



UNIVERSIDAD JOSÉ ANTONIO PÁEZ

**PLATAFORMA COLABORATIVA PARA LA
GESTIÓN DE PROYECTOS DE SOFTWARE
APLICANDO ALGORITMOS DE
APRENDIZAJE PROFUNDO.**

Autores:

León, Enmanuel
Sánchez, Alejandro

Urb. Yuma II, Calle N° 3, Municipio San Diego
Teléfono: (0241) 8714240 (Máster) - Fax: (0241) 8712394



REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN

PLATAFORMA COLABORATIVA PARA LA GESTIÓN DE PROYECTOS
DE SOFTWARE APLICANDO ALGORITMOS DE APRENDIZAJE
PROFUNDO

Trabajo de grado presentado como requisito para optar al título de
INGENIERO DE COMPUTACIÓN

Autores:

León, Enmanuel

C.I. 27.725.294

Sánchez, Alejandro

C.I. 27.242.594

Tutor académico:

MSc. Jiménez, Oneida

San Diego, Febrero 2020



FI-C-009-2020-1CR (TG)

Valencia, 19 de junio de 2020

Ciudadanos:

León P., Enmanuel J.

27.725.294

Sánchez B., Alejandro E.

27.242.594

Presente-

Cumplo con informarle que la Comisión de Trabajo de Grado y Pasantías de la Facultad de Ingeniería en su reunión N° 04-2020 de fecha 13-02-2020 aprobó el proyecto de trabajo de grado titulado **PLATAFORMA COLABORATIVA PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE APLICANDO ALGORITMOS DE APRENDIZAJE PROFUNDO** presentado por usted (es) como requisito para optar al título de Ingeniero en Computación.

Se ratifica la designación de la Ing. Oneida Jiménez C.I: 10.227.464 como Tutora Académica que los asesorara en el desarrollo de este proyecto.

Atentamente,



Prof. Luís Lira

Decano de la Facultad de Ingeniería

c.c. Coordinación de Pasantías y Trabajo de Grado (1).

Ll/a.a.



REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERIA
ESCUELA DE COMPUTACION

CONSTANCIA DE APROBACIÓN PARA LA PRESENTACIÓN PÚBLICA DEL TRABAJO DE GRADO

Quien suscribe, **Oneida Jiménez**, portador(a) de la cédula de identidad N° **V-10.227.464**, en mi carácter de tutor (a) del trabajo de grado presentado por el(la) los ciudadano(a) **Alejandro Enrique Sánchez Betancourt** y **Enmanuel Jose Leon Pizzano**, portador(es) de la cédula de identidad N° **V-27.242.594** y N° **V-27.725.294**, titulado **PLATAFORMA COLABORATIVA PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE APLICANDO ALGORITMOS DE APRENDIZAJE PROFUNDO** presentado como requisito parcial para optar al título de **Ingeniero en Computación**, considero que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del jurado examinador que se designe.

En San Diego, a los 6 días del mes de Julio del año dos mil veinte.

(Firma autógrafa del tutor)
Nombres y apellidos
N° de la Cédula de Identidad

DEDICATORIA

Dedicamos esta tesis de grado:

En primera instancia, a **DIOS** por darnos la fuerza física y espiritual para la correcta culminación de esta etapa académica.

Y a nuestras **FAMILIAS**, en especial a nuestros padres, por ser los pilares de todo el crecimiento de los y fomentar sus conocimientos, ya que, sin el apoyo y su solidaridad, este proyecto de tesis no se hubiese podido realizar.

¡A todos ustedes, gracias!

Alejandro y Enmanuel

AGRADECIMIENTO

Queremos expresar nuestro profundo agradecimiento a las personas, instituciones y organizaciones que hicieron posible el haber culminado satisfactoriamente esta tesis de grado, para convertirnos en Ingenieros de la República.

A la **UNIVERSIDAD JOSE ANTONIO PAEZ**, que a pesar de los obstáculos encontrados a lo largo de nuestra estancia en esta casa de estudios, nos permitió culminar con éxito este proyecto de vida.

A **INTELIX SINERGY C.A.**, especialmente al **ING. ANGEL NARVAEZ** y al **LICDO. JAVIER ROJAS** por prestarnos todo su apoyo para el proceso de obtención de datos para el desarrollo de la presente tesis de grado.

A la **MsC ING. ONEIDA JIMENEZ** por la disposición y dedicación hacia nosotros, para iniciar y culminar lo que hoy presentamos.

A los Ingenieros **VICTOR HERNANDEZ, JOSE FRANCISCO MEDINA Y PEDRO EMILIO FLORES** por haber certificado el método de recolección de datos empleado para la obtención de la información del correcto desarrollo del sistema.

A nuestros amigos, **DARIANA, LEONARDO, JUAN, HERMES, VICTOR Y FERNANDO** Por haber sido pilares fundamentales de compañerismo para hacer la vida universitaria más interesante y amena.

Al programa **GITHUB EDUCATION**, por proporcionarnos las herramientas necesarias para el desarrollo de esta plataforma colaborativa.

¡A todos nuestro eterno agradecimiento!

Alejandro y Enmanuel

“El éxito no es definitivo, el fracaso no es fatal.
Lo que cuenta es el valor para continuar”

Sir Winston Churchill

ÍNDICE GENERAL

	PP.
RESUMEN.....	X
INTRODUCCIÓN	1
CAPÍTULO	
I EL PROBLEMA	
1.1 Planteamiento del problema	3
1.2 Formulación del problema.....	5
1.3 Objetivos.....	6
1.3.1 Objetivo general.....	6
1.3.2 Objetivos específicos	6
1.4 Justificación.....	6
1.5 Alcance	7
II MARCO TEÓRICO	
2.1 Antecedentes de la investigación.....	8
2.2 Bases teóricas	12
2.2.1 Análisis predictivo	13
2.2.2 Aprendizaje automático	13
2.2.3 Aprendizaje profundo	13
2.2.4 Aprendizaje supervisado.....	14
2.2.5 Ciencia de datos	14
2.2.6 Gestión de proyectos.....	14
2.2.7 Inteligencia artificial.....	15
2.2.8 Manejador de base de datos	16
2.2.9 Multiplataforma	16

2.2.10 Plataforma colaborativa	16
2.3 Bases legales.....	16
2.4 Definición de términos básicos	17
III MARCO METODOLÓGICO	
3.1 Tipo de investigación	19
3.2 Diseño de la investigación.....	19
3.3 Nivel de la investigación	20
3.4 Población y muestra	20
3.5 Técnicas e instrumentos de recolección de datos	21
3.6 Fases de la investigación	22
IV RESULTADOS	
4.1 Fase I	24
4.2 Fase II	45
4.3 Fase III.....	47
4.4 Fase IV.....	62
4.5 Fase V	63
V CONCLUSIONES Y RECOMENDACIONES	
5.1 Conclusiones.....	69
5.2 Recomendaciones	71
ANEXOS	72
REFERENCIAS.....	75
Electrónicas	75

ÍNDICE DE TABLAS Y GRAFICOS

	PP.
TABLAS	
Tabla 1: Respuesta pregunta n° 1. Entrevistas.....	25
Tabla 2: Respuesta pregunta n° 2. Entrevistas	27
Tabla 3: Respuesta pregunta n° 3. Entrevistas.....	29
Tabla 4: Respuesta pregunta n° 4. Entrevistas.....	30
Tabla 5: Respuesta pregunta n° 5. Entrevistas.....	32
Tabla 6: Respuesta pregunta n° 6. Entrevistas.....	33
Tabla 7: Respuesta pregunta n° 7. Entrevistas	34
Tabla 8: Respuesta pregunta n° 8. Entrevistas	35
Tabla 9: Respuesta pregunta n° 9. Entrevistas	37
Tabla 10: Respuesta pregunta n° 10. Entrevistas	38
Tabla 11: Respuesta pregunta n° 11. Entrevistas	40
Tabla 12: Caso de uso (Registrarse).....	46
Tabla 13: Caso de uso (Inicias sesión).....	46
Tabla 14: Caso de uso (Gestionar aptitudes).....	47
Tabla 15: Caso de uso (Gestionar perfil)	47
Tabla 16: Caso de uso (Visualizar notificaciones).....	48
Tabla 17: Caso de uso (Gestionar solicitudes).....	48
Tabla 18: Caso de uso (Crear organización).....	49
Tabla 19: Caso de uso (Gestionar usuarios de la organización)	49
Tabla 20: Caso de uso (Crear proyecto).....	50
Tabla 21: Caso de uso (Gestionar proyecto administrativo de proyecto)	50
Tabla 22: Caso de uso (Gestionar proyecto desarrollador).....	51
Tabla 23: Módulos de plataforma	52
Tabla 24: Inicio de sesión, nuevo usuario.....	59

Tabla 25: Registro de la organización.....	60
Tabla 26: Creación/Inicialización de un proyecto	60
Tabla 27: Registro de aptitudes de un desarrollador.....	61
Tabla 28: Creación/Inicialización de un proyecto (Método automático).....	62
Tabla 29: Privacidad de información	63
Tabla 30: Redireccionamiento por falta de datos de sesión.....	63
Tabla 31: Guardado de imagen de perfil.....	64

GRAFICOS

Gráfico 1: Diagrama de procesos	43
Gráfico 2: Diagrama de caso de administración de proyectos	44
Gráfico 3: Diagrama de uso desarrollador	45
Gráfico 4: Diagrama de base de datos.....	53
Gráfico 5: Diagrama de arquitectura de la plataforma.....	54
Gráfico 6: Esquema de diseño (Página de inicio)	55
Gráfico 7: Esquema de diseño diseño (Estructura interna de básica de navegación en la plataforma)	55
Gráfico 8: Esquema de diseño (Paleta de colores)	56
Gráfico 9: Captura de pantalla (Página de inicio)	57
Gráfico 10: Captura de pantalla (Página de organización).....	57
Gráfico 11: Carta estructurada (Mapa de navegación).....	58



REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN

**PLATAFORMA COLABORATIVA PARA LA GESTIÓN
DE PROYECTOS DE SOFTWARE APLICANDO ALGORITMOS
DE APRENDIZAJE PROFUNDO.**

Autores: Enmanuel León
Alejandro Sánchez
Tutor: MSc. Oneida Jiménez
Fecha: Febrero 2020

RESUMEN

La presente investigación tenía como objetivo desarrollar una plataforma colaborativa, implementando algoritmos de aprendizaje profundo para elevar la eficiencia de la gestión de proyectos, permitiendo la optimización de un modelo efectivo y eficiente en la conformación de equipos de trabajo de desarrollo de software, facilitando incrementar los estándares de calidad y preservando el prestigio de la organización, tomando en cuenta los requerimientos que afectan el desarrollo de software y algunas características o eventos externos. Por otra parte, no se tiene en cuenta únicamente la imagen pública de la empresa, sino también la presentación de futuras tecnologías, considerando como factor primordial la satisfacción del usuario final. Para esto se realiza la presente investigación que se enmarca en el modelo de proyecto especial, con un diseño de campo para recolección de información, paralelamente se hará uso de la metodología de desarrollo XP. Aunado a ello, se realizará enfocado en la línea de investigación, gestión de proyectos de tecnologías de información y comunicación. Para finalizar, se desarrollará un sistema que permita la optimización de la conformación de los equipos de trabajo en el área del desarrollo de software.

Descriptor: Plataforma colaborativa, aprendizaje profundo, gestión de proyectos, equipos de trabajo, desarrollo de software, inteligencia artificial.

INTRODUCCIÓN

La tecnología informática ha evolucionado de tal manera a lo largo de los últimos años, llegando a ser una herramienta indispensable en diferentes áreas laborales de la sociedad actual. Por otra parte, también esta facilita o impulsa pequeñas mejoras o facilidades en la vida cotidiana de los seres humanos. Como ejemplo de ello se tienen las comunicaciones móviles y lo que estas permiten en los diferentes métodos de pagos automatizados, métodos de compras, entre otras. La mayoría de estas tecnologías traen consigo la finalidad de automatizar los procesos y para llegar a desarrollar algunas de ellas, es necesario que la programación evolucionara a ritmos similares que la tecnología informática.

Detrás de toda esta evolución, se presentan cambios significativos lo que generan nuevas tecnologías enfocadas al desarrollo en cada generación, lo que, a través de ellas, las personas, grupos u organizaciones se orientan al desarrollo de tales tecnologías. Estos cambios también se reflejan en las estructuras, sobre todo lo relacionado con los nuevos desarrollos o proyectos a implementar y que han ido evolucionando. Tanto desde el punto de vista metodológico, basado en nuevas formas de trabajar, dependiendo de los diferentes factores que intervienen como la dificultad, el tiempo, los recursos, entre otros, así como desde el punto de vista de los desarrolladores, el conjunto de tecnologías que existen actualmente para el desarrollo de sistemas informáticos, es muy amplio y generalmente una persona se enfoca solo en algunas áreas.

En síntesis, la gestión de proyectos específicamente en el área del software, se ve afectada por diversos factores, incluidas las dificultades de las tecnologías utilizadas, lo que conlleva a que en algunos casos no se presente una correcta planificación, una efectiva gestión de los proyectos o una validación de la factibilidad de los mismos. Cabe destacar que estas planificaciones también pueden verse afectadas por elementos externos que no están planteados y que generan problemas

en la reestructuración de tales planificaciones. Aunado a ello, también se encuentra la problemática de revisar múltiples resúmenes sobre las capacidades individuales en grupos extensos.

En tal sentido, la presente investigación tenía como objetivo el desarrollar una plataforma colaborativa, implementando algoritmos de aprendizaje profundo para elevar la eficiencia de la gestión de proyectos, que permita la optimización de un modelo efectivo y eficiente en la conformación de equipos de trabajo de desarrollo de software.

Con la finalidad de presentar los resultados de la investigación, se han estructurado en cinco (5) capítulos, los cuales se describen a continuación:

Capítulo I, referido a la descripción general del problema, justificación, objetivos de la investigación, incluyendo objetivos específicos y el alcance de la investigación.

Capítulo II, comprende el marco teórico de la investigación, (los antecedentes más influyentes), las bases teóricas, las bases legales y los términos básicos que sustentan la investigación.

Capítulo III, contiene el marco metodológico, el cual comprende la descripción de las metodologías utilizadas en el desarrollo de la investigación y la metodología aplicada en el desarrollo del sistema, que en este caso sería la plataforma.

Capítulo IV, describe los resultados obtenidos en la realización del proyecto, especificada fase por fase.

Capítulo V, conlleva las conclusiones y recomendaciones que se han definido en la realización de todo el proceso investigativo.

CAPÍTULO I

EL PROBLEMA

1.1 Planteamiento del problema

La función de desarrollar software recae en el programador o en una compañía comercial que se dedica a uno o más aspectos del proceso de desarrollo de software. Se trata de un ámbito más amplio de la programación algorítmica. Las empresas de desarrollo de software son empresas que cuentan con personal capacitado (comúnmente conocidos como programadores o desarrolladores de software) los cuales son capaces de concebir y elaborar sistemas informáticos, así como de implementarlos y ponerlos a punto, utilizando uno o varios lenguajes de programación. En estas empresas surgen diversas problemáticas a solucionar y/o proyectos a ejecutar, los cuales hay que entenderlos dependiendo de su naturaleza, ya sean menores, mayores o urgentes, y estas soluciones conllevan una toma de decisiones respecto a lo que se realizará, por lo que hay procesos, herramientas, modelos, que ayudan a esta tarea, que recae en administradores, ingenieros, responsables y líderes.

La capacidad de resolver problemas es la eficacia y agilidad para dar soluciones, emprendiendo las acciones correctivas necesarias con sentido común, sentido del coste e iniciativa. Esta cualidad supone tomar acción de manera proactiva, ante las dificultades, sin pérdida de tiempo y atendiendo a las soluciones que marca el sentido común, pensando en las repercusiones que pueden tener en un plazo más amplio. El solucionar problemas no siempre es tarea fácil, ya que lleva un proceso que involucra no solo decisiones, sino personas para ejecutarlas y recursos que se necesitan y debe atenderse, ya que es un factor de suma importancia en las empresas que conlleva riesgos, ya que no siempre será fácil hacerlo, razón por la

cual es necesario dotarse de buenos elementos de capital humano y conocer las técnicas y herramientas que ayudan para poder primero, identificar los problemas y aplicar los conocimientos para llegar a la mejor solución posible.

Conociendo todo esto, se puede definir como proyectos de software a esas soluciones de problemas o aquellas herramientas diseñadas para facilitar el objetivo propuesto en los sistemas de información. A la hora de llevar a cabo este proceso de desarrollo, surgen diversas situaciones problemáticas que existen cuando hay una desviación entre el desempeño actual de un proceso y el nivel deseado de desempeño. Usualmente cuando se habla lo relacionado a la eficiencia y productividad, se enfoca en el trabajador y esta proposición suele ser mal enfocada, ya que lo más adecuado es verificar la existencia de problemas de organización. Estas posibles deficiencias son las que se quieren reducir con la aplicación de esta plataforma colaborativa.

Aquí es donde entra la planificación de equipos, donde se requiere liderar los talentos, evaluar y regular continuamente las acciones necesarias y suficientes. Por lo antes propuesto, se sabe que en el mundo laboral y en la organización interna de empresas de software, suele ser difícil decidir qué proyectos son posibles de realizar en un período fijo de tiempo, ya sea por dificultad de recursos disponibles o por las aptitudes de los integrantes del equipo. También se percibe que, aunque se posean los anteriores requisitos, no es tan fácil gestionar un proyecto, ni elaborar un modelo óptimo de organización de equipos.

El gerente de proyectos es la persona encargada de la organización de los mismos y en ciertas ocasiones, es un representante del cliente y debe determinar e implementar la satisfacción y atención de sus necesidades e inquietudes exactas, basándose en el conocimiento de la firma que representa. La habilidad de adaptar los múltiples procedimientos internos de la parte contratante y la forma de estrechar los lazos con los representantes seleccionados, es esencial para asegurar que los objetivos clave de alcance, costo, tiempo, calidad y, sobre todo, satisfacción del cliente y con buen clima interpersonal, se hagan realidad.

Aunque ninguno de los problemas antes mencionados tiene una influencia directa en las ventas, la situación es que, de forma muy superficial se van generando dificultades en el funcionamiento del equipo y debido a ello, se generan modelos de planificación inconvenientes e ineficientes, pudiendo distribuir de una forma muy equivocada las capacidades de desarrollo de la empresa en distintas áreas, y pudiendo terminar en la no finalización de proyectos en los tiempos establecidos o en el incumplimiento de los requerimientos u objetivos del mismo.

De igual forma, estos problemas de organización traen consigo otras consecuencias, tales como, no llevar una correcta administración del tiempo, fallas en el cumplimiento de los procesos y tareas, insuficiencia de recursos como puede ser el personal capacitado y deficiencias en el sistema de control interno.

Para superar dicho problema, es importante manejar una planificación y estrategias adecuadas. Este modelo de gestión de proyectos es de suma importancia en los equipos de trabajo, ya que plantea posibles mejoras en el tiempo de desarrollo para que funcione como herramienta que identifique alguna debilidad en la empresa como puede ser la insuficiencia o falta de capacidad de desarrollo en un área específica, aparte de poder evaluar el rendimiento de las personas en ese equipo y así asegurarse que cada individuo cumpla las métricas establecidas para que estos datos conlleven a soluciones que permita la toma de decisiones efectivas y de acuerdo a lo que se hará al respecto.

1.2 Formulación del problema

Dentro de este marco propuesto y considerando las consecuencias que tiene esta falta de organización, surge la siguiente interrogante: ¿Como elevar la eficiencia de la gestión de proyectos de software a través de herramientas computacionales?

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar una plataforma colaborativa para la gestión de proyectos de software aplicando algoritmos de aprendizaje profundo

1.3.2 Objetivos específicos

- Diagnosticar la situación actual de las plataformas de desarrollo colaborativo de trabajo o similares al sistema en desarrollo, aplicando las técnicas respectivas de recolección de datos.
- Determinar los requerimientos funcionales y no funcionales para una plataforma colaborativa basada en algoritmo de aprendizaje profundo, en la gestión de proyectos de software, en función al diagnóstico que se realizara.
- Diseñar las bases de los modelos de gestión de proyectos para la plataforma colaborativa mediante la metodología XP.
- Desarrollar la plataforma colaborativa para la gestión de proyectos de software usando herramientas computacionales.
- Ejecutar un plan de pruebas de software para la verificación del correcto funcionamiento de la plataforma colaborativa.

1.4 Justificación

El presente trabajo de investigación tiene como justificación la necesidad de crear una plataforma de desarrollo colaborativo, aplicando algoritmos de aprendizaje profundo que permita ayudar a evaluar y tomar decisiones respecto a la distribución de equipos y cargas de trabajo, aparte de ayudar a identificar las debilidades que el equipo y/o cada integrante puedan presentar.

En tal sentido, es posible percibir que esta plataforma puede ser de gran impacto en la empresa de software, ya que esta facilitaría una forma de mantener sus estándares de calidad en los de tiempos requeridos y de igual forma su prestigio.

Esta plataforma al exponer algunas debilidades del grupo de trabajo, permite realizar cambios para mejorar el entorno. Primero, se identificara si esta deficiencia

es por falta de capacidad del equipo para la tarea o por alguna otra situación, ya sea por factores externos que afecten a uno o varios de los integrantes del grupo, para luego ejecutar las acciones que se considere necesarias las cuales pueden ser ampliación del equipo de trabajo o exponer la necesidad de capacitación de algunos integrantes, o la posible solución a algún problema de interacción laboral entre algunos integrantes que puede terminar en acciones más drásticas como despedir a algún integrante, sin embargo estas decisiones siempre tienen que ser aprobadas por el personal encargado.

Este proyecto también aportara información al área académica como soporte en próximas investigaciones, para mejorar la gestión de proyectos. Por otra parte, los recursos de evaluación de rendimiento, la cual es un área de investigación difícil de tratar por los muchos factores que están inmersos, serian efectuados mediante algoritmos de aprendizaje profundo.

1.5 Alcance

La plataforma colaborativa tiene la funcionalidad de ayudar a solucionar los problemas de organización y podrá ayudar en la toma de decisiones para el control de proyectos, así como también servirá para identificar algunas debilidades o falta de rendimiento de forma grupal e individualmente.

Esta plataforma se dividió en dos partes, en primer lugar, para la red neuronal artificial, se utilizará como lenguaje de programación Python y segundo para el desarrollo de la plataforma web, se hará uso como lenguaje de programación Javascript y uso de HTML, CSS y BOOTSTRAP para el diseño de la interfaz.

También se estipulo que la metodología aplicada es la Extreme Programming (XP), la cual se centra en potenciar las relaciones interpersonales del equipo de desarrollo como clave del éxito mediante el trabajo en equipo, el aprendizaje continuo y el buen clima de trabajo.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes de la investigación

Esthory (2019), elaboró un trabajo de investigación para obtener la maestría en matemáticas y computación en la Universidad de Carabobo, Venezuela titulado **“Sistema de redes neuronales para la evaluación de programas de postgrado”**, esta investigación utiliza tecnología de Redes Neuronales y Deep Learning para diseñar un sistema de aprendizaje que procese encuestas de satisfacción hechas a los estudiantes de postgrado. Se quiere medir de manera independiente el impacto del uso de un sistema basado en redes neuronales en las encuestas de satisfacción de los estudiantes de postgrado. La modalidad de la investigación se enmarca en los lineamientos de proyecto factible. El diseño de la investigación es de campo, los datos de interés se recogen directamente de la realidad, mediante el trabajo directo y concreto del investigador y su equipo.

La población de la investigación es la Coordinación de los programas de Postgrado de Ingeniería. La muestra que se toma para la evaluación de los programas de postgrado de Ingeniería es censal, es decir, todos los programas de postgrado de Ingeniería de la Universidad de Carabobo y de esta forma se comprueba su funcionamiento. Este modelo permite el alcance del objetivo, también ha sido capaz de pronosticar de manera eficiente las evaluaciones que son realizadas por personas.

Esta forma investigación es una importante base de información a la hora del desarrollo de la plataforma en la investigación actual, más específicamente en el área de predicciones que emplea algoritmos de aprendizaje profundo o Deep Learning. Los más relevantes son el exceso de potencia que supone las redes neuronales para algunos pronósticos y la adaptabilidad que estas tienen, aunando en esto y como en

desarrollo y ejecución en el tiempo que dure el proyecto. Este sistema logro también reducir notablemente los tiempos de generación y consulta de reportes.

De este modo concluye, superando satisfactoriamente con los objetivos planteados en la implementación del sistema y aclara que la parte más fundamental de los sistemas de información es una documentación completa y adecuada lo cual es muchas ocasiones no se tiene como prioridad, lo que permite un mejor manejo del sistema por parte del usuario. Para tales efectos esta investigación también da una perspectiva importante, ya que se puede apreciar un método simple para gestionar proyectos en distintas fases y usuarios con diferentes roles en el proyecto. También es importante destacar que solo algunos roles pueden determinar si alguna fase está finalizada o si se puede empezar paralelamente otra fase.

Además, Blanco (2018), realizó un trabajo de investigación titulado “**Módulo de gestión de órdenes de servicio y actividades para el sistema web de la empresa CrediNat C.A.**” en la Universidad de Carabobo, Venezuela. Esta investigación está centrada en la necesidad de la empresa en un escritorio de ayuda o mejor dicho un centro de soporte centralizado para las distintas sedes de la empresa, para esto se desarrolló un módulo web y se integró a la página de la empresa la cual se encontraba en fase de desarrollo.

Finalmente concluye en que el módulo desarrollado cumple con los objetivos pautados, también propone que el módulo fue desarrollado externamente a la página web de la empresa y pudo ser integrado sin mayores inconvenientes, todo esto haciendo énfasis en la modularidad que puede presentar una plataforma. Esta aporta una perspectiva distinta a la resolución de problema, que se puede modelar en la plataforma planteada en la investigación actual como un sistema de Solicitudes para la gestión de revisión de fases del proyecto o de una intervención necesaria por problemas en el desarrollo, y también un registro histórico de problemas que se presenten en el mismo y se planteen como objetivos en medio del desarrollo. Por otra parte, es posible rescatar como algo fundamental la modularidad que siempre ha de

estar presente en la programación, que es dividir un problema en partes más pequeñas y así poder llegar a una solución más fácilmente.

Así mismo, Argomedo y Ochoa (2017), elaboraron un trabajo de investigación para optar al título profesional de Ingeniero de Sistemas en la Universidad Nacional de Trujillo, Facultad de Ingeniería, (Perú) y que se tituló **“implementación de un sistema de información web para mejorar la gestión de proyectos siguiendo los lineamientos de la guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®) para la empresa SIGCOMT S.A.C”**, tiene la finalidad de mejorar la gestión de proyectos, mediante el desarrollo e implementación de un Sistema de Información web logrando que los proyectos tengan un mejor manejo, desarrollo y ejecución en el tiempo que dure el proyecto, este sistema logro también reducir notablemente los tiempos de generación y consulta de reportes, cabe destacar que los investigadores llegaron a la conclusión de que mediante la implementación de un Sistema de Información Web se logra mejorar la gestión de proyectos.

También concluyen en que los sistemas de información requieren una documentación completa y adecuada para poder facilitar el uso de esta y posibles mejoras a futuro. Esta investigación también da una perspectiva importante, ya que se puede apreciar un método simple para gestionar proyectos en distintas fases y usuarios con diferentes roles en el proyecto. También es importante destacar que solo algunos roles pueden determinar si alguna fase está finalizada o si se puede empezar paralelamente otra fase.

Al mismo tiempo, Araque (2015), elaboró un trabajo de investigación titulado **“Gestión de riesgos en proyectos de software a desarrollar en empresas privadas”**, en la Universidad Militar Nueva Granada en Bogotá, Colombia. En esta investigación se evidencia la gestión de riesgos para el desarrollo de un software para una entidad privada basada en la guía del PMBOK, donde se hace una identificación de los riesgos más importantes en cada una de las fases del desarrollo, de igual forma se hace un análisis cualitativo, por medio del cual se haya la probabilidad de

ocurrencia y el impacto de cada uno de los riesgos identificados, seguidamente se hace una clasificación por el nivel del riesgo, para posteriormente proponer un plan de respuesta a los riesgos más críticos.

Al respecto de esto la investigación concluye en que es importante entender de forma específica y clara cada una de las fases del desarrollo de software, ya que de estas se deben identificar los riesgos más importantes que las afectan, posteriormente se debe realizar una adecuada determinación del nivel del riesgo por cada uno de los riesgos para proponer el plan de respuesta que más se ajuste al riesgo. Esta investigación forma una parte importante en la gestión de proyectos el cual es la identificación de posibles riesgos en un proyecto de software, lo cual tiene un considerable impacto en la hora de toma de decisiones en cada modelo de gestión, llama la atención que actualmente existe una cantidad limitada de investigaciones específicamente de la gestión de riesgos en esta investigación y es algo que siempre debe estar presente en el día a día del desarrollo de proyectos.

Paralelamente, Rojas, Pérez y Echeverría (2019), realizaron un artículo de investigación titulado **“Predicción del gasto público en los municipios de la región andina de Venezuela empleando redes neuronales artificiales”**, artículo el cual esta publicado en la Revista Ibérica de Sistemas y Tecnologías de Información. En esta investigación se presenta una aplicación de las redes neuronales artificiales para la predicción del gasto de las entidades municipales de la “Región Andina de Venezuela”, incorporando información de variables socioeconómicas y demográficas. Se realiza un exhaustivo análisis estadístico univariante y multivariante, se ajustan modelos para realizar dos tipos de predicción: predicción individual para cada municipio y predicción promedio para los grupos identificados en el análisis estadístico.

En última instancia, al realizar la validación de ambos modelos de redes neuronales con los dos datos de los dos últimos años de las series de tiempo bajo estudio, se determinó que los modelos contaron con una buena capacidad predictiva,

dado que los errores de estimación tendieron a disminuir en el último año de la validación, por lo que los valores obtenidos en la salida se asemejaron a los valores reales del presupuesto total y promedio de gastos de los municipios de la Región Andina. Esta investigación influye un método diferente para realizarlas las predicciones, aparte de plantear el ajuste estadístico de las variables de entrada, esto en la presente investigación permite la visualización de un espectro más grande de posibilidades a la hora de predecir el rendimiento promedio que puedan tener los equipos de trabajo en los proyectos y a calcular su factibilidad.

Finalmente, Carranza (2016), desarrollo un artículo de investigación titulado **“Gestión en proyectos de software”**, elaborado en la Universidad Distrital Francisco José de Caldas, Bogotá Colombia, este artículo tiene como finalidad estudiar los contrastes y delimitaciones más importantes en la gestión de proyectos de software, también plantea aspectos que se deben evaluar a la hora de gestionar un proyecto de este tipo, sobre todo en el área de conformación de equipos de trabajo y una evaluación de la factibilidad del proyecto y los diversos procesos que son necesarios tener en cuenta para dicha gestión.

Tal investigación concluye en lo importante que es la gestión de proyectos en la consolidación de una organización empresarial, que sin importar el tamaño o dificultad de un proyecto debe ser gestionado y que los aspectos estudiados intervienen de manera considerable en el éxito o fracaso dentro del proceso del proyecto. Cabe destacar que este artículo aporta métodos para la evaluación de características que son requeridas y conocimientos pertinentes a la evaluación de la factibilidad del mismo.

2.2 Bases teóricas

Según Bavaresco (2006), las bases teóricas tienen que ver con las teorías que brindan al investigador el apoyo inicial dentro del conocimiento del objeto de estudio, es decir, cada problema posee algún referente teórico, lo que indica, que el

investigador no puede hacer abstracción por el desconocimiento, salvo que sus estudios se soporten en investigaciones puras o bien exploratorias.

2.2.1 Análisis predictivo

Nyce (2007), expone que “El análisis predictivo agrupa una variedad de técnicas estadísticas de modelización, aprendizaje automático y minería de datos que analiza los datos actuales e históricos reales para hacer predicciones acerca del futuro o acontecimientos no conocidos”. En base a esta concepción, es posible deducir que el análisis predictivo implementa herramientas estadísticas cuyo propósito es predecir sucesos próximos basados en eventos presentes o pasados.

2.2.2 Aprendizaje automático

Russell y Norvig (2009), indican que “El aprendizaje automático o aprendizaje automatizado o aprendizaje de máquinas (del inglés, machine learning) es el subcampo de las ciencias de la computación y una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan que las computadoras aprendan” por lo que comprender, este subcampo de la inteligencia artificial, permite a los sistemas “aprender” automáticamente, que en este contexto quiere decir, identificar patrones complejos en una gran cantidad de datos.

2.2.3 Aprendizaje profundo

Rouse, M. (2017), define aprendizaje profundo de la siguiente forma “El aprendizaje profundo, también conocido como redes neuronales profundas, es un aspecto de la inteligencia artificial (AI) que se ocupa de emular el enfoque de aprendizaje que los seres humanos utilizan para obtener ciertos tipos de conocimiento. En su forma más simple, el aprendizaje profundo puede considerarse como una forma de automatizar el análisis predictivo”. Como aclara Rouse este aspecto se aplica en el análisis predictivo, lo cual en esta investigación es algo fundamental que se plantea implementar en la generación de modelos para la gestión de proyectos.

2.2.4 Aprendizaje supervisado

Simeone (2018), señala que “En el aprendizaje supervisado, los algoritmos trabajan con datos “etiquetados” (labeled data), intentado encontrar una función que, dadas las variables de entrada (input data), les asigne la etiqueta de salida adecuada. El algoritmo se entrena con un “histórico” de datos y así “aprende” a asignar la etiqueta de salida adecuada a un nuevo valor, es decir, predice el valor de salida.”. De este concepto interpretamos que el aprendizaje supervisado consta de variables de entrada y salida, cuya información está clasificada y aplicando algoritmos de aprendizaje automático permite identificar patrones en los datos y así aprender a predecir una salida en base a datos suministrados.

2.2.5 Ciencia de datos

Liu (2015), propone que la ciencia de datos “es un campo interdisciplinario sobre procesos y sistemas para extraer conocimiento o ideas de grandes volúmenes de datos en diversas formas, ya sea estructuradas o no estructuradas, que es una continuación de algunos de los campos de análisis de datos, como la minería de datos y el análisis predictivo, así como el descubrimiento de conocimiento y minería de datos.” De este concepto se discierne que la ciencia de datos permite obtener información basada en datos, ya sea que estén clasificados o no, para así poder realizar análisis estadísticos en la información obtenida.

2.2.6 Gestión de proyectos

Según Amejide, L. (2016), “La dirección o gestión de proyectos, es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos del mismo. Se logra mediante la aplicación e integración adecuada de una serie de procesos agrupados”, del cual se puede interpretar como toda la metodología necesaria para poder llevar a un buen proceso de desarrollo de proyectos.

2.2.7 Inteligencia artificial

Kaplan y Haenlein (2019) “Definen la Inteligencia Artificial como la capacidad de un sistema para interpretar correctamente datos externos, aprender de dichos datos y usar esos aprendizajes para lograr objetivos y tareas específicos a través de una adaptación flexible.” Es decir, sistemas inteligentes, con la posibilidad de recibir información, y generar una salida útil para el fin con que fue desarrollado, todo esto considerando el previo entrenamiento del sistema.

Además de esto Russell y Norvig (2009), clasifican los sistemas inteligentes de la siguiente manera:

- Sistemas que piensan como humanos. - Estos sistemas tratan de emular el pensamiento humano; por ejemplo, las redes neuronales artificiales. La automatización de actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolución de problemas y aprendizaje
- Sistemas que actúan como humanos. Estos sistemas tratan de actuar como humanos; es decir, imitan el comportamiento humano; por ejemplo, la robótica. El estudio de cómo lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor.
- Sistemas que piensan racionalmente. Es decir, con lógica (idealmente), tratan de imitar el pensamiento racional del ser humano; por ejemplo, los sistemas expertos. El estudio de los cálculos que hacen posible percibir, razonar y actuar.
- Sistemas que actúan racionalmente. Tratan de emular de forma racional el comportamiento humano; por ejemplo, los agentes inteligentes. Está relacionado con conductas inteligentes en artefactos.

Por lo que se indica que, existen diferentes clasificaciones para los sistemas de inteligencia artificial en base al uso que se requiera.

2.2.8 Manejador de base de datos

Según Álvarez, S. (2007), “Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a realizar las siguientes acciones: definición de los datos, mantenimiento de la integridad de los datos dentro de la base de datos, control de la seguridad y privacidad de los datos y manipulación de los datos”, por lo cual se entiende como manejador de base de datos a la herramienta que permite el fácil manejo de la información contenida en dicha base.

2.2.9 Multiplataforma

Según Alegsa, L. (2018), “Multiplataforma es la capacidad de un software de soportar múltiples plataformas informáticas. Cross-platform en inglés. Esto significa que el software que es multiplataforma, tiene la característica de funcionar de forma similar en distintas plataformas”, del cual como se puede inferir es la capacidad de un software de adaptarse a diferentes plataformas de forma muy similares tanto en rendimiento, como en características del mismo.

2.2.10 Plataforma colaborativa

Rouse (2017), plantea lo siguiente: “Una plataforma de colaboración es una categoría de software empresarial que agrega amplias capacidades de redes”, en síntesis, son herramientas virtuales actualmente utilizadas para la gestión de diferentes procesos en una organización, ya sea mediante la implementación de recursos internos o externos para mejorar el uso de la misma y así poder cumplir con su función.

2.3 Bases Legales

Según Palella y Stracruzzi (2017), las bases legales "son las normativas jurídicas que sustenta el estudio a través de la Carta Magna, las Leyes Orgánicas, las Resoluciones, los Decretos, entre otros". Por lo tanto, las bases legales de esta investigación se encuentran representadas en la Constitución de la República Bolivariana de Venezuela (1999).

Artículo 98. La creación cultural es libre. Esta libertad comprende el derecho a la inversión, producción y divulgación de la obra creativa, científica, tecnológica y humanística, incluyendo la protección legal de los derechos del autor o de la autora sobre sus obras. El Estado reconocerá y protegerá la propiedad intelectual sobre las obras científicas, literarias y artísticas, invenciones, innovaciones, denominaciones, patentes, marcas y lemas de acuerdo con las condiciones y excepciones que establezcan la ley y los tratados internacionales suscritos y ratificados por la República en esta materia.

Artículo 110. El Estado reconocerá el interés público de la ciencia, la tecnología, el conocimiento, la innovación y sus aplicaciones y los servicios de información necesarios por ser instrumentos fundamentales para el desarrollo económico, social y político del país, así como para la seguridad y soberanía nacional. Para el fomento y desarrollo de esas actividades, el Estado destinará recursos suficientes y creará el sistema nacional de ciencia y tecnología de acuerdo con la ley. El sector privado deberá aportar recursos para los mismos. El Estado garantizará el cumplimiento de los principios éticos y legales que deben regir las actividades de investigación científica, humanística y tecnológica. La ley determinará los modos y medios para dar cumplimiento a esta garantía.

2.4 Definición de términos básicos

- **Algoritmos:** es un conjunto de instrucciones o reglas definidas y no-ambiguas, ordenadas y finitas que permite típicamente solucionar un problema, realizar un cómputo, procesar datos y llevar a cabo otras tareas o actividades.
- **API:** es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones.
- **Aprendizaje automático:** El aprendizaje automático, o el automatizado de máquinas (del inglés machine learning) es el subcampo de las ciencias de la computación y una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan que las computadoras aprendan.
- **Aprendizaje profundo:** El aprendizaje profundo (también conocido como aprendizaje estructurado profundo o aprendizaje jerárquico), es parte de una familia más amplia de métodos de aprendizaje automático, basados en redes

neuronales artificiales. El aprendizaje puede ser supervisado, semi-supervisado o sin supervisión.

- **Plataforma:** Una plataforma digital es un lugar de Internet que sirve para almacenar diferentes tipos de información tanto personal como nivel de negocios. Estas plataformas funcionan con determinados tipos de sistemas operativos y ejecutan programas o aplicaciones con diferentes contenidos, como pueden ser juegos, imágenes, texto, cálculos, simulaciones y vídeo, entre otros.
- **Red Neuronal Profunda:** Una red neuronal profunda (DNN), es una red neuronal artificial (ANN), con múltiples capas entre las capas de entrada y salida. El DNN encuentra la manipulación matemática correcta para convertir la entrada en la salida, ya sea una relación lineal o no lineal. La red se mueve a través de las capas calculando la probabilidad de cada salida.

CAPÍTULO III

MARCO METODOLÓGICO

3.1 Tipo de investigación

El presente trabajo de investigación es de tipo “Proyecto Especial”, definido por Mijares y García (2007), de la siguiente manera “Consistirá en las creaciones tangibles, susceptibles de ser utilizadas como soluciones a problemas demostrados o que respondan a necesidades e intereses de tipo cultural”, la cual también afirma que “se incluyen en esta categoría los trabajos de elaboración de libros de textos y de materiales de apoyo educativo, el desarrollo de software y hardware, prototipos y de productos tecnológicos en general”, por lo tanto conociendo que como el objetivo es desarrollar un software, entonces pertenece a este tipo de investigación.

De la misma forma, según Pérez, A. (2012), “Se entiende por proyectos especiales a las propuestas tangibles sistemáticamente elaboradas con base en conocimientos preestablecidos y valiéndose de procesos imaginativos y de herramientas del diseño y de la planificación, para ser presentadas como soluciones novedosas y creativas ante necesidades o problemas determinados, que hacen posibles mejores condiciones para los seres humanos. Equivale a decir que son intervenciones que independientemente de su grado de complejidad tienen como propósito específico o especial resolver aquellos problemas que surgen en cualquier ámbito del desempeño humano, con el uso de los conocimientos existentes”.

3.2 Diseño de la investigación

Según Arias (2012), “El diseño de investigación es la estrategia general que adopta el investigador para responder al problema planteado. En atención al diseño, la investigación se clasifica en: documental, de campo y experimental”. También,

según Arias (2006) se refiere la investigación de campo como: “aquella que consiste en la recolección de datos directamente de los sujetos investigados, o de la realidad donde ocurren los hechos (datos primarios), sin manipular o controlar variable alguna”. Teniendo estos conceptos presentes se define que esta investigación se inclina a ser una investigación de campo, debido a que los datos necesarios para el desarrollo de la plataforma son recolectados de la empresa y esta misma información proviene de una igualdad de condiciones y no se ve alterada por parte de los investigadores como ocurre en las investigaciones experimentales.

3.3 Nivel de investigación

El nivel de investigación se refiere al grado de profundidad con que se aborda un fenómeno u objeto de estudio. La presente investigación es de tipo descriptivo, ya que comprende la descripción, registro y análisis de todos los integrantes que están definidos para la plataforma colaborativa para optimizar la gestión de proyectos.

Según Arias F. (2012), “la investigación descriptiva consiste en la caracterización de un hecho, fenómeno, individuo o grupo, con el fin de establecer su estructura o comportamiento. Los resultados de este tipo de investigación se ubican en un nivel intermedio en cuanto a la profundidad de los conocimientos se refiere”.

3.4 Población y muestra

Se define la población como “(...) conjunto finito o infinito de elementos con características comunes, para los cuales serán extensivas las conclusiones de la investigación. Esta queda limitada por el problema y por los objetivos de estudio”. Arias, F. (2012)

Se define Muestra cómo “(...) un subconjunto representativo y finito que se extrae de la población accesible” Arias, F. (2012)

Para esta investigación se utilizará una muestra intencionada, debido a la extensa población de la empresa y la diversidad de áreas de trabajo, por lo tanto, se decidió tomar como población, los jefes o gerentes de proyecto y los desarrolladores

para recopilar información y una muestra de aproximadamente 6 personas de esta organización.

3.5 Técnicas e instrumentos de recolección de datos

Según Arias, F. (2012) “las técnicas de recolección de datos son el procedimiento o formas particulares de obtener la información”. Teniendo en cuenta que las técnicas son los procedimientos, entonces para el desarrollo de la investigación se emplearan las siguientes técnicas e instrumentos de recolección de datos. En primer lugar, se hará uso de la recopilación documental, que Torrealba y Rodríguez (2009) definen como “La recopilación documental es un instrumento o técnica de investigación general cuya finalidad es obtener datos e información a partir de fuentes documentales con el fin de ser utilizados dentro de los límites de una investigación en concreto.”, en síntesis, es la revisión bibliográfica de diversas fuentes cuya información es aplicable o necesaria en la investigación. Cabe destacar que estará centrada hacia el área metodológica de ingeniería del software.

En primer se aplica la observación directa, que es un método de recolección de datos cuyo proceso consiste en la visualización consistente de un fenómeno en estudio dentro de una situación particular, esto se realiza en el entorno sin modificar o alterar las condiciones ambientales, además de la no intervención del procedimiento de recolección de información ya que esto ocasionaría la invalidez de la misma.

Por último, se tiene a la entrevista que Vargas, (2012) define como “Una conversación considerada como “el arte de realizar preguntas y escuchar”. Como se puede interpretar, es un dialogo entre el investigador y una persona involucrada en el tópico. Esta ofrece la ventaja de que el entrevistado está relacionado con el tema y la información que este pueda aportar será una base de la investigación, esta será aplicada de forma no estructurada para obtener algunos de los requerimientos y características que una plataforma como la planteada en esta investigación necesitan para elevar la eficiencia en la gestión de proyectos.

Una lista de cotejo es una herramienta para facilitar la recolección de información en el desarrollo de una investigación, la función más general de esta es la posibilidad de comparar los objetivos planteados, en contraste a la información o al aprendizaje obtenido. Por otra parte, este es un gran complemento con lo que respecta a la entrevista, porque como se menciona anteriormente permite a modo general verificar la información obtenida en la entrevista. Motivo por el cual es un gran complemento a una de las técnicas de recolección de información en la presente investigación.

3.6 Fases de la investigación

El desarrollo de esta plataforma se llevó a cabo mediante la implementación de la metodología XP, la cual, según Izquierdo, J. (2014) la define como “una metodología basada en la comunicación, la reutilización del código desarrollado y la realimentación”. Esta metodología es idónea para proyectos que necesitan de buena interacción del equipo de desarrollo y como dice Calvo, D. (2018) “esta metodología pone el énfasis en la retroalimentación continua entre cliente y el equipo de desarrollo y es idónea para proyectos con requisitos imprecisos y muy cambiantes”. También se pueden aplicar estos términos de requisitos a las funciones internas de la plataforma que tomarán forma y evolucionarán en su desarrollo para llegar a los resultados esperados de la misma. Esta metodología se divide en las siguientes fases:

- **Fase I: Diagnóstico de la situación actual de las plataformas de desarrollo colaborativo de trabajo o similares al sistema en desarrollo, aplicando las técnicas respectivas de recolección de datos.**

Para lograr el objetivo de diagnosticar la situación actual de las plataformas de desarrollo colaborativo o similares, se propuso aplicar las técnicas de recolección de información mediante la utilización de un instrumento como es la entrevista y aplicarla de forma no estructurada

- **Fase II: Determinación de los requerimientos funcionales y no funcionales para una plataforma colaborativa basada en algoritmo de**

aprendizaje profundo, en la gestión de proyectos de software, en función al diagnóstico que se realizara.

En esta fase se establecieron los requerimientos funcionales y no funcionales para una plataforma colaborativa, basada en algoritmos de aprendizaje profundo para gestión de proyectos de software, en función al diagnóstico de la situación en el momento que fue realizada la investigación.

- **Fase III: Diseño de las bases de los modelos de gestión de proyectos para la plataforma colaborativa mediante la metodología XP.**

Se elaboraron las bases de los modelos de gestión de proyectos para la plataforma colaborativa mediante la aplicación de la metodología XP.

- **Fase IV: Desarrollo de la plataforma colaborativa para la gestión de proyectos de software usando herramientas computacionales.**

Se realizó el proceso de codificación de la plataforma colaborativa a través de su desarrollo para la gestión de proyectos de software, usando herramientas computacionales.

- **Fase V: Ejecución de un plan de pruebas de software para la verificación de el correcto funcionamiento de la plataforma colaborativa.**

En esta fase se ejecutó un plan de pruebas de software, para verificar el correcto funcionamiento de la plataforma colaborativa aprovechando los recursos que se tengan a disposición.

CAPÍTULO IV

RESULTADOS

4.1 Fase I: Diagnóstico de la situación actual de las plataformas de desarrollo colaborativo de trabajo o similares al sistema en desarrollo, aplicando las técnicas respectivas de recolección de datos.

Mediante la elaboración de un guion para la entrevista y la aplicación de todos los métodos y técnicas mencionados con anterioridad, se obtuvo una visión general de cómo se maneja todo el conjunto de información tanto en la organización en cuestión como en otras herramientas similares, entiendo mejor como se realiza el procesamiento de datos y definiendo los parámetros necesarios para el mismo. Aunando a esto, con esta información se extrajeron todos los datos necesarios para definir los requisitos funcionales y no funcionales de la plataforma colaborativa en cuestión.

4.1.1 Actividad I: Investigación de sistemas con similitudes a la plataforma que se planteó desarrollar.

Se investigaron y probaron diferentes herramientas que cumplieran con la característica de facilitar la gestión de los proyectos de software, tales como Jira, GitHub, Slack, Trello, etc. El proceso de pruebas se centro en entender las funcionalidades y su utilidad en la gestión de proyectos para así comprender de forma mas amplia que especto abarcaba cada uno de estos sistemas y poder rescatar, mejorar y adaptar algunas características de estas, entre esto están incluidas las API de las mismas como forma de integración de esos sistemas.

4.1.2 Actividad II: Elaboración del guion para la entrevista y aplicación del mismo.

En esta actividad se elaboro el guion para la entrevista en base a la investigación previamente realizada, con la finalidad de poder analizar la mejora de ciertas características que ya se plantearon y poder ajustar estas mas

a la necesidad de la empresa, también para poder comprender con mayor claridad como se realizan los procesos de toma de decisión para la conformación de los equipos y así mediante algoritmos de aprendizaje profundo para la elaboración de una red neuronal profunda llegar a una automatización de este proceso. En cuanto a los resultados de esta entrevista tenemos que:

1. De acuerdo a su opinión profesional, ¿cuáles considera que deban ser los criterios para la distribución y asignación de desarrolladores a proyectos?

Tabla 1. Respuestas pregunta n°1 entrevista.

INFORMANTE CLAVE N°	RESPUESTAS
1	<p>Depende de la forma en la cual se aborde el proyecto, si se tienen restricciones de tecnología a usar, se deberían buscar los desarrolladores con mejor conocimiento de la tecnología que estén disponibles.</p> <p>Si se puede seleccionar la tecnología es importante también tomar en cuenta si se cuenta con desarrolladores que conozcan las reglas de negocio o el modelo de negocio que se desea implementar.</p> <p>La idea de todo esto es aplanar la curva de aprendizaje lo más posible, para minimizar los tiempos de aprendizaje y desarrollo del proyecto.</p> <p>Otro aspecto importante es lograr equipos multidisciplinarios, donde algunos miembros sean fuertes en algunas áreas y se complementan con las fortalezas de otras personas en otras áreas, el ejemplo clásico de esto es tener alguien con sólidos conocimientos en front-end y otra persona de igual manera en back-end.</p>
2	<p>a.-Organización del equipo de acuerdo al skill técnico a fin de lograr el mejor desempeño dentro del proyecto.</p> <p>b.-Designar al líder (técnico), quien debe tener el conocimiento previo de la solicitud</p> <p>c.-Definir las etapas y actividades del proyecto</p> <p>d.-Distribución de las actividades dentro del equipo, haciendo un</p>

	balanceo en tiempo y esfuerzo.
3	<p>Primero se debe determinar las tareas que se van a desarrollar, el alcance del proyecto, cronograma y tecnología.</p> <p>Con esta base se establece el número de desarrolladores y las capacidades que se requieren y sólo entonces ya se puede distribuir las tareas a los desarrolladores.</p> <p>Es imprescindible poner especial atención a:</p> <ul style="list-style-type: none"> •Capacidades individuales y sus destrezas (Conocer características del desarrollador) •Actitud – Si bien no es medible cuantitativamente es necesaria para emprender un desarrollo. La pro-actividad, disposición al trabajo y de servicio al cliente son características exclusivas de los recursos humanos y necesarias al asignar tareas. •Experiencia – El conocimiento de un desarrollador sobre las actividades y reglas de negocio de un cliente determinan quienes deben realizar o coordinar ciertas tareas.
4	<ul style="list-style-type: none"> •Nivel de conocimiento de la tecnología. •Complejidad de las tareas. •Disponibilidad del recurso.
5	En base a los requerimientos se extraen las herramientas necesarias y se previene la dificultad de los mismos, para posteriormente contrastarlo con los recursos disponibles.
6	Se debe analizar el proyecto, organizar el equipo de trabajo, definir el alcance y prioridad del mismo, en base a esos tópicos definidos, se realiza una relación con los recursos que se cuentan y luego se dividen equitativamente las cargas de trabajo.

Fuente: León, Sánchez (2020)

Breve análisis: En base a la información proporcionada, los criterios que se deben tomar en cuenta para hacer una correcta distribución de los equipos, se debe considerar el nivel de conocimiento de cada uno de los integrantes con respecto al proyecto en desarrollo, así como también las restricciones de tecnología que existan en el mismo, considerar los recursos disponibles y el conocimiento del modelo de negocio del cliente, además se debe establecer una estructura jerárquica en el equipo de trabajo para dividir las cargas equitativamente y cumplir con el cronograma establecido.

2. ¿Cómo determina usted el tiempo que requiere para efectuar un proyecto?

Tabla 2. Respuestas pregunta n°2 entrevista.

INFORMANTE CLAVE N°	RESPUESTAS
1	<p>Me parece que lo primero es delimitar el alcance del proyecto para saber exactamente qué se requiere lograr, para poder segmentar en etapas o fases.</p> <p>Luego se estiman las actividades que generalmente se realizarán según el tipo del proyecto a realizar.</p> <p>Finalmente se asigna un tiempo aproximado para cada actividad, tomando en cuenta elementos claves como curva de aprendizaje, prelación entre actividades, recursos disponibles y pruebas, entre otros, para poder estimar un consolidado de la duración del proyecto.</p>
2	<p>El tiempo va comprendido desde el primer momento donde te reúnes con el cliente para conocer el alcance y dimensión del proyecto hasta el tiempo de estabilización.</p> <p>Se establecen las distintas etapas de acuerdo a lo solicitado y luego en pequeños entregables dentro de cada etapa, a cada entregable se le asigna un tiempo. La sumatoria de tiempo es el tiempo del proyecto.</p>
3	<p>A través del cálculo de la Ruta Crítica y se obtiene sumando el tiempo de las actividades que van en secuencia y se identifica la ruta o secuencia más larga.</p> <p>La ruta más larga es la duración estimada del proyecto. Con metodologías Ágiles se van realizando entregas parciales y el proyecto se divide en micro-proyectos con entregables implementables y la suma de ellos es la duración estimada del proyecto.</p>
4	<p>Valoración empírica de:</p> <ul style="list-style-type: none"> • Base del proyecto: hacer desde 0, proyectos similares, basado en solución existente. • Pertinencia: prioridad, obligatoriedad técnica o normativa. • Tamaño del equipo. • Proyección de esfuerzo general: mejora simple, mejora media, mejora compleja, proyecto simple, proyecto medio, proyecto complejo.

	<ul style="list-style-type: none"> • Revisión de esfuerzo por tarea con el equipo de trabajo.
5	Con análisis de la prioridad del proyecto, comprensión de las necesidades del mismo, cantidad de trabajadores en el mismo y en base a esto se realiza una planificación y una estimación de los tiempos de entrega.
6	En base al análisis ya realizado y la prioridad ya definida del cliente se verifica el equipo disponible y cada una de sus características, una vez realizado esto, se realizan estimaciones de tiempo, considerando un porcentaje de margen de error por cualquier imprevisto que pueda suceder.

Fuente: León, Sánchez (2020)

Breve análisis: En base a las respuestas proporcionadas por los informantes claves, el tiempo o análisis de tiempo requerido para realizar determinado proyecto, se debe considerar diversos factores que permiten hacer una estimación del periodo de tiempo necesario para ejecutar la fase esperada en el mismo, entre ellos se deben considerar el alcance del plan a ejecutar, los requerimientos del cliente los cuales deben ser establecidos y analizados correctamente para llevarlos a cabo con éxito en la fase de desarrollo. En cuanto a la manera de cómo se gestionará el mismo, deben seleccionarse las metodologías que permitan llevar a cabo el proyecto con éxito, además de la prioridad que se tenga y la calidad del equipo con que se cuente (Experiencia y conocimiento).

3. Enumere las causas que se han presentado por las cuales no se han podido concluir los proyectos en el tiempo planeado.

Tabla 3. Respuestas pregunta n°3 entrevista.

INFORMANTE CLAVE N°	RESPUESTAS
1	1. Errores al estimar el tiempo de las actividades. 2. No tener claro el alcance del proyecto.

	<ol style="list-style-type: none"> 3. Desconocimiento de procesos claves o reglas de negocio del proyecto. 4. Cambios de los requerimientos acordados.
2	<ol style="list-style-type: none"> 1. El alcance no estaba bien establecido 2. No hay claridad del requerimiento 3. El cliente no sabe lo que realmente quiere o necesita 4. Los tiempos son subjetivos y puesto por el líder sin un análisis previo 5. Los desarrolladores no comprenden las especificaciones 6. Los desarrolladores no tienen claridad de la complejidad de los procesos
3	<ol style="list-style-type: none"> 1. Mal diseño (Levantamiento incompleto de información, fallas al establecer los alcances, desviaciones en la ruta crítica por asignación incorrecta de tareas y definición de las mismas, entre otras) 2. Desarrolladores (Retiros inesperados, exceso de confianza u optimismo, falta de seguimiento del responsable del proyecto, falta de conocimiento en la arquitectura a desarrollar – Error de selección del recurso). 3. Modificación de los alcances por parte del cliente. 4. Sobreasignación de tareas (mala distribución) y equipo desmotivado. 5. Incumplimiento de terceros, fallas de tecnología que forman parte de la solución, pero no del desarrollo (Servidores, conexiones, licencias), otros problemas (Electricidad, Internet, transporte, entre otros).
4	<ol style="list-style-type: none"> 1. Requerimientos y especificaciones incompletas. 2. Cambios frecuentes en los requerimientos y especificaciones. 3. Rotación de personal. 4. Escasa participación del usuario. 5. Planificación realizada sin consideración de riesgos.
5	<ol style="list-style-type: none"> 1. Mala comunicación con el cliente, no se especifican los requerimientos con precisión, mala planificación o falta de recursos para el desarrollo del mismo.
6	<ol style="list-style-type: none"> 1. No se tienen claro los requerimientos 2. Falta de interés y responsabilidad en el trabajo a realizar. 3. Mala comunicación 4. Liderazgo deficiente 5. Cambios en el equipo de trabajo

Fuente: León, Sánchez (2020)

Breve análisis: Las causas arrojadas por los resultados de la entrevista a cada uno de los informantes nos permiten dar a conocer las principales razones por la cual un proyecto no es ejecutado satisfactoriamente, entre ellos podemos mencionar la mala estimación de tiempo o alcance del mismo, desconocimiento o falta de interés por parte de los responsables de ejecución, el cliente no tiene claro lo que requiere o continuos cambios de requerimientos, falta de comunicación, movimiento o rotación de personal y presión excesiva. Cada uno de estos motivos puede causar el incumplimiento de los tiempos establecidos para la elaboración del mismo.

4. En su experiencia personal y profesional, ¿qué criterios utilizaría usted para evaluar la calidad de un código?

Tabla 4. Respuestas pregunta n°4 entrevista.

INFORMANTE CLAVE N°	RESPUESTAS
1	<ul style="list-style-type: none"> · Cumplimiento de estándares de la tecnología. · Optimización de llamadas a base de datos · Uso de archivos de propiedades en vez de código duro. · Escalabilidad del código.
2	El uso de herramientas que realicen la evaluación correspondiente, donde el análisis tenga una buena interpretación de sus resultados.
3	<ul style="list-style-type: none"> • Debe cumplir con los estándares decodificación y las mejores prácticas establecidas en la compañía • Pruebas unitarias y pruebas funcionales • Aplicar prueba para descartar fallos potenciales • Debe estar correctamente comentado • Evitar dependencias cíclicas • Por último y más importante, evaluar que cumpla con lo establecido en la tarea (Lee lo que debe leer, graba lo que debe grabar y retorna lo que debe retornar),
4	<ul style="list-style-type: none"> • Grado de cumplimiento de los requisitos y especificaciones planteados. • Precisión en su desempeño. • Eficiencia en su desempeño.

	<ul style="list-style-type: none"> • Eficiencia en la gestión de datos. • Cumplimiento de estándares propios de la unidad de desarrollo, y en caso de que estos no se encuentren definidos, cumplimiento de estándares referenciales ‘Clean Code’. • Facilidad de comprensión y mantenimiento. • Facilidad de reutilización. • Integridad de acceso y/o auditoría. • Adaptabilidad post-productiva. • Trazabilidad.
5	Eficiencia en los procesos, facilidad de comprensión para futuras reutilizaciones u actualizaciones, integridad en los resultados y lograr los requerimientos exigidos.
6	Para evaluar la calidad de código, se debe seguir una serie de estándares de programación tales como un conjunto de pruebas unitarias para verificar posibles fallas en el código desarrollado.

Fuente: León, Sánchez (2020)

Breve análisis: Evaluar la calidad de un código es uno de los tópicos más complejos en el área de la informática, por diversos factores que se escapan al método de cuantificación de variables, es decir, no se puede medir, sin embargo, existen otros métodos que permiten evaluar la calidad del software desarrollado. En base a las réplicas proporcionadas por los informantes, dichas etapas pueden englobarse de la siguiente manera.

El cumplimiento de los estándares establecidos, la escalabilidad del software, pruebas unitarias realizadas a los módulos desarrollados. El cumplimiento de cada uno de estas fases, permite visualizar los errores que existan, corregirlos y hacer continuas del sistema, con esto se logra desarrollar un código estable a prueba de errores.

5. ¿Cuáles considera deberían ser los principales factores para evaluar la calidad de un desarrollador?

Tabla 5. Respuestas pregunta n°5 entrevista.

INFORMANTE CLAVE N°	RESPUESTAS
------------------------	------------

1	<ul style="list-style-type: none"> · Cumplimiento de los desarrollos. · Escalabilidad. · Documentación de su código. · Inclinación a buscar nuevas tecnologías para dar mejores soluciones.
2	Calidad de su código, buena batería de pruebas y buen evaluador de los criterios de aceptación.
3	<ul style="list-style-type: none"> • Capacidad de análisis – Entender que se quiere, saber cómo lograrlo de la manera más simple, rápida y confiable - + Análisis y - Intuición • Responsabilidad • Manejo de tecnología, aplicaciones, lenguajes y repositorios • Capacidad de Aprendizaje - No necesitar saber todo, pero si donde encontrarlo • Actitud y aptitud • Interés por conocer el negocio. - Un buen desarrollador sabe lo que su cliente necesita porque sabe lo que su cliente hace • Madurez y Ética • Trabajo en equipo
4	<ul style="list-style-type: none"> • Curiosidad e interés por el desarrollo. • Comprensión de requisitos. • Buen uso de frameworks. • Codificación coherente y organizada. • Uso correcto de algoritmos, patrones y estructuras. • Uso correcto de librerías, clases,. variables, archivos y otros elementos inherentes al desarrollo • Buen manejo de errores y excepciones. • Aplicación de pruebas unitarias y funcionales. • Cumplimiento de requisitos.
5	Codificación bien estructurada y comprensible, variedad de conocimientos, constante evolución de los mismos, buen manejo de herramientas y cumplimiento de sus deberes.
6	El cumplimiento de estándares de programación, la documentación realizada, actitud frente a los problemas que se presenten y capacidad de trabajo en equipo.

Fuente: León, Sánchez (2020)

Breve análisis: En contrastación con la pregunta anterior, para definir la calidad de un desarrollador en sí y no del código desarrollado, se pueden tomar algunas

características del desarrollador, teniendo en cuenta las respuestas de los informantes claves, podemos definir los siguientes tópicos.

El correcto cumplimiento del diseño y/o los requerimientos establecidos, la creación de documentación corregida y entendible para usuarios no técnicos, el manejo y experiencia de la tecnología, la capacidad de trabajo en equipo, la capacidad de análisis y el interés prestado a determinado trabajo a realizar. El correcto cumplimiento de los mismos puede garantizar que el desarrollador seleccionado es de calidad.

6. En relación a los proyectos que no son entregados en el tiempo planeado. Este número, ¿es menor al deseado, dentro de los límites deseados o mayor al deseado?

Tabla 6. Respuestas pregunta n°6 entrevista.

INFORMANTE CLAVE N°	RESPUESTAS
1	Siempre que no sea 0, el número va a ser mayor al deseado, la idea es ser realista con los tiempos de entrega para no prometer proyectos que no se puedan entregar a tiempo.
2	Mayor al deseado
3	En promedio “Menor al deseado” – Siempre hay espacio a mejorar.
4	El número es mayor al deseado.
5	Mayor al deseado.
6	Mayor al deseado.

Fuente: León, Sánchez (2020)

Breve análisis: Según las réplicas proporcionadas por los informantes claves, el número de proyectos que no son entregados en el tiempo planeado siempre es mayor a deseado, esto quiere decir que existen proyectos que no son entregados a tiempo. Es importante tener definidas los requerimientos para no comprometerse a entregar resultados en un tiempo improbable.

7. En su entorno laboral, ¿aplican metodologías de desarrollo de software y que aspectos piensa que son los más importantes de estas?

Tabla 7. Respuestas pregunta n°7 entrevista.

INFORMANTE CLAVE N°	RESPUESTAS
1	Sí, es usa la metodología Ágil, SCRUM, no estrictamente aplicada, pero sí con sus bases, se realiza el trabajo por iteración, donde en cada una se planifican entregables, se asignan puntos de dificultad a las actividades para facilitar la distribución equitativa y que todo el equipo tenga una carga de trabajo similar.
2	Siempre el uso de la metodología es importante para controlar el seguimiento de las etapas del proyecto y evaluar los resultados que se van obteniendo.
3	<p>Metodologías Ágiles</p> <ul style="list-style-type: none"> • Permiten adaptar la forma de trabajo a las condiciones del proyecto • Es flexible al cambio y a la adaptación • Reduce costos e incrementa la productividad • Al escuchar las opiniones (cliente / equipo) se aumenta la motivación y se obtienen más respuestas y menos desviaciones. • Esta metodología intenta siempre mantenerse dentro de los tiempos de entrega. • Detección rápida de errores. La entrega continua a avances permite corregir errores e implementar mejoras. • La interacción entre el cliente y los desarrolladores tiene como objetivo cumplir las necesidades del cliente.
4	<p>Sí, dependiendo del tipo de proyecto:</p> <ul style="list-style-type: none"> • Metodología de cascada: en desarrollos con requisitos y metas claras, sobre los que ya se tienen experiencias previas similares. • Prototipo: para el desarrollo de sistemas experimentales que puede servir de modelo para evaluar la viabilidad de un sistema más amplio. • Metodologías ágiles: para proyectos con nivel de incertidumbre alto y susceptibilidad a cambios frecuentes.
5	<p>Si se aplicación en relación al proyecto:</p> <ul style="list-style-type: none"> • Prototipo: Para los proyectos experimentales, para la evaluación de viabilidad del sistema es un aspecto más amplio y con todas sus funcionalidades. • Cascada: Aplicada con frecuencia a los equipos con mayor

	<p>experiencia para optimizar los resultados y tiempos de entrega.</p> <ul style="list-style-type: none"> • Mitologías ágiles: Para proyectos cambiantes, estos son los que cambian frecuentemente ya sea por mala extracción de los requisitos por parte del equipo o falta de especificación por el usuario.
6	Se aplican diversas metodologías de desarrollo según el proyecto que se presente y las necesidades del cliente, sin embargo, no son rigurosamente empleadas, solo se implementan para definir las etapas del mismo.

Fuente: León, Sánchez (2020)

Breve análisis: En base a las reacciones de los informantes claves, efectivamente si se aplican metodologías de desarrollo de software, sobre todo Ágil y Scrum, las mismas no se aplican estrictamente, sin embargo se planifican entregables y se realiza una distribución equitativa de las etapas del proyecto y así garantizar que los integrantes del equipo tengan cargas de trabajo similares, esto también permite realizar un seguimiento de las etapas del proyecto y evaluar los resultados que te obtienen, además de la posibilidad de la detección de errores rápida y la reducción de costos.

8. Ante una eventual falta de organización y/o comunicación (situaciones extraordinarias), ¿cree usted que las metodologías de desarrollo de software se adaptan y ayudan a remediar estas?

Tabla 8. Respuestas pregunta n°8 entrevista.

INFORMANTE CLAVE N°	RESPUESTAS
1	<p>Si, las metodologías ayudan a que exista una organización del proyecto, mientras que se cumplan las normas.</p> <p>Mientras que, si la comunicación falla, no importa que metodología se use, es muy probable que el proyecto no sea exitoso.</p>
2	Ayudan a remediar las situaciones extraordinarias siempre y cuando se tenga un buen control del proyecto
3	Las metodologías ágiles se caracterizan por su adaptabilidad, sin embargo, considero que la prolongada falta de organización y/o

	comunicación a la larga no es manejable por ninguna metodología.
4	<p>Las metodologías de desarrollo de software correctamente implementadas permiten de trabajar con cierto grado de incertidumbre.</p> <p>Levantando la información de algunos de los riesgos latentes de un proyecto se puede generar un plan para adaptarse o minimizar el impacto de ciertas situaciones extraordinarias.</p> <p>También existirán situaciones o riesgos no considerados para los cuales se deberá actuar de forma reactiva y resolutiva.</p>
5	<p>Si, sobre todo las metodologías ágiles si son bien implementadas permiten trabajar haciendo cambios precisos para no afectar en gran medida los resultados.</p> <p>Cabe destacar que son capaces de tener previsto algunos riesgos en la planificación y permiten adaptarse fácilmente en casos extraordinarios no considerados.</p>
6	Ayudan a remediar los posibles inconvenientes que se presenten siempre y cuando no sean demasiado devastadores además de considerar los problemas de comunicación, cosa que no son controladas mediante ninguna metodología de desarrollo.

Fuente: León, Sánchez (2020)

Breve análisis: Ante las respuestas proporcionadas por cada uno de los informantes claves, detallamos lo siguiente, en la existencia de una situación extraordinaria en un proyecto, ya sea por inconvenientes u organización y/o comunicación, si, las metodologías de desarrollo de software ayudan a repeler dichos obstáculos, teniendo en cuenta que estas metodologías se caracterizan por su adaptabilidad y mientras se tenga un buen control del proyecto se solucionan los problemas más fácilmente, sin embargo cabe destacar que en la falta de comunicación, ninguna metodología ayudará a solucionar las dificultades presentadas.

9. En relación a situaciones extraordinarias que presenten un cambio en la situación del proyecto (Como el retiro de una persona, algún problema que le impida trabajar o similares) ¿Qué curso de acciones tomaría usted para reevaluar el proyecto en curso?

Tabla 9. Respuestas pregunta n°9 entrevista.

INFORMANTE CLAVE N°	RESPUESTAS
1	Se debe evaluar si existe alguna persona que sustituya temporalmente a la persona afectada, para continuar el desarrollo con la menor afectación posible. De no estar disponible otro desarrollador, tocaría dividir las actividades restantes entre los otros miembros del equipo.
2	Evaluar el impacto de la situación extraordinaria y tomar la acción que corresponda de acuerdo al evento extraordinario.
3	<ul style="list-style-type: none"> • Identificar las tareas asignadas a esa persona • Evaluar la disposición de desarrolladores con características similares • Reunir al equipo para identificar si alguna tarea puede ser reasignada a lo interno a otro desarrollador o distribuida entre varios desarrolladores. • Si las tareas forman parte de la ruta crítica realizar los ajustes requeridos en caso de que sea inevitable una desviación • Ajustar los planes de seguimiento para intentar cumplir el plazo • Notificar al cliente la desviación en caso de ser imposible lograr las metas en el tiempo estimado.
4	<ol style="list-style-type: none"> 1. Reevaluar el proyecto bajo el nuevo contexto. 2. Revisar el impacto de la situación en el alcance y tiempo de las tareas. 3. Realizar ajustes en la asignación de recursos, distribución y/o duración de las tareas. 4. Re planificar y/o cambiar alcance (si aplican).
5	Reevaluar el proyecto en su totalidad, considerando el impacto que producirá esto en los resultados y luego de evaluar la factibilidad de esos realizar la nueva planificación y los ajustes necesarios.
6	Verificar los avances realizados y comprobar si existe otro recurso que permita suplir al desarrollador retirado, en cualquier caso, se debe distribuir las nuevas cargas de trabajo y informar al cliente de lo ocurrido.

Fuente: León, Sánchez (2020)

Breve análisis: En el caso de que exista el retiro de algún miembro del equipo de desarrollo de software, los informantes clave mencionan que se debe verificar la

existencia de algún otro desarrollador disponible que pueda cubrir dicha ausencia, en el caso que no exista nadie disponible, se debe reevaluar el proyecto y distribuir las cargas a los miembros que queden, además de esto re planificar y modificar el alcance, todo esto considerando el conocimiento del cliente ante la situación.

10. ¿Qué funciones cree que son necesarios en una plataforma para la gestión de proyectos de software para ser útil al nivel de las empresas?

Tabla 10. Respuestas pregunta n°10 entrevista.

INFORMANTE CLAVE N°	RESPUESTAS
1	Básicamente lo necesario sería: planificación, seguimiento de actividades, reporte de horas por parte de los miembros del proyecto, estadísticas del proyecto y retroalimentación.
2	Colaborativa Accesible Segura
3	<ul style="list-style-type: none"> • Estimación y control de proyectos • Visualización de la ruta critica • Calendario de eventos
4	<ul style="list-style-type: none"> • Plataforma colaborativa. • Flujo de trabajo que abarque: <ul style="list-style-type: none"> -Solicitud del proyecto con detalle de requerimientos y especificaciones. -Estimación preliminar de tamaño del proyecto y fecha de atención. -Aprobación del proyecto. -Asignación de recursos. -Levantamiento de información. -Planificación del proyecto. -Seguimientos de etapas y tareas, con sus cumplimientos o desviaciones, notas de seguimiento. -Notificaciones. -Documentación del proyecto. -Hitos y cierre. • Integración con gestor de recursos, registro de horas, repositorio de documentos asociados al proyecto, GSuite. • Homologación y visualización del proyecto en múltiples

	formatos: Gantt, calendario, tarjetas de seguimiento, informes. <ul style="list-style-type: none"> • Simulación de escenarios. • Listas de selección, filtros, búsqueda, códigos de color, etiquetas, actualizaciones de estado, plantillas, incorporación de adjuntos y otros utilitarios para organizar y visualizar la información.
5	Debe abarcar: <ul style="list-style-type: none"> • Solicitudes de proyectos • Estimaciones o aproximaciones de tiempos de entrega. • Gestión y asignación de recursos • Informes periódicos • Seguimiento de los proyectos
6	<ul style="list-style-type: none"> • Módulo de notificaciones para cada uno de los integrantes • Simulación de escenarios, y eventos extraordinarios • Cronograma de ejecución

Fuente: León, Sánchez (2020)

Breve análisis: Esta pregunta se corresponde directamente con los requerimientos que deben existir una plataforma colaborativa para la gestión de proyectos de software, las cuales según para informantes clave son necesarios las siguientes características:

- Seguimiento de actividades en el proyecto
- Estadísticas o métricas del proyecto,
- Retroalimentación por parte de los integrantes del equipo de desarrollo
- Ser colaborativa
- Detallar un flujo de trabajo
- Gestionar las solicitudes de proyectos
- Generar estimaciones y aproximaciones en los tiempos de entrega
- Asignación de recursos

Todos y cada uno de estos requerimientos funcionales permiten apoyar al equipo de desarrollo y sobre todo al líder de proyectos para la correcta organización y garantizar el cumplimiento de las metas establecidas.

Por otra parte, los requerimientos no funcionales descritos se definen a continuación:

- Colaborativa

- Accesible
- Segura

Los cuales con la integración de cada uno de estos requerimientos se garantizan un correcto funcionamiento de la plataforma y cada uno de los requerimientos funcionales.

11. ¿Qué herramientas externas (GitHub, SonarQube, etc.) opina usted que ¿son necesarias de integrar en este tipo de plataformas y por qué?

Tabla 11. Respuestas pregunta n°11 entrevista.

INFORMANTE CLAVE N°	RESPUESTAS
1	<p>Eso depende de las necesidades de integración que se tengan, si unas de las prioridades es evaluar la calidad de código, sería importante tener herramientas como SonarQuebe, por ejemplo.</p> <p>O si ya se ha alcanzado una madurez en el nivel de desarrollo, se pueden integrar herramientas como jenkins, para lograr que despliegues de las aplicaciones marquen tareas como finalizadas en tu sistema de seguimiento del proyecto.</p>
2	<p>Herramientas como slack que permiten establecer una plataforma comunicativa que podría complementar la plataforma de gestión de proyectos.</p>
3	<ul style="list-style-type: none"> • GitHub: Herramienta de colaboración, revisión y gestión de código • Docker: Contenedores para construir, gestionar y proteger aplicaciones software y para desplegarlas en cualquier lugar • Jenkins: Servidor de automatización de código abierto, en concreto, un servidor de integración continua. • Slack: Herramientas de colaboración. La gestión de proyectos en Slack se realiza principalmente a través de canales como chats y conversaciones sobre proyectos. • Katalon Studio: Prueba de automatización • Proyectos: MS Project, Trello, Asana y Clockify • Escritorio Remoto: Múltiples herramientas de acceso remoto.
4	<p>La integración de herramientas externas de apoyo al desarrollo puede considerarse dentro de un dashboard de todas las aplicaciones que se utilizaran en el ciclo de vida del proyecto.</p>

	También se puede considerar su integración para relacionar de forma automática entregables del proyecto, código, calidad del código, documentación asociada, test realizados, etc., a partir de los cuales se pudiesen generar indicadores adicionales para la valoración del proyecto y el equipo de proyecto.
5	La implementación de herramientas externas puede ser bastante útil para la evaluación de los proyectos y por consiguiente en la gestión de las mismas, en lo particular herramientas para el control de versiones como Git que faciliten la revisión de los cambios o herramientas que permitan la evaluación de calidad del código
6	Integración de herramientas como github permiten llevar un control o estadística acerca de cómo se están desarrollando los proyectos en tiempo real y además también la integración de instrumentos como clockify o Trello permiten al líder de proyectos ver como se están aprovechando las horas en la ejecución del plan.

Fuente: León, Sánchez (2020)

Breve análisis: Las herramientas de desarrollo de código como GitHub o SonarQube, ayudan al programador en el proceso de desarrollo de software, ya sea mediante un sistema de control de versiones o una herramienta de evaluación de código fuente, los informantes claves mencionan que la integración de estas herramientas dependen de las necesidades que se tengan, desde la implementación de SonarQube hasta la utilización de Jenkins o como también herramientas de gestión de tareas como Trello o Clockify para el reporte de horas por desarrollador, con el propósito de evaluar el rendimiento de los programadores y/o la calidad del código.

4.2 Fase II: Determinación de los requerimientos funcionales y no funcionales para una plataforma colaborativa basada en algoritmo de aprendizaje profundo, en la gestión de proyectos de software, en función al diagnóstico que se realizara.

Luego de realizado toda la recolección de datos se procedió con un análisis riguroso de toda la información obtenida, para luego proceder con la determinación de los requerimientos, tanto funcionales como no funcionales, de la plataforma colaborativa

en cuestión y así llegar a una conclusión mucho más clara, específica y total funcional para realizar su posterior aplicación y realización de pruebas.

4.2.2 Actividad I: Análisis y definición de los requisitos funcionales y no funcionales de la plataforma colaborativa.

Se realizó el análisis requerido de la información obtenida a través de las respectivas herramientas de recolección de datos, las cuales fueron definidas con anterioridad. Adicional, se tomaron en consideración las observaciones adicionales que se produjeron en el proceso de la entrevista, con la finalidad de definir lo mejor posible las bases fundamentales de la plataforma colaborativa, que son los requerimientos funcionales y no funcionales.

A continuación, se presentarán en forma de lista los requerimientos:

•Requerimientos funcionales:

- * Registrar usuarios
- * Gestionar perfil
- * Registrar y gestionar organizaciones
- * Registrar y gestionar proyectos
- * Realizar y gestionar solicitudes
- * Integrar herramientas externas (Git o SonarQube).

•Requerimientos no funcionales:

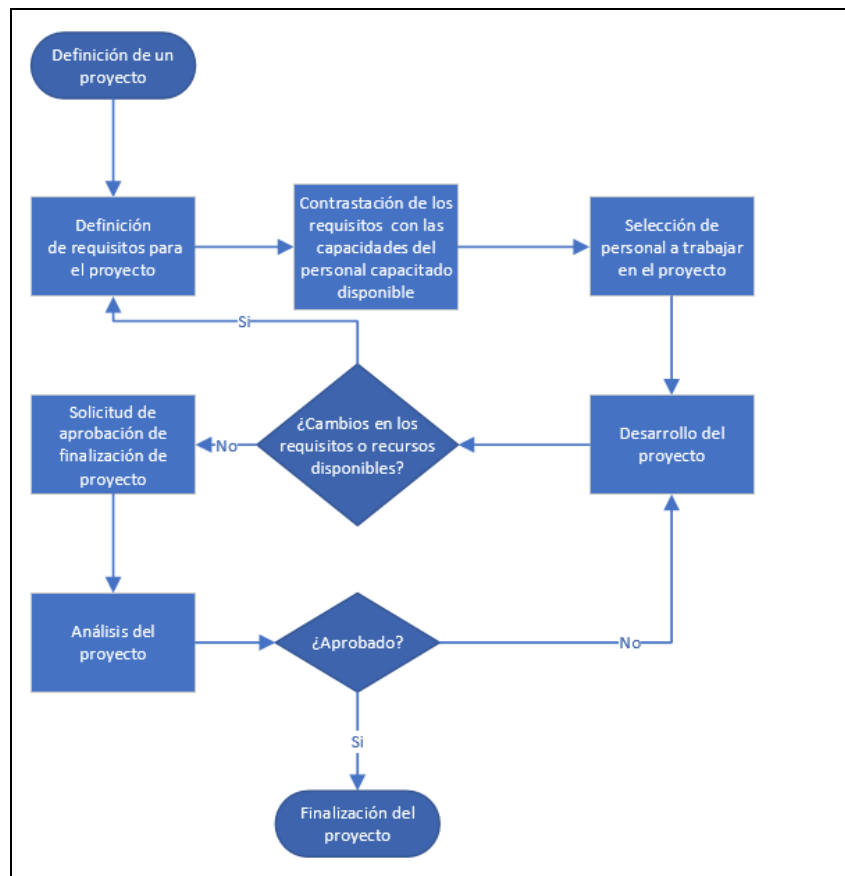
- * Realizar respaldo de la base de datos de forma periódica.
- * Proteger el sistema con los estándares básicos de seguridad.
- * Mantener un diseño uniforme en la interfaz en toda la plataforma.
- * Realizar todas las validaciones necesarias para garantizar la integridad de los datos ingresados en la plataforma.
- * Garantizar la escalabilidad del sistema mediante una adecuada documentación, diseño y desarrollo.

4.3 Fase III: Diseño de las bases de los modelos de gestión de proyectos para la plataforma colaborativa mediante la metodología XP.

Una vez completada la fase de análisis y determinados los requerimientos funcionales y no funcionales del sistema, se procedió a diseñar los modelos de gestión de proyectos para aplicarlos en la plataforma colaborativa, adicional a esto se realizó el modelado de toda la diagramación básica de los sistemas de información que incluye modelación de base de datos, diseño de casos de uso y diseño de la estructura general de la plataforma colaborativa.

4.3.1 Actividad I: Modelado De proceso de gestión de un proyecto.

Gráfico 1. Diagrama de proceso.



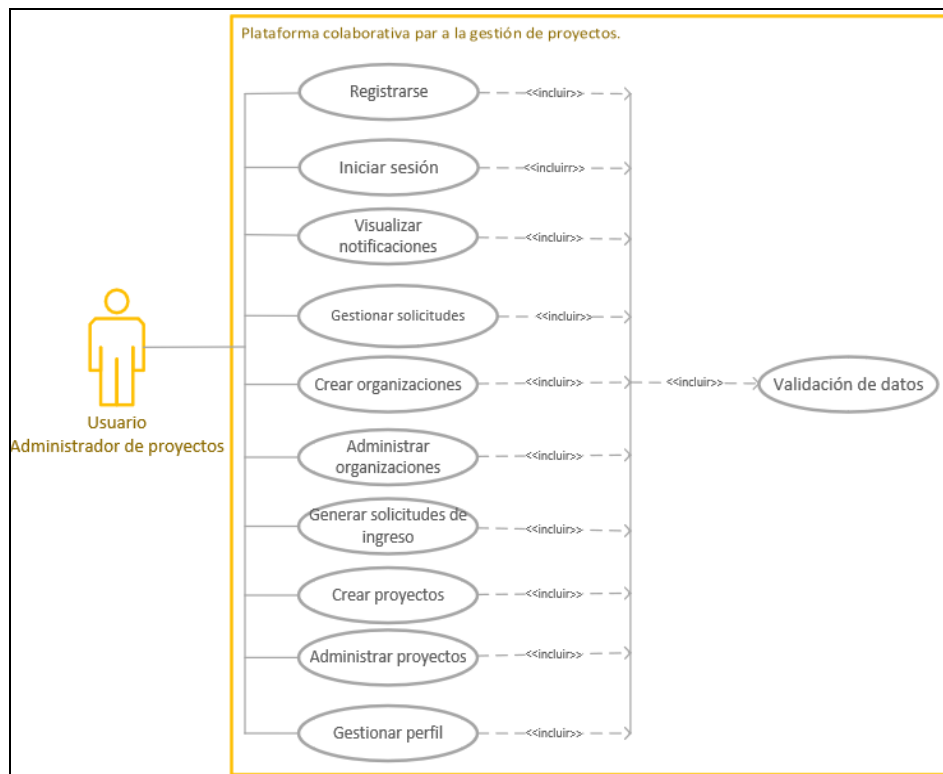
Fuente: León, Sánchez (2020)

4.3.2 Actividad II: Diseño de casos de uso.

Los casos de uso son parte del conjunto incluido en el lenguaje de Modelado Unificado (UML), que permite mostrar las diferentes acciones que pueden tomar los diferentes actores dentro del sistema. A continuación, se muestran los diagramas correspondientes a cada actor y sus respectivas descripciones:

Administradores de proyectos: Estos son los encargados de gestionar organizaciones y proyectos, modificando los miembros que integran a estos 2 en tiempo real, se ramifican en diferentes grados que son definidos para limitar y representar de forma mas adecuada el control que se lleva en una organización.

Gráfico 2. Diagrama de caso de uso (Administrador de proyectos).

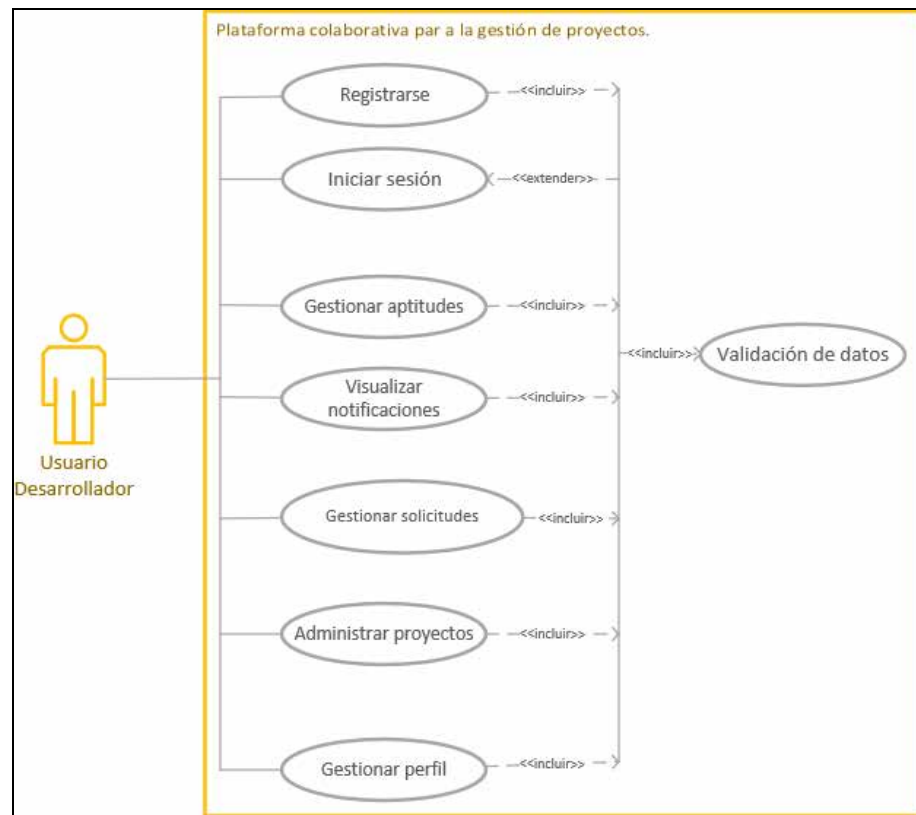


Fuente: León, Sánchez (2020)

Desarrollador: Es el usuario con menores funciones disponible en el sistema, pero el mas indispensable en el mismo. Estos dentro de la plataforma podran

definir sus aptitudes en las diferentes herramientas que se encuentran definidas en el mismo, también realizan funciones similares al administrador (Gestión de perfil, notificaciones, etc.), pero a diferencia de estos no cumplen funciones de gestión de organización y no están encargados de dar la aprobación de un proyecto ni crear uno dentro de la plataforma. Cabe destacar, que aunque no definen la aprobación de un proyecto, estos sí pueden gestionarlo internamente con los controles de versiones, generación de solicitudes de aprobación, verificación del mismo y otras funciones adicionales.

Gráfico 3. Diagrama de caso de uso (Desarrollador).



Fuente: León, Sánchez (2020)

4.3.3 Actividad III: Descripción de casos de uso.

Tabla 12. Caso de uso (Registrarse).

Registrarse	
Actor: Administrador de proyecto, Desarrollador.	
Objetivo: Obtener credenciales para acceder a las funciones de la plataforma.	
Precondición: No estar registrado en la plataforma.	
Flujo normal: <ol style="list-style-type: none"> 1. Entrar a la sección de ingreso de la plataforma. 2. Posicionarse en la sección de registro. 3. Llenar el formulario con los datos solicitados. 4. Presionar el botón para registrar. 5. Esperar la validación de los datos. 6. Entrada a la plataforma. 	Flujo alterno: <ol style="list-style-type: none"> 1. Las credenciales ingresadas en el sistema son inválidas. 2. Se genera una alerta para informar al usuario del error. 3. No se da acceso a la plataforma, hasta ingresar las credenciales correctas.
Postcondición: Se accede a la plataforma y se registran las credenciales.	

Fuente: León, Sánchez (2020)

Tabla 13. Caso de uso (Iniciar sesión).

Iniciar sesión	
Actor: Administrador de proyecto, Desarrollador.	
Objetivo: Acceder a las funciones de la plataforma.	
Precondición: Estar registrado en la plataforma.	
Flujo normal: <ol style="list-style-type: none"> 1. Entrar a la sección de ingreso de la plataforma. 2. Ingresar las credenciales (Correo y contraseña) en los campos requeridos. 3. Presionar el botón para ingresar. 4. Esperar la validación de los datos. 5. Entrada a la plataforma. 	Flujo alterno: <ol style="list-style-type: none"> 1. Las credenciales ingresadas en el sistema son inválidas. 2. Se genera una alerta para informar al usuario del error. 3. No se da acceso a la plataforma, hasta ingresar las credenciales correctas.
Postcondición: Se accede a la plataforma.	

Fuente: León, Sánchez (2020)

Tabla 14. Caso de uso (Gestionar aptitudes).

Gestionar aptitudes	
Actor: Desarrollador.	
Objetivo: Definir sus aptitudes en la plataforma.	
Precondición: Ingresar a la plataforma y no tener aptitudes inicializadas o solicitar permiso a los administradores del sistema para redefinirlas.	
Flujo normal: <ol style="list-style-type: none"> 1. Ingresar con sus credenciales a la plataforma. 2. Llenar el formulario. 3. Presionar el botón para guardar los datos del formulario. 4. Aptitudes registradas. 	Flujo alterno: <ol style="list-style-type: none"> 1. El usuario no registra los datos en el formulario. 2. Se redirige al usuario a esta misma sección hasta que inicialice (Nuevo en la plataforma) o redefina sus aptitudes (Solicitud)
Postcondición: Se accede a las funciones básicas de la plataforma.	

Fuente: León, Sánchez (2020)

Tabla 15. Caso de uso (Gestionar perfil).

Gestionar perfil	
Actor: Administrador de proyecto, Desarrollador.	
Objetivo: Editar su perfil en la plataforma.	
Precondición: Ingresar en la plataforma y solicitar permiso para editar a los administradores de la plataforma.	
Flujo normal: <ol style="list-style-type: none"> 1. Ingresar con sus credenciales a la plataforma, 2. Dirigirse a la sección de perfil. 3. Editar los datos en el formulario. 4. Presionar el botón de guardar para registrar los cambios. 5. Registro de los cambios. 	Flujo alterno: <ol style="list-style-type: none"> 1. Los datos ingresados en la plataforma son inválidos. 2. Se informa al usuario mediante una alerta.
Postcondición:	

Fuente: León, Sánchez (2020)

Tabla 16. Caso de uso (Visualizar notificaciones).

Visualizar notificaciones	
Actor: Administrador de proyecto, Desarrollador.	
Objetivo: Ver las notificaciones que están asociadas a su cuenta en la plataforma.	
Precondición: Ingresar en la plataforma.	
Flujo normal: <ol style="list-style-type: none"> 1. Ingresar con sus credenciales a la plataforma, 2. Dirigirse a la sección de inicio. 3. Visualizar la lista que se muestra en esta sección. 	Flujo alterno: <ol style="list-style-type: none"> 1. No hay notificaciones asociadas al usuario 2. Se muestra una lista de notificaciones vacías al usuario.
Postcondición:	

Fuente: León, Sánchez (2020)

Tabla 17. Caso de uso (Gestionar solicitudes).

Gestionar solicitudes	
Actor: Administrador de proyecto, Desarrollador.	
Objetivo: Dar respuesta a las solicitudes.	
Precondición: Ingresar en la plataforma en la sección de organizaciones y tener notificaciones/solicitudes asociadas a la cuenta.	
Flujo normal: <ol style="list-style-type: none"> 1. Visualizar la lista que se muestra en la sección de organizaciones. 2. Presionar el botón de acción. 3. Aceptar o rechazar la solicitud. 4. Si es aceptada la solicitud se genera alerta si el usuario ya pertenece a una organización 5. Luego de validar los datos se ingresa al usuario en cuestión en la organización. 	Flujo alterno: <ol style="list-style-type: none"> 1. No hay solicitudes asociadas al usuario 2. No se muestra una lista de solicitudes.
Postcondición: Ingresar a una organización o rechazar una solicitud.	

Fuente: León, Sánchez (2020)

Tabla 18. Caso de uso (Crear organización).

Crear organización.	
Actor: Administrador de proyecto.	
Objetivo: Registrar a la organización en la plataforma.	
Precondición: Ingresar en la plataforma en la sección de organizaciones y no pertenecer a una organización.	
Flujo normal: <ol style="list-style-type: none"> 1. Presionar el botón de crear organización. 2. Rellenar el formulario. 3. Presionar el botón para enviar la solicitud. 4. Se registra la solicitud 	Flujo alterno: <ol style="list-style-type: none"> 1. No se rellena el formulario correctamente. 2. Se informa mediante una alerta al usuario del error. 3. No se registra la solicitud.
Postcondición: Registrar la organización en la plataforma.	

Fuente: León, Sánchez (2020)

Tabla 19. Caso de uso (Gestionar usuarios de la organización).

Gestionar usuarios de la organización.	
Actor: Administrador de proyecto.	
Objetivo: Gestionar los integrantes de la organización.	
Precondición: Ingresar en la plataforma en la sección de organizaciones, pertenecer a una organización y tener el grado necesario dentro de la misma para esta función.	
Flujo normal: <p style="margin-left: 20px;">Flujo gestión:</p> <ol style="list-style-type: none"> a. Presionar el botón de acciones en el usuario a gestionar. b. Seleccionar acción. c. Validar la acción en la alerta que se presenta. <p style="margin-left: 20px;">Flujo añadir integrante:</p> <ol style="list-style-type: none"> a. Presionar le botón de añadir usuario. b. Ingresar el correo del usuario a añadir. c. Validar los datos. <ol style="list-style-type: none"> 2. Generar notificaciones respectivas a la solicitud o cambio. 	Flujo alterno: <ol style="list-style-type: none"> 1. No se rellena el formulario correctamente. 2. Se informa mediante una alerta al usuario del error. 3. No se registra la solicitud o el cambio.

3. Se registra la solicitud o el cambio.	
Postcondición: Registrar la organización en la plataforma.	

Fuente: León, Sánchez (2020)

Tabla 20. Caso de uso (Crear proyecto).

Crear proyecto.	
Actor: Administrador de proyecto.	
Objetivo: Registrar un nuevo proyecto en la plataforma.	
Precondición: Ingresar en la plataforma en la sección de proyectos.	
Flujo normal: <ol style="list-style-type: none"> 1. Rellenar el formulario en la sección de nuevos proyectos. 2. Presionar el botón de crear organización. 3. Seleccionar los integrantes del proyecto (se presentará una lista recomendada si la opción de selección automática fue seleccionada). 4. Presionar el botón para registrar el proyecto. 5. Se registra el proyecto. 	Flujo alterno: <ol style="list-style-type: none"> 1. No se rellenan los formularios correctamente. 2. Se informa mediante una alerta al usuario del error. 3. No se registra el proyecto.
Postcondición: Se notifica a todos los integrantes del nuevo proyecto sobre la creación del mismo.	

Fuente: León, Sánchez (2020)

Tabla 21. Caso de uso (Gestionar proyecto, administrador de proyectos).

Gestionar proyecto	
Actor: Administrador de proyecto.	
Objetivo: Gestionar solicitudes de revisión de proyecto o modificar los integrantes del mismo.	
Precondición: Ingresar en la plataforma en la sección de proyectos.	
Flujo normal: <ol style="list-style-type: none"> 1. Ingresar a la sección de proyectos activos. 2. Seleccionar el proyecto a gestionar. Flujo para avances:	Flujo alterno: <ol style="list-style-type: none"> 1. No se rellenan con los datos correctamente. 2. Se informa mediante una alerta al usuario del error. 3. No se registra los cambios.

<ul style="list-style-type: none"> a. Entrar a la sección de solicitud de avance. b. Seleccionar el botón revisión. c. Registrar aprobación o no. <p>Flujo para integrantes:</p> <ul style="list-style-type: none"> a. Entrar a la sección de integrantes. b. Seleccionar a un integrante en la lista para expulsar o presionar el botón para añadir un nuevo integrante. c. Se realiza una validación de la solicitud del cambio al usuario mediante una alerta. <p>3. Se registran los cambios.</p>	
Postcondición:	

Fuente: León, Sánchez (2020)

Tabla 22. Caso de uso (Gestionar proyecto, desarrollador).

Gestionar proyecto	
Actor: Desarrollador.	
Objetivo: Generar solicitudes de revisión de proyecto.	
Precondición: Ingresar en la plataforma en la sección de proyectos.	
<p>Flujo normal:</p> <ul style="list-style-type: none"> 1. Ingresar a la sección de proyectos activos. 2. Seleccionar el proyecto a gestionar. 3. Entrar a la sección de solicitud de avance. 4. Rellonar el formulario. 5. Presionar el botón para enviar la solicitud. 6. Se registra la solicitud 	<p>Flujo alterno:</p> <ul style="list-style-type: none"> 1. No se rellena el formulario correctamente. 2. Se informa mediante una alerta al usuario del error. 3. No se registra la solicitud.
Postcondición:	

Fuente: León, Sánchez (2020)

4.3.4 Actividad IV: Declaración de módulos y de estados en la plataforma.

Una vez definidos los casos de usos podemos definir y dividir sus acciones en diferentes módulos para poder hacer uso de interfaces que no estén cargadas de información excesiva e innecesaria y facilitar la navegación y visualización, todo esto para la comodidad del usuario.

Tabla 23. Módulos de la plataforma.

Módulos	
Nombre	Descripción
Inicio	Esta comprende la página de inicio, que contiene información básica sobre la plataforma y algo de su arquitectura y documentación.
Ingreso/Registro	Este contiene los formularios tanto para inicio de sesión como para el registro de usuarios.
Inicio (Post ingreso)	Es un modulo que contiene notificaciones básicas del sistema para que el usuario este notificado de los cambios que ocurran dentro de la organización.
Proyectos (Post ingreso)	
Organización (Post ingreso)	Contiene toda la información básica de la organización, desde el nombre hasta los integrantes y roles, en este se ejecutan los cambios de los roles, expulsión e invitación de los usuarios, también en este se muestran las invitaciones y se aceptan/rechazan por el mismo.
Perfil (Post ingreso)	Contiene los datos básicos del usuario.
Aptitudes (Post ingreso)	Aquí se definen las aptitudes de los desarrolladores (solo desarrolladores o ajustes automáticos del sistema).

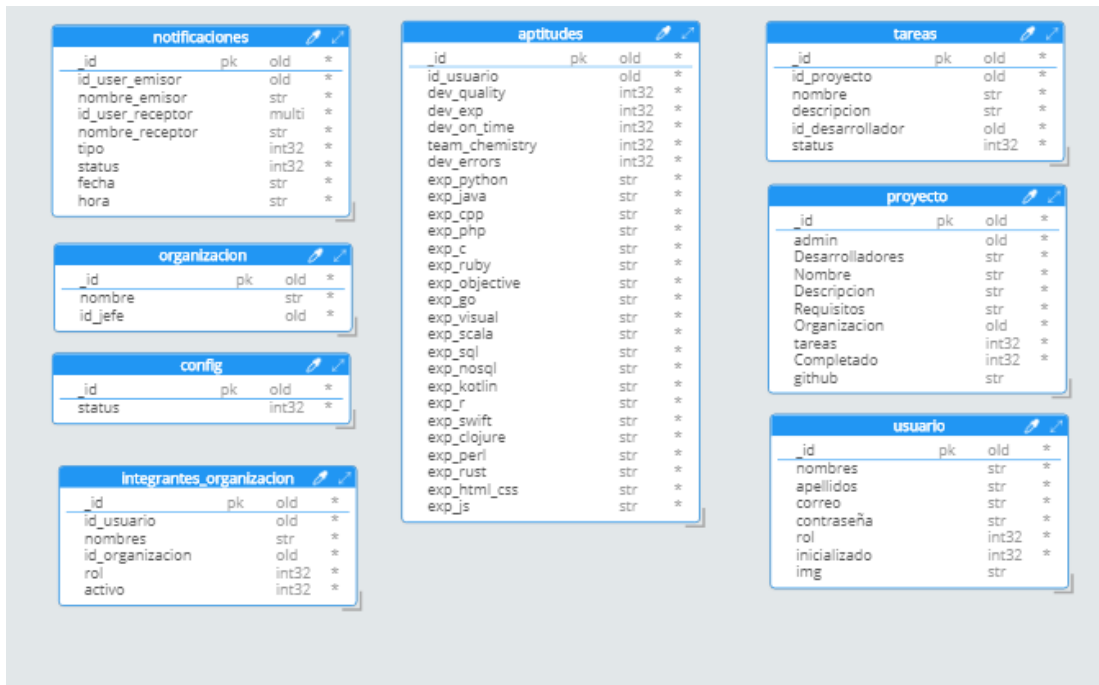
Fuente: León, Sánchez (2020)

4.3.5 Actividad V: Modelado de base de datos.

Una vez determinados los actores y sus casos de uso, se creó el modelo de base de datos, en este caso modelo un modelo relacional no SQL, que nos da una

representación gráfica de cómo serán almacenados los datos en la plataforma colaborativa. En base a este modelo, será implementada la base de datos.

Gráfico 4. Diagrama de base de datos.

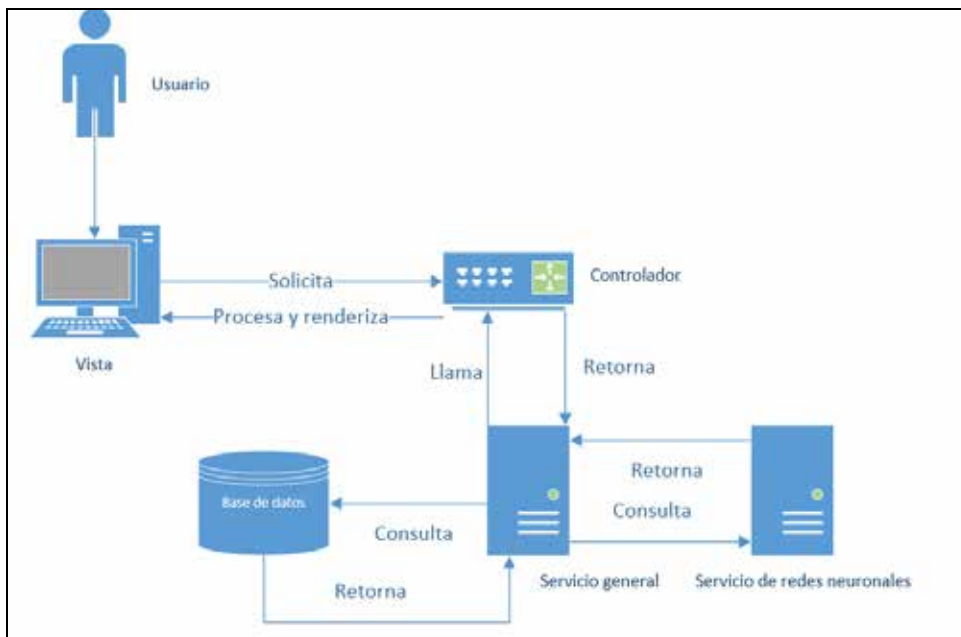


Fuente: León, Sánchez (2020)

4.3.6 Actividad VI: Descripción de la arquitectura del sistema.

La arquitectura del sistema es un conjunto de patrones los cuales dictan como será desarrollado el sistema, así mismo, proporcionan un marco que facilita el desarrollo del mismo. Para el presente proyecto se hizo uso del patrón Modelo Vista-Controlador, este plantea la separación de los componentes que componen el sistema en capas, las cuales son: las interfaces con el usuario, la lógica de programación y la interfaz con los datos. De igual manera la lógica se dividió en 2 servicios separados, los cuales son, un servicio que conlleva las acciones o funciones básicas, el cual se encarga de realizar las consultas al otro servicio, y este ultimo se encarga de las operaciones que realizan las redes neuronales.

Gráfico 5. Diagrama de arquitectura de la plataforma.

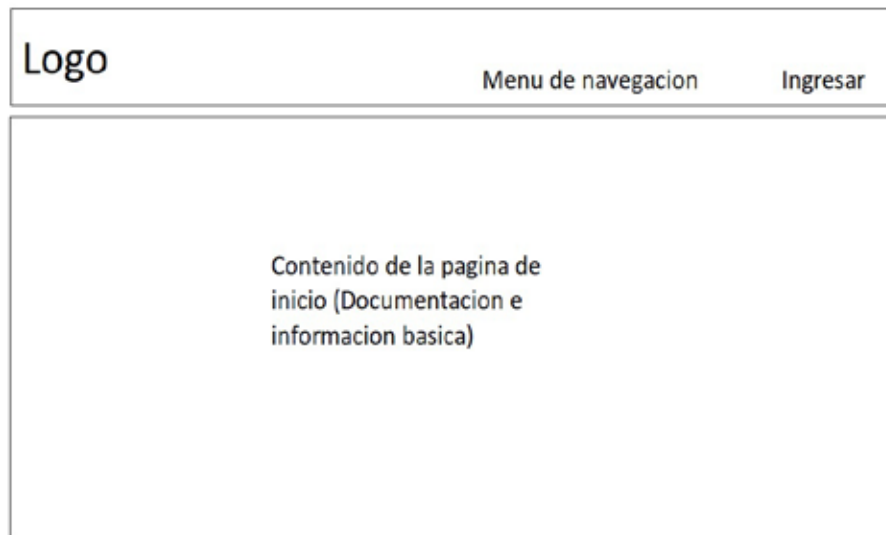


Fuente: León, Sánchez (2020)

4.3.7 Actividad VII: Diseño de interfaces.

Para este diseño se tuvo en consideración que estas interfaces deben permitir al usuario realizar todos los procesos de forma sencilla e intuitiva, aplicando un patrón visual a lo largo de toda la plataforma. Adicionalmente se mantuvo una paleta de colores oscuros por las crecientes corrientes de diseños de interfaces actual. Para este diseño se realizaron varias secciones con distintas funciones, las cuales se verán representadas a continuación con esquemas de los diseños.

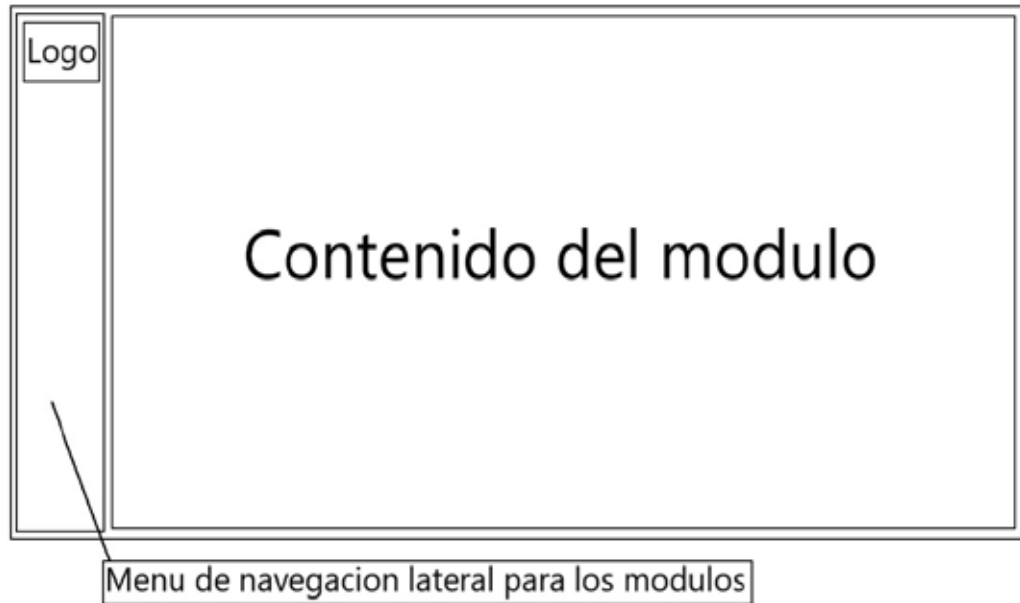
Gráfico 6. Esquema de diseño (Página de inicio).



Fuente: León, Sánchez (2020)

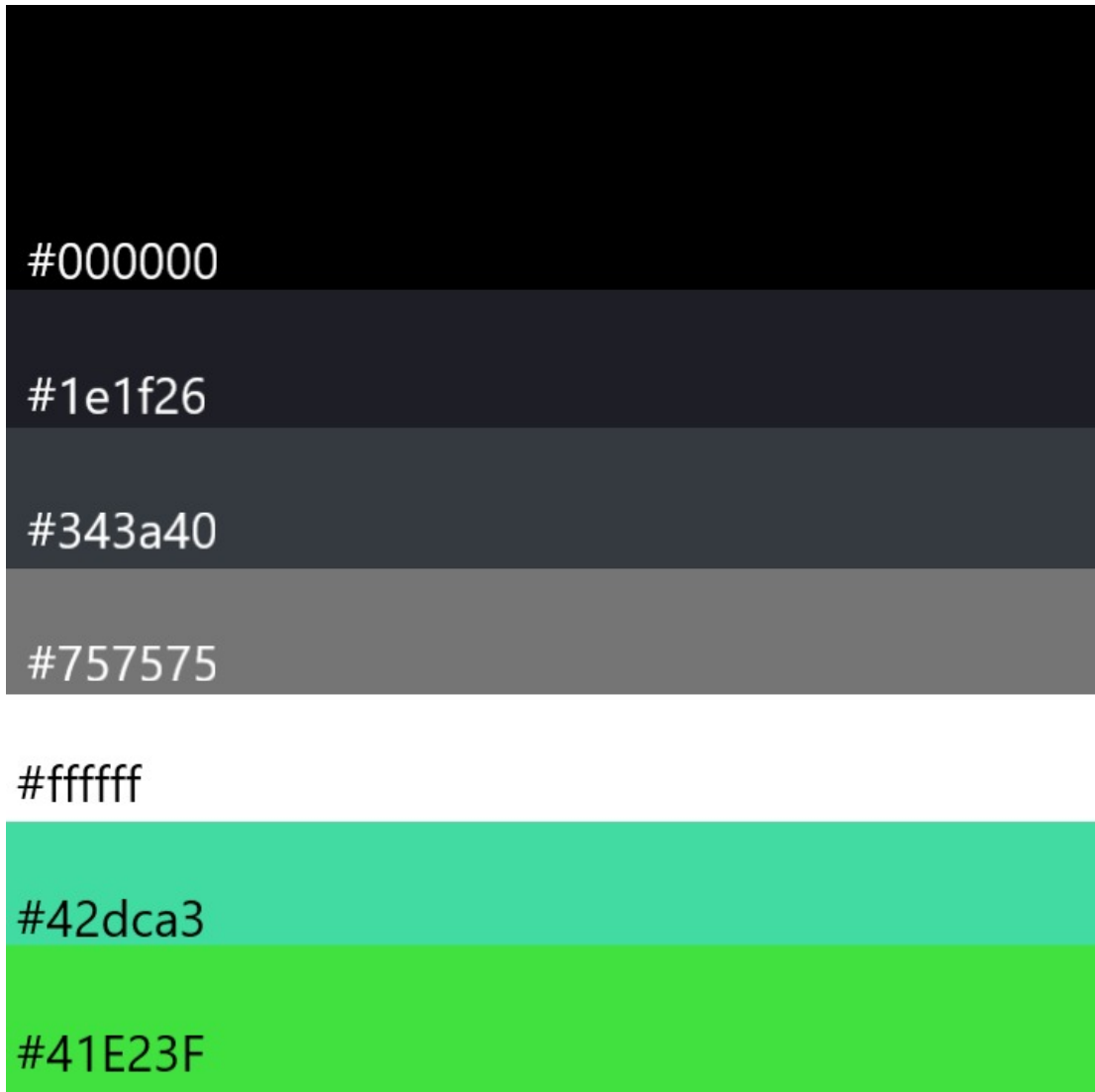
Estructura interna de básica de navegación en la plataforma.

Gráfico 7. Esquema de diseño (Estructura interna de básica de navegación en la plataforma)



Fuente: León, Sánchez (2020)

Gráfico 8. Esquema de diseño (Paleta de colores).



Fuente: León, Sánchez (2020)

Capturas de pantalla (De contener datos personales como los correos o los nombres, serán ocultados por la privacidad de los usuarios en las capturas):

Gráfico 9. Capturas de pantalla (Página de inicio).

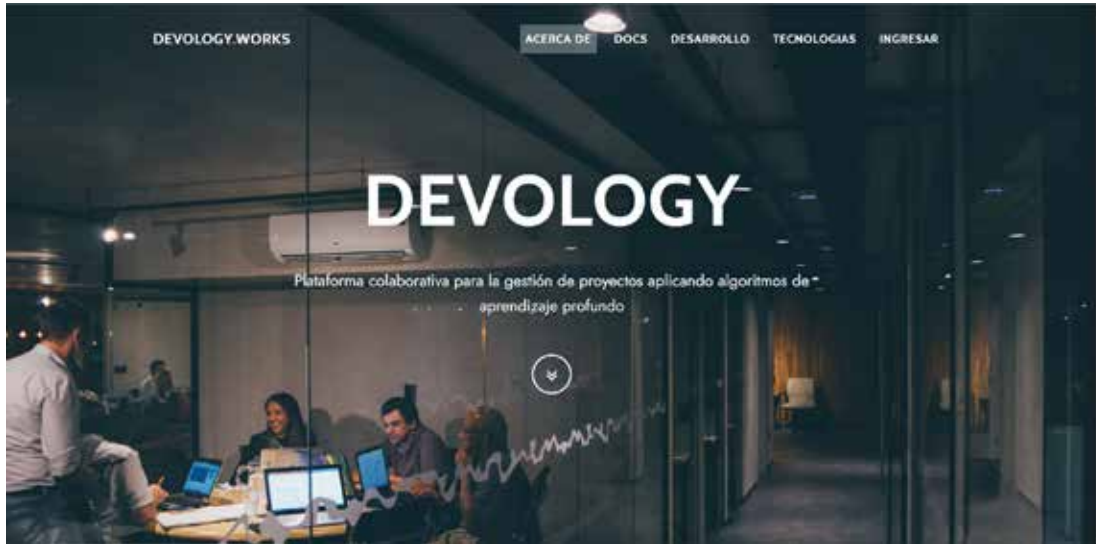
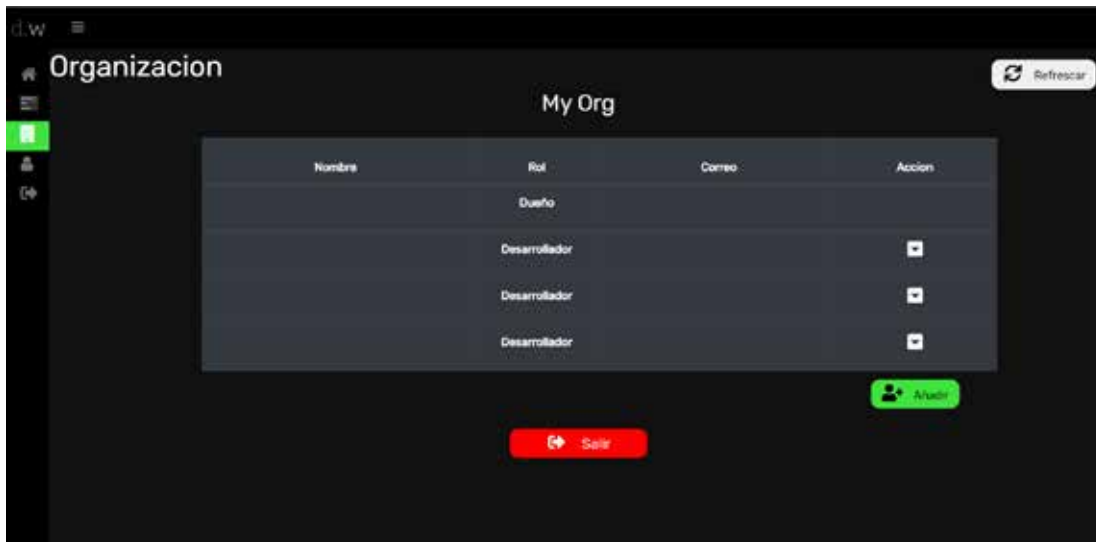


Gráfico 10. Capturas de pantalla (Página de organización).



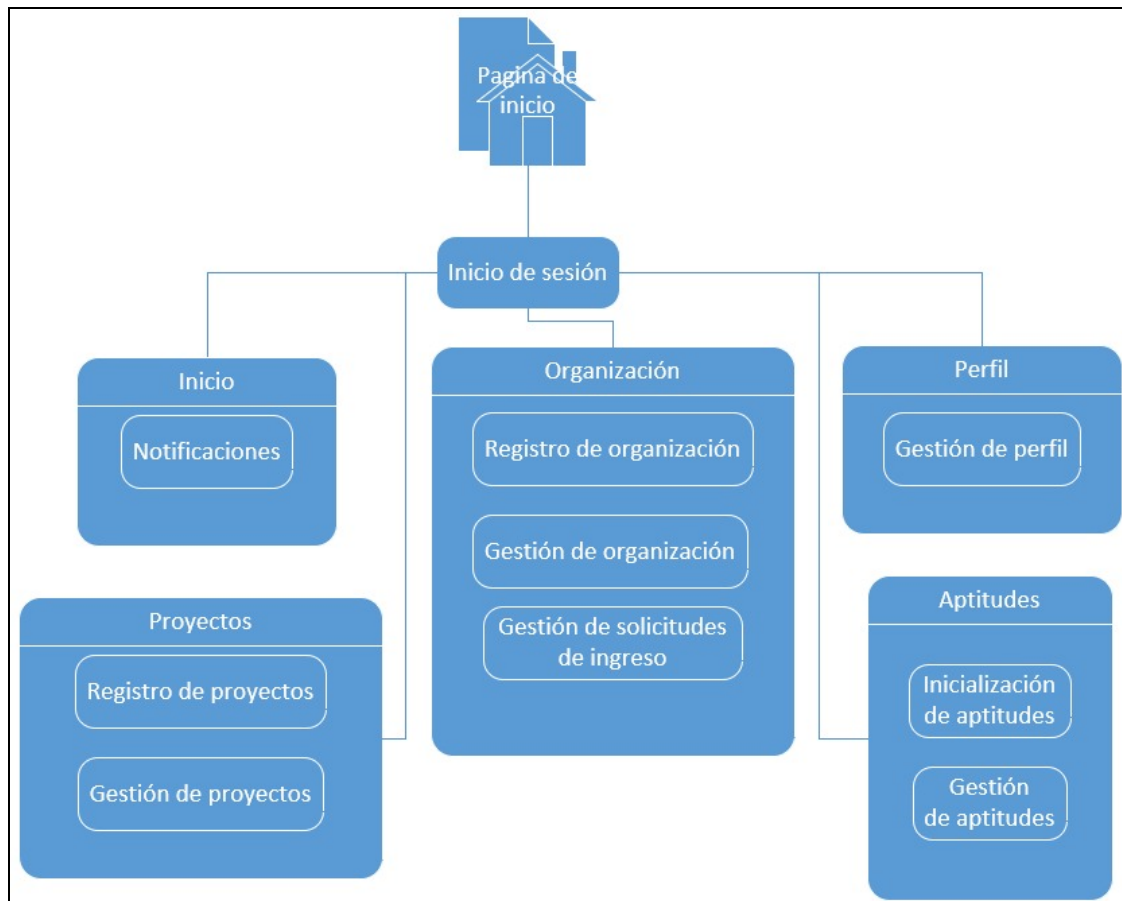
Fuente: León, Sánchez (2020)

4.3.8 Actividad VIII: Estructuración de las secciones de la plataforma.

Para esta actividad se plasmó de forma conceptual la forma en que se gestionarían las rutas por grupos, las direcciones que se deben seguir para tener acceso a estas y se agrupa en grupos módulos que estarán contenidos en cada sección, los cuales no estarán presentes dependiendo de la condición del usuario o de la organización,

ejemplo de esto sería no se podría acceder a la gestión de un proyecto si no se pertenece al mismo o no está creado.

Gráfico 11. Carta estructurada (Mapa de navegación).



Fuente: León, Sánchez (2020)

4.4 Fase IV: Desarrollo de la plataforma colaborativa para la gestión de proyectos de software usando herramientas computacionales.

Se realizó la codificación de la plataforma, esto se dividió en diferentes fases que comprender el desarrollo del servicio destinado al procesamiento y gestión de las solicitudes del usuario (BackEnd), por otra parte el desarrollo de la interfaz de usuario(FrontEnd) y por último, todo el proceso que conlleva el entrenamiento de las redes neuronales(Aplicada en forma de REST API externa), cabe destacar que estos procesos se realizaron de forma simultánea, dando únicamente prioridad a las

redes neuronales en las primeras, el mapa de desarrollo fue llevado a cabo de forma similar al mapa de navegación de la plataforma y el resto de secciones fueron desarrollados en base a los previos planteamientos conceptuales realizados en la fase 3.

4.5 Fase V: Ejecución de un plan de pruebas de software para la verificación de el correcto funcionamiento de la plataforma colaborativa.

Durante y después de la codificación del sistema, se realizaron y aplicaron pruebas en cada módulo, componente y servicio de la plataforma, todo esto con el objetivo de comprobar la buena funcionalidad y validar los resultados expresados en la plataforma. Se realizaron pruebas de caja blanca y caja negra.

Pruebas de Caja Negra:

Las pruebas de caja negra, es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software. En las pruebas de caja negra, nos enfocamos solamente en las entradas y salidas del sistema, sin preocuparnos en tener conocimiento de la estructura interna del programa de software.

Tabla 24, Inicio de sesión, nuevo usuario

CASO DE PRUEBA		
Número de prueba	Caso de Uso	Inicio de sesión, nuevo usuario
	Estrategia	Prueba de caja negra
1		
Descripción	El usuario desea registrarse en la plataforma	
Entradas	Nombre completo, correo, contraseña y tipo de usuario	
Resultado Esperado	El usuario se registrar exitosamente en la plataforma	
Resultado Esperado	El usuario registra exitosamente su compañía y lograr ingresar al sistema	

Resultado	Exitoso
Observación	El usuario no tuvo dificultades en registrarse y logro ingresar de manera exitosa al sistema.

Fuente: Leon, Sánchez (2020)

Tabla 25, Registro de la organización

CASO DE PRUEBA		
Número de prueba 2	Caso de Uso	Registro de la organización
	Estrategia	Prueba de caja negra
Descripción	El usuario desea registrar la organización en la cual desempeña sus labores.	
Entradas	Nombre de la organización.	
Resultado Esperado	El usuario registra exitosamente su organización y tiene acceso a las funciones de gestión de la misma	
Resultado	Exitoso	
Observación	El usuario pudo registrar con éxito la organización y posteriormente gestionar sus integrantes, todo esto con gran facilidad	

Fuente: Leon, Sánchez (2020)

Tabla 26, Creación/Inicialización de un proyecto

CASO DE PRUEBA		
Número de prueba 3	Caso de Uso	Creación/Inicialización de un proyecto
	Estrategia	Prueba de caja negra
Descripción	El usuario desea iniciar un proyecto, luego de que se realizara una recolección de requerimientos	
Entradas	Nombre del proyecto, descripción, integrantes y	

	requerimientos
Resultado Esperado	El usuario registra exitosamente el proyecto en cuestión
Resultado	Exitoso
Observación	El usuario expreso incomodidad visual en el método de presentación para escoger a los integrantes
Solución	Se realizaron ajustes y se pasó de un modelo de selección de múltiples modales en formas de cartas a un modal que integra una tabla de selección múltiple.

Fuente: Leon, Sánchez (2020)

Tabla 27, Registro de aptitudes de un desarrollador

CASO DE PRUEBA		
Número de prueba	Caso de Uso	Registro de aptitudes de un desarrollador
4	Estrategia	Prueba de caja negra
Descripción	El usuario tratara de registrar sus aptitudes con el formulario que se le presenta.	
Entradas	Selección de porcentaje de conocimiento en la aptitud reflejada en el formulario	
Resultado Esperado	El usuario carga exitosamente sus aptitudes y tiene acceso a otras funciones del sistema	
Resultado	Fallido	
Observación	El usuario tuvo dificultades al registrar sus aptitudes, expresando que la disposición del formulario era excesivamente grande y resultaba tedioso el llenado del mismo	
Solución	Editar el formato de llenado del formulario e implementar	

	un diseño más amigable con el usuario
--	---------------------------------------

Fuente: Leon, Sánchez (2020)

Pruebas de Caja Blanca:

En estas se pretende investigar sobre la estructura interna del código, exceptuando detalles referidos a datos de entrada o salida, para probar la lógica del programa desde el punto de vista algorítmico. Realizan un seguimiento del código fuente según se va ejecutando los casos de prueba, determinándose de manera concreta las instrucciones, bloques, etc. que han sido ejecutados por los casos de prueba.

Tabla 28, Creación/Inicialización de un proyecto (Método automático)

CASO DE PRUEBA		
Número de prueba	Caso de Uso	Creación/Inicialización de un proyecto (Método automático)
5	Estrategia	Prueba de caja blanca
Descripción	El usuario inicializa un proyecto definiendo los valores básicos que se solicitan y procede a la selección de integrantes con el método automático.	
Entradas	Nombre del proyecto, descripción del proyecto y requerimientos del proyecto	
Resultado Esperado	Se muestran los integrantes con aptitudes adecuadas	
Resultado	Fallido	
Observación	Algunos de los candidatos a integrantes recomendados no coincidían con los resultados esperados	
Solución	Se realizo un cambio en las funciones utilizadas para el entrenamiento de la red neuronal profunda y así poder	

	obtener una mayor precisión en las predicciones
--	---

Fuente: Chacón, Reyes (2019)

Tabla 29, Privacidad de información

CASO DE PRUEBA		
Número de prueba 6	Caso de Uso	Privacidad de datos
	Estrategia	Prueba de caja blanca
Descripción	El usuario ingresado al sistema trata de acceder a la información de otra organización editando datos en cache o realizando peticiones al servidor con herramientas externas	
Entradas	Identificador del usuario, identificador de la organización, entre otras cosas	
Resultado Esperado	El usuario no logra de ninguna forma acceder o ver la información de otra compañía.	
Resultado	Exitoso	
Observación	Las rutas del servidor están protegidas por el uso de sesión y validaciones adicionales en base a los datos recibidos, por lo tanto, las consultas son rechazadas	

Fuente: Leon, Sánchez (2020)

Tabla 30, Redireccionamiento por falta de datos de sesión

CASO DE PRUEBA		
Número de prueba 7	Caso de Uso	Redireccionamiento por falta de datos de sesión
	Estrategia	Prueba de caja blanca
Descripción	El usuario trata de acceder a un módulo del sistema mediante la ruta sin haber iniciado sesión.	

Entradas	Ruta de la sección de organización.
Resultado Esperado	El usuario no accede al módulo debido a que debe iniciar sesión primero y se le presentan mensajes de error para indicar esto
Resultado	Exitoso
Observación	El usuario no logra acceder al sistema hasta haber iniciado sesión

Fuente: Leon, Sánchez (2020)

Tabla 31, Guardado de imagen de perfil

CASO DE PRUEBA		
Número de prueba	Caso de Uso	Guardado de imagen de perfil
	Estrategia	Prueba de caja blanca
8		
Descripción	El usuario accede a la sección de perfil y trata de cargar su imagen de perfil	
Entradas	Imagen mediante un formulario	
Resultado Esperado	Se guardan exitosamente la imagen y se puede apreciar en la sección de perfil	
Resultado	Fallido	
Observación	El usuario cargo un archivo de texto, aunque fue guardado con éxito no puede ser representado en la interfaz grafica	
Solución	Se implementaron métodos de validación para la recepción de archivos y, por lo tanto, se limitó a solo formatos comunes de imágenes	

Fuente: Leon, Sánchez (2020)

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Al examinar los resultados obtenidos en cada una de las fases descritas previamente, conjuntamente con el desarrollo y despliegue de un sistema, debidamente realizadas las pruebas del mismo, los investigadores llegaron a una serie de desenlaces relacionados a los objetivos definidos en la presente investigación.

En la fase inicial o primera fase referente al diagnóstico de las plataformas de desarrollo colaborativo de trabajo o similares al sistema desarrollado, aplicando técnicas de recolección de datos, los investigadores lograron realizar una indagación de los sistemas desarrollados para tal fin, con ayuda de un conjunto de preguntas elaboradas aplicando el método de recolección de datos entrevista no estructurada, donde se logró examinar el conjunto de herramientas existentes destinadas para tal fin y conceptualizar el procedimiento general que es necesario para los procesos de gestión de proyectos.

En la segunda fase, siguiendo los lineamientos establecidos en el presente documento, se logró descubrir haciendo uso del recurso de recolección de datos antes mencionado. Los requerimientos que debe presentar la plataforma desarrollada, derivando en los requerimientos funcionales del sistema tales como la creación de proyectos, estimación de rendimiento de desarrolladores basado en sus características y simulación de duración de un determinado proyecto aplicando algoritmos de aprendizaje profundo y automático, es decir el uso de redes neuronales artificiales para el cálculo y estimación de las métricas requeridas, módulo de notificaciones para establecer un correcto sistema organizativo dentro de la plataforma, creación de organizaciones, gestión de solicitudes de proyectos, asignación de recursos disponibles, seguimiento de los mismos y la posibilidad de ser colaborativa.

Con respecto a los requerimientos no funcionales, se logró establecer un conjunto de directrices que son fundamentales en el presente sistemas tal y como la seguridad proporcionada en el mismo, correspondiente a rendimiento de la privacidad de los diferentes usuarios y roles actuales en sistema presentado. Además de esto la creación de interfaces fáciles e intuitivas para la correcta utilización de la herramienta.

En la tercera fase, se logró establecer los modelos de gestión de proyectos para la plataforma colaborativa mediante la metodología XP, en los cuales se elaboraran todos los diagramas y tablas requeridas por las bases fundamentales en la ingeniería de software entre los cuales destacan el diagrama de base de datos, la representación y especificación de los casos de usos por roles de usuario y la arquitectura del sistema; por otra parte, se plantearon y maquetaron las bases del diseño que se implemento en la plataforma realizando cartas de diseño y la paleta de colores para mantener la uniformidad a lo largo de toda la plataforma, cabe destacar que para estos se siguieron hitos de diseño actuales entre los cuales el mas destacado son tonalidades oscuros.

Para la ejecución de la **cuarta fase** descrita en el presente documento, se desarrolló un plataforma colaborativa para la gestión de proyectos usando herramientas computacionales que abarcan desde el Frontend de la plataforma con el Framework VueJS o el Backend del mismo el cual fue desarrollado implementando el marco de aplicación web Expressjs el cual corre bajo el entorno de ejecución de JavaScript NodeJS, cabe destacar que se realiza la base de datos mediante un entorno no relacional (NoSQL) tal como es MongoDB, también para el desarrollo de la API que implementaba las redes neuronales profundas se usaron Tensorflow que es una librería desarrollada por Google la cual implementa los métodos necesarios para el entrenamiento de las redes neuronales, Keras una librería que se ejecuta sobre Tensorflow para agilizar el proceso de aprendizaje y Flask como el marco de desarrollo web de Python para el manejo de las solicitudes y funciones de la API.

En la última fase, de acuerdo a lo puntualizado en el trabajo de grado, se ejecutó un plan de pruebas para verificar el correcto funcionamiento de todos y cada uno de los módulos que integran dicha plataforma.

5.2 Recomendaciones

En concordancia con la información obtenida durante el desarrollo de las fases, para futuras aplicaciones del sistema propuesto, se recomienda aplicar las recomendaciones expuestas a continuación.

En vista de que los sistemas computacionales se encuentran en una constante evolución y cambio, naturalmente los requerimientos funcionales y no funcionales variaran, lo que significa que en el caso de uso del sistema pueden realizarse actualizaciones y/o modificaciones en los diferentes módulos que componen al sistema.

El reentrenamiento de las redes neuronales es una opción para mejorar los resultados de las futuras predicciones, teniendo en cuenta los nuevos datos y la consideración fundamental de no realizar sobre-entrenamiento en las mismas, para obtener resultados deficientes.

La adición de nuevas habilidades, no necesariamente enfocadas a lenguajes de programación, sino también a la experiencia y conocimiento en otro software y/o habilidades requeridas en la gestión de proyectos. Este último punto para la elaboración de sistemas con procesos más generales y con un mayor alcance en proyectos de diferentes áreas de tecnología, ciencias, ingeniería y afines

ANEXOS

Certificación de la entrevista

- **Modelo:**

1. De acuerdo a su opinión profesional, ¿cuáles considera que deban ser los criterios para la distribución y asignación de desarrolladores a proyectos? hacen la distribución de los proyectos en los desarrolladores? ¿Bajo qué criterios lo hacen?
2. ¿Como determina usted el tiempo que requiere para efectuar un proyecto?
3. Enumere las causas que se han presentado por las cuales no se han podido concluir los proyectos en el tiempo planeado.
4. En su experiencia personal y profesional, ¿qué criterios utilizaría usted para evaluar la calidad de un código?
5. ¿Cuáles considera deberían ser los principales factores para evaluar la calidad de un desarrollador?
6. En relación a los proyectos que no son entregados en el tiempo planeado. Este número, ¿es menor al deseado, dentro de los límites deseados o mayor al deseado?
7. En su entorno laboral, ¿aplican metodologías de desarrollo de software y que aspectos piensa que son los más importantes de estas?

8. Ante una eventual falta de organización y/o comunicación (situaciones extraordinarias), ¿cree usted que las metodologías de desarrollo de software se adaptan y ayudan a remediar estas?

9. En relación a situaciones extraordinarias que presenten un cambio en la situación del proyecto (Como el retiro de una persona, algún problema que le impida trabajar o similares) ¿Qué curso de acciones tomaría usted para reevaluar el proyecto en curso?

10. ¿Qué funciones cree que son necesarios en una plataforma para la gestión de proyectos de software para ser útil al nivel de las empresas?

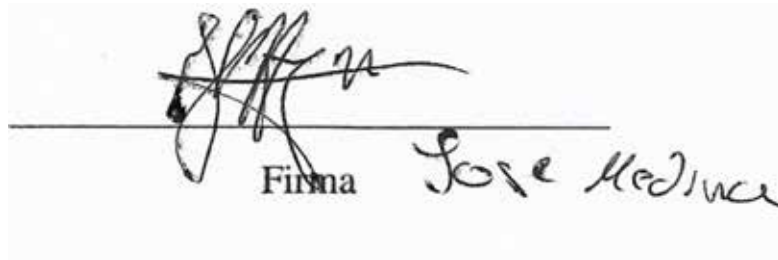
11. ¿Qué herramientas externas (GitHub, SonarQube, etc.) opina usted que ¿son necesarias de integrar en este tipo de plataformas y por qué?

- Firmas de certificación de expertos



Firma
Victor Hernandez

Ing. Victor Hernandez
Service Delivery Manager
B-YOND, INC.



Firma Jose Medina

Ing. Jose Francisco Medina
Product Manager of A&S
HUAWEI TECHNOLOGIES CIA. LTDA.



Firma

Ing. Pedro Emilio Flores Fusco
Full-Stack Web Developer
ARM HOLDINGS.

REFERENCIAS BIBLIOGRÁFICAS

Electrónicas:

- Alegsa, Leandro (2018). **Definición de Multiplataforma (informática)**. Recuperado de: <http://www.alegsa.com.ar/Dic/multiplataforma.php>.
- Álvarez, Sara (2007). **Sistemas gestores de bases de datos**. Recuperado de: <https://desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
- Amejide, Laura (2016). **Gestión de proyectos según el PMI**. Recuperado de: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/45590/7/lameijideTFC0116memoria.pdf>.
- Araque, María (2015). **Gestión de riesgos en proyecto de software a desarrollar en empresa privada**. (Trabajo de grado). Recuperado de: <https://pdfs.semanticscholar.org/8e28/b0b7e21c69dd77e956e7be7ebd09fd2b7f43.pdf>.
- Argomedo, Richard y Ochoa, Fernando (2017). **Implementación de un Sistema de Información Web para mejorar la gestión de proyectos siguiendo los lineamientos de la Guía de los Fundamentos para la Dirección de Proyectos (Guía del PMBOK®) para la empresa SIGCOMT S.A.C.** (Tesis para optar por el título profesional de Ingeniero de Sistemas). Recuperado de: <http://dspace.unitru.edu.pe/handle/UNITRU/10035>.
- Arias (2006). **El proyecto de investigación**. Recuperado de: https://www.academia.edu/9153815/Fidias_G._Arias_El_Proyecto_de_Investigaci%C3%B3n_5ta._Edici%C3%B3n.
- Arias (2012). **El proyecto de investigación**. Recuperado de: <https://evidencia.com/wp-content/uploads/2014/12/EL-PROYECTO-DE-INVESTIGACI%C3%93N-6ta-Ed.-FIDIAS-G.-ARIAS.pdf>.
- Bavaresco (2006). **Bases teóricas**. Recuperado de: <http://trabajodegradobarinas.blogspot.com/2011/11/bases-teoricas.html>.

- Blanco Jesús (2018). **Módulo de Gestión de Ordenes de Servicio y Actividades para el Sistema Web de la empresa CrediNat C.A.**”. (Trabajo especial de grado). Recuperado de: <https://es.slideshare.net/JesusJimenez123/tesis-desarrollo-de-aplicacion-web-tipo-help-integrado-en-sistema-web-existente>
- Calvo, D. (2018). **Metodología XP Programación Extrema (Metodología ágil)**. Recuperado de: <http://www.diegocalvo.es/metodologia-xp-programacion-extrema-metodologia-ágil/>.
- Carranza, Leidy (2016). **Gestión en proyectos de software**. (Artículo de investigación) Recuperado de: <https://revistas.udistrital.edu.co/ojs/index.php/tia/article/download/7609/pdf>.
- Constitución de la República Bolivariana de Venezuela (1999). **Constitución de la República Bolivariana de Venezuela**. Recuperado de: http://www.uc.edu.ve/archivos/pdf_pers_adm_obr/constitucion.pdf.
- Esthory, Eduardo (2019). **Sistema de redes neuronales para la evaluación de programas de postgrado**. (Trabajo presentado ante el Área de estudios de Postgrado de la Universidad de Carabobo para optar al Título de Magíster en Matemática y Computación). Recuperado de: <http://mriuc.bc.uc.edu.ve/bitstream/handle/123456789/8216/esthory.pdf?sequence=1>
- Izquierdo, J. (2014). **¿Qué es el XP Programming?**. Recuperado de: <https://www.iebschool.com/blog/que-es-el-xp-programming-agile-scrum/>.
- Kaplan, Andreas y Haenlein, Michael (2019). **Siri, Siri, in my hand: Who’s the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence**. Recuperado de: <https://web.archive.org/web/20181121191205/https://www.sciencedirect.com/science/article/pii/S0007681318301393>.
- Liu, Alex (2015). **Data Science and Data Scientist**. Recuperado de: <http://www.researchmethods.org/DataScienceDataScientists.pdf>.

- Mijares, Héctor y García, Luis (2007). **Normas para la elaboración y presentación de los anteproyectos, proyectos y trabajos de grado**. Recuperado de: https://www.academia.edu/4070723/normas_de_trabajo_de_grado.
- Nyce, Charles (2007). **Predictive Analytics White Paper**. Recuperado de: <https://web.archive.org/web/20130319130138/http://www.aicpcu.org/doc/predictivemodelingwhitepaper.pdf>.
- Parella y Stracruzzi (2017). **Bases legales**. Recuperado de: <https://metinvest.jimdofree.com/marco-te%C3%B3rico/>.
- Pérez, A. (2012) **Proyectos especiales**. Recuperado de: <http://antonioperezvillegas.blogspot.com/2012/04/proyectos-especiales.html>.
- Rojas, Andrea; Pérez, Anna y Echeverría Francklin (2019). **predicción del gasto público en los municipios de la región andina de Venezuela empleando redes neuronales artificiales**. (Artículo de investigación). Recuperado de: <https://search.proquest.com/openview/9eec2424343f6bd662a8b06b2b9c8555/1?pq-origsite=gscholar&cbl=1006393>.
- Russell, Stuart y Norvig, Peter (2009). **Artificial Intelligence A Modern Approach Third Edition**. Recuperado de: <https://faculty.psau.edu.sa/filedownload/doc-7-pdf-a154ffbec538a4161a406abf62f5b76-original.pdf>.
- Rouse, Margaret (2017). **Aprendizaje profundo (deep learning)**. Recuperado de: <https://searchdatacenter.techtarget.com/es/definicion/Aprendizaje-profundo-deep-learning>.
- Rouse, Margaret (2017). **Plataforma de colaboración**. Recuperado de: <https://searchdatacenter.techtarget.com/es/definicion/Plataforma-de-colaboracion>.
- Simeone (2018). **A Very Brief Introduction to Machine Learning With Applications to Communication Systems**. Recuperado de: <https://arxiv.org/pdf/1808.02342.pdf>.
- Torrealba y Rodríguez (2009). **La recopilación documental como técnica de investigación** Recuperado de: <http://dani14238551.blogspot.com/2009/03/la-recopilacion-documental-como-tecnica.html>

Vargas, Ileana (2012). **La entrevista en la investigación cualitativa: nuevas tendencias y retos**. Recuperado de: http://biblioteca.icap.ac.cr/blivi/coleccion_unpan/bol_diciembre_2013_69/uned/2012/investigacion_cualitativa.pdf.