDD DE MINERÍA DE DATOS PARA EL PROCESO DE PLANIFICACIÓN DE LA PRODUCCIÓN DE LA INDUSTRIA FARMACÉUTICA ECUATORIANA.” Universidad de Guayaquil, 2020. [Online]. Available: <http://repositorio.ug.edu.ec/handle/redug/49567>
- [193] M. L. Tello, H. J. Eslava, and L. B. Tobías, “Análisis y evaluación del nivel de riesgo en el otorgamiento de créditos financieros utilizando técnicas de minería de datos Analysis and evaluation of risk levels on loan approval using data mining techniques Resumen En

este artículo se presenta la aplica,” *Rev. Univ. distritalista*, vol. 7, no. 1, 2013, [Online]. Available:

<https://revistas.udistrital.edu.co/index.php/visele/article/view/4389/6093>

- [194] S. R. Timarán Pereira, I. Hernández Arteaga, S. J. Caicedo Zambrano, A. Hidalgo Troya, and J. C. Alvarado Pérez, “El proceso de descubrimiento de conocimiento en bases de datos,” in *Descubrimiento de patrones de desempeño académico con árboles de decisión en las competencias genéricas de la formación profesional*, Ediciones., Bogotá, 2016, pp. 63–86. doi: <http://dx.doi.org/10.16925/9789587600490>.
- [195] S. García, S. Ramírez Gallego, J. Luengo, and F. Herrera, “Big Data: Preprocesamiento y calidad de datos,” Universidad de Granada, 2016. [Online]. Available: http://sci2s.ugr.es/sites/default/files/ficherosPublicaciones/2133_Nv237-Digital-sramirez.pdf
- [196] J. S. Burgos Vargas and K. L. González Cubero, “Implementación de un Sistema de Gestión Basado en la Metodología KDD de Minería de Datos para el proceso de Planificación de la Producción de la Industria Farmacéutica Ecuatoriana,” Universidad de Guayaquil, 2020.
- [197] A. Jaramillo and H. P. Paz Arias, “Aplicación de Técnicas de Minería de Datos para Determinar las Interacciones de los Estudiantes en un Entorno Virtual de Aprendizaje,” *Rev. Tecnológica ESPOL – RTE*, vol. 28, no. 1, pp. 64–90, 2015, [Online]. Available: <http://www.rte.espol.edu.ec/index.php/tecnologica/article/view/351>
- [198] M. Niño, “CRISP-DM: Fase de ‘Modelado’ (Modeling),” 2016. <http://www.mikelnino.com/2016/11/crisp-dm-metodologia-data-mining-modelado-modeling.html>
- [199] F. O. Rodríguez Miranda, “Implementación de KDD para mejorar el Proceso de Identificación de Estilos de Aprendizaje en la Universidad Autónoma del Perú,” Universidad Autónoma del Perú, 2018.
- [200] V. Lopez Güemes, “Business Intelligence para la toma de decisiones estratégicas: Un caso de aplicación de Minería de datos dentro del sector bancario,” Universidad de la Cantabria, 2018. [Online]. Available: <http://hdl.handle.net/10902/17546>

6. ANEXOS

ENCUESTA PARA DETERMINAR FACTORES QUE INCIDEN EN LA INTEGRACIÓN DE LA INTELIGENCIA ARTIFICIAL CON LA INGENIERÍA DE SOFTWARE

Esta encuesta permitirá descubrir factores que influyen en la integración de la Inteligencia Artificial con el Desarrollo de Software. Esta encuesta está dirigida a personas que tengan experiencia en el área de Inteligencia Artificial y Desarrollo de Software, las respuestas que se obtengan de este cuestionario serán utilizadas estrictamente con fines de investigación. La encuesta está dividida en 3 secciones. La sección 1 está relacionada con la información de las personas que llenarán la encuesta. La sección 2 corresponde a preguntas que determinen la influencia de los factores que inciden en la integración de la Inteligencia Artificial con el Desarrollo de Software. La sección 3 contiene preguntas que complementan el estudio.

GRACIAS POR SU COLABORACIÓN.

Sección 1: Información General

1. Edad: _____

2. Género

Masculino	Femenino

3. Área de especialización o profesionalización

Inteligencia Artificial	
Sistemas e Informática	
Desarrollo de Software	

4. Experiencia

1 – 3 años	
4 – 6 años	
7 – 9 años	
Más de 10 años	

Sección 2: Factores que influyen en la integración del Desarrollo de Software con Inteligencia Artificial.

Conteste las siguientes preguntas de acuerdo la escala del 1 al 3 considerando el nivel de influencia:

1 = No influye

2 = Baja influencia

3 = Alta influencia

1. Considera usted que en la fase de Planificación influye en la integración del Desarrollo de Software con Inteligencia Artificial.

1 = No influye	2 = Baja influencia	3 = Alta influencia

2. Considera usted que la fase de Análisis influye en la integración del Desarrollo de Software con Inteligencia Artificial.

1 = No influye	2 = Baja influencia	3 = Alta influencia

3. Considera usted que la fase Iteraciones influye en la integración del Desarrollo de Software con Inteligencia Artificial.

1 = No influye	2 = Baja influencia	3 = Alta influencia

4. Considera usted que la fase de Desarrollo influye en la integración del Desarrollo de Software con Inteligencia Artificial.

1 = No influye	2 = Baja influencia	3 = Alta influencia

5. Considera usted que la fase de Codificación influye en la integración del Desarrollo de Software con Inteligencia Artificial.

1 = No influye	2 = Baja influencia	3 = Alta influencia

6. Considera usted que la fase de Pruebas influye en la integración del Desarrollo de Software con Inteligencia Artificial.

1 = No influye	2 = Baja influencia	3 = Alta influencia

7. Considera usted que la fase de Mantenimiento del software influye en la integración del Desarrollo de Software con Inteligencia Artificial.

1 = No influye	2 = Baja influencia	3 = Alta influencia

8. Considera usted que la fase de Culminación del Proyecto influye en la integración del Desarrollo de Software con Inteligencia Artificial.

1 = No influye	2 = Baja influencia	3 = Alta influencia

Sección 3: Preguntas que complementan el estudio.

Conteste las siguientes preguntas de acuerdo la escala del 1 al 3 considerando el nivel de influencia:

- 1 = Nada importante
- 2 = Poco importante
- 3 = Muy importante

1. Considera usted importante que, en la fase de Análisis se pueda determinar un proceso de selección de variables que influyan en el desarrollo del software y la construcción de modelos de Inteligencia Artificial.

1 = Nada importante	2 = Poco importante	3 = Muy importante

2. Considera usted importante que, en la fase de Desarrollo se pueda establecer una etapa de Preprocesamiento que permita el análisis de la data y la mejora del rendimiento de los algoritmos de aprendizaje automático aplicados en el desarrollo de software.

1 = Nada importante	2 = Poco importante	3 = Muy importante

3. Considera usted importante que, en la fase de Codificación respecto al desarrollo de software, se puede establecer el modelado de algoritmos de Inteligencia Artificial.

1 = Nada importante	2 = Poco importante	3 = Muy importante

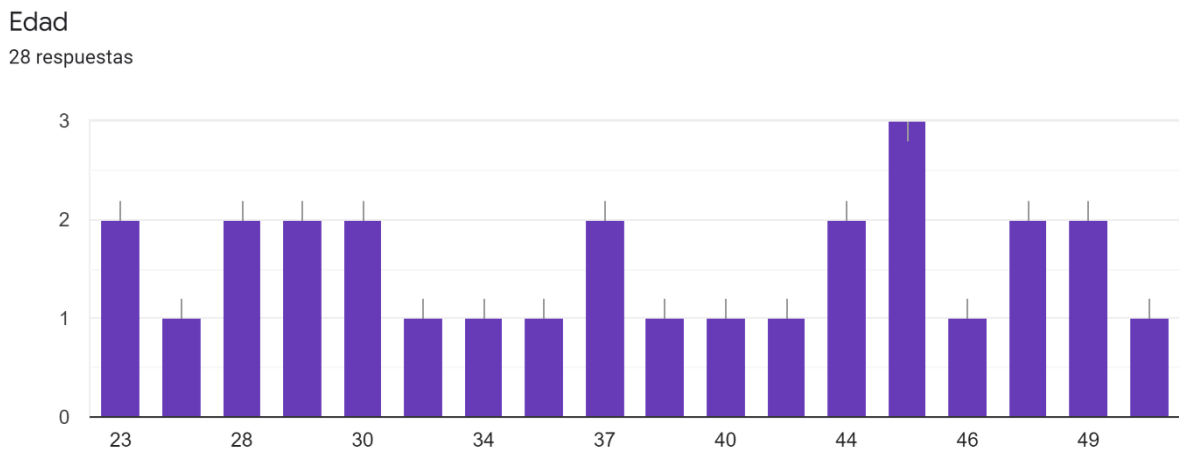
4. Considera usted importante que, en la fase de Pruebas respecto al desarrollo de software, se pueda determinar un proceso de validación de los algoritmos de aprendizaje automático estableciendo métricas estadísticas de evaluación.

1 = Nada importante	2 = Poco importante	3 = Muy importante

RESULTADOS DE LA ENCUESTA (Google Forms)

Sección 1: Información General

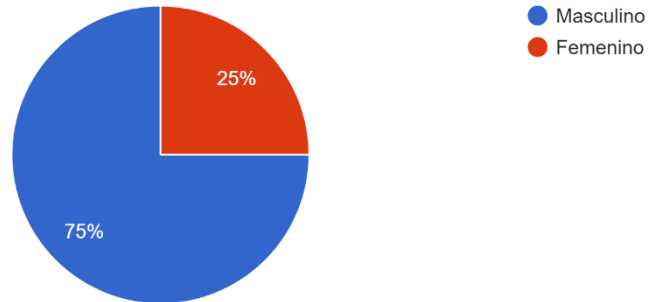
Figura 27. Resultados estadísticos de la encuesta.



Fuente: Google Forms.

Figura 28. Gráfica de resultados estadísticos por género.

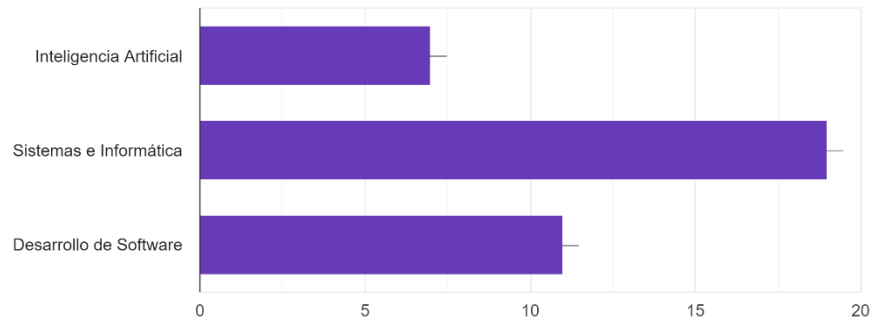
2. Género
28 respuestas



Fuente: Google Forms.

Figura 29. Gráfica de resultados estadísticos por área de especialización o profesionalización.

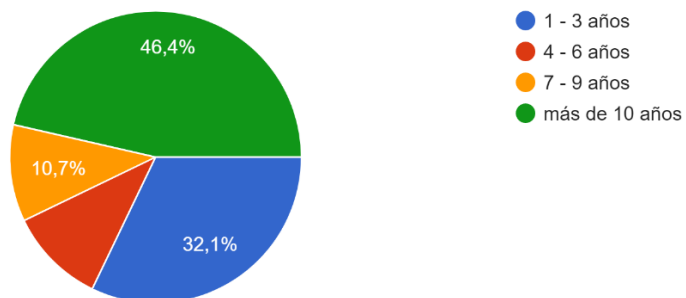
3. Área de especialización o profesionalización
28 respuestas



Fuente: Google Forms.

Figura 30. Gráfica de resultados estadísticos por años de experiencia.

4. Experiencia
28 respuestas

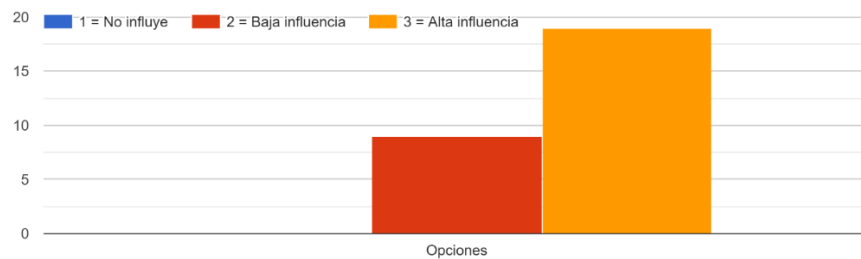


Fuente: Google Forms.

Sección 2: Factores que influyen en la integración del Desarrollo de Software con Inteligencia Artificial.

Figura 31. Gráfica de resultados estadísticos Pregunta 1, sección 2 de la encuesta.

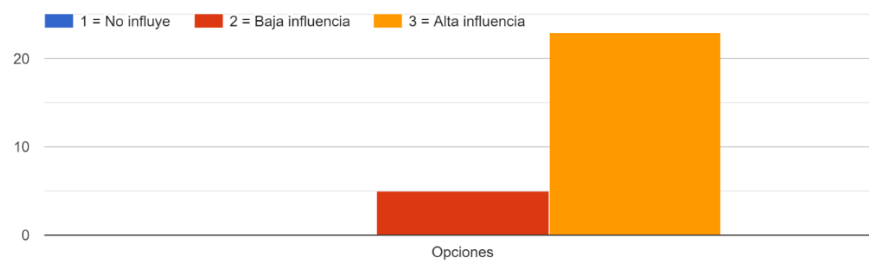
1. Considera usted que en la fase de Planificación influye en la integración del Desarrollo de Software con Inteligencia Artificial.



Fuente: Google Forms.

Figura 32. Gráfica de resultados estadísticos Pregunta 2, sección 2 de la encuesta.

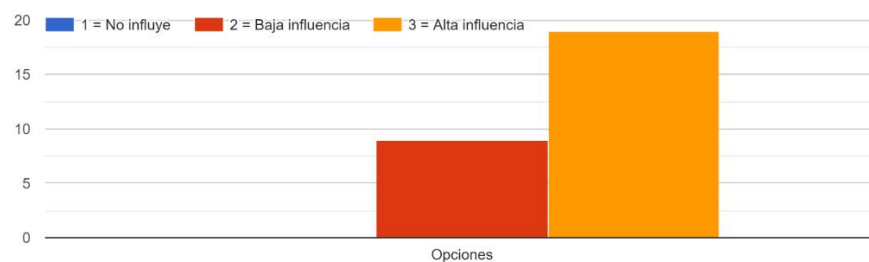
2. Considera usted que la fase de Análisis influye en la integración del Desarrollo de Software con Inteligencia Artificial.



Fuente: Google Forms.

Figura 33. Gráfica de resultados estadísticos Pregunta 3, sección 2 de la encuesta.

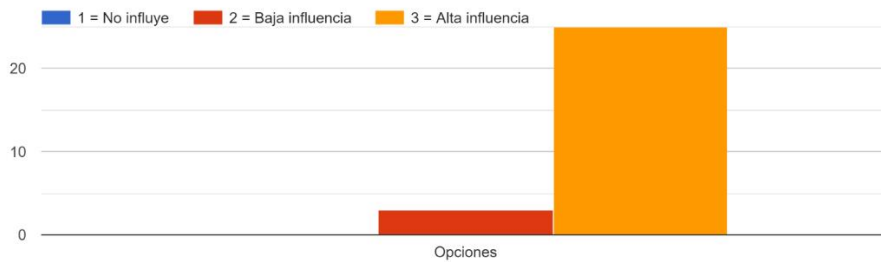
3. Considera usted que la fase Iteraciones influye en la integración del Desarrollo de Software con Inteligencia Artificial.



Fuente: Google Forms.

Figura 34. Gráfica de resultados estadísticos Pregunta 4, sección 2 de la encuesta.

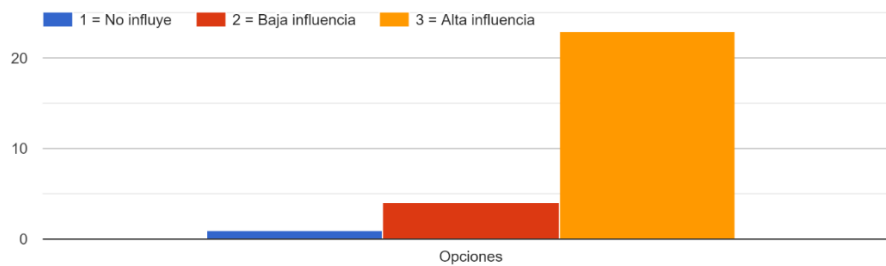
4. Considera usted que la fase de Desarrollo influye en la integración del Desarrollo de Software con Inteligencia Artificial.



Fuente: Google Forms.

Figura 35. Gráfica de resultados estadísticos Pregunta 5, sección 2 de la encuesta.

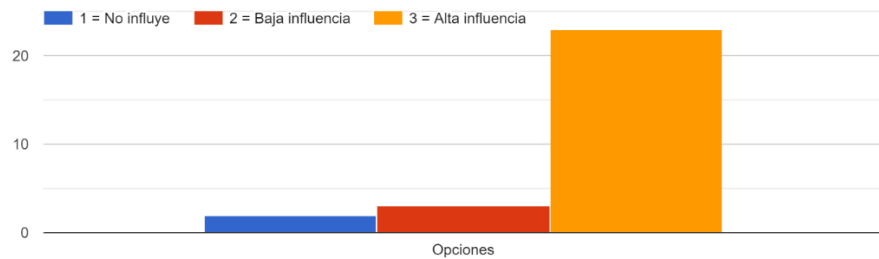
5. Considera usted que la fase de Codificación influye en la integración del Desarrollo de Software con Inteligencia Artificial.



Fuente: Google Forms.

Figura 36. Gráfica de resultados estadísticos Pregunta 6, sección 2 de la encuesta.

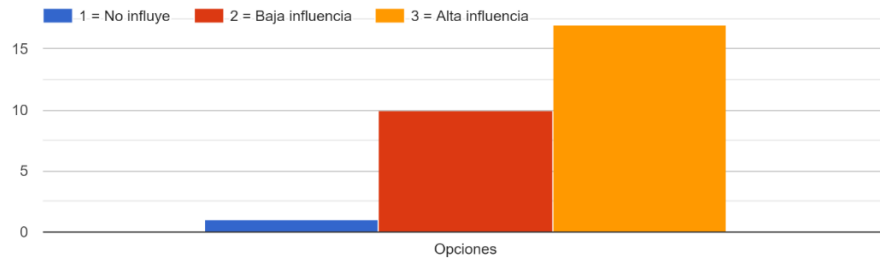
6. Considera usted que la fase de Pruebas influye en la integración del Desarrollo de Software con Inteligencia Artificial.



Fuente: Google Forms.

Figura 37. Gráfica de resultados estadísticos Pregunta 7, sección 2 de la encuesta.

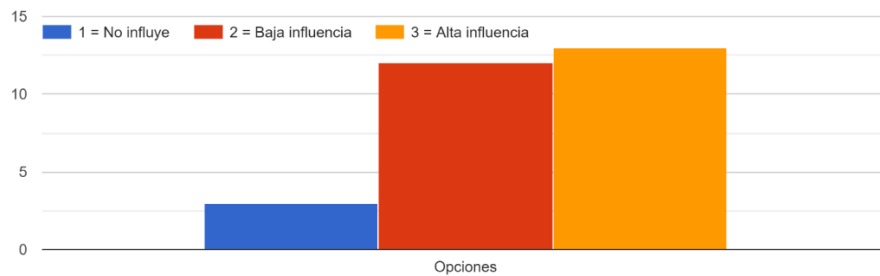
7. Considera usted que la fase de Mantenimiento del software influye en la integración del Desarrollo de Software con Inteligencia Artificial.



Fuente: Google Forms.

Figura 38. Gráfica de resultados estadísticos Pregunta 8, sección 2 de la encuesta.

8. Considera usted que la fase de Culminación del Proyecto influye en la integración del Desarrollo de Software con Inteligencia Artificial.

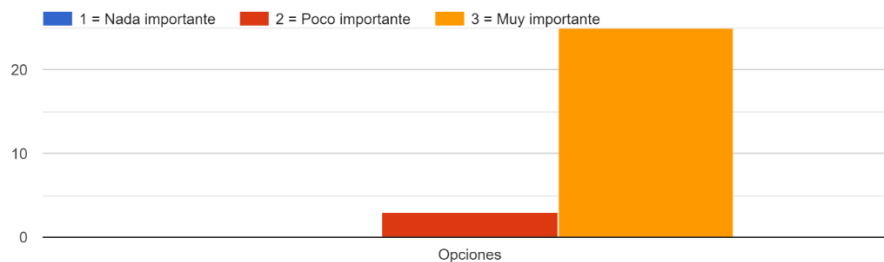


Fuente: Google Forms.

Sección 3: Preguntas que complementan el estudio.

Figura 39. Gráfica de resultados estadísticos Pregunta 1, sección 3 de la encuesta.

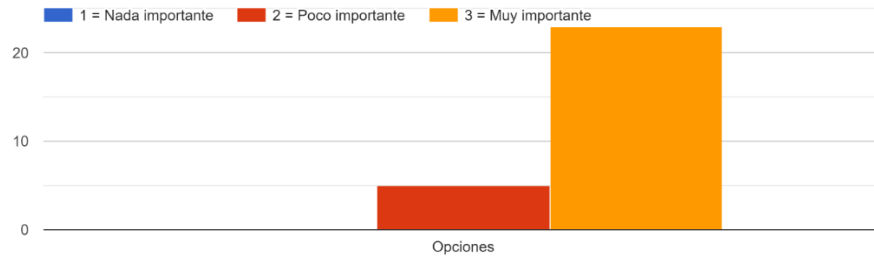
1. Considera usted importante que, en la fase de Análisis se pueda determinar un proceso de selección de variables que influyan en el desarroll...construcción de modelos de Inteligencia Artificial.



Fuente: Google Forms.

Figura 40. Gráfica de resultados estadísticos Pregunta 2, sección 3 de la encuesta.

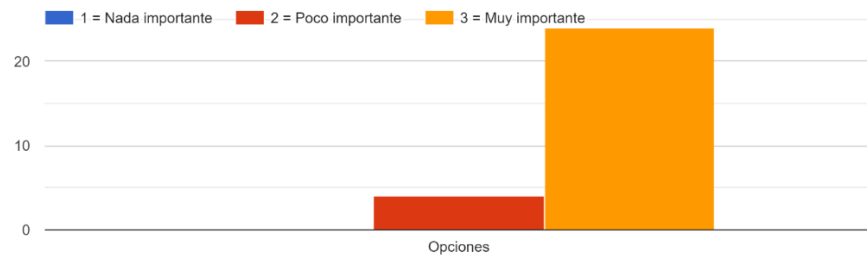
2. Considera usted importante que, en la fase de Desarrollo se pueda establecer una etapa de Preprocesamiento que permita el análisis de la dat...utomático aplicados en el desarrollo de software.



Fuente: Google Forms.

Figura 41. Gráfica de resultados estadísticos Pregunta 3, sección 3 de la encuesta.

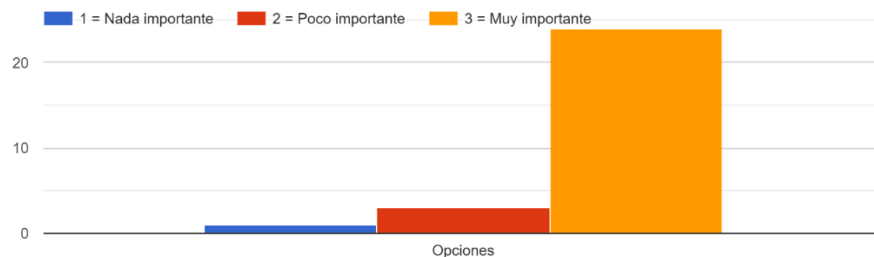
3. Considera usted importante que, en la fase de Codificación respecto al desarrollo de software, se puede establecer el modelado de algoritmos de Inteligencia Artificial.



Fuente: Google Forms.

Figura 42. Gráfica de resultados estadísticos Pregunta 3, sección 3 de la encuesta.

4. Considera usted importante que, en la fase de Pruebas respecto al desarrollo de software, se pueda determinar un proceso de validación de los ...stableciendo métricas estadísticas de evaluación.



Fuente: Google Forms.

RESULTADOS DE LA ENCUESTA (SPSS)

Tablas de resultados

- **Sección 1:** Información General de los encuestados.

Tabla 49. Resultados de la encuesta por Edad del encuestado.

Edad del encuestado					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	23	2	7,1	7,1	7,1
	25	1	3,6	3,6	10,7
	28	2	7,1	7,1	17,9
	29	2	7,1	7,1	25,0
	30	2	7,1	7,1	32,1
	32	1	3,6	3,6	35,7
	34	1	3,6	3,6	39,3
	35	1	3,6	3,6	42,9
	37	2	7,1	7,1	50,0
	38	1	3,6	3,6	53,6
	40	1	3,6	3,6	57,1
	43	1	3,6	3,6	60,7
	44	2	7,1	7,1	67,9
	45	3	10,7	10,7	78,6
	46	1	3,6	3,6	82,1
	48	2	7,1	7,1	89,3
	49	2	7,1	7,1	96,4
54	1	3,6	3,6	100,0	
Total		28	100,0	100,0	

Fuente: Elaboración propia.

Tabla 50. Resultados de la encuesta por Género del encuestado.

Género del encuestado					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Masculino	21	75,0	75,0	75,0
	Femenino	7	25,0	25,0	100,0
	Total	28	100,0	100,0	

Fuente: Elaboración propia.

Tabla 51. Resultados de la encuesta por Área de especialización o profesionalización.

Área de especialización o profesionalización					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Inteligencia Artificial	3	10,7	10,7	10,7
	Sistemas e Informática	13	46,4	46,4	57,1
	Desarrollo de Software	5	17,9	17,9	75,0
	Inteligencia Artificial, Desarrollo de Software	1	3,6	3,6	78,6
	Inteligencia Artificial, Sistemas e Informática	1	3,6	3,6	82,1
	Sistemas e Informática, Desarrollo de Software	3	10,7	10,7	92,9
	Inteligencia Artificial, Sistemas e Informática, Desarrollo de Software	2	7,1	7,1	100,0
	Total	28	100,0	100,0	

Fuente: Elaboración propia.

Tabla 52. Resultados de la encuesta por Años de experiencia del encuestado.

Años de experiencia del encuestador					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	1 a 3 años	9	32,1	32,1	32,1
	4 a 6 años	3	10,7	10,7	42,9
	7 a 9 años	3	10,7	10,7	53,6
	más de 10 años	13	46,4	46,4	100,0
	Total	28	100,0	100,0	

Fuente: Elaboración propia.

- **Sección 2:** Factores que influyen en la integración del desarrollo de software con inteligencia artificial.

Tabla 53. Resultados de la pregunta 1, sección 2 de la encuesta.

Considera usted que en la fase de Planificación influye en la integración del Desarrollo de Software con Inteligencia Artificial					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Baja influencia	9	32,1	32,1	32,1
	Alta influencia	19	67,9	67,9	100,0
	Total	28	100,0	100,0	

Fuente: Elaboración propia.

Tabla 54. Resultados de la pregunta 2, sección 2 de la encuesta.

Considera usted que la fase de Análisis influye en la integración del Desarrollo de Software con Inteligencia Artificial					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Baja influencia	5	17,9	17,9	17,9
	Alta influencia	23	82,1	82,1	100,0
	Total	28	100,0	100,0	

Fuente: Elaboración propia.

Tabla 55. Resultados de la pregunta 3, sección 2 de la encuesta.

Considera usted que la fase Iteraciones influye en la integración del Desarrollo de Software con Inteligencia Artificial.					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Baja influencia	9	32,1	32,1	32,1
	Alta influencia	19	67,9	67,9	100,0
	Total	28	100,0	100,0	

Fuente: Elaboración propia.

Tabla 56. Resultados de la pregunta 4, sección 2 de la encuesta.

Considera usted que la fase de Desarrollo influye en la integración del Desarrollo de Software con Inteligencia Artificial.					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Baja influencia	3	10,7	10,7	10,7
	Alta influencia	25	89,3	89,3	100,0
	Total	28	100,0	100,0	

Fuente: Elaboración propia.

Tabla 57. Resultados de la pregunta 5, sección 2 de la encuesta.

Considera usted que la fase de Codificación influye en la integración del Desarrollo de Software con Inteligencia Artificial.					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	No influye	1	3,6	3,6	3,6
	Baja influencia	4	14,3	14,3	17,9
	Alta influencia	23	82,1	82,1	100,0
	Total	28	100,0	100,0	

Fuente: Elaboración propia.

Tabla 58. Resultados de la pregunta 6, sección 2 de la encuesta.

Considera usted que la fase de Pruebas influye en la integración del Desarrollo de Software con Inteligencia Artificial.					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	No influye	2	7,1	7,1	7,1
	Baja influencia	3	10,7	10,7	17,9
	Alta influencia	23	82,1	82,1	100,0
	Total	28	100,0	100,0	

Fuente: Elaboración propia.

Tabla 59. Resultados de la pregunta 7, sección 2 de la encuesta.

Considera usted que la fase de Mantenimiento del software influye en la integración del Desarrollo de Software con Inteligencia Artificial.					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	No influye	1	3,6	3,6	3,6
	Baja influencia	10	35,7	35,7	39,3
	Alta influencia	17	60,7	60,7	100,0
	Total	28	100,0	100,0	

Fuente: Elaboración propia.

Tabla 60. Resultados de la pregunta 8, sección 2 de la encuesta.

Considera usted que la fase de Culminación del Proyecto influye en la integración del Desarrollo de Software con Inteligencia Artificial.					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	No influye	3	10,7	10,7	10,7
	Baja influencia	12	42,9	42,9	53,6
	Alta influencia	13	46,4	46,4	100,0
	Total	28	100,0	100,0	

Fuente: Elaboración propia.

- **Sección 3:** Preguntas que complementan el estudio.

Tabla 61. Resultados de la pregunta 1, sección 3 de la encuesta.

Considera usted importante que, en la fase de Análisis se pueda determinar un proceso de selección de variables que influyan en el desarrollo del software y la construcción de modelos de Inteligencia Artificial.					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Poco importante	3	10,7	10,7	10,7
	Muy importante	25	89,3	89,3	100,0
	Total	28	100,0	100,0	

Fuente: Elaboración propia.

Tabla 62. Resultados de la pregunta 2, sección 3 de la encuesta.

Considera usted importante que, en la fase de Desarrollo se pueda establecer una etapa de Preprocesamiento que permita el análisis de la data y la mejora del rendimiento de los algoritmos de aprendizaje automático aplicados en el desarrollo de software.					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Poco importante	5	17,9	17,9	17,9
	Muy importante	23	82,1	82,1	100,0
	Total	28	100,0	100,0	

Fuente: Elaboración propia.

Tabla 63. Resultados de la pregunta 3, sección 3 de la encuesta.

Considera usted importante que, en la fase de Codificación respecto al desarrollo de software, se puede establecer el modelado de algoritmos de Inteligencia Artificial.					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Poco importante	4	14,3	14,3	14,3
	Muy importante	24	85,7	85,7	100,0
	Total	28	100,0	100,0	

Fuente: Elaboración propia.

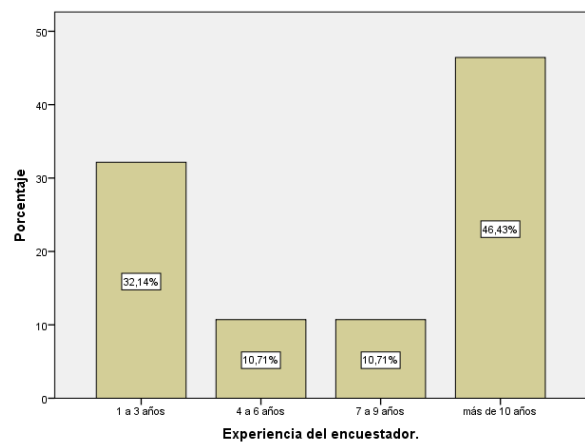
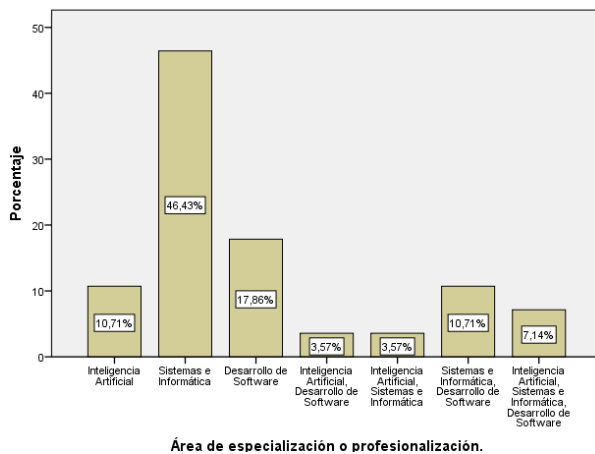
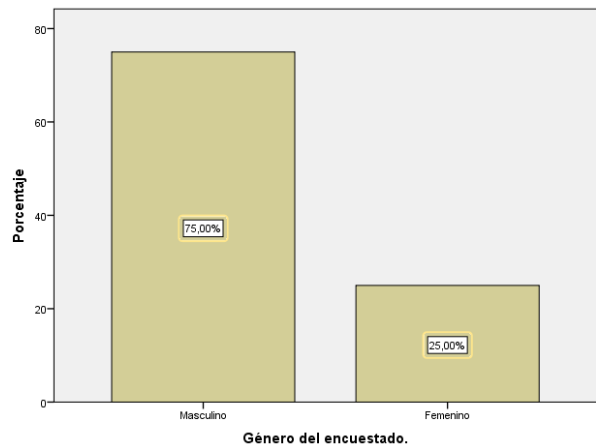
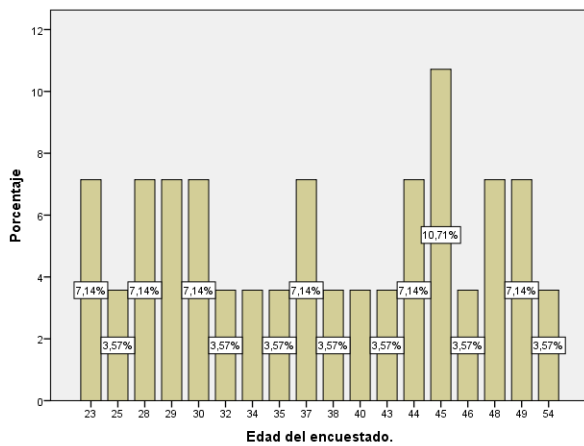
Tabla 64. Resultados de la pregunta 4, sección 3 de la encuesta.

Considera usted importante que, en la fase de Pruebas respecto al desarrollo de software, se pueda determinar un proceso de validación de los algoritmos de aprendizaje automático estableciendo métricas estadísticas de evaluación.					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Nada importante	1	3,6	3,6	3,6
	Poco importante	3	10,7	10,7	14,3
	Muy importante	24	85,7	85,7	100,0
	Total	28	100,0	100,0	

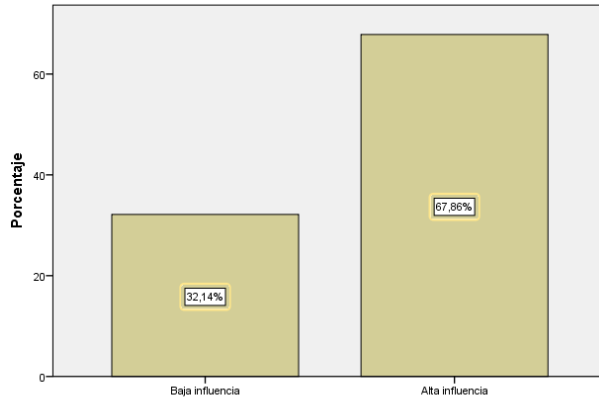
Fuente: Elaboración propia.

Gráficas Estadísticas

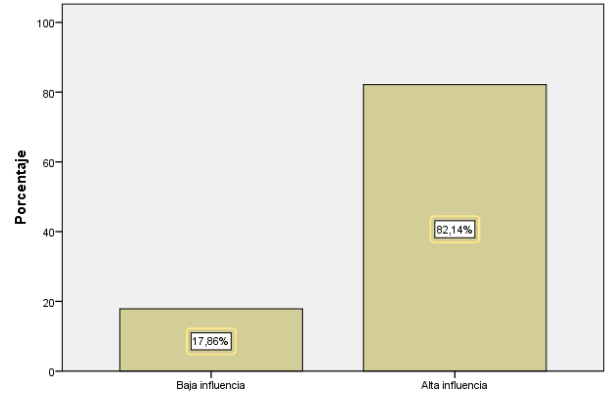
- Sección 1: Información General**



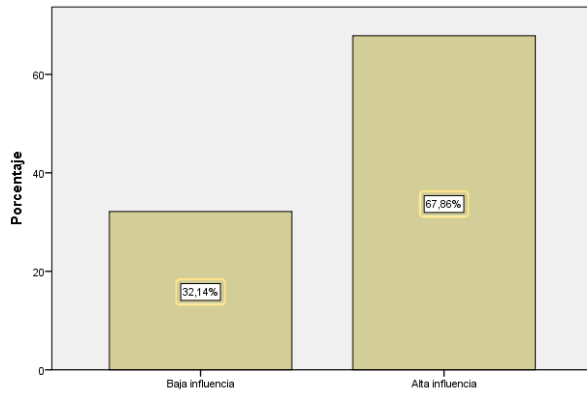
Sección 2: Factores que influyen en la integración del Desarrollo de Software con Inteligencia Artificial.



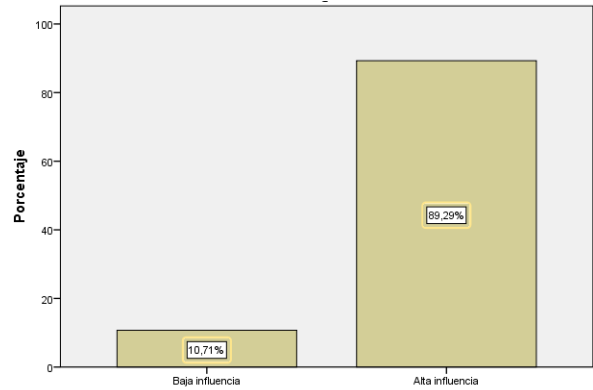
Considera usted que en la fase de Planificación influye en la integración del Desarrollo de Software con Inteligencia Artificial.



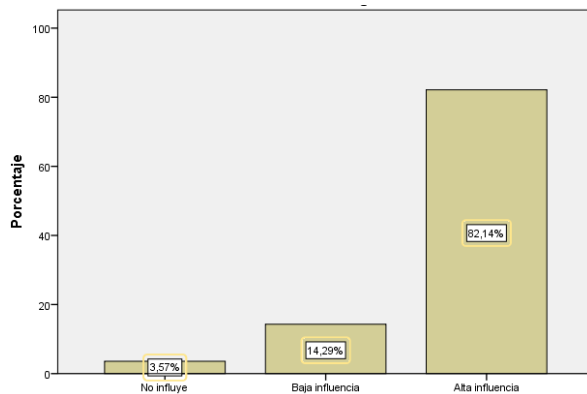
Considera usted que la fase de Análisis influye en la integración del Desarrollo de Software con Inteligencia Artificial.



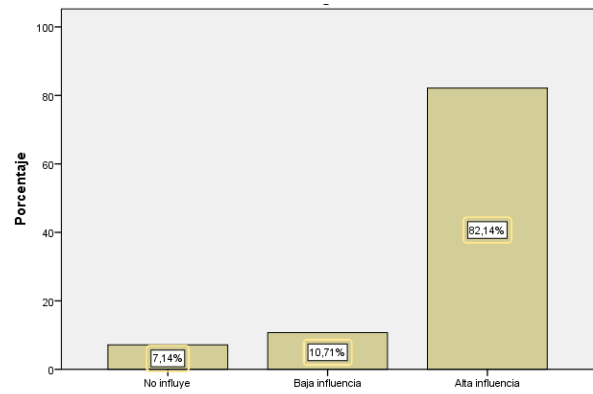
Considera usted que la fase Iteraciones influye en la integración del Desarrollo de Software con Inteligencia Artificial.



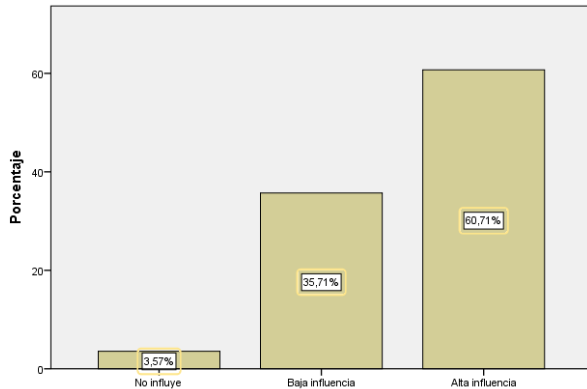
Considera usted que la fase de Desarrollo influye en la integración del Desarrollo de Software con Inteligencia Artificial.



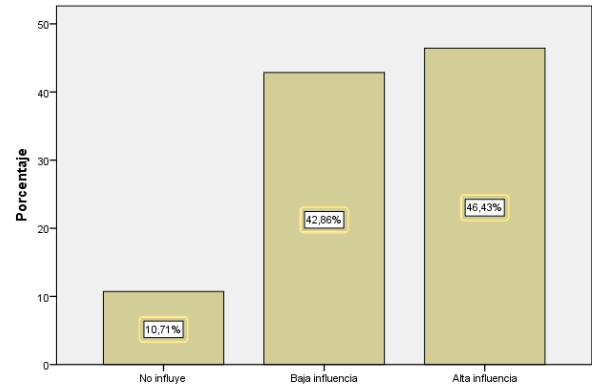
Considera usted que la fase de Codificación influye en la integración del Desarrollo de Software con Inteligencia Artificial.



Considera usted que la fase de Pruebas influye en la integración del Desarrollo de Software con Inteligencia Artificial.

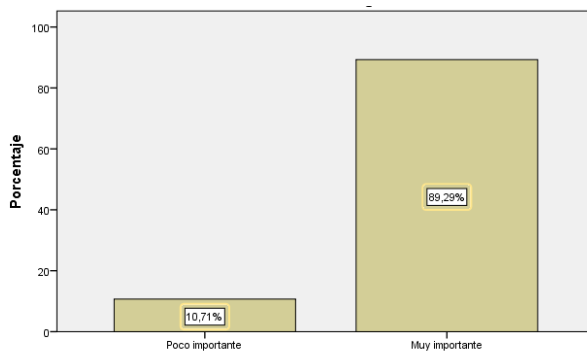


Considera usted que la fase de Mantenimiento del software influye en la integración del Desarrollo de Software con Inteligencia Artificial.

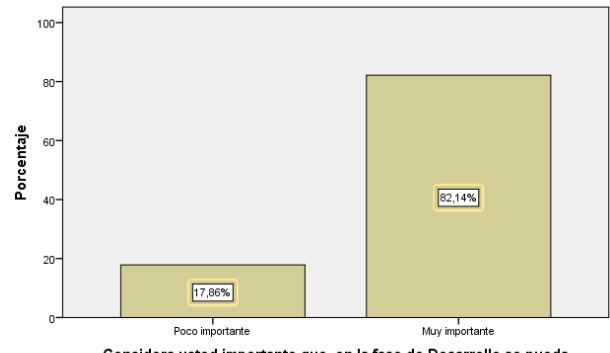


Considera usted que la fase de Culminación del Proyecto influye en la integración del Desarrollo de Software con Inteligencia Artificial.

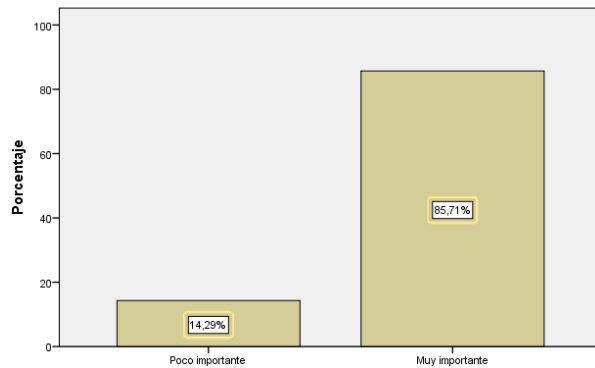
Sección 3: Preguntas que complementan el estudio.



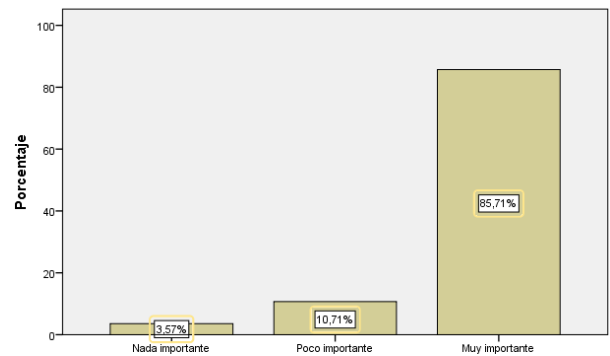
Considera usted importante que, en la fase de Análisis se pueda determinar un proceso de selección de variables que influyan en el desarrollo del software y la construcción de modelos de Inteligencia Artificial.



Considera usted importante que, en la fase de Desarrollo se pueda establecer una etapa de Preprocesamiento que permita el análisis de la data y la mejora del rendimiento de los algoritmos de aprendizaje automático aplicados en el desarrollo de software.



Considera usted importante que, en la fase de Codificación respecto al desarrollo de software, se pueda establecer el modelado de algoritmos de Inteligencia Artificial.



Considera usted importante que, en la fase de Pruebas respecto al desarrollo de software, se pueda determinar un proceso de validación de los algoritmos de aprendizaje automático estableciendo métricas estadísticas de evaluación.