

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Ingeniería en Sistemas Computacionales

TEMA:

DESARROLLO DE UN SISTEMA WEB PARA LA AUTOMATIZACIÓN DEL PROCESO DE MAPEO SISTEMÁTICO DE LA LITERATURA, Y VALIDADO MEDIANTE UN MARCO DE TRABAJO DE CALIDAD DE USO BASADO EN LAS NORMAS ISO/IEC 25000 PARA MEJORAR EL PROCESO DE INVESTIGACIÓN EN LOS DOCENTES DE LA UNIVERSIDAD TÉCNICA DEL NORTE.

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO
EN SISTEMAS COMPUTACIONALES

Autor:

Edwin Alfredo Bastidas Bastidas

Director:

MSc. José Antonio Quiña Mera

Ibarra, 2020



UNIVERSIDAD TÉCNICA DEL NORTE BIBLIOTECA UNIVERSITARIA

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

1. IDENTIFICACIÓN DE LA OBRA

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

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD:	1759774589
APELLIDOS Y NOMBRES:	BASTIDAS BASTIDAS EDWIN ALFREDO
DIRECCIÓN:	IBARRA, SÁNCHEZ Y CIFUENTES 11-47 COLÓN
EMAIL:	eabastidasb@utn.edu.ec
TELÉFONO MÓVIL:	0967607610

DATOS DE LA OBRA	
TÍTULO:	DESARROLLO DE UN SISTEMA WEB PARA LA AUTOMATIZACIÓN DEL PROCESO DE MAPEO SISTEMÁTICO DE LA LITERATURA, Y VALIDADO MEDIANTE UN MARCO DE TRABAJO DE CALIDAD DE USO BASADO EN LAS NORMAS ISO/IEC 25000 PARA MEJORAR EL PROCESO DE INVESTIGACIÓN EN LOS DOCENTES DE LA UNIVERSIDAD TÉCNICA DEL NORTE.
AUTOR:	BASTIDAS BASTIDAS EDWIN ALFREDO
FECHA:	2020/02/18
PROGRAMA:	PREGRADO
TÍTULO POR EL QUE OPTA:	INGENIERO EN SISTEMAS COMPUTACIONALES
DIRECTOR:	MSc. ANTONIO QUIÑA

2. CONSTANCIAS

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

Ibarra, a los 18 días del mes de febrero del 2020

A handwritten signature in blue ink, consisting of the letters 'EAB' with a stylized flourish underneath.

Nombre: Edwin Alfredo Bastidas Bastidas

Cédula: 1759774589



UNIVERSIDAD TÉCNICA DEL NORTE FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN DEL DIRECTOR

Certifico que el trabajo de grado “DESARROLLO DE UN SISTEMA WEB PARA LA AUTOMATIZACIÓN DEL PROCESO DE MAPEO SISTEMÁTICO DE LA LITERATURA, Y VALIDADO MEDIANTE UN MARCO DE TRABAJO DE CALIDAD DE USO BASADO EN LAS NORMAS ISO/IEC 25000 PARA MEJORAR EL PROCESO DE INVESTIGACIÓN EN LOS DOCENTES DE LA UNIVERSIDAD TÉCNICA DEL NORTE.”, ha sido desarrollado en su totalidad por la señor: Edwin Alfredo Bastidas Bastidas, portadora de la cédula de identidad número 1759774589.

MSc. Antonio Quiña

DIRECTOR DE TRABAJO DE GRADO

MSc. Antonio Quiña
DOCENTE FICA UTN

Dedicatoria

Dedicado a mi mamá (Nancy) y a mi papá (Alfredo) por el esfuerzo y gran sacrificio que han realizado para que cumpla mis sueños, a mi hermana (Johana) por ser un pilar fundamental en mi vida, a mis abuelos (Regina, Aurelio, Catalina) por ser mis segundos padres, a Day y a mis amigos (Tardones) por estar conmigo durante esta etapa (la vida universitaria) y hacerla más grandiosa.

Agradecimiento

Gracias a Dios por todas las bendiciones recibidas y permitirme cumplir un sueño más.

Gracias infinitas a mi mamá (Nancy), a mi papá (Alfredo) y a mi hermana (Johana) porque sin ellos nada de esto sería posible, a mis abuelos (Regina, Aurelio, Catalina) por tomar un rol de padres cuando lo necesité.

A mi tutor MSc. Antonio Quiña, MSc. Irving Reascos y MSc. Diego Trejo por brindarme su conocimiento y experiencia. Y como no olvidar aquellos docentes de la escuela, colegio y universidad.

Tablas De Contenidos

INTRODUCCIÓN	1
Antecedentes	1
Situación actual	1
Prospectiva.....	1
Planteamiento del Problema.....	2
Objetivos.....	3
Objetivo General	3
Objetivos Específicos	3
Alcance.....	3
Justificación	5
Contexto	6
CAPÍTULO 1.....	7
Marco Teórico	7
1.1 Mapeo Sistemático de Literatura	7
1.1.1 Fase 1: Planificación de la Revisión	7
1.1.2 Fase 2: Identificación del Estudio.....	8
1.1.3 Fase 3: Extracción de Datos y Clasificación	9
1.2 Conceptualización Tecnológica	10
1.2.1 Sistema Web.....	10
1.2.2 Arquitecturas Web	12
1.2.3 Interfaz de Programación de Aplicaciones (API)	14
1.2.5 Herramientas Tecnológicas.....	16
1.3 Metodologías y Estándares.....	18
1.3.1 Estándar ISO/IEC 25000, SQuaRE (System and Software Quality Requirements and Evaluation)	18
1.3.2 Metodología Scrum.....	21
CAPÍTULO 2.....	25
Desarrollo	25
2.1 Análisis	25
2.1.1 Equipo Scrum	25
2.1.2 Definición de Requisitos	25
2.1.2 Product Backlog.....	33
2.2 Diseño – Sprint 0	34
2.2.1 Resumen Sprint 0	34
2.2.2 Arquitectura Tecnológica.....	34
2.2.3 Esquema Base de Datos Inicial	35
2.2.4 Diagrama de Procesos	35
2.3 Desarrollo del Sistema Web SMS.....	37

2.3.1 Sprint 1	38
2.3.2 Sprint 2.....	42
2.3.3 Sprint 3.....	47
2.3.4 Sprint 4.....	53
2.3.5 Sprint 5.....	57
2.4 Acta Entrega Recepción.....	62
CAPÍTULO 3.....	65
Validación de Resultados	65
3.1 Definición del Modelo de Calidad en Uso.....	65
3.2 Medición del Modelo de Calidad en Uso	66
3.2.1 Definición de la Muestra	66
3.2.2 Taller Práctico	66
3.2.3 Encuesta SUS.....	67
3.3 Evaluación Estadística de los Instrumentos de Medición	68
3.3.1 Fiabilidad del Taller.....	68
3.3.2 Fiabilidad de la Encuesta SUS.....	69
3.4 Evaluación del Modelo de Calidad en Uso	70
3.4.1 Característica: Eficacia	70
3.4.2 Característica: Eficiencia	71
3.4.3 Característica: Satisfacción	72
3.5 Resultados del Modelo de Calidad en Uso.....	73
CONCLUSIONES	77
RECOMENDACIONES	79
GLOSARIO TEMAS	80
BIBLIOGRAFÍA	81
ANEXOS	87
Anexo A. Taller Práctico.....	87
Anexo B. Instrucciones Taller	88
Anexo C. Tabulación del Taller.....	90
Anexo D. Matriz Taller (CSV).....	93
Anexo E. Algoritmo Alfa de Cronbach	94
Anexo F. Tabulación Encuesta SUS	95
Anexo G. Análisis Factorial Confirmatorio	97

Índice de Figuras

Fig. 1: Árbol de problemas.....	2
Fig. 2: Proceso mapeo sistemático de literatura.....	3
Fig. 3: Módulos y funcionalidades del sistema web.....	4
Fig. 4: Arquitectura tecnológica propuesta.....	5
Fig. 5: Proceso de estudios de mapeo sistemáticos.....	7
Fig. 6: Fase planificación de la revisión.....	7
Fig. 7: Fase identificación del estudio.....	8
Fig. 8: Fase extracción de datos y clasificación.....	9
Fig. 9: Esquema básico de una aplicación web.....	10
Fig. 10: Arquitectura monolítica.....	12
Fig. 11: Arquitectura en capas.....	13
Fig. 12: Fracción de código GraphQL.....	16
Fig. 13: Divisiones de las normas SQuaRE.....	19
Fig. 14: Escala de medición.....	21
Fig. 15: Ciclo de vida Scrum.....	22
Fig. 16: Estructura capítulo 2.....	25
Fig. 17: Arquitectura tecnológica.....	35
Fig. 18: Esquema inicial de Base de Datos.....	35
Fig. 19: Diagrama de procesos de la aplicación.....	36
Fig. 20: Vista para crear usuario.....	40
Fig. 21: Vista para actualizar usuario.....	40
Fig. 22: Vista listar usuarios.....	41
Fig. 23: Vista para crear base de datos bibliográfica.....	41
Fig. 24: Vista para actualizar base de datos bibliográfica.....	41
Fig. 25: Vista para listar las base de datos bibliográficas.....	41
Fig. 26: Vista para el registro del investigador.....	42
Fig. 27: Vista para la autenticación.....	44
Fig. 28: Vista para crear SMS.....	44
Fig. 29: Vista para actualizar SMS.....	45
Fig. 30: Vista para listar SMS.....	45
Fig. 31: Vista para seleccionar base de datos bibliográfica.....	45
Fig. 32: Vista para deseleccionar y listar base de datos bibliográficas seleccionadas.....	45
Fig. 33: Vista para crear pregunta.....	46
Fig. 34: Vista para actualizar pregunta.....	46
Fig. 35: Vista para listar preguntas.....	46
Fig. 36: Vista para crear búsqueda de estudios.....	49
Fig. 37: Vista para actualizar búsqueda de estudios.....	50
Fig. 38: Vista para listar búsquedas de estudios.....	50
Fig. 39: Vista para crear primer filtro.....	51
Fig. 40: Vista para actualizar primer filtro.....	51
Fig. 41: Vista para listar los primeros filtros.....	51
Fig. 42: Vista para crear filtro de estudios.....	52
Fig. 43: Vista para editar filtro de estudios.....	52
Fig. 44: Vista para listar los filtros de estudios.....	52
Fig. 45: Vista para crear búsqueda profunda.....	55
Fig. 46: Vista para actualizar búsqueda profunda.....	55

Fig. 47: Vista para listar búsquedas profundas.....	56
Fig. 48: Vista para importar estudios.....	56
Fig. 49: Vista para crear campos personalizados.....	56
Fig. 50: Vista para actualizar campo personalizado.....	56
Fig. 51: Vista para listar los campos personalizados.....	57
Fig. 52: Vista para listar los estudios.....	59
Fig. 53: Vista para completar la información.....	60
Fig. 54: Vista para exportar CSV.....	60
Fig. 55: Ejemplo archivo CSV exportado.....	60
Fig. 56: Lista de los SMS para generar reporte.....	61
Fig. 57: Breve resumen del proceso SMS.....	61
Fig. 58: Ejemplo de reporte generado – Parte 1.....	61
Fig. 59: Ejemplo de reporte generado - Parte 2.....	62
Fig. 60: Estructura capítulo 3.....	65
Fig. 61: Alfa de Cronbach del taller.....	69
Fig. 62: Correlación de las preguntas.....	69
Fig. 63: Resultado en la escala de medición.....	75

Índice de Tablas

Tabla 1: Modelo de calidad en uso	20
Tabla 2: Roles del proyecto	25
Tabla 3: Historia de Usuario 1	26
Tabla 4: Historia de Usuario 2	27
Tabla 5: Historia de Usuario 3	27
Tabla 6: Historia de Usuario 4	28
Tabla 7: Historia de Usuario 5	28
Tabla 8: Historia de Usuario 6	29
Tabla 9: Historia de Usuario 7	29
Tabla 10: Historia de Usuario 8	30
Tabla 11: Historia de Usuario 9	30
Tabla 12: Historia de Usuario 10	31
Tabla 13: Historia de Usuario 11	31
Tabla 14: Historia de Usuario 12	32
Tabla 15: Historia de Usuario 13	32
Tabla 16: Historia de Usuario 14	32
Tabla 17: Historia de Usuario 15	33
Tabla 18: Product Backlog.....	33
Tabla 19: Tareas - Sprint 0.....	34
Tabla 20: Descripción del diagrama de procesos de la aplicación.....	36
Tabla 21: Resumen Sprints	38
Tabla 22: Sprint Backlog - Sprint 1	38
Tabla 23: Pruebas de aceptación Sprint 1	39
Tabla 24: Plan de mejora.....	42
Tabla 25: Sprint Backlog - Sprint 2.....	43
Tabla 26: Pruebas de aceptación Sprint 2	44
Tabla 27: Plan de mejora.....	47
Tabla 28: Sprint Backlog - Sprint 3.....	47
Tabla 29: Pruebas de aceptación Sprint 3	48
Tabla 30: Plan de mejora.....	53
Tabla 31: Sprint Backlog - Sprint 4.....	53
Tabla 32: Pruebas de aceptación Sprint 4	54
Tabla 33: Plan de mejora.....	57
Tabla 34: Sprint Backlog - Sprint 5.....	58
Tabla 35: Pruebas de aceptación Sprint 5	58
Tabla 36: Plan de mejora.....	62
Tabla 37: Definición modelo de calidad en uso	65
Tabla 38: Objetivos y tareas del taller	66
Tabla 39: Peso respuestas	72
Tabla 40: Resultados SUS - Utilidad	72
Tabla 41: Resultados SUS - Comodidad	73
Tabla 42: Resultados calidad en uso	74

Resumen

Se realizó un marco de teórico del proceso de mapeo sistemático de literatura y la tecnología usada, la cual sirvió como base para realizar el desarrollo del sistema web para la automatización del mapeo sistemático (SMS); el SMS cuenta con tres fases que son: (i) planificación de la revisión, (ii) identificación del estudio, (iii) extracción de datos y clasificación.

El sistema web para la automatización del SMS permite realizar SMS de manera organizada y llevar acabo cada una de sus fases y generar un *dataset* de estudios para su análisis y un reporte del proceso realizado. El sistema web tiene una arquitectura orientada a microservicios, utilizando las tecnologías Apollo, Express, GraphQL, Neo4j, React, Bootstrap y la API Mendelej. Se utilizó Scrum como una metodología ágil en el desarrollo realizando un total de 15 historias de usuario y 5 Sprints cada uno con una duración de 42 horas excepto el Sprint 5 que tiene 32 horas.

Después del desarrollo se realizó una evaluación con un marco de trabajo de calidad en uso basado en las normas ISO/IEC 25000, el cual se obtuvo como resultado 84,11% de calidad en uso, lo que quiere decir que el uso del sistema web es satisfactorio.

Palabras clave: Mapeo Sistemático, SMS, GraphQL, Microservicios, Scrum.

Abstract

A theoretical framework of the systematic mapping process of literature and the technology used was made, which served as the basis for developing the web system for the automation of systematic mapping study (SMS); The SMS has three phases that are: (i) review planning, (ii) study identification, (iii) data extraction and classification.

The web system for SMS automation allows do SMS in an organized manner and to carry out each of its phases and generate a set of data for its analysis and a report of the process carried out. The web system has a microservice-oriented architecture, using Apollo, Express, GraphQL, Neo4j, React, Bootstrap and Mendeley API technologies. Scrum was used as an agile methodology in the development, there was of a total of 15 user stories and 5 Sprints and each with a duration of 42 hours except the Sprint 5 that was 32 hours.

After the development, an evaluation was carried out with a quality framework in use based on the ISO/IEC 25000 standards, which resulted in 84,11% quality in use, which means that the use of the web system It is satisfactory.

Keywords: Systematic Mapping, SMS, GraphQL, Microservices, Scrum.

INTRODUCCIÓN

Antecedentes

El mapeo sistemático de literatura o Systematic Mapping Study (SMS) es una metodología utilizada en el campo de la investigación para elaborar esquemas de clasificación y estructurar un campo de interés con resultados (K. Petersen, Feldt, R. Mujtaba, & Mattsson, 2008). Con el fin de establecer una línea base de conocimiento en cualquier tema, la información encontrada se puede combinar para responder preguntas específicas realizadas por los investigadores (Macchi & Solari, 2012). El uso de esta metodología hace que el análisis y seguimiento de revisiones bibliográficas sean más eficientes, sin embargo, no existe una aplicación web que automatice el proceso del mapeo sistemático de literatura.

En la Universidad Técnica del Norte cuenta con acceso a bases de datos bibliográficas como: Scopus, Science Direct, Springer, IEEE, entre otras; donde los docentes investigadores de la Universidad pueden encontrar artículos científicos para realizar sus investigaciones, sin embargo, la gran cantidad de informes y resultados disponibles que van en aumento, por lo cual es importante resumir y tener una visión general (K. Petersen et al., 2008) mediante la aplicación del mapeo sistemático como metodología.

Situación actual

Al momento de realizar una investigación los docentes de investigación de la Universidad Técnica del Norte deben explorar diferentes bases de datos bibliográficas las cuales no están integradas y se debe realizar la misma búsqueda en cada una de estas, por lo cual este proceso toma más tiempo y esfuerzos de investigación.

Actualmente los investigadores cuando buscan los artículos científicos (*papers*) leen el título, el *abstract* y a veces otras partes de estos; la idea es tener una imagen amplia e identificar fácilmente las brechas de investigación (Galindo, Benavides, Trinidad, Gutiérrez-Fernández, & Ruiz-Cortés, 2018) pero no existe una aplicación web para los docentes de la Universidad Técnica del Norte que permita realizar un SMS. Dado este panorama se plantea automatizar este proceso mediante una aplicación web para que los docentes investigadores de la Universidad Técnica del Norte realicen la organización de sus investigaciones de manera eficiente aplicando la metodología SMS y así puedan reducir tiempo y esfuerzo empleado en sus proyectos.

Prospectiva

Existe un gran número de artículos científicos en las diferentes base de datos bibliográficas, y es importante hacer un resumen y tener una visión general del tema de

investigación (K. Petersen et al., 2008) para que el docente investigador de la Universidad Técnica del Norte pueda abstraer la información que necesita.

Con el fin de mejorar el proceso de mapeo sistemático de literatura se planteó el proyecto de automatización, con esto se obtiene una caracterización y clasificación de artículos más heterogéneos (Galindo et al., 2018), que permita una manera más fácil el estudio de los resultados.

Planteamiento del Problema

En la literatura de investigación se puede encontrar estudios primarios y secundarios, los primarios son nuevas investigaciones y originales, y los secundarios son el resumen del estado actual de una investigación; este último requiere búsquedas exhaustivas de las investigaciones publicadas (Jalali & Wohlin, 2012). Con el fin de tener una organización más eficiente de los estudios y reducir los tiempos empleados, además de tener una visión general, se ha pensado en la automatización del proceso de mapeo sistemático de literatura mediante una aplicación web para los docentes de investigación de la Universidad Técnica del Norte, ver Fig. 1: Árbol de problemas.

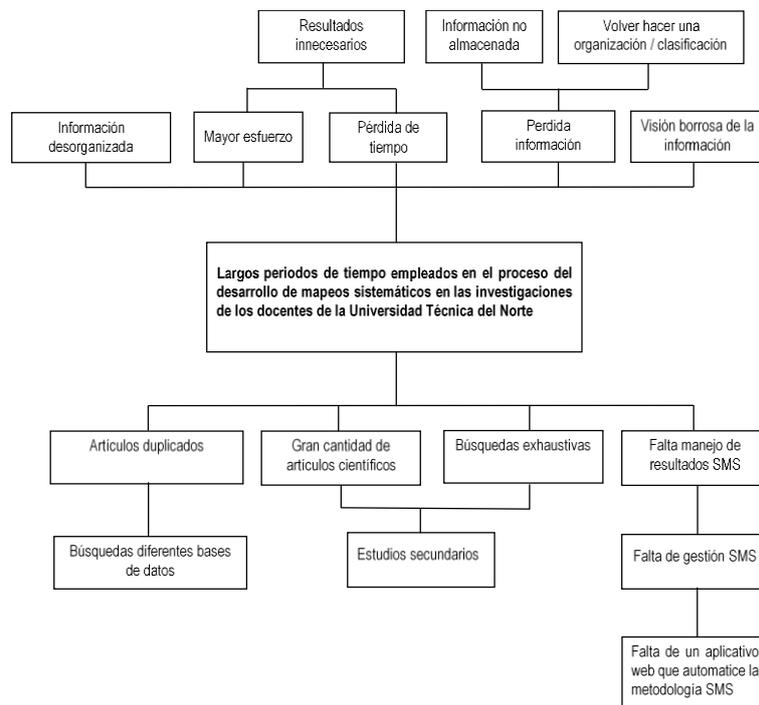


Fig. 1: Árbol de problemas
Fuente: Propia

Objetivos

Objetivo General

Desarrollar un sistema web para la automatización del proceso de mapeo sistemático de literatura, y validado mediante un marco de trabajo de calidad de uso basado en las normas ISO/IEC 25000 para mejorar el proceso de investigación en los docentes de la Universidad Técnica del Norte.

Objetivos Específicos

- Elaborar un marco teórico del proceso de mapeo sistemático de literatura y la tecnología usada.
- Desarrollar un sistema web que automatice el proceso de mapeo sistemático de literatura.
- Evaluar la aplicación web mediante un marco de trabajo de calidad en uso basado en las normas ISO/IEC 25000.
- Validar resultados.

Alcance

El proyecto planteado tiene la finalidad automatizar el proceso de mapeo sistemático de literatura mediante un aplicativo web para los docentes de investigación de la Universidad Técnica del Norte, y así mejorar el tiempo empleado en las investigaciones. Los SMS proporcionan salidas visuales como tabulaciones y gráficos que permiten una fácil gestión del proceso de revisión bibliográfica de la literatura (Galindo et al., 2018), ver Fig. 2.

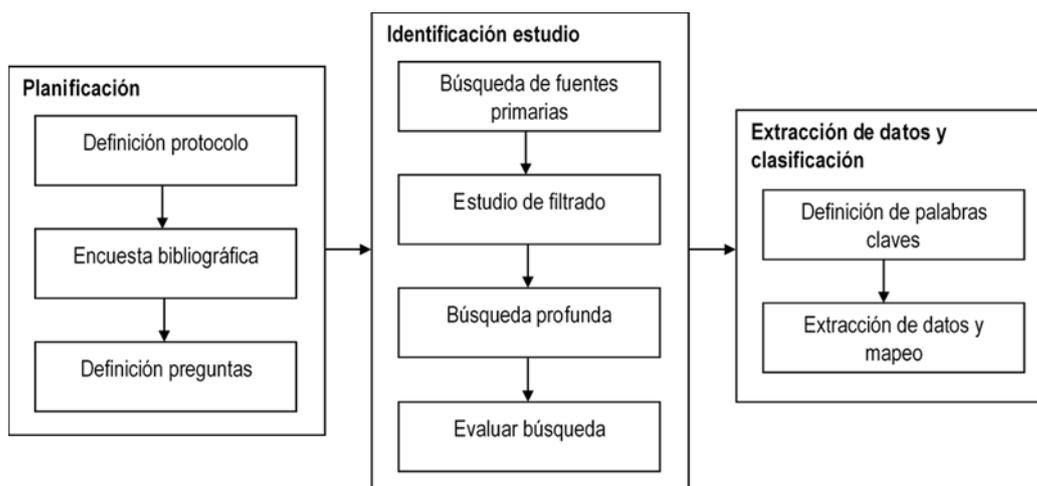


Fig. 2: Proceso mapeo sistemático de literatura
Fuente: (Galindo et al., 2018).

El proyecto contará con una aplicación web que tendrá los siguientes módulos (ver Fig. 3):

Módulo	Funcionalidades	Descripción	
Administración SMS	Gestión de los SMS	CRUD de los SMS's	
Planificación SMS	Registro protocolo	Se define las pautas para realizar el mapeo sistemático	Establecer el alcance del estudio del SMS
	Registro encuestas bibliográficas	Establecer en que bases de datos se va a realizar las búsquedas	
	Registro preguntas de investigación	Definir preguntas de investigación, según el modelo 5W+1H	
Identificación de estudio	Registro de las búsquedas	Registrar los resultados de las búsquedas apoyándonos en la API REST de Mendeley	Selección de los estudios primarios que se aplicarán en el SMS
	Filtrado	Registrar de criterios de inclusión y exclusión	
	Registro de búsqueda profunda	Agregar más estudios	
	Evaluación de la búsqueda	Obtención final de fuentes primarias	
Extracción de datos y clasificación	Registro de palabras claves	Agrupación y clasificación de los trabajos	Despliegue de reportes para responder a las preguntas planteadas en la planificación del SMS
	Extracción de datos y mapeo	Reportes en tablas que responden las preguntas planteadas	

Fig. 3: Módulos y funcionalidades del sistema web
Fuente: Propia

- Además, el proyecto contará con una API GraphQL para la gestión de los SMS. Las tecnologías que se emplearán en el desarrollo de la aplicación serán (ver Fig. 4):
- Servidor GraphQL Express
 - Node.js
 - Base de datos Neo4j
 - Apollo Client
 - React
 - Bootstrap

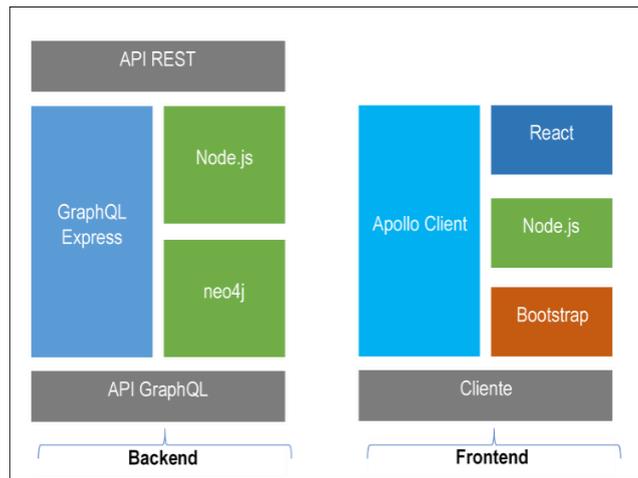


Fig. 4: Arquitectura tecnológica propuesta
Fuente: Propia

En el desarrollo se utilizará la metodología SCRUM.

Al finalizar la aplicación se hará una validación mediante un marco de trabajo de calidad de uso con las normas ISO/IEC 25000 y además se validará los resultados.

Justificación

La falta de un aplicativo web que automatice el proceso de SMS y permita a los docentes investigadores de la Universidad Técnica del Norte organizar de manera eficiente la información encontrada en sus estudios justifica el desarrollo de este proyecto.

El proyecto se enfoca al objetivo 9 Industria, Innovación e Infraestructura de los Objetivos de Desarrollo Sostenible:

9.b Apoyar el desarrollo de tecnologías, la investigación y la innovación nacionales en los países en desarrollo, incluso garantizando un entorno normativo propicio a la diversificación industrial y la adición de valor a los productos básicos, entre otras cosas (Naciones Unidas, 2016).

9.c Aumentar significativamente el acceso a la tecnología de la información y las comunicaciones y esforzarse por proporcionar acceso universal y asequible a Internet en los países menos adelantados de aquí a 2020 (Naciones Unidas, 2016).

Este proyecto tiene un impacto social puesto que permite tener una idea e identificar fácilmente las brechas de investigación para entender fácilmente el estado actual y futuro de un determinado tema (Galindo et al., 2018) y así poder profundizar en dichos temas y aportar a la comunidad científica.

Se pensó desarrollar una API GraphQL, porque proporciona una solución a ciertas limitaciones e ineficiencias experimentadas por desarrolladores que interactúan con API REST (Ghebremicael, 2017). GraphQL fue creado por Facebook en 2012 y a partir del 2015 comenzó a ser de código abierto (Facebook, 2018); es una tecnología API del futuro por las

ventajas que tiene, los diferentes beneficios que aporta a los desarrolladores y mejora los flujos de trabajo lo cual hace que cambie la manera en cómo se crea y se consume las API (Prisma, 2018).

Contexto

En el repositorio digital de la Universidad Técnica del Norte no existe estudios o proyectos donde hacen referencia a la automatización de mapeo sistemático de literatura.

A nivel nacional existen varios trabajos de titulación y revistas donde hace uso de la metodología pero no utilizan una aplicación web que automatice el procesos de SMS; una de las revistas es de la Escuela Superior Politécnica de Chimborazo donde utiliza como metodología el mapeo sistemático de literatura, la revista se titula "MAPEO SISTEMATICO DE LITERATURA DE UN DATA LAKE" (Aucancela, Naranjo, Vizúete, & Betún, 2018). Un trabajo similar en la Universidad de las Fuerzas Armadas ESPE donde aplica un mapeo sistemático de literatura a un estudio, el trabajo de titulación tiene el título: "ESTUDIO EMPÍRICO DE LOS FACTORES TECNOLÓGICOS QUE INCIDEN EN LA PRODUCCIÓN JUST IN TIME DE LAS ROSAS MEDIANTE LA APLICACIÓN DE UN SYSTEMATIC MAPPING STUDY" (CHUQUIMARCA ARANHA, 2017).

CAPÍTULO 1

Marco Teórico

1.1 Mapeo Sistemático de Literatura

Los estudios de mapeo sistemáticos o SMS por sus siglas en inglés (Systematic Mapping Studies) permiten tener una visión general de un área de investigación mediante una clasificación (Kai Petersen, Vakkalanka, & Kuzniarz, 2015). El SMS se enfoca en realizar una búsqueda en la literatura y conocer los temas que se han cubierto y donde se han publicado (K. Petersen et al., 2008). Las fases principales del SMS son: planificación de la revisión, identificación del estudio y extracción de datos y clasificación (Galindo et al., 2018), ver Fig. 5.

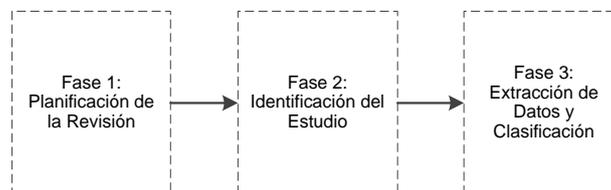


Fig. 5: Proceso de estudios de mapeo sistemáticos
Fuente: (Galindo et al., 2018)

Un SMS permite satisfacer a un conjunto de preguntas de investigación, facilitando un instrumento imparcial, objetivo y sistemático a través del análisis de todos los aportes de investigación importantes en un área de estudio específico (Francesco, Malavolta, & Lago, 2017).

1.1.1 Fase 1: Planificación de la Revisión

En esta fase se cumplen tres pasos: (i) definición del protocolo, (ii) encuesta bibliográfica y (iii) definición de las preguntas de investigación, que consiste en establecer las preguntas a responder (Galindo et al., 2018), ver Fig. 6.

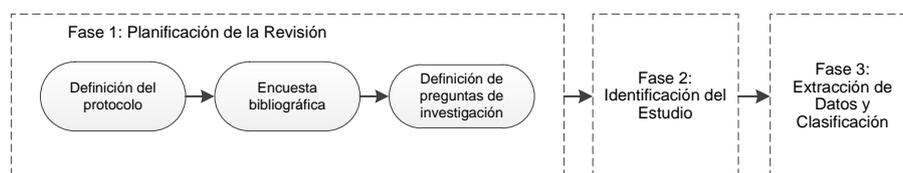


Fig. 6: Fase planificación de la revisión
Fuente: (Galindo et al., 2018)

- Definición del Protocolo

Donde se decide como realizar la revisión y como reducir las amenazas a la validez. El protocolo para el proceso del SMS es adaptado de (K. Petersen et al., 2008) y (Galindo et al., 2018).

- Encuesta Bibliográfica

Se consigue la base de conocimiento del área de investigación a estudiar (Galindo et al., 2018).

- Definición de las Preguntas de Investigación

Las preguntas de investigación (*Research Question - RQ*) están relacionadas con los objetivos de los SMS, el principal objetivo de un SMS es tener una visión de un área de investigación e identificar el número y el tipo de investigación y los resultados disponibles sobre ella; otro de los propósitos es reconocer en que foros se ha publicado (K. Petersen et al., 2008). Existe una propuesta la cual se basa en el modelo 5W+1H: *Who, Why, What, Where, When* y *How* (Quién, Por qué, Qué, Dónde, Cuándo y Cómo) (Hart, 1996; Pan & Kosicki, 1993). Este modelo permite a las personas con poca experiencia hacer preguntas iniciales de investigación.

1.1.2 Fase 2: Identificación del Estudio

En esta fase se realiza (i) búsqueda de fuentes primarias, (ii) estudio de filtrado, (iii) búsqueda profunda y (iv) evaluación de la búsqueda, ver Fig. 7.

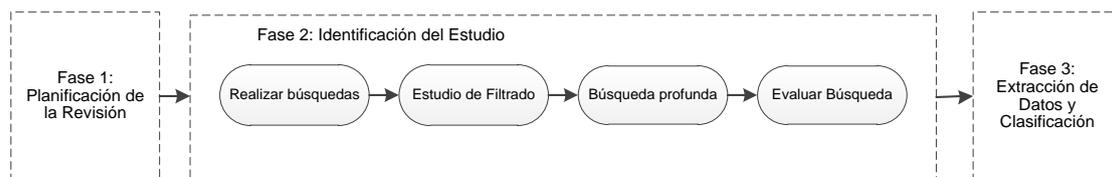


Fig. 7: Fase identificación del estudio
Fuente: (Galindo et al., 2018)

- Realizar Búsqueda de Fuentes Primarias

Los estudios primarios (investigaciones nuevas y originales) investigan una pregunta de investigación específica, para encontrar estos estudios se busca en las bases de datos científicas utilizando cadenas de búsqueda o navegando manualmente en actas de congresos importantes o publicaciones de revistas (K. Petersen et al., 2008).

Para identificar palabras clave y estructurar cadenas de búsquedas basadas en las preguntas de investigación, se recomienda utilizar PICO (Población, Intervención,

Comparación y Resultados) propuesto por Kitchenham y Charters (Kitchenham & Charters, 2007).

En población se define el área que se va a estudiar; en intervención lo que se va a analizar, por ejemplo: metodología, herramienta o procedimiento; en comparación se realiza comparaciones de intervenciones para identificar las mejores opciones; y en resultados (*outcomes*) los resultados esperados.

- Estudio de Filtrado

Se establece pautas de inclusión y exclusión para eliminar estudios que no son relevantes para las preguntas de investigación (K. Petersen et al., 2008).

- Búsqueda Profunda

Para evitar perder trabajos importantes se realiza nuevas búsquedas por ejemplo, con la técnica de muestreo bola de nieve (K. Petersen et al., 2008).

El muestreo bola de nieve es una técnica que se utilizada para identificar nuevos sujetos, en este caso serían nuevos estudios. Esta técnica se define en tres pasos: (i) Hacer búsquedas en las principales revistas y / o en los actas de congresos para obtener un conjunto inicial de artículos, (ii) ir hacia atrás revisando las listas de referencias de los artículos relevantes que se encuentra en el paso uno y dos (iterar hasta que no se identifiquen nuevos trabajos), y (iii) continuar identificando los artículos, citando los que han sido identificados en los pasos anteriores (Jalali & Wohlin, 2012).

- Evaluar Búsqueda

Además de las pautas de inclusión y exclusión, se debe evaluar la calidad de los estudios primarios, para lo cual se detalla más criterios de inclusión y exclusión (Kitchenham & Charters, 2007).

1.1.3 Fase 3: Extracción de Datos y Clasificación

Comprende dos pasos (i) temas de palabras clave; (ii) extracción de datos y mapeo, ver Fig. 8.

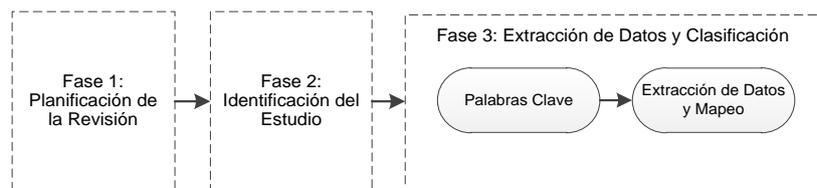


Fig. 8: Fase extracción de datos y clasificación
Fuente: (Galindo et al., 2018)

- Palabras Clave

Las *Keywording*¹ es una forma de reducir el tiempo para realizar el esquema de clasificación y garantizar que tenga en cuenta los estudios existentes. Primero, los lectores leen resúmenes y buscan palabras clave y conceptos que reflejen la contribución del *paper*²; cuando los resúmenes son de baja calidad pueden leer la introducción o la conclusión del documento. Hecho esto, el conjunto de palabras clave de diferentes artículos se combinan para desarrollar un alto nivel comprensión de la investigación. Las palabras clave se pueden agrupar y categorizar el mapa (K. Petersen et al., 2008).

- Extracción de Datos y Mapeo

Para extraer datos de los estudios primarios identificados se crea una tabla con los campos que contienen cada categoría de la clasificación (Kai Petersen et al., 2015).

El resultado de los estudios de mapeos sistemáticos de la literatura se concentran en mostrar las frecuencias de las investigaciones por categoría generales o específicas de un tema de investigación, con lo que se puede observar de una manera estructurada como se han distribuido las publicaciones de un tema específico, y así también responder las preguntas de investigación planteadas por el investigador (K. Petersen et al., 2008).

1.2 Conceptualización Tecnológica

1.2.1 Sistema Web

Sistema web o aplicación web es un tipo de software que se ejecuta en un navegador web conectado a una red; las aplicaciones web tienen una comunicación del tipo cliente/servidor mediante un protocolo HTTP, por sus siglas en inglés *Hypertext Transfer Protocol*, es decir Protocolo de Transferencia de Hipertexto (Neosoft, 2018), ver Fig. 9.

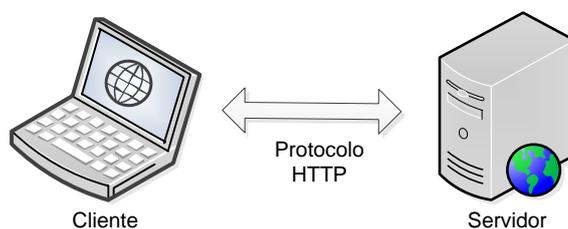


Fig. 9: Esquema básico de una aplicación web
Fuente: (Neosoft, 2018)

¹ Palabras clave.

² Artículo científico publicado en una revista especializada.

Las aplicaciones web se han convertido en un pilar importante para la sociedad actual, estas aplicaciones proporcionan diversos servicios, desde comercio electrónico hasta banca en línea (Mesbah, 2015).

- Protocolo HTTP

HTTP pertenece a los protocolos de comunicación TCP/IP utilizados en Internet. Facilita la comunicación con el cliente y el servidor (Luján Mora, 2001), basado en el esquema solicitud/respuesta, el cliente solicita un recurso al servidor y este envía una respuesta. El protocolo HTTP puede transportar hojas de estilo en cascada (CS), applets de Java, páginas de servidor activas (ASP), scripts de Perl, entre otros lenguajes y herramientas utilizadas en los servidores web (Goralski & Goralski, 2017).

- El Cliente

El cliente web es un software, normalmente un navegador web donde el usuario puede interactuar y solicitar al servidor web el envío de recursos mediante el protocolo HTTP. Las páginas web están formadas de código HTML por sus siglas en inglés *HyperText Markup Language*, es decir Lenguaje de Marcas de Hipertexto y algunas partes de scripts como JavaScript³ (Roldán Martínez, Valderas Aranda, & Pastor López, 2010) Tim Berners-Lee desarrolló HTML en 1989, con el objetivo de mostrar información en las páginas web, este código contiene etiquetas que son leídas por el navegador una vez se procesa la página y utiliza esta información para definir como se mostrará el contenido. (Harrington, Cross, Harrington, & Cross, 2015).

- El Servidor

El servidor web es un software que está recibiendo peticiones constantemente mediante el protocolo HTTP por parte de los clientes (Berenguel Gómez, 2015). Comúnmente la parte del servidor de la aplicación está formado por: (i) páginas estáticas (HTML), que muestran el mismo contenido; (ii) recursos como multimedia (por ejemplo, imágenes, videos, audios, etc.), documentos, etc. los cuales se pueden utilizar en la página mismo o descargar para su utilización en el cliente; (iii) scripts que son ejecutados por el servidor cuando el cliente hace alguna petición, la respuesta a esta petición suele ser una página HTML que se envía al cliente (Luján Mora, 2001).

³ Lenguaje de programación imperativo más omnipresentes en la actualidad en el mundo de la computación, basado en objetos.

1.2.2 Arquitecturas Web

Las organizaciones empresariales han evolucionado en los diferentes enfoques arquitectónicos. Hasta los años 80, las estructuras eran mayormente verticales y aisladas hasta que, en los años 90 optaron por modelos horizontales centrados en los procesos empresariales. Ahora necesitan entregar continuamente diferentes servicios. Esta necesidad lleva a la migración de la arquitectura monolítica a la Arquitectura Orientada a Servicios (*Service Oriented Architecture* - SOA) y a los Microservicios (Endrei & IBM., 2004).

- Arquitectura Monolítica

En la arquitectura monolítica la interfaz de usuario, la lógica de negocio y los datos se combinan a la vez, ver Fig. 10. Cuando la aplicación y el equipo de trabajo crece, en una arquitectura como ésta, es difícil para los integrantes del equipo entender y modificar el código y sobre todo la dificultad aumenta para los nuevos miembros (Seda et al., 2018). A estos inconvenientes se le suman: (i) dificultad en el despliegue continuo: en una aplicación monolítica grande es un obstáculo realizar despliegues continuos. Para actualizar un componente o módulo, es debe desplegar toda la aplicación; (ii) problemas de escalabilidad: los componentes en esta arquitectura tienen diferentes requisitos de recursos y no se puede escalar individualmente cada componente. Esta arquitectura es escalable solo en una dimensión (agregando recursos: CPU, RAM, etc.); (iii) retrasa el proceso de desarrollo: si el código base se vuelve demasiado grande, el Entorno de Desarrollo Integrado (IDE) se hace más lento y ocasiona que los desarrolladores sean menos productivos; (iv) imposible vender módulos por separado: no se puede separar las funcionalidades en módulos y vender a terceros. Si desea vender debe ser toda la aplicación lo cual resulta costoso (Seda et al., 2018).

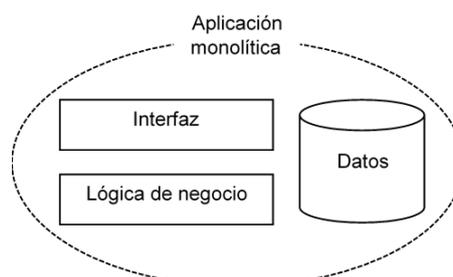


Fig. 10: Arquitectura monolítica
Fuente: (Seda et al., 2018)

- Arquitectura en Capas

En esta arquitectura existen módulos separados que se comportan como si estuvieran en la misma capa lo cual es beneficioso cuando el sistema tiene diferentes niveles de

funcionalidad. El principal beneficio de esta arquitectura en capas es que los diferentes niveles podrían distribuirse, implementarse de manera fácil e independiente. (Seda et al., 2018). Las capas que tiene esta arquitectura comúnmente son (desde el nivel inferior): capa de acceso a datos, capa de lógica de negocios también llamada capa de servicio, capa de presentación, capa de servicio web y capa de interfaz gráfica de usuario (Seda et al., 2018), ver Fig. 11.

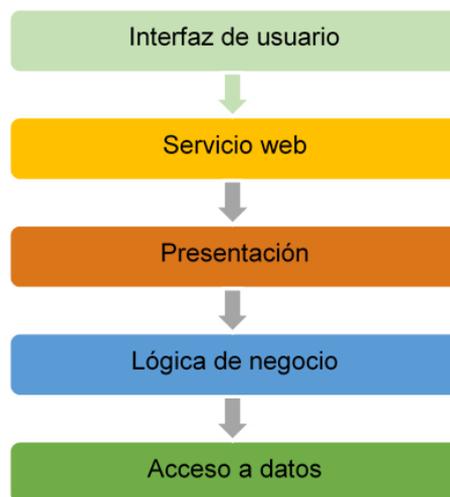


Fig. 11: Arquitectura en capas
Fuente: (Seda et al., 2018)

La arquitectura en capas básicamente está conformada por tres grupos: (i) arquitectura de un nivel, (ii) arquitectura de dos niveles y (iii) arquitectura de tres niveles o más.

La arquitectura de un nivel es la más simple, todas las capas necesarias se encuentran en la misma máquina. Es fácil de diseñar, pero es la menos escalable. Las ventajas de esta arquitectura son: (i) se requiere poco mantenimiento para el desarrollo en el nivel del cliente, (ii) fácil de actualizar (Seda et al., 2018).

En arquitectura de dos niveles existe una comunicación entre el cliente y servidor, por ejemplo, el cliente (navegador web) realiza una petición al servidor, éste procesa la solicitud y retorna una respuesta (página web). Estos enfoques permitieron mejorar la escalabilidad en la forma de dividir la vista del usuario de las capas de datos. Las ventajas de esta arquitectura son: (i) un mejor intercambio de carga, (ii) se realiza poco trabajo en el cliente, (iii) la presentación puede ser personalizada para cada cliente (Seda et al., 2018).

En la arquitectura de tres niveles o más se utiliza en el desarrollo de aplicaciones web, esta arquitectura consta de tres o más capas, las más comunes son: (i) presentación, (ii) lógica de negocio y (iii) capa de datos. Una de las principales ventajas de esta arquitectura es la facilidad con la que se puede reemplazar las capas y se pueden dividir las capas del servidor (Seda et al., 2018).

- Arquitectura Orientada a Servicios (SOA)

SOA es una arquitectura propuesta por *International Business Machines* (IBM). Esta arquitectura presenta un enfoque para construcción de sistemas distribuidos que ofrecen funcionalidades como servicios a aplicaciones de usuario final u otros servicios (Endrei & IBM., 2004). Para la implementación de SOA, por lo general se utilizan servicios web basados en WSDL y en el protocolo SOAP (Seda et al., 2018).

Los servicios web son aplicaciones empresariales autodescriptivas y modulares que exportan la lógica empresarial como servicio por medio de Internet para ser consumidos por los clientes (Wagh & Thool, 2012).

WSDL siglas de *Web Services Description Language* (Lenguaje de Descripción de Servicios Web) describe las características operativas de un servicio web utilizando un documento XML y proporciona una notación para responder las preguntas: (i) ¿de qué se trata este servicio?, (ii) ¿dónde (reside)? y (iii) ¿cómo (puede ser invocado)? (Endrei & IBM., 2004).

SOAP siglas de *Simple Object Access Protocol* (Protocolo Simple de Acceso a Objetos) es un protocolo que permite al cliente comunicarse con el proveedor de servicios. El formato del mensaje es XML (Endrei & IBM., 2004).

- Microservicios

Los microservicios se componen de una serie de pequeños servicios que funcionan de forma independiente. La arquitectura de estos ha ido ganando popularidad en respuesta a la innovación de aplicaciones y plataformas distribuidas a gran escala (Krylovskiy, Jahn, & Patti, 2015). Los beneficios que puede ofrecer la arquitectura de microservicios son: (i) diseñar, desarrollar, probar y lanzar servicios con gran agilidad; (ii) permitir la entrega continua, con la automatización de la infraestructura; (iii) la gestión de datos permite que estos servicios trabajen independientemente, así se evita que una aplicación se acomode a una sola tecnología (Francesco et al., 2017).

1.2.3 Interfaz de Programación de Aplicaciones (API)

- REST

REST o transferencia de estado representacional (*Representation State Transfer*) es un estilo arquitectónico que provee un conjunto de restricciones para sistemas hipertexto distribuidos, donde los recursos identificados por una URI⁴ (*Uniform Resource Identifier*)

⁴ Identificador de recursos uniforme: sirve para identificar recursos en internet.

vinculados se comunican mediante el intercambio de representaciones del estado de los recursos (Fielding & Taylor, 2000).

REST no es un protocolo pero se basa en el protocolo HTTP (Seda et al., 2018), se centra en la flexibilidad, escalabilidad y la reducción de latencia de una interacción (Fielding & Taylor, 2000). Al basarse en HTTP utiliza encabezados de ese protocolo y se intercambian y procesan las representaciones de recursos, las cuales suelen estar en formatos XML⁵ (*eXtensible Markup Language*) o JSON⁶ (*JavaScript Object Notation*) (Bernstein & Newcomer, 2009). REST tiene cuatro operaciones principales: (i) GET devuelve datos, (ii) POST ingresa un nuevo registro, (iii) PUT actualiza un registro y (iv) DELETE elimina un registro (Klein & Namjoshi, 2011).

Para importar los estudios a la aplicación web de automatización del SMS se utiliza el API de Mendeley. Mendeley es una herramienta tecnológica utilizada para la gestión de citas y como una red social con más de dos millones de usuarios académicos (Rodgers & Barbrow, 2013). La API de Mendeley ofrece sus recursos que son usados por investigadores, bibliotecarios, profesionales de la información y hasta empresas enfocadas en el área académica. Algunos de los principales usos son para la comprensión detallada de una área específica, facilitar la vida laboral, ampliar los límites de la biblioteca digital, análisis de métricas y entre otros usos que se puede dar con la información que nos proporciona la API de Mendeley (Mendeley, 2019).

- GraphQL

Facebook lanza públicamente GraphQL convirtiéndolo en código abierto desde 2015. En el 2012 es desarrollado como un lenguaje de consulta y manejo de datos para las API (Rodríguez-Echeverría, Cánovas Izquierdo, & Cabot, 2018). Surge por la necesidad de: (i) reducir la sobrecarga de datos transferidos y número de consultas que se requieren por separado que se da en las arquitecturas similares a REST; (ii) reducir la cantidad de errores por consultas no válidas causadas por parte del cliente; (iii) admite un modelo de datos en evolución sin la necesidad de una versión API (Bryant, 2017).

Mediante un *schema* (esquema) como se muestra en la Fig. 12 se especifica todos los tipos y operaciones tales como *query* (consulta), *mutation* (mutación) y *subscription* (suscripción) que se van a admitir. El servidor GraphQL debe verificar internamente si una solicitud es sintácticamente correcta, inequívoca y sin errores (Vogel, Weber, & Zirpins, 2018).

⁵ Lenguaje de marcado extensible: sirve para la organización de y etiquetado de documentos.

⁶ Formato de texto para intercambiar información.

```

schema {
  query : Query,
  mutation: Mutation,
  subscription: Subscription
}

type Query {
  maintenance(id: String): Maintenance
}

type Property{
  id : String!,
  customer: Customer
  ...
}

type Customer {
  id : String!,
  country: String!,
  city: String,
  ...
}

type Maintenance{
  id : String!,
  start: String!,
  property: Property
  ...
}

```

Fig. 12: Fracción de código GraphQL
Fuente: (Vogel et al., 2018)

Consulta (*query*) debe proporcionarse obligatoriamente, puesto que es el punto de partida para las consultas que el cliente realice. (Vogel et al., 2018).

Mutación es opcional y si no se proporciona el servicio no va a permitir que los clientes agreguen o modifiquen registros; el tipo *mutation* se puede definir como un punto de entrada para las solicitudes respectivas y debe ser tipo objeto (Vogel et al., 2018).

Suscripción, este tipo también es opcional y si se agrega permite un punto de entrada para el intercambio de datos en tiempo real entre los clientes y el servidor; si se agrega debe ser de tipo objeto (Vogel et al., 2018).

1.2.5 Herramientas Tecnológicas

- Node.js

Comúnmente conocido como Node, es un entorno de JavaScript ubicado en la capa del servidor y enfocado a soportar procesos de servidor de extensa ejecución; está basado en una arquitectura orientada a eventos de I/O⁷ asíncrono y en el motor de Google V8 que admite JavaScript en el navegador (comúnmente Chrome); V8 al igual que Node son de código abierto y están desarrollados en C y C++, enfocados en el rendimiento y el bajo consumo de memoria. (Tilkov & Vinoski, 2010).

- Express.js

Conocido como Express, proporciona una envoltura alrededor del nivel inferior de la interfaz de Node, proporcionando al desarrollador un medio conveniente para manejar el enrutamiento y las operaciones HTTP (como GET y POST) (Poulter, Johnston, & Cox, 2015). Express soluciona el problema de código repetitivo, así los desarrolladores pueden simplificar las API de Node.js y agregar nuevas funciones (Hahn, 2016). Las características principales de Express son: (i) *middleware*, la aplicación se divide en pequeñas partes, y en secuencia se llama uno a uno; (ii) enrutamiento, funciona como *middleware* dividiendo la aplicación en

⁷ Periférico de Entrada/Salida que permite interacción de forma bidireccional.

funciones más pequeñas, pero a diferencia de *middleware* se ejecutan dependiendo de que URL⁸ y método HTTP envía el cliente; (iv) enrutadores, permite dividir en partes aún más pequeñas las aplicaciones grandes (Hahn, 2016).

- React.js

Mayormente conocido como React es un *framework* moderno para desarrollar interfaz de usuario. React ofrece al desarrollador un DOM⁹ (*Document Object Model*) virtual para que se lo renderice en lugar del DOM real, puesto que la manipulación de DOM es una operación costosa y debe reducirse, además React reconoce que la manipulación a mano de DOM resulta en un código extenso repetitivo propenso a errores (Vipul A. M. & Sonpatki, 2016). React permite crear interfaces de usuario interactivas, sabe que es lo que debe actualizar y no toda la página, sino solo la parte que es necesario. Las vistas declarativas permiten que el código sea más predecible, sea sencillo y fácil de entender (Vipul A. M. & Sonpatki, 2016).

- Apollo Client

Apollo Client permite crear aplicaciones cliente usando GraphQL. Está diseñado para desarrollar una interfaz usuario que recupera datos con GraphQL, puede utilizarse con cualquier *front-end*¹⁰ de JavaScript (Apollo, 2019). Apollo Client es el cliente GraphQL muy flexible para React, JavaScript y plataformas nativas. Apollo Client hace automáticamente varias implementaciones que serían complejas y difíciles de implementar. Se encarga de todas las solicitudes, incluyendo el seguimiento de los estados de carga y los errores. También cuenta con un caché para localizar los datos localmente y aumentar la velocidad de la aplicación con muy poca configuración (Kimokoti, 2018).

- Neo4j

Actualmente los datos que las aplicaciones informáticas deben tratar han crecido considerablemente al igual que su complejidad, a esto se le suma el cambio frecuente de su estructura y el acceso a los datos al mismo tiempo; la mejor solución para estos aspectos no son las bases de datos relacionales, para ello se han creado tecnologías para solventar estos problemas, por ejemplo, bases de datos NoSQL¹¹, Neo4j es la más utilizada (Perçuku, Minkovska, & Stoyanova, 2017).

⁸ Uniform Resource Locator: es una dirección específica de un recurso.

⁹ Interfaz y lenguaje independiente de plataforma que permite a los programas y scripts acceder y actualizar dinámicamente el contenido, la estructura y el estilo del documento.

¹⁰ Tecnologías que están a lado del cliente, navegador web.

¹¹ No utilizan SQL como el principal lenguaje para hacer consultas.

Neo4j es una base de datos enfocada a grafos de código abierto, en lugar de tablas tradicionales almacena datos conectados como una estructura de red de grafos. Neo4j tiene cuatro elementos básicos: nodos, relaciones, propiedades y etiquetas. Los nodos son los principales elementos de datos, los cuales están conectados a otros nodos a través de relaciones direccionales. Las etiquetas son utilizadas para especificar los roles de los nodos en el grafo y organizarlos en diferentes grupos. (Zhu, Zhou, & Shao, 2019).

Un grafo es un conjunto de nodos que se relacionan entre si a través de arcos dirigidos. Se componen de un nodo inicial y uno final que pueden tener propiedades. Los grafos representan entidades como nodos, y la forma en que esas entidades se relacionan como relaciones (Perçuku et al., 2017).

Las principales ventajas que tiene Neo4j según (Perçuku et al., 2017) son:

- a) Es muy fácil y rápido recuperar, recorrer o navegar por más datos conectados.
- b) Es muy fácil representar datos conectados.
- c) Representa datos semiestructurados muy fácilmente.
- d) Los comandos del lenguaje de consulta Neo4j CQL¹² están en un formato fácil de leer y muy fácil de aprender.
- e) Utiliza un modelo de datos simple y potente.
- f) No requiere uniones complejas para recuperar los datos conectados o relacionados, ya que es muy fácil recuperar los detalles de sus nodos o relaciones adyacentes sin uniones o índices.

- Bootstrap

Desarrollado por Mark Otto y Jacob Thornton en 2011, es el *framework* de código abierto más popular, se creó con el fin de estandarizar las herramientas de *front-end* (Spurlock, 2013). Bootstrap permite crear proyectos que se adapten a la biblioteca de componentes de *front-end*, este *framework* posee un conjunto de herramientas para desarrollar con HTML, CSS y JavaScript (Bootstrap, 2019).

1.3 Metodologías y Estándares

1.3.1 Estándar ISO/IEC 25000, SQuaRE (System and Software Quality Requirements and Evaluation)

La familia de normas ISO/IEC 25000 llamada SQuaRE (*System and Software Quality Requirements and Evaluation*) tiene como meta el desarrollo de un marco de trabajo común para la evaluación de la calidad del producto software. La familia SQuaRE está formada por cinco divisiones (ISO 25000, 2019), como se indica en la Fig. 13.

¹² Cypher Query Language: Lenguaje de consulta declarativo.



Fig. 13: Divisiones de las normas SQuaRE
Fuente: (ISO 25000, 2019)

- a) ISO/IEC 2500n - División de Gestión de Calidad: Establece todos los modelos, términos y definiciones comunes referidos en las normas de la familia SQuaRE (ISO 25000, 2019).
- b) ISO/IEC 2501n – División de Modelo de Calidad: Expone un modelo de referencia de la medición de la calidad del producto, definiciones y una guía práctica para su uso (ISO 25000, 2019).
- c) ISO/IEC 2502n – División de Medición de Calidad: Contiene un modelo de referencia de la medición de la calidad del producto, definiciones de medidas de calidad y guías prácticas para su aplicación (ISO 25000, 2019).
- d) ISO/IEC 2503n – División de Requisitos de Calidad: Ayudan a especificar requisitos de calidad que pueden ser utilizados en el proceso de elicitación de requisitos de calidad del producto software a desarrollar o como entrada del proceso de evaluación (ISO 25000, 2019).
- e) ISO/IEC 2504n – División de Evaluación de Calidad: Provee normas que proporcionan requisitos, recomendaciones y guías para llevar a cabo el proceso de evaluación del producto software (ISO 25000, 2019).

Para evaluar la calidad en uso de un software se debe diseñar, medir y evaluar un modelo de calidad. A) Para el diseño del modelo de calidad en uso se debe basar en normas ISO/IEC 25010 donde enumera las características y subcaracterísticas a medir y evaluar. B) Para la medición se define las métricas correspondientes a cada una de las subcaracterísticas basándose en la ISO/IEC 25022. C) Para la evaluación de modelo de calidad se toma en cuenta la mediación de sus subcaracterísticas y se la evalúa conforme a lo que recomienda la ISO/IEC 25040.

- Modelo de Calidad en Uso

En la Tabla 1 se muestra el modelo de calidad en uso con las características, subcaracterísticas y métricas establecida en la ISO/IEC 25010 y 25022.

Tabla 1: Modelo de calidad en uso

MODELO DE CALIDAD EN USO			
Características	Subcaracterística	Métricas	
Eficacia	Tareas completas	$X = A / B$ A = Número de tareas únicas completado B = Número total de tareas únicas intentadas	
	Objetivos logrados	$\{X = 1 - \sum A_i \mid X \geq 0\}$ A _i = valor proporcional de cada objetivo que falta u objetivo incorrecto en la salida de la tarea (valor máximo = 1)	
	Los errores en una tarea	$X = A$ A = número de errores cometidos por el usuario durante una tarea	
	Tareas con errores	$X = A / B$ A = Número de tareas con errores B = Número total de tareas	
	Intensidad de errores de tareas	$X = A / B$ A = Número de usuarios de cometer un error B = Número total de usuarios realizar la tarea	
	Eficiencia	Tiempo de tareas	$X = T$ T = Tiempo de Tarea
		Eficiencia del Tiempo	$X = A / T$ A = Número de objetivos alcanzados T = Tiempo
La rentabilidad		$X = A / B$ A = Coste total de la realización de la tarea B = Número de objetivos alcanzados	
Productivo relación del tiempo		$X = T_a / T_b$ T _a = tiempo productivo = tiempo tomado para completar la tarea - el tiempo pasado obtener ayuda o asistencia - tiempo necesario recuperarse de errores - tiempo tomado buscar inútilmente T _b = tiempo de tareas	
Comportamiento Innecesario		$X = A / B$ A = Número de acciones que en realidad no fueron necesarias para lograr la tarea B = Número de acciones realizadas por el usuario.	
Consecuencias de la fatiga		$X = 1 - A / B$ A = El rendimiento actual B = rendimiento inicial.	
Satisfacción		Utilidad	$X = S/E$ S = Número de usuarios satisfechos E = Número de usuarios encuestados
	Confianza	$X = A / T, C=1-X$ X = % reclamos, % Confianza A = Número de quejas presentadas	

Comodidad

T = Total de encuestados
 $X = (A+B+C+D+F)/E$
 A = Muy de acuerdo;
 B=Algo de acuerdo;
 C=Ni de acuerdo ni en desacuerdo;
 D= Algo en desacuerdo; F=Muy en desacuerdo.
 E = Número de usuarios encuestados

Fuente: (ISO/IEC 25022, 2016)

- Escala de evaluación

En la ISO/IEC 25040 describe que la evaluación de un proyecto se define conforme al contexto en el que se desenvuelve, por lo cual en el presente trabajo de investigación se tomó en cuenta la escala de evaluación que propone la ISO/IEC 25000 para evaluar de manera cualitativa los resultados de la medición del modelo de calidad. La escala se divide en dos categorías y cuatro subcategorías (ISO/IEC 25000, 2014), ver Fig. 14.

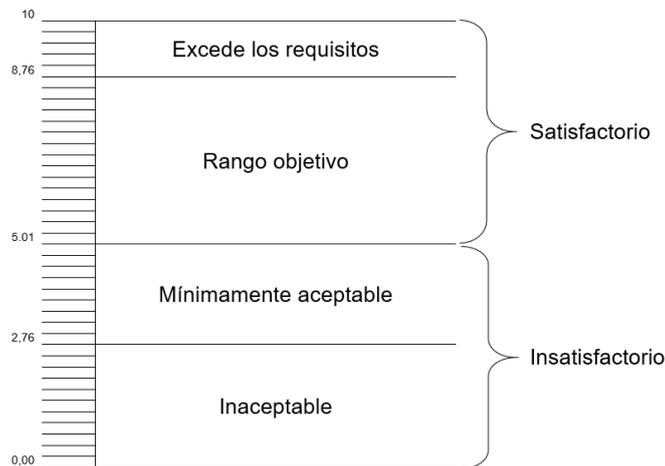


Fig. 14: Escala de medición
 Fuente: Basada en (ISO/IEC 14598-1, 1999)

1.3.2 Metodología Scrum

Scrum es una metodología ágil, que establece un ciclo de vida iterativo y permite priorizar elementos de tareas grandes a más pequeños y manejables. Además, promueve una planificación adaptativa, desarrollo y entrega evolutivos, un enfoque iterativo e impulsa a una respuesta rápida al cambio (Ungan, Cizmeli, & Demirors, 2014). (Schwaber & Sutherland, 2017) establecen que los Equipos Scrum y sus roles, los eventos, los artefactos y las reglas asociadas tiene una finalidad y son esenciales para el éxito y el uso de Scrum. La Fig. 15: Ciclo de vida Scrum representa el ciclo de vida de Scrum, el que contiene los eventos y los artefactos.

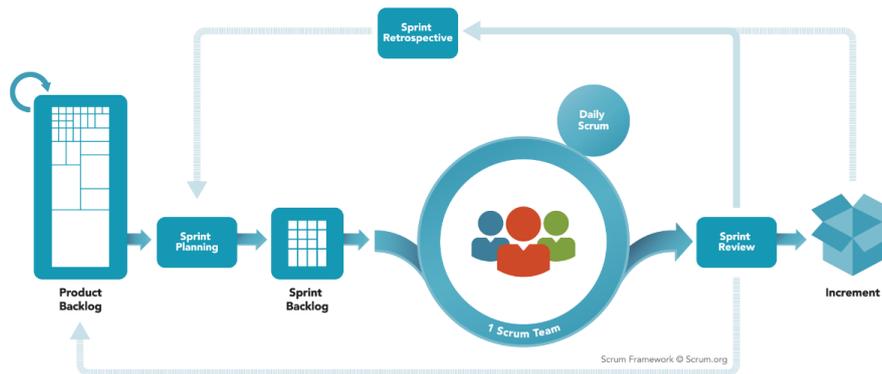


Fig. 15: Ciclo de vida Scrum
Fuente: (Acosta, 2018)

- Los Equipos Scrum

Los equipos Scrum están compuestos por: un *Product Owner* (Dueño del Producto), el Equipo de Desarrollo y un Scrum Master. Los equipos Scrum no necesitan ser dirigidos por alguien externo al equipo, ellos son autoorganizados. Además, son multifuncionales, es decir que para realizar un trabajo ellos tienen todas las competencias requeridas. Las entregas de productos son incrementales e interactivas logrando maximizar las oportunidades de retroalimentación. Cuando se entrega un producto terminado de manera incremental se logra tener un versión útil del producto (Schwaber & Sutherland, 2017).

Product Owner es el dueño del producto y sobre el recae la responsabilidad de maximizar el valor del producto que el Equipo de Desarrollo entrega. Además, es el único responsable de gestionar el *Product Backlog* (Schwaber & Sutherland, 2017).

Equipo de Desarrollo lo conforman profesionales que solo ellos crean el Incremento el cual debe ser un producto “terminado” al final de cada Sprint y debe ser potencialmente liberable. En la Revisión del Sprint requiere de un incremento “terminado”.

Scrum Master está a la cabeza del Equipo Scrum, sirviendo como un líder y vigilando que todos los miembros del equipo trabajen de acuerdo a la teoría, prácticas y reglas de Scrum. El Scrum Master asegura que Scrum sea claro y adoptado. Además, de organizar las reuniones y servir como un guía (Schwaber & Sutherland, 2017).

- Artefactos

Product Backlog es una lista de todas las características, funcionalidades, requisitos, mejoras y correcciones en el producto. Está lista debe ser ordenada (Schwaber & Sutherland, 2017).

Sprint Backlog es la agrupación de *Product Backlog* elegidos para el Sprint, adicional a eso un plan para entregar el Incremento del producto y lograr cumplir el objetivo del planteado (Schwaber & Sutherland, 2017).

Incremento es resultado después de finalizar un Sprint, el cual debe estar “terminado” y debe estar listo para ser utilizado (Schwaber & Sutherland, 2017).

- Eventos Scrum

Los eventos permiten tener regularidad y reducir la necesidad de reuniones no planificadas. Estos eventos tienen un tiempo máximo definido y no puede reducirse o prologarse una vez iniciado un Sprint.

Durante el Sprint se desarrolla un incremento de producto “terminado” el cual debe estar listo para ser usado y altamente listo para el despliegue; esto debes hacerse durante un mes o menos. Cuando un Sprint finaliza debe comenzar uno nuevo (Schwaber & Sutherland, 2017).

Planificación del Sprint es una reunión que puede durar ocho horas máximo para un Sprint de un mes. El Scrum Master es el encargado de este evento, además de que los participantes comprendan el propósito (Schwaber & Sutherland, 2017).

Cada integrante del Equipo Scrum eligen un grupo de *Product Backlog Items* y así se obtendrá el *Sprint Backlog* que iniciará al siguiente Sprint (Acosta, 2018).

Scrum Diario es una reunión diaria que debe durar máximo 15 minutos. Cada integrante durante la reunión debe responder: ¿Qué hizo ayer?, ¿Qué tiene planeado para hoy? Y ¿Qué obstáculos tuvo? (Schwaber & Sutherland, 2017).

Revisión del Sprint se realiza una revisión del Incremento, esta revisión el Equipo Scrum y los interesados participan acerca de lo que se hizo en el Sprint y lo que se puede mejorar (Schwaber & Sutherland, 2017).

Retrospectiva del Sprint en donde Product Owner se reúne con el Equipo de Desarrollo y el Scrum Master para analizar en las fallas que se ha tenido y en que se puede mejorar, enfocándose en las personas y el proceso. Esta acción se hace después de la Revisión del Sprint (Acosta, 2018).

CAPÍTULO 2

Desarrollo

En este capítulo se muestra las fases de desarrollo del sistema de automatización del Systematic Mapping Study “SMS”, utilizando la metodología Scrum. En la Fig. 16 se presenta la estructura del capítulo.

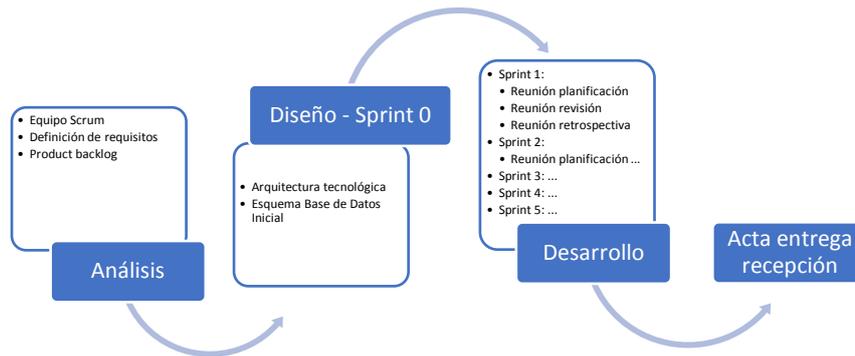


Fig. 16: Estructura capítulo 2
Fuente: Propia

2.1 Análisis

2.1.1 Equipo Scrum

Es en esta sección se define el equipo Scrum para el desarrollo de la aplicación Systematic Mapping Study “SMS”. En la Tabla 2 se especifica los roles de cada integrante del equipo.

Tabla 2: Roles del proyecto

Nombre	Rol	Cargo
MSc. Antonio Quiña	Product Owner.	Director del presente Trabajo de Grado y Docente de la Universidad Técnica del Norte
Edwin Bastidas	Scrum Master. Equipo de Desarrollo.	Tesista
PhD. Pablo Fernández Montés	Stakeholders	Docente titular - Universidad de Sevilla - España
PhD. José María García		

2.1.2 Definición de Requisitos

Los requisitos están definidos en historias de usuario de acuerdo con la metodología Scrum, las cuales se han realizado en conjunto con el Product Owner y Scrum Master del

proyecto. La plantilla usada en las historias de usuarios es adaptación de Scrum Manager (Scrum Manager, 2014).

Tabla 3: Historia de Usuario 1

Historias de usuario		
ID: HSMS-01	Usuario: Administrador	
Nombre: Administración de Usuarios		
Prioridad: Alta	Dependencia: N/A	Estimación: 12 h
Descripción: Como administrador quiero administrar los usuarios, mediante la creación, actualización y listado de los mismos. Los usuarios deben tener roles: investigador o administrador. Los atributos que al menos debe incluir son: <ul style="list-style-type: none">• Nombres• Correo electrónico• Nombre de usuario• Contraseña		
Pruebas de aceptación: <ul style="list-style-type: none">• Al ingresar un correo con un formato inválido el sistema debe validar y debe mostrar un mensaje de error.• Al ingresar un correo ya registrado, el sistema debe validar y mostrar un mensaje de error.• Al ingresar un usuario ya registrado, el sistema debe validar y mostrar un mensaje de error.• Al ingresar la contraseña y confirmar la contraseña si no son iguales el sistema debe validar y mostrar un mensaje de error.• Si el ingreso o la actualización son correctas debe mostrarse un mensaje de éxito.• Si existe un error al momento de ingresar o actualizar debe mostrar un mensaje de error.		

Tabla 4: Historia de Usuario 2

Historias de usuario		
ID: HSMS-02	Usuario: Administrador	
Nombre: Administración de bases de datos bibliográficas		
Prioridad: Alta	Dependencia: N/A	Estimación: 12 h
Descripción: Como administrador quiero administrar las bases de datos bibliográficas, mediante la creación, actualización y listado de las mismas, en donde tenga por lo menos los siguientes atributos:		
<ul style="list-style-type: none"> • Nombre • Descripción 		
Pruebas de aceptación:		
<ul style="list-style-type: none"> • Al ingresar el nombre de una base de datos bibliográfica que ya existe, el sistema debe validar y mostrar un mensaje de error. • Si no se ingresa el nombre y/o la descripción debe deshabilitarse el botón de guardar. • Si el ingreso o actualización son correctas debe mostrarse un mensaje de éxito. • Si existe un error técnico en los servicios de base de datos o servidor de aplicaciones al momento de ingresar o actualizar el registro, debe mostrar un mensaje de error. 		

Tabla 5: Historia de Usuario 3

Historias de usuario		
ID: HSMS-03	Usuario: Investigador	
Nombre: Registro investigador		
Prioridad: Alta	Dependencia: N/A	Estimación: 8 h
Descripción: Como investigador quiero tener una opción en la pantalla principal en la que pueda acceder a una opción para registrarme como investigador, los campos mínimos solicitados son:		
<ul style="list-style-type: none"> • Nombres • Correo electrónico • Nombre de usuario • Contraseña 		
Pruebas de aceptación:		
<ul style="list-style-type: none"> • Al ingresar un correo con un formato inválido el sistema debe validar y debe mostrar un mensaje de error. • Al ingresar un correo ya registrado, el sistema debe validar y mostrar un mensaje de error. • Al ingresar un usuario ya registrado, el sistema debe validar y mostrar un mensaje de error. • Al ingresar la contraseña errónea el sistema debe validar y mostrar un mensaje de error. • Si el ingreso o la actualización son correctas debe mostrarse un mensaje de éxito. • Si se produce un error al momento de ingresar o actualizar un registro se debe describir en un mensaje de error. 		

Tabla 6: Historia de Usuario 4

Historias de usuario		
ID: HSMS-04	Usuario: Administrador/Investigador	
Nombre: Autenticación		
Prioridad: Alta	Dependencia: 1,3	Estimación: 12 h
Descripción: Como usuario del sistema quiero iniciar sesión ingresando el nombre de usuario y contraseña, dependiendo del rol debe mostrarme la página de inicio correspondiente a mi rol.		
Pruebas de aceptación:		
<ul style="list-style-type: none"> • Al ingresar un usuario no registrado el sistema debe validar y mostrar un mensaje de error. • Al ingresar un usuario registrado y una contraseña incorrecta el sistema debe validar y mostrar un mensaje de error. 		

Tabla 7: Historia de Usuario 5

Historias de usuario		
ID: HSMS-05	Usuario: Investigador	
Nombre: Administración Systematic Mapping Study "SMS"		
Prioridad: Alta	Dependencia: N/A	Estimación: 12 h
Descripción: Como investigador quiero administrar los SMS, creando, actualizando y listando los mismos. El SMS debe al menos los siguientes atributos:		
<ul style="list-style-type: none"> • Título • Fecha de creación • Usuario de quien creo • Fecha de la última modificación • Descripción • Estado (proceso, completo) 		
Pruebas de aceptación:		
<ul style="list-style-type: none"> • Si un usuario ingresa un título que ya existe para ese mismo usuario el sistema debe validar y mostrar un mensaje de error. • Si no se ingresa el título y/o la descripción se debe deshabilitar el botón para guardar o actualizar. • Si el ingreso o la actualización son correctas debe mostrarse un mensaje de éxito. • Si existe un error al momento de ingresar o actualizar debe mostrar un mensaje de error. 		

Tabla 8: Historia de Usuario 6

Historias de usuario

ID: HSMS-06

Usuario: Investigador

Nombre: Planificación SMS

Prioridad: Alta

Dependencia: 2,5

Estimación: 12 h

Descripción: Después de creado el SMS con datos generales como investigador quiero seleccionar las bases de datos bibliográficas donde voy a realizar las búsquedas. También quiero administrar las preguntas de investigación, mediante la creación, actualización y listado de las mismas. Además, se debe mostrar una breve explicación sobre la Fase 1 y de cada uno de los pasos.

Pruebas de aceptación:

- Al seleccionar una base de datos bibliográfica ya seleccionada el sistema debe verificar y mostrar un mensaje de error.
- Al ingresar una pregunta de investigación ya ingresada el sistema debe verificar y mostrar un mensaje de error.
- Si no se ingresa la pregunta de investigación se debe deshabilitar el botón para guardar o actualizar.
- Si el ingreso o la actualización son correctas debe mostrarse un mensaje de éxito.
- Si existe un error al momento de ingresar o actualizar debe mostrar un mensaje de error.
- Debe tener al menos un registro para poder continuar.

Tabla 9: Historia de Usuario 7

Historias de usuario

ID: HSMS-07

Usuario: Investigador

Nombre: Identificación de estudios - Registro de búsquedas

Prioridad: Alta

Dependencia: 6

Estimación: 12 h

Descripción: Como investigador quiero administrar las búsquedas de estudios, mediante la creación, actualización y listado de las mismas. Debe tener al menos los siguientes atributos:

- Base de datos bibliográfica
- Cadena de búsqueda
- Rango de fechas
- Ordenado por
- Otro criterio
- Número de documentos encontrados

También quiero guardar un archivo como evidencia (opcional) y descargar dicho archivo.

Pruebas de aceptación:

- Si seleccionó una base de datos e ingreso una cadena de búsqueda ya existente el sistema debe verificar y mostrar un mensaje de error.
- Si el ingreso o la actualización son correctas debe mostrarse un mensaje de éxito.
- Si existe un error al momento de ingresar o actualizar debe mostrar un mensaje de error.
- Debe tener al menos un registro para poder continuar.

Tabla 10: Historia de Usuario 8

Historias de usuario**ID:** HSMS-08**Usuario:** Investigador**Nombre:** Identificación de estudios - Primer filtro**Prioridad:** Alta**Dependencia:** 7**Estimación:** 12 h

Descripción: Como investigador quiero administrar los primeros filtros mediante, la creación, actualización y listado de los mismos de cada una de las búsquedas realizadas. Debe tener al menos los siguientes atributos:

- Búsqueda de estudios
- Criterio

También quiero guardar un archivo como evidencia (opcional) y descargar dicho archivo.

Pruebas de aceptación:

- No se puede ingresar dos veces el mismo filtro a una búsqueda.
- Si el ingreso o la actualización son correctas debe mostrarse un mensaje de éxito.
- Si existe un error al momento de ingresar o actualizar debe mostrar un mensaje de error.
- Debe tener al menos un registro para poder continuar.

Tabla 11: Historia de Usuario 9

Historias de usuario**ID:** HSMS-09**Usuario:** Investigador**Nombre:** Identificación de estudios - Filtro estudios**Prioridad:** Alta**Dependencia:** 8**Estimación:** 12 h

Descripción: Como investigador quiero administrar los filtros de inclusión y/o exclusión mediante la creación, actualización y listado de los mismos. Debe tener al menos los siguientes atributos:

- Tipo de criterio inclusión o exclusión
- Criterio

También quiero guardar un archivo como evidencia (opcional) y descargar dicho archivo.

Pruebas de aceptación:

- No se puede ingresar dos veces el mismo filtro, si lo hace el sistema debe mostrar un mensaje de error.
- Si el ingreso o la actualización son correctas debe mostrarse un mensaje de éxito.
- Si existe un error al momento de ingresar o actualizar debe mostrar un mensaje de error.
- Debe tener al menos un registro para poder continuar.

Tabla 12: Historia de Usuario 10

Historias de usuario**ID:** HSMS-10**Usuario:** Investigador**Nombre:** Identificación de estudios - Búsqueda profunda**Prioridad:** Media**Dependencia:** 9**Estimación:** 12 h

Descripción: Como investigador quiero administrar las búsquedas profundas mediante la creación, actualización y listado de los mismos. Debe tener al menos los siguientes atributos:

- Estrategia
- Descripción
- Número de documentos encontrados en la búsqueda profunda realizada

También podrá cargar (opcional) y descargar una evidencia.

Pruebas de aceptación:

- Este paso es opcional.
- Si el ingreso o la actualización son correctas debe mostrarse un mensaje de éxito.
- Si existe un error al momento de ingresar o actualizar debe mostrar un mensaje de error.
- Debe tener al menos un registro para poder continuar.

Tabla 13: Historia de Usuario 11

Historias de usuario**ID:** HSMS-11**Usuario:** Investigador**Nombre:** Extracción y clasificación - Importar**Prioridad:** Alta**Dependencia:** 9**Estimación:** 12 h

Descripción: Como investigador quiero importar los documentos exportados en las diferentes carpetas de Mendeley.

Pruebas de aceptación:

- No se puede importar un documento dos veces.
- Debe tener al menos un registro para poder continuar.

Tabla 14: Historia de Usuario 12

Historias de usuario		
ID: HSMS-12	Usuario: Investigador	
Nombre: Extracción y clasificación - Campos personalizados		
Prioridad: Media	Dependencia: 11	Estimación: 12 h
Descripción: Como investigador quiero administrar los campos (columnas) personalizados, mediante la creación, actualización y listado de los mismos. Debe tener al menos los siguientes atributos:		
<ul style="list-style-type: none"> • Nombre. • A que pregunta responde. 		
Pruebas de aceptación:		
<ul style="list-style-type: none"> • No se puede ingresar dos veces el mismo campo. 		

Tabla 15: Historia de Usuario 13

Historias de usuario		
ID: HSMS-13	Usuario: Investigador	
Nombre: Extracción y clasificación - Completar información		
Prioridad: Alta	Dependencia: 11	Estimación: 12 h
Descripción: Como investigador quiero completar la información de cada uno de los documentos, debe estar dividida en cuatro secciones. La primera sección debe contener la información básica, la segunda la información restante del documento, la tercera a que pregunta responde y por último los campos personalizados creados.		
Pruebas de aceptación:		
<ul style="list-style-type: none"> • La información básica es obligatorio completar. • No debe ingresar dos veces o más el mismo código de pregunta. 		

Tabla 16: Historia de Usuario 14

Historias de usuario		
ID: HSMS-14	Usuario: Investigador	
Nombre: Extracción y clasificación - Exportar CSV		
Prioridad: Alta	Dependencia: 13	Estimación: 12 h
Descripción: Como investigador quiero exportar la información en archivo CSV de los documentos importados.		
Pruebas de aceptación:		
<ul style="list-style-type: none"> • No debe existir inconsistencia en los datos. 		

Tabla 17: Historia de Usuario 15

Historias de usuario		
ID: HSMS-15	Usuario: Investigador	
Nombre: Extracción y clasificación - Reporte		
Prioridad: Alta	Dependencia: 13	Estimación: 6 h
Descripción: Como investigador quiero generar un reporte en PDF del proceso realizado en un Systematic Mapping Study "SMS", mostrando la información en tablas.		
Pruebas de aceptación:		
<ul style="list-style-type: none"> • No debe existir inconsistencias en los datos. 		

2.1.2 Product Backlog

A partir de las historias de usuarios definidas, el Product Owner las prioriza y asigna un orden para el desarrollo del producto de software, ver Tabla 18.

Tabla 18: Product Backlog

Orden	ID	Descripción	Estimación (Horas)
01	HSMS-01	Administrador de Usuarios	17
02	HSMS-03	Registro investigador	8
03	HSMS-02	Administración de bases de datos bibliográficas	12
04	HSMS-04	Autenticación	12
05	HSMS-05	Administración Systematic Mapping Study "SMS"	12
06	HSMS-06	Planificación SMS	12
07	HSMS-07	Identificación de estudios - Registro de búsquedas	12
08	HSMS-08	Identificación de estudios - Primer filtro	12
09	HSMS-09	Identificación de estudios - Filtro Estudios	12
10	HSMS-10	Identificación de estudios - Búsqueda profunda	12
11	HSMS-11	Extracción y clasificación - Importar	12
12	HSMS-12	Extracción y clasificación - Campos personalizados	12
13	HSMS-13	Extracción y clasificación - Completar información	12
14	HSMS-14	Extracción y clasificación - Exportar CSV	6
15	HSMS-15	Extracción y clasificación - Reporte	8

2.2 Diseño – Sprint 0

El diseño del proyecto de desarrollo de software se lo realizó en una iteración denominada Sprint 0, en donde se definió la arquitectura tecnológica, un esquema inicial de la base de datos y un diagrama de procesos del sistema.

2.2.1 Resumen Sprint 0

En la reunión de planificación realizada el día 03/06/2019 en presencia del Product Owner, Scrum Master y Equipo de Desarrollo se obtuvo como resultado las tareas a realiza, ver Tabla 19.

Tabla 19: Tareas - Sprint 0

Tarea	Horas
Diseño arquitectura tecnológica	4
Estructura inicial de base de datos	5
Diagrama de procesos	5
Reunión Planificación	2
Reunión Revisión	2
Reunión Retrospectiva	2

En la reunión de revisión y retrospectiva se las realizó el día 10/06/2019, en presencia del Product Owner, Scrum Master y Equipo de Desarrollo se obtuvo como resultado el incremento entregable de la arquitectura tecnológica, estructura inicial de la base de datos y diagrama de proceso.

2.2.2 Arquitectura Tecnológica

La aplicación web se desarrolló utilizando una arquitectura orientada a microservicios en un entorno de ejecución Node.js. En lado del servidor se encuentra las tecnologías: Apollo, GraphQL y la base de datos Neo4j; y en el cliente las tecnologías: Apollo, Express, React y Bootstrap y la API de Mendeley como se muestra en la Fig. 17.

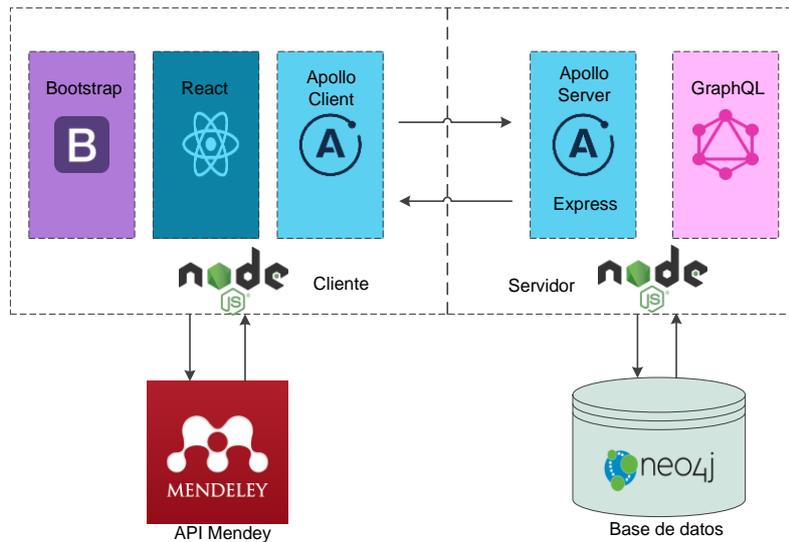


Fig. 17: Arquitectura tecnológica
Fuente: Propia

2.2.3 Esquema Base de Datos Inicial

En esta sección cabe mencionar que se utilizó una base de datos NoSQL, que es orientada a grafos. En el esquema inicial de la base de datos se muestra los principales nodos, etiquetas y sus relaciones, ver Fig. 18.

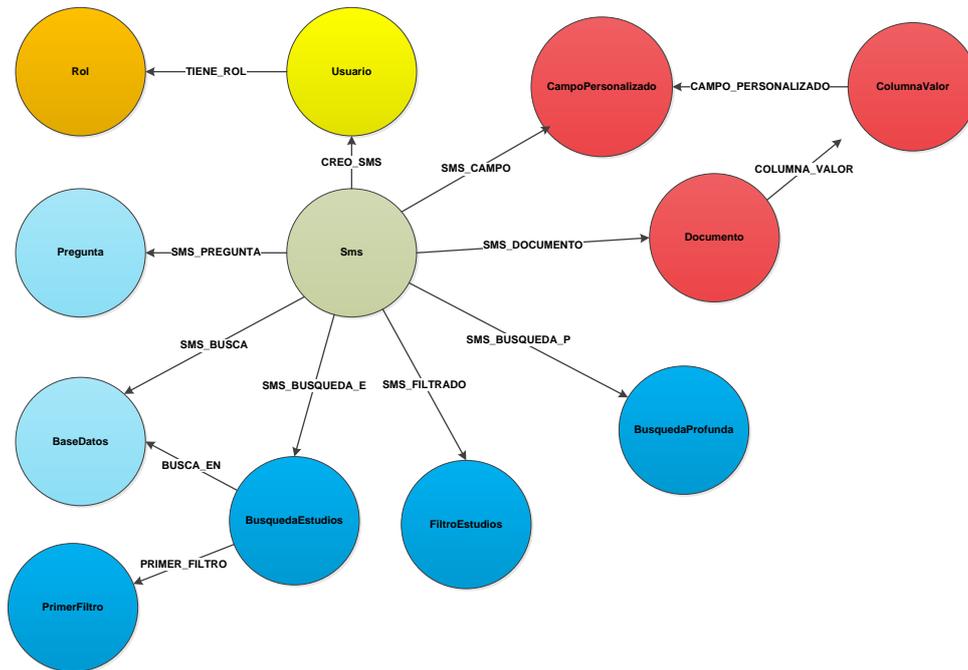


Fig. 18: Esquema inicial de Base de Datos
Fuente: Propia

2.2.4 Diagrama de Procesos

En la Fig. 19 se muestra el diagrama de proceso de la aplicación para la automatización del SMS y en la Tabla 20 se describe cada una de las actividades.

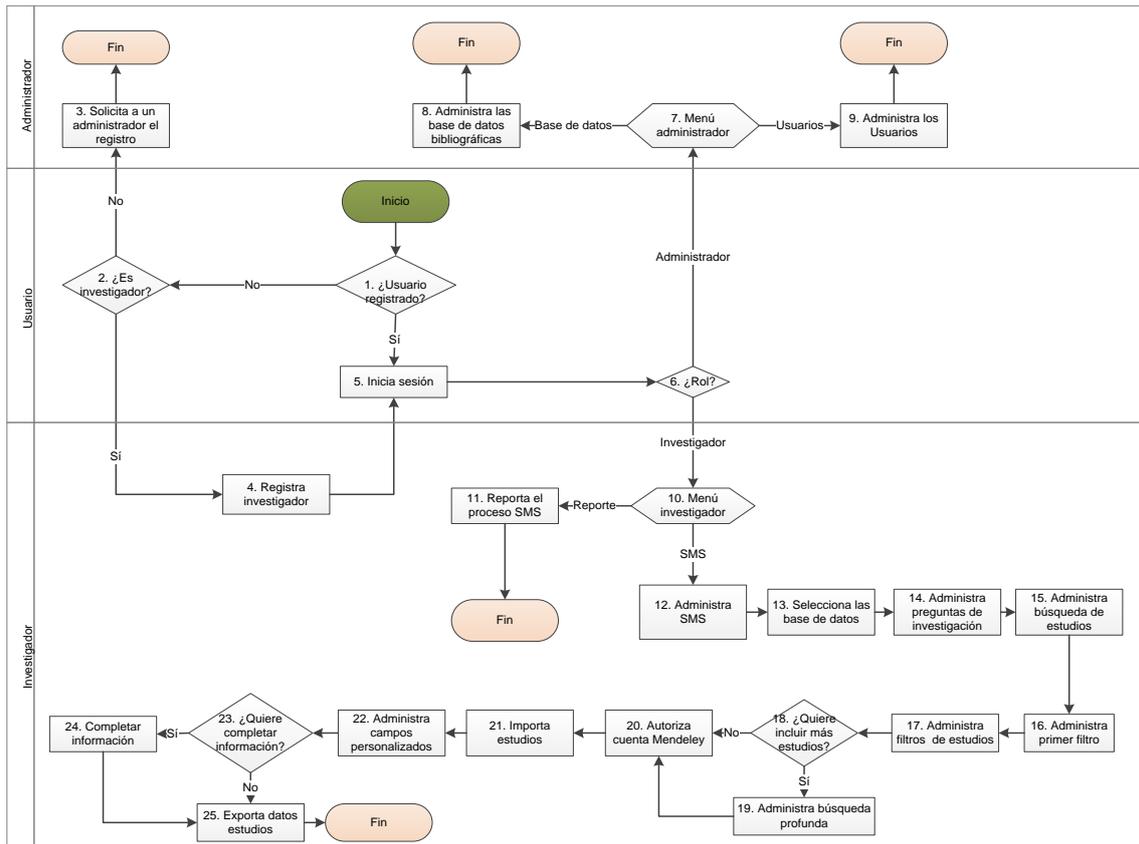


Fig. 19: Diagrama de procesos de la aplicación
Fuente: Propia

Tabla 20: Descripción del diagrama de procesos de la aplicación

N°	Actividad	Descripción	Responsable
1	¿Usuario registrado?	Si el usuario esta registrado ir a la actividad Usuario N° 5, sino a la actividad N° 2.	Usuario
2	¿Es investigador?	Si el usuario es investigador ir a la actividad Usuario N° 4. Caso contrario ir a la actividad N° 3.	Usuario
3	Solicita a un administrador el registro	Debe solicitar que un administrador lo registre en el sistema.	Administrador
4	Registra investigador	Debe ingresar los datos solicitados para crear una cuenta como investigador.	Investigador
5	Inicia sesión	Debe ingresar las credenciales de inicio de sesión.	Usuario
6	¿Rol?	Si tiene rol de administrador ir a la actividad Usuario N° 7, sino ir a la N° 10.	Usuario
7	Menú administrador	Si selecciona Usuarios ir a la actividad N° 9, caso contrario a la N° 8.	Administrador
8	Administra las base de datos bibliográficas	Crea, actualiza y lista las bases de datos bibliográficas.	Administrador
9	Administra los Usuarios	Crea, actualiza y lista usuarios.	Administrador
10	Menú investigador	Si selecciona Reporte ir a la actividad N° 11. Caso contrario ir a la N° 12.	Investigador

11	Reporta el proceso SMS	Se genera un reporte PDF del proceso realizado en un SMS.	Investigador
12	Administra SMS	Crea, actualiza y lista los SMS.	Investigador
13	Selecciona las bases de datos.	Selecciona las bases de datos bibliográficas en donde va a realizar las búsquedas.	Investigador
14	Administra las preguntas de investigación.	Crea, actualiza y lista las preguntas de investigación.	Investigador
15	Administra búsqueda de estudios	Crea, actualiza y lista los parámetros de las búsquedas de estudios realizadas.	Investigador
16	Administra primer filtro	Crea, actualiza y lista el primero filtro utilizado en cada una de las búsquedas.	Investigador
17	Administra filtros de estudios	Crea, actualiza y lista los filtros de estudios realizados.	Investigador
18	¿Quiere incluir más estudios?	Si quiere incluir más estudios ir a la actividad N° 19, sino ir a la N° 20.	Investigador
19	Administra búsqueda profunda	Crea, actualiza y lista las entretejías utilizada para la búsqueda profunda.	Investigador
20	Autoriza cuenta de Mendeley	Autorizar el uso de la cuenta de Mendeley para importar los estudios.	Investigador
21	Importar estudios	Seleccionar la carpeta donde se encuentra los estudios e importar.	Investigador
22	Administra campos personalizados	Crea, actualiza y lista los campos personalizados para la clasificación de los estudios.	Investigador
23	¿Quiere completar la información?	Si quiere completar la información ir a la actividad N° 24, caso contrario ir a la N° 25.	Investigador
24	Completar información	Completar información de cada uno de los estudios importados.	Investigador
25	Exporta datos estudios	Exportar en un archivo CSV los estudios con sus datos y los campo personalizados creados.	Investigador

2.3 Desarrollo del Sistema Web SMS

El desarrollo se lo realizó de forma iterativa-incremental, en donde una iteración en Scrum se denomina Sprint, los cuales contienen los siguientes eventos: Reunión de planificación, reuniones diarias, reunión de revisión y reunión de retrospectiva.

NOTA: Es necesario mencionar que durante el desarrollo del sistema no se realizó las reuniones diarias ya que el equipo de desarrollo se conforma de una sola persona.

A continuación, en la Tabla 21 se detalla un resumen de los Sprints ejecutados en el proyecto.

Tabla 21: Resumen Sprints

Sprint	Fecha inicio	Fecha fin	Duración (Horas)
Sprint 0	03/06/2019	11/06/2019	20
Sprint 1	11/06/2019	11/07/2019	42
Sprint 2	12/07/2019	12/08/2019	42
Sprint 3	02/09/2019	02/10/2019	42
Sprint 4	03/10/2019	04/11/2019	42
Sprint 5	05/11/2019	06/12/2019	32

2.3.1 Sprint 1

- Reunión de Planificación

Fecha: 11/06/2019

Asistentes: Product Owner, Scrum Master y Equipo de Desarrollo

Resultado: Sprint Backlog – Sprint 1

- Sprint Backlog – Sprint 1

El Sprint Backlog contiene las historias de usuarios que se van a desarrollar en el Sprint, además se detalla las tareas designadas al equipo de desarrollo.

Tabla 22: Sprint Backlog - Sprint 1

Historias de Usuario	Nombre	Tarea	Horas
HSMS-01	Administrador de Usuarios	Configurar base de datos	4
		Codificar en el servidor GraphQL para crear, actualizar y listar usuarios	4
		Desarrollo de la vistas para crear, actualizar y listar usuarios	4
		Pruebas	2
		Corrección de errores y mejoras	2
HSMS-02	Administración de bases de datos bibliográficas	Codificar mutación y resolver para crear, actualizar y listar base de datos bibliográficas	4
		Codificar vistas para crear, actualizar y listar base de datos bibliográficas	4
		Pruebas	2

		Corrección de errores y mejoras	2
HSMS-03	Registro investigador	Codificar vistas para el ingreso de los datos para el registro autónomo del investigador	4
		Pruebas	2
		Corrección de errores y mejoras	2
Reuniones Scrum	Planificación		2
	Revisión		2
	Retrospectiva		2

- Reunión de Revisión

Se realiza una revisión del incremento del producto desarrollado en el Sprint.

Fecha: 11/07/2019

Asistentes: Product Owner, Scrum Master y Equipo de Desarrollo

Resultado: Sprint Backlog revisado (pruebas de aceptación), y el incremento del producto potencialmente entregable.

- Pruebas de Aceptación

Para comprobar el cumplimiento de los requisitos definidos en el Sprint, se ha realizado las pruebas de aceptación.

Tabla 23: Pruebas de aceptación Sprint 1

Historia de Usuario	Nombre	Funcionalidad	Aceptación	
			SI	NO
HSMS-01	Administración de Usuarios	Crear usuarios	X	
		Actualizar usuarios	X	
		Consultar usuarios	X	
HSMS-02	Administración de bases de datos bibliográficas	Crear bases bibliográficas	X	
		Actualizar bibliográficas	X	
		Consultar bibliográficas	X	
HSMS-03	Registro investigador	Registrar investigador	X	

- Incremento

Para la administración de usuarios, se puede registrar los usuarios que accederán al sistema, a esta opción solo podrá ingresar el rol administrador, ver Fig. 20, Fig. 21 y Fig. 22.

Agregar usuario ✕

Nombres

Apellidos

Correo institucional

Usuario

Rol

Contraseña

Confirmar contraseña

Fig. 20: Vista para crear usuario
Fuente: Propia

Editar usuario ✕

Nombres

Apellidos

Correo institucional

Usuario

Activo

Fig. 21: Vista para actualizar usuario
Fuente: Propia

Opción	ID	Nombres	Apellidos	Usuario	Email	Rol	Activo
 	1	Edwin	Bastidas	admin	admin@admin.com	Rol administrador	
 	2	Antonio	Quiña	aquina	aquina@utn.edu.ec	Rol investigador	

Total registros: 2, mostrando 1 al 2

Atrás 0 Siguiendo

Fig. 22: Vista listar usuarios
Fuente: Propia

Para la gestión de las bases de datos bibliográficas el usuario con rol administrador puede crear, actualizar y consultar, ver vistas en la Fig. 23, Fig. 24 y Fig. 25.

Agregar base de datos bibliográficas

Nombre
Nombre de base de datos bibliográfica

Descripción
Descripción de base de datos bibliográfica

 Guardar  Cancelar

Fig. 23: Vista para crear base de datos bibliográfica
Fuente: Propia

Editar base de datos bibliográfica

Nombre
Scopus

Descripción
Scopus es una base de datos bibliográfica de resúmenes y citas de artículos de revistas científicas.

Activo
SI

 Guardar  Cancelar

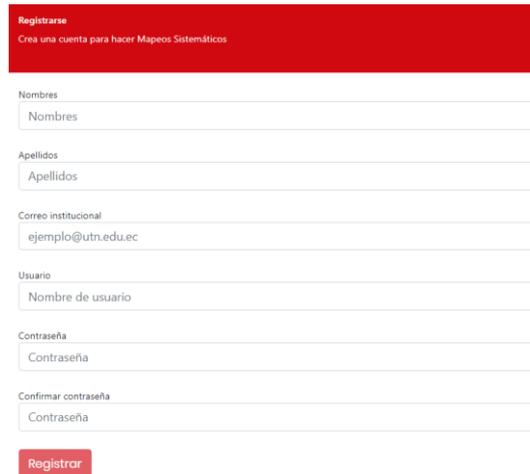
Fig. 24: Vista para actualizar base de datos bibliográfica
Fuente: Propia

Opción	Nombre	Descripción	Activo
	IEEE Xplore	IEEE Xplore es una base de datos de investigación para el descubrimiento.	
	Scopus	Scopus es una base de datos bibliográfica de resúmenes y citas de artículos de revistas científicas.	

Atrás 0 Siguiendo

Fig. 25: Vista para listar las base de datos bibliográficas
Fuente: Propia

En la Fig. 26 se muestra la vista para el registro del investigador.



El formulario de registro de investigador tiene un encabezado rojo con el texto "Registrarse" y "Crea una cuenta para hacer Mapeos Sistemáticos". A continuación, hay campos de entrada para: Nombres, Apellidos, Correo institucional (con el ejemplo "ejemplo@utn.edu.ec"), Usuario (con el texto "Nombre de usuario"), Contraseña y Confirmar contraseña (ambos con el texto "Contraseña"). Al final del formulario hay un botón rojo con el texto "Registrar".

Fig. 26: Vista para el registro del investigador
Fuente: Propia

- Reunión de Retrospectiva – Sprint 1

Fecha: 11/07/2019

Asistentes: Product Owner, Scrum Master y Equipo de Desarrollo

Resultado: Plan de mejora

- Plan de Mejora

El plan de mejora se define en tres componentes: aciertos (¿qué salió bien del Sprint?), errores (¿qué no salió bien del Sprint?) y mejoras (¿qué mejoras se implementará?).

Tabla 24: Plan de mejora

Plan de mejora
Aciertos: <ul style="list-style-type: none">• Servidor GraphQL conectado con Neo4j correctamente.
Problemas: <ul style="list-style-type: none">• Escasa información sobre GraphQL y Neo4j.• Falta de conocimiento en React.
Mejoras: <ul style="list-style-type: none">• Investigar proyectos realizados con GraphQL y Neo4j• Capacitación en React.

2.3.2 Sprint 2

- Reunión de Planificación

Fecha: 12/07/2019

Asistentes: Product Owner, Scrum Master y Equipo de Desarrollo

Resultado: Sprint Backlog – Sprint 2

- Sprint Backlog – Sprint 2

Tabla 25: Sprint Backlog - Sprint 2

Historias de Usuario	Nombre	Tarea	Horas
HSMS-04	Autenticación	Codificar mutación y resolver para la autenticación	4
		Codificar vista para la autenticación	4
		Pruebas de autenticación	2
		Corrección de errores y mejoras	2
HSMS-05	Administración Systematic Mapping Study "SMS"	Codificar mutación y resolver para crear, actualizar y listar de SMS	4
		Codificar vistas para crear, actualizar y listar SMS	4
		Pruebas	2
		Corrección de errores y mejoras	2
HSMS-06	Planificación SMS	Codificar en el servidor GraphQL para seleccionar, deseleccionar y listar las bases de datos bibliográficas,	2
		Codificar en el servidor GraphQL para crear, actualizar y listar las Preguntas	2
		Codificar vistas para seleccionar, deseleccionar y listar las bases de datos bibliográficas	2
		Codificar vistas para crear, actualizar y listar las preguntas	2
		Pruebas	2
		Corrección de errores y mejoras	2
Reuniones Scrum	Planificación		2
	Revisión		2
	Retrospectiva		2

- Reunión de Revisión

Fecha: 12/08/2019

Asistentes: Product Owner, Scrum Master y Equipo de Desarrollo

Resultado: Sprint Backlog revisado e incremento del producto potencialmente entregable

- Pruebas de Aceptación

Tabla 26: Pruebas de aceptación Sprint 2

Historia de Usuario	Nombre	Funcionalidad	Aceptación	
			SI	NO
HSMS-04	Autenticación	Iniciar sesión	X	
		Redirigir a la página de inicio correspondiente	X	
HSMS-05	Administración Systematic Mapping Study "SMS"	Crear bases SMS	X	
		Actualizar SMS	X	
		Consultar SMS	X	
HSMS-06	Planificación SMS	Seleccionar bases bibliográficas	X	

- Incremento

Los usuarios pueden iniciar sesión con usuario y contraseña. En las Fig. 27 se muestra el vista para la autenticación.

Fig. 27: Vista para la autenticación
Fuente: Propia

Los investigadores pueden crear, actualizar y consultar SMS. En las Fig. 27, Fig. 28, Fig. 29 se presenta los vistas.

Fig. 28: Vista para crear SMS
Fuente: Propia

Editar SMS ✕

Título

Descripción

Activo

Guardar
Cancelar

Fig. 29: Vista para actualizar SMS
Fuente: Propia

+	Opción	Título	Creación	Actualización	Estado	Activo
-		GraphQL	2020-01-02 14:01:17	2020-01-02 14:01:17	En proceso	SI
-		REST API	2020-01-06 16:01:43	2020-01-06 16:01:43	En proceso	SI

5 Total registros: 2, mostrando 1 al 2 Atrás 0 Siguiente

Fig. 30: Vista para listar SMS
Fuente: Propia

El investigador en la planificación del SMS selecciona las bases de datos bibliográficas y registra las preguntas de investigación. En la Fig. 31 y Fig. 32 se muestran las vistas para seleccionar, deseleccionar y listar las bases de datos bibliográficas. En las Fig. 33, Fig. 34 y Fig. 35 se muestran las vistas para crear, actualizar y listar las preguntas de investigación.

Bases de datos bibliográficas ✕

Seleccione la base de datos donde realizará la búsqueda

✓ Seleccionar
✕ Cancelar

Fig. 31: Vista para seleccionar base de datos bibliográfica
Fuente: Propia

Buscar

Opción	Nombre
	Scopus

5 Atrás 0 Siguiente

Fig. 32: Vista para deseleccionar y listar base de datos bibliográficas seleccionadas
Fuente: Propia

Preguntas de investigación ✕

Ejemplos de preguntas basadas en el modelo: 5W + 1H ✕

- ¿Quién está investigando?
- ¿Qué se está investigando?
- ¿Dónde se está investigando?
- ¿Cuándo se investigó?
- ¿Por qué se está investigando?
- ¿Cómo se está investigando?

Ingrese una pregunta

Ej: Quién está investigando? i

Guardar
Cancelar

Fig. 33: Vista para crear pregunta
Fuente: Propia

Preguntas de investigación ✕

Ingrese una pregunta

¿Qué se está investigando? i

Activo

SI ▼

Guardar
Cancelar

Fig. 34: Vista para actualizar pregunta
Fuente: Propia

Buscar

Opción	Código	Pregunta	Activo
	Q1	¿Qué se está investigando?	SI
	Q2	¿Dónde se está investigando?	SI

5

Atrás
0
Siguiente

Fig. 35: Vista para listar preguntas
Fuente: Propia

- Reunión de Retrospectiva – Sprint 2

Fecha: 12/08/2019

Asistentes: Product Owner, Scrum Master y Equipo de Desarrollo

Resultado: Plan de mejora

- Plan de Mejora

Tabla 27: Plan de mejora

Plan de mejora
<p>Aciertos:</p> <ul style="list-style-type: none"> • Se investigó sobre el uso de JWT para almacenar información del usuario. • La Planificación SMS de cumple con los requerimientos. <p>Problemas:</p> <ul style="list-style-type: none"> • Falta de experiencia en el manejo de Express. <p>Mejoras:</p> <ul style="list-style-type: none"> • Capacitación en Express.

2.3.3 Sprint 3

- Reunión de Planificación

Fecha: 02/09/2019

Asistentes: Product Owner, Scrum Master y Equipo de Desarrollo

Resultado: Sprint Backlog – Sprint 3

- Sprint Backlog – Sprint 3

Tabla 28: Sprint Backlog - Sprint 3

Historias de Usuario	Nombre	Tarea	Horas
HSMS-07	Identificación de estudios - Registro de búsquedas	Codificar en el servidor GraphQL para crear, actualizar y listar registro de búsquedas	4
		Codificar vistas para crear, actualizar y listar registro de búsquedas	4
		Pruebas	2
		Corrección de errores y mejoras	2
HSMS-08	Identificación de estudios - Primer filtro	Codificar en el servidor GraphQL para crear, actualizar y listar el primer filtro	4
		Codificar vistas para crear, actualizar y listar primer filtro	4
		Pruebas	2
		Corrección de errores y mejoras	2

HSMS-09	Identificación de estudios - Filtro estudios	Codificar en el servidor GraphQL para crear, actualizar y listar filtro de estudios	4
		Codificar vistas para crear, actualizar y listar filtro de estudios	4
		Pruebas	2
		Corrección de errores y mejoras	2
Reuniones Scrum	Planificación		2
	Revisión		2
	Retrospectiva		2

- Reunión de Revisión

Fecha: 02/10/2019

Asistentes: Product Owner, Scrum Master y Equipo de Desarrollo

Resultado: Sprint Backlog revisado e incremento del producto potencialmente entregable

- Pruebas de Aceptación

Tabla 29: Pruebas de aceptación Sprint 3

Historia de Usuario	Nombre	Funcionalidad	Aceptación	
			SI	NO
HSMS-07	Identificación de estudios - Registro de búsquedas	Crear búsqueda estudios	X	
		Actualizar búsqueda estudios	X	
		Consultar búsqueda estudios	X	
		Cargar evidencia	X	
		Descargar evidencia	X	
HSMS-08	Identificación de estudios - Primer filtro	Crear primer filtro	X	
		Actualizar primer filtro	X	
		Consultar primer filtro	X	
		Cargar evidencia	X	
		Descargar evidencia	X	
HSMS-09	Identificación de estudios - Filtro estudios	Crear filtro estudios	X	
		Actualizar filtro estudios	X	
		Consultar filtro estudios	X	
		Cargar evidencia	X	
		Descargar evidencia	X	

- Incremento

Los investigadores en la identificación de estudios pueden registrar las búsquedas realizadas, los primeros filtros de cada búsqueda, los filtros de estudios y las búsquedas profundas (este último de desarrolla en el Sprint 4).

En las Fig. 36, Fig. 37 y Fig. 38 muestran las vistas para crear, actualizar y listar las búsquedas de los estudios. En las Fig. 39, Fig. 40 y Fig. 41 muestran las vistas para el primer filtro y en las Fig. 42, Fig. 43 y Fig. 44 las vistas para filtros de estudios.

Parámetros búsqueda estudios

Base de datos bibliográfica
Seleccione una

Cadena de búsqueda
Ej: GraphQL

Rango fechas
Ej: 2011-2019

Ordenado por
Ej: Ordenado por relevancia

Otros criterios
Ej: Autor

Número de documentos encontrados
0

Evidencia
Cargar evidencia No ha seleccionado archivo

Guardar **Cancelar**

Fig. 36: Vista para crear búsqueda de estudios
Fuente: Propia

Editar búsqueda estudios ✕

Base de datos bibliográfica
 ⌵ ⊗

Cadena de búsqueda
 ⊗

Rango fechas
 ⊗

Ordenado por
 ⊗

Otros criterios
 ⊗

Número de documentos encontrados
 ⊗

Evidencia
⬇ Cargar evidencia graphql (2).csv
⬆ Descargar evidencia

Activo
 ⌵

💾 Guardar
✕ Cancelar

Fig. 37: Vista para actualizar búsqueda de estudios
Fuente: Propia

Opción	Búsqueda	Base de datos	Cadena	Rango	Ordenado por	Otros	# Documentos	Evidencia	Activo
	2020-01-02 15:01:55	Scopus	API GraphQL	2015 - 2019	Relevancia	Ninguno	99	graphql (1).csv	
	2020-01-02 15:01:01	Scopus	Web service graphql	2015 - 2019	Relevancia	Ninguno	12	completo.csv	
	2020-01-02 14:01:32	Scopus	Api graphi api rest	2015 - 2019	Más citas		123	Sin evidencia	
	2020-01-02 14:01:15	IEEE Xplore	Arquitectura graphql	2015 - 2019	Relevancia	Ninguno	12	graphql (2).csv	

5
Atrás 0 Siguiente

Fig. 38: Vista para listar búsquedas de estudios
Fuente: Propia

Agregar parámetros primer filtro por búsqueda ✕

Búsqueda (Base de datos - cadena de búsqueda)

Seleccione una ⓘ

Filtro

Ej: Los 200 Más citados ⓘ

Número documentos

0 ⓘ

Evidencia

⬆️ Cargar evidencia No ha seleccionado archivo

💾 Guardar ✖️ Cancelar

Fig. 39: Vista para crear primer filtro
Fuente: Propia

Editar parámetros primer filtro por búsqueda ✕

Búsqueda (Base de datos - cadena de búsqueda)

IEEE Xplore - Arquitectura graphql ⓘ

Estudios primarios

12

Filtro

Los 10 más citados ⓘ

Número documentos

10 ⓘ

Evidencia

⬆️ Cargar evidencia Sin evidencia

💾 Guardar ✖️ Cancelar

Fig. 40: Vista para actualizar primer filtro
Fuente: Propia

Buscar

Opción	Fecha ↕	Base datos ↕	Cadena	Criterio	# Documentos	Evidencia	Activo
✎	2020-01-06 17:01:28	IEEE Xplore	Arquitectura graphql	Los 10 más citados	10	Sin evidencia	SI
✎	2020-01-06 17:01:59	Scopus	Web service graphql	10 más relevantes	10	graphql (1).csv	SI

5 ⌵

Atrás 0 Siguiente

Fig. 41: Vista para listar los primeros filtros
Fuente: Propia

Agregar filtro ✕

Tipo filtro

Criterio

Número documentos

Evidencia
 No ha seleccionado archivo

Fig. 42: Vista para crear filtro de estudios
Fuente: Propia

Editar filtro ✕

Tipo filtro

Criterio

Número documentos

Evidencia
 Sin evidencia

Fig. 43: Vista para editar filtro de estudios
Fuente: Propia

Buscar

Opción	Fecha	Tipo	Criterio	# Documentos	Evidencia	Activo
	2020-01-06 17:01:52	Exclusión	Abstract del estudio	3	Sin evidencia	
	2020-01-06 17:01:45	Inclusión	Estudios de conferencias	9	Sin evidencia	

5 Atrás Siguiente

Fig. 44: Vista para listar los filtros de estudios
Fuente: Propia

- Reunión de Retrospectiva – Sprint 3

Fecha: 02/10/2019

Asistentes: Product Owner, Scrum Master y Equipo de Desarrollo

Resultado: Plan de mejora

- Plan de Mejora

Tabla 30: Plan de mejora

Plan de mejora
<p>Aciertos:</p> <ul style="list-style-type: none"> • Se cumplió con las tareas planteadas. <p>Problemas:</p> <ul style="list-style-type: none"> • Problemas en el manejo de la caché con Apollo Client. <p>Mejoras:</p> <ul style="list-style-type: none"> • Investigar soluciones para corregir los errores con el manejo de la caché.

2.3.4 Sprint 4

- Reunión de Planificación

Fecha: 03/10/2019

Asistentes: Product Owner, Scrum Master y Equipo de Desarrollo

Resultado: Sprint Backlog – Sprint 4

- Sprint Backlog – Sprint 4

Tabla 31: Sprint Backlog - Sprint 4

Historias de Usuario	Nombre	Tarea	Horas
HSMS-10	Identificación de estudios - Búsqueda profunda	Codificar en el servidor GraphQL para crear, actualizar y listar búsqueda profunda	4
		Codificar vistas para crear, actualizar y listar búsqueda profunda	4
		Pruebas	2
		Corrección de errores y mejoras	2
HSMS-11	Extracción y clasificación - Importar	Codificar en el servidor GraphQL para importar documentos	4
		Codificar vistas para importar documentos	4

		Pruebas	2
		Corrección de errores y mejoras	2
HSMS-12	Extracción y clasificación - Campos personalizados	Codificar en el servidor GraphQL para crear, actualizar y listar campos personalizados	4
		Codificar vistas para crear, actualizar y listar de campos personalizados	4
		Pruebas	2
		Corrección de errores y mejoras	2
Reuniones Scrum	Planificación		2
	Revisión		2
	Retrospectiva		2

- Reunión de Revisión

Fecha: 04/11/2019

Asistentes: Product Owner, Scrum Master y Equipo de Desarrollo

Resultado: Sprint Backlog revisado e incremento del producto potencialmente entregable

- Pruebas de Aceptación

Tabla 32: Pruebas de aceptación Sprint 4

Historia de Usuario	Nombre	Funcionalidad	Aceptación	
			SI	NO
HSMS-10	Identificación de estudios - Búsqueda profunda	Crear búsqueda profunda	X	
		Actualizar búsqueda profunda	X	
		Consultar búsqueda profunda	X	
		Cargar evidencia	X	
		Descargar evidencia	X	
HSMS-11	Extracción y clasificación - Importar	Importar documentos	X	
HSMS-12	Extracción y clasificación - Campos personalizados	Crear campos personalizados	X	
		Actualizar campos personalizados	X	
		Consultar campos personalizados	X	

- Incremento

La búsqueda profunda es el último pase de la identificación de estudios y es opcional. En las Fig. 45, Fig. 46 y Fig. 48 se muestra los vistas para crear, actualizar y listar las búsquedas profundas.

The screenshot shows a form titled "Parámetros búsqueda profunda" with a close button (X) in the top right corner. The form contains the following fields and controls:

- Estrategia:** A text input field containing "Estrategia de búsqueda" with a red circular icon containing a white 'i' on the right.
- Descripción:** A text input field containing "Descripción de la estrategia" with a red circular icon containing a white 'i' on the right.
- Número de documentos:** A text input field containing "0" with a red circular icon containing a white 'i' on the right.
- Evidencia:** A green button labeled "Cargar evidencia" with an upward arrow icon, followed by the text "No ha seleccionado archivo".
- Buttons:** A blue button labeled "Guardar" with a floppy disk icon and a red button labeled "Cancelar" with an 'X' icon.

Fig. 45: Vista para crear búsqueda profunda
Fuente: Propia

The screenshot shows a form titled "Editar búsqueda profunda" with a close button (X) in the top right corner. The form contains the following fields and controls:

- Estrategia:** A text input field containing "Bola de nieve" with a red circular icon containing a white 'i' on the right.
- Descripción:** A text input field containing "Ir de documento en documento hasta encontrar nuevos estudios." with a red circular icon containing a white 'i' on the right.
- Número de documentos:** A text input field containing "10" with a red circular icon containing a white 'i' on the right.
- Evidencia:** A green button labeled "Cargar evidencia" with an upward arrow icon, followed by the text "Sin evidencia". Below it is a blue button labeled "Descargar evidencia" with a downward arrow icon.
- Activo:** A dropdown menu showing "SI".
- Buttons:** A green button labeled "Guardar" with a floppy disk icon and a red button labeled "Cancelar" with an 'X' icon.

Fig. 46: Vista para actualizar búsqueda profunda
Fuente: Propia

Opción	Búsqueda	Estrategia	Descripción	# Documentos	Evidencia	Activo
	2020-01-06 17:01:30	Bola de nieve	Ir de documento en documento hasta encontrar nuevos estudios.	10	Sin evidencia	

5

Atrás 0 Siguiente

Fig. 47: Vista para listar búsquedas profundas
Fuente: Propia

El investigador en la extracción y clasificación de datos puede importar los estudios desde Mendeley, registrar campos personalizados.

En la Fig. 48 se muestra la vista importar estudios.

Importar estudios

Seleccione una carpeta

Desconectar

Importar Cancelar

Fig. 48: Vista para importar estudios
Fuente: Propia

En las Fig. 49, Fig. 50 y Fig. 51 se muestra las vistas para crear, actualizar y listar los campos personalizados.

Agregar campos personalizados

Campos Preguntas a responder

Nombre campo Q1 - ¿Qué se está invest

Crear campo

Guardar Cancelar

Fig. 49: Vista para crear campos personalizados
Fuente: Propia

Editar campos personalizados

Campos Preguntas a responder

País del primer autor Q2 - ¿Dónde se está inv

Guardar Cancelar

Fig. 50: Vista para actualizar campo personalizado
Fuente: Propia



Fig. 51: Vista para listar los campos personalizados
Fuente: Propia

- Reunión de Retrospectiva – Sprint 4

Fecha: 04/11/2019

Asistentes: Product Owner, Scrum Master y Equipo de Desarrollo

Resultado: Plan de mejora

- Plan de Mejora

Tabla 33: Plan de mejora

Plan de mejora

Aciertos:

- Se cumplió con las tareas planteadas.

Problemas:

- Falta de conocimiento en la API de Mendeley provocó errores en la obtención de documentos.

Mejoras:

- Investigar más sobre la API de Mendeley y solucionar problemas de autorización.
-

2.3.5 Sprint 5

- Reunión de Planificación

Fecha: 05/11/2019

Asistentes: Product Owner, Scrum Master y Equipo de Desarrollo

Resultado: Sprint Backlog – Sprint 5

- Sprint Backlog – Sprint 5

Tabla 34: Sprint Backlog - Sprint 5

Historias de Usuario	Nombre	Tarea	Horas
HSMS-13	Extracción y clasificación - Completar información	Codificar en el servidor GraphQL completar información	4
		Codificar vistas para completar información	4
		Pruebas	2
		Corrección de errores y mejoras	2
HSMS-14	Extracción y clasificación - Exportar CSV	Codificar vistas exportar CSV	2
		Pruebas	2
		Corrección de errores y mejoras	2
HSMS-15	Extracción y clasificación - Reporte	Codificar vistas reporte PDF	4
		Pruebas	2
		Corrección de errores y mejoras	2
Reuniones Scrum	Planificación		2
	Revisión		2
	Retrospectiva		2

- Reunión de Revisión

Fecha: 06/12/2019

Asistentes: Product Owner, Scrum Master y Equipo de Desarrollo

Resultado: Sprint Backlog revisado e incremento del producto potencialmente entregable

- Pruebas de Aceptación

Tabla 35: Pruebas de aceptación Sprint 5

Historia de Usuario	Nombre	Funcionalidad	Aceptación	
			SI	NO
HSMS-13	Extracción y clasificación - Completar información	Completar información documento	X	
HSMS-14	Extracción y clasificación - Exportar CSV	Exportar archivo CSV	X	

- Incremento

Continuando con la extracción y clasificación de datos el investigador puede completar la información de los estudios, exportar los estudios en un archivo CSV y generar un reporte PDF del proceso realizado en el SMS.

La Fig. 52 muestra la lista de los estudios importados desde Mendeley para completar la información. En la Fig. 53 se muestra la vista para completar la información del estudio seleccionado.

Buscar							
Opción	Título	Autores	Año	Añadido	Importado	Completo	Publicado en
	An empirical analysis of GraphQL API schemas in open code repositories and package registries	Kim, Yun Wan; Consens, Mariano P.; Hartig, Olaf	2019	2019-11-28	2020-01-06 19:01:55		CEUR Workshop Proceedings
	Efficient Architecture Design for Software as a Service in Cloud Environments	Seda, Pavel; Masek, Pavel; Sedova, Jindriska; Seda, Milos; Krejci, Jan; Hasek, Jiri	2019	2019-11-28	2020-01-06 19:01:55		International Congress on Ultra Modern Telecommunications and Control Systems and Workshops
	Experiences on Migrating RESTful Web Services to GraphQL	Vogel, Maximilian; Weber, Sebastian; Zirpins, Christian	2018	2019-11-28	2020-01-06 19:01:53		Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)

Fig. 52: Vista para listar los estudios
Fuente: Propia

Detalles documento ✕

Básico ● Más Preguntas Campos personalizados

Título
An empirical analysis of GraphQL API schemas in open code repositories and package registries

Autores
Kim, Yun Wan; Consens, Mariano P.; Hartig, Olaf

Palabras clave
Ingrese palabras clave

Resumen
GraphQL is a query language for APIs that has been increasingly adopted by Web developers since its

Año
2019

Editor / Base de datos
Ingrese editor / base de datos
Ej: Scopus, IEEE Xplore

Publicado en
CEUR Workshop Proceedings

Comentarios
Ingrese comentario o comentarios

Guardar
Cancelar

Fig. 53: Vista para completar la información
Fuente: Propia

En la Fig. 54 se muestra vista para exportar los datos de los estudios en un archivo CSV y en la Fig. 55 un ejemplo del archivo exportado.

Exportar a CSV



Fig. 54: Vista para exportar CSV
Fuente: Propia

	A	B	C	D	E	F	G	H	I	J	K
1	TITLE	ABSTRACT	ACCESSED	AUTHORED	AUTHORS	CHAPTER	CITATION KE	CITY	CODE	PAÍS DEL PRII	REGIÓN-Q2
2	An empirical GraphQL	is a query language for APIs		Kim, Yun Wan; Consens, Mariano P.; Hartig, Olaf							
3	Efficient Arc4	With the increasing popularity of Softw		Seda, Pavel; Masek, Pavel; Sedova, Jindriska; Seda, Milos; Krejci, Jan; Hosek, Jiri							
4	Experiences	Web service APIs are central hubs of m		Vogel, Maximilian; Weber, Sebastian; Zirpins, Christian							
5	GraphQLLD: The Linked Open Data cloud	has the po		Taelman, Ruben; Vander Sande, Miel; Verborgh, Ruben							
6	Migrating to GraphQL is a novel query language	proj		Brito, Gleison; Mombach, Thais; Valente, Marco Tulio							
7	Performance GraphQL is a new concept in building	a		Hartina, Dewi Ayu; Lawi, Armin; Panggabean, Benny Leonard Enrico							
8	Towards a U Web APIs have become first-class	citize		Rodriguez-Echeverria, Roberto; Cánovas Izquierdo, Javier Luis; Cabot, Jordi							
9	Web-based c	There is a quiet revolution in the IT-are		Xingyue, Yu; Wiedemann, Thomas; Krug, Wilfried							

Fig. 55: Ejemplo archivo CSV exportado
Fuente: Propia

En las Fig. 56 se muestra lista de los SMS para generar el reporte, en la Fig. 57 un breve resumen del proceso realizado en el SMS y en la Fig. 58 y Fig. 59 se muestra un ejemplo en archivo PDF del reporte generado.

Opción	Título	Descripción
	GraphQL	Test de graphql sms
	REST API	SMS de API REST

5 Atrás 0 Siguiente

Fig. 56: Lista de los SMS para generar reporte
Fuente: Propia

Reporte ✕

SMS título:
GraphQL

Definición de preguntas de investigación:
¿Qué se está investigando? (Q1)
¿Dónde se está investigando? (Q2)

Búsquedas realizadas:
Base de datos: IEEE Xplore
Cadena de búsqueda: Arquitectura graphql
Documentos encontrados: 12
Rango fechas: 2015 - 2019
Ordenado por: Relevancia
Otros criterios: Ninguno
Primer filtro por búsqueda: Los 10 más citados
Documentos después del primer filtro: 10

Base de datos: Scopus
Cadena de búsqueda: Api graphl api rest
Documentos encontrados: 123

Generar PDF
Aceptar

Fig. 57: Breve resumen del proceso SMS
Fuente: Propia

Proceso Systematic Mapping Study (SMS)
Proyecto: GraphQL Ultimo estado
Fecha: 2020-01-08 14:01:01

Fase 1: Planificación

1.1 Base de datos bibliográficas

Base de datos
IEEE Xplore
Scopus

1.2 Preguntas de investigación

Pregunta	Código	Activo
¿Qué se está investigando?	Q1	SI
¿Dónde se está investigando?	Q2	SI

Fase 2: Identificación de estudios

2.1 Búsqueda de estudios

Fecha	Base de datos	Cadena	Rango	Orden	Otro criterio	# Doc.	Activo
2020-01-02 14:01:15	IEEE Xplore	Arquitectura graphql	2015 - 2019	Relevancia	Ninguno	12	SI
2020-01-02 14:01:32	Scopus	Api graphl api rest	2015 - 2019	Más citas		123	SI
2020-01-02 15:01:01	Scopus	Web service graphql	2015 - 2019	Relevancia	Ninguno	12	SI
2020-01-02 15:01:55	Scopus	API GraphQL	2011-2019	Relevancia	Ninguno	99	SI

2.2 Primer filtro por búsqueda

Fig. 58: Ejemplo de reporte generado – Parte 1
Fuente: Propia

Fecha	Base de datos - Cadena	Criterio	# Doc. antes filtro	# Doc. después filtro	Activo
2020-01-06 17:01:28	IEEE Xplore - Arquitectura graphql	Los 10 más citados	12	10	SI
	Scopus - Api graphl api rest	No se ha realizado un primer filtro.	123	No se ha realizado un primer filtro.	NO
2020-01-06 17:01:59	Scopus - Web service graphql	10 más relevantes	12	10	SI
	Scopus - API GraphQL	No se ha realizado un primer filtro.	99	No se ha realizado un primer filtro.	NO

Fecha	Tipo	Criterio	# Doc. resultantes	Activo
2020-01-06 17:01:45	Inclusión	Estudios de conferencias	9	SI
2020-01-06 17:01:52	Exclusión	Abstract del estudio	3	SI

Fecha	Estrategia	Descripción	# Doc.	Activo
2020-01-06 17:01:30	Bola de nieve	Ir de documento en documento hasta encontrar nuevos estudios.	10	SI

Fase 3: Extracción y clasificación de datos

#	Título	Autores	Año	DOI
1	An empirical analysis of GraphQL API schemas in open code repositories and package registries	Kim, Yon Wan; Consens, Mariano P.; Hwang, Daeh	2019	
2	Efficient Architecture Design for Software as a Service in Cloud Environments	Seda, Pavel; Masek, Pavel; Sedova, Jindriska; Seda, Milos; Krejci, Jan; Hisek, Jiri	2019	10.1109/ICUMT.2018.8631237
3	Experiences on Migrating RESTful Web Services to	Vogel, Maximilian; Weber, Sebastian;	2018	10.1007/978-3-319-91764-1_23

Fig. 59: Ejemplo de reporte generado - Parte 2
Fuente: Propia

- Reunión de Retrospectiva – Sprint 5

Fecha: 06/12/2019

Asistentes: Product Owner, Scrum Master y Equipo de Desarrollo

Resultado: Plan de mejora

- Plan de Mejora

Tabla 36: Plan de mejora

Plan de mejora

Aciertos:

- Se cumplió con las tareas planteadas.

Problemas:

- Falta de experiencia en el reporte en archivos PDF.

Mejoras:

- Presentar la información más detallada en el reporte PDF.

2.4 Acta Entrega Recepción

Luego de haber desarrollado todas las historias de usuario del Product Backlog en la sección 2.3, se obtuvo como resultado un producto software terminado y revisado, el cual será entregado al Product Owner del proyecto mediante una acta entrega recepción.



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

ACTA DE ENTREGA RECEPCIÓN

TEMA: SISTEMA WEB DE AUTOMATIZACIÓN DEL MAPEO SISTEMÁTICO

En las instalaciones de la Universidad Técnica del Norte de la Ciudad de Ibarra, el día 18 de febrero del 2020, Edwin Alfredo Bastidas Bastidas hace la entrega del sistema web "Automatización del mapeo sistemático", al Ing. Antonio Quiña, Product Owner del proyecto.

El sistema web "Automatización del mapeo sistemático", ofrece las funcionalidades para gestionar los mapeos sistemáticos y administración de usuarios y bases de datos bibliográficas.

El sistema fue desarrollado conforme a los requisitos funcionales solicitados por los interesados del sistema.

Productos entregados:

- Proyecto de desarrollo de software (Código Fuente).
- Implementación del proyecto en el servidor de la carrera.
- Manuales: técnico y de usuario.
- Pruebas de funcionalidad, operatividad con la tecnología asignada.
- Capacitación al funcionario encargado.

Atentamente, ENTREGA CONFORME

Edwin Bastidas
Tesista
UNIVERSIDAD TÉCNICA DEL NORTE

RECIBE CONFORME,

MSc. Antonio Quiña
Docente tutor
UNIVERSIDAD TÉCNICA DEL NORTE

MSc. Antonio Quiña
DOCENTE FICA UTN

CAPÍTULO 3

Validación de Resultados

En este capítulo se estructura conforme a lo que se muestra en el siguiente gráfico, ver Fig. 60



Fig. 60: Estructura capítulo 3
Fuente: Propia

3.1 Definición del Modelo de Calidad en Uso

El modelo de calidad en uso se definió en conjunto con el Product Owner y el Scrum Master. Conforme indica la ISO/IEC 25010 se identificó las características y subcaracterísticas más importantes en el contexto que se desarrolló el proyecto (ISO/IEC 25010, 2015). Las características escogidas son: eficacia, eficiencia y satisfacción, a las cuales se asignó un peso dependiendo la necesidad y perspectiva de negocio, ver Tabla 37.

Tabla 37: Definición modelo de calidad en uso

MODELO DE CALIDAD EN USO			
Características	Subcaracterísticas	Peso caract.	Peso subcaract.
Eficacia	Tareas completas	42%	16%
	Objetivos logrados		16%
	Los errores en una tarea		10%
Eficiencia	Tiempo de tareas	32%	16%
	Eficiencia del Tiempo		16%
Satisfacción	Utilidad	26%	10%
	Confianza		8%
	Comodidad		8%
	Total	100%	100%

3.2 Medición del Modelo de Calidad en Uso

Para realizar la medición de la calidad en uso, se basó en las ISO/IEC 25022 (ISO/IEC 25022, 2016), que muestra la manera como medir cada una de las subcaracterísticas del modelo de calidad definido en la Tabla 37. Para obtener los elementos de las métricas se estableció dos instrumentos de levantamiento de datos: (i) taller práctico para levantar datos de eficacia y eficiencia de calidad en uso; (ii) encuesta SUS¹³ para medir la satisfacción del usuario.

3.2.1 Definición de la Muestra

La muestra¹⁴ se seleccionó mediante un muestreo no probabilístico por conveniencia, seleccionando a dos grupos: 25 estudiantes de la Carrera de Ingeniería en Sistemas Computacionales y 15 de la Carrera de Software de la Universidad Técnica el Norte.

3.2.2 Taller Práctico

El taller tuvo lugar en los laboratorios de computó de los Facultad de Ingeniería en Ciencias Aplicadas el día 28 de enero de 2020, el cual tuvo una duración de 1 hora y 45 minutos en cada uno de los grupos seleccionados. A continuación, se detalla el diseño y la ejecución del taller.

- Diseño

El taller práctico consistió en establecer tareas (requisitos funcionales), para utilizar las principales funcionalidades del sistema web para la automatización del mapeo sistemático, ver Tabla 38.

Tabla 38: Objetivos y tareas del taller

Nro.	Objetivos	Tareas
1	Registro como investigador e inicio de sesión	Registro Inicio sesión
2	Crear SMS y realizar la Fase 1 del SMS	Crear SMS Seleccionar base de datos Crear pregunta
3	Realizar la Fase 2 del SMS	Registrar búsqueda Registrar primer filtro Registrar filtros estudios

¹³ System Usability Scale: (Escala de Usabilidad del Sistema) herramienta metodológica para medir usabilidad.

¹⁴ Parte de una población.

- Ejecución del Taller

- Pre-requisitos

- a) Para usar el sistema web SMS se debe tener una cuenta en Mendeley y acceso a bases de datos bibliográficas, en este taller se utilizó Scopus.
 - b) Se realizó una capacitación del uso de las herramientas Mendeley y Scopus y también se realizó una explicación del proceso del mapeo sistemático.

- Realización del Taller

Sujetos: Se tomó en cuenta las personas definidas en la muestra de la población establecida en la Sección 3.2.1.

Objetivos: Medir eficiencia y eficacia de la calidad en uso.

Método:

- a) Se estableció una lista de tareas que son parte de los requisitos funcionales del sistema web SMS (Diseño del taller), las cuales están divididas en 4 objetivos.
 - b) Se repartió la lista con las tareas, y una tabla de medición de tiempos y completitud a los sujetos del taller (Anexo A).
 - c) Los sujetos ejecutaron las tareas definidas con las instrucciones establecidas (Anexo B) y midieron en tiempo y completitud de los requisitos solicitados en el taller.
 - d) Los sujetos registraron los resultados en una tabla medición y reportaron quejas presentadas en la utilización del sistema.

3.2.3 Encuesta SUS

La encuesta se la realizó después de ejecutar el taller, la cual tuvo una duración de 15 minutos. A continuación, se detalla el diseño y la ejecución de la encuesta.

- Diseño de la Encuesta SUS

SUS (*System Usability Scale* - Escala de Usabilidad del Sistema) propone 10 preguntas para medir la usabilidad de un sistema, las cuales están en una escala Likert¹⁵, donde 1

¹⁵ Escala de medición para medir el nivel de acuerdo o desacuerdo de una pregunta.

significa *Muy en desacuerdo* y 5 *Muy de acuerdo* (Fabio Devin, 2017). Estas preguntas permitieron levantar datos para medir las subcaracterísticas utilidad y comodidad, donde las preguntas P1, P6 y P9 fueron seleccionadas inicialmente para la utilidad y las preguntas P2, P3, P4, P5, P7, P8 y P9 para la comodidad. A continuación, se detalla el enunciado de cada una de ellas:

P1: ¿Considera usted qué usaría este sistema frecuentemente?

P2: ¿Considera usted qué este sistema es innecesariamente complejo?

P3: ¿Considera usted qué el sistema fue fácil de usar?

P4: ¿Considera usted qué necesitaría ayuda de una persona con conocimientos técnicos para usar este sistema?

P5: ¿Considera usted qué las funciones de este sistema están bien integradas?

P6: ¿Considera usted qué el sistema es inconsistente?

P7: ¿Considera usted qué la mayoría de la gente aprendería a usar este sistema en forma rápida?

P8: ¿Considera usted qué el sistema es difícil de usar?

P9: ¿Se siente confiado al usar el sistema?

P10: ¿Considera usted qué necesita aprender muchas cosas tecnológicas antes de usar el sistema?

- Ejecución de la Encuesta

- a) La encuesta se diseñó en Forms de Office 365.

- b) Después del taller los sujetos ejecutaron la encuesta.

- Observaciones

El 97,5% de los encuestados respondieron la encuesta y el 2,5% no respondió.

3.3 Evaluación Estadística de los Instrumentos de Medición

3.3.1 Fiabilidad del Taller

Luego de la ejecución del taller y la tabulación de los datos (Anexo C) se procedió a verificar la fiabilidad del taller utilizando el método estadístico coeficiente Alfa de Cronbach, el cual nos permite conocer que tan fiable es un instrumento de medida, en este caso el taller. Este coeficiente tiene una escala de 0 a 1, donde los valores: (i) mayores a 0,9 son excelentes, (ii) entre 0,8 y 0,9 son buenos, (iii) mayores o iguales a 0,7 son aceptables y (iv) menores de 0,5 a 0 poseen una pobre fiabilidad (Juan Mendoza, 2018).

Para el análisis se creó una matriz de los datos tabulados con las columnas: tareas intentadas, tareas completadas, objetivos completados, errores en una tarea, tiempo tareas, relación experto, objetivos alcanzados, número quejas y confianza. Esa matriz se exportó en un archivo CSV (Anexo D) y se ejecutó el algoritmo de Alfa de Cronbach (Anexo E) en el entorno de desarrollo RStudio¹⁶. En la Fig. 61 se puede observar el resultado del Alfa de Cronbach del taller el cual dio como resultado 0,89 lo cual quiere decir que la fiabilidad del taller es buena.

```
Reliability analysis
Call: psych::alpha(x = setValidacion)

raw_alpha std.alpha G6(smc) average_r S/N ase mean sd median_r
0.89      0.89      0.81      0.81 8.5 0.033 0.45 0.16 0.81
```

Fig. 61: Alfa de Cronbach del taller
Fuente: Propia

3.3.2 Fiabilidad de la Encuesta SUS

Después de obtener los resultados de la encuesta SUS y la tabulación de los datos (Anexo F) se procedió a usar la técnica estadística Análisis Factorial Confirmatorio (AFC)¹⁷ para verificar la fiabilidad de la encuesta, el AFC valida el instrumento (encuesta) y estructura factorial (factores: utilidad, comodidad). El análisis se lo ejecutó en RStudio (Anexo G) con la matriz de los resultados de la encuesta y se conoció la correlación de las preguntas. En la Fig. 62 se puede observar que existe una correlación entre las preguntas P1, P3, P5, P7, P8 y P9 en el primer factor ML1 (utilidad) porque existe una correlación mayor a 0,3. Y para el segundo factor ML2 (comodidad) las preguntas P2, P4, P6, P8 y P10 cumplen con la correlación de ser mayor a 0,3.

	ML1	ML2
P1	0.73	-0.08
P2	-0.05	0.46
P3	0.77	-0.07
P4	0.02	0.70
P5	0.74	0.02
P6	0.11	0.52
P7	0.87	0.07
P8	0.39	0.47
P9	0.73	-0.03
P10	-0.11	0.75

Fig. 62: Correlación de las preguntas
Fuente: Propia

¹⁶ IDE para el lenguaje de programación R, para la computación estadística y gráficos.

¹⁷ Evalúa un posible modelo de medición.

Las preguntas seleccionadas inicialmente (Sección 3.2.3) para la utilidad fueron P1, P6 y P9, por lo cual se descartó las preguntas P3, P5, P7 y P8 a pesar de que existía una correlación para este factor, y la pregunta P1 también fue descartada porque generaba niveles bajos para obtener una encuesta viable; después de este análisis las preguntas seleccionadas finales para la utilidad fueron P6 y P9. Las preguntas seleccionadas inicialmente para la comodidad fueron P2, P3, P4, P5, P7, P8, P9, de las cuales fueron tomadas en cuenta para la selección final las preguntas P2, P4, P8 y P10, puesto que están si tienen una correlación entre sí y las demás preguntas (P3, P5, P7, P9) ya fueron descartadas.

Luego de seleccionar las preguntas se calculó el índice CFI (Comparative Fit Index)¹⁸ y el índice TLI (Tucker y Lewis)¹⁹ con el mismo algoritmo del Análisis Factorial Confirmatorio utilizado en la sección anterior, para conocer el resultado final y a partir de los índices determinar si la encuesta es fiable. La escala de CFI se define de la siguiente manera: si es mayor o igual a 0,95 se considera bueno o si es mayor o igual a 0,90 es aceptable; con TLI ocurre lo mismo (Tóth-Király et al., 2017). Los índices obtenidos para la encuesta SUS fueron: CFI 0,94 y TLI 0,90 lo cual quiere decir que la **encuesta es fiable y valida**.

3.4 Evaluación del Modelo de Calidad en Uso

Para evaluar el modelo de calidad, primero se tabuló los datos obtenidos de la ejecución del taller práctico (Anexo C) y encuesta SUS (Anexo F), luego se aplicó las métricas establecidas en la Tabla 1. A continuación, se detalla las mediciones del modelo establecido.

3.4.1 Característica: Eficacia

- Subcaracterística: Tareas Completadas

$$\text{Fórmula: } X = \frac{A}{B}$$

Donde,

- A = número de tareas únicas completadas = 465 (sumatoria de todas las tareas completadas por los 40 sujetos)
- B = número de tareas únicas intentadas = 480 (sumatoria de todas las tareas intentadas por los sujetos 12*40)

Remplazando en la formula se obtiene: $X = \frac{465}{480} = 0,97$. Lo cual quiere decir que el 97% de las tareas fueron completadas por los sujetos.

¹⁸ Muestra que tan bueno es un modelo, valores próximos a 1 son fiables.

¹⁹ Comparación entre el modelo propuesto y uno que no tiene relación entre variables. Valor superior a 0.9 es aceptable.

- Subcaracterística: Objetivos Alcanzados

Cada sujeto tenía 4 objetivos y cada objetivo tiene varias tareas para completar, ver Tabla 38. Si una tarea no fue completada, el objetivo correspondiente a esa tarea también se lo consideró no completo. Por lo cual en la tabulación del taller se obtuvo 155 objetivos totales alcanzados por los sujetos, de los 160 (40*4) objetivos esperados en la ejecución de taller práctico. Por lo cual se obtuvo un 97% (0,97) de los objetivos alcanzados.

- Subcaracterística: Errores en Una Tarea

$$\text{Formula: } X = 1 - \frac{A}{B}$$

Donde,

- A = errores en las tareas completadas = 20 (sumatoria de todos los errores en tareas completadas, los errores cometidos en tareas no completadas no se tomaron en cuenta)
- B = número de tareas únicas intentadas = 480

Remplazando en la formula se obtiene: $X = 1 - \frac{20}{480} = 0,96$. Con ese resultado se puede decir que el 96% de las tareas intentadas no tienen errores.

3.4.2 Característica: Eficiencia

- Subcaracterística: Tiempo de Tareas

$$\text{Formula: } X = \frac{A}{B}$$

Donde,

- A = tiempo empleado por un usuario experto al realizar una tarea
- B = tiempo empleado por un usuario normal al realizar una tarea

Esta fórmula se la aplicó para cada uno de los sujetos que usaron el sistema y luego se realizó el promedio de todas las relaciones, como resultado se alcanzó 0,78. Lo cual quiere decir que el sujeto tiene 78% de eficiencia comparado con un usuario experto al momento de completar las tareas.

- Subcaracterística: Eficiencia del Tiempo

$$\text{Formula: } X = \frac{A}{B}$$

Donde,

- A = tiempo empleado por un usuario experto al completar un objetivo.
- B = tiempo empleado por un usuario normal al completar un objetivo.

Esa relación se la aplicó para cada uno de los sujetos que usaron el sistema y luego se realizó el promedio de todas las relaciones, se obtuvo 0,80. Lo cual quiere decir que la eficiencia del sujeto al completar los objetivos es del 80% comparado con un usuario experto.

3.4.3 Característica: Satisfacción

- Subcaracterística: Utilidad

Las preguntas de la encuesta SUS seleccionadas para recolectar datos de medición de la “Utilidad” luego de la evaluación de fiabilidad de la encuesta (Sección 3.3.2) fueron: P6 y P9. Utilizando la escala de Likert se estableció pesos a las respuestas dadas por los sujetos, ver Tabla 39.

Tabla 39: Peso respuestas

Escala	Respuesta	Peso
5	Muy de acuerdo	1
4	Algo de acuerdo	0,8
3	Ni de acuerdo ni en desacuerdo	0,6
2	Algo desacuerdo	0,4
1	Muy en desacuerdo	0,2

Luego se calculó las respuestas conforme a la escala establecida y para el resultado se sumó el total de cada pregunta, ver Tabla 40.

Tabla 40: Resultados SUS - Utilidad

Pregunta	Sujetos satisfechos
P6	27,6 de 39
P9	27,4 de 39
Total	27,50 de 39

Para obtener el valor total de los sujetos satisfechos se calculó el promedio de las dos preguntas.

$$\text{Formula: } X = \frac{A}{B}$$

Donde,

- A = usuarios satisfechos (ver Tabla 40) = 27,50
- B = usuarios encuestados = 39

Remplazando en la formula se obtiene: $X = \frac{27,50}{39} = 0,71$. Lo que quiere decir que el 71% de los sujetos estuvieron satisfechos al usar el sistema web.

- Subcaracterística: Confianza

Formula: $X = 1 - \% C$, $C = \frac{A}{B}$

Donde,

- A = número de reclamos = 12
- B = número de usuarios = 40
- C= % reclamos

Reemplazando en la ecuación se obtiene: $C = \frac{12}{40} = 0,3$, $X = 1 - 0,3 = 0,70$. Lo cual quiere decir que el nivel de confianza de los sujetos es de 70%.

- Subcaracterística: Comodidad

Las preguntas de la encuesta SUS que sirvió para recolectar datos de medición de la “Comodidad” fueron: P2, P4, P8 y P10 (detalladas en la Sección 3.2.3), en la Tabla 41 se muestra el resultado de la ejecución de la encuesta.

Tabla 41: Resultados SUS - Comodidad

Escala	Peso escala	Usuarios	Satisfacción
Muy de acuerdo	1	5	0,128
Algo de acuerdo	0,8	15	0,308
Ni de acuerdo ni en desacuerdo	0,6	15	0,231
Algo desacuerdo	0,4	3	0,031
Muy en desacuerdo	0,2	1	0,005

Formula: $X = A + B + C + D + E$

Donde,

- A = muy de acuerdo = 0,128
- B = algo de acuerdo = 0,308
- C = ni de acuerdo ni en desacuerdo = 0,231
- D = algo en desacuerdo = 0,031
- E = muy en desacuerdo = 0,005

Reemplazando en la formula se obtiene: $X = 0,128 + 0,308 + 0,231 + 0,031 + 0,005 = 0,7$. Lo que quiere que el 70% de usuarios están cómodos al usar el sistema.

3.5 Resultados del Modelo de Calidad en Uso

En la Tabla 42 se puede observar los resultados de cada una de las características, subcaracterísticas y el nivel de calidad en uso total del sistema web.

Tabla 42: Resultados calidad en uso

Tabla de resultados						
Características	Subcaracterísticas	Peso categoría	Peso subcat.	Medición	Resultado	Resultado Característica
Eficacia	Tareas completas	42%	16%	0,97	15,50%	40,58%
	Objetivos logrados		16%	0,97	15,50%	
	Los errores en una tarea		10%	0,96	9,58%	
Eficiencia	Tiempo de tareas	32%	16%	0,78	12,51%	25,25%
	Eficiencia del Tiempo		16%	0,80	12,74%	
Satisfacción	Utilidad	26%	10%	0,71	7,05%	18,27%
	Confianza		8%	0,70	5,60%	
	Comodidad		8%	0,70	5,62%	
	Total	100%				84,11%

El resultado final de la evaluación del sistema web para la automatización del SMS arrojó 84,11% (8,41) en calidad en uso, lo que quiere decir que la calidad en uso del sistema web para la automatización del SMS está dentro del rango objetivo y es satisfactoria según la escala de medición establecida en la Fig. 14. En la Fig. 63 se muestra el valor obtenido en la escala de medición.

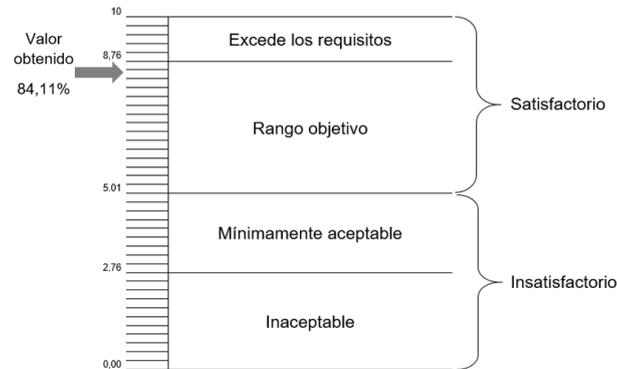


Fig. 63: Resultado en la escala de medición
Fuente: Basada en (ISO/IEC 14598-1, 1999)

Con los resultados obtenidos en cada una de las características se puede concluir que la satisfacción de los usuarios al usar el sistema web es aceptable y pueden usarlo sin asistencia de un experto y también pueden cumplir correctamente con los objetivos planteados. Además de tener una eficiencia aceptable en comparación con un usuario experto.

CONCLUSIONES

- El marco teórico que se desarrolló en el Capítulo 1, permitió fundamentar conceptualmente el proceso del mapeo sistemático de literatura (SMS), la tecnología usada y la metodología ágil empleada en el desarrollo del sistema web para la automatización del SMS y también la base conceptual de la ISO/IEC 25000 para evaluar el sistema web en calidad en uso.
- Con el desarrollo del sistema web para la automatización del SMS se puede hacer un aporte de mejora al proceso de investigación de los docentes de la Universidad Técnica del Norte mediante la automatización de la metodología de investigación SMS, en donde el sistema web se encuentra en la nube y se puede acceder mediante el internet. Además, el investigador puede registrar el proceso en cada una de las fases del SMS de manera organizada y genera un conjunto de datos de los estudios para su análisis. También genera un reporte del proceso realizado el cual puede ser compartido a otros investigadores. Cabe mencionar que cuenta con la integración de Mendeley lo cual ayuda tener una mejor organización de las referencias bibliográficas.
- La metodología ágil Scrum utilizada en el desarrollo del sistema web permitió tener un proceso iterativo e incremental, desarrollando cada iteración (Sprint) de forma organizada para cumplir con las actividades planificadas; obteniendo así un producto final entregable.
- También se pudo observar y verificar que el uso de la base de datos NoSQL Neo4j y el lenguaje de consulta y entorno de ejecución GraphQL reducen esfuerzos de programación; debido a que la combinación de estas dos tecnologías simplifica el desarrollo en los tipos de componentes de manipulación de datos como: *queries*, *mutations*; ya que generan automáticamente las funciones básicas de la utilización de los componentes antes descritos.
- El uso de GraphQL permitió generar una API que posee un sistema de tipo de datos, *queries* y *mutations* las cuales pueden ser usadas por un cliente *front-end* y acceder a los datos generados. Con la utilización de esta tecnología se puede hacer consultas específicas de los datos que expone el API GraphQL.
- La evaluación de la calidad en uso del sistema web desarrollado, se realizó con un marco de trabajo basado en las normas ISO/IEC 25000 (requisitos y evaluación de sistemas y calidad de software - *SQuaRE*), específicamente las ISO/IEC 25010 (modelos de calidad del sistema y software), ISO/IEC 25022 (medición de calidad en uso), ISO/IEC 25040 (proceso de evaluación). A partir de esto se estableció un modelo de calidad (Tabla 37), con el cual se tomó como base para medición de las funcionalidades del sistema web, que se evaluó con los lineamientos de las normas utilizadas. Dando como resultado final

de la evaluación del modelo de calidad con el 84,11% de cumplimiento de los requisitos de calidad esperados, lo que corresponde a una calificación de “*Satisfactorio*” del rango objetivo. En conclusión, se puede decir que la utilización del marco de trabajo utilizado para la evaluación de calidad en uso es muy recomendable ya que se pudo identificar y reportar el resultado de la interacción cuando el software es usado.

RECOMENDACIONES

- Para redactar el marco teórico consultar en las bases de datos de citas y resúmenes bibliográficos como por ejemplo Scopus para tener una buena base en los fundamentos teóricos, además de apoyarse en las herramientas de gestión de referencias bibliográficas para llevar una mejor organización; una de ellas es Mendeley la cual cuenta con una aplicación de escritorio y web.
- Para realizar un SMS (basado en las pautas de Petersen) se recomienda seguir las instrucciones detalladas en las fases y realizar las búsquedas en diferentes bases de datos bibliográficas para así abarcar un mayor número de estudios.
- Para reducir esfuerzos de programación utilizar herramientas que se complementen para simplificar el desarrollo de operaciones, como por ejemplo la base de datos Neo4j y el lenguaje de consulta y entorno de ejecución GraphQL; las cuales cuentan con librerías que permiten el desarrollo automático de componentes que son utilizados para el manejo de datos.
- Para llevar un orden en el desarrollo de las actividades planificadas en el desarrollo de un sistema web utilizar la metodología ágil Scrum, la cual permite entregar cada cierto tiempo un incremento (producto parcialmente entregable), además de realizar reuniones periódicas para planificar las actividades a desarrollar, revisar el incremento y realizar una retrospectiva para generar un plan de mejora para aplicar en el siguiente Sprint.
- Evaluar un producto software con las ISO/IEC 25000 en calidad en uso para conocer el nivel de calidad que tiene el software y así darle un valor adicional al producto. Las normas ISO/IEC 25000 ofrecen una guía para el diseño, medición y evaluación de la calidad en uso, aunque no son tan descriptivas y no cuentan con ejemplos prácticos, por lo cual al momento de evaluar con las normas ISO se recomienda apoyarse en trabajos donde han utilizado estas normas y así tener una guía y un ejemplo más claro.

GLOSARIO TEMAS

CQL: Lenguaje de consulta declarativo.

CSV: Es un documento de texto que almacena datos en forma de tabla.

DOM: Interfaz y lenguaje independiente de plataforma que permite a los programas y scripts acceder y actualizar dinámicamente el contenido, la estructura y el estilo del documento.

Framework: Marco de trabajo.

Front-end: Tecnologías que están a lado del cliente, navegador web.

I/O: Periférico de Entrada/Salida que permite interacción de forma bidireccional.

ISO: Organización encargada para la creación de estándares internacionales.

Javascript: Lenguaje de programación imperativo más omnipresentes en la actualidad en el mundo de la computación, basado en objetos.

JSON: Formato de texto para intercambiar información.

Keywords: Palabras clave.

Likert: Escala de medición para medir el nivel de acuerdo o desacuerdo de una pregunta.

Muestra: Parte de una población.

NoSql: No utilizan SQL como el principal lenguaje para hacer consultas.

Paper: Artículo científico publicado en una revista especializada.

SUS: System Usability Scale por sus siglas en inglés, es herramienta metodológica para medir usabilidad.

XML: Lenguaje de marcado extensible: sirve para la organización de y etiquetado de documentos.

BIBLIOGRAFÍA

- Acosta, J. (2018). Guía rápida para aprender Scrum | OpenWebinars. Retrieved May 13, 2019, from <https://openwebinars.net/blog/la-guia-para-aprender-scrum/>
- Apollo. (2019). Introduction - Apollo Docs. Retrieved May 5, 2019, from <https://www.apollographql.com/docs/react/>
- Aucancela, M., Naranjo, M., Vizuete, K., & Betún, J. (2018). MAPEO SISTEMATICO DE LITERATURA DE UN DATA LAKE, 17. Retrieved from <http://revistas.espech.edu.ec/index.php/mktDESCUBRE/article/view/77771>
- Berenguel Gómez, J. L. 1981-. (2015). *Desarrollo de aplicaciones web en el entorno servidor, UF1844*. Paraninfo.
- Bernstein, P. A., & Newcomer, E. (2009). *Principles of transaction processing*. Morgan Kaufmann Publishers.
- Bootstrap. (2019). Bootstrap · The most popular HTML, CSS, and JS library in the world. Retrieved February 20, 2019, from <https://getbootstrap.com/>
- Bryant, M. (2017). GraphQL for archival metadata: An overview of the EHRI GraphQL API. In *2017 IEEE International Conference on Big Data (Big Data)* (pp. 2225–2230). IEEE. <https://doi.org/10.1109/BigData.2017.8258173>
- CHUQUIMARCA ARANHA, S. E. (2017). *ESTUDIO EMPÍRICO DE LOS FACTORES TECNOLÓGICOS QUE INCIDEN EN LA PRODUCCIÓN JUST IN TIME DE LAS ROSAS MEDIANTE LA APLICACIÓN DE UN SYSTEMATIC MAPPING STUDY*. Sangolquí. Retrieved from <https://repositorio.espe.edu.ec/bitstream/21000/13557/1/T-ESPE-053899.pdf>
- Endrei, M., & IBM. (2004). *Patterns : service-oriented architecture and web services*. IBM Corp., International Technical Support.
- Fabio Devin. (2017). Sistema de Escalas de Usabilidad: ¿qué es y para qué sirve? | UXpañol. Retrieved February 3, 2020, from <https://uxpanol.com/teoria/sistema-de-escalas-de-usabilidad-que-es-y-para-que-sirve/>
- Facebook. (2018). GraphQL. Retrieved December 18, 2018, from <https://facebook.github.io/graphql/June2018/#sec-Overview>
- Fielding, R., & Taylor, R. (2000). Architectural styles and the design of network-based software architectures. Retrieved from https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf
- Francesco, P. Di, Malavolta, I., & Lago, P. (2017). Research on Architecting Microservices: Trends, Focus, and Potential for Industrial Adoption. In *2017 IEEE International Conference on Software Architecture (ICSA)* (pp. 21–30). IEEE. <https://doi.org/10.1109/ICSA.2017.24>

- Galindo, J. A., Benavides, D., Trinidad, P., Gutiérrez-Fernández, A.-M., & Ruiz-Cortés, A. (2018). Automated analysis of feature models: Quo vadis? *Computing*, 1–47. <https://doi.org/10.1007/s00607-018-0646-1>
- Ghebremicael, E. S. (2017). *Transformation of REST API to GraphQL for OpenTOSCA*. Retrieved from [https://elib.uni-stuttgart.de/bitstream/11682/9369/1/Master Thesis_Transformation of REST API to GraphQL for OpenTOSCA.pdf](https://elib.uni-stuttgart.de/bitstream/11682/9369/1/Master_Thesis_Transformation_of_REST_API_to_GraphQL_for_OpenTOSCA.pdf)
- Goralski, W., & Goralski, W. (2017). Hypertext Transfer Protocol. *The Illustrated Network*, 661–684. <https://doi.org/10.1016/B978-0-12-811027-0.00026-6>
- Hahn, E. M. (2016). *Express in Action: Writing, building, and testing Node.js applications*. (Manning Publications, Ed.). Retrieved from www.it-ebooks.info
- Harrington, M., Cross, M., Harrington, M., & Cross, M. (2015). KML/XML/HTML. *Google Earth Forensics*, 49–68. <https://doi.org/10.1016/B978-0-12-800216-2.00004-0>
- Hart, G. (1996). The Five W's: An Old Tool for the New Task of Task Analysis, 43. Retrieved from <https://www.ingentaconnect.com/content/stc/tc/1996/00000043/00000002/art00004>
- ISO/IEC 14598-1. (1999). Tecnologías de la información - Evaluación del producto de software - Parte 1: Visión general.
- ISO/IEC 25000. (2014). SOFTWARE ENGINEERING -- SOFTWARE PRODUCT QUALITY REQUIREMENTS AND EVALUATION (SQuaRE) -- GUIDE TO SQuaRE.
- ISO/IEC 25010. (2015). SYSTEMS AND SOFTWARE ENGINEERING — SYSTEMS AND SOFTWARE QUALITY REQUIREMENTS AND EVALUATION (SQuaRE) — SYSTEM AND SOFTWARE QUALITY MODELS.
- ISO/IEC 25022. (2016). SYSTEMS AND SOFTWARE ENGINEERING — SYSTEMS AND SOFTWARE QUALITY REQUIREMENTS AND EVALUATION (SQuaRE) — Measurement of quality in use.
- ISO 25000. (2019). NORMAS ISO 25000. Retrieved May 8, 2019, from <https://iso25000.com/index.php/normas-iso-25000?limit=4&limitstart=0>
- Jalali, S., & Wohlin, C. (2012). *Systematic Literature Studies: Database Searches vs. Backward Snowballing*. Retrieved from <http://scholar.google.com/intl/en/scholar/about.html>
- Juan Mendoza. (2018). RPubs - Alfa de Cronbach - Psicometría con R. Retrieved February 6, 2020, from https://rpubs.com/jboscomendoza/alfa_cronbach_r
- Kimokoti, B. (2018). *Beginning GraphQL : fetch data faster and more efficiently whilst improving the overall performance of your web application*. (Packt Publishing, Ed.).
- Kitchenham, B., & Charters, S. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=F4AD5761DAF1810658701A>

CBA65EE715?doi=10.1.1.117.471&rep=rep1&type=pdf

- Klein, U., & Namjoshi, K. S. (2011). Formalization and Automated Verification of RESTful Behavior (pp. 541–556). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-22110-1_43
- Krylovskiy, A., Jahn, M., & Patti, E. (2015). Designing a Smart City Internet of Things Platform with Microservice Architecture. In *2015 3rd International Conference on Future Internet of Things and Cloud* (pp. 25–30). IEEE. <https://doi.org/10.1109/FiCloud.2015.55>
- Luján Mora, S. (2001). *Programación en Internet: cliente web*. Club Universitario. Retrieved from <http://rua.ua.es/dspace/handle/10045/16994>
- Macchi, D., & Solari, M. (2012). *Mapeo sistemático de la literatura sobre la Adopción de Inspecciones de Software*. Retrieved from <https://pdfs.semanticscholar.org/7dc1/85c7638fb792e167519613d067e8dcf418e4.pdf>
- Mendeley. (2019). How To Use The API - Mendeley Developer Portal. Retrieved April 20, 2019, from https://dev.mendeley.com/overview/how_to_use_the_api.html
- Mesbah, A. (2015). Advances in Testing JavaScript-Based Web Applications. *Advances in Computers*, 97, 201–235. <https://doi.org/10.1016/BS.ADCOM.2014.12.003>
- Naciones Unidas. (2016). Agenda 2030 y los Objetivos de Desarrollo sostenible. Una oportunidad para América Latina y el Caribe. [https://doi.org/10.1016/0950-4230\(91\)80011-I](https://doi.org/10.1016/0950-4230(91)80011-I)
- Neosoft. (2018). ¿Qué es una aplicación Web? – Blog Neosoft Sistemas. Retrieved January 8, 2020, from <https://www.neosoft.es/blog/que-es-una-aplicacion-web/>
- Pan, Z., & Kosicki, G. (1993). Framing analysis: An approach to news discourse. *Political Communication*, 10(1), 55–75. <https://doi.org/10.1080/10584609.1993.9962963>
- Perçuku, A., Minkovska, D., & Stoyanova, L. (2017). Modeling and Processing Big Data of Power Transmission Grid Substation Using Neo4j. *Procedia Computer Science*, 113, 9–16. <https://doi.org/10.1016/J.PROCS.2017.08.276>
- Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic Mapping Studies in Software Engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering* (pp. 68–77). Retrieved from <https://www.researchgate.net/publication/228350426>
- Petersen, Kai, Vakkalanka, S., & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64, 1–18. <https://doi.org/10.1016/J.INFSOF.2015.03.007>
- Poulter, A. J., Johnston, S. J., & Cox, S. J. (2015). Using the MEAN stack to implement a RESTful service for an Internet of Things application. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)* (pp. 280–285). IEEE. <https://doi.org/10.1109/WF->

IoT.2015.7389066

- Prisma. (2018). Top 5 Reasons to Use GraphQL. Retrieved December 18, 2018, from <https://www.prisma.io/blog/top-5-reasons-to-use-graphql-b60cfa683511>
- Rodgers, E., & Barbrow, S. (2013). A Look at Altmetrics and Its Growing Significance to Research Libraries. Retrieved from <https://deepblue.lib.umich.edu/handle/2027.42/99709>
- Rodriguez-Echeverria, R., Cánovas Izquierdo, J. L., & Cabot, J. (2018). Towards a UML and IFML Mapping to GraphQL. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 10544 LNCS, pp. 149–155). https://doi.org/10.1007/978-3-319-74433-9_13
- Roldán Martínez, D., Valderas Aranda, P. J., & Pastor López, O. (2010). *Aplicaciones web : un enfoque práctico*. Ra-Ma.
- Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide™ The Definitive Guide to Scrum: The Rules of the Game*. Retrieved from <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>
- Scrum Manager. (2014). Historia de usuario - Scrum Manager BoK. Retrieved February 4, 2020, from https://www.scrummanager.net/bok/index.php/Historia_de_usuario
- Seda, P., Masek, P., Sedova, J., Seda, M., Krejci, J., & Hosek, J. (2018). Efficient Architecture Design for Software as a Service in Cloud Environments. *2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 317–322. <https://doi.org/10.1109/ICUMT.2018.8631237>
- Spurlock, J. (2013). *Bootstrap*. O'Reilly Media. Retrieved from https://books.google.com.ec/books?id=LZm7Cxxgi3aQC&dq=bootstrap+css&lr=&hl=es&source=gbs_navlinks_s
- Tilkov, S., & Vinoski, S. (2010). Node.js: Using JavaScript to Build High-Performance Network Programs. *IEEE Internet Computing*, 14(6), 80–83. <https://doi.org/10.1109/MIC.2010.145>
- Tóth-Király, I., Orosz, G., Dombi, E., Jagodics, B., Farkas, D., & Amoura, C. (2017). Cross-cultural comparative examination of the Academic Motivation Scale using exploratory structural equation modeling. *Personality and Individual Differences*, 106, 130–135. <https://doi.org/10.1016/j.paid.2016.10.048>
- Ungan, E., Cizmeli, N., & Demirors, O. (2014). Comparison of Functional Size Based Estimation and Story Points, Based on Effort Estimation Effectiveness in SCRUM Projects. In *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications* (pp. 77–80). IEEE. <https://doi.org/10.1109/SEAA.2014.83>
- Vipul A. M., & Sonpatki, P. (2016). *ReactJS by example : building modern web applications*

with React: get up and running with ReactJS. (Packt Publishing, Ed.). Retrieved from https://books.google.com.ec/books?id=Ht3JDAAAQBAJ&dq=reactjs&lr=&hl=es&source=gbs_navlinks_s

Vogel, M., Weber, S., & Zirpins, C. (2018). Experiences on Migrating RESTful Web Services to GraphQL. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 10797 LNCS, pp. 283–295). https://doi.org/10.1007/978-3-319-91764-1_23

Wagh, K., & Thool, R. (2012). A Comparative Study of SOAP Vs REST Web Services Provisioning Techniques for Mobile Host. *Journal of Information Engineering and Applications*, 2(5), 12–16. Retrieved from <https://www.iiste.org/Journals/index.php/JIEA/article/view/2063>

Zhu, Z., Zhou, X., & Shao, K. (2019). A novel approach based on Neo4j for multi-constrained flexible job shop scheduling problem. *Computers & Industrial Engineering*, 130, 671–686. <https://doi.org/10.1016/J.CIE.2019.03.022>

ANEXOS

Anexo A. Taller Práctico

Taller para el uso del sistema para la automatización del mapeo sistemático de literatura -
SMS

NOMBRE: _____ FECHA: _____

Objetivo: Utilizar el sistema para la automatización del mapeo sistemático de literatura – SMS para realizar las siguientes tareas:

- **Registro como investigador e inicio de sesión**
 - Registro / Inicio de sesión
- **Crear SMS y realizar la fase 1**
 - Crear SMS / Seleccionar base de datos, Crear pregunta
- **Realizar la fase 2 (no cargar evidencia)**
 - Registrar búsqueda / Registrar primer filtro / Registrar filtros estudios / (Omitir paso 4)
- **Realizar la fase 3**
 - Importar estudios de la carpeta GraphQL / Crear campo personalizado / Completar información documento / Exportar CSV

Registro y seguimiento

#	Objetivo	Tarea	Completo (x)	Hora inicio (hh:mm)	Hora fin (hh:mm)
1	Registro como investigador e inicio de sesión	Registro			
		Inicio sesión			
2	Crear SMS y realizar la Fase 1	Crear SMS			
		Seleccionar base de datos			
		Crear pregunta			
3	Realizar la Fase 2	Registrar búsqueda			
		Registrar primer filtro			
		Registrar filtros estudios			
4	Realizar la Fase 3	Importar estudios			
		Crear campo personalizado			
		Completar información			
		Exportar CSV			

Quejas

Nro. tarea: _____ queja: _____

Observaciones: _____

Anexo B. Instrucciones Taller

Taller para el uso del sistema para la automatización del mapeo sistemático de literatura - SMS

Objetivo: Utilizar el sistema para la automatización del mapeo sistemático de literatura – SMS para realizar las siguientes tareas:

Nota: Las tareas *** no se las realiza desde el sistema web.

- **Registro como investigador e inicio de sesión**
 - Registro
 - Inicio de sesión
- **Crear SMS y realizar la Fase 1**
 - Crear SMS
 - *Título: Graphql*
 - *Descripción: Sms para el taller*
 - Fase 1 - Paso 1: Seleccionar base de datos
 - *Scopus*
 - Fase 1 – Paso 2: Crear pregunta
 - *¿Dónde se está investigando graphql?*
- **Realizar la fase 2 (no cargar evidencia)**
 - ***Realizar búsqueda en Scopus con los siguientes parámetros:
 - ***Cadena: *Graphql*
 - ***Rango fechas: *2016-2019*
 - ***Ordenado por: *Más citas*
 - Fase 2 – Paso 1: Registrar búsqueda estudios
 - Base de datos: *Scopus*
 - Cadena: *Graphql*
 - Rango fechas: *2016-2019*
 - Ordenado por: *Más citas*
 - Otros criterios: (opcional)
 - Número de documentos: (Ingresar número de documentos encontrados en Scopus)
 - Fase 2 – Paso 2: Registrar primer filtro
 - ***Nota: *Crear carpeta en Mendeley: Graphql (Si crea en Mendeley Desktop sincronizar)*
 - ***Importar a Mendeley los 5 primeros estudios a la carpeta Graphql y luego registrar en el primer filtro en el sistema web del SMS el filtro y número de documentos. Sincronizar en Mendeley.

- Filtro: 5 más citados.
 - Número de documentos. 5
- Fase 2– Paso 3: Registrar filtros estudios
 - Tipo: *Exclusión*
 - Criterio: *Título que no tenga la palabra GraphQL*
 - ****Eliminar estudios en Mendeley (carpeta GraphQL) que no tengan la palabra GraphQL en el título. Después de eliminar sincronizar.*
 - *Número de documentos (ingresar el número de documentos restantes)*
- **Omitir paso 4 de la Fase 2**
- **Realizar la fase 3**
 - Fase 3 – Paso 1: Importar estudios de la carpeta *GraphQL*
 - Fase 3 – Paso 1: Crear campo personalizado
 - País primer autor
 - Seleccionar pregunta: ¿Dónde se está investigando GraphQL?
 - Fase 3 – Paso 3: Completar información documento
 - **Nota:** Obligatorio completar la información en las pestañas Básico y Campos personalizados.
 - **Nota 2:** En el campo personalizado *País del primer autor* ingresar *España*
 - Fase 3 – Paso 4: Exportar CSV.

Anexo C. Tabulación del Taller

Eficacia				Eficiencia					
Total tareas intentadas	Tareas completadas	Objetivos completados	Errores en una tarea	Tiempo taller (min)	Tiempo tareas (min)	Relación exp	Objetivos alcanzados	Nro. Quejas	Confianza (al menos una queja)
12	12	100%	1	40	36	72%	4	1	1
12	12	100%	1	38	38	68%	4	0	0
12	12	100%	0	43	26	100%	4	0	0
12	12	100%	1	42	42	62%	4	0	0
12	12	100%	0	40	38	68%	4	0	0
12	12	100%	0	41	41	63%	4	0	0
12	12	100%	0	31	32	81%	4	0	0
12	12	100%	0	41	39	67%	4	0	0
12	12	100%	0	35	35	74%	4	0	0
12	12	100%	0	29	28	93%	4	0	0
12	12	100%	0	35	28	93%	4	1	1
12	12	100%	2	31	27	96%	4	1	1
12	12	100%	0	27	26	100%	4	0	0
12	12	100%	0	25	25	104%	4	0	0
12	12	100%	0	35	35	74%	4	0	0
12	8	75%	1	31	30	57%	3	1	1

12	8	50%	1	30	22	77%	2	0	0
12	12	100%	0	39	29	90%	4	2	1
12	12	100%	1	47	45	58%	4	0	0
12	12	100%	4	52	49	53%	4	0	0
12	8	75,00%	1	15	13	77%	3	0	0
12	12	100%	0	44	44	59%	4	1	1
12	12	100%	0	41	41	63%	4	0	0
12	12	100%	1	47	39	67%	4	0	0
12	12	100%	0	37	37	70%	4	0	0
12	12	100%	0	52	46	57%	4	1	1
12	12	100%	0	40	37	70%	4	0	0
12	12	100%	4	31	31	84%	4	1	1
12	12	100%	1	48	33	79%	4	0	0
12	12	100%	0	41	41	63%	4	0	0
12	12	100%	0	40	35	74%	4	0	0
12	12	100%	3	44	24	108%	4	1	1
12	12	100%	0	38	29	90%	4	0	0
12	12	100%	1	28	27	96%	4	1	1
12	12	100%	0	33	32	81%	4	0	0
12	12	100%	0	34	29	90%	4	0	0
12	12	100%	0	37	26	100%	4	2	1
12	12	100%	0	27	26	100%	4	0	0

12	12	100%	0	35	24	108%	4	0	0
12	9	75,00%	1	52	45	40%	3	1	1
480	465	97%	24	1496	33,25	78%	155	14	12

Anexo D. Matriz Taller (CSV)

TotalIntentadas,TareasCompletadas,ObjetivosCompletados,ErroresEnUnaTarea ,TiempoTareas,RelacionExperto,ObjetivosAlcanzados,NumeroQuejas,Confianza ,Errores

12,12,1,1,36,0.722222222,4,1,1,Si
12,12,1,1,38,0.684210526,4,0,0,Si
12,12,1,0,26,1,4,0,0,No
12,12,1,1,42,0.619047619,4,0,0,Si
12,12,1,0,38,0.684210526,4,0,0,No
12,12,1,0,41,0.634146341,4,0,0,No
12,12,1,0,32,0.8125,4,0,0,No
12,12,1,0,39,0.666666667,4,0,0,No
12,12,1,0,35,0.742857143,4,0,0,No
12,12,1,0,28,0.928571429,4,0,0,No
12,12,1,0,28,0.928571429,4,1,1,No
12,12,1,2,27,0.962962963,4,1,1,Si
12,12,1,0,26,1,4,0,0,No
12,12,1,0,25,1.04,4,0,0,No
12,12,1,0,35,0.742857143,4,0,0,No
12,8,0.75,1,30,0.566666667,3,1,1,Si
12,8,0.5,1,22,0.772727273,2,0,0,Si
12,12,1,0,29,0.896551724,4,2,1,No
12,12,1,1,45,0.577777778,4,0,0,Si
12,12,1,4,49,0.530612245,4,0,0,Si
12,8,0.75,1,13,0.772727273,3,0,0,Si
12,12,1,0,44,0.590909091,4,1,1,No
12,12,1,0,41,0.634146341,4,0,0,No
12,12,1,1,39,0.666666667,4,0,0,Si
12,12,1,0,37,0.702702703,4,0,0,No
12,12,1,0,46,0.565217391,4,1,1,No
12,12,1,0,37,0.702702703,4,0,0,No
12,12,1,4,31,0.838709677,4,1,1,Si
12,12,1,1,33,0.787878788,4,0,0,Si
12,12,1,0,41,0.634146341,4,0,0,No
12,12,1,0,35,0.742857143,4,0,0,No
12,12,1,3,24,1.083333333,4,1,1,Si
12,12,1,0,29,0.896551724,4,0,0,No
12,12,1,1,27,0.962962963,4,1,1,Si
12,12,1,0,32,0.8125,4,0,0,No
12,12,1,0,29,0.896551724,4,0,0,No
12,12,1,0,26,1,4,2,1,No
12,12,1,0,26,1,4,0,0,No
12,12,1,0,24,1.083333333,4,0,0,No
12,9,0.75,1,45,0.4,3,1,1,Si

Anexo E. Algoritmo Alfa de Cronbach

```
library(readr)
master = read_csv("matrizTaller.csv")
View(master)
setValidacion=master[,c(5,6)]
# la variable 6 estaba en relación inversa más tiempo menos RelacionExperto
# se invirtió la escala resto de 1
setValidacion$RelacionExperto=1-setValidacion$RelacionExperto
# la variable 5 tiempo en minutos le escalamos div tiempo max
setValidacion$TiempoTareas=setValidacion$TiempoTareas/max(setValidacion$TiempoT
areas)
psych::alpha(setValidacion)
```

Anexo F. Tabulación Encuesta SUS

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
3	4	5	4	4	4	4	4	3	4
2	3	2	3	3	3	2	3	2	1
4	2	3	3	3	3	4	3	3	3
3	3	4	2	5	4	3	4	4	4
3	1	2	4	4	2	3	3	2	4
4	3	4	2	3	3	4	4	5	5
3	1	3	3	3	4	3	1	3	3
3	2	3	2	3	4	3	3	3	4
4	2	4	4	4	5	4	4	4	4
4	3	4	4	5	3	4	2	4	3
4	2	4	3	4	3	3	2	5	3
3	3	4	5	4	5	4	4	4	5
3	3	3	3	3	4	2	5	3	4
4	2	5	3	5	4	4	5	3	2
4	4	3	5	5	5	5	4	4	5
3	3	5	4	5	4	5	5	5	5
2	3	4	3	4	3	4	4	4	3
4	5	4	4	4	5	5	5	4	4
5	2	4	1	4	2	4	2	4	2
3	3	3	5	2	3	2	4	3	4
4	2	3	4	4	3	4	2	3	4
3	4	1	2	3	3	2	4	2	5
2	2	2	4	1	4	1	2	2	5
4	2	2	3	3	2	3	2	4	2
3	1	1	1	3	3	1	2	3	2
4	3	4	4	5	4	3	4	3	4
5	3	4	5	5	5	4	5	5	3

5	5	2	5	5	5	5	5	5	5
2	4	2	4	2	5	2	3	2	4
3	3	3	4	4	3	4	4	3	4
4	3	5	1	5	3	5	4	3	3
4	3	4	3	5	2	5	2	5	3
1	5	1	4	4	3	1	1	1	5
5	1	4	5	5	4	5	5	5	5
2	4	2	2	3	2	3	4	4	2
3	3	4	2	4	3	4	2	4	2
3	2	2	3	3	5	2	3	4	4
4	1	4	2	5	5	4	2	4	2
3	2	3	4	3	3	4	2	3	4

Anexo G. Análisis Factorial Confirmatorio

```
library(readr)
master = read_csv("encuestaSUS.csv")
View(master)
summary(master)
library(lavaan)
library(semPlot)
summary(master)
apply(master,2,sd, na.rm=T)
boxplot(master)
mahal=mahalanobis(master, colMeans(master,na.rm=T),
                  cov(master, use="pairwise.complete"))
mahal
summary(mahal)
cutoff=qchisq(1-0.001, ncol(master))
summary(mahal<cutoff)
noout=subset(master, mahal<cutoff)
boxplot(noout)
correl=cor(noout, use="pairwise.complete.obs")
View(correl)
symnum(correl)
library(corrplot)
corrplot(correl, method = "circle")
random=rchisq(nrow(noout),7)
fake=lm(random ~. ,data=noout) #lm (linear model)
fake
standardized=rstudent(fake)
fitted=scale(fake$fitted.values)
hist(standardized)
qqnorm(standardized)
abline(0,1)
plot(fitted,standardized)
abline(0,0)
abline(v=0)
library(psych)
library(GPArotation)
```

```

cortest.bartlett(correl,n=nrow(noout))
KMO(correl)
screplot(princomp(noout,cor=TRUE),type="lines",
          main=deparse(substitute(Teorico)))
nofactors=fa.parallel(noout,fm="ml",fa="fa")
round1=fa(noout, nfactors=2, rotate = "oblimin", fm="ml")
round1
finalmodel=fa(noout[ ,-c(9,2,4,5)], nfactors=2, rotate = "oblimin", fm="ml")
finalmodel
library(lavaan)
library(semPlot)
three.model=
  'Utilidad =~ P6 + P9
  Comodidad =~ P2 + P4 + P8 + P10'
three.fit = cfa(three.model, data = noout)
semPaths(three.fit,whatLabels="std", layout="tree", edge.label.cex= .80, mar= c(9,9,9,9))
summary(three.fit, standardized= TRUE, rsquare = TRUE)
fitMeasures(three.fit)

```