

Maestría en ingeniería del software

Trabajo Fin de Maestría

PROPUESTA CONCEPTUAL PARA LA GESTIÓN DE LA PRODUCCIÓN CIENTÍFICA EN LA UNIVERSIDAD TÉCNICA LUIS VARGAS TORRES DE ESMERALDAS SEDE SANTO DOMINGO DE LOS TSÁCHILAS

AUTOR: LUIS ALFONSO NÚÑEZ FREIRE

LA CONCORDIA, ECUADOR

PROFESOR: RODRIGO CADENA MARTÍNEZ

DICIEMBRE DE 2021

Resumen

Esta tesis pertenece a la línea de investigación de la Ingeniería de Requisitos, Auditoría e Ingeniería de Software definida por la universidad UNADE, está direccionada a elaborar una propuesta conceptual que ayudará posteriormente en el desarrollo de un software que facilite la gestión de las publicaciones científicas efectuadas en la Universidad Técnica Luis Vargas Torres de Esmeraldas Sede Santo Domingo de los Tsáchilas y con ello llevar un control de los estándares e indicadores para evidenciar los trabajos realizados por la función sustantiva de la investigación.

En la recolección de datos se aplicó la técnica de la entrevista, utilizando como guía para las preguntas el instrumento del cuestionario, el mismo que se evidencia en un documento digital, las respuestas recibidas del Responsable del departamento ponen de manifiesto que los mecanismos empleados en el Departamento de Investigación para el registro de los trabajos científicos han ocasionado que la información no se encuentre actualizada y que la presentación de reportes estadísticos sea compleja.

Para solucionar el problema detectado se hizo una propuesta conceptual aplicando una metodología ágil y el marco de trabajo *scrum*, donde se especificó los integrantes del scrum team y sus respectivas tareas, se empleó el modelado UML y modelo entidad relación.

Palabras claves: ingeniería de software, producción científica; scrum; UML.

Abstract

This thesis belongs to the research line of Requirements Engineering, Auditing and Software Engineering defined by the UNADE university, it is aimed at developing a conceptual proposal that will later help in the development of a software that facilitates the management of scientific publications made at the Universidad Técnica Luis Vargas Torres de Esmeraldas Sede Santo Domingo de los Tsáchilas, and thereby control the standards and indicators to demonstrate the work carried out by the substantive function of the investigation.

In the data collection, the interview technique was applied, using the questionnaire instrument as a guide for the questions, the same one that is evidenced in a digital document, the responses received from the Head of the department show that the mechanisms used in the Research Department for the registration of scientific works have caused that the information is not updated and that the presentation of statistical reports is complex.

To solve the detected problem, a conceptual proposal was made applying an agile methodology and the scrum framework, where the members of the scrum team and their respective tasks were specified, UML modeling and entity relationship model were used.

Keywords: Software engineering, scientific production; scrum; UML.

Dedicatoria

La presente tesis está dedicada a Dios Padre todo poderoso, quien guía mis pasos por los caminos de la vida, llenándolos de bendiciones y fuerzas para seguir adelante en la consecución de mis metas.

A mi familia, por ser ejemplo de lucha, buenos valores y trabajo constante, que inspiran mis acciones.

Agradecimientos

Agradezco a Dios, por permitirme disfrutar de las metas alcanzadas a nivel personal, académico y profesional.

A mi familia, que son el soporte para afrontar las diversas situaciones, apoyándome emocionalmente y con consejos.

A mis amigos y a las personas que me han apoyado de manera desinteresada durante el proceso de investigación de esta tesis.

A la universidad UNADE por ser patrocinadores del conocimiento adquirido en el transcurso de esta maestría.

CONTENIDOS

Resumen ii

Abstract iii

Dedicatoria iv

Agradecimientos v

Capítulo 1. Marco introductorio 1

1.1. Antecedentes 1

1.2. Problema de investigación 2

1.3. Justificación 3

1.4. Establecimiento de hipótesis y objetivos 3

1.4.1. Hipótesis 3

1.4.2. Objetivos 4

1.4.2.1. Objetivo general 4

1.4.2.2. Objetivos específicos 4

Capítulo 2. Marco teórico 5

2.1. Producción científica 5

2.1.1. Divulgación científica 5

2.1.2. Revistas científicas 5

2.1.3. Artículos científicos 6

2.1.4. Libro científico 6

2.1.5. Productividad científica 7

2.2. Ingeniería del software 7

2.2.1. Software 8

2.2.2. Sistema de información 8

2.2.3. Metodologías de desarrollo de software 8

2.2.3.1. Metodologías ágiles 9

2.2.4. Scrum 10

2.2.4.1. Roles 12

2.2.4.1.1. Product owner 12

2.2.4.1.2. Scrum master 12

2.2.4.1.3. Development team 13

2.2.4.2. Artefactos 14

2.2.4.2.1. Product backlog 14

2.2.4.2.2. Sprint backlog 14

2.2.4.2.3. Burndown charts 14

2.2.4.2.4. Definition of done 15

2.2.4.3. Actividades 15

2.2.4.3.1. Sprint planning 15

2.2.4.3.2. Sprint execution 16

2.2.4.3.3. Scrum daily meeting 16

2.2.4.3.4. Sprint review 17

2.2.4.3.5. Sprint retrospective 17

2.2.4.3.6. Product backlog refinement..... 17

2.2.5. Lenguaje unificado de modelado UML..... 17

2.2.5.1. Tipos de diagramas estructurales 19

2.2.5.1.1. Diagramas de clases 19

2.2.5.1.2. Diagrama de objetos..... 19

2.2.5.1.3. Diagrama de componentes 19

2.2.5.2. Tipos de diagramas de comportamiento..... 19

2.2.5.2.1. Diagrama de actividades 19

2.2.5.2.2. Diagrama de casos de uso 20

2.2.5.2.3. Diagrama de secuencia..... 20

2.2.6. Bases de datos 20

2.2.6.1. Modelo de datos 20

2.2.6.1.1. Modelo entidad relación..... 21

2.2.6.1.2. Diseño conceptual 21

2.2.6.1.3. Diseño físico..... 21

2.2.6.2. Elementos del modelo relacional 22

2.2.6.2.1. Entidades 22

2.2.6.2.2. Atributos..... 22

2.2.6.2.3. Tipo de dato..... 22

2.2.6.2.4. Clave primaria 22

2.2.6.2.5. Clave foránea..... 23

2.2.6.2.6. Cardinalidad 23

2.2.6.3. Sistema gestor de bases de datos 23

2.2.6.3.1. MariaDB 24

2.2.6.3.2. PostgreSQL 24

2.2.6.4. Lenguaje de consulta SQL 24

2.2.6.4.1. Lenguaje de definición de datos 25

2.2.6.4.2. Lenguaje de manipulación de datos 25

2.2.6.4.3. Lenguaje de control de datos 25

Capítulo 3. Marco empírico 26

3.1. Desarrollo de la investigación 26

3.1.1. Clasificación de la investigación 26

3.1.2. Diseño de la investigación 27

3.2. Metodología scrum 27

3.2.1. Designación de roles 27

3.2.1.1. Product owner 27

3.2.1.2. Scrum master 27

3.2.1.3. Develoment team 28

3.2.2. Análisis de requerimientos 28

3.2.2.1. Requerimientos de la app 28

3.2.3. Historias de usuarios 29

3.2.4. Generación del product backlog 31

3.2.5. Estimación de los sprints 31

3.2.6. Tareas de los sprints 32

3.2.6.1. Tareas del sprint 1 32

3.2.6.2. Tareas del sprint 2 34

3.2.6.3. Tareas del sprint 3 36

3.3. Diagramas UML 38

3.3.1. Diagrama de clases 38

3.3.2. Diagrama de casos de uso 39

3.3.2.1. Descripción de los casos de uso 40

3.3.3. Diagramas de secuencia 46

3.4. Diseño de la base de datos 58

3.4.1. Modelo entidad relación 58

3.4.2. Modelo conceptual 59

3.4.3. Modelo físico 60

Capítulo 4. Conclusiones y recomendaciones 61

4.1. Conclusiones 61

4.2. Recomendaciones 61

Referencias bibliográficas 63

Apéndices 69

ÍNDICE DE TABLAS Y FIGURAS

Tablas

Tabla 1.Requerimientos de la app..... 28

Tabla 2.Historias de usuarios..... 29

Tabla 3.Product backlog 31

Tabla 4.Estimación del sprint 1 31

Tabla 5.Estimación del sprint 2 31

Tabla 6.Estimación sprint 3 32

Tabla 7.Tareas sprint 1, historia de usuario creación de la base de datos. 32

Tabla 8. Tareas sprint 1, historia de usuario acceso al sistema. 32

Tabla 9.Tareas sprint 1, historia de usuario mantenimiento datos proyectos. 33

Tabla 10. Tareas sprint 1, historia de usuario mantenimiento datos carreras. 33

Tabla 11. Tareas sprint 1, historia de usuario mantenimiento datos docentes..... 34

Tabla 12.Tareas sprint 2, historia de usuario crear informe de publicaciones por carrera. 35

Tabla 13. Tareas sprint 2, historia de usuario crear informe de publicaciones por proyectos. 35

Tabla 14. Tareas sprint 2, historia de usuario crear informe de publicaciones por docente. 36

Tabla 15.Tareas sprint 3, historia de usuario mantenimiento de datos revistas..... 36

Tabla 16.Tareas sprint 3, historia de usuario mantenimiento de datos publicaciones.... 37

Tabla 17. Descripción caso de uso mantener datos proyectos 40

Tabla 18. Descripción caso de uso mantener datos carreras 41

Tabla 19.Descripción caso de uso mantener datos docentes 42

Tabla 20.Descripción caso de uso reporte publicaciones 43

Tabla 21.Descripción caso de uso mantener datos revistas 44

Tabla 22.Descripción caso de uso mantener datos publicaciones 45

Figuras

Figura 1: Revistas científicas 6

Figura 2: Ingeniería de software 7

Figura 3: Etapas del desarrollo ágil 10

Figura 4: Elementos de scrum 11

Figura 5: Diagramas UML 18

Figura 6: Diagrama entidad relación 21

Figura 7: Diagrama de clases 38

Figura 8: Diagrama de casos de uso 39

Figura 9: Diagrama de secuencia edición proyectos 46

Figura 10: Diagrama de secuencia edición carreras 47

Figura 11: Diagrama de secuencia edición docentes 48

Figura 12: Diagrama de secuencia reporte publicaciones 49

Figura 13: Diagrama de secuencia edición revistas 50

Figura 14: Diagrama de secuencia edición publicaciones 51

Figura 15: Diagrama de actividades edición proyectos 52

Figura 16: Diagrama de actividades edición carreras 53

Figura 17: Diagrama de actividades edición docentes 54

Figura 18: Diagrama de actividades reporte publicaciones 55

Figura 19: Diagrama de actividades edición revistas 56

Figura 20: Diagrama de actividades edición publicaciones 57

Figura 21: Modelo entidad relación 58

Figura 22: Modelo conceptual de la base de datos 59

Figura 23: Modelo físico de la base datos 60

Capítulo 1. Marco introductorio

1.1. Antecedentes

El intelectual prusiano Wilhelm Von Humboldt consideró que en la universidad la enseñanza y la investigación deberían formar una unidad. Razón por la cual la educación superior no puede concebirse sin la producción de conocimiento, es decir, sin el proceso de investigación que la sustenta (Avendaño, 2020).

Cada IES (Institución de Educación Superior) planifica las actividades investigativas según las líneas y sublíneas de investigación, objetivos y políticas las cuales son ejecutadas a través de programas o proyectos con el apoyo interno y externo. La divulgación y difusión de los resultados de estas investigaciones permitirá el uso del conocimiento por parte de la sociedad con la finalidad de generar nuevos productos, procesos o servicios (CACES, 2019). Entre los medios para publicar los trabajos científicos se encuentran los libros, capítulos de libros, artículos científicos en revistas indexadas, ponencias, etc.

El CACES (Consejo de Aseguramiento de la Calidad de la Educación Superior) realiza la evaluación de las IES enfocada en las funciones sustantivas de docencia, investigación y vinculación, conformados por 20 estándares. En el criterio de la Academia se encuentran: producción científica, investigación regional, libros y capítulos revisados por partes (Molina et al., 2020).

Emplear los sistemas de información para llevar un registro pormenorizado de todas las publicaciones efectuadas es fundamental, ya que esto permite medir los indicadores respectivos y con ello diseñar estrategias que permitan alcanzar los estándares solicitados por la entidad de control.

En una publicación desarrollada por Medina et al. (2019) en la revista MediSur hace referencia que con la finalidad de solventar ciertas deficiencias tales como: duplicidad de información, dificultad en el acceso oportuno y rápido, falta de un mecanismo que garantice la información almacenada de la producción científica crean un sistema para la gestión de dicha producción desarrollada en la Universidad de Ciencias Médicas de Cienfuegos.

En el desarrollo de una propuesta de *software* es fundamental seguir una metodología y un marco de trabajo que permita establecer claramente las etapas del ciclo de vida de éste y también los individuos participantes, Gaete et al. (2021) presentan en la revista chilena Ingeniare un enfoque de aplicación ágil con Scrum, Lean y Kanban, publicación que permite tener una visión de estos enfoques para los proyectos de *software.*

1.2. Problema de investigación

Para el control de la producción científica se considera los datos, por ejemplo, de un artículo científico: el nombre del artículo, autores, revista, ISSN, base de datos indexada, DOI, proyecto, área, fecha de publicación, entre otros. Estos datos permiten realizar los cálculos con respecto a los indicadores que se debe cumplir, con esto también se debe demostrar la veracidad de la información proporcionada a los evaluadores internos y externos.

Actualmente el responsable de Investigación realiza el registro de la producción científica desarrollada por los profesores y estudiantes de las distintas carreras de la Universidad Técnica Luis Vargas Torres de Esmeraldas Sede Santo Domingo de los Tsáchilas a través de hojas electrónicas, cada que se requiere actualizar la información

es enviado a través del correo institucional un archivo con el formato requerido para que sea llenado por los autores de las publicaciones, posteriormente se alimenta una matriz.

Esta forma de trabajar ocasiona muchas de las veces duplicidad de los datos e inconsistencia; para lograr una depuración se requiere de una revisión manual y pormenorizada la cual emplea un tiempo considerable.

1.3. Justificación

Esta tesis se encuentra enmarcada en la línea de investigación de Requisitos, Auditoría e Ingeniería de Software definida por la Universidad Americana de Europa.

Dado el progreso tecnológico y a la necesidad de dar respuesta pronta y veraz a las solicitudes y requerimientos por parte de la comunidad universitaria y los entes externos de control, es conveniente que el Departamento de Investigación de la Universidad Técnica Luis Vargas Torres de Esmeraldas Sede Santo Domingo de los Tsáchilas cuente con una propuesta para conceptualizar un sistema que permita gestionar el control de la producción científica llevada a cabo por los profesores y estudiantes.

1.4. Establecimiento de hipótesis y objetivos

1.4.1. Hipótesis

El diseño de la propuesta para la gestión de la producción científica en la Universidad Técnica Luis Vargas Torres de Esmeraldas Sede Santo Domingo de los Tsáchilas permitirá implementar de manera más ágil un software que automatice los procesos del departamento.

1.4.2. Objetivos

1.4.2.1.Objetivo general

Desarrollar una propuesta conceptual para la gestión de la producción científica en la Universidad Técnica Luis Vargas Torres de Esmeraldas Sede Santo Domingo de los Tsáchilas.

1.4.2.2.Objetivos específicos

- Estudiar y analizar el estado del arte de las tecnologías empleadas en la gestión y almacenamiento de los datos.
- Análisis de los procesos que intervienen en la gestión de la producción científica.
- Diseño conceptual de la solución para la gestión de la producción científica.

Capítulo 2. Marco teórico

2.1. Producción científica

La producción científica hace referencia, en términos generales, a la publicación o difusión de los trabajos científicos efectuados por los profesores, estudiantes o investigadores. Los resultados científicos tienen la condición de publicados o en proceso de arbitraje (Tarango et al., 2017).

2.1.1. Divulgación científica

El objetivo de la divulgación científica es permitir que el público pueda acceder a los conocimientos, y con ello comparar, cuestionar, interpretar e incluso reinterpretar los resultados generados en las investigaciones publicadas, propiciando de esta forma la creación de estudios y proyectos (Rivas-Torres, 2017). Existen varios formatos de divulgar los resultados fruto de las investigaciones, entre ellos el artículo científico, libros y capítulos de libros, congresos, posters, documentales, etc.

2.1.2. Revistas científicas

La revista científica es un medio que realiza publicaciones periódicas de manera impresa (ver figura 1), o de manera digital, especialmente de artículos científicos, con información de actualidad en varias áreas del conocimiento, cuya finalidad es la de contribuir en el progreso de la ciencia, cada entrega está numerada o fechada consecutivamente (Unesco, 2002, como se citó en Tarango et al., 2017). “Las revistas científicas muestran características peculiares: cada una posee su propia identidad definida a través de sus objetivos, delimitación temática, patrones aceptados de autoría, tipo y extensión del contenido y referencias” (Tarango et al., 2017, p. 68).

Figura 1
Revistas científicas



Nota. Tomado de Índice de revistas mexicanas de divulgación científica y tecnológica, por A. Sánchez, Azul Web (https://www.azulweb.net/indice-de-revistas-mexicanas-de-divulgacion-cientifica-y-tecnologica/). CC BY.

2.1.3. Artículos científicos

Un artículo es un documento original e inédito, que brinda avances novedosos y de suma importancia y relevancia científica que se escribe con la finalidad de que una revista especializada en un determinado campo de investigación lo publique y de esta manera los lectores puedan reproducir los experimentos o sistemas propuestos por el autor o los autores (Muñoz et al., 2018).

2.1.4. Libro científico

"El libro científico trata sobre una temática específica dentro de los campos científicos, tecnológicos, humanísticos, o de diferentes fenómenos de orden históricos psicológicos sociológicos". Su contenido describe, narra o explica en profundidad sobre una temática de la ciencia, además debe ser original, ya sea a partir de materiales

de referencia o de su análisis (Corona-Lisboa, 2015; Ejarque, 2005, como se citó en Tarango et al., 2017, p. 71).

2.1.5. Productividad científica

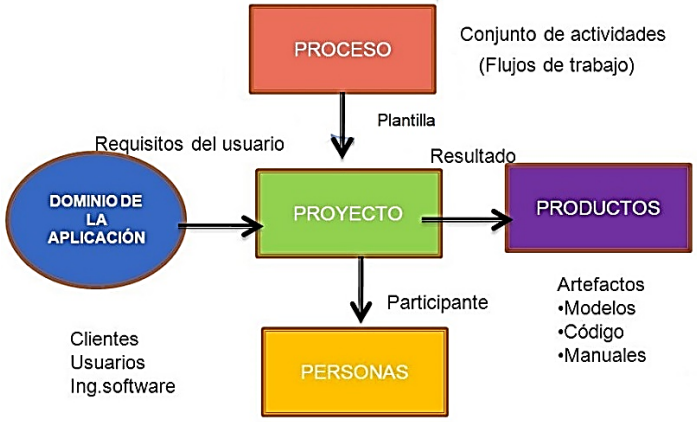
De Filippo y Fernández mencionan que la productividad científica es un indicador que expresa el número de publicaciones desarrolladas por cada investigador, que permite medir el desempeño del sistema científico en la producción de conocimiento (como se citó en Barra, 2019).

2.2. Ingeniería del software

La ingeniería de software es una disciplina que mediante la aplicación de los principios de la informática y las matemáticas busca ejecutar soluciones que optimicen relación entre coste y beneficio a los problemas de *software*. También involucra la aplicación de metodología, herramientas y tecnologías en el análisis, desarrollo, operación y mantenimiento de *software* (Noriega, 2017). La figura 2 muestra los factores de la ingeniería de *software* en el desarrollo de proyecto de *software*.

Figura 2

Ingeniería de software



Nota. Tomado de Proceso y modelos en la ingeniería de software, por S. Godoy, 2015, Slideplayer (https://slideplayer.es/slide/159895/). CC BY 2.0

2.2.1. Software

El *software* es considerado como un elemento intangible y el soporte lógico que gobierna el funcionamiento de un sistema informático, este soporte está formado por un conjunto de programas y rutinas que hacen factible la ejecución de tareas específicas que puede incluir elementos como: *drivers*, documentos, bases de datos, compiladores, interpretes, entre otras (Gómez & Moraleda, 2020).

2.2.2. Sistema de información

Los sistemas de información son herramientas de gran utilidad para las organizaciones, sean públicas o privadas, ya que contribuyen en el procesamiento de los datos permitiéndoles de esta forma generar información para ser utilizadas en la toma de decisiones en sus distintas operaciones y actividades, técnicamente los sistemas de información son un conjunto de elementos tales como: *hardware*, *software*, humanos y económicos, que están relacionados entre sí con el objetivo de capturar, procesar, guardar y distribuir el activo más importante que poseen las empresas, es decir, los datos (Vargas et al., 2019).

2.2.3. Metodologías de desarrollo de software

Las metodologías de desarrollo de software son de gran importancia para lograr poner en funcionamiento cualquier sistema informático debido a que presentan un conglomerado de técnicas y métodos que se emplean para diseñarlos, Cendejas argumentó que son “un enfoque estructurado para el desarrollo de software que incluye modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos” (como lo citó Zumba, 2018, p. 25).

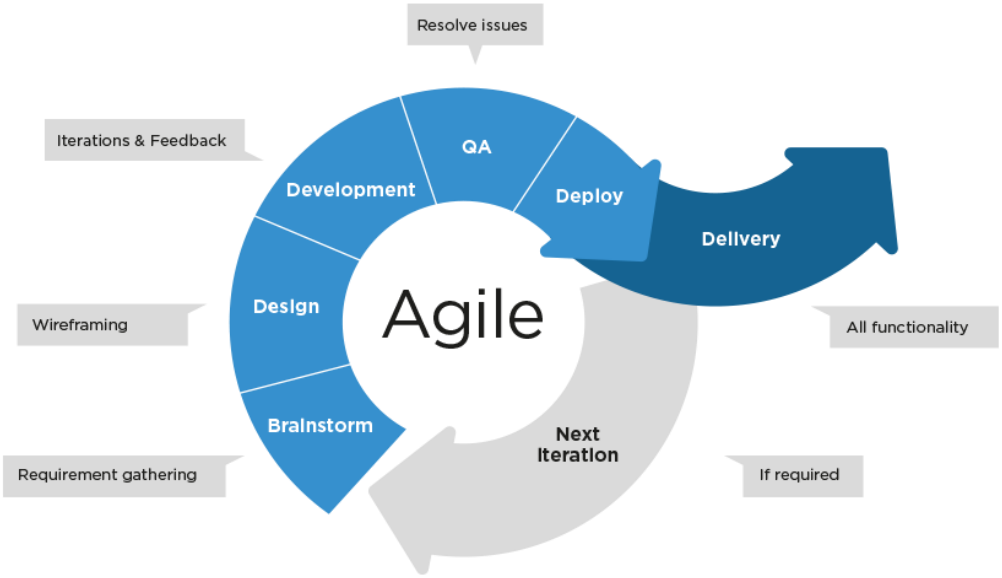
2.2.3.1.Metodologías ágiles

> Cadavid et al. (2013) mencionan que las metodologías ágiles particionan el proyecto que se está ejecutando haciéndolos más flexibles, son sumamente colaborativos, adaptables a los cambios y participativos dado que la interacción con el usuario o cliente es continua al momento de realizar la entrega de las partes y su revisión correspondiente. (como se citó en Molina-Montero et al., 2018, p. 116)

En la página de Red Hat (s.f.) se menciona que las metodologías ágiles de desarrollo de software son una guía cuya intención es dividir en fragmentos funcionales el sistema de software y que se ajusten a las necesidades del cliente, para conseguir este objetivo se usan enfoques flexibles que permitan que el trabajo en equipo evolucione en continuas mejoras del producto, cada fragmento estará a cargo de pequeños equipos de desarrollo, los mismos que se reúnen periódicamente con un encargado por parte de la empresa durante el ciclo de vida del desarrollo de software. Esto permite que los procesos de desarrollo y prueba sean constantes y se lleven en paralelos. La metodología presenta cuatro características prioritarias fundamentales tales como: personas e interacciones antes que procesos y herramientas, *software* funcional sobre la documentación, trabajo colaborativo con el cliente sobre aspectos concernientes al contrato y planes flexibles para responder al cambio. La figura 3 muestra las diferentes etapas planificadas en una iteración, las mismas que se deben llevarse a cabo para conseguir un sistema completamente funcional, en el caso de que no exista la funcionalidad requerida se debe realizar otra iteración.

Figura 3

Etapas del desarrollo ágil



Nota. Tomado de Gestión ágil de proyectos, desarrollo ágil de software y gestión de proyectos, Pngwing (https://www.pngwing.com/es/free-png-hrgcm). BY NC

2.2.4. Scrum

Este marco de trabajo forma parte de las metodologías ágiles y es utilizado para guiar actividades de desarrollo como el de requerimientos, análisis, diseño, evolución y entrega. El tratamiento de cada una de estas actividades es efectuado dentro de un *sprint*, la cantidad de *sprint* dependerá del alcance y la complejidad del producto que se está desarrollando. Scrum es flexible, razón por la cual el trabajo ejecutado se adapta al problema en cuestión y no es de extrañarse que se modifique con frecuencia en tiempo real según el criterio técnico del equipo scrum (Pressman, 2010).

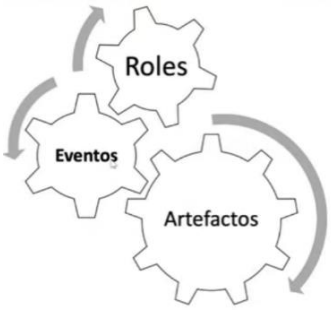
Scrum se acopla de forma eficaz a proyectos cuyos plazos para ser entregados son apremiados, también con los proyectos en los que los requerimientos cambian, esto se debe al uso de patrones de proceso del *software*. Scrum proporciona un marco de trabajo que evita las tergiversaciones, reduciendo los errores, ya que permite aplicar

buenas prácticas en la construcción de un proyecto. En las iteraciones cada uno de los miembros que integran los equipos de trabajo tienen pleno conocimiento de las actividades, los pasos y los plazos que deben cumplir para concretar con éxito los objetivos propuestos (Gaete et al., 2021).Monte (2016) realiza un acercamiento a los roles, artefactos y actividades también conocidas como eventos, siendo estos tres elementos esenciales del scrum, los mismos que deben funcionar como un engranaje para potenciar los resultados del proyecto (ver figura 4):

Los roles establecen la distribución de las responsabilidades en un proyecto. Es así que se tiene el *scrum team* cuya tarea es el desarrollo del producto, el *product owner* en representación del negocio y el *scrum master* llevan el gobierno del proyecto, como conocedores del negocio y sus necesidades se encuentran los *stakeholders*. Los artefactos son las herramientas con las cuales el equipo de trabajo ejecuta y controla el proyecto, apoyado en listas de control y en el *scrum board*, y las actividades constituyen el tiempo requerido para el desarrollo del proyecto. Para brindar siempre una visión del producto que se está construyendo el tiempo es dividido en ciclos autoorganizables y evaluables.

Figura 4

Elementos de scrum



Nota. Tomado de Introducción a scrum, por A. Betancourt, 2021, Awakelab (https://www.awakelab.cl/las-claves-de-scrum/). BY NC

2.2.4.1.Roles

2.2.4.1.1. Product owner

De acuerdo con Rojas (2021), el *product owner* es la voz oficial del cliente, tiene como responsabilidad establecer el enlace entre el cliente y el equipo de trabajo, él conoce y entiende los requerimientos del negocio y también el de los usuarios. Entre las funciones que debe cumplir están:

- Establecer el orden de las tareas para cumplir con las metas y los objetivos.
- Llevar el control de la lista de tareas enfocadas en el desarrollo del nuevo producto.
- Emplear estrategias para optimizar el trabajo ejecutado por el equipo.
- Lograr que los demás integrantes del equipo comprendan las tareas a efectuarse.
- Ser el único nexo entre las autoridades involucradas al desarrollo del producto.

2.2.4.1.2. Scrum master

Con respecto a *scrum master,* Canal (2015) menciona que es el encargado de liderar los equipos en la gestión eficaz de proyectos para que puedan sobre llevar cualquier dificultad que se presente en el camino hasta llegar al sprint final y con ello alcanzar los objetivos propuestos.

Un *scrum master* debe realizar:

- Asesorar a cada uno de los integrantes del equipo para que puedan autoorganizarse y trabajar mancomunadamente.
- Eliminar los impedimentos en el equipo para obtener sus objetivos finales.
- Explicará a cada desarrollador los pormenores de las tareas asignadas y aprendan a utilizar la metodología adecuada.

- Actúan como guía y moderador cuando los integrantes del equipo presentan a los *stakeholders* el sprint final.
- En los *daily standups* se encarga de revisar el progreso y realizar actualizaciones que el equipo requiere para el desarrollo del proyecto.
- Trabaja estrechamente con el producto owner definiendo desde un principio los objetivos del proyecto.

2.2.4.1.3. Development team

ProyectosAgiles (2021) indica que el equipo de desarrollo es un conjunto de personas técnicas que de manera conjunta desarrollan el producto del proyecto. Este equipo tiene un objetivo común, son responsables de realizar el trabajo en cada iteración y por consiguiente en el proyecto.

Para que este equipo llegue a un alto rendimiento debe cumplir las siguientes características:

- Cada integrante del equipo posee las habilidades requeridas para identificar y ejecutar las tareas que permitan suministrar al cliente los requisitos definidos para la iteración.
- La dependencia de personas que no pertenecen al equipo debe ser mínima con el fin de evitar inconvenientes el cumplimiento de lo acordado para cada iteración.
- Complementar las experiencias, conocimientos y habilidades de los miembros permiten ideas creativas.
- El tamaño del equipo debe estar comprendido entre 5 y 9 personas, menos de 5 al presentarse alguna complicación puede perjudicar los objetivos previsto para

la iteración, más de 9 se corre el riego de que se formen subgrupos dificultando la comunicación.

- Evitar en lo mínimo las rotaciones de los integrantes de los grupos de trabajo, es recomendable que los miembros sean asignados a tiempo completo a los proyectos.
- Debe interactuar con los *stakeholders,* quienes emitirán el conocimiento de negocio y con ello establecer las prioridades medir los resultados del valor entregado al cliente.

2.2.4.2.Artefactos

2.2.4.2.1. Product backlog

La pila de producto "es la lista de tareas clave para el desarrollo de producto, del cual es responsable el *Product Owner*, las actividades de *Product Backlog* están ordenadas en función de otorgar el máximo valor al producto en desarrollo" (Rojas, 2021, párr. 19).

2.2.4.2.2. Sprint backlog

"Es el listado de tareas del *Product Backlog* que se emplearán en un *Sprint*, pues tiene una duración máxima de 6 semanas, mientras que el *Product Backlog* puede enlistar una gran cantidad de tareas que requiera ser distribuido en varios *Sprints*" (Rojas, 2021, párr. 20).

2.2.4.2.3. Burndown charts

Los gráficos de trabajo pendientes permiten ver el ritmo al que se está cumpliendo los objetivos acordes a los requisitos, ayuda a proyectar si el tiempo

estimado para realizar el trabajo es el adecuado, el trabajo pendiente se puede graficar en base a los días pendientes para completar los requisitos del proyecto y también en relación a las horas pendientes para contemplar las tareas de la iteración (ProyectosAgiles, 2021).

2.2.4.2.4. Definition of done

Definición de hechos o definición de terminado se especifica todo lo que se requiere realizar para los elementos este artefacto es idóneo para efectuar la documentación de los tipos de pruebas del proyecto, se puede establecer como terminado cuando un representante del dueño del producto haya participado de las pruebas ejecutadas (Rad & Turley, 2019).

2.2.4.3.Actividades

2.2.4.3.1. Sprint planning

Para Piattini et al. (2014) la planificación de la iteración tiene como objetivo identificar qué parte del *product backlog* se convertirá en *software* funcional en este *sprint*, esta actividad se divide en dos:

- En el qué, *sprint planning meeting* general el cliente presenta la lista de requerimientos en orden de prioridad y el equipo la examina estableciendo los objetivos/requisitos para la iteración.
- En el cómo, *sprint planning meeting* local el equipo planifica la iteración, crea la estrategia para conseguir el resultado más óptimo, se definen las tareas y con ello se crea la lista de la iteración, se estima el esfuerzo requerido para realizar las tareas, la distribución de las tareas es en base a una autoasignación por parte de los miembros del equipo.

2.2.4.3.2. Sprint execution

Según ProyectosAgiles (2021) kos proyectos se ejecutan en bloques temporalmente comprendidos de un mes o quince días, cada iteración debe emitir un resultado completo, consiguiendo con esto incrementos con altas posibilidades de ser entregables y que puedan ser utilizados. En las iteraciones los miembros del equipo realizan la siguiente dinámica:

Cada miembro del equipo inspecciona el trabajo de sus compañeros y emite de ser necesario adaptaciones para las dificultades encontradas, actualiza la lista de las tareas de la iteración y los gráficos de trabajo pendientes.

El *scrum master* elimina los inconvenientes que no pueden ser solventados por el equipo, evitando así interrupciones externas que perjudiquen la productividad o consecución del objetivo de la iteración.

2.2.4.3.3. Scrum daily meeting

En el libro *Gestión de proyectos aplicada al PMBOK* de Sarmiento et al. (2020) menciona que la reunión diaria se planifica para desarrollarse en 15 minutos de preferencia a primeras horas de la mañana dirigida por el *scrum master* para realizar un seguimiento de los problemas identificados, en esta reunión participan principalmente los integrantes del equipo con la finalidad de coordinar el trabajo y exponer los problemas que enfrenten, sin que esto quiera decir que se ofrezcan posibles soluciones. Básicamente la reunión se centra en responder tres preguntas para saber que está haciendo cada miembro del equipo para lograr los objetivos de la iteración y evaluar su progreso:

- ¿Lo que ha hecho a partir de la última reunión?
- ¿Lo que planea hacer hoy?

- ¿Hay dificultades que impiden el progreso?

2.2.4.3.4. Sprint review

La demostración de requisitos completados es una reunión informal en la que el equipo revisa los resultados alcanzados con relación al objetivo de la iteración, los problemas y las soluciones. También presenta los incrementos del producto preparado para la entrega que solventan los requisitos completados (ProyectosAgiles, 2021).

2.2.4.3.5. Sprint retrospective

"Después de la revisión de la iteración, pero, antes de la próxima reunión de planificación de iteración, el equipo de desarrollo realiza una actividad de inspección y adaptación" (Sarmiento et al., 2020, p. 162). Permite que el equipo tenga en cuenta los comentarios u observaciones efectuados en la revisión de la iteración por parte del propietario del producto.

2.2.4.3.6. Product backlog refinement

Se entiende por refinamiento de la lista de requisitos y cambios en el proyecto al proceso continuo en el que el cliente y el equipo de desarrollo pueden añadir detalle, estimaciones y secuencia a los elementos de dicha lista, para ello se examinan y revisan sus elementos. Los elementos de la lista que se encuentran en la parte más alta poseen mayor claridad y detalle, mientras que los que están en la parte más baja tienen menor grado de claridad y detalle (Schwaber & Sutherland, 2016).

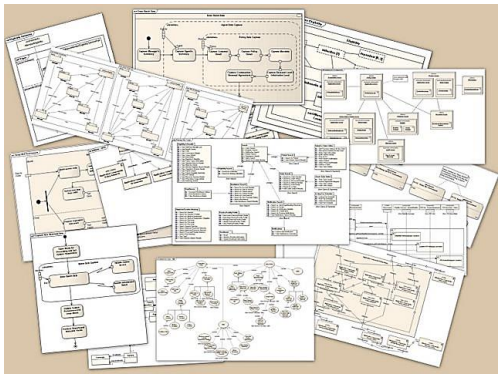
2.2.5. Lenguaje unificado de modelado UML

El UML es un lenguaje visual estándar empleado para realizar representaciones esquemáticas de los programas ya que permite especificar, construir y documentar los componentes que forman o formarán parte de un sistema de información. UML permite crear modelos que ayudan a comunicar las ideas para la resolución de problemas, es independiente del lenguaje de programación que se requiera utilizar para la codificación del sistema (Molinero, 2018).

Microsoft 365 Team (2019) menciona que en UML se puede representar varios tipos de diagramas como se puede apreciar en la figura 5, existe dos grupos principales: los diagramas estructurales quienes permiten diagramar y visualizar las diferentes estructuras estáticas que conforman un *software* o sistema, muestran los niveles de abstracción e implementación, ayudan a jerarquizar los componentes o módulos, además, cómo estos están conectados e interactuando, y los diagramas de comportamiento que son empleados para trabajar con los aspectos dinámicos del sistema de *software* o proceso, permiten ver la funcionalidad y lo que debe ocurrir en el sistema que se está modelando.

Figura 5

Diagramas UML



Nota. Tomado de Collage de diagramas uml, por Kishorekumar 62, 2009, Wikimedia Commons (https://commons.wikimedia.org/wiki/File:UML_Diagrams.jpg). BY CC

2.2.5.1.Tipos de diagramas estructurales

2.2.5.1.1. Diagramas de clases

Este diagrama representa el diseño lógico y físico de un sistema, y muestra sus clases y la forma en la que se interrelacionan: que puede ser asociación, dependencia, especialización o en paquetes, la estructura interna de una clase está formada por sus atributos y las operaciones que se pueden realizar (García, 2018).

2.2.5.1.2. Diagrama de objetos

Generalmente son empleados para probar la precisión de los diagramas de clases, dado que los objetos son instancias de las clases, permite ver los objetos de un sistema y como están relacionados en un momento en particular (Microsoft 365 Team, 2019).

2.2.5.1.3. Diagrama de componentes

Sirven para mostrar las partes del *software*, sus interfaces y las dependencias entre ellos, tienen nivel más alto de abstracción que el diagrama de clases. Un componente se puede implementar por una o varias clases en tiempo de ejecución (Gómez & Moraleda, 2020).

2.2.5.2.Tipos de diagramas de comportamiento

2.2.5.2.1. Diagrama de actividades

Permiten describir cómo se implementa la funcionalidad de un sistema, capturan acciones y sus resultados, la secuencia de actividades, es decir, el flujo de trabajo desde que empieza la actividad hasta que concluye, permite ver las bifurcaciones de una condición, indica si se ha ejecutado un proceso paralelo a la actividad (García, 2018).

2.2.5.2.2. Diagrama de casos de uso

Los diagramas de casos de uso hacen posible graficar los casos de uso, cada caso consiste en la iteración que los actores mantienen con el sistema en respuesta a un determinado evento generado por dicho actor (Trujillo, 2018).

2.2.5.2.3. Diagrama de secuencia

Muestra la secuencia cronológica de mensajes e interacciones entre actores y objetos. Es conveniente generar un diagrama de secuencia para cada acción que ha generado el usuario en los casos de uso y estos a su vez responden a la definición de las clases y sus respectivos atributos y métodos (Vidal-Silva et al., 2019).

2.2.6. Bases de datos

Las bases de datos son un conjunto de datos almacenados sobre un soporte físico, este conjunto debe estar estructurado de tal forma que permita agruparse en entidades con el fin de interconectarse y solventar las necesidades de información que poseen las instituciones. La estructura debe permitir acceso directo, rápido y eficiente a los datos para utilizarlos de manera compartida por todos los usuarios de la organización (Capacho & Nieto, 2017).

2.2.6.1.Modelo de datos

Beynon-Davies (2018) manifiesta que “cualquier base de datos y sistema gestor de base de datos utiliza un determinado modelo de datos. Un modelo de datos es una arquitectura para los datos. Describe la estructura la estructura general de organización de los datos” (p. 144).

2.2.6.1.1. Modelo entidad relación

Representa una idea del mundo real, en la figura 6 se puede visualizar de forma general que estos diagramas están compuestos por entidades y vinculadas entre sí a través de las relaciones, cada entidad posee atributos los cuales describen propiedades o características de cada una de las entidades. Este modelo se convierte en un pilar fundamental para el diseño de la base de datos debido a que permite especificar un esquema lógico completo acorde al contexto propio de la empresa (Tejada & Martínez, 2019).

Figura 6

Diagrama entidad relación



Nota. La entidad 1 se encuentra relacionada con la entidad 2, Elaboración propia

2.2.6.1.2. Diseño conceptual

El diseño conceptual permite realizar la abstracción de los datos inherentes al contexto y presentar una descripción entendible, se identifican las entidades que forman parte de la estructura, los atributos, tipos de datos, claves primarias, es independiente del sistema gestor de bases de datos.

2.2.6.1.3. Diseño físico

El modelo físico permite desarrollar toda la estructura de la base de datos, partiendo por las tablas de las que se puede especificar las columnas, tipo de datos, clave primaria y la relación entre las tablas a través de las claves foráneas. El modelo

físico obedece a las particularidades del sistema gestor de bases de datos (Pulido et al., 2019).

2.2.6.2.Elementos del modelo relacional

2.2.6.2.1. Entidades

Según Martínez (2019), una entidad permite representar un objeto o cosa a través de un conjunto de propiedades o atributos que la permiten diferenciarse de otras.

2.2.6.2.2. Atributos

Son las propiedades o características que conforman una entidad, cada atributo permite almacenar un dato (Martínez, 2019).

2.2.6.2.3. Tipo de dato

Los tipos de datos son especificaciones que se les otorga a los atributos con el fin de determinar qué valores se pueden almacenar en cada atributo, por ejemplo: carácter, número, fecha, entre otros. Los tipos de datos dependerá del sistema gestor de bases de datos que se esté empleando (Martínez, 2019).

2.2.6.2.4. Clave primaria

Es un atributo que permite diferenciar de manera unívoca a los diferentes registros que forman parte de las entidades, la clave primaria siempre debe contener datos, es decir, no puede quedar vacía (Guerrero, 2014).

2.2.6.2.5. Clave foránea

Es un atributo que se utiliza para referenciar los datos de una entidad en otra, para ser clave foránea en una entidad es necesario que esta sea clave principal en la otra entidad, los datos almacenados en este tipo de atributo pueden quedar vacío o contener datos que se repitan (Guerrero, 2014).

2.2.6.2.6. Cardinalidad

Especifica la cantidad de relaciones que pueden existir entre dos entidades o conjunto de entidades. Existiendo así la cardinalidad mínima, donde señala la cantidad mínima de relaciones en las que puede aparecer cada ejemplar de la entidad, y la cardinalidad máxima que hace referencia a la cantidad máxima de relaciones (Trujillo, 2018). De manera general se puede mencionar las siguientes cardinalidades:

- Uno a ninguno (1, 0).
- Muchos a ninguno (n, 0).
- Uno a uno (1, 1).
- Uno a muchos (1, n).
- Muchos a uno (n, 1).
- Muchos a muchos (n, n).

2.2.6.3.Sistema gestor de bases de datos

Es un conjunto de aplicaciones que permiten acceder, almacenar y gestionar una o varias bases de datos facilitando a los usuarios principalmente cuatro interacciones tales como: el mantenimiento estructural, el procesamiento de transacciones, la

recuperación de información y la administración de las bases de datos, a través de herramientas específicas para cada una de estas acciones (Beynon-Davies, 2018).

2.2.6.3.1. MariaDB

MariaDB Server es uno de los SGBD de código abierto con mayor popularidad para ser empleados en modelos relacionales. Está desarrollado con la misma esencia de MYSQL con el objetivo de mantener una distribución libre, esta esencia permite trabajar con la sintaxis MYSQL, posee buenos valores de apertura, desempeño y estabilidad, está presente en un gran porcentaje de distribuciones de Linux.

Para mejorar el acoplamiento de los *clusters* de bases de datos y trabajar con varios nodos y tener alta disponibilidad, MariaDB incorpora Galera Cluster 4, trabaja con Temporal Data Tables para de esta manera consultar datos con valores anteriores (MariaDB Foundation, 2019).

2.2.6.3.2. PostgreSQL

Es un gestor de bases de datos objeto-relacional de código abierto y distribuido bajo licencia Berkeley Software Distribution (BSD), emplea la arquitectura cliente servidor, para garantizar la estabilidad del sistema utiliza multiprocesos con el objetivo de que si falla un proceso no afecte a los demás y por ende el sistema siga trabajando (PostgreSQL, como se citó en Zea et al., 2017).

2.2.6.4.Lenguaje de consulta SQL

El lenguaje de consulta estructurado es el estándar empleado por los sistemas gestores de bases de datos relacionales, aunque cada SGBD maneja sus particularidades en la sintaxis al momento de escribir, está compuesto de cláusulas, comandos, funciones

y operadores, permitiendo de esta manera en términos generales crear y manipular las bases de datos. SQL a su vez se encuentra dividido en DDL (lenguaje de definición de datos), DML (lenguaje de manipulación de datos) y DCL (lenguaje de control de datos)(Camuña, 2014).

2.2.6.4.1. Lenguaje de definición de datos

Es un lenguaje constituido por comandos que se emplean para establecer la base de datos, es decir, se puede crear (*créate*), modificar (*alter*) o eliminar (*drop*) su estructura (Bisson & Godoc, 2018).

2.2.6.4.2. Lenguaje de manipulación de datos

Este lenguaje es el encargado de realizar las consultas necesarias para trabajar con la información almacenada en la base de datos, dicho de otra manera, se puede ingresar (*insert*), modificar (*update*), eliminar (*delete*) y seleccionar (*select*) los datos (Bisson & Godoc, 2018).

2.2.6.4.3. Lenguaje de control de datos

Es empleado para la administración de la base de datos, mediante este lenguaje se puede establecer quiénes son los usuarios autorizados y los privilegios asignados a cada uno de ellos(Beynon-Davies, 2018).

Capítulo 3. Marco empírico

3.1. Desarrollo de la investigación

3.1.1. Clasificación de la investigación

La investigación de campo, a decir de Monroy y Nava (2018), son las que se realizan en el lugar de los acontecimientos permitiendo obtener información de fuentes directas, utiliza la observación de fenómenos en condiciones reales, sin manipular o controlar variables. El presente trabajo se enmarca en este tipo de investigación porque la información se la ha obtenido realizando una visita *in situ* al Departamento de Investigación de la Universidad Técnica Luis Vargas Torres de Esmeraldas Sede Santo Domingo, donde se pudo ver el proceso que realizan para el registro de los datos de las diferentes publicaciones científicas y cómo obtienen las estadísticas para los informes pertinentes solicitados por los órganos de control tanto internos como externos. Además, es una investigación descriptiva en virtud de que se detallan las características relevantes de los diferentes actores y objetos que están relacionados con el control de las publicaciones de los trabajos científicos desarrollados en la institución, Báez (2018) permite sostener esto ya que menciona que la investigación descriptiva

> trata de describir las características más importantes de un determinado objeto de estudio con respecto a su aparición y comportamiento, o simplemente el investigador buscará describir las maneras o formas en que éste se diferencie del mismo en otra situación. (p. 25)

La investigación bibliográfica ha permitido desarrollar la conceptualización del proyecto mediante una búsqueda rigurosa del conocimiento sobre teorías pertinentes a la temática, estas teorías se encuentran plasmadas en fuentes secundarias como: libros, artículos científicos y sitios web. José Martínez de Sousa menciona que “la investigación bibliográfica se considera una búsqueda sistemática y exhaustiva de

material editorial sobre un tema determinado" (como lo citó Martín & Lafuente, 2017, p. 152).

3.1.2. Diseño de la investigación

Hernández (2014) menciona que en el diseño de investigación no experimental las situaciones no son provocadas intencionalmente por parte del investigador, las variables inmersas en el estudio no son manipuladas, por lo tanto, no se tiene control sobre dichas variables.

3.2. Metodología scrum

3.2.1. Designación de roles

Para el desarrollo del proyecto se estableció tres roles que van acorde a las actividades que desempeña cada uno de los miembros.

3.2.1.1.Product owner

En virtud de que el magister Efraín Prado es el Responsable del Departamento de Investigación fue asignado con este rol dado que él conoce de las principales necesidades que deben ser cubiertas.

3.2.1.2.Scrum master

El magister Ricardo Celi es el miembro del grupo de trabajo con el que se ha mantenido reuniones permanentes de forma presencial y telemática ayudando a canalizar la información con respecto a los requerimientos.

3.2.1.3.Develoment team

La parte del desarrollo del proyecto estuvo a cargo del ingeniero Luis Núñez, aquí se cumplió con la ejecución de los *sprint* establecidos previamente.

3.2.2. Análisis de requerimientos

3.2.2.1.Requerimientos de la app

En la tabla 1 se establecen los requerimientos funcionales y los requerimientos no funcionales que la aplicación debe integrar.

Tabla 1.
Requerimientos de la app

Requerimientos funcionales	**Requerimientos no funcionales**
Creación de la base de datos.	Las tablas deben permitir manejar los códigos empleados en la universidad. Mantener un respaldo de la base de datos cada 24 horas.
Creación de la página home.	Colores distintivos de la universidad.
Acceso al sistema.	Controlar la cantidad de intentos erróneos.
CRUD de datos de los proyectos.	Mantenimiento sencillo de los datos.
CRUD de datos de los docentes.	Mantenimiento sencillo de los datos.
CRUD de datos de las carreras.	Mantenimiento sencillo de los datos.
CRUD de datos de las revistas.	Mantenimiento sencillo de los datos.
CRUD de datos de las publicaciones.	Mantenimiento sencillo de los datos.
Informe de publicaciones.	Los informes deben permitir definir proyectos, carreras, docentes, revistas, rango de fechas y periodos académicos.

Nota. Elaboración propia.

3.2.3. Historias de usuarios

Gracias a la entrevista mantenida con el Responsable del Departamento de Investigación se estableció las historias de usuario, las mismas que se encuentran detalladas en la tabla 2.

Tabla 2.
Historias de usuarios

HISTORIAS DE USUARIO				
Id.	Como…	Quiero…	Para…	Descripción
1	Responsable del Departamento de Investigación.	Disponer de un repositorio de datos.	Brindar información actualizada y veraz referente a las publicaciones realizadas en la universidad.	Se diseñará el modelado de la base de datos para que este permita generar el esquema en un SGBD, cargar la información para efectuar las transacciones a través de la app.
2	Responsable del Departamento de Investigación.	Crear la página home.	Tener la entrada principal a la app a las diferentes opciones de la app.	Se desarrollará una interface amigable que permita la fácil utilización de la app.
3	Responsable del Departamento de Investigación.	Acceso al sistema.	Controlar que el sistema solo sea utilizado por el personal autorizado.	Se usará un usuario con su respectiva contraseña, las mismas que deben estar almacenadas en la base de datos.
4	Responsable del Departamento de Investigación.	Registrar datos de proyectos.	Llevar control de las publicaciones científicas relacionadas con los proyectos	El usuario podrá crear nuevos proyectos con todos los datos necesarios como: código, nombre, línea, sublínea, fecha de inicio, estado, etc.
5	Responsable del Departamento de Investigación.	Actualizar datos de proyectos.	Realizar cambios en los datos de los proyectos de ser necesarios.	Se podrá realizar cambios de ciertos datos de los proyectos.
6	Responsable del Departamento de Investigación.	Registrar datos de docentes.	Tener control de los datos de los docentes.	El usuario podrá crear nuevos docentes con los siguientes datos: código, cédula, nombre, contraseña, título tercer nivel, título cuarto nivel, título phd, correo institucional, correo personal, tipo de contrato, estado.
7	Responsable del Departamento de Investigación.	Actualizar datos de docentes.	Tener la posibilidad de realizar cambios en los datos del docente.	Permitirá cambiar ciertos datos de docente, como por ejemplo uno de los títulos, el estado, el tipo de contrato.

8	Responsable del Departamento de Investigación.	Registrar datos de carreras.	Tener control de las carreras existentes en la universidad.	Registrará nuevas carreras que incluya los siguientes datos: código, nombre, estado.
9	Responsable del Departamento de Investigación.	Actualizar datos de carreras.	Tener la posibilidad de realizar cambios en los datos de las carreras.	Se podrá cambiar el nombre y el estado de las carreras.
10	Responsable del Departamento de Investigación.	Emitir informes de publicaciones por carrera.	Mantener la información detallada de las publicaciones que realiza cada carrera en los respectivos periodos.	Se podrá especificar los informes por cada carrera o carreras involucradas en una publicación, se seleccionará el periodo académico o rango de fechas.
11	Responsable del Departamento de Investigación.	Emitir informes de publicaciones por proyecto.	Presentar información de las publicaciones pertenecientes a los proyectos vigentes y también de las publicaciones independientes.	Mostrará todas las publicaciones efectuadas por un determinado proyecto, así mismo las publicaciones independientes.
12	Responsable del Departamento de Investigación.	Emitir informes de publicaciones por docente.	Medir la producción científica de cada docente.	Con esta opción se podrá hacer un seguimiento al rendimiento de cada docente. Esta opción estará habilitada para el responsable del departamento y con ciertas restricciones para el docente.
13	Docente	Registrar datos de revistas.	Llevar control de los datos de las revistas en las que se están realizando las publicaciones de los artículos.	Permitirá ingresar datos como: código, nombre, campo amplio, campo específico campo detallado, ISSN, frecuencia, país, idioma.
14	Docente	Actualizar datos de revistas.	Tener la posibilidad de realizar cambios en los datos de las revistas.	De ser necesario realizar cambios en ciertos datos de las revistas.
15	Docente	Registrar datos de publicaciones.	Llevar control de los datos de las publicaciones científicas.	Permitirá ingresar datos de los artículos científicos como: código, título, DOI, URL, idioma, archivo, año, estado.
16	Docente	Actualizar datos de publicaciones	Tener la posibilidad de realizar cambios en los datos de las publicaciones.	Ayudará a realizar modificaciones a ciertos datos de las publicaciones.

Nota. Elaboración propia.

3.2.4. Generación del product backlog

En la tabla 3, tanto la complejidad como la prioridad están establecidas en una escala de: mínima, baja, media, alta, muy alta, donde 1 significa mínima y 5 muy alta.

Tabla 3.
Product backlog

Descripción	Prioridad	Complejidad	Días estimados
Creación de la base de datos.	5	5	5
Acceso al sistema	5	4	5
Mantenimiento datos proyectos.	5	4	5
Mantenimiento datos carreras.	5	4	5
Mantenimiento datos docente	5	4	5
Crear informe de publicaciones por carrera.	4	5	8
Crear informe de publicaciones por proyectos.	4	5	8
Crear informe de publicaciones por docente.	4	5	8
Mantenimiento datos de revistas.	5	4	5
Mantenimiento de datos de publicaciones.	5	4	5

Nota. Elaboración propia.

3.2.5. Estimación de los sprints

Para agilitar el cumplimiento de los ítems del *product backlog* la metodología *scrum* emplea los *sprints*, de esta forma cada *sprint* contiene un grupo de estos ítems, de allí que, en las tablas 4, 5 y 6 se detallan los *sprints* 1, 2 y 3, respectivamente.

Tabla 4.
Estimación del sprint 1

Descripción	Prioridad	Complejidad	Días estimados
Creación de la base de datos.	5	5	5
Acceso al sistema	5	4	5
Mantenimiento datos proyectos.	5	4	5
Mantenimiento datos carreras.	5	4	5
Mantenimiento datos docente	5	4	5

Nota. Elaboración propia.

Tabla 5.
Estimación del sprint 2

Descripción	Prioridad	Complejidad	Días estimados
Crear informe de publicaciones por carrera.	4	5	8
Crear informe de publicaciones por proyectos.	4	5	8
Crear informe de publicaciones por docente.	4	5	8

Nota. Elaboración propia.

Tabla 6.
Estimación sprint 3

Descripción	Prioridad	Complejidad	Días estimados
Mantenimiento datos de revistas.	5	4	5
Mantenimiento de datos de publicaciones.	5	4	5

Nota. Elaboración propia.

3.2.6. Tareas de los sprints

3.2.6.1.Tareas del sprint 1

Con la finalidad de que la historia de usuario que permite crear la base de datos se satisfaga se deben llevar a cabo varias tareas, como lo indica la tabla 7.

Tabla 7.
Tareas sprint 1, historia de usuario creación de la base de datos.

Id	Rol	Descripción de la historia de usuario.	Tareas	Esfuerzo requerido.
1	Desarrollador	Se requiere gestionar los datos de la universidad.	Diseño del modelo entidad relación.	16
			Diseño del modelo conceptual.	8
			Selección del SGBD.	2
			Diseño del modelo físico.	2
			Generar script con las sentencias DDL.	1
			Instalación del SGBD.	2
			Configuración del SGBD.	2
			Ejecución del script.	1
			Pruebas de la base de datos.	6
			Total	40

Nota. Elaboración propia.

En la tabla 8 se muestran las tareas que se deben ejecutar para la historia de usuario que permite el acceso al sistema.

Tabla 8.
Tareas sprint 1, historia de usuario acceso al sistema.

Id	Rol	Descripción de la historia de usuario.	Tareas	Esfuerzo requerido.
2	Usuarios dados de alta	Acceder a la app a través de un formulario de logeo.	Diseñar la página principal para la app.	16
			Desarrollar la página de autenticación	8
			Establecer las urls para direccionamiento según el tipo de usuario.	16
			Total	40

Nota. Elaboración propia.

Las tareas para el mantenimiento de datos de los proyectos se encuentran detalladas en la tabla 9, las mismas que pertenecen al *sprint* 1.

Tabla 9.
Tareas sprint 1, historia de usuario mantenimiento datos proyectos.

Id	Rol	Descripción de la historia de usuario.	Tareas	Esfuerzo requerido.
3	Responsable	Se requiere ingresar, modificar y eliminar los datos de los proyectos desarrollados y que se están desarrollando en la universidad.	Crear las clases y métodos para trabajar con las funciones y procedimientos que permitan editar los datos de los proyectos.	8
			Diseñar el formulario para la visualización de la información de los proyectos.	8
			Crear el formulario para ingresar los datos del nuevo proyecto.	4
			Validar los campos requeridos de los proyectos.	4
			Desarrollar la vista con la información del proyecto a modificar.	4
			Controlar los campos permitidos para la modificación.	4
			Controlar la eliminación de un determinado proyecto.	4
			Probar las opciones para la edición de los datos de los proyectos.	4
			Total	40

Nota. Elaboración propia.

Siguiendo con el *sprint* 1, en la tabla 10 se puede observar las tareas para cumplir con el mantenimiento de los datos de las carreras aperturadas en la universidad.

Tabla 10.
Tareas sprint 1, historia de usuario mantenimiento datos carreras.

Id	Rol	Descripción de la historia de usuario.	Tareas	Esfuerzo requerido.
4	Responsable	Se requiere ingresar, modificar y eliminar los datos de las carreras aperturadas en la universidad.	Crear las clases y métodos para trabajar con las funciones y procedimientos que permitan la edición de los datos de las carreras.	8
			Diseñar el formulario para la visualización de la información de las carreras.	8
			Crear el formulario para ingresar los datos de la nueva carrera.	4
			Validar los campos requeridos de la carrera.	4

			Desarrollar la vista con la información de la carrera a modificar.	4
			Controlar los campos permitidos para la modificación.	4
			Controlar la eliminación de una determinada carrera.	4
			Probar las opciones para la edición de los datos de las carreras.	4
			Total	40

Nota. Elaboración propia.

La tabla 11 muestra las tareas especificadas para el mantenimiento de los datos de los docentes y con ello completar el *sprint* 1.

Tabla 11.
Tareas sprint 1, historia de usuario mantenimiento datos docentes.

Id	Rol	Descripción de la historia de usuario.	Tareas	Esfuerzo requerido.
5	Responsable	Se requiere ingresar, modificar y eliminar los datos de los docentes que prestan y han prestado sus servicios en la universidad.	Crear las clases y métodos para trabajar con las funciones y procedimientos que permitan la edición de los datos de los docentes.	8
			Diseñar el formulario para la visualización de la información de los docentes.	8
			Crear el formulario para ingresar los datos del nuevo docente.	4
			Validar los campos requeridos del docente.	4
			Desarrollar la vista con la información del docente a modificar.	4
			Controlar los campos permitidos para la modificación.	4
			Controlar la eliminación de un determinado docente.	4
			Probar las opciones para la edición de los datos de los docentes.	4
			Total	40

Nota. Elaboración propia.

3.2.6.2.Tareas del sprint 2

Las tareas definidas para el *sprint* 2 se muestran en las tres siguientes tablas, específicamente en la tabla 12, están las tareas para la historia de usuario que requiere

crear un informe de las publicaciones desarrolladas por cada una de las carreras. En la tabla 13 se puede visualizar las tareas pertinentes a crear informes relacionados con las publicaciones ejecutadas en los respectivos proyectos y, por último, en este mismo sprint están las tareas para emitir los informes de las publicaciones escritas por los docentes (ver tabla 14).

Tabla 12.
Tareas sprint 2, historia de usuario crear informe de publicaciones por carrera.

Id	Rol	Descripción de la historia de usuario.	Tareas	Esfuerzo requerido.
6	Responsable	Se requiere emitir reportes de las publicaciones realizadas en cada una de las carreras.	Crear las clases y métodos para trabajar con las funciones y procedimientos que permitan visualizar la información pertinente.	16
			Diseñar el formulario para la visualización de la información de las publicaciones desarrolladas en cada carrera.	8
			Crear los controles para enviar los argumentos necesarios para emitir el reporte.	8
			Validar los campos requeridos para el reporte.	8
			Crear el formato de los reportes.	16
			Realizar las pruebas necesarias.	8
			Total	64

Nota. Elaboración propia.

Tabla 13.
Tareas sprint 2, historia de usuario crear informe de publicaciones por proyectos.

Id	Rol	Descripción de la historia de usuario.	Tareas	Esfuerzo requerido.
7	Responsable	Se requiere emitir reportes de las publicaciones realizadas en cada uno de los proyectos.	Crear las clases y métodos para trabajar con las funciones y procedimientos que permitan visualizar la información pertinente.	16
			Diseñar el formulario para la visualización de la información de las publicaciones desarrolladas en cada proyecto.	8
			Crear los controles para enviar los argumentos necesarios para emitir el reporte.	8
			Validar los campos requeridos para el reporte.	8
			Crear el formato de los reportes.	16
			Realizar las pruebas necesarias.	8
			Total	64

Nota. Elaboración propia.

Tabla 14.
Tareas sprint 2, historia de usuario crear informe de publicaciones por docente.

Id	Rol	Descripción de la historia de usuario.	Tareas	Esfuerzo requerido.
8	Responsable	Se requiere emitir reportes de las publicaciones realizadas por cada uno de los docentes de la universidad.	Crear las clases y métodos para trabajar con las funciones y procedimientos que permitan visualizar la información pertinente.	16
			Diseñar el formulario para la visualización de la información de las publicaciones desarrolladas por los docentes.	8
			Crear los controles para enviar los argumentos necesarios para emitir el reporte.	8
			Validar los campos requeridos para el reporte.	4
			Crear el formato de los reportes.	4
			Realizar las pruebas necesarias.	4
	Docente	Se requiere visualizar cada una de las publicaciones en las que ha participado el docente.	Controlar que la información visualizada sea exclusivamente de las publicaciones en las que el docente ha participado.	8
			Validar los campos requeridos para el reporte.	4
			Crear el formato de los reportes.	4
			Realizar las pruebas necesarias.	4
			Total	64

Nota. Elaboración propia.

3.2.6.3.Tareas del sprint 3

La tabla 15 permite ver las tareas para llevar a cabo el mantenimiento de los datos de las revistas en las que la universidad ha realizado las publicaciones y con esto empezar el *sprint* 3.

Tabla 15.
Tareas sprint 3, historia de usuario mantenimiento de datos revistas.

Id	Rol	Descripción de la historia de usuario.	Tareas	Esfuerzo requerido.
9	Docente	Se requiere ingresar, modificar los datos de las revistas en las que el docente ha desarrollado la publicación.	Crear las clases y métodos para trabajar con las funciones y procedimientos que permitan el ingreso y la modificación de los datos de las revistas.	8
			Diseñar el formulario para la visualización de la información de las revistas.	8

		Crear el formulario para ingresar los datos de la nueva revista.	4
		Validar los campos requeridos de las revistas.	4
		Desarrollar la vista con la información de la revista a modificar.	4
		Controlar los campos permitidos para la modificación.	4
Responsable	Se requiere eliminar los datos de las revistas en la que se ha desarrollado publicaciones.	Controlar la eliminación de un determinado proyecto.	4
		Probar las opciones para la edición de los datos de los proyectos.	4
		Total	40

Nota. Elaboración propia.

El *sprint* 3 culmina con las tareas necesarias para que el mantenimiento de los datos de las publicaciones se realice adecuadamente (ver tabla 16).

Tabla 16.
Tareas sprint 3, historia de usuario mantenimiento de datos publicaciones.

Id	Rol	Descripción de la historia de usuario.	Tareas	Esfuerzo requerido.
10	Docente	Se requiere ingresar, modificar y eliminar los datos de las publicaciones realizadas por el docente.	Crear las clases y métodos para trabajar con las funciones y procedimientos que permitan el ingreso, modificación y eliminación de los datos de las publicaciones.	8
			Diseñar el formulario para la visualización de la información de las publicaciones.	8
			Crear el formulario para ingresar los datos de la nueva publicación.	4
			Validar los campos requeridos de las publicaciones.	4
			Desarrollar la vista con la información de la publicación a modificar.	4
			Controlar los campos permitidos para la modificación.	4
			Controlar la eliminación de un determinado proyecto.	4
			Probar las opciones para la edición de los datos de los proyectos.	4
			Total	40

Nota. Elaboración propia.

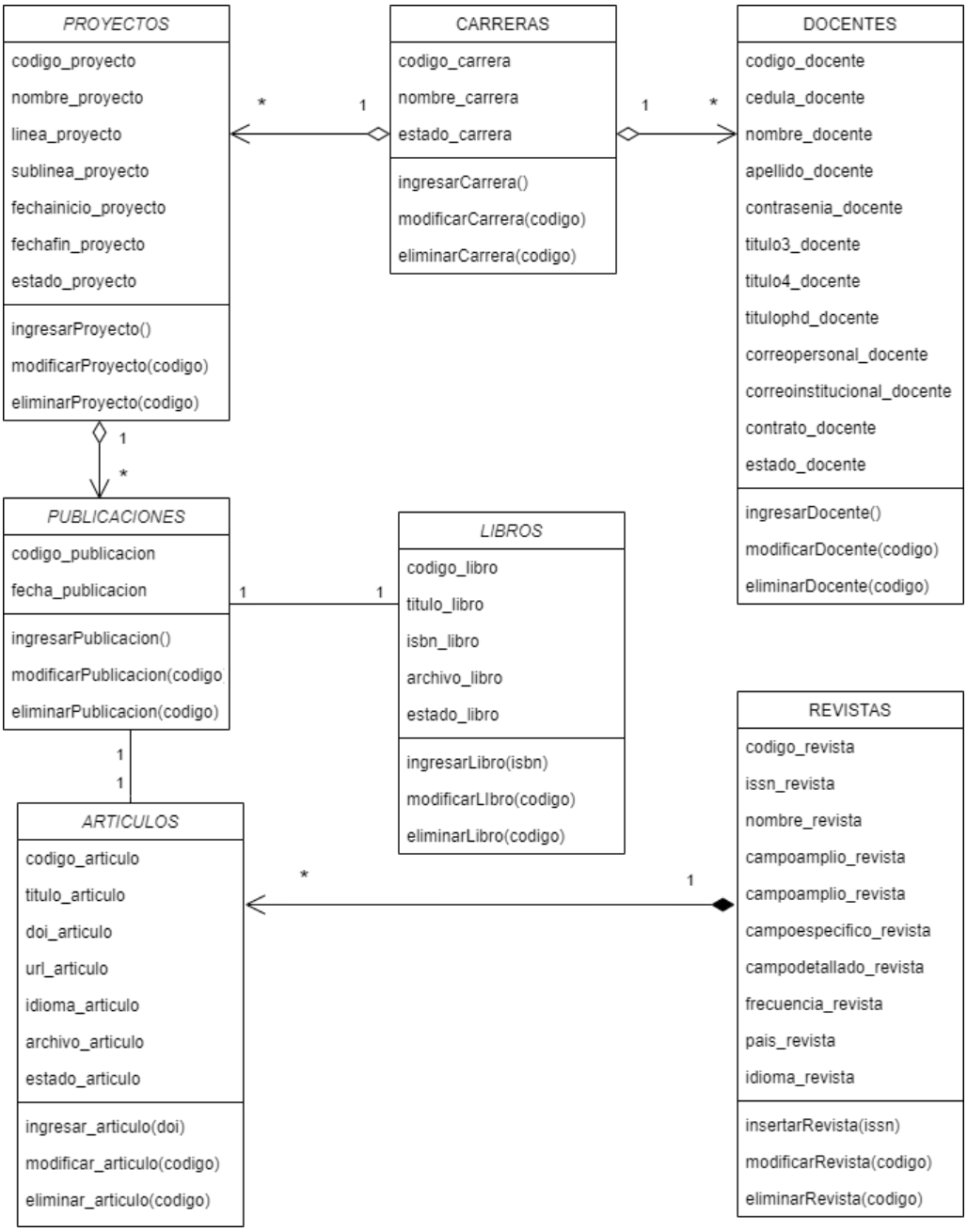
3.3. Diagramas UML

3.3.1. Diagrama de clases

La figura 7 muestra la estructura del sistema, con sus respectivas clases, atributos y métodos, y la relación que existe entre los diferentes objetos.

Figura 7

Diagrama de clases



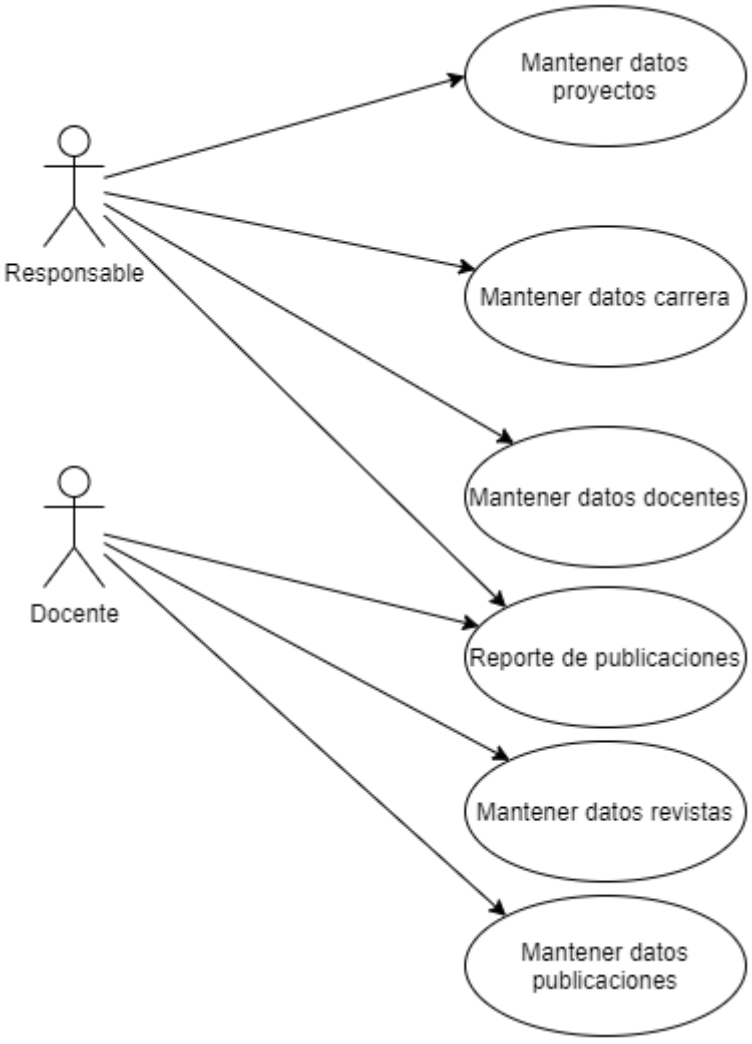
Nota. Elaboración propia.

3.3.2. Diagrama de casos de uso

En el diagrama de caso de uso, representado en la figura 8, se observa la relación entre los requerimientos para el sistema y los actores.

Figura 8

Diagrama de casos de uso



Nota. Elaboración propia.

3.3.2.1.Descripción de los casos de uso

La tabla 17 describe la interacción entre el Responsable del Departamento de Investigación y el sistema para mantener los datos de los proyectos y con ello cumplir con el objetivo especificado para el caso de uso correspondiente.

Tabla 17.
Descripción caso de uso mantener datos proyectos

Caso de uso 1	Nombre de caso de uso: Mantener datos proyectos
Actores	Responsable del departamento de investigación.
Objetivo	Permitir al responsable del departamento de investigación realizar el ingreso, actualización y eliminación de los datos de los proyectos de investigación que se desarrollan en la universidad.
Precondiciones	El responsable del departamento debe tener una cuenta administradora.
Postcondición	El responsable del departamento podrá mantener en correcto estado los datos relacionados con los proyectos.
Flujo de eventos	
Acción del actor	**Respuesta del sistema**
1. Clic en el enlace Edición Proyectos.	2. Carga el formulario con la información de los proyectos y las opciones: nuevo, modificar, eliminar.
3. Escoge la opción: nuevo.	4. Carga el formulario de la opción nuevo.
5. Registra los datos correspondientes al nuevo proyecto.	
6. Clic en el botón guardar.	7. Chequea que los datos obligatorios estén presentes. 8. Emite un mensaje sobre el éxito o fracaso de la transacción.
9. Selecciona la opción modificar del registro pertinente.	10. Muestra el formulario con la información del proyecto seleccionado. 11. Habilita los campos que se pueden modificar.
12. Ingresa los nuevos valores.	13. Cheque que los datos obligatorios estén presentes. 14. Emite un mensaje sobre el éxito o fracaso de la transacción.
15. Selecciona la opción eliminar del registro pertinente.	16. Muestra un mensaje de confirmación para evitar una eliminación involuntaria. 17. Emite un mensaje de éxito o error de la transacción.
Situaciones excepcionales.	
• No se puede establecer la conexión con la base de datos. • Faltan los campos requeridos. • La fecha de inicio del proyecto es mayor a la fecha de fin. • El proyecto ya existe.	

Nota. Elaboración propia.

El caso de uso para mantener los datos de las carreras permite que el Responsable del departamento realice las acciones pertinentes que le lleven a cumplir con el objetivo y que el sistema responda de forma efectiva, la tabla 18 describe esta interacción.

Tabla 18.
Descripción caso de uso mantener datos carreras

Caso de uso 2	Nombre de caso de uso: Mantener datos carreras
Actores	Responsable del departamento de investigación.
Objetivo	Permitir al responsable del departamento de investigación realizar el ingreso, actualización y eliminación de los datos de las carreras aperturadas en la universidad.
Precondiciones	El responsable del departamento debe tener una cuenta administradora.
Postcondición	El responsable del departamento podrá mantener en correcto estado los datos relacionados con las carreras.
Flujo de eventos	
Acción del actor	**Respuesta del sistema**
1. Clic en el enlace Edición Carreas.	2. Carga el formulario con la información de las carreras y las opciones: nuevo, modificar, eliminar.
3. Escoge la opción nuevo.	4. Carga el formulario de la opción nuevo.
5. Registra los datos correspondientes a la nueva carrera.	
6. Clic en el botón guardar.	7. Chequea que los datos obligatorios estén presentes. 8. Emite un mensaje sobre el éxito o fracaso de la transacción.
9. Selecciona la opción modificar del registro pertinente.	10. Muestra el formulario con la información de la carrera seleccionada. 11. Habilita los campos que se pueden modificar.
12. Ingresa los nuevos valores.	13. Cheque que los datos obligatorios estén presentes. 14. Emite un mensaje sobre el éxito o fracaso de la transacción.
15. Selecciona la opción eliminar del registro pertinente.	16. Muestra un mensaje de confirmación para evitar una eliminación involuntaria. 17. Emite un mensaje de éxito o error de la transacción.
Situaciones excepcionales.	
• No se puede establecer la conexión con la base de datos. • Faltan los campos requeridos. • La carrera ya existe.	

Nota. Elaboración propia.

El caso de uso para mantener los datos de los docentes se describe en la tabla 19.

Tabla 19.

Descripción caso de uso mantener datos docentes

Caso de uso 3	Nombre de caso de uso: Mantener datos docentes
Actores	Responsable del departamento de investigación.
Objetivo	Permitir al responsable del departamento de investigación realizar el ingreso, actualización y eliminación de los datos de los docentes de la universidad.
Precondiciones	El responsable del departamento debe tener una cuenta administradora.
Postcondición	El responsable del departamento podrá mantener en correcto estado los datos relacionados con los docentes.
Flujo de eventos	
Acción del actor	**Respuesta del sistema**
1. Clic en el enlace Edición Docentes.	2. Carga el formulario con la información de los docentes y las opciones: nuevo, modificar, eliminar.
3. Escoge la opción: nuevo.	4. Carga el formulario de la opción nuevo.
5. Registra los datos correspondientes a los nuevos docentes.	
6. Clic en el botón guardar.	7. Chequea que los datos obligatorios estén presentes. 8. Emite un mensaje sobre el éxito o fracaso de la transacción.
9. Selecciona la opción modificar del registro pertinente.	10. Muestra el formulario con la información del docente seleccionado. 11. Habilita los campos que se pueden modificar.
12. Ingresa los nuevos valores.	13. Cheque que los datos obligatorios estén presentes. 14. Emite un mensaje sobre el éxito o fracaso de la transacción.
15. Selecciona la opción eliminar del registro pertinente.	16. Muestra un mensaje de confirmación para evitar una eliminación involuntaria. 17. Emite un mensaje de éxito o error de la transacción.
Situaciones excepcionales.	
• No se puede establecer la conexión con la base de datos. • Faltan los campos requeridos. • El docente ya existe.	

Nota. Elaboración propia.

En la tabla 20 se detalla las acciones necesarias que el Responsable tiene que realizar para que los reportes de las publicaciones se efectúen con éxito.

Tabla 20.
Descripción caso de uso reporte publicaciones

Caso de uso 4	Nombre de caso de uso: Reporte publicaciones.
Actores	Responsable del departamento de investigación y docentes.
Objetivo	Permitir al responsable del departamento de investigación y los docentes realicen reportes de las publicaciones desarrolladas en la universidad.
Precondiciones	El responsable del departamento y los docentes deben tener una cuenta en el sistema.
Postcondición	El responsable del departamento y los docentes podrán realizar reportes de las publicaciones según sus roles.
Flujo de eventos	
Acción del actor	**Respuesta del sistema**
1. Clic en el enlace Reporte Publicaciones.	2. Carga el formulario con la información de las publicaciones acorde al rol del usuario.
3. Selecciona los filtros por los cuales desea especificar los reportes.	4. Chequea que los argumentos especificados sean correctos. 5. Emite los respectivos reportes en el caso de ejecutarse con éxito la consulta o un mensaje de error en caso de haber inconvenientes.
6. Selecciona el destino de los reportes.	7. Emite un mensaje sobre el éxito o fracaso de la consulta.
Situaciones excepcionales.	
• No se puede establecer la conexión con la base de datos. • Los parámetros indicados son incorrectos. • La fecha indicada es mayor a la actual. • Sus privilegios no le permiten realizar este reporte.	

Nota. Elaboración propia.

La descripción del caso de uso que se encarga de mantener los datos de las revistas en las que la universidad ha realizado sus publicaciones se encuentra detallada en la tabla 21.

Tabla 21.
Descripción caso de uso mantener datos revistas

Caso de uso 5	Nombre de caso de uso: Mantener datos revistas.
Actores	Docentes.
Objetivo	Permitir a los docentes realizar el ingreso, actualización y eliminación de los datos de las revistas en las que está contenido una publicación.
Precondiciones	El docente debe poseer una cuenta registrada en el sistema.
Postcondición	El docente podrá mantener en correcto estado de los datos relacionados con las revistas.
Flujo de eventos	
Acción del actor	**Respuesta del sistema**
1. Clic en el enlace Edición Revistas.	2. Carga el formulario con la información de las revistas y las opciones: nuevo, modificar, eliminar.
3. Escoge la opción: nuevo.	4. Carga el formulario de la opción nuevo.
5. Registra los datos correspondientes a la nueva revista.	6. Chequea que los datos obligatorios estén presentes.
7. Clic en el botón guardar.	8. Consulta el ISSN ingresado. 9. Emite un mensaje sobre el éxito o fracaso de la transacción.
10. Selecciona la opción modificar del registro pertinente.	11. Muestra el formulario con la información de la revista seleccionada. 12. Habilita los campos que se pueden modificar.
13. Ingresa los nuevos valores.	14. Cheque que los datos obligatorios estén presentes. 15. Emite un mensaje sobre el éxito o fracaso de la transacción.
16. Selecciona la opción eliminar del registro pertinente.	17. Muestra un mensaje de confirmación para evitar una eliminación involuntaria. 18. Emite un mensaje de éxito o error de la transacción.
Situaciones excepcionales.	
• No se puede establecer la conexión con la base de datos. • Faltan los campos requeridos. • La revista ya existe.	

Nota. Elaboración propia.

En el siguiente caso de uso los docentes pueden realizar el mantenimiento de los datos concernientes a las publicaciones, la tabla 22 muestra este proceso.

Tabla 22.
Descripción caso de uso mantener datos publicaciones

Caso de uso 6	Nombre de caso de uso: Mantener datos publicaciones.
Actores	Docentes.
Objetivo	Permitir a los docentes realizar el ingreso, actualización y eliminación de los datos de las publicaciones efectuadas.
Precondiciones	El docente debe tener una cuenta registrada en el sistema.
Postcondición	El docente podrá mantener en correcto estado de los datos relacionados con las publicaciones desarrolladas por él.
Flujo de eventos	
Acción del actor	**Respuesta del sistema**
1. Clic en el enlace Edición Publicaciones.	2. Carga el formulario con la información de las publicaciones por el docente logueado y las opciones: nuevo, modificar, eliminar.
3. Escoge la opción nuevo.	
	4. Carga el formulario de la opción nuevo.
5. Registra los datos correspondientes a la nueva publicación.	
	6. Chequea que los datos obligatorios estén presentes.
7. Clic en el botón guardar.	
	8. Consulta el DOI ingresado.
10. Selecciona la opción modificar del registro pertinente.	9. Emite un mensaje sobre el éxito o fracaso de la transacción.
	11. Muestra el formulario con la información de la publicación seleccionada.
13. Ingresa los nuevos valores.	12. Habilita los campos que se pueden modificar.
	14. Cheque que los datos obligatorios estén presentes.
16. Selecciona la opción eliminar del registro pertinente.	15. Emite un mensaje sobre el éxito o fracaso de la transacción.
	17. Muestra un mensaje de confirmación para evitar una eliminación involuntaria.
	18. Emite un mensaje de éxito o error de la transacción.
Situaciones excepcionales.	
• No se puede establecer la conexión con la base de datos. • Faltan los campos requeridos. • La publicación ya existe.	

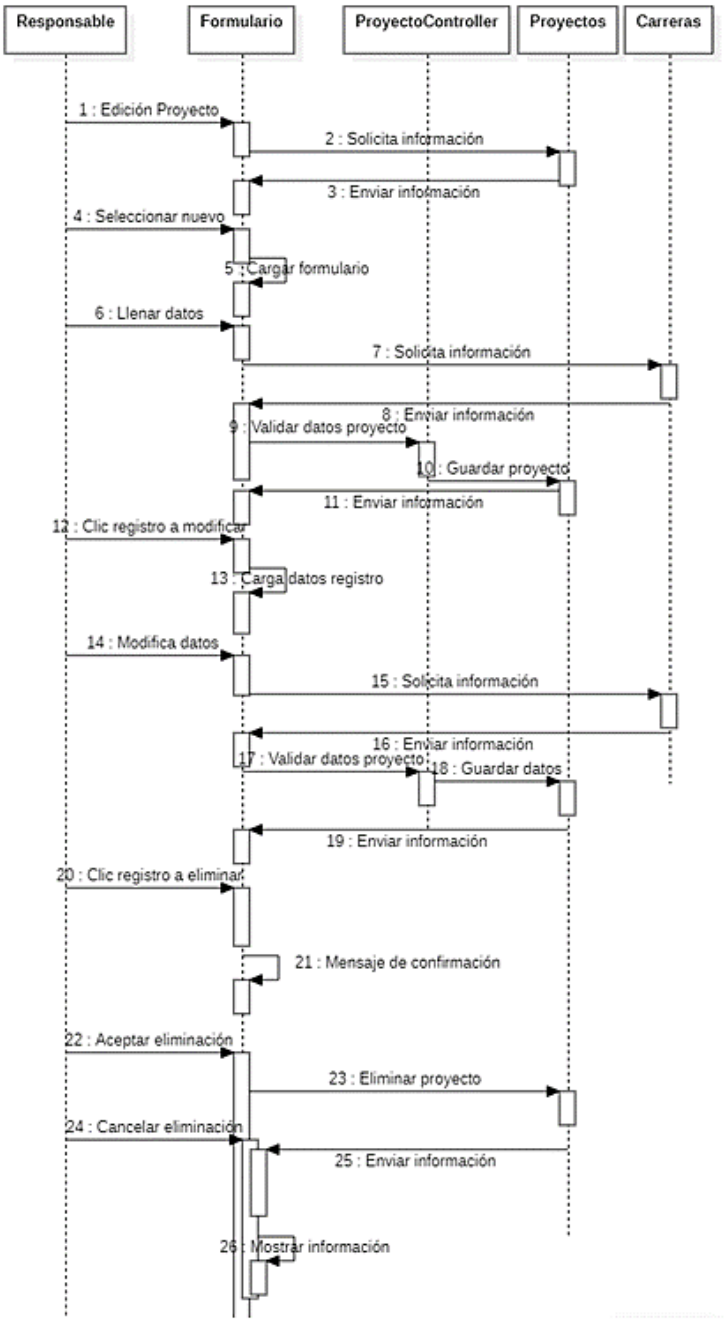
Nota. Elaboración propia.

3.3.3. Diagramas de secuencia

La figura 9 permite ver el comportamiento del sistema frente al intercambio de los diferentes mensajes emitidos por los objetos relacionados para efectuar la edición de los datos de los proyectos.

Figura 9

Diagrama de secuencia edición proyectos

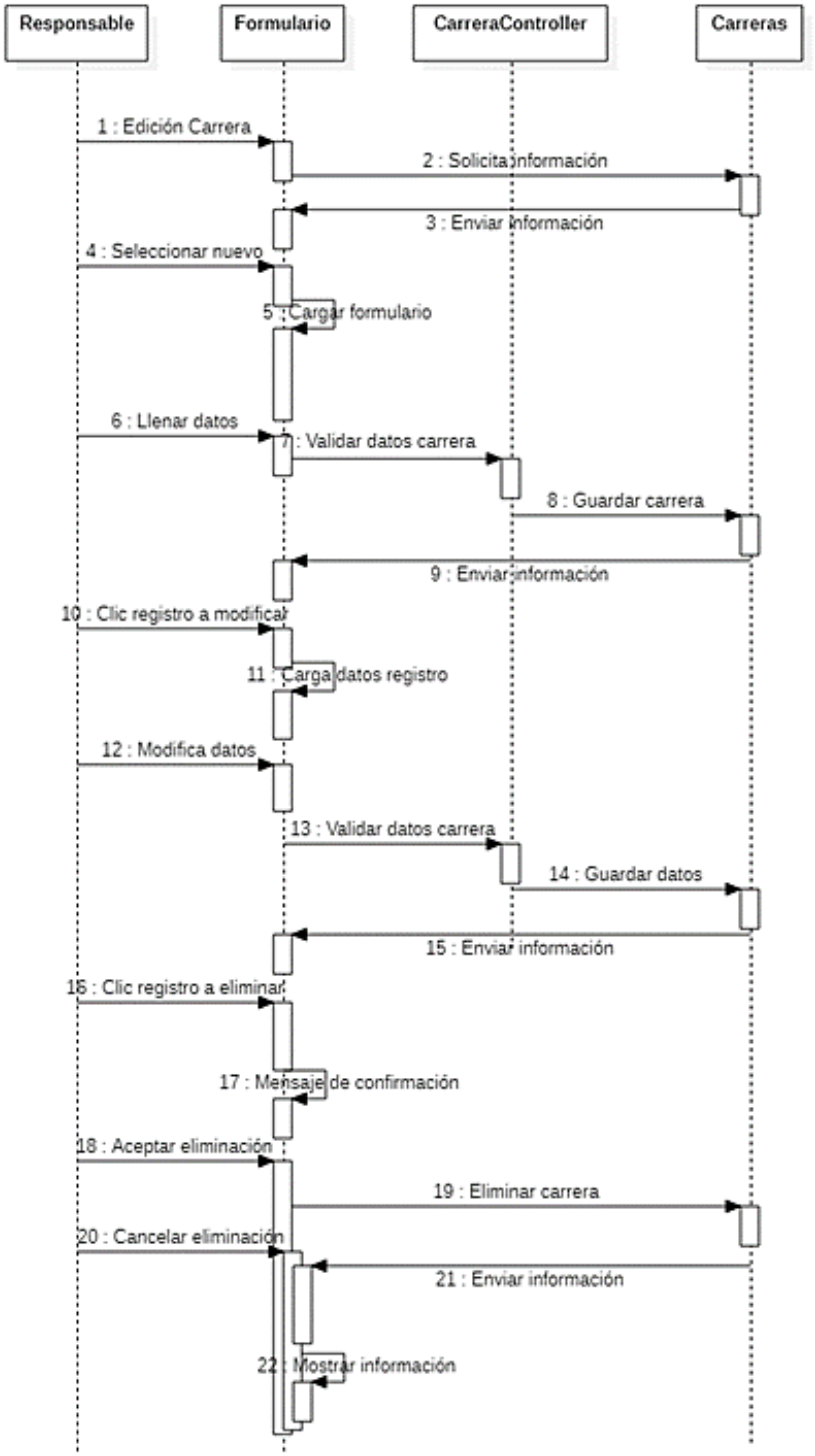


Nota. Elaboración propia.

El comportamiento de los objetos al recibir y enviar los mensajes para editar las carreras que plasmado en la figura 10.

Figura 10

Diagrama de secuencia edición carreras

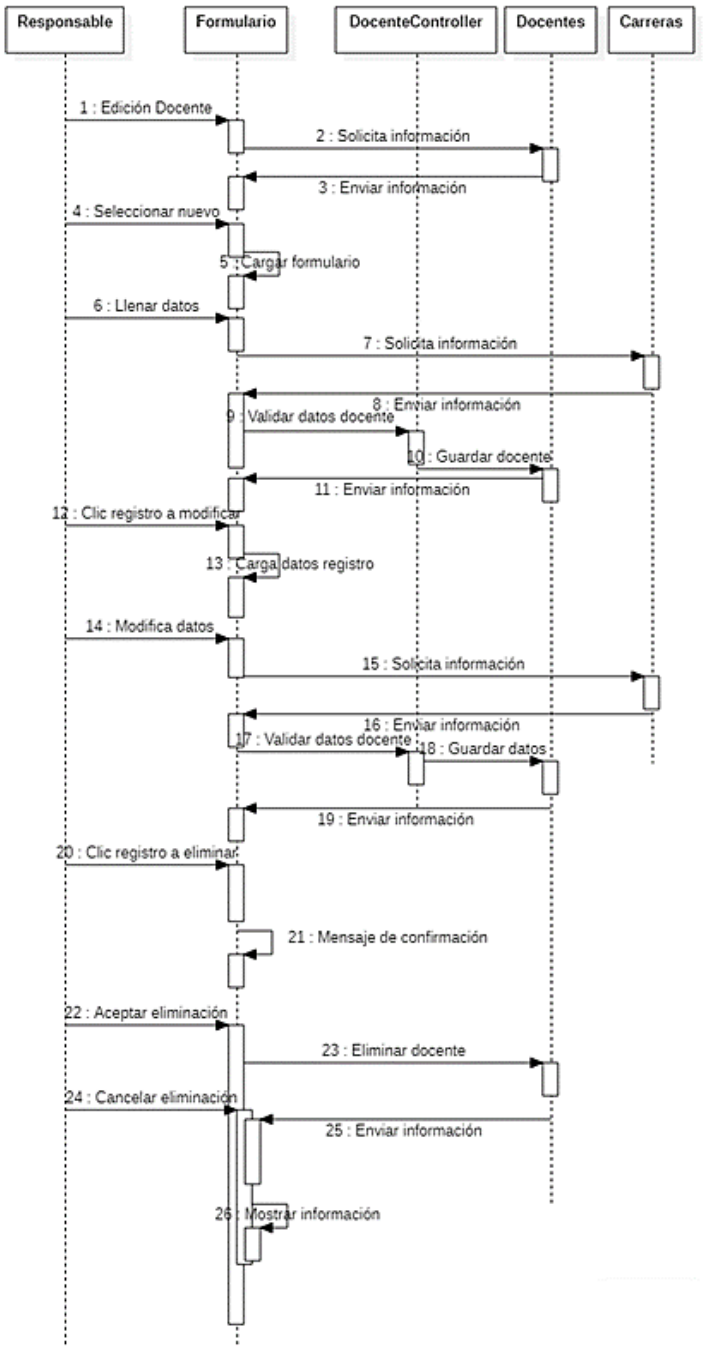


Nota. Elaboración propia.

La secuencia de los mensajes modelados en el diagrama, mostrado en la figura 11, permite realizar la edición de los datos de los docentes que están vinculados a la universidad.

Figura 11

Diagrama de secuencia edición docentes

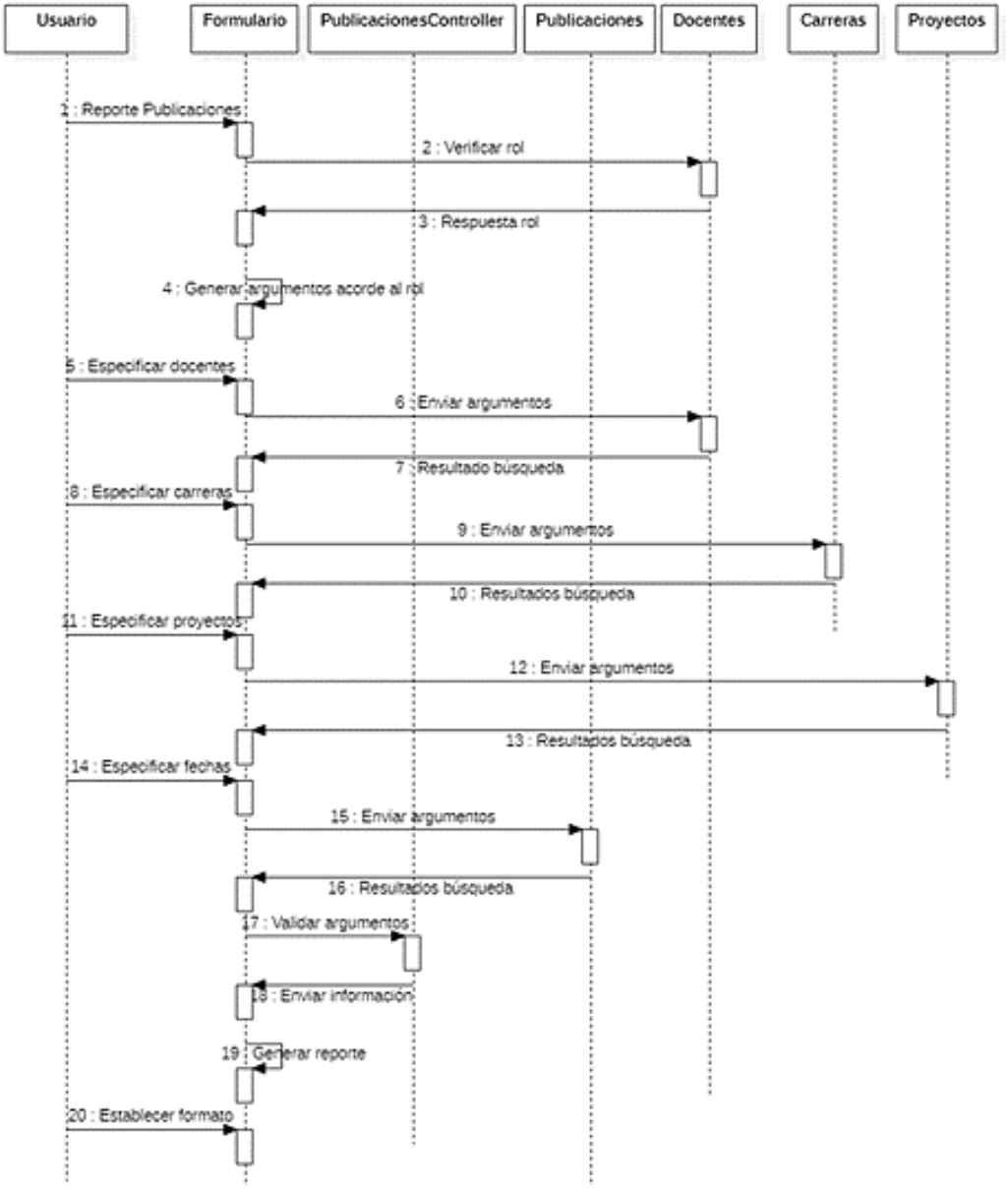


Nota. Elaboración propia.

El reporte de las publicaciones hechas por los docentes de la universidad se puede establecer con la secuencia de los mensajes modelados en el diagrama mostrado en la figura 12.

Figura 12:

Diagrama de secuencia reporte publicaciones

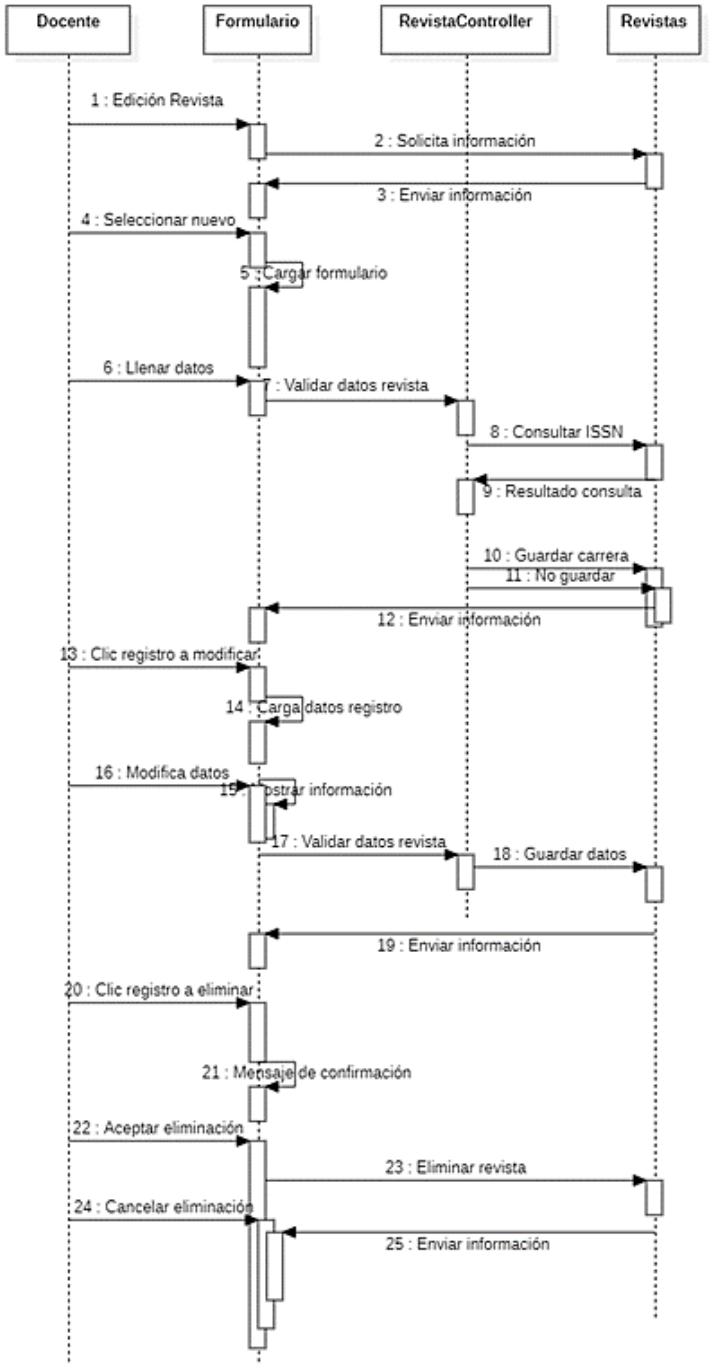


Nota. Elaboración propia.

La figura 13 muestra el diagrama de secuencia, en el cual se puede apreciar la interrelación entre los objetos y de esta forma editar los datos de las revistas en las cuales se han realizado las publicaciones de los artículos científicos.

Figura 13

Diagrama de secuencia edición revistas

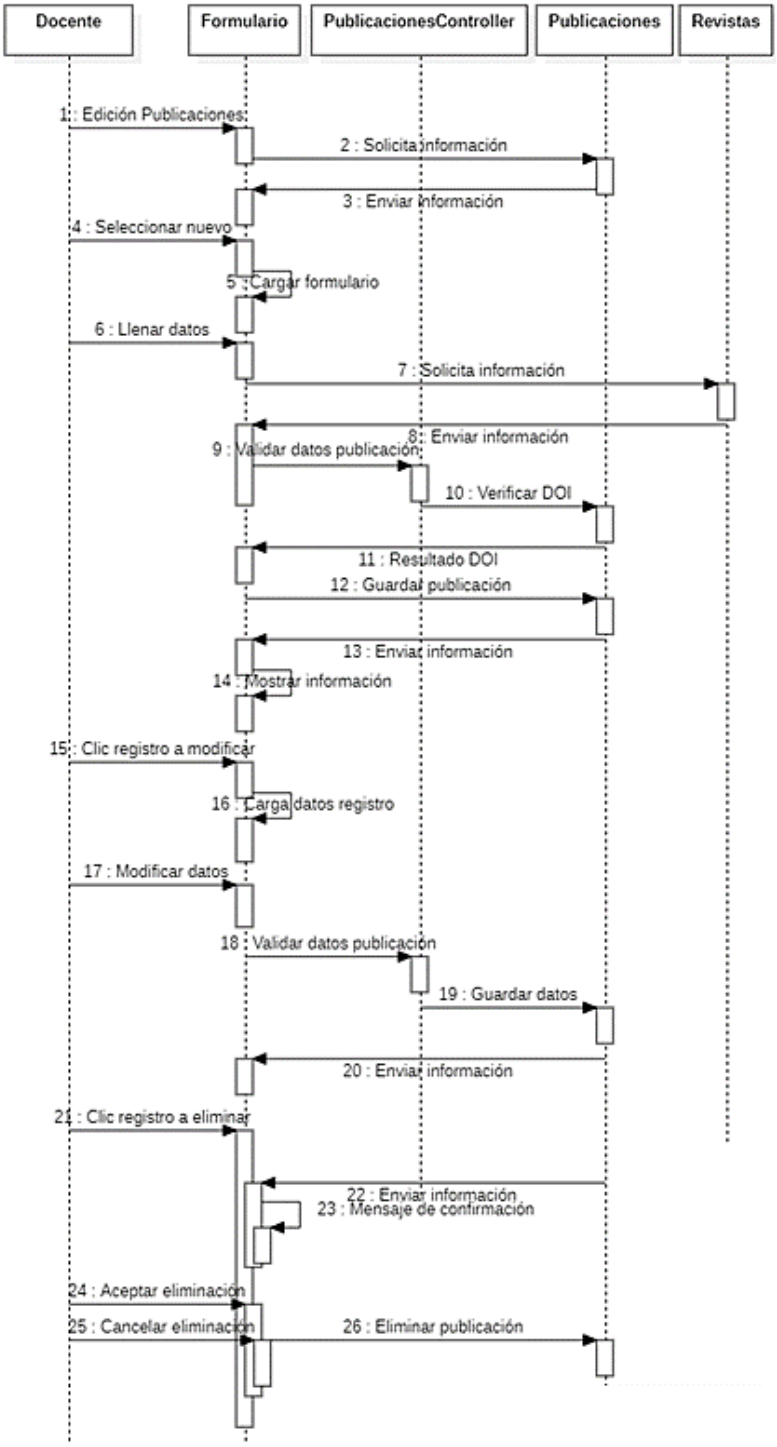


Nota. Elaboración propia.

Los mensajes emitidos y recibidos por los objetos involucrados en el proceso que permite editar las publicaciones generadas se pueden observar en la figura 14.

Figura 14

Diagrama de secuencia edición publicaciones



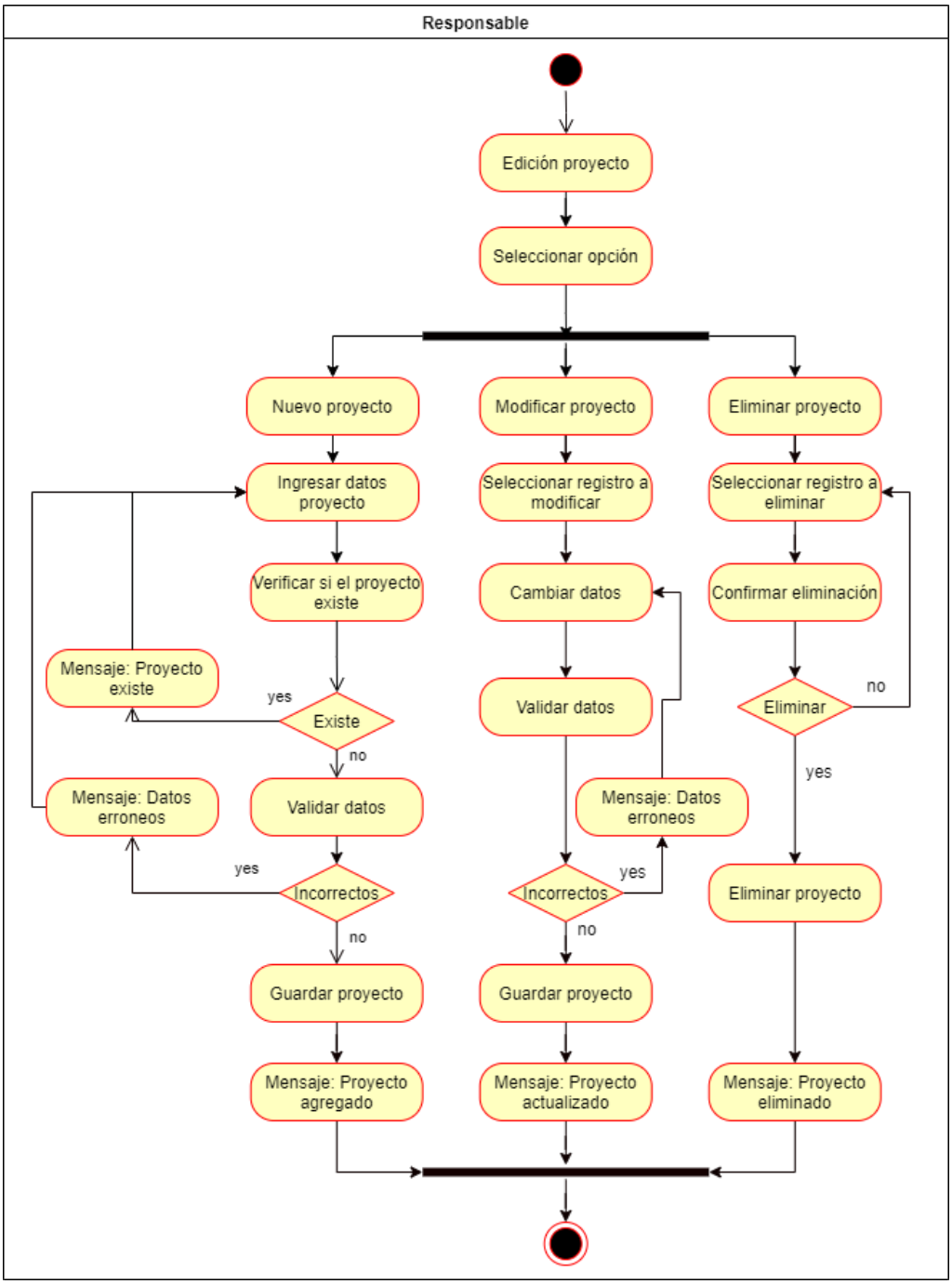
Nota. Elaboración propia.

Diagrama de actividades

Por medio de la figura 15 se puede ver el flujo que sigue las actividades ejecutadas por la aplicación para editar los datos pertenecientes a los proyectos.

Figura 15

Diagrama de actividades edición proyectos

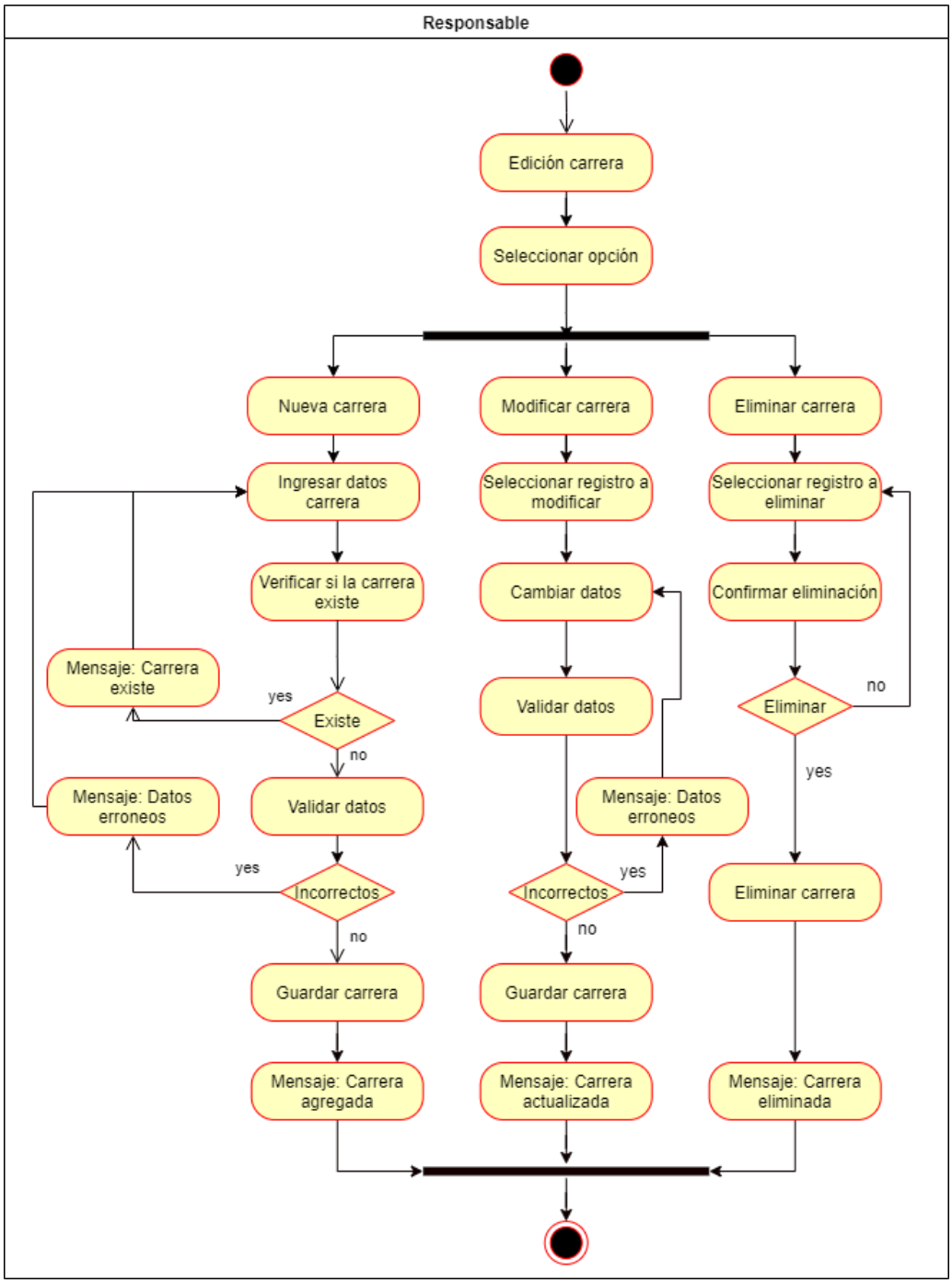


Nota. Elaboración propia.

La figura 16 permite tener una representación gráfica de los diferentes procesos que conllevan a la edición de los datos de las carreras que aportan al departamento de investigación.

Figura 16

Diagrama de actividades edición carreras

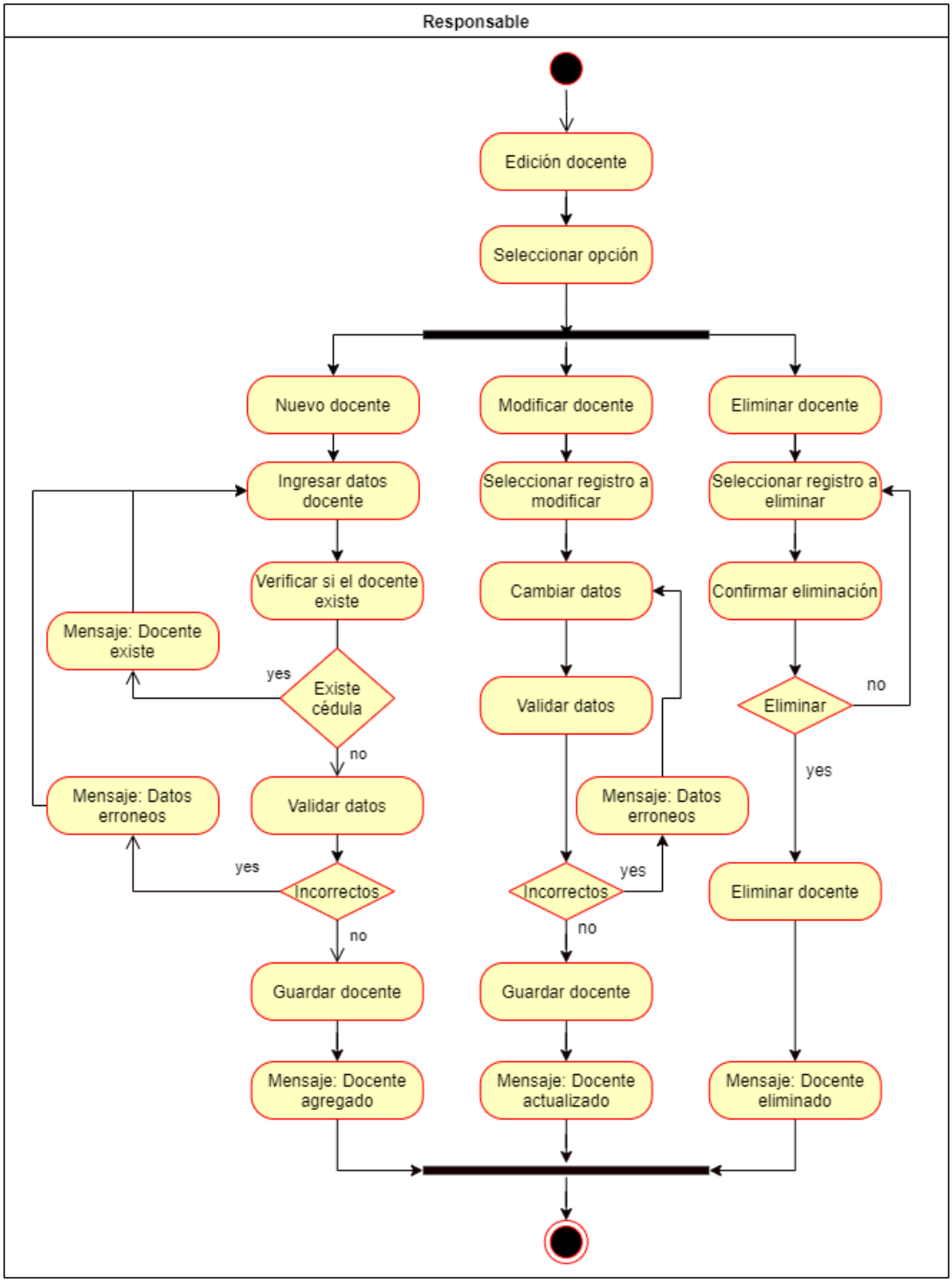


Nota. Elaboración propia.

La secuencia de las actividades está lógicamente ordenada para mostrar el trabajo que el sistema debe ejecutar para editar los datos de los docentes (ver figura 17).

Figura 17

Diagrama de actividades edición docentes

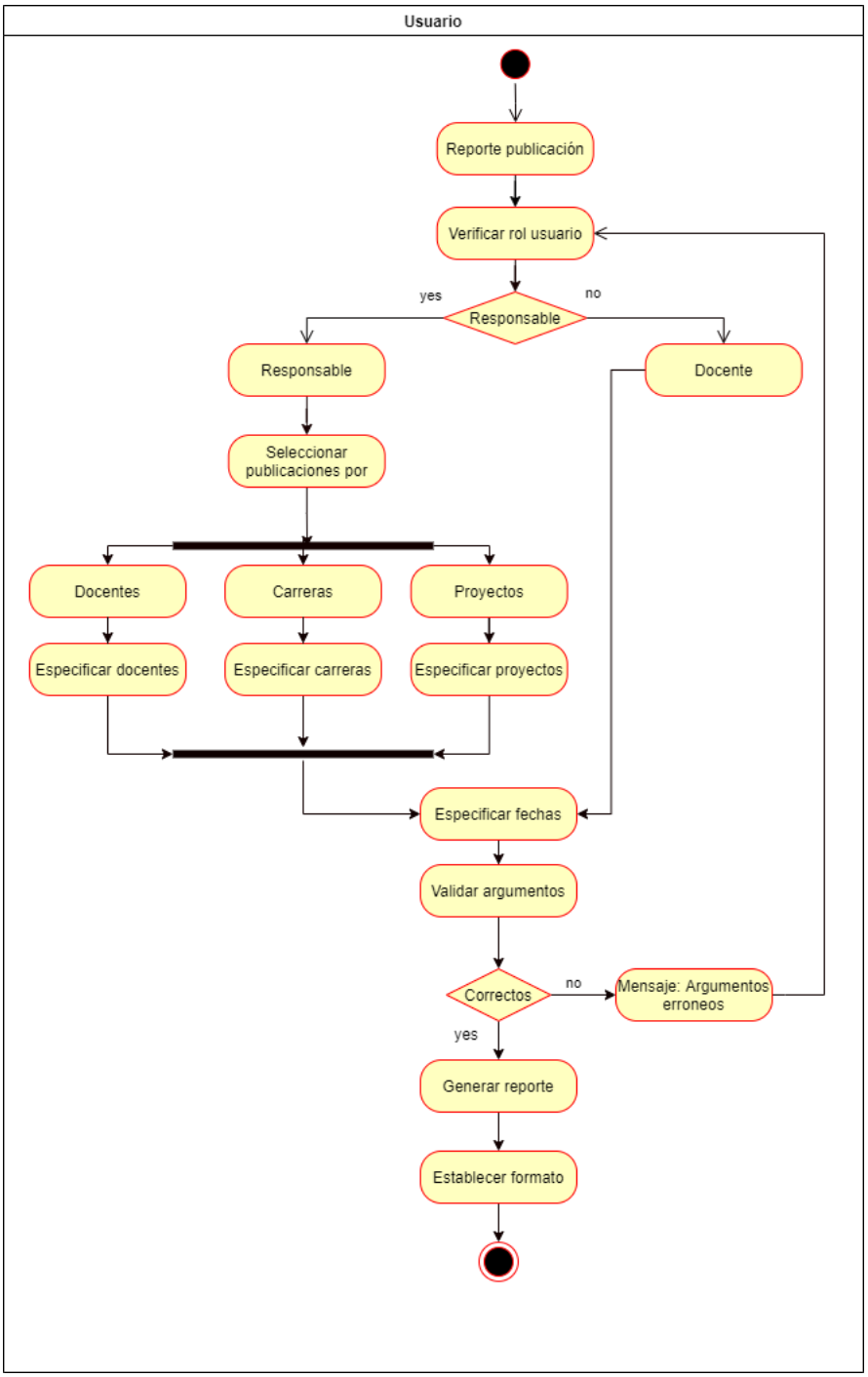


Nota. Elaboración propia.

Las actividades necesarias para crear el reporte de las publicaciones desarrolladas por la universidad están plasmadas en el diagrama expuesto en la figura 18.

Figura18

Diagrama de actividades reporte publicaciones

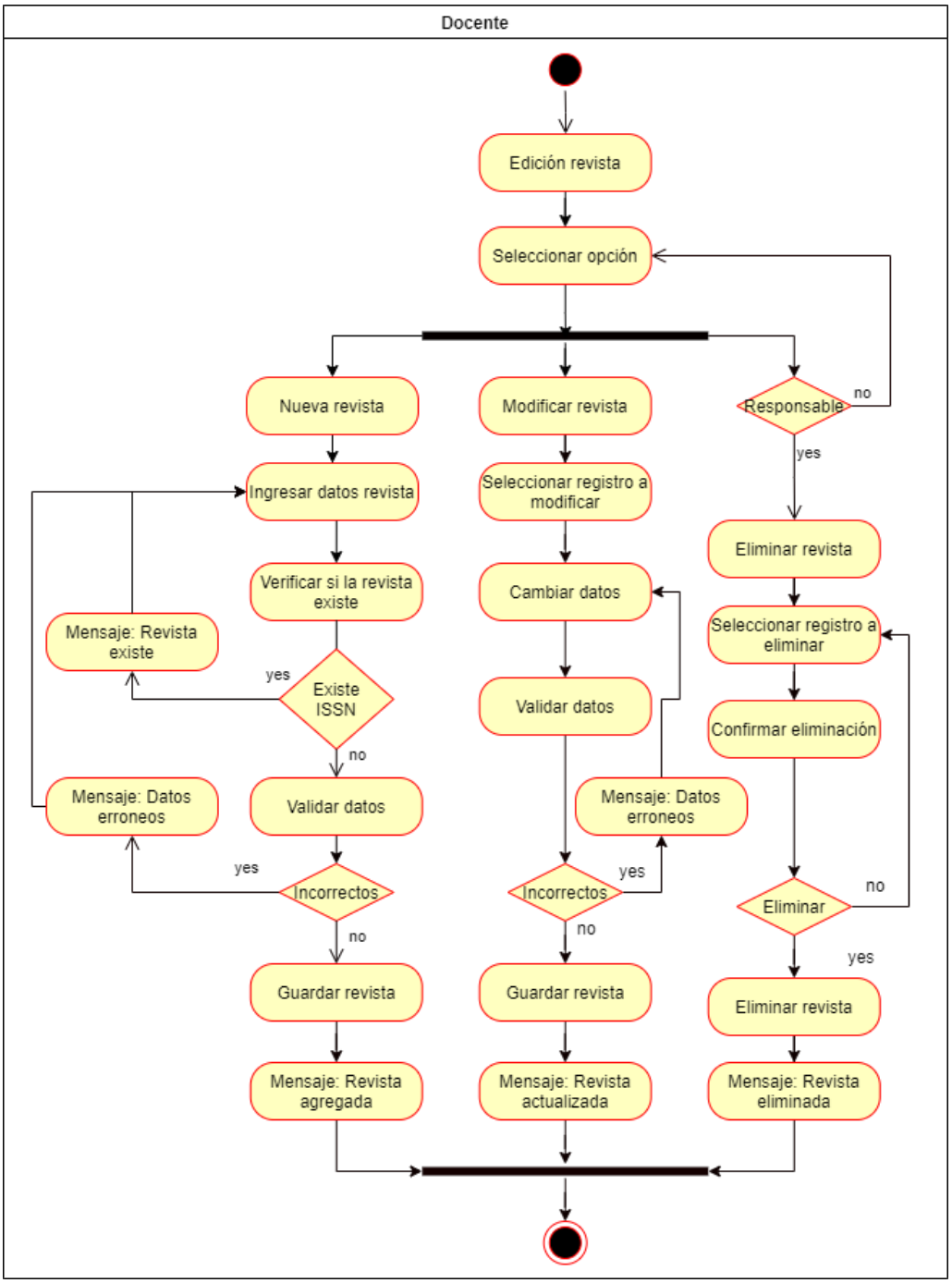


Nota. Elaboración propia.

El diagrama que permite editar los datos de las revistas se puede apreciar en la figura 19, donde se puede establecer el flujo de las actividades que se ejecutan para cumplir con esta opción.

Figura 19

Diagrama de actividades edición revistas

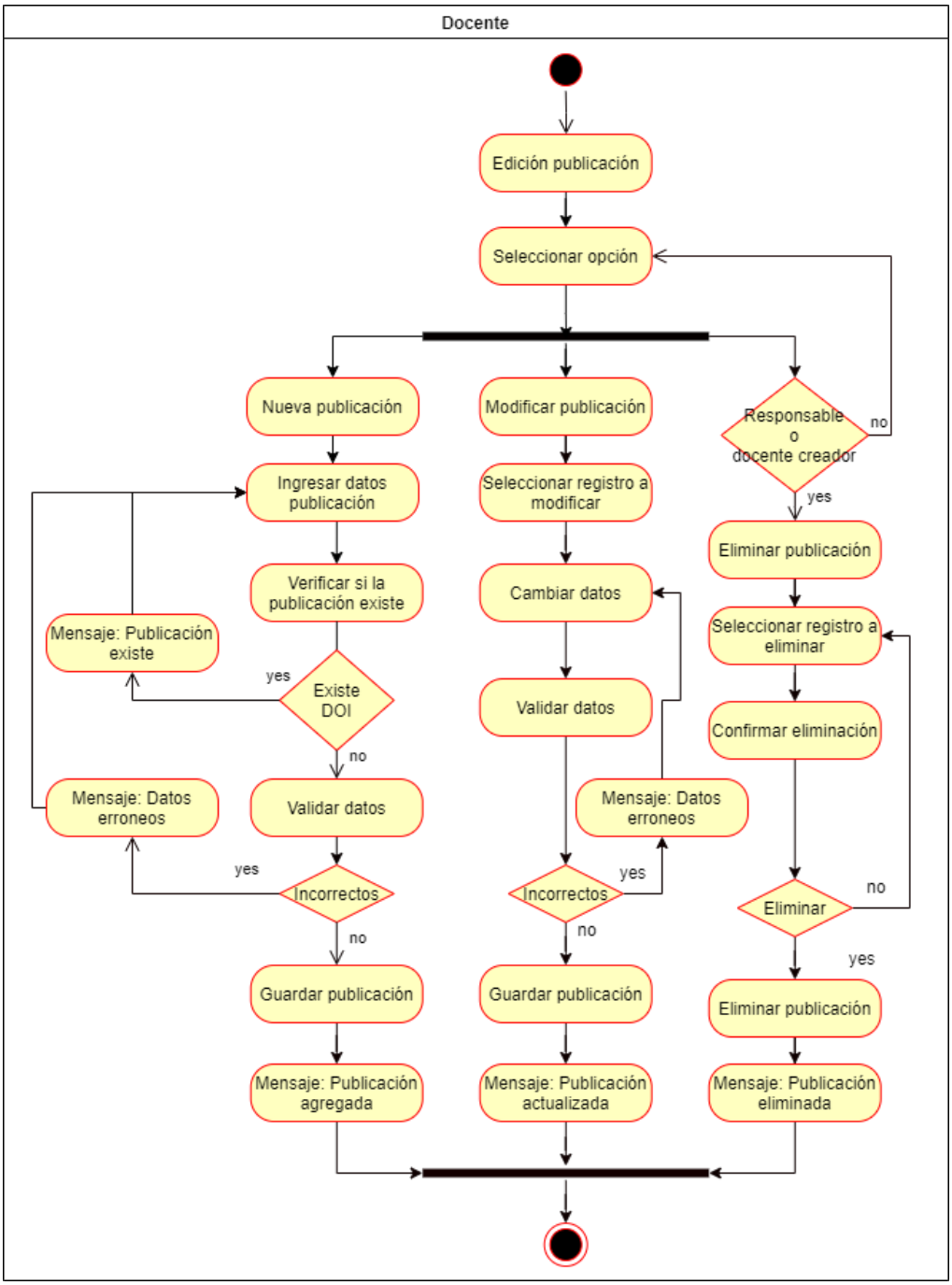


Nota. Elaboración propia.

En la figura 20 se puede ver el flujograma establecido para realizar las actividades que permitan cumplir con éxito la ejecución de la edición de los datos de las publicaciones.

Figura 20

Diagrama de actividades edición publicaciones



Nota. Elaboración propia.

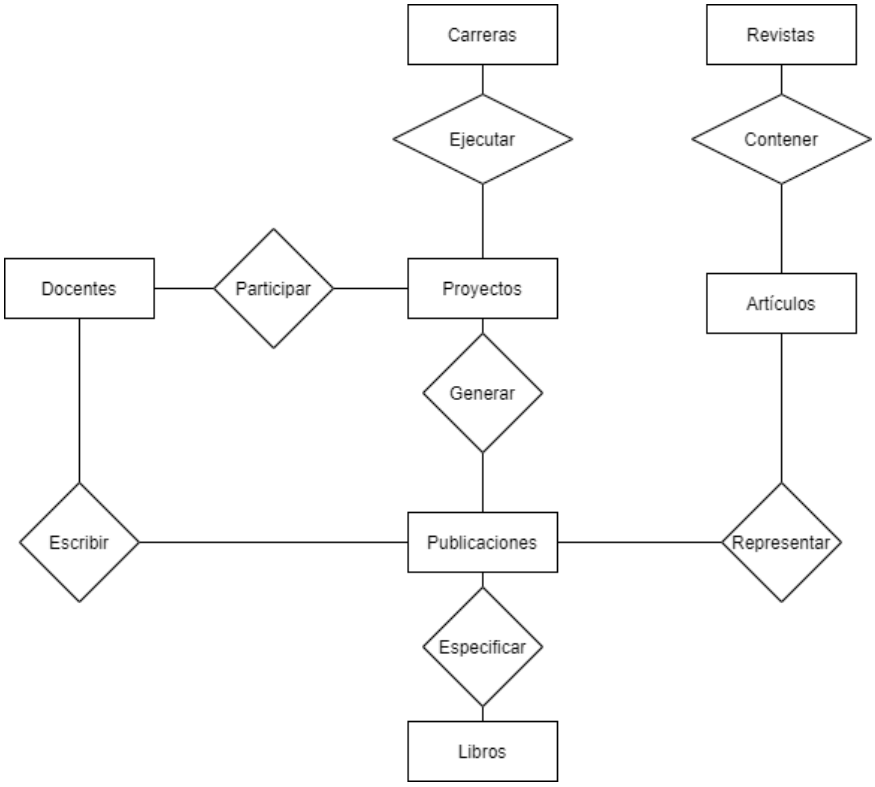
3.4. Diseño de la base de datos

3.4.1. Modelo entidad relación

El siguiente modelo (ver figura 21) permite determinar las entidades necesarias y las relaciones que existen entre ellas, de esta forma establecer una estructura que enmarca el diseño de la base de datos.

Figura 21

Modelo entidad relación



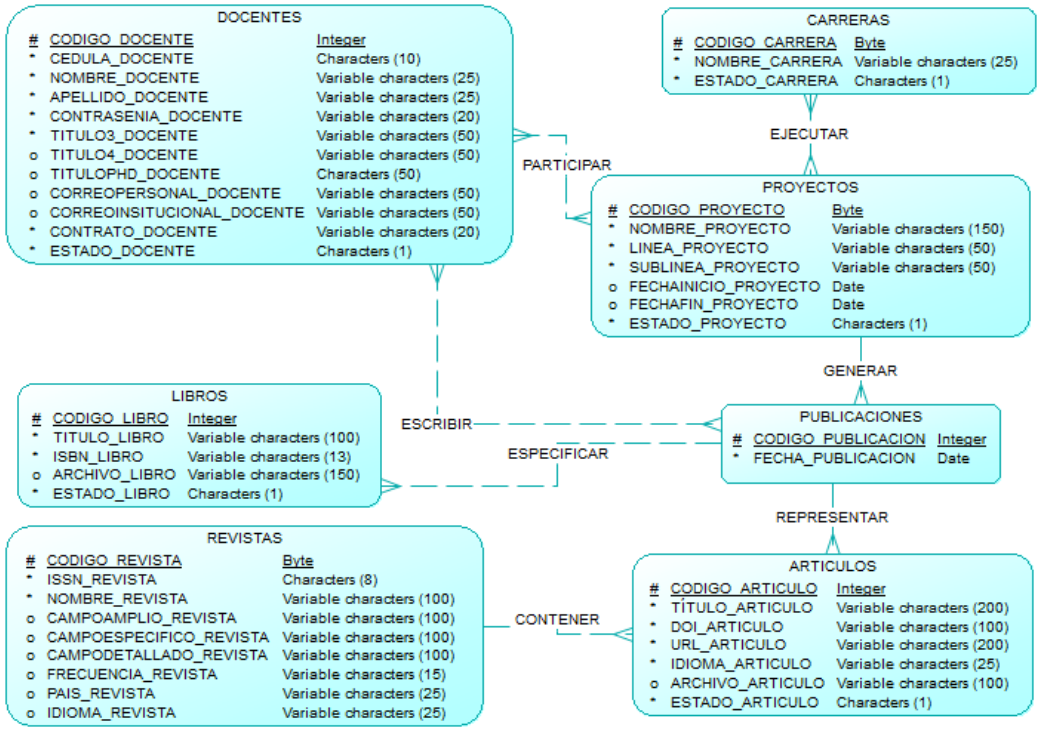
Nota. Elaboración propia.

3.4.2. Modelo conceptual

Siguiendo con el modelado de la base de datos, posterior al entidad-relación se tiene el modelo conceptual (ver figura 22), donde se especifica para cada entidad sus respectivos atributos con el tipo de dato apropiado, las claves primarias, además, de la cardinalidad presente en la relación.

Figura 22

Modelo conceptual de la base de datos



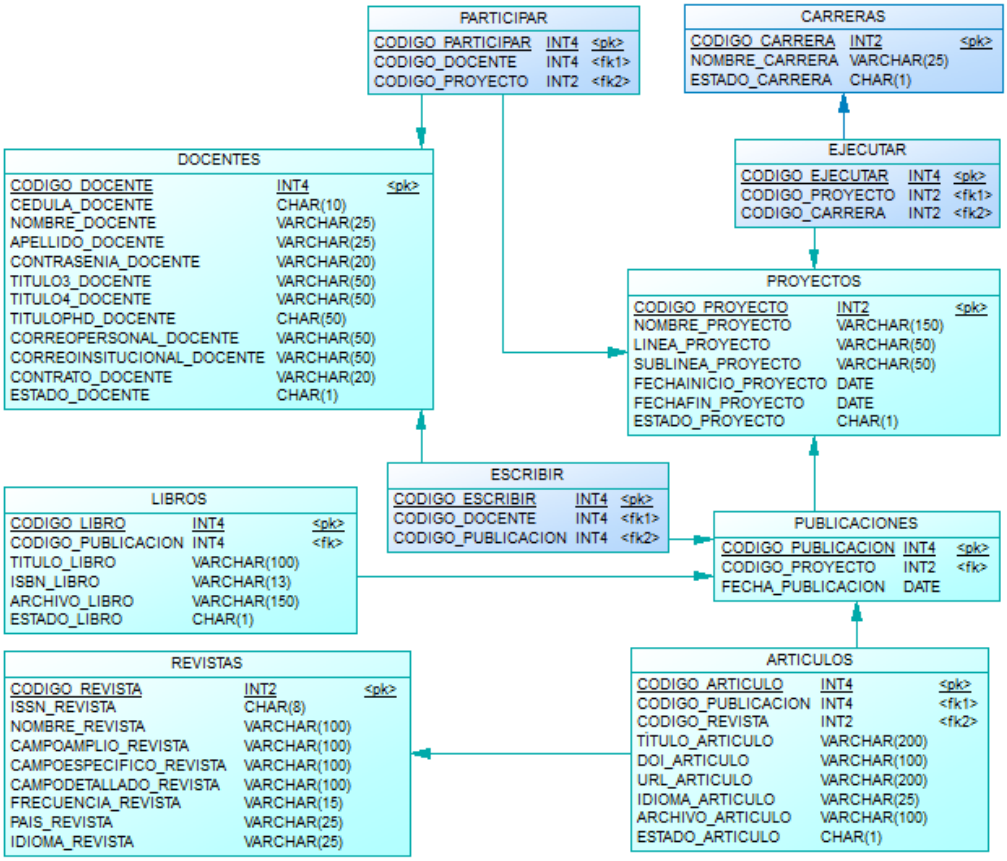
Nota. Elaboración propia.

3.4.3. Modelo físico

La figura 23 permite visualizar el modelo físico, en este modelo a más de las particularidades del diseño conceptual se definen las claves foráneas, siendo estas las que establecen la relación entre las tablas. La implementación de la base de datos es desarrollada a partir de esta estructura.

Figura 23

Modelo físico de la base datos



Nota. Elaboración propia.

Capítulo 4. Conclusiones y recomendaciones

4.1. Conclusiones

La revisión sistemática y pormenorizada de la literatura referente al desarrollo de proyectos de software contribuye de manera directa en la calidad del producto, además de dar directrices para optimizar los recursos involucrados en este proceso.

Realizar el análisis de requerimientos con base en las historias de usuario permitió conocer a detalle las necesidades del departamento de investigación con respecto al registro de las publicaciones científicas originadas en la universidad.

La aplicación de las metodologías ágiles permite cumplir con los plazos determinados aun si se presentarán cambios o imprevistos en los requerimientos inicialmente establecidos.

El marco de trabajo permite dar los lineamientos sobre las tareas que se deben realizar y los encargados de ejecutarlas para concretar con el desarrollo del *software* a medida que facilite el registro de las publicaciones científicas.

4.2. Recomendaciones

Mantener una constante actualización de conocimientos mediante la revisión de la literatura especializada en el desarrollo de software a medida.

Emplear el tiempo necesario para crear las historias de usuarios, ya que de esta forma los requerimientos serán plenamente captados y con ello minimizar cambios o imprevistos en los requerimientos.

Utilizar metodologías ágiles en la construcción de *software*, las cuales ayudan a conseguir avances paralelos en las distintas fases de su desarrollo, esto permitirá tener entregables funcionales a medida que se ejecuta el proyecto.

Implementar la aplicación informática ajustada al marco de trabajo propuesto con la finalidad de cumplir con los requerimientos analizados y considerados con el *product owner*.

Referencias bibliográficas

Avendaño, F. (2020). *Quehaceres de la investigación.* Rosario: Homo Sapiens Ediciones.

Báez, Y. (2018). *Guía para una investigación de campoo.* México: Grupo Editorial Éxodo.

Barra, A. (2019). La Importancia de la Productividad Científica en la Acreditación Institucional de Universidades Chilenas. *Formación universitaria*, 101-110. Obtenido de https://www.scielo.cl/pdf/formuniv/v12n3/0718-5006-formuniv-12-03-00101.pdf

Betancourt, A. (18 de Enero de 2021). *Las claves de Scrum con Antonio Betancourt*. Obtenido de https://www.awakelab.cl/las-claves-de-scrum/

Beynon-Davies, P. (2018). *Sistemas de información: Introducción a la informática en las organizaciones.* Barcelona: Reverté.

Bisson, A.-C., & Godoc, E. (2018). *SQL Los fundamentos del lenguaje*. Barcelona: Ediciones ENI.

CACES. (2019). *Modelo de evaluación externa de universidades y escuelas politécnicas 2019.* Quito: Consejo de Aseguramiento de la Calidad de la Educación Superior.

Camuña, J. (2014). *Lenguaje de definición y modificación de datos SQL*. Málaga: IC Editorial.

Canal, P. (10 de septiembre de 2015). *https://www.iebschool.com/blog/*. Obtenido de Definición y características de scrum master:

https://www.iebschool.com/blog/definicion-y-caracteristicas-del-scrum-master-agile-scrum/

Capacho, J., & Nieto, W. (2017). *Diseño de bases de datos.* Barranquilla: Universidad del Norte.

Gaete, J., Villarroel, R., Figueroa, I., Cornide-Reyes, H., & Muñoz, R. (2021). Enfoque de aplicación ágil con Serum, Lean y Kanban. *Ingeniare. Revista chilena de ingeniería*, 141-157.

García, J. (2018). *Aprende a modelar aplicaciones con UML.* Vigo: IT Campus Academy.

Godoy, S. (2015). *slideplayer.es*. Obtenido de https://slideplayer.es/slide/159895/

Gómez, S., & Moraleda, E. (2020). *Aproximación a la ingeniería del software.* Madrid: Centro de Estudios Ramón Areces SA.

Guerrero, M. (2014). *Base de datos relacionales y modelado de datos.* Málaga: Elearning S.L.

Hernández, R. (2014). *Metodología de la investigación.* México: McGraw Hill.

MariaDB Foundation. (13 de Noviembre de 2019). *MariaDB Server: The open source relational database*. Obtenido de https://mariadb.org/

Martín, S., & Lafuente, V. (2017). Referencias bibliográficas: indicadores para su evaluación en trabajos científicos. *Investigación Bibliotecnológica*, 151-180. Obtenido de http://www.scielo.org.mx/pdf/ib/v31n71/2448-8321-ib-31-71-00151.pdf

Martínez, D. (2019). *Manual de bases de datos.* Santiago de los Caballeros: Universidad Abierta para Adultos (UAPA).

Medina, O., Pérez, M., Toledo, F., Carrillo, A., Rosales, R., & Trujillo, J. (2019). Sistema de información para la gestión de la producción científica de la Universidad de Ciencias Médicas de Cienfuegos. *MediSur*, 764-770. Obtenido de http://scielo.sld.cu/pdf/ms/v17n6/1727-897X-ms-17-06-764.pdf

Microsoft 365 Team. (24 de septiembre de 2019). *https://www.microsoft.com/*. Obtenido de https://www.microsoft.com/es-ww/microsoft-365/business-insights-ideas: https://www.microsoft.com/es-ww/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling

Molina, T., Burbano, L., Lizcano, C., & Viteri, J. (2020). Evaluación y acreditación de las Universidades en Ecuador. Caso: Universidad Regional Autónoma de Los Andes. *Espacios*, 297 - 309. Obtenido de http://www.revistaespacios.com/a20v41n20/a20v41n20p24.pdf

Molina-Montero, B., Vite-Cevallos, H., & Dávila-Cuesta, J. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espirales recista multidisciplinaria de invesigación*, 113-121. Obtenido de https://www.researchgate.net/publication/327537074_Metodologias_agiles_frente_a_las_tradicionales_en_el_proceso_de_desarrollo_de_software

Molinero, J. (2018). *UF2218 - Desarrollo de un CMS.* Granada: Elearning, S.L.

Monroy, M., & Nava, N. (2018). *Metodología de la investigación.* México: Lapislázuli Ediciones.

Monte, J. (2016). *Implantar scrum con éxito.* Barcelona: UOC.

Muñoz, J., López, D., & Rivas, M. (2018). *La publicaciones científicas: guía de sobrevivencia cadémica.* Santiago de Chile: RIL Editores.

Noriega, R. (2017). *El proceso de desarrollo de software.* Londres: IT Campus Academy.

Piattini, M., Vizcaíno, A., & García, F. (2014). *Desarrollo global de software.* Madrid: Grupo Editorial RA-MA.

PNGWINGLE. (s.f.). *Gestión ágil de proyectos desarrollo ágil de software, gestión de proyectos, texto, logo, negocio png | PNGWing*. Obtenido de https://www.pngwing.com/es/free-png-hrgcm

Pressman, R. (2010). *Ingeniería del software: Un enfoque práctico.* México: McGraw Hill Educación.

ProyectosAgiles. (16 de 01 de 2021). *ProyectosAgiles.org*. Obtenido de Cómo funciona Scrum: https://proyectosagiles.org/como-funciona-scrum/

Pulido, E., Escobar, Ó., & Núñez, J. (2019). *Base de datos.* México: Grupo Editorial Patria.

Rad, N., & Turley, F. (2019). *Los fundamentos de agile scrum.* Ámsterdam: Van Haren.

Red Hat. (2021). *DEVOPS*. Obtenido de https://www.redhat.com/es/devops/what-is-agile-methodology

Rivas-Torres, F. (2017). La importancia de la Divulgación Científica en la investigación. *Sapienza Organizacional*, 241-244. Obtenido de https://www.redalyc.org/articulo.oa?id=553056607011

Rojas, P. (30 de Septiembre de 2021). *Blog de InGenio Learning*. Obtenido de Componentes y marco de Scrum: Elementos básicos para dominarlo: https://ingenio.edu.pe/blog/elementos-basicos-para-dominar-scrum/

Sánchez, Á. (27 de 07 de 2016). *Azul Web*. Obtenido de https://www.azulweb.net/indice-de-revistas-mexicanas-de-divulgacion-cientifica-y-tecnologica/

Sarmiento, J., Correa, C., & Jiménez, D. (2020). *Gestión de proyectos aplicada al PMBOK.* Tunja: Editorial de la Universidad Pedagógica y Tecnológica de Colombia - UTPC.

Schwaber, k., & Sutherland, J. (1 de julio de 2016). *La guía definitiva de scrum TM.* Obtenido de La guía definitiva de scrum: las reglas del juego: http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish.pdf

Tarango, J., Machin, J., & Romo, J. (2017). *Gestión de la producción y comunicación científica en instituciones de conocimiento.* Buenos Aires: Alfagrama Ediciones.

Tejada, L., & Martínez, S. (2019). *Manual de bases de datos.* Santiago de los Caballeros: Ediciones UAPA.

Trujillo, S. (2018). *UF2213 - Modelos de datos y visión conceptual de una base de datos.* Madrid: Editorial Elearning, S.L.

Vargas, E., Rengifo, R., Guizado, F., & Sánchez, F. (2019). Sistemas de información como herramienta para reorganizar procesos de manufactura. *Revista Venezolana de Gerencia*, 265-279. Obtenido de https://www.redalyc.org/journal/290/29058864015/29058864015.pdf

Vidal-Silva, C., Villarroel, R., López-Cortés, X., & Rubio, J. (2019). Una Propuesta de Algoritmo Spin / Promela para el Análisis y Diagnóstico de Errores en Diagramas de Secuencia UML. *Información Tecnológica*, 263-272. Obtenido de https://www.scielo.cl/pdf/infotec/v30n1/0718-0764-infotec-30-01-263.pdf

Wikimedia Commons. (9 de Febrero de 2019). *Archivo-Leer*. Obtenido de https://commons.wikimedia.org/wiki/File:UML_Diagrams.jpg

Zea, M., Molina, J., & Redrován, F. (2017). *Administración de bases de datos con PostgreSQL.* Alicante: 3Ciencias.

Zumba, J., & León, C. (2018). Evolución de las metodologías y modelos utilizados en el desarrollo de software. *Innova Research Journal*, 20-33. Obtenido de https://revistas.uide.edu.ec/index.php/innova/article/view/651/802

Apéndices

Apéndice 1. Matriz diseñada en Excel en la cual realizan el registro de los libros publicados por los docentes de la universidad.

UNIVERSIDAD TÉCNICA "LUIS VARGAS TORRES" DE ESMERALDAS
PROYECTO : INVESTIGACIÓN FORMATIVA CON FINES PEDAGÓGICOS Y DE SERVICIO
MATRIZ DE PRODUCCIÓN CIENTÍFICA - ACADÉMICA

N°	TÍTULO DE OBRA	TIPO OBRA	AUTOR(ES)	IDENTIFICACIÓN AUTOR(ES)	FECHA ENTREGA	CÓDIGO ISBN	REVISIÓN POR PARES	LÍNEA DE INVESTIGACIÓN	FACULTAD/ SEDE	CARRERA
1										
2										
3										
4										
5										
6										

Esmeraldas, ...(fecha)

CODIRECTOR DE FACULTAD