



UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

**CARRERA DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS
COMPUTACIONALES**

PROYECTO DE INVESTIGACIÓN

TEMA:

**“DESARROLLO DE UN PROTOTIPO INTELIGENTE PARA
DETERMINAR PROBLEMAS DE DEPENDENCIA EN
APLICACIONES BASADAS EN MICROSERVICIOS”**

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

AUTORES:

Alcasiga Quilumbaquin Carlos Stalin

Tarco Taipe Bryan Jhoel

TUTORA:

Ing. MSc. Tapia Cerda Verónica del Consuelo

LATACUNGA – ECUADOR

Marzo 2022

DECLARACIÓN DE AUTORÍA



Ingeniería
Informática Y Sistemas
Computacionales

DECLARACIÓN DE AUTORÍA

Nosotros, **CARLOS STALIN ALCASIGA QUILUMBAQUIN** con C.I.: 172723602-6 y **BRYAN JHOEL TARCO TAIPE** con C.I.: 050410091-8, declaramos ser los autores del presente Proyecto de Investigación: “**DESARROLLO DE UN PROTOTIPO INTELIGENTE PARA DETERMINAR PROBLEMAS DE DEPENDENCIA EN APLICACIONES BASADAS EN MICROSERVICIOS**”, siendo la Ing. MSc. Verónica del Consuelo Tapia Cerda, tutora del presente trabajo, eximio expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

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

Atentamente,

Estudiante 1
Carlos Stalin Alcasiga Quilumbaquin
C.C.: 172723602-6

Estudiante 2
Bryan Jhoel Tarco Taipe
C.C.: 050410091-8

RENUNCIA DE DERECHOS



Ingeniería
Informática Y Sistemas
Computacionales

RENUNCIA DE DERECHOS

Nosotros **CARLOS STALIN ALCASIGA QUILUMBAQUIN** con C.I.: **172723602-6** y **BRYAN JHOEL TARCO TAIBE** con C.I.: **050410091-8**, sedemos los derechos de autoría intelectual total y/o parcial del presente trabajo de titulación con el tema “**DESARROLLO DE UN PROTOTIPO INTELIGENTE PARA DETERMINAR PROBLEMAS DE DEPENDENCIA EN APLICACIONES BASADAS EN MICROSERVICIOS**”, a la Universidad Técnica de Cotopaxi.

Latacunga, marzo, 2022.

Atentamente,

Autor 1

Carlos Stalin Alcasiga Quilumbaquin

CI: 172723602-6

.....

Autor 2

Bryan Jhoel Tarco Taipe

CI: 050410091-8

AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN



Ingeniería
Informática Y Sistemas
Computacionales

AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN

En calidad de Tutor del Trabajo de Investigación con el título:

“DESARROLLO DE UN PROTOTIPO INTELIGENTE PARA DETERMINAR PROBLEMAS DE DEPENDENCIA EN APLICACIONES BASADAS EN MICROSERVICIOS”, de los estudiantes: **CARLOS STALIN ALCASIGA QUILUMBAQUIN** y **BRYAN JHOEL TARCO TAIPE**, de la carrera de Ingeniería en Informática y Sistemas Computacionales, considero que dicho Informe Investigativo cumple con los requerimientos metodológicos y aportes científico-técnicos suficientes para ser sometidos a la evaluación del Tribunal de Validación de Proyecto que el Honorable Consejo Directivo de la Facultad de Ciencias de la Ingeniería y Aplicadas de la Universidad Técnica de Cotopaxi designe para su correspondiente estudio y calificación.

Latacunga, marzo 2022

A handwritten signature in blue ink, appearing to read 'Verónica del Consuelo', written over a dotted line.

Ing. MSc. Tapia Cerda Verónica del Consuelo

C.I.: 0502053697

Tutor

APROBACIÓN DEL TRIBUNAL DE TITULACIÓN



Ingeniería
Informática Y Sistemas
Computacionales

APROBACIÓN DEL TRIBUNAL DE TITULACIÓN

En calidad de Tribunal de Lectores, aprueban el presente Informe de Investigación de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y por la Facultad de **CIENCIAS DE LA INGENIERÍA Y APLICADAS**; por cuanto, los postulantes: **CARLOS STALIN ALCASIGA QUILUMBAQUIN** y **BRYAN JHOEL TARCO TAIPE**, con el título del proyecto de investigación: **“DESARROLLO DE UN PROTOTIPO INTELIGENTE PARA DETERMINAR PROBLEMAS DE DEPENDENCIA EN APLICACIONES BASADAS EN MICROSERVICIOS”**, ha considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de Sustentación del Proyecto.

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

Latacunga, marzo 2022

.....
Lector 1 (Presidente)

Nombre: Mg. Iza Carate Miryan Dorila
C.C.: 0501957617

.....
Lector 2

Nombre: Mg. Llano Casa Alex Christian
C.C.: 0502589864

.....
Lector 3

Nombre: Dr. Chancusig Chisag Juan Carlos
C.C.: 0502275779

AGRADECIMIENTO

En primer lugar, agradezco a Dios, por darme salud y por permitir caminar junto a mis padres en cada etapa formación personal y estudiantil.

A todos los docentes de la carrera de Ingeniería en Informática Sistemas Computacionales, que durante todos estos años impartieron sus conocimientos, por la motivación y consejos para desenvolverme en el mundo profesional.

A la Ing. Verónica Tapia por el apoyo y conocimientos compartidos, por la motivación y dedicación brindada para la culminación de la tesis.

Alcasiga Carlos

A mi madre por siempre darme un ejemplo de respeto y responsabilidad hacia todas las cosas, a toda mi familia que me dieron la motivación para continuar adelante con este proyecto. A todas las personas que me guiaron y corrigieron durante el desarrollo de este proyecto. A los expertos que me ayudaron a validar el proyecto, gracias por aceptar la petición de unos desconocidos.

Tarco Bryan

DEDICATORIA

Este trabajo de titulación va dedicado a mis Padres y hermana, por su cariño, amor y apoyo constante que día a día me han ayudado a seguir adelante, cada uno de sus consejos han sido muy valiosos y que gracias a ella he logrado conseguir lo que soy hoy en día.

Alcasiga Carlos

Este trabajo está dedicado a toda mi carrera de Ingeniería en informática y Sistemas Computacionales, puesto que se aplicó todo lo aprendido durante la misma, además fue necesario una ardua investigación para lograr cumplir con los objetivos del proyecto, se revisó una gran cantidad de documentos que no fueron parte de este documento, pero que sirvieron para ayudarme a entender y aprender un sin fin de nuevos conocimientos, aplicables para esta carrera como para mi vida como persona.

Tarco Bryan

ÍNDICE GENERAL

PORTADA	i
DECLARACIÓN DE AUTORÍA	ii
RENUNCIA DE DERECHOS	iii
AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN	iv
APROBACIÓN DEL TRIBUNAL DE TITULACIÓN.....	v
AGRADECIMIENTO	vi
DEDICATORIA.....	vii
ÍNDICE GENERAL.....	viii
ÍNDICE DE TABLAS.....	xi
ÍNDICE DE FIGURAS	xiii
ÍNDICE DE FÓRMULAS	xiv
ÍNDICE DE ANEXOS	xv
RESUMEN	xvi
ABSTRACT	xvii
AVAL DE TRADUCCIÓN.....	xviii
1. INFORMACIÓN GENERAL	1
2. INTRODUCCIÓN.....	3
2.1. EL PROBLEMA	3
2.1.1. Situación Problemática.....	3
2.1.2. Formulación del problema:.....	4
2.3. OBJETO Y CAMPO DE ACCIÓN	4
2.3.1. Objeto:	4
2.3.2. Campo:	4
2.4. BENEFICIARIOS	4
2.5. JUSTIFICACIÓN.....	5
2.6. HIPÓTESIS	6
2.7. OBJETIVOS.....	6
2.7.1. General:	6
2.7.2. Específicos:.....	6
2.8. SISTEMA DE TAREAS	7
3. FUNDAMENTACIÓN TEÓRICA	8
3.1. INTELIGENCIA ARTIFICIAL.....	8

3.1.1. Definición	8
3.1.2. Característica	8
3.1.3. Aplicaciones Prácticas de la Inteligencia Artificial.....	9
3.1.4. Aplicaciones Técnicas de la Inteligencia Artificial.....	9
3.1.5. Ventajas y Desventajas	10
3.1.6. Clasificación	10
3.2. PROCESAMIENTO DE LENGUAJE NATURAL (PLN)	13
3.2.1. Definición de procesamiento de lenguaje natural.....	13
3.2.2. Aplicaciones del Procesamiento de Lenguaje Natural	13
3.2.3. Arquitectura de un sistema de Procesamiento de Lenguaje Natural	14
3.2.4. Herramientas que permiten realizar el Procesamiento de Lenguaje Natural.....	16
3.3. SPACY	19
3.3.1. Definición	19
3.3.2. Características.....	19
3.3.3. Técnica de procesamiento de datos de Spacy.....	20
3.4. APLICACIÓN	24
3.4.1. Definición	24
3.4.2. Tipos de arquitectura de Aplicaciones.....	24
3.4.3. Arquitectura Monolítica vs Arquitectura basada en Microservicios	25
3.5. CALIDAD DE SOFTWARE	27
3.5.1. Cómo obtener un software de calidad	27
3.5.2. Calidad de aplicaciones basadas en microservicios	28
3.5.3. Problemas de calidad en aplicaciones basadas en microservicios.....	28
3.6 METODOLOGÍA DE LA INVESTIGACIÓN.....	32
3.6.1. Tipo de investigación	32
3.6.2. Nivel de investigación	33
3.6.3. Diseño de la Investigación.....	33
3.6.4. Instrumentos de recolección de datos	33
3.6.5. Técnicas para el análisis de datos	34
3.7. METODOLOGÍA DE DESARROLLO.....	34
3.7.1. Metodologías de Desarrollo de Software Ágiles.....	34
4. MATERIALES Y MÉTODOS.....	39
4.1 LÍNEA DE TIEMPO DE LA INVESTIGACIÓN	39
4.2. HERRAMIENTAS DE DESARROLLO	40
4.2.1. Visual Studio Code.....	40

4.2.2. Python.....	40
4.2.3. spaCy	40
4.2.4. Django	41
4.2.5. SQLite.....	41
4.2.6. Bootstrap.....	41
4.2.7. Speech Recognition	41
4.2.8. PyAudio.....	41
4.2.9. D3	41
4.3. METODOLOGÍA.....	42
4.3.1. Metodología de la investigación.....	42
4.4. METODOLOGÍA DE DESARROLLO DE SOFTWARE	43
4.4.1. Prácticas Ágiles	43
5. ANÁLISIS Y DISCUSIÓN DE RESULTADOS	55
5.1. Análisis de resultados de acuerdo a las métricas de dependencia que permita verificar que la hipótesis se cumplió.....	55
5.2. Validación del Prototipo por Juicio de Expertos	56
5.2.1. Confiabilidad de la Validación	57
5.3. Valoración del Proyecto de Investigación	57
5.3.1. Valoración Económica	57
5.3.2. Gastos Directos del Proyecto.....	58
5.3.3. Gastos Indirectos del Proyecto	58
5.3.4. Gasto Total del Proyecto	59
6. CONCLUSIONES Y RECOMENDACIONES	60
6.1. CONCLUSIONES.....	60
6.2. RECOMENDACIONES	60
7. BIBLIOGRAFÍA	61
8. ANEXOS	65

ÍNDICE DE TABLAS

Tabla 2.1. Beneficiarios directos e indirectos.....	4
Tabla 2.2. Sistema de tareas.	7
Tabla 3.1. Lenguajes de programación, nombres de herramientas y licencia para el procesamiento de lenguaje natural [17].....	16
Tabla 3.2. “Continuación” Lenguajes de programación, nombres de herramientas y licencia para el procesamiento de lenguaje natural [17].....	17
Tabla 3.3. Ventajas y desventajas de librerías de lenguaje de programación Python para procesamiento de lenguaje natural.	17
Tabla 3.4. “Continuación” Ventajas y desventajas de librerías de lenguaje de programación Python para procesamiento de lenguaje natural.....	18
Tabla 3.5. Características y descripción de la librería de Python spaCy [18].	19
Tabla 3.6. “Continuación” Características y descripción de la librería de Python spaCy [18].....	20
Tabla 3.7. Objetos de contenedor [20].....	21
Tabla 3.8. Atributos del canal del procesamiento [20].	22
Tabla 3.9. “Continuación” Atributos del canal del procesamiento [20].....	23
Tabla 3.10. Comparadores y descripción [20].	23
Tabla 3.11. Características arquitectura monolítica vs arquitectura basada en microservicios [27].	25
Tabla 3.12. “Continuación” Características arquitectura monolítica vs arquitectura basada en microservicios [27].	26
Tabla 3.13. Formato para definir de roles del proyecto.....	35
Tabla 3.14. Formato para definir una historia de usuario.....	35
Tabla 3.15. Formato para definir Product Backlog.	36
Tabla 3.16. Formato para definir Release Plan.	37
Tabla 3.17. Formato para definir un Sprint.	37
Tabla 3.18. Formato para definir las pruebas unitarias.	38
Tabla 3.19. Formato para definir casos de prueba.....	38
Tabla 3.20. “Continuación”Formato para definir casos de prueba.....	39
Tabla 4.1. Grado de confiabilidad del Alfa de Cronbach	43
Tabla 4.2. Roles y actividades.	44
Tabla 4.3. Product Backlog.	45
Tabla 4.4. Release Plan.....	46

Tabla 4.5. “Continuación” Release Plan.....	47
Tabla 4.6. Planificación del Sprint 1.	47
Tabla 4.7. Planificación del Sprint 2.	48
Tabla 4.8. Planificación del Sprint 3.	48
Tabla 4.9. Planificación del Sprint 4.	49
Tabla 4.10. Caso de prueba 01.....	50
Tabla 4.11. Caso de prueba 02.....	51
Tabla 4.12. Caso de prueba 03.....	52
Tabla 4.13. Caso de prueba 04.....	53
Tabla 5.1. Escala de valoración del funcionamiento del Prototipo Inteligente.	56
Tabla 5.2. Resultados de las encuestas	56
Tabla 5.3. Alfa de Cronbach para la validación de expertos.....	57
Tabla 5.4. Gastos directos del proyecto.....	58
Tabla 5.5. Gastos indirectos del proyecto.....	59
Tabla 5.6. Gastos Total del Proyecto.....	59

ÍNDICE DE FIGURAS

Figura 3.1.	Diagrama de flujo del aprendizaje supervisado [6].	11
Figura 3.2.	Diagrama de flujo del aprendizaje no supervisado [6].	11
Figura 3.3.	Diagrama de flujo del aprendizaje por refuerzo [6].	12
Figura 3.4.	Técnica de procesamiento de datos de Spacy [20].	20
Figura 3.5.	Canal de procesamiento [20].	22
Figura 3.6.	Arquitectura monolítica [25].	24
Figura 3.7.	Arquitectura Monolítica y de Microservicios [28].	27
Figura 3.8.	Beneficios de los microservicios.	28
Figura 4.1.	Línea de tiempo de las actividades realizadas para el proyecto de investigación.	39
Figura 4.2.	Línea de tiempo de las actividades realizadas para el proyecto de investigación.	40
Figura 4.3.	Ingreso de la solicitud del usuario a través de voz y archivo de audio...	50
Figura 4.4.	Ingresar el usuario responsable.	50
Figura 4.5.	Lista de historias de usuario.	51
Figura 4.6.	Generar Product Backlog.	52
Figura 4.7.	Ordenar Historias de Usuario.	53
Figura 4.8.	Relación de dependencias entre microservicios.	54
Figura 4.9.	Descripción de dependencias entre microservicios.	54

ÍNDICE DE FÓRMULAS

Fórmula (3.1). Diccionario de los valores de similitud semántica	29
Fórmula (3.2). Similitud semántica Individual	29
Fórmula (3.3). Similitud semántica Total	29
Fórmula (3.4). Importancia absoluta del microservicio Individual.....	30
Fórmula (3.5). Importancia absoluta del microservicio Total.....	30
Fórmula (3.6). Dependencia absoluta del microservicio Individual	30
Fórmula (3.7). Dependencia absoluta del microservicio Total	30
Fórmula (3.8). Interdependencia de los microservicios Individual.....	30
Fórmula (3.9). Interdependencia de los microservicios Total.....	30
Fórmula (3.10). Vector métricas de dependencia	31
Fórmula (3.11). Dependencia total del sistema.....	31
Fórmula (5.1). Coeficiente del Alfa de Cronbach.....	57

ÍNDICE DE ANEXOS

Anexo A:	Datos informativos estudiante 1.	65
Anexo B:	Datos informativos estudiante 2.	66
Anexo C:	Historias de usuario	67
Anexo D:	Pruebas unitarias.....	73
Anexo E:	Estimación del proyecto a través de Puntos de Historia “SP”.....	83
Anexo F:	Formulario de validación de expertos	85
Anexo G:	Perfil de los expertos que validaron la propuesta	88
Anexo H:	Respuestas de Juicio de Expertos	89
Anexo I:	Hoja de vida de Expertos.....	92

UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

TÍTULO: “DESARROLLO DE UN PROTOTIPO INTELIGENTE PARA DETERMINAR PROBLEMAS DE DEPENDENCIA EN APLICACIONES BASADAS EN MICROSERVICIOS”.

Autores:

Carlos Stalin Alcasiga Quilumbaquin

Bryan Jhoel Tarco Taipe

RESUMEN

El presente proyecto tiene el objetivo de proponer un prototipo inteligente para determinar los problemas de dependencia en aplicaciones basadas en microservicios mediante la utilización de técnicas de procesamiento de lenguaje natural, para el cumplimiento del mismo se utiliza la metodología de investigación cuantitativa con un enfoque exploratorio, la fundamentación teórica se realizó mediante la investigación bibliográfica. Dentro de la metodología de desarrollo se aplicaron prácticas ágiles, para el desarrollo del prototipo se utilizaron las siguientes herramientas: Visual Studio Code, Python, PyAudio, Speech Recognition, spaCy para el procesamiento de lenguaje natural, D3 para generar el gráfico de dependencia, el gestor de base de datos SQLite y Bootstrap para el diseño de interfaces. El resultado principal del proyecto de investigación es el prototipo desarrollado que permite procesar los requisitos o necesidades de un usuario expresado de forma oral, con el propósito de transformar a texto y obtener las historias de usuario, posteriormente generar el Product Backlog en donde el desarrollador o analista puede priorizar de forma manual las historias de usuario; por último, muestra un gráfico donde se visualizan los microservicios y la relación de dependencia entre los mismos, esto basándose en la métrica de similitud semántica entre los sustantivos determinados para generar el gráfico de microservicios, se debe tomar en cuenta que el resultado de la métrica debe ser mayor a 0.85, en un rango de 0 a 1, donde 1 significa una similitud exacta y 0 que no existe similitud alguna. Para validar la propuesta se utilizó la estrategia del Juicio de Expertos, para lo cual se aplicó el Coeficiente del Alfa de Cronbach en donde se obtiene 0,89 que significa que el prototipo tiene un nivel alto de confiabilidad.

Palabras Claves: Arquitecturas de software, procesamiento de lenguaje natural, microservicios, dependencia, prácticas de desarrollo ágil, spaCy.

TECHNICAL UNIVERSITY OF COTOPAXI

FACULTY OF ENGINEERING AND APPLIED SCIENCES

TITLE: "DEVELOPMENT OF AN INTELLIGENT PROTOTYPE TO DETERMINE DEPENDENCY PROBLEMS IN APPLICATIONS BASED ON MICROSERVICES".

Authors:

Carlos Stalin Alcasiga Quilumbaquin

Bryan Jhoel Tarco Taipe

ABSTRACT

The objective of this project is to propose an intelligent prototype to determine dependency problems in applications based on microservices through the use of natural language processing techniques, using quantitative research methodology with an exploratory approach, the theoretical foundation was made through bibliographic research. Agile practices were applied in the development methodology; the following tools were used to develop the prototype: Visual Studio Code, Python, PyAudio, Speech Recognition, spaCy for natural language processing, D3 to generate the dependency graph, SQLite database manager and Bootstrap for interface design. The main result of the research project is the developed prototype that allows to process the requirements or needs of a user expressed orally, in order to transform them into text and obtain the user stories, then generate the Product Backlog where the developer or analyst can manually prioritize the user stories; Finally, it shows a graph where the microservices and the dependency relationship between them are displayed, based on the semantic similarity metric between the nouns determined to generate the graph of microservices, it should be taken into account that the result of the metric must be greater than 0.85, in a range from 0 to 1, where 1 means an exact similarity and 0 means that there is no similarity at all. To validate the proposal, the Expert Judgment strategy was used, for which the Cronbach's Alpha Coefficient was applied, where 0.89 was obtained, which means that the prototype has a high level of reliability.

Keywords: Software architectures, natural language processing, microservices, dependency, agile development practices, spaCy.

AVAL DE TRADUCCIÓN



CENTRO
DE IDIOMAS

AVAL DE TRADUCCIÓN

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

La traducción del resumen al idioma Inglés del proyecto de titulación cuyo título versa: **“DESARROLLO DE UN PROTOTIPO INTELIGENTE PARA DETERMINAR PROBLEMAS DE DEPENDENCIA EN APLICACIONES BASADAS EN MICROSERVICIOS”** presentado por: **Alcasiga Quilumbaquin Carlos Stalin y Tarco Taipe Bryan Jhoel**, estudiantes de la Carrera de **Ingeniería en Informática y Sistemas Computacionales** perteneciente a la **Facultad de Ciencias de la Ingeniería y Aplicadas**, lo realizaron bajo mi supervisión y cumple con una correcta estructura gramatical del Idioma.

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

Latacunga, 17 marzo del 2022

Atentamente,



CENTRO
DE IDIOMAS

Mg. Marco Paúl Beltrán Semblantes

DOCENTE CENTRO DE IDIOMAS-UTC
CI: 0502666514

1. INFORMACIÓN GENERAL

Título del proyecto: “Desarrollo de un prototipo inteligente, para determinar problemas de dependencia, en aplicaciones basadas en microservicios”

Fecha de inicio: 25 de octubre de 2021.

Fecha de finalización: marzo 2022.

Lugar de ejecución: Carrera Sistemas de Información de la Universidad Técnica de Cotopaxi.

Facultad que auspicia: Facultad de Ciencias de la Ingeniería y Aplicadas.

Carrera que auspicia: Ingeniería en Informática y Sistemas Computacionales.

Proyecto de investigación vinculado:

Aplicación del modelo Iterativo - Incremental en el desarrollo de Herramientas Informáticas para instituciones, organizaciones y empresas del entorno educativo, productivo y comercial de la Provincia de Cotopaxi. Proyecto vinculado a los estudiantes de la carrera de Ingeniería en Informática y Sistemas Computacionales.

Equipo de Trabajo:

COORDINADOR:

Nombre: Verónica Consuelo Tapia Cerda

Nacionalidad: Ecuatoriana

E-mail: veronica.tapia@utc.edu.ec

ESTUDIANTE 1:

Nombre: Bryan Jhoel Tarco Taipe

Nacionalidad: Ecuatoriana

Fecha de Nacimiento: 22 de marzo de 1999.

Estado Civil: Soltero

Residencia: Parroquia Juan Montalvo, calle Quijano y Ordoñez.

Correo: bryan.tarco0918@utc.edu.ec

Teléfono: 0984725989

ESTUDIANTE 2:

Nombre: Carlos Stalin Alcasiga Quilumbaquin

Nacionalidad: Ecuatoriana

Fecha de Nacimiento: 13 de diciembre de 1997.

Estado Civil: Soltero

Residencia: Barrio San Felipe, Av. Simón Rodríguez y México S/N.

Correo: carlos.alcasig6026@utc.edu.ec

Celular: 0967459735

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

Línea de investigación:

Tecnologías de la Información y Comunicación (TICs).

Sub línea de investigación de la Carrera:

Ciencias Informáticas para la modelación de Sistemas de Información a través del desarrollo de software.

2. INTRODUCCIÓN

2.1. EL PROBLEMA

2.1.1. Situación Problemática

El desarrollo de aplicaciones de software conlleva una serie de pasos y etapas muy importantes, una de las más vitales es la etapa inicial de análisis y diseño, puesto que los errores que aparecen aquí crecen a medida que se desarrolla el proyecto. Uno de los inconvenientes más comunes y difíciles de detectar en las etapas iniciales es la dependencia que existe entre los microservicios de una aplicación, puesto que, si un microservicio es muy dependiente de otro, producirá un error de manera global en la aplicación en producción puesto que uno necesita del otro para poder funcionar.

Según [1], analiza cómo estructurar el código de procedimiento para reducir la dependencia y complejidad. En la primera página del libro, escribe: “Hay mucho más en el tema de la dependencia y complejidad que simplemente intentar minimizar la complejidad local de cada parte de un programa. Un tipo de complejidad mucho más importante es la complejidad global: la complejidad de la estructura general de un programa o sistema (es decir, el grado de asociación o interdependencia entre las partes principales de un programa).” Basándose en el contexto anterior, la complejidad local es la complejidad de cada microservicio individual, mientras que la complejidad global es la complejidad de todo el sistema. La complejidad local depende de la implementación de un servicio; La complejidad global está definida por las interacciones y dependencias entre los servicios.

En cuanto al proyecto de investigación se pretende realizar un prototipo inteligente el mismo que permita convertir el relato de las necesidades del usuario/cliente, a través de algoritmos de inteligencia artificial para el procesamiento de lenguaje natural y reconocimiento de voz, con el objetivo de transformar la mayoría de estos relatos en historias de usuario donde se describen el autor, la acción que realiza y el objetivo que se espera conseguir. Como una opción para ayudar a los equipos de desarrollo de software a tener una solución general en el diseño de la una aplicación basada en microservicios. En resumen, la escasa cantidad de opciones de solución al momento de diseñar o especificar los requisitos de software de un proyecto genera una baja calidad en el producto final entregado o en algunos casos el producto no satisface las necesidades del usuario.

2.1.2. Formulación del problema:

¿Cómo desarrollar un prototipo inteligente para determinar problemas de dependencia en aplicaciones basadas en microservicios?

2.3. OBJETO Y CAMPO DE ACCIÓN

2.3.1. Objeto:

Inteligencia Artificial: Técnicas de Procesamiento de Lenguaje Natural.

2.3.2. Campo:

Calidad de Aplicaciones Basadas en Microservicios.

2.4. BENEFICIARIOS

Tabla 2.1. Beneficiarios directos e indirectos.

Beneficiarios Directos	Beneficiarios Indirectos
- Equipos de desarrollo de Aplicaciones basadas en microservicios.	- Usuarios de las Aplicaciones basadas en microservicios.

2.5. JUSTIFICACIÓN

Las aplicaciones basadas en microservicios son una tendencia relativamente nueva para el desarrollo de aplicaciones, por ello carecen de varios aspectos de calidad en su desarrollo, como es el caso particular de los problemas de dependencia que se estudia en el presente proyecto de investigación, particularmente apoyándose en el desarrollo de un prototipo basado en inteligencia artificial usando técnicas de procesamiento del lenguaje natural y reconocimiento de voz, para encontrar historias de usuario a partir de un texto y que cada una de estas se conviertan en un microservicio y visualizar si existe dependencia entre los mismos, con esto se puede brindar una posible solución de arquitectura en las etapas iniciales del proyecto para tomar una decisión de continuar o realizar cambios para conseguir una aplicación con baja dependencia, logrando que si un microservicio presenta un error inesperado no afecte a toda la aplicación.

Con el desarrollo del prototipo inteligente se busca determinar los problemas de dependencia, para ello el prototipo inteligente permitirá al usuario ingresar el relato de su necesidad o problema de forma oral o a su vez por un archivo de audio, posteriormente se procesa la información por medio de técnicas de inteligencia artificial basadas en el procesamiento de lenguaje natural, para de esta manera obtener historias de usuario, formar el Product Backlog, y crear un modelo de dependencia en el cual se busca establecer que cada una de las historias de usuario forme parte de un solo microservicio.

Las aplicaciones basadas en microservicios están enfocadas en la creación de sistemas complejos, en particular el desarrollo de estas aplicaciones parten de historias de usuario por lo cual es muy importante conocer sus aspectos fundamentales, como los problemas de dependencia para tomar las decisiones de diseño y arquitectura que tendrá un sistema, además el prototipo realizado es un caso relativamente nuevo en la Universidad Técnica de Cotopaxi, debido a que no se han creado proyectos similares basados en inteligencia artificial con lo cual es un tema de interés y relevancia en el aspecto académico investigativo.

El prototipo inteligente está enfocado en ayudar a la toma de decisiones de los desarrolladores de software en las fases iniciales del proyecto, específicamente en la fase de análisis de requisitos y modelado del sistema, ofreciendo alternativas de solución o un panorama general, como son los requisitos del cliente que están dentro de las historias de usuario que pueden aumentar la dependencia pero que no son detectados hasta fases posteriores del proyecto.

2.6. HIPÓTESIS

Un prototipo inteligente para el diseño de aplicaciones basadas en microservicios, puede ayudar a reducir la incidencia de problemas de dependencia en este tipo de aplicaciones.

2.7. OBJETIVOS

2.7.1. General:

- Proponer un prototipo inteligente para determinar problemas de dependencia en aplicaciones basadas en microservicios, mediante la utilización de técnicas de procesamiento de lenguaje natural.

2.7.2. Específicos:

- Realizar investigación bibliográfica para sustentar el marco teórico del proyecto de investigación.
- Aplicar técnicas de procesamiento de lenguaje natural para el desarrollo del prototipo inteligente.
- Validar el prototipo inteligente creado para determinar problemas de dependencia en aplicaciones basadas en microservicios.

2.8. SISTEMA DE TAREAS

Tabla 2.2. Sistema de tareas.

Objetivos específicos	Actividades(tareas)	Resultados esperados	Técnicas, medios e instrumentos
Realizar investigación bibliográfica para sustentar el marco teórico del proyecto de investigación.	<ul style="list-style-type: none"> -Identificar antecedentes investigativos. -Buscar información en fuentes de consulta confiables. -Sintetizar la información más relevante. 	-Fundamentación teórica.	<ul style="list-style-type: none"> -Investigación bibliográfica. -Fichas bibliográficas.
Aplicar las técnicas de procesamiento de lenguaje natural para el desarrollo de prototipo.	<ul style="list-style-type: none"> -Codificar cada uno de los módulos del sistema. - Diseñar un GUI que permite la utilización del prototipo. -Definir un plan de pruebas para verificar el funcionamiento del prototipo. 	<ul style="list-style-type: none"> -Funcionalidades implementadas. -Pantallas del prototipo. -Plan de pruebas. 	-Prácticas ágiles para el desarrollo de software.
Validar el prototipo inteligente para determinar problemas de dependencia de aplicaciones basadas en microservicios.	<ul style="list-style-type: none"> -Encuesta a expertos sobre la utilidad de la investigación y el prototipo. -Analizar los resultados de la validación de los expertos. 	<ul style="list-style-type: none"> -Valoración de Expertos. -Cuestionario de encuesta. -Documento de resultados. 	<ul style="list-style-type: none"> -Encuesta. -Validación por expertos a través del método del alfa de Cronbach.

3. FUNDAMENTACIÓN TEÓRICA

3.1. INTELIGENCIA ARTIFICIAL

3.1.1. Definición

La Inteligencia Artificial (IA) es la habilidad de los ordenadores para hacer actividades que normalmente requieren inteligencia humana como ver (visión artificial), oír (reconocimiento de voz) y entender (procesamiento del lenguaje natural). Tiene dos objetivos, un objetivo tecnológico, consiste en utilizar los ordenadores para hacer cosas útiles y un objetivo científico, usar conceptos y modelos de IA que ayuden a resolver necesidades del ser humano y los demás seres vivos [1].

Según menciona el autor [2], inteligencia artificial es la capacidad que tienen las máquinas para emplear algoritmos, aprender en base a un conjunto de datos y utilizar lo aprendido para tomar una resolución similar a como lo haría un ser humano, además las máquinas que están programadas con algoritmos de inteligencia artificial no necesitan descansar y pueden analizar grandes volúmenes de información a la vez y la proporción de errores es significativamente menor.

Este concepto es utilizado dentro del proyecto para lograr el reconocimiento del lenguaje natural del usuario, en este caso específico para entender el relato o problema del usuario, para obtener resultados de la investigación se crea un prototipo que procese la solicitud del usuario, luego filtrar ambigüedades, las acciones que realicen los actores se deben ver reflejados en un Product Backlog, luego se debe crear un gráfico para poder evidenciar problemas de dependencia en el desarrollo de aplicaciones basadas en microservicios. Todo esto con el fin de aumentar las opciones de solución para el equipo de desarrollo y mejorar las decisiones que se tomen y disminuir los riesgos que se puedan presentar a medida que el proyecto avance y conseguir los objetivos del proyecto.

3.1.2. Característica

La Inteligencia Artificial se caracteriza por la eliminación de tareas monótonas, minimizando los errores y los costos humanos. Puede manejar una gran cantidad de datos, transformando datos en información de utilidad para la toma de decisiones de los usuarios. Su principal objetivo es la imitación de la cognición humana, como puede ser la visión, la audición y el entendimiento [3].

Adicionalmente, la Inteligencia Artificial va en continua evolución logrando inmiscuirse cada vez más en las tareas cotidianas de la sociedad y convirtiéndose en una necesidad para la competitividad de los negocios y empresas, además ayuda a obtener un mejor análisis general del ambiente, ofreciendo varias soluciones para abordar un problema. Cuenta con varias técnicas las cuales pueden utilizar diferentes algoritmos, cada técnica se enfoca en obtener determinados resultados por lo cual dependiendo del objetivo del proyecto se deben utilizar una u otra técnica o combinar técnicas.

3.1.3. Aplicaciones Prácticas de la Inteligencia Artificial

- Asistentes personales virtuales: hace referencia a chatbots interactivos que podrán sugerir productos según el historial, estos pueden ser restaurantes, hoteles, servicios de búsquedas.
- Finanzas: este tipo de tecnologías inteligentes pueden ayudar a los bancos a detectar el fraude, además de predecir patrones de mercado e incluso aconsejar operaciones a sus clientes.
- Comercial: con la inteligencia artificial se puede hacer pronósticos de ventas y elegir el producto adecuado para sugerir al cliente, cabe recalcar que empresas como Amazon utilizan robots para determinar si un producto tendrá o no aceptación.
- Agrícolas: la inteligencia artificial ayuda en la agricultura por medio de análisis predictivos que ayudan a mejorar los rendimientos agrícolas y advierten de impactos ambientales adversos.
- Sanidad: hace referencia a la recopilación de datos que genera patrones que ayudan a identificar factores genéticos susceptibles de desarrollar una enfermedad [4].

3.1.4. Aplicaciones Técnicas de la Inteligencia Artificial

- Reconocimiento de imágenes estáticas, clasificación y etiquetado.
- Mejoras del desempeño de la estrategia algorítmica comercial.
- Procesamiento eficiente y escalable de datos de pacientes.
- Mantenimiento predictivo.
- Detección y clasificación de objetos.
- Distribución de contenido en las redes sociales.
- Protección contra amenazas de seguridad cibernética [1].

3.1.5. Ventajas y Desventajas

3.1.5.1. Ventajas

- Obtención de más datos, crear servicios tan útiles que la gente esté dispuesta a permitir que sus datos sean utilizados por el servicio.
- Ofrecer productos garantizados basados en Inteligencia Artificial, que son utilizadas en empresas como Google y Facebook, la experiencia de usuario es personalizada de tal manera que sea importante y relevante.
- Mayor número de usuarios esto se da cuando una persona tiene una experiencia agradable con un producto o servicio, tiende a recomendarlo a sus amigos.

3.1.5.2. Desventajas

- El entrenamiento de los algoritmos requiere una gran cantidad de datos para que los resultados puedan tener una alta confiabilidad.
- Existen limitaciones de software y hardware, debido a que el manejo de una gran cantidad de datos consume varios recursos de los equipos.
- Los algoritmos en ocasiones pueden ser de alta complejidad con lo cual se dificulta el análisis del proceso que siguen los datos para obtener determinados resultados.

3.1.6. Clasificación

3.1.6.1. Aprendizaje automático

El aprendizaje automático (Machine Learning), cuyo objetivo es desarrollar técnicas que permitan que las computadoras aprendan. Se dice que un algoritmo aprende cuando su desempeño mejora en base a la experiencia y mediante el uso de datos; es decir, cuando la habilidad no estaba presente en su genotipo o rasgos de nacimiento [5].

a) Tipos de Aprendizaje Automático

- **Aprendizaje supervisado (Supervised Machine Learning)**

Según postula [6], acerca de los algoritmos de aprendizaje supervisado dice que se genera un modelo predictivo, el mismo que está basado en datos de entrada y salida. La palabra principal “supervisado” hace referencia a tener la idea de una agrupación de datos anteriormente etiquetado y clasificado, es decir, tener un conjunto de muestra, el cual ya se sabe a qué grupo, valor o categoría pertenecen cada uno de los ejemplos. Con este grupo de datos conocidos también como datos de entrenamiento, se realiza el ajuste al modelo inicial planteado. De esta manera es como el algoritmo va “aprendiendo” a clasificar las muestras de entrada comparando el resultado del modelo, y la etiqueta real de la muestra, realizando las compensaciones si se da

cumplimiento al modelo de acuerdo a cada error en la estimación del resultado, como se puede apreciar en la Figura 3.1.

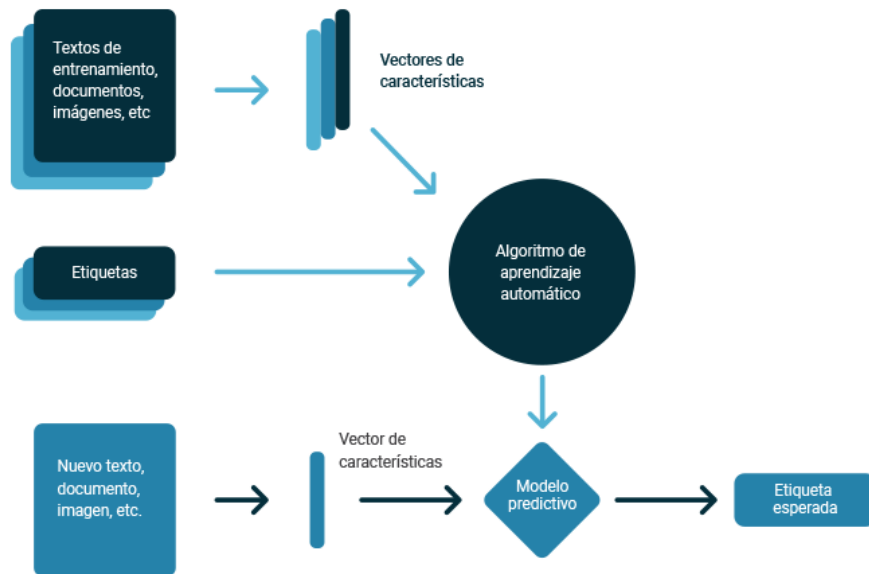


Figura 3.1. Diagrama de flujo del aprendizaje supervisado [6].

- **Aprendizaje no supervisado (Unsupervised Machine Learning)**

Este tipo de algoritmos no usan ningún tipo de dato etiquetado con anterioridad para indicar cómo tendría que ser clasificada la nueva información, sino que tienen que encontrar la manera de clasificarlas por sí mismos, además es importante mencionar que este tipo de algoritmos no requiere la intervención humana el funcionamiento de este modelo se puede apreciar en la Figura 3.2 [7].

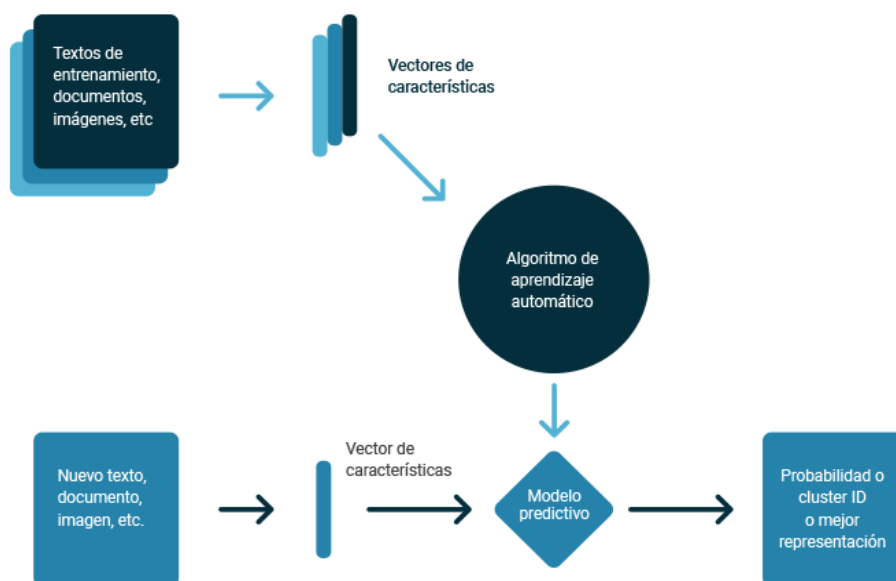


Figura 3.2. Diagrama de flujo del aprendizaje no supervisado [6].

- **Aprendizaje por refuerzo (Reinforcement Learning)**

Desde el punto de vista de [6], plantea que los algoritmos de aprendizaje por refuerzo definen modelos y funciones enfocadas en maximizar el grado de confianza, con la finalidad de obtener “recompensas”, basados en “acciones” y al ambiente en el que el agente inteligente se desempeñará. Este algoritmo es el más apegado a la psicología conductista de los humanos, debido a que es un modelo acción-recompensa, puesto que busca que el algoritmo se ajuste a la mejor “recompensa” dada por el ambiente en el que se desarrolla el proceso, y sus acciones por tomar están sujetas a estas recompensas, los procesos que se llevan a cabo en este modelo de aprendizaje se muestran en la Figura 3.3.



Figura 3.3. Diagrama de flujo del aprendizaje por refuerzo [6].

3.1.6.2. Aprendizaje Profundo

De acuerdo con [8], el aprendizaje profundo o deep learning, se trata de un subcampo del aprendizaje automático, este tipo de aprendizaje se utiliza para resolver problemas sumamente complejos y tienen grandes volúmenes de datos. Este tipo de aprendizaje se realiza mediante el uso de redes neuronales, que se organizan en capas para reconocer relaciones y patrones complejos en los datos, su aplicación requiere un enorme conjunto de información y una potente capacidad de procesamiento. Es muy frecuentemente utilizado para el reconocimiento de voz, el procesamiento del lenguaje natural, la visión artificial y la identificación de vehículos en los sistemas de asistencia al conductor.

3.1.6.3. Procesamiento de lenguaje natural

De acuerdo con [9], el procesamiento de lenguaje natural (NLP, por sus siglas en inglés) es una rama de la inteligencia artificial que ayuda a las computadoras a entender, interpretar y manipular el lenguaje humano. NLP toma elementos prestados de muchas disciplinas,

incluyendo la ciencia de la computación y la lingüística computacional, en su afán por cerrar la brecha entre la comunicación humana y el entendimiento de las computadoras.

3.1.6.4. Reconocimiento automático del habla

Desde el punto de vista [8], el reconocimiento automático del habla permite a las computadoras tener la capacidad de recibir mensajes orales, tomando como entrada señal acústica recogida por un micrófono, el objetivo final es descodificar el contenido en la onda acústica para realizar posteriormente varias acciones pertinentes. En su desarrollo intervienen varias disciplinas, como es el caso de la fisiología, la acústica, la lingüística, el procesamiento de señales, la inteligencia artificial y la ciencia de la computación.

3.1.6.5. Planificación

Según [3], el objetivo de la planificación es que un sistema encuentre por sí solo una secuencia ordenada de acciones que permiten alcanzar metas a partir de la situación y recursos iniciales. Constan de tres entradas, una descripción del estado inicial del mundo, una descripción del objetivo a alcanzar y un conjunto de acciones posibles.

3.2. PROCESAMIENTO DE LENGUAJE NATURAL (PLN)

3.2.1. Definición de procesamiento de lenguaje natural

De acuerdo con [10], menciona que una de las tareas fundamentales de la Inteligencia Artificial (IA) es la manipulación de lenguajes naturales usando herramientas de computación, en esta los lenguajes de programación son fundamentales, debido a que forman el enlace necesario entre los lenguajes naturales y su manipulación por una máquina. El procesamiento de lenguaje natural consiste en la aplicación de un lenguaje natural para poder establecer comunicación con la computadora, debiendo ésta entender las oraciones que le sean empleadas por el ser humano, además es importante mencionar que el uso de estos lenguajes naturales facilita el desarrollo de programas para que realicen tareas relacionadas con el lenguaje o bien, desarrollar patrones que ayuden a entender los mecanismos humanos relacionados con el lenguaje.

3.2.2. Aplicaciones del Procesamiento de Lenguaje Natural

Según menciona [11], las aplicaciones del Procesamiento de Lenguaje Natural (PLN) son muy variadas, debido a que su alcance es muy grande, algunas de las aplicaciones son:

- Resúmenes de textos, se fundamenta en encontrar la idea principal del texto y omitir lo que no sea destacado.
- Chatbots, son capaces de mantener una charla fluida con el usuario y responder a sus

preguntas de manera automática.

- Recomendaciones de entidades, consiste en detectar personas, entidades comerciales o gubernamentales, países, ciudades, marcas, etc.
- Análisis de sentimientos, se basa en comprender si un comentario es positivo o negativo y en qué magnitud podría llegar a ser este neutro, este tipo de análisis es frecuentemente utilizado en opiniones de productos redes sociales y otros.
- Machine Translation, se caracteriza por brindar la posibilidad de traducir texto o el audio de un idioma a otro componente y cada vez con más precisión.
- Clasificación automática de textos, en categorías preexistentes, detectar temas recurrentes y crear las categorías [11].

3.2.3. Arquitectura de un sistema de Procesamiento de Lenguaje Natural

Teniendo en cuenta a [12] y [10], la arquitectura de un sistema de Procesamiento de Lenguaje Natural se sustenta en una definición del lenguaje natural por niveles, estos son: fonológico, morfológico, sintáctico, semántico y pragmático, es importante mencionar que no todos los niveles que se describen son aplicables para cualquier tarea de PLN, sino más bien depende del objetivo de la aplicación que se vaya a desarrollar.

3.2.3.1. Nivel Fonológico

Según menciona [13], el nivel fonológico se trata de cómo las palabras se relacionan con los sonidos que representan, se ocupa primordialmente de la organización mental de los sonidos de una lengua particular.

3.2.3.2. Nivel Morfológico

De acuerdo con [10], se trata de cómo las palabras se construyen a partir de unas unidades de significado más pequeñas llamadas morfemas, en la Lingüística general, la Morfología suele cubrir dos grandes zonas, ambas de considerable amplitud.

- Para [14], una de las tareas principales de la Morfología, es tratar de comprender lo que ocurre dentro de las palabras que conforman un texto, unidad que, a diferencia de lo que sucede en el plano sintáctico, deja de ser mínima, es importante mencionar además que los morfemas, unidades mínimas en este plano.
- Desde el punto de vista de [14], afirma que la Morfología es la teoría de las categorías léxicas, las cuales que están conformadas por varias características como son: sustantivo, adjetivo, verbo o preposición en la Lingüística general es un trabajo bastante más complejo.

3.2.3.3. Nivel Sintáctico

Este nivel da a conocer cómo las palabras pueden unirse para formar oraciones, fijando el papel estructural que cada palabra juega en la oración y que sintagmas son parte de otros sintagmas [15].

3.2.3.4. Nivel Semántico

Según afirma [16], este nivel proporciona la interpretación de las oraciones, una vez eliminadas las ambigüedades morfosintácticas, el análisis semántico está basado en estudiar las relaciones entre temas tratados en un texto, para lo cual se definen varios patrones de relación que se tomarán en cuenta, dos aspectos principales que son las relaciones gramaticales o relaciones más propiamente semánticas.

3.2.3.5. Nivel Pragmático

Desde el punto de vista de [15], el nivel pragmático trata de cómo las oraciones se usan en diversas situaciones y de cómo el uso afecta al significado de las oraciones. Se inspecciona un subnivel recursivo: discursivo, el cual trata la manera en que el significado de una oración se ve afectado por las oraciones.

Por otra parte [10] da a conocer, que la arquitectura del sistema de procesamiento de lenguaje natural, muestra como la computadora interpreta y analiza oraciones que le sean proporcionadas la explicación se lleva a cabo en los siguientes pasos:

- a) El usuario le expresa de forma oral a la computadora que es lo que quiere realizar.
- b) La computadora analiza las oraciones del usuario, en el sentido morfológico y sintáctico, esto dicho de otra manera si las frases contienen palabras compuestas por morfemas y si la estructura de las oraciones es correcta.
- c) El siguiente paso es analizar las oraciones semánticamente, es decir saber cuál es el significado de cada oración, y asignar el significado de estas a expresiones lógicas (cierto o falso).
- d) Una vez que se haya culminado el paso anterior y haber analizado las oraciones por separado posteriormente se analizan todas juntas, tomando en cuenta la situación de cada oración, una vez realizado este paso, la computadora ya sabe que es lo que va a realizar como expresión final.
- e) Una vez obtenida la expresión final, el siguiente paso es la ejecución de esta, para obtener así el resultado y poder proporcionarle al usuario [10].

3.2.4. Herramientas que permiten realizar el Procesamiento de Lenguaje Natural

En la actualidad, existen multitud de aplicaciones, herramientas, librerías y entornos de trabajo que facilitan el procesamiento del lenguaje natural. Implementan todo tipo de técnicas, métodos y algoritmos, están disponibles para diferentes lenguajes de programación, bajo diferentes tipos de licencias, tanto comerciales como libres [17].

Tabla 3.1. Lenguajes de programación, nombres de herramientas y licencia para el procesamiento de lenguaje natural [17].

Lenguaje de programación	Nombre de la herramienta	Licencia
.NET Framework	Antelope framework	Libre para investigación
	NooJ (basado en INTEX)	Libre para investigación
	Rossete	Comercial
C	Ellogon	LGPL
	Rosette	Comercial
C#	Antelope framework	Libre para investigación
C++	Apertium,	GPL
	DELPH-IN	LGPL, MIT
	Distinguo	Comercial
	Ellogon	LGPL
	FreeLing	GPL
	Rasp	LGPL
	Rosette	Comercial
	UIMA	Apache 2.0
	UniteX	LGPL
	VisualText	Libre (uso no comercial)
Java	WebLab-project	LGPL
	Apertium	GPL
	ClearTK	Nueva licencia BSD
	Factorie	Apache
	General Architecture for Text Engineering (GATE)	LGPL
	Graph Expression	Apache

Tabla 3.2. “Continuación” Lenguajes de programación, nombres de herramientas y licencia para el procesamiento de lenguaje natural [17].

Lenguaje de programación	Nombre de la herramienta	Licencia
Java	IceNLP	LGPL
Node JS	ClearTK	Nueva licencia BSD
Others	Factorie	Apache
	General Architecture for Text Engineering (GATE)	LGPL
Perl	Graph Expression	Apache
Python Lenguaje de Programación	NLTK	Libre para investigación
	Spacy	Libre para investigación
	Scikit-learn	Licencia BSD
	Gensim	Licencia GNU LGPL
	Pattern	Licencia BSD.
	Polyglot	Licencia GPLv3

En las Tablas 3.1 y 3.2, se puede observar varios lenguajes de programación tanto como las herramientas y el tipo de licencia, que sirven para el procesamiento de lenguaje natural en este caso se opta por escoger el lenguaje de programación Python puesto que el equipo de desarrollo tiene conocimientos previos en este lenguaje de programación, además de esto, Python cuenta varias opciones de herramientas con las que se puede trabajar y la mayoría son de uso libre para la investigación.

Tabla 3.3. Ventajas y desventajas de librerías de lenguaje de programación Python para procesamiento de lenguaje natural.

Librerías PLN Python	Ventajas	Desventajas
NLTK	La librería de PNL más conocida y completa. Muchas extensiones de terceros. Numerosos enfoques para cada tarea de PNL. Admiten mayor cantidad de idiomas.	Complicado de aprender y usar. Bastante lento en la tokenización de oraciones, NLTK solo divide textos por oraciones, sin analizar la estructura semántica. Procesa cadenas que no son muy típicas para lenguaje Python orientado a objetos.

Tabla 3.4. “Continuación” Ventajas y desventajas de librerías de lenguaje de programación Python para procesamiento de lenguaje natural.

Librerías PLN Python	Ventajas	Desventajas
Spacy	<p>Es una librería de PNL es rápida e intuitiva.</p> <p>Fácil de aprender y usar porque tiene una herramienta única altamente optimizada para cada tarea.</p> <p>Proporciona modelos pre entrenados de diferentes lenguajes. Utiliza sintaxis clara hace que sea ideal para principiantes en el campo de PLN. Permite crear modelos nuevos o reentrenar los modelos.</p>	<p>Carece de flexibilidad en comparación con NLTK.</p> <p>No es compatible con muchos idiomas. Existen modelos solo para 7 idiomas.</p>
Scikit-learn	<p>Tiene funciones que ayudan a utilizar el método de bolsa de palabras de creación de características para los problemas de clasificación.</p> <p>Proporciona una amplia variedad de algoritmos para construir modelos de inteligencia artificial automático.</p> <p>Tiene una buena documentación de clases y métodos.</p>	<p>Para cosas de preprocesamiento más sofisticadas (por ejemplo, el post etiquetado), se debería utilizar otra librería y sólo después se puede utilizar.</p> <p>No usa redes neuronales para el preprocesamiento del texto.</p>
Gensim	<p>Funciona con grandes conjuntos de datos y preprocesa los mismos.</p> <p>Proporciona una vectorización de TF-IDF, Word2VEC, Document2VEC, análisis semántico latente. Apoya al aprendizaje profundo.</p>	<p>Diseñado principalmente para personas sin supervisión de modelado de texto.</p> <p>No tiene suficientes herramientas para proporcionar canalización de PNL, por lo que debe usarse con algunos de otra librería como Spacy o NLTK.</p>
Pattern	<p>Permitir etiquetado de parte del discurso, búsqueda de n-gramas, análisis de sentimientos, espacio vectorial y agrupamiento. Rastreador web, analizador DOM, y funcionan con algunas apps.</p>	<p>Es un minero web puede ser suficientemente optimizado para algunas tareas específicas de PNL.</p>
Polyglot	<p>Admite una gran cantidad de idiomas.</p>	<p>No es tan popular como Spacy o NLTK ya que tiene soluciones más lentas y el soporte solo es comunitario.</p>

En las Tablas 3.3 y 3.4, se puede observar distintas librerías para procesamiento de lenguaje natural de Python, conjuntamente con sus ventajas y desventajas de cada una, para el estudio realizado se optó por escoger la librería spaCy debido a que tiene herramientas específicas para cada tarea de procesamiento de texto, proporciona modelos pre entrenados de diferentes lenguajes, utiliza sintaxis clara hace que sea ideal para principiantes en el campo de PLN y sobre todo porque permite crear modelos nuevos o reentrenar los modelos en campos específicos.

3.3. SPACY

3.3.1. Definición

De acuerdo con [18], spaCy es una librería gratuita de código abierto creado por Matt Honnibal para el procesamiento avanzado de lenguaje natural (PLN) en Python.

Por otra parte [19], menciona que spaCy además proporciona modelos pre entrenados de diferentes lenguajes, lo cual, junto a una sintaxis clara hace que sea ideal para principiantes en el campo de PLN, por otra parte esta librería permite crear modelos nuevos o reentrenar los modelos que proporciona con datos propios para obtener modelos enfocados en campos específicos, es importante mencionar que spaCy está orientado al uso de producción, proporcionando un framework para construir aplicaciones completas que requieran procesamiento de lenguaje natural, por último los modelos de spaCy garantizan velocidad y precisión.

3.3.2. Características

Según [18], menciona diferentes características y capacidades de spaCy, algunos de ellos se refieren a conceptos lingüísticos, mientras que otros están relacionados con una funcionalidad más general de aprendizaje automático como se detalla en las Tablas 3.5 y 3.6.

Tabla 3.5. Características y descripción de la librería de Python spaCy [18].

Característica	Descripción
Tokenización	Hace referencia a segmentar el texto en palabras, signos de puntuación, etc.
Categorización Gramatical (Part-of-Speech, POS)	Asignar tipos de palabras a tokens, como verbo o sustantivo.
Análisis de dependencia	Asignar etiquetas de dependencia sintáctica, describiendo las relaciones entre tokens individuales, como sujeto u objeto.
Lematización	Asignar las formas base de las palabras. Por ejemplo, el lema de "era" es "ser" y el lema de "ratas" es "rata".
Detección de límites de oración (SBD)	Encontrar y segmentar frases individuales.

Tabla 3.6. “Continuación” Características y descripción de la librería de Python spaCy [18].

Característica	Descripción
Reconocimiento de entidad nombrada (NER)	Etiquetado de objetos denominados "del mundo real", como personas, empresas o ubicaciones.
Vinculación de entidades (EL)	Desambiguar entidades textuales a identificadores únicos en una base de conocimiento.
Semejanza	Comparar palabras, tramos de texto y documentos y qué tan similares son entre sí.
Clasificación de texto	Asignar categorías o etiquetas a un documento completo o partes de un documento.
Coincidencia basada en reglas	Detectar secuencias de tokens basadas en sus textos y anotaciones lingüísticas, similares a las expresiones regulares.
Capacitación	Actualizar y mejorar las predicciones de un modelo estadístico.
Publicación por entregas	Guardar objetos en archivos o cadenas de texto.

3.3.3. Técnica de procesamiento de datos de Spacy



Figura 3.4. Técnica de procesamiento de datos de Spacy [20].

Las estructuras de datos centrales en spaCy son las clases de Language, el Vocabulario (vocab) y el objeto Doc. La clase Language se usa para procesar un texto y convertirlo en un objeto Doc. Normalmente se almacena como una variable llamada nlp. El objeto Doc posee la secuencia de tokens y todas sus anotaciones, es crucial mencionar que, al centralizar cadenas, vectores de palabras y atributos léxicos en el Vocab, se evita almacenar múltiples copias de estos datos y esto a su vez ahorra memoria y asegura que haya una única fuente de verdad [20].

Las anotaciones de texto también están diseñadas para permitir una única fuente de verdad: el objeto Doc es propietario de los datos, Span y Token son vistas que apuntan hacia él. El objeto Doc es construido por el Tokenizer y luego modificado en su lugar por los componentes de la tubería. El objeto Language coordina estos componentes. Toma texto sin formato y lo envía a través de la canalización, devolviendo un documento anotado. También organiza el entrenamiento y la serialización [20].

3.3.3.1. Objetos de contenedor

Tabla 3.7. Objetos de contenedor [20].

Nombre	Descripción
Doc	Un contenedor para acceder a anotaciones lingüísticas.
DocBin	Una colección de objetos Doc para una serialización binaria eficiente. También se utiliza para datos de entrenamiento .
Example	Una colección de anotaciones de entrenamiento, que contiene dos objetos Doc: los datos de referencia y las predicciones.
Language	Clase de procesamiento que convierte texto en objetos Doc. Los diferentes lenguajes implementan sus propias subclases. La variable se suele llamar nlp.
Lexeme	Una entrada en el vocabulario. Es un tipo de palabra sin contexto, a diferencia de un símbolo de palabra. Por lo tanto, no tiene etiqueta de parte del discurso, análisis de dependencia, etc.
Span	Un trozo de un objeto Doc.
SpanGroup	Una colección con nombre de tramos que pertenece a un Doc.
Token	Un token individual, es decir, una palabra, un símbolo de puntuación, un espacio en blanco, etc.

En la Tabla 3.7 se puede observar los nombres y una descripción general de objetos que conforman el contenedor de spaCy, los mismos que son importantes para saber cómo se lleva a cabo el procesamiento de lenguaje natural.

3.3.3.2. Canal de procesamiento

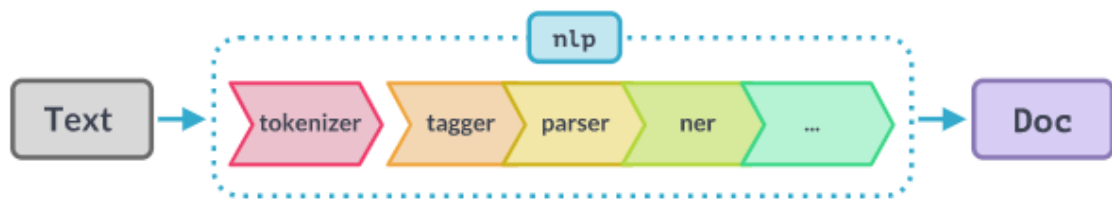


Figura 3.5. Canal de procesamiento [20].

Según menciona [20], la canalización de procesamiento consta de uno o más componentes de la canalización que se llaman en orden en el Doc. El tokenizador se ejecuta antes que los componentes. Los componentes de la tubería se pueden agregar usando `Language.add_pipe`. Pueden contener un modelo estadístico y pesos entrenados, o solo realizar modificaciones basadas en reglas en Doc. `spaCy` proporciona una gama de componentes integrados para diferentes tareas de procesamiento de idiomas y también permite agregar componentes personalizados como se puede observar en la Figura 3.5.

Tabla 3.8. Atributos del canal del procesamiento [20].

Nombre	Descripción
AttributeRuler	Establezca los atributos del token mediante reglas de emparejamiento.
DependencyParser	Predecir dependencias sintácticas.
EntityLinker	Elimine la ambigüedad de entidades con nombre en nodos en una base de conocimiento.
EntityRecognizer	Predecir entidades nombradas, por ejemplo, personas o productos.
EntityRuler	Agregue intervalos de entidad al Doc usando reglas basadas en tokens o coincidencias de frases exactas.
Lemmatizer	Determina las formas básicas de las palabras.
Morphologizer	Predecir características morfológicas y etiquetas de parte del discurso de grano grueso.
SentenceRecognizer	Predecir los límites de las oraciones.
Sentencizer	Implemente la detección de límites de oraciones basada en reglas que no requieran el análisis de dependencia.
Tagger	Predecir etiquetas de parte del discurso.
TextCategorizer	Predecir categorías o etiquetas en todo el documento.
Tok2Vec	Aplique un modelo de "token a vector" y establezca sus salidas.

Tabla 3.9. “Continuación” Atributos del canal del procesamiento [20].

Nombre	Descripción
Tokenizer	Segmenta el texto sin procesar y crea objetos Doc a partir de las palabras.
TrainablePipe	Clase de la que heredan todos los componentes de canalización entrenables.
Transformer	Utilice un modelo de transformador y configure sus salidas.
Otras funciones	Aplicar automáticamente algo al Doc, por ejemplo, para fusionar tramos de tokens.

En las Tablas 3.8 y 3.9, se puede observar los diferentes atributos y su descripción los cuales forman parte del canal de procesamiento de spaCy, es importante conocer cada una de las funcionalidades de estos atributos puesto que servirán para desarrollar un algoritmo capaz de poder encontrar historias de usuario a partir de texto.

3.3.3.3. Matchers

De acuerdo con [20], los comparadores lo ayudan a encontrar y extraer información de objetos Doc, basados en patrones de coincidencia que describen las secuencias que está buscando. Un comparador opera en un Doc y les da acceso a los tokens coincidentes en contexto, existen tres tipos de Matchers como se describe en la Tabla 3.6.

Tabla 3.10. Comparadores y descripción [20].

Nombre	Descripción
DependencyMatcher	Haga coincidir secuencias de tokens según los árboles de dependencia mediante operadores Semgrep .
Matcher	Coincidir con secuencias de tokens, basadas en reglas de patrones, similares a las expresiones regulares.
PhraseMatcher	Combina secuencias de fichas basadas en frases.

3.4. APLICACIÓN

3.4.1. Definición

Según menciona [21], una aplicación es un software que procesa datos para el usuario, a excepción del "software del sistema", que proporciona la infraestructura en la computadora (sistema operativo, utilidades y componentes relacionados), todos los programas de software son programas de aplicación.

Las aplicaciones hacen mención a prácticamente cualquier tipo de programa, desde hojas de cálculo, hasta reproductores multimedia o juegos de realidad virtual entre otros.

3.4.2. Tipos de arquitectura de Aplicaciones

Algunos autores como [22], definen a una Arquitectura de Software más bien como el diseño de un Sistema Informático, el cual es único para cada arquitectura propiamente dicha, está definida como única para cada sistema que se vaya a implementar y de ser el caso se le puede dar una definición diferente o similar dependiendo el punto de vista.

3.4.2.1. Arquitectura monolítica

a. Definición

Según [23], define una arquitectura monolítica como un único ejecutable lógico, es decir que cualquier cambio realizado en una parte de un sistema de este tipo implica la construcción y despliegue de una nueva versión de todo el sistema. Una Arquitectura Monolítica según [24], es una simple unidad, donde las aplicaciones están construidas en tres partes: una Interfaz de Usuario Cliente, Aplicación del lado del Servidor y una Base de Datos.

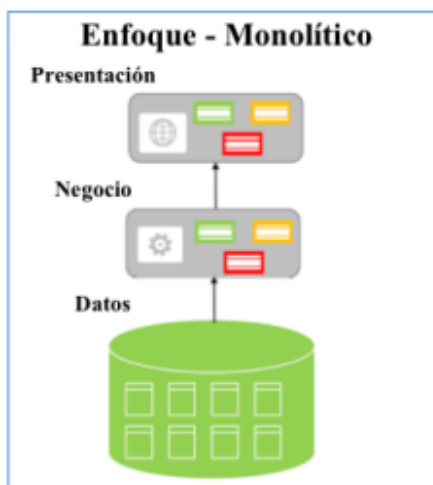


Figura 3.6. Arquitectura monolítica [25].

En la Figura 3.6, se puede apreciar un ejemplo de un sistema construido con Arquitectura Monolítica. Por lo general, las Aplicaciones con esta Arquitectura involucran aspectos de presentación, procesamiento y almacenamiento de información dentro de un sólo componente de Software. Además, son elaboradas con un supuesto de que serán ejecutadas en un único servidor.

3.4.2.2. Arquitectura Basada en Microservicios

a. Definición

La Arquitectura Basada en Microservicios es relativamente nueva y aún no existe una definición formal, sin embargo, tomamos un par de conceptos referenciales.

El primero viene de [26], quien menciona que este tipo de arquitectura es usada para construir aplicaciones grandes, complejas y horizontalmente escalables, compuestas de procesos pequeños, independientes y altamente desacoplables, comunicándose cada uno usando API's - Application Programming Interface (Interfaz de Programación de Aplicaciones en español).

Por otro lado [22], define una arquitectura de Microservicios como una propuesta innovadora para producir Software basándose en la agrupación de servicios individuales, que tienen su propio proceso o hilo de proceso en el sistema principal y mantienen una comunicación a través de mecanismos ligeros. La comunicación entre cada uno de estos servicios se hace comúnmente usando el protocolo HTTP - Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto en español).

3.4.3. Arquitectura Monolítica vs Arquitectura basada en Microservicios

A continuación, se muestra una lista de características de Arquitectura Monolítica y Arquitectura de Microservicios que propone [27].

Tabla 3.11. Características arquitectura monolítica vs arquitectura basada en microservicios [27].

Monolítica	Microservicios
Su gran tamaño y complejidad dificulta el mantenimiento.	Implementa una cantidad limitada de funcionalidades, lo cual reduce el número de errores posteriores.
Existe estricta dependencia entre las partes que componen el sistema.	Posibilita planear la migración a nuevas versiones de una tecnología.

Tabla 3.12. “Continuación” Características arquitectura monolítica vs arquitectura basada en microservicios [27].

Monolítica	Microservicios
Cambios en un módulo implica reiniciar un sistema completo.	El manejo modular de los Microservicios permite evitar el reinicio completo de un sistema, y enfocarse únicamente en una parte.
El despliegue de un Sistema Monolítico no siempre es exitoso debido a conflictos de requerimiento de recursos.	Los Microservicios se prestan para el uso de contenedores. Esto le permite a un DevOps poder adecuar un ambiente de desarrollo compartido y único.
Limita la escalabilidad del sistema.	La escalabilidad no implica la duplicidad de sus componentes, sino, a conveniencia se pueden crear nuevas instancias de servicios dependiendo de la carga de trabajo.
Representa una tecnología cerrada para los desarrolladores, quienes están obligados a usar los mismos lenguajes de programación de la Aplicación original.	Su comunicación en red entre componentes se restringe al uso de protocolos como HTTP. A parte de eso, un desarrollador es libre de elegir el lenguaje y Framework que quiera usar.
Complejidad para aplicar la Integración / Despliegue Continuo.	Va de la mano con la Integración / Despliegue Continuo, fácil integración.
El Integrated Development Environment – IDE (Ambiente de Desarrollo Integrado, en español), se vuelve lento cuando carga incluso lo que nunca se utiliza para desarrollar la aplicación.	El enfocarse en una parte de la aplicación le permite al desarrollador tener en su IDE solamente las herramientas necesarias para esta parte en desarrollo.

En las Tablas 3.11 y 3.12 se describe la comparación entre arquitectura monolítica y arquitectura basada en microservicios para lo cual se dan a conocer sus características, de las cuales se pueden destacar que las aplicaciones monolíticas destacan por agrupar todas las funciones en un solo gran código, mientras que una app basada en microservicios está constituida por diversos módulos y componentes que se comunican entre sí.

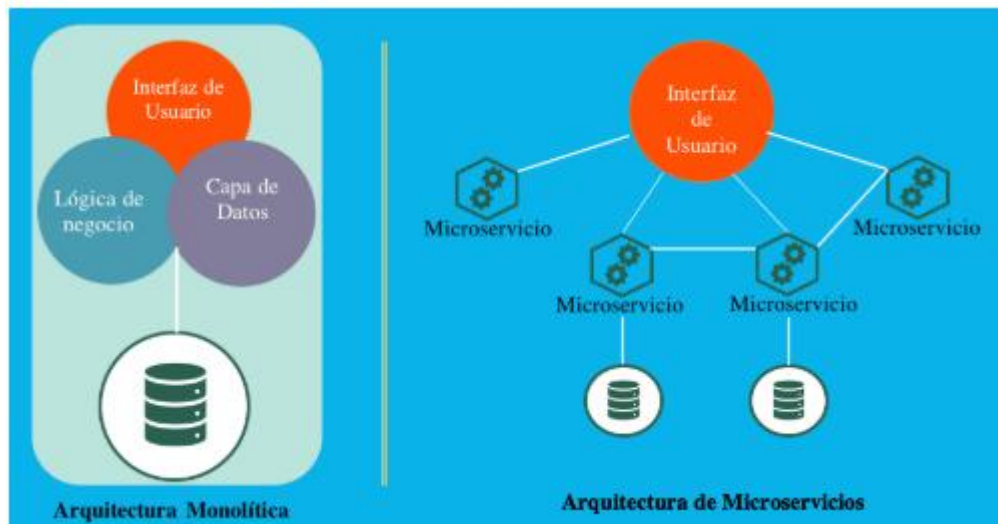


Figura 3.7. Arquitectura Monolítica y de Microservicios [28].

De forma general, la Figura 3.7 hace referencia básicamente cómo se vería una aplicación en base a la arquitectura monolítica y arquitectura basada en microservicios.

3.5. CALIDAD DE SOFTWARE

Según [29], menciona que la calidad del software hace referencia a un conjunto de cualidades que lo caracterizan y que determinan su utilidad, por otra parte es importante tomar en cuenta que calidad es sinónimo de eficiencia, flexibilidad por ende son varios aspectos a tomar en cuenta para validar si un software cuenta con los parámetros de calidad, es importante tomar en cuenta que la calidad del software puede medirse después de elaborado el producto, pero esto puede resultar muy costoso si se detectan problemas, por lo que es imprescindible tener en cuenta la obtención de la calidad durante las etapas del ciclo de vida del software.

3.5.1. Cómo obtener un software de calidad

Según menciona [29], la obtención de un software de calidad implica utilizar metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software en desarrollo con la finalidad de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad para el control de la calidad del software.

- El principio tecnológico define las técnicas a utilizar en el proceso de desarrollo del software.
- El principio administrativo contempla las funciones de planificación y control del desarrollo del software, así como la organización del ambiente o centro de ingeniería de software.
- El principio ergonómico define la interfaz entre el usuario y el ambiente automatizado.

3.5.2. Calidad de aplicaciones basadas en microservicios

Para [30], una de las ventajas de utilizar microservicios es la capacidad de publicar una aplicación grande como un conjunto de pequeñas aplicaciones (microservicios) que se pueden desarrollar, desplegar, escalar, manejar y visualizar de forma independiente. Los microservicios permiten a las empresas gestionar las aplicaciones de código base grande usando una metodología más práctica, donde las mejoras incrementales son ejecutadas por pequeños equipos en bases de código y despliegues independientes, como se puede evidenciar los beneficios de las aplicaciones basadas en microservicios en la Figura 3.8.

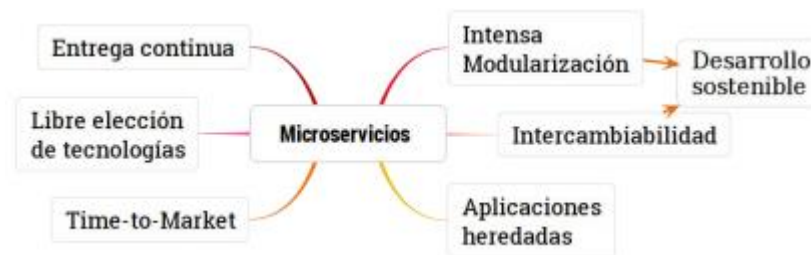


Figura 3.8. Beneficios de los microservicios.

3.5.3. Problemas de calidad en aplicaciones basadas en microservicios

3.5.3.1. Dependencia

a. Definición

Es el grado en que un componente de software necesita de otro para su funcionamiento [8]. Un alto nivel de acoplamiento o dependencia en un componente de software, hace que el mismo no pueda funcionar correctamente sin el otro componente, si se cambia un componente obligatoriamente se debe cambiar los componentes que dependen de este.

b. Métricas para calcular la Dependencia

- **Similitud semántica (SsT):** Según [31], menciona que la similitud semántica utiliza algoritmos de evaluación de la distancia léxica para señalar los servicios que contienen componentes no relacionados o acciones no relacionadas que obstaculizan la cohesión.

Se utilizó la librería para realizar el procesamiento de lenguaje natural llamada Spacy [20], esto con la finalidad de calcular la similitud semántica entre los microservicios que conformaran el sistema; es importante mencionar además que cuya similitud se establece relacionando vectores de palabras o "incrustaciones de palabras", las cuales hacen referencia a representaciones de significado multidimensional de que conforma una palabra [32]. Se calculó la similitud semántica entre cada microservicio que está conformado por una historia de usuario, uniendo su nombre y su descripción. Para lo cual se calculó SsT de la siguiente manera:

1. Seleccionamos los sustantivos (entidades) del nombre y la descripción de la historia del usuario.
2. Identificamos los lemas del sustantivo en cada historia de usuario.
3. Definimos un diccionario que contiene los valores de similitud semántica entre las historias de usuarios para lo cual se aplica la formula (3.1):

$$DSS = [\langle "hu_1 - hu_2", a_{1-2} \rangle, \langle "hu_1 - hu_3", a_{1-3} \rangle, \dots, \langle "hu_j - hu_k", a_{j-k} \rangle] \quad (3.1)$$

Donde: DSS: Diccionario de similitud semántica entre las historias de usuario, “hui y huj” son los identificadores de las historias de usuario que se están comparando. “huj - huk” corresponde a la llave del diccionario, la cual se forma concatenando los identificadores de las historias de usuario que se comparan y aj-k corresponde al valor del diccionario, son los valores de similitud semántica entre las historias huj y huk obtenidos por Spacy, Son números reales entre 0 y 1. Entre más cercano a uno sea el valor, más similares son las historias de usuario.

4. Se calculó la similitud semántica apoyándose de las fórmulas (3.2) y (3.3), en donde SS del i-ésimo microservicio como el promedio de los valores de similitud semántica entre sus historias de usuarios. La similitud semántica total de MSBA fue el promedio de la similitud semántica de cada microservicio. Para obtener un valor de similitud semántica entre 0 y 100 multiplicamos el promedio por 100 [8].

$$SSj = \frac{1}{c} \sum_{j=1}^m a_j - K \quad (3.2)$$

$$SST = \frac{100}{n} \sum_{i=1}^n SSi \quad (3.3)$$

5. Donde: aj-k corresponden a los valores de similitud semántica de la comparación realizada entre las historias de usuarios de msi. c hace referencia al número de comparaciones realizadas para calcular SS. n corresponde al número de microservicios de MSBA [8].

- **Importancia absoluta del microservicio (AIS):** es el número de otros microservicios que invocan al menos una operación de la interfaz de un microservicio para obtener la respuesta se hace uso de las fórmulas (3.4) y (3.5) [2].

AIS_i es un número de clientes (otros microservicios) que invocan al menos una operación de ms_i y a nivel del sistema se define el vector \overrightarrow{AIS} , que contiene el valor AIS calculado para cada microservicio para calcular el valor total de AIS a nivel de sistema (AisT), se calcula la norma del vector \overrightarrow{AIS} . Donde n es el número de microservicios de MSBA, por lo tanto:

$$\overrightarrow{AIS} = [AIS_1, AIS_2, \dots, AIS_n] \quad (3.4)$$

$$AisT = |\overrightarrow{AIS}| = \sqrt{AIS_1^2 + AIS_2^2 + \dots + AIS_n^2} \quad (3.5)$$

- **Dependencia absoluta del microservicio (ADS):** [2] es el número de otros microservicios de los que dependen el microservicio para obtener las respuestas se hace uso de las fórmulas (3.6) y (3.7).

ADS_i es el número de microservicios de los cuales se invoca al menos una operación de ms_i y para calcular el valor total de ADS a nivel de sistema (AdsT), se calcula la norma del vector ADS. Entonces:

$$\overrightarrow{ADS} = [ADS_1, ADS_2, \dots, ADS_n] \quad (3.6)$$

$$AdsT = |\overrightarrow{ADS}| = \sqrt{ADS_1^2 + ADS_2^2 + \dots + ADS_n^2} \quad (3.7)$$

- **Interdependencia de los microservicios (SIY):** según menciona [2], es el número de pares de microservicios interdependientes. En este caso SIY_i define el número de pares de microservicios que dependen bidireccionalmente uno del otro dividido por el número total de microservicios para obtener el valor del mismo se aplica las fórmulas (3.8) y (3.9).

$$\overrightarrow{SIY} = [SIY_1, SIY_2, \dots, SIY_n] \quad (3.8)$$

$$SiyT = |\overrightarrow{SIY}| = \sqrt{SIY_1^2 + SIY_2^2 + \dots + SIY_n^2} \quad (3.9)$$

Se define el vector \vec{Cp} como el vector que contiene las métricas de acoplamiento calculadas para MSBA, calculando la norma del vector \vec{Cp} tenemos el valor de acoplamiento para la aplicación (CpT) para obtener las respuestas se aplicaron las fórmulas (3.10) y (3.11) [2].

$$\vec{Cp} = [AisT, AdsT, SiyT] \quad (3.10)$$

$$CpT = |\vec{Cp}| = \sqrt{AisT^2 + AdsT^2 + SiyT^2} \quad (3.11)$$

c. Proceso para el Cálculo de la Dependencia

- Limpiar las ambigüedades de las Historias de Usuario.
- Reconocer las entidades involucradas.
- Aplicar algoritmos de similitud semántica.
- Conectar las historias de usuario con una similitud semántica en sus entidades, mayor a 0.85.
- Crear los microservicios.
- Calcular las llamadas (calls) y pedidos (request) de cada microservicio.
- Aplicar las fórmulas para calcular las métricas de dependencia.

3.5.3.2. Complejidad

a. Definición

La complejidad corresponde a una gran cantidad de temáticas a analizar, se puede comprender como la cantidad de procesos, funciones, métodos o características que posee un microservicio y todo el esfuerzo y los recursos necesarios para completarlos. La complejidad en los microservicios debe ser baja, para que puedan ser modificados en un par de semanas, reescritos y mejorados rápidamente. Si la complejidad es alta, entonces el costo del cambio es mayor [2].

3.5.3.4. Granularidad

Los microservicios se pueden declarar con diferentes niveles de capacidad, el tamaño de esta funcionalidad se suele denominar granularidad, es decir, la complejidad funcional codificada en un servicio o número de casos de uso implementados por un microservicio [33].

Por lo tanto, es interesante lograr un nivel óptimo de granularidad a los desarrolladores de aplicaciones los factores clave que contribuye a lo que se puede resumir como:

Impulsado por la necesidad o la capacidad empresarial, las necesidades de una empresa pueden estar cambiando rápidamente y exigiendo nuevas funciones desde una aplicación. Este

crecimiento puede no ser manejable dentro de la arquitectura de la aplicación existente y, por lo tanto, se adopta un enfoque granular. Es típico de la aplicación que los desarrolladores utilicen la propia funcionalidad para establecer el alcance que determina el tamaño de un microservicio [34].

Tamaño de aplicación, para aplicaciones más pequeñas, el nivel de granularidad podría ser muy detallada. Para empresas (más grandes) aplicaciones, es probable que la granularidad esté en un nivel superior (más grueso) con cada microservicio construido a partir de microservicios cruzados [35].

Diseño de bases de datos, el diseño de una base de datos puede tener un impacto sobre granularidad. Por ejemplo, en un escenario minorista si hay un servicio de producto y un servicio de pedido, el concepto funcional es probable que la composición haya conducido a la implementación de repositorios de datos separados para cada servicio. Cualquier asociación de los datos entre las bases de datos se implementará en nivel de código, lo que lleva a microservicios más detallados [35].

Se puede evidenciar que dependencia es uno de los principales problemas en aplicaciones que están basadas en microservicios, puesto que es una nueva tendencia para desarrollar aplicaciones, basándose en ello se puede evidenciar la factibilidad de estudio para indagar la manera de que cada microservicio que componga la aplicación funcione de modo prácticamente independiente para de este modo no afectar en el funcionamiento y garantizar la estabilidad de aplicación en general.

3.6 METODOLOGÍA DE LA INVESTIGACIÓN

3.6.1. Tipo de investigación

3.6.1.1. Investigación Cuantitativa

Desde el punto de vista de [36], la investigación cuantitativa utiliza la recolección de datos para probar hipótesis con base en la medición numérica y el análisis estadístico, con el fin establecer pautas de comportamiento y probar teorías. Se analizan las mediciones obtenidas utilizando métodos estadísticos, y se extrae una serie de conclusiones. Se prueban hipótesis con la finalidad de aceptarlas o rechazarlas.

3.6.2. Nivel de investigación

3.6.2.1. Investigación Exploratoria

Para [36], los estudios exploratorios se realizan cuando el objetivo es examinar un tema o problema de investigación poco estudiado, del cual se tienen muchas dudas o no se ha abordado antes. Es decir, cuando la revisión de la literatura reveló que tan sólo hay guías no investigadas e ideas vagamente relacionadas con el problema de estudio, o bien, si deseamos indagar sobre temas y áreas desde nuevas perspectivas es importante además mencionar que los estudios exploratorios sirven para familiarizarnos con fenómenos relativamente desconocidos, obtener información sobre la posibilidad de llevar a cabo una investigación más completa respecto de un contexto particular, los estudios exploratorios generalmente determinan tendencias, identifican áreas, ambientes, contextos y situaciones de estudio, relaciones potenciales entre variables; o establecen el “tono” de investigaciones posteriores más elaboradas y rigurosas esta clase de estudios son comunes en la investigación, sobre todo en situaciones en las que existe poca información

3.6.3. Diseño de la Investigación

3.6.3.1. Investigación documental o bibliográfica

Según menciona [37], la investigación documental es un proceso basado en la búsqueda, recuperación, análisis, crítica e interpretación de datos secundarios, es decir, los obtenidos y registrados por otros investigadores en fuentes documentales: impresas, audiovisuales o electrónicas, por otra parte, como en toda investigación, el propósito de este diseño permite el aporte de nuevos conocimientos.

3.6.4. Instrumentos de recolección de datos

3.6.4.1. Encuesta

Para [38], la encuesta es una técnica que se lleva a cabo mediante la aplicación de un cuestionario a una muestra de personas, además la encuesta se aplica para poder probar una hipótesis o descubrir una solución a un problema, identificar e interpretar, de la manera más metódica posible, un conjunto de testimonios que puedan cumplir un determinado propósito.

3.6.5. Técnicas para el análisis de datos

3.6.5.1. Coeficiente del alfa de Cronbach

Según [39], el coeficiente alfa fue descrito en por Lee J. Cronbach, el cual menciona que es un índice usado para medir la confiabilidad del tipo de consistencia interna de una determinada escala, es decir, para evaluar importancia en que los ítems de un instrumento están vinculado con otras palabras, el alfa de Cronbach es el promedio de las correlaciones entre los ítems que hacen parte de un instrumento también se puede concebir este coeficiente como la medida en la cual alguna creación, concepto o factor medido está presente en cada ítem del instrumento, por lo general un grupo de ítems que explora un factor en general muestra un elevado valor de alfa de Cronbach.

3.7. METODOLOGÍA DE DESARROLLO

Según menciona [40], el transcurso de desarrollo es utilizado con la finalidad de promover productos de software seguros que pueda resistir o sostenerse a ataques ya anticipados, recuperarse rápidamente y mitigar el daño provocado por los ataques que no pueden ser suprimidos o resistidos.

Para [41], la metodología de desarrollo se usa en la creación de un software, no solo porque busca cumplir con las expectativas del cliente, sino también para reformar los procesos internos en la elaboración de un producto que ayudan a las organizaciones a mejorar la calidad de sus procesos, para el crecimiento y posicionamiento de una empresa.

De acuerdo con [40] y [41], las metodologías son usadas en el desarrollo de software, dependiendo del proceso que se utilice, se puede contar con sistemas informáticos confiables y seguros, a su vez con la terminación del mismo en un tiempo estimado. Siguiendo estos procesos se puede reparar los daños que esté presente o que sea fácil su mantenimiento.

3.7.1. Metodologías de Desarrollo de Software Ágiles

Para [42], las metodologías ágiles tienen dos diferencias fundamentales con las metodologías tradicionales; la primera es que los métodos ágiles son adaptativos y no predictivos, debido a que contrasta con la predictibilidad buscada por las metodologías tradicionales por otra parte la segunda diferencia es que las metodologías ágiles son orientadas a las personas y no orientadas a los procesos.

El trabajar con metodologías ágiles permite una rápida respuesta a cambios de requisitos a lo largo del desarrollo, así como entregas continuas en plazos cortos del software funcional. Se

puede trabajar entre el cliente y el equipo de desarrollo, lo cual ayuda a minimizar los costos frente a cambios que se manifiestan a menudo eliminando el trabajo innecesario. En estas metodologías, el equipo de desarrollo no malgasta el tiempo y dinero del cliente desarrollando soluciones innecesariamente generales y complejas que en realidad no son requisito del cliente, logrando que cada componente del producto final sea probado y que satisfaga los requerimientos.

3.7.1.1. Roles del proyecto

Tabla 3.13. Formato para definir de roles del proyecto.

ROL	ACTIVIDADES	RESPONSABLES
Product Owner		
Scrum Master		
Scrum Team		

Para un adecuado seguimiento del desarrollo de prototipo se requiere definir tres roles. El primero es el Scrum Máster quien lidera las reuniones y ayuda al equipo en caso de presentarse problemas y también tiene el rol de minimizar los obstáculos para cumplir cada uno de los Sprint. El segundo rol es el Product Owner, habla con el cliente y asegura que el equipo cumpla con cada uno de los requerimientos del sistema y el Scrum Team que son los encargados de desarrollar y cumplir con cada una de las exigencias que les asigna el Product Owner.

3.7.1.2. Historias de usuario

Tabla 3.14. Formato para definir una historia de usuario.

HISTORIA DE USUARIO (HU)			
Código HU:		Fecha:	
Sprint:		Prioridad:	
Actor(es):		Puntos:	
Nombre:			
Descripción:			
Como:			
Quiero:			
Para:			
Restricciones:			
Criterios de aceptación:			
DoD (Definition of Done):			

Las historias de usuario son fundamentales en la metodología de desarrollo ágil, debido a que se utilizan para detallar de forma general los requisitos del prototipo inteligente, además que estas historias de usuario conforman los diferentes Sprint que ayudan a desarrollar el prototipo inteligente.

a) Estimación de puntos de historia basados en Planning Poker.

Según [43], esta técnica se usa para asignar estimaciones a las historias de usuario para ello, se considera a todo el equipo que participará del proyecto (desarrolladores, testers, diseñadores y Product Owners), entregándoles un número determinado de cartas 1, 2, 3, 5, 8, 13, 21, ?, cada carta representa un número de puntos de historia.

Para realizar estimaciones con esta técnica se realiza el siguiente proceso:

- El Product Owner da a conocer la historia de usuario, qué desea lograr y cómo aportará valor.
- Las personas encargadas del proyecto debaten brevemente. ¿Qué implica? ¿Qué riesgos existen? escuchar los diferentes puntos de vista es fundamental.
- Se establece un determinado tiempo para que cada persona piense y elija su carta.
- Una vez terminado el tiempo todas las personas muestran la carta seleccionada.
- Se cuenta el número de personas que eligieron cada carta, la mayoría suele elegir un valor y sólo unos pocos eligen uno sumamente superior o inferior.
- Es bueno escuchar las opiniones, es decir, aquellas personas que tienen estimaciones muy bajas o muy altas, debido a que pueden tener información que el resto desconoce u omite. Por supuesto que también pudo desconocer o malinterpretarse algún punto de la historia de usuario de igual modo, resulta relevante escuchar sus argumentos y, en base a ellos, decidir si se desea repetir la estimación.
- Siempre se llega a un consenso, que los valores sean discretos ayuda mucho [43].

3.7.1.3. Product Backlog

Tabla 3.15. Formato para definir Product Backlog.

Id	Descripción	Tarea	Prioridad	Sprint

El Product Backlog engloba de manera general la lista de tareas o historias de usuario que se van a realizar para cumplir con el desarrollo de un proyecto, además debe contener la descripción de las funcionalidades las mismas que están ordenadas descendientemente de acuerdo a la prioridad para de esta manera desarrollar primero las historias que tienen mayor prioridad [44].

a) Técnica de priorización MoSCoW

Se utiliza para realizar la priorización de las tareas que van a conformar el Product Backlog se utilizó el método MoSCoW, que es una técnica que ayuda a la gestión de proyectos ágiles puesto que ordena el grado de prioridad de acuerdo a las funcionalidades del prototipo en base a los objetivos: debe tener, debería tener, podría tener y no tendrá en esta ocasión.

3.7.1.4. Release plan

Tabla 3.16. Formato para definir Release Plan.

RELEASE PLAN						
Release	Sprint	Épicas	HU	Responsable de la HU	Inicio	Fin

El “Release Plan” o planificación de entregas se realiza para mostrar el conjunto de historias agrupadas las mismas que conforman las versiones del proyecto conocidas también como Release [45].

3.7.1.5. Sprint

Tabla 3.17. Formato para definir un Sprint.

DATOS DE SPRINT		
Número.		
Fecha de inicio.		
Fecha de culminación.		
Tareas a desarrollar		
Prioridad.	Descripción.	Responsable.

Cada Sprint está establecido en un lapso de dos semanas, tiempo durante el cual se realizan reuniones presenciales para poder realizar la revisión del avance del prototipo inteligente, con la finalidad de ver el porcentaje de cumplimiento de las tareas establecidas y de ser el caso adaptarse a un cambio en específico o de la misma manera aprobar el sprint y designar nuevas tareas para el desarrollo del siguiente sprint [44].

3.7.1.6. Pruebas unitarias

Tabla 3.18. Formato para definir las pruebas unitarias.

Código Prueba Unitaria	Fecha de ejecución
Historia de Usuario:	
Responsable:	
Descripción del caso de prueba:	
Prerrequisitos:	
Prueba Unitaria:	
Fragmento de Código:	
Resultados Esperados:	
Datos Ingresados:	
Observaciones:	

Son utilizadas para comprobar y validar el correcto funcionamiento de un fragmento de código, función o método, para ahorrar tiempo y evitar errores futuros cuando se lo implemente conjuntamente con el resto de funciones del sistema. Son realizadas por el equipo de trabajo.

3.7.1.7. Sprint Review.

Tabla 3.19. Formato para definir casos de prueba.

Caso de prueba:	
Objetivo:	
Alcance:	
Fecha:	
Responsables:	
# Caso de prueba.	
# Sprint Review.	
# H.U.	

Tabla 3.20. “Continuación” Formato para definir casos de prueba.

Descripción.	
Condiciones.	
Entradas.	
Evaluación de la Prueba.	

Se lleva a cabo la revisión del Sprint con el Product Owner, Scrum Master, Scrum Team con la finalidad de evaluar los resultados al finalizar la fecha de entrega del Sprint y decidir si aprobar o realizar futuras adaptaciones [46].

4. MATERIALES Y MÉTODOS

4.1 LÍNEA DE TIEMPO DE LA INVESTIGACIÓN

Se utilizó la línea de tiempo para determinar las diferentes actividades y fechas para desarrollar el presente proyecto de investigación como se muestra en las Figuras 4.1. y 4.2.

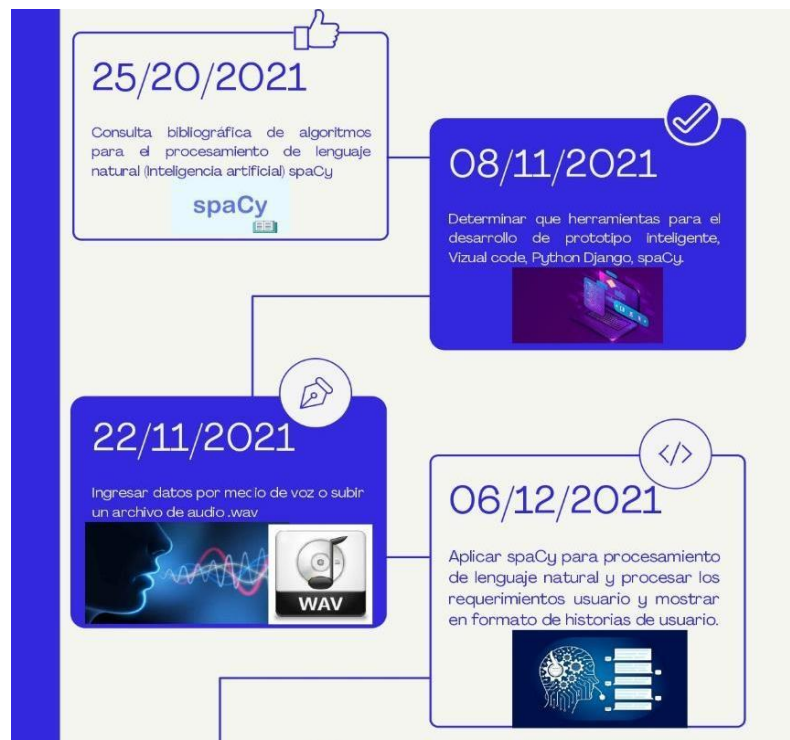


Figura 4.1. Línea de tiempo de las actividades realizadas para el proyecto de investigación.

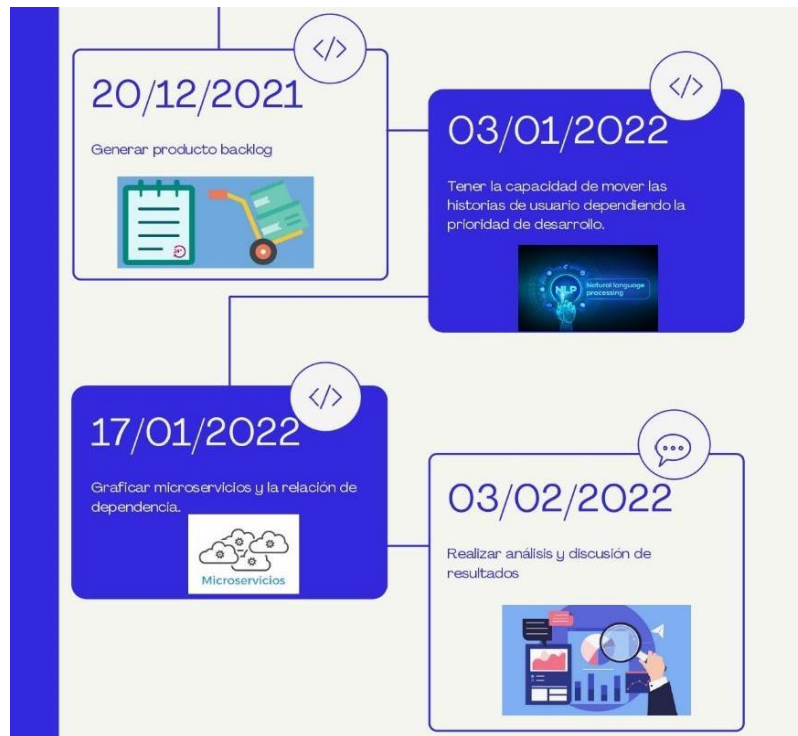


Figura 4.2. “Continuación” Línea de tiempo de las actividades realizadas para el proyecto de investigación.

4.2. HERRAMIENTAS DE DESARROLLO

Para efectuar el desarrollo del proyecto de investigación propuesto, se hizo uso de las siguientes herramientas los cuales servirán para el desarrollo del código fuente, base de datos, procesamiento de lenguaje natural y diseño de interfaces.

4.2.1. Visual Studio Code

El Software Visual Studio Code se utiliza para escribir el código fuente, debido a que permite la integración del framework de Django junto con el lenguaje de programación Python, de esta manera se puede ejecutar el código fuente.

4.2.2. Python

El lenguaje de programación Python se utiliza para crear clases y poder ingresar los requerimientos del usuario para posteriormente clasificarlos en historias de usuario, además sirve para crear métodos y llamar a las plantillas de html, también permite realizar validaciones y facilita la instalación de varios módulos como es el caso de la librería spaCy.

4.2.3. spaCy

Es una librería de Python que sirve para el procesamiento de lenguaje natural, como es el caso de eliminar todos los signos de puntuación para posteriormente poder clasificar texto, el cual

utiliza un algoritmo que permite un aprendizaje multitarea, como poder clasificar el pronombre, sustantivo, y demás caracteres del texto que tenga el relato, de esta manera poder obtener historias de usuario a partir de un texto.

4.2.4. Django

El framework Django se utiliza para generar el modelo-vista-controlador, además que permite crear aplicaciones directamente con el lenguaje de programación Python y la base de datos SQLite.

4.2.5. SQLite

El gestor de base de datos SQLite se utiliza para poder almacenar la base de datos, en el cual se agregan diferentes tablas con sus respectivas relaciones, las mismas que permitirá el funcionamiento adecuado del sistema.

4.2.6. Bootstrap

El framework Bootstrap se utiliza para el desarrollo de front-end de las diferentes interfaces del prototipo, el mismo que está conformado por varios formularios, botones, cuadros de navegación, entre otros elementos basados en HTML, CSS y JavaScript.

4.2.7. Speech Recognition

Es una librería de Python que sirve para el manejo de habla natural, tiene la finalidad de reconocer información por medio de voz y transformar a texto, para posteriormente aplicar técnicas de inteligencia artificial para el procesamiento de lenguaje natural.

4.2.8. PyAudio

Es una librería de Python que sirve para utilizar el micrófono de la computadora en cualquier navegador web, esta librería es utilizada en el prototipo inteligente con la finalidad de utilizar el micrófono y grabar las necesidades o requerimientos del usuario.

4.2.9. D3

Es una librería de JavaScript, sirve para realizar gráficos a partir de datos, a la vez permite realizar infogramas dinámicos, en el presente proyecto de investigación esta librería se utiliza con la finalidad de realizar el gráfico de microservicios y al mismo tiempo señalar los problemas de dependencias entre los mismos.

4.3. METODOLOGÍA

4.3.1. Metodología de la investigación

4.3.1.1. Tipo de investigación

Para nuestro proyecto de investigación se determinó que la investigación cuantitativa es la que mejor se acopla, puesto que se realiza encuestas a expertos para la recolección de datos con el propósito de valorar la confiabilidad del prototipo, en base a la medición numérica y el análisis estadístico para de esta manera establecer conclusiones.

4.3.1.2. Nivel de investigación

El proceso investigativo tuvo un nivel exploratorio, puesto que el objetivo de esta investigación es examinar un tema poco estudiado, como es el caso de incorporar técnicas de procesamiento de lenguaje natural con la finalidad de determinar problemas de dependencia en aplicaciones basadas en microservicios, este tema surge debido a que, al llevar a cabo la revisión de la literatura existe solamente una investigación con relación a este caso de estudio propuesto, debido a esto se determina que el proyecto de investigación, el objetivo de estudio es una tendencia.

4.3.1.3. Diseño de la Investigación

En el diseño de la investigación se utilizó la documental o bibliográfica puesto que permite obtener información de diferentes fuentes, debido a que el proyecto debe contar con una cantidad considerable de teoría y conceptos que respalden los procesos y métodos utilizados para resolver el problema, debe utilizar fuentes bibliográficas verídicas de documentos o textos actuales, a excepción de documentos o libros que sean los principales referentes del área de estudio.

4.3.1.4. Instrumentos de la investigación

Como instrumento se determinó la encuesta, la cual se aplica a expertos, para reunir varios criterios profesionales de validez y confiabilidad del “Prototipo Inteligente”. La validez del “Prototipo Inteligente”, los parámetros a evaluar es la transformación de audio a texto, la validación de las historias de usuario obtenidas tras el relato ingresado en el “Prototipo Inteligente” y la utilidad del gráfico de dependencias entre los microservicios generados. Es aquí donde la tarea del experto se convierte en una labor fundamental para eliminar aspectos irrelevantes, incorporar los que son imprescindibles y/o modificar aquellos que lo requieran.

Para la aplicación del método de Juicio de Expertos se establece adecuado contar con la colaboración de al menos 3 profesionales en el área de arquitectura de aplicaciones basadas en

microservicios y procesamiento de lenguaje natural, quienes serán los encargados de validar el funcionamiento del “Prototipo Inteligente” en base al formulario descrito en el **Anexo D: Formulario de Validación de Expertos**.

4.3.1.5. Técnicas para el análisis de datos

Para analizar los datos obtenidos de la encuesta realizada a los expertos se aplicará la técnica del coeficiente del alfa de Cronbach para obtener la confiabilidad del prototipo, el cual se basa en una medición del índice de consistencia interna del grado de confiabilidad que va en un rango de 0 a 1, para de esta manera determinar el valor de alfa cuyo resultado será el grado de confiabilidad de la validación de expertos. En la Tabla 4.1. se muestra el grado confiabilidad y su equivalencia.

Tabla 4.1. Grado de confiabilidad del Alfa de Cronbach.

Muy baja	Baja	Moderada	Buena	Alta
0 a 0,20	0,21 – 0,40	0,41 a 0,60	0,61 a 0,80	0,81 a 1,00

4.4. METODOLOGÍA DE DESARROLLO DE SOFTWARE

El presente proyecto de investigación debe tener una guía para estructurar cada una de las partes o etapas del mismo, con lo cual se utiliza una metodología que esté acorde con los objetivos del proyecto, de esta manera se pretende mantener la calidad de las etapas de planificación y diseño, otorgar pautas para facilitar el desarrollo del prototipo, y ayudar a la implementación y distribución del producto final. Adicionalmente, permite una mejor comunicación con los clientes, puesto que proporciona medios de presentación de los avances del proyecto a los clientes, para que puedan ser verificados por el equipo de trabajo y validados por los clientes.

4.4.1. Prácticas Ágiles

Para el desarrollo del prototipo se utilizó prácticas ágiles como es el caso de la definición de los roles, generación de las historias de usuario, Product Backlog, planificación de los Sprint, desarrollo de los Sprint, para de esta manera cumplir con el objetivo que es controlar y planificar el proyecto de software.

4.4.1.1. Roles

Tabla 4.2. Roles y actividades.

ROL	ACTIVIDADES	RESPONSABLES
Product Owner	Habla con el cliente y asegura que el equipo cumpla con cada uno de los requerimientos del sistema.	BRYAN TARCO CARLOS ALCASIGA
Scrum Master	Lidera las reuniones y ayuda al equipo en caso de presentarse problemas. También tiene el rol de minimizar los obstáculos para cumplir cada uno de los Sprints.	ING. VERÓNICA TAPIA
Scrum Team	Son los encargados de desarrollar y cumplir con cada una de las exigencias que les asigna el Product Owner.	BRYAN TARCO CARLOS ALCASIGA

4.4.1.2. Product Backlog

En la Tabla 4.3, se presenta la pila de producto o Product Backlog del sistema, en el que se representa la visión del prototipo con respecto a los objetivos y las entregas funcionales de software con la respectiva priorización de cada una de las funcionalidades, tomando en cuenta si es de prioridad Alta, Media o Baja según la importancia de la funcionalidad dentro del prototipo. Para realizar la estimación se utiliza la técnica de MoSCoW la cual ordena el grado de prioridad en base a los objetivos: debe tener, debería tener, podría tener y no tendrá en esta ocasión.

Tabla 4.3. Product Backlog.

Id	Descripción	Tarea	Prioridad	Sprint
HU00 1	Ingresar datos al Prototipo Inteligente.	Ingresar datos al prototipo de forma oral o por archivo de audio.	Alta	1
HU00 2	Ingresar el tipo de usuario.	Crear un campo para ingresar el usuario responsable de la solicitud.	Alta	
HU00 3	Reconocer historias de usuario.	Utilizar spaCy para desarrollar un algoritmo capaz de obtener historias de usuario.	Alta	2
HU00 4	Generar Product Backlog.	Resumir historias de usuario y mostrar.	Alta	
HU00 5	Ordenar prioridad de historias de usuario.	Permitir al usuario ordenar de forma manual las historias de usuario.	Media	3
HU00 6	Mejorar el algoritmo de Procesamiento de Lenguaje Natural.	Mejorar el algoritmo utilizando spaCy para obtener historias de usuario de una manera correcta.	Alta	
HU00 7	Graficar los microservicios y la relación de dependencia.	Generar un gráfico de los microservicios y las relaciones de dependencia entre los mismos.	Alta	4

4.4.1.3. Historias de Usuario

Son parte de un enfoque de las prácticas ágiles que sirven para dar una explicación general de una función de software descrita desde la perspectiva de la persona, generalmente usuario o cliente, para el desarrollo del prototipo se utilizaron siete historias de usuario que se describen en el **Anexo C**.

4.4.1.4. Prueba Unitarias

Son utilizadas para comprobar y validar el correcto funcionamiento de un fragmento de código, función o método, para ahorrar tiempo y evitar errores futuros cuando se lo implemente

conjuntamente con el resto de funciones del sistema, son realizadas por el equipo de trabajo y se describen en el **Anexo D**.

4.4.1.5. Release Plan

El “Release Plan” o planificación de entregas se realiza para mostrar el conjunto de historias agrupadas las mismas que conforman las versiones del proyecto conocidas también como Release.

Tabla 4.4. Release Plan.

RELEASE PLAN						
Release	Sprint	Épicas	HU	Responsable de la HU	Inicio	Fin
1	Sprint 1	Gestión del ingreso de Datos en el Sistema	Ingresar datos al Prototipo Inteligente.	Carlos Alcasiga. Bryan Tarco	25 de octubre del 2021.	17 de diciembre del 2021.
			Ingresar el tipo de usuario.	Carlos Alcasiga. Bryan Tarco		
	Sprint 2	Encontrar las historias de usuario	Reconocer historias de usuario.	Carlos Alcasiga. Bryan Tarco		
		Gestión del Product Backlog	Generar Product Backlog.	Carlos Alcasiga. Bryan Tarco		

Tabla 4.5. “Continuación” Release Plan.

RELEASE PLAN						
Release	Sprint	Épicas	HU	Responsable HU.	Inicio	Fin
2	Sprint 3	Gestión del Product Backlog	Ordenar prioridad de historias de usuario.	Carlos Alcasiga. Bryan Tarco	20 de diciembre del 2021.	4 de febrero del 2022.
		Encontrar las historias de usuario	Mejorar el algoritmo de Procesamiento de Lenguaje Natural.	Carlos Alcasiga. Bryan Tarco		
	Sprint 4	Generar un gráfico de Dependencia	Graficar los microservicios y la relación de dependencia.	Carlos Alcasiga. Bryan Tarco		

4.4.1.6. Planificación de Sprint.

Para la planificación de los Sprints se tomó en cuenta las historias de usuario que conforman el Product Backlog, a continuación, se presentan los respectivos Sprints.

a) Sprint 1.

Durante el proceso de desarrollo del Sprint 1, se realizó el ingreso de datos, es decir el ingreso de la solicitud del usuario de manera oral o por archivo de audio y la transformación a texto además del ingreso del tipo de usuario responsable de la solicitud.

Tabla 4.6. Planificación del Sprint 1.

DATOS DE SPRINT		
Número.	S001	
Fecha de inicio.	25 de octubre del 2021.	
Fecha de culminación.	19 de noviembre del 2021.	
Tareas a desarrollar		
Prioridad.	Descripción.	Responsable.
Alta	HU001: Ingresar datos al Prototipo Inteligente.	Bryan Tarco, Carlos Alcasiga.
Alta	HU002: Ingresar el tipo usuario.	Bryan Tarco, Carlos Alcasiga.

b) Sprint 2.

Durante el proceso de desarrollo del Sprint 2, se realizó la búsqueda de historias de usuario a través de técnicas de inteligencia artificial, con algoritmos de procesamiento de lenguaje natural también se desarrolló una nueva pestaña que se pueda generar y mostrar el Product Backlog.

Tabla 4.7. Planificación del Sprint 2.

DATOS DE SPRINT		
Número.	S002	
Fecha de inicio.	22 de noviembre del 2021.	
Fecha de culminación.	17 de diciembre del 2021.	
Tareas a desarrollar		
Prioridad.	Descripción.	Responsable.
Alta	HU003: Reconocer historias de usuario.	Bryan Tarco, Carlos Alcasiga.
Media	HU004: Generar Product Backlog con las HU encontradas.	Bryan Tarco, Carlos Alcasiga.

c) Sprint 3.

Durante el proceso de desarrollo del Sprint 3, se realizó el proceso de otorgar al usuario del sistema ordenar de forma manual las historias de usuario para determinar la prioridad de cada una, además de mejorar el algoritmo de procesamiento de lenguaje natural para detectar de mejor manera las historias de usuario.

Tabla 4.8. Planificación del Sprint 3.

DATOS DE SPRINT		
Número.	S003	
Fecha de inicio.	20 de diciembre del 2021.	
Fecha de culminación.	14 de enero del 2022.	
Tareas a desarrollar		
Prioridad.	Descripción.	Responsable.
Alta	HU005: Ordenar prioridad de historias de usuario.	Bryan Tarco, Carlos Alcasiga.
Alta	HU006: Mejorar el algoritmo para detectar historias de usuario.	Bryan Tarco, Carlos Alcasiga.

d) Sprint 4.

Durante el proceso de desarrollo del Sprint 4, se realizó una nueva pestaña en el cual se pueda graficar cada uno de los microservicios correspondientes a las historias de usuario y las relaciones de dependencia.

Tabla 4.9. Planificación del Sprint 4.

DATOS DE SPRINT		
Número.	S004	
Fecha de inicio.	17 de enero del 2022.	
Fecha de culminación.	4 de febrero del 2022.	
Tareas a desarrollar		
Prioridad.	Descripción.	Responsable.
Alta	HU007: Graficar microservicios y la relación de dependencia.	Bryan Tarco, Carlos Alcasiga.

4.4.1.7. Sprint Review.

Se lleva a cabo la revisión del Sprint con el tutor de la investigación, con la finalidad de inspeccionar los resultados al finalizar la fecha de entrega para aprobar o realizar futuras adaptaciones.

a) Sprint Review 1

Caso de prueba: Ingreso de la solicitud del usuario y tipo de usuario.

Objetivo: Verificar el ingreso de la solicitud del usuario, es decir el ingreso por voz o archivo de audio y la transformación a texto y permitir al usuario del sistema digitar el tipo de usuario.

Alcance: La historia de usuario 1 tiene como función el ingreso de la solicitud del usuario y la historia de usuario 2, ingresar el usuario responsable.

Fecha: 19 de noviembre del 2021.

Responsables: Carlos Alcasiga, Bryan Tarco.

Tabla 4.10. Caso de prueba 01.

# Caso de prueba.	Ingreso de la solicitud del usuario y tipo de usuario.
# Sprint Review.	1
# H.U.	HU001, HU002
Condiciones.	Usuario del prototipo ingresa datos.
Entradas.	<ol style="list-style-type: none"> 1. El Usuario ingresa al sistema e ingresa requerimientos de forma oral o su vez sube un archivo de audio .wav. 2. El Usuario digita el tipo de usuario que le corresponden las historias de usuario previamente ingresadas de manera oral.
Evaluación de la Prueba.	Prueba superada.

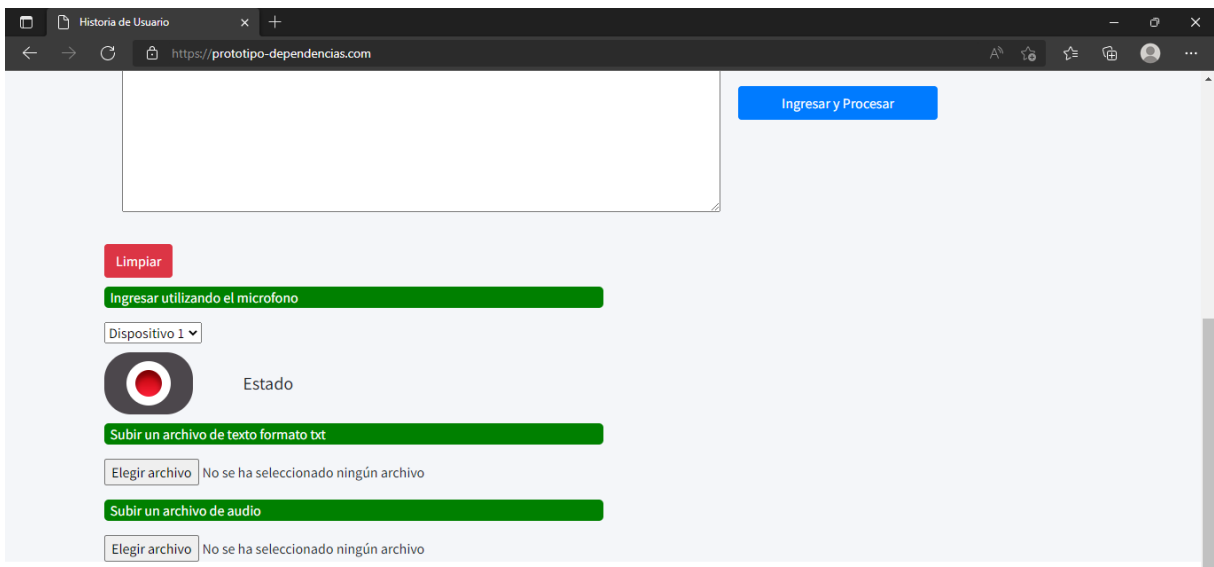


Figura 4.3. Ingreso de la solicitud del usuario a través de voz y archivo de audio.

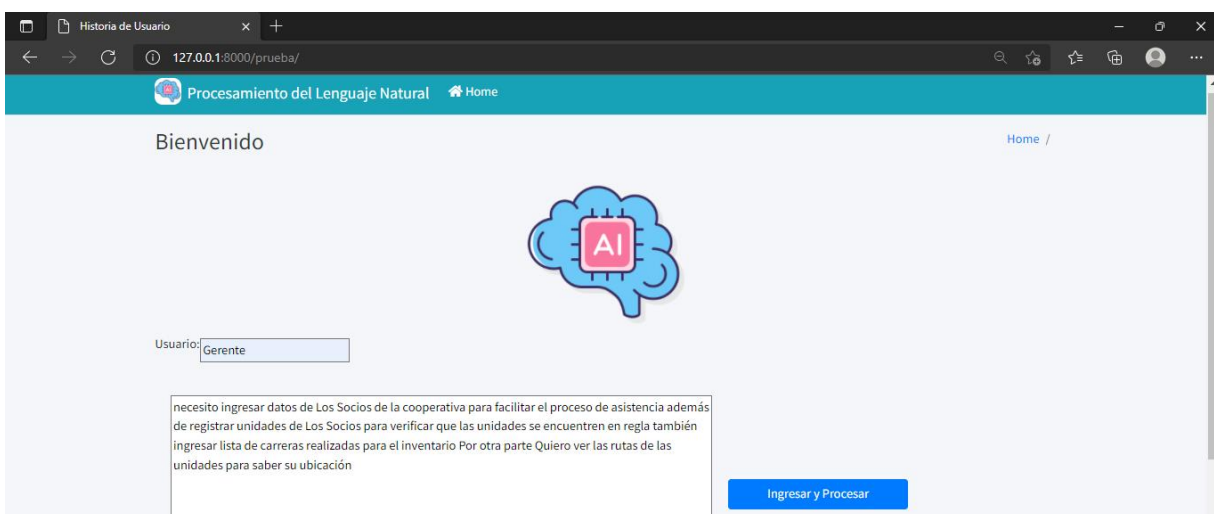


Figura 4.4. Ingresar el usuario responsable.

b) Sprint Review 2.

Caso de prueba: Detectar historias de usuario y generar Product Backlog.

Objetivo: Verificar que la búsqueda de historias de usuario se realiza de manera correcta a las mismas que están establecidas por técnicas de inteligencia artificial y crear una nueva pestaña en la cual se pueda generar y mostrar el Product Backlog

Alcance: La historia de usuario 3 tiene como función detectar historias de usuario y la historia de usuario 4 Generar Product backlog.

Fecha: 19 de diciembre del 2021

Responsables: Carlos Alcasiga, Bryan Tarco.

Tabla 4.11. Caso de prueba 02.

# Caso de prueba.	Detectar historias de usuario y ordenar.
# Sprint Review.	2
# H.U.	HU003, HU004
Condiciones.	Prototipo inteligente buscará historias de usuario y el usuario podrá generar el Product Backlog.
Entradas.	<ol style="list-style-type: none">1. Usuario presiona el botón ingresar y procesar.2. Usuario presiona el botón generar Product Backlog.
Evaluación de la Prueba.	Prueba superada.

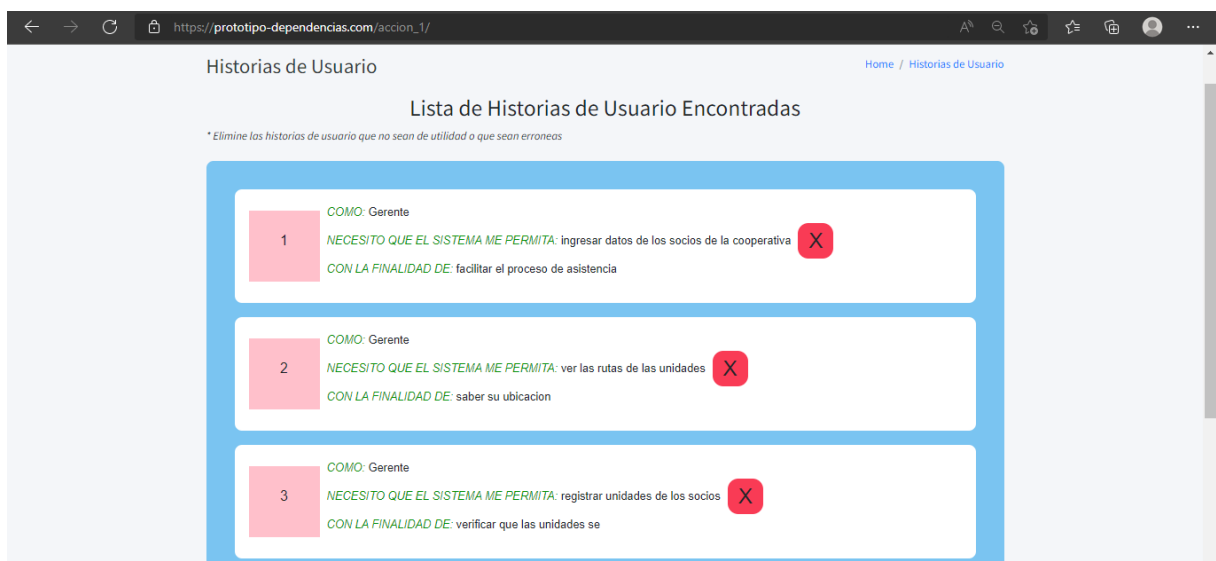


Figura 4.5. Lista de historias de usuario.

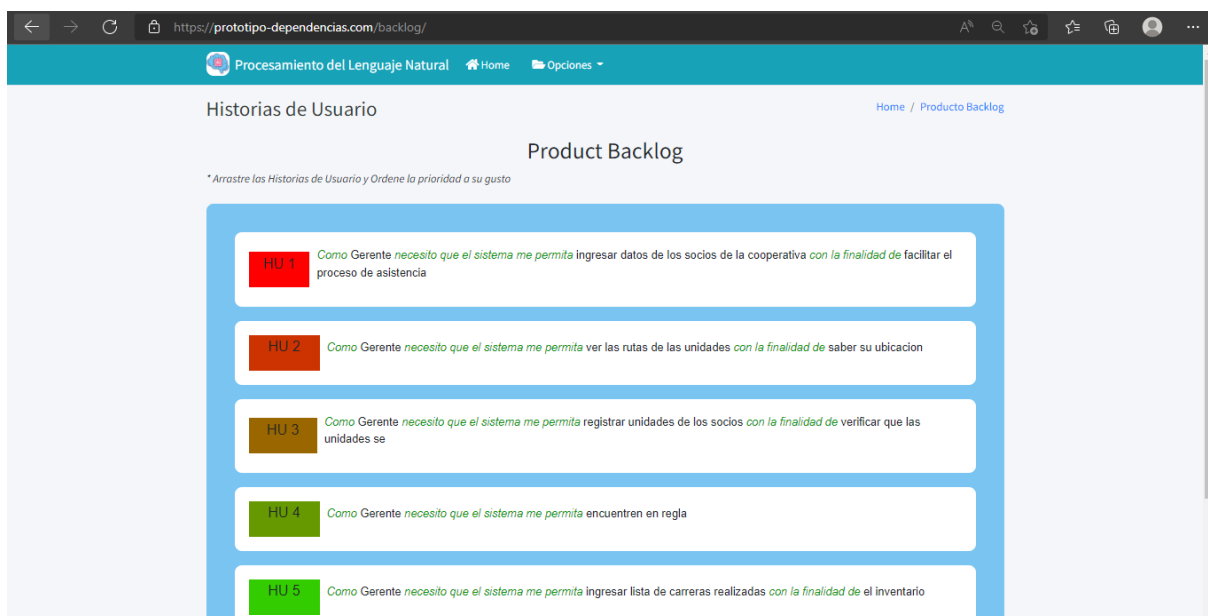


Figura 4.6. Generar Product Backlog.

c) Sprint Review 3

Caso de prueba: Ordenar prioridad de historias de usuario y mejorar algoritmo PLN.

Objetivo: Otorgar al usuario del sistema la capacidad de ordenar las historias de usuario para determinar la prioridad además de mejorar el algoritmo de procesamiento de lenguaje natural para detectar de mejor manera las historias de usuario.

Alcance: La historia de usuario 5 tiene como función generar el Producto Backlog y la historia de usuario 6 mejorar el algoritmo para detectar de manera más efectiva las historias de usuario.

Fecha: 14 de enero de 2022.

Responsables: Carlos Alcasiga, Bryan Tarco.

Tabla 4.12. Caso de prueba 03.

# Caso de prueba.	Ordenar prioridad de historias de usuario y mejorar algoritmo PLN.
# Sprint Review.	3
# H.U.	HU005, HU006.
Condiciones.	Usuario del prototipo podrá visualizar Product Backlog y ordenar la prioridad de historias de usuario.
Entradas.	1. Usuario puede mover de forma manual las historias de usuario que conforman el Product Backlog.
Evaluación de la Prueba.	Prueba superada.

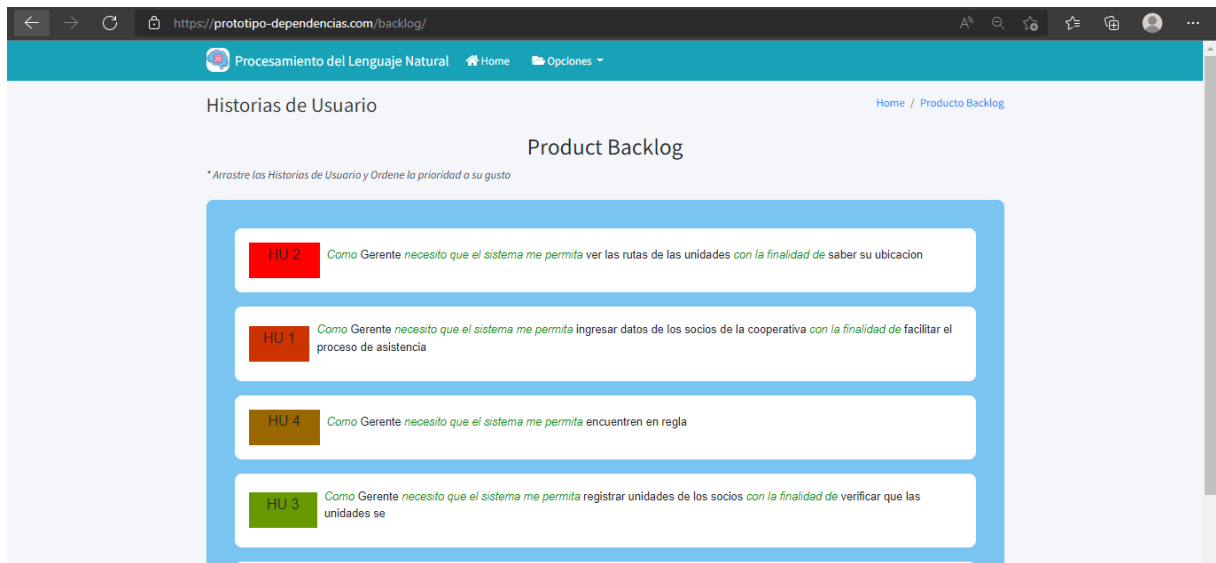


Figura 4.7. Ordenar Historias de Usuario.

d) Sprint Review 4

Caso de prueba: Generar gráfico de los microservicios y las relaciones de dependencia entre los mismos.

Objetivo: Verificar nueva pantalla en el cual se pueda graficar los microservicios y la relación de dependencia.

Alcance: Realizar la historia de usuario 7 que tiene como función generar un gráfico de los microservicios y la relación de dependencia con respecto a las historias de usuario.

Fecha: 28 de enero de 2022.

Responsables: Carlos Alcasiga, Bryan Tarco.

Tabla 4.13. Caso de prueba 04.

# Caso de prueba.	Generar el gráfico de los microservicios y las relaciones de dependencia entre los mismos.
# Sprint Review.	4
# H.U.	HU007
Condiciones.	Usuario del prototipo podrá visualizar gráficos de microservicios basados en historias de usuario.
Entradas.	1. El Usuario presiona el botón Gráfico de dependencias.
Evaluación de la Prueba.	Prueba superada.



Figura 4.8. Relación de dependencias entre microservicios.

The screenshot shows the same web browser as Figure 4.8, but with a modal window open over the dependency graph. The modal is titled "Información de la Historia de Usuario" and contains the following information:

- ID:** H1
- Descripción:** Como Gerente quiero que el sistema ingrese datos de los socios de la cooperativa para facilitar el proceso de asistencia
- Depedencias:**
 - H3 -> H1

 At the bottom of the modal is a blue button labeled "Aceptar" with a checkmark icon.

Figura 4.9. Descripción de dependencias entre microservicios.

5. ANÁLISIS Y DISCUSIÓN DE RESULTADOS

5.1. Análisis de resultados de acuerdo a las métricas de dependencia que permita verificar que la hipótesis se cumplió.

La métrica utilizada en el prototipo desarrollado es la Similitud Semántica apoyándose de la librería spaCy que permite calcular el grado de semejanza entre las entidades o sustantivos de las historias de usuario, esta similitud está en un rango de 0 a 1, siendo 1 una similitud exacta (entidades iguales) y 0 no existe similitud. Para este cálculo, primero se agrupan las entidades del relato del usuario que tengan una similitud semántica mayor a 0.75 y luego se comparan las entidades de cada grupo con las entidades de cada historia de usuario y se le asigna el identificador del grupo a la historia de usuario si alguna de sus entidades tiene una similitud semántica mayor a 0.85. Esto debido a que el prototipo se basa en que a partir de una historia de usuario se genera un microservicio, entonces al determinar la similitud semántica de varias historias de usuario referidas a un propósito similar, se pueden evidenciar los problemas de dependencia. En el gráfico que proporciona el prototipo se muestra la relación de dependencia entre microservicios, entonces, el desarrollador basándose en los problemas que determina el prototipo puede eliminar esas dependencias agrupando los microservicios relacionados en un solo microservicio y de esta manera cumplir uno de los principios de las aplicaciones basadas en microservicios, esto es, un microservicio tiene su propia base de datos independiente y solo se accede a los datos a través de la interfaz del servicio.

Además, es importante tomar en cuenta que para determinar la dependencia total del sistema que se quiere desarrollar se utilizaron las siguientes métricas: Importancia Absoluta del Microservicio (AIS) que indica el valor de importancia de un microservicio correspondiente al número de microservicios que dependerán del mismo para su correcto funcionamiento, Dependencia Absoluta del Microservicio (ADS) esta métrica es utilizada para determinar la cantidad de microservicios necesarios para el correcto funcionamiento de un microservicio, Interdependencia entre los Microservicios (SIY) donde se calcula la cantidad de microservicios que tienen dependencia bi-direccional, por último se calcula los totales de cada una de las métricas utilizando las fórmulas (3.5),(3.7),(3.9), para luego calcular la dependencia total del sistema en base a la fórmula (3.11), con este resultado se puede realizar una comparativa entre el grado de dependencia total y el número de microservicios que conforma el sistema.

En conclusión, un prototipo inteligente para el diseño de aplicaciones basadas en microservicios puede ayudar a reducir la incidencia de problemas de dependencia en este tipo de aplicaciones. En base a las métricas de dependencia aplicadas, se determina la dependencia entre los microservicios y la dependencia total del sistema, es decir, el equipo de desarrollo podrá apoyarse del prototipo para ver los posibles problemas de dependencias y tomar decisiones en etapas tempranas del desarrollo.

5.2. Validación del Prototipo por Juicio de Expertos

Para realizar la validación del prototipo se han escogido a tres ingenieros que tienen un perfil de expertos en el área de aplicaciones basadas en microservicios y procesamiento de lenguaje natural, los perfiles profesionales se encuentran detallados en el **Anexo G**.

Para cuantificar el correcto funcionamiento del “Prototipo Inteligente” se procede a validarlos de manera individual de acuerdo con la siguiente escala:

Tabla 5.1. Escala de valoración del correcto funcionamiento del “Prototipo Inteligente”.

ESCALA CUANTITATIVA	EQUIVALENCIA CUALITATIVA
1	Totalmente en desacuerdo
2	En desacuerdo
3	En desacuerdo más que en acuerdo
4	De acuerdo más que en desacuerdo
5	De acuerdo
6	Totalmente de acuerdo

Los tres expertos, proceden a revisar el prototipo en base a tres componentes principales que son: procesamiento de lenguaje natural, problemas de dependencia y utilidad del prototipo, y se obtienen los siguientes resultados que se describen en el Tabla 5.2.

Tabla 5.2. Resultados de las encuestas

Resultados de las encuestas aplicadas a expertos							
ENCUESTADOS	ITEMS						
	Procesamiento de lenguaje Natural			Problemas de dependencia		Utilidad del prototipo	
	item1	item2	item3	item4	item5	item6	item7
EXTERTO 1	5	6	5	5	5	6	6
EXTERTO 2	6	6	5	6	5	6	6
EXTERTO 3	5	5	4	5	5	5	5

5.2.1. Confiabilidad de la Validación

Luego de realizado el consenso de los criterios emitidos por los expertos se aplica la prueba de Alfa de Cronbach para relacionar los criterios correspondientes a procesamiento de lenguaje natural, problemas de dependencia y utilidad del prototipo.

El cálculo se realiza mediante la siguiente fórmula:

$$\alpha = \frac{K}{K-1} \left[1 - \frac{\sum Vi}{Vt} \right] \quad (5.1)$$

Donde:

K = número de ítems

Vi= varianza del ítem i

Vt= varianza de los puntajes brutos de los sujetos

Tabla 5.3. Alfa de Cronbach para la validación de expertos.

Detalle	Valor
K (número de ítems)	7
Vi (Varianza de cada ítem)	1,7
Vt (varianza total)	6,8
a(Alfa)	0,89

Una vez aplicado la formula se obtiene como resultado del Alfa de 0,89 y de acuerdo con el grado de confiabilidad del Alfa de Cronbach que refleja la Tabla 4.1 el cual establece que si la equivalencia es de 0,81 a 1,0 garantiza que la confiabilidad es Alta, por ende se puede establecer que la propuesta investigativa ha sido realizada de manera satisfactoria y se determina que el prototipo inteligente para determinar problemas de dependencia en aplicaciones basadas en microservicios si ayudará al equipo de desarrollo a mejorar el diseño de aplicaciones basadas en microservicios.

5.3. Valoración del Proyecto de Investigación

5.3.1. Valoración Económica

Para el desarrollo del proyecto de investigación se ha tomado en cuenta diferentes elementos los mismos que son necesarios para complementar el proyecto de investigación como los gastos directos e indirectos detallados a continuación.

5.3.2. Gastos Directos del Proyecto

Por parte de los gastos directos se tiene un conjunto de herramientas de software que son necesarios para el desarrollo del Prototipo, en donde los investigadores consideran oportuno emplear herramientas de software libre con licenciamiento gratuito, sin embargo, los gastos directos principales es el costo derivado del proceso de desarrollo, que se estima a través de los puntos de historia llegando a tener así un total de 189 puntos que se detallan en el **Anexo E**. Los valores de gastos directos del proyecto se encuentran resumidos en la siguiente tabla:

Estimación del proyecto a través de Puntos de Historia

Tabla 5.4. Gastos directos del proyecto.

Gastos Directos			
Detalle	Cantidad	Valor Unitario	Total
Internet (Investigadores)	12 meses	\$20.00	\$480.00
Python (Lenguaje de Programación)	2	Licencia gratuita	\$0.00
Navegador Web Google Chrome	2	Licencia gratuita	\$0.00
Navegador Web Firefox	2	Licencia gratuita	\$0.00
PostgreSQL	1	Licencia gratuita	\$0.00
SQLite	1	Licencia gratuita	\$0.00
Javascript	2	Licencia gratuita	\$0.00
Visual Studio Code	2	Licencia gratuita	\$0.00
Paquete de Office 2016 (Documentación)	2	\$40.00	\$80.00
spaCy	2	Licencia gratuita	\$0.00
Speech Recognition	2	Licencia gratuita	\$0.00
Django	2	Licencia gratuita	\$0.00
Bootstrap	2	Licencia gratuita	\$0.00
Desarrollo de prototipo inteligente.	1	\$4.560,00	\$4.560,00
TOTAL			\$5.120.00

5.3.3. Gastos Indirectos del Proyecto

El total correspondiente a los gastos indirectos a considerar para el desarrollo del Prototipo en relación de la inversión económica requerida para los gastos directos representa un valor inferior como se puede visualizar a continuación:

Tabla 5.5. Gastos indirectos del proyecto.

Gastos Indirectos			
Detalle	Cantidad	Valor Unitario	Total
Pasajes	40	\$1.00	\$40.00
Alimentación	40	\$2.50	\$100.00
Impresiones	150	\$0.15	\$22.50
Comunicación	15	\$3	\$45.00
TOTAL			\$207.50

5.3.4. Gasto Total del Proyecto

Una vez obtenido el total de los gastos directos e indirectos se procede a calcular el gasto total de Proyecto de Investigación en donde es importante considerar el 10% para imprevistos obteniendo como resultado.

Tabla 5.6. Gastos Total del Proyecto

Gasto Total del Proyecto	
Detalle	Total
Total de Gastos Directos	\$5120.00
Total de Gastos Indirectos	\$207.50
Gastos Directos + Gastos Indirectos	\$5327.50
Imprevistos (10%)	\$532.75
TOTAL	\$5.860,25

6. CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

- La investigación bibliográfica ha sido fundamental para el desarrollo del proyecto de investigación, puesto que a través de la revisión de antecedentes conceptuales se pudo definir las herramientas para el procesamiento de lenguaje natural como es el caso de la librería spaCy, además de los problemas en aplicaciones basadas en microservicios principalmente de dependencia.
- El Desarrollo del Prototipo Inteligente está basado en técnicas de procesamiento de lenguaje natural enfocadas principalmente en la librería spaCy, el mismo que permite procesar texto para así obtener historias de usuario, además de utilizar el nivel de similitud semántica para ayudar a determinar los problemas de dependencia en aplicaciones basadas en microservicios.
- La validación del “Prototipo inteligente” atreves del método de Juicio de Expertos y la técnica para el análisis de datos por medio del Coeficiente del Alfa de Cronbach y se obtiene como resultado 0,89 y de acuerdo con el grado de confiabilidad del Alfa de Cronbach se establece que si la equivalencia es de 0,81 a 1,0 garantiza que la confiabilidad es Alta, de modo que el “Prototipo Inteligente” tiene un alto grado de confiabilidad.

6.2. RECOMENDACIONES

- Es importante utilizar un gestor de bibliografías para recabar información de diferentes fuentes, mismas que tienen que garantizar la confiabilidad de un proyecto de investigación.
- Combinar diferentes técnicas de procesamiento de lenguaje natural para el desarrollo del prototipo para de esta manera obtener mejores resultados en cuanto al procesamiento de texto y verificación cálculos de dependencia entre microservicios.
- Ampliar las temáticas a estudiar sobre los problemas en el desarrollo de aplicaciones basadas en microservicios como es el caso de la complejidad, debido a que es un factor muy importante y complejo de encontrar en las etapas tempranas del desarrollo de un proyecto.

7. BIBLIOGRAFÍA

- [1] M. A Bonden, *Inteligencia Artificial*, Primera Ed. 28006 Madrid: Turner Publicaciones S.L., 2017.
- [2] L. P. Rouhiainen, “Inteligencia artificial,” in *Inteligencia artificial 101 cosas que debes saber hoy sobre nuestro futuro*, Noviembre., 2018 Editorial Planeta, S.A., Ed. 08034 Barcelona: B. 24.220-2018, 2018, p. 17.
- [3] M. Virginie, *Inteligencia Artificial para desarrolladores*, Editions E. Cornella de Llobergat Barcelona, 2018.
- [4] Iberdrola.S.A, “APLICACIONES PRÁCTICAS DE LA INTELIGENCIA ARTIFICIAL,” 2019. <https://www.iberdrola.com/innovacion/que-es-inteligencia-artificial> (accessed Feb. 04, 2022).
- [5] S. J. Rusell and P. Norvig, *Artificial intelligence : a modern approach / Stuart J. Russell and Peter Norvig.*, Fourth edi. Updated edition of popular textbook on Artificial Intelligence., 2021.
- [6] J. Luna Gonzales, “Tipos de aprendizaje automático,” *SoldAI*, 2018. <https://medium.com/soldai/tipos-de-aprendizaje-automático-6413e3c615e2> (accessed Nov. 28, 2021).
- [7] L. Judith Sandoval, “ALGORITMOS DE APRENDIZAJE AUTOMÁTICO PARA ANÁLISIS Y PREDICCIÓN DE DATOS,” *Esc. Espec. en Ing. ITCA - FEPADE*, p. 38, 2018, [Online]. Available: http://www.redicces.org.sv/jspui/bitstream/10972/3626/1/Art6_RT2018.pdf.
- [8] F. H. Vera Rivera, “MODELO INTELIGENTE DE ESPECIFICACIÓN DE LA GRANULARIDAD DE APLICACIONES BASADAS EN MICROSERVICIOS.,” Universidad del Valle Cali-Colombia, 2019.
- [9] SAS Institute Inc, “Procesamiento del lenguaje natural,” *El procesamiento del lenguaje natural (NLP, por sus siglas en inglés) es una rama de la inteligencia artificial que ayuda a las computadoras a entender, interpretar y manipular el lenguaje humano. NLP toma elementos prestados de muchas disciplinas, incluy*, 2021. https://www.sas.com/es_ar/insights/analytics/what-is-natural-language-processing-nlp.html (accessed Nov. 28, 2021).
- [10] A. C. Vásquez, H. Huerta Vega, and J. P. Quispe, “Procesamiento de lenguaje natural,” *Rev. Ing. Sist. e Informática*, vol. 6, no. 2, pp. 47–52, 2009, [Online]. Available:

- <https://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/view/5923>.
- [11] Baoss Analytics Everywhere, “Procesamiento del Lenguaje Natural (PLN),” *Usos de PLN*, 2020. <https://www.baoss.es/procesamiento-del-lenguaje-natural-pln-con-python/>.
- [12] I. de I. del Conocimiento, “Componentes del procesamiento del lenguaje natural,” 2019. <https://www.iic.uam.es/inteligencia/que-es-procesamiento-del-lenguaje-natural/> (accessed Dec. 12, 2021).
- [13] D. A. G. L. T. Burquest, *Análisis fonológico Un enfoque funcional*, Third. Dallas, 2009.
- [14] I. BOSQUE, “La morfología. Introducción a la lingüística.,” Madrid: Al., 2016.
- [15] Itelligent, “Procesamiento del Lenguaje Natural y sus aplicaciones,” *Itelligent*, 2017. <https://itelligent.es/es/procesamiento-del-leguaje-natural-aplicaciones/> (accessed Dec. 23, 2021).
- [16] R. Colle, “El análisis de contenido de las comunicaciones,” Sociedad L., A. A. Abreu, Ed. 2011, p. 15.
- [17] G. De la Calle Velasco, “Modelo basado en técnicas de procesamiento de lenguaje natural para extraer y anotar información de publicaciones científicas,” Madrid, pp. 35–37, Apr. 2014.
- [18] spaCy, “Que es spaCy,” 2021. <https://spacy.io/usage/spacy-101>.
- [19] stratebi OPEN BUSINESS INTELLIGENCE, “spaCy Business Intelligence-Machine Learning- Big Data,” Jun. 24, 2020.
- [20] spaCy, “Library Architecture spaCy,” 2021. <https://spacy.io/api> (accessed Nov. 28, 2021).
- [21] The Computer Language Company Inc., “Aplication.” <https://www.computerlanguage.com/results.php?definition=Aplication>.
- [22] J. Lewis and M. Fowler, “Microservices,” 2014. <https://martinfowler.com/articles/microservices.html> (accessed Nov. 29, 2021).
- [23] C. Nielsen Djernæs, “Investigate availability and maintainability within a microservice architecture,” *The microservice architectural style is a way of building software systems. A microservice system consists of many small services that communicate with each other over a network. Each service has an individual software development lifecycle. To work corre*, 2015.

- [24] F. Robin and H. Athon, “Performance characteristics between monolithic and microservice-based systems,” 2017. <http://www.diva-portal.org/smash/record.jsf?dswid=-1035&pid=diva2%3A1119785> (accessed Nov. 28, 2021).
- [25] M Fussell, “Why a microservices approach to building applications?,” 2017. <https://docs.microsoft.com/es-es/azure/service-fabric/service-fabric-overview-microservices> (accessed Nov. 28, 2021).
- [26] K. Nane, “About Microservices, Containers and their Underestimated Impact on Network Performance,” *Conference: Cloud Computing 2015*, 2015. https://www.researchgate.net/publication/273456042_About_Microservices_Containers_and_their_Underestimated_Impact_on_Network_Performance?channel=doi&linkId=550f34070cf2ac2905ae0177&showFulltext=true (accessed Nov. 29, 2021).
- [27] L. S. Nicola Dragoni, Saverio Giallorenzo, Alberto Lluch Lafuente, Manuel Mazzara, Fabrizio Montesi, Ruslan Mustafin, “Microservices: yesterday, today, and tomorrow,” 2016. <http://arxiv.org/abs/1606.04036> (accessed Nov. 30, 2021).
- [28] A. F. Saransig Chiza, ““ANÁLISIS DE RENDIMIENTO ENTRE UNA ARQUITECTURA MONOLÍTICA Y UNA ARQUITECTURA DE MICROSERVICIOS – TECNOLOGÍA BASADA EN CONTENEDORES,”” 2018.
- [29] D. Fernández Carrasco, Oscar M García León and A. Beltrán Benavides, “Un enfoque actual sobre la calidad del software,” *Scielo*, no. 1024–9435, 2018, [Online]. Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94351995000300005.
- [30] D. Lopez and M. Edgar, “Arquitectura de Software basada en Microservicios para Desarrollo,” in *Séptima Conferencia de Directores de Tecnología de Información, TICAL 2017 Gestión de las TICs para la Investigación y la Colaboración, San José, del XX al XX de julio de 2017*, 2017, pp. 5–7.
- [31] M. Cojocar, A. Uta, and A. Oprescu, “MicroValid: A Validation Framework for Automatically Decomposed Microservices,” *Int. Conf. Cloud Comput. Technol. Sci.*, pp. 78–86, 2019, doi: 10.1109.
- [32] spaCy Usage Documentation - Linguistic Features, “Word vectors and semantic similarity,” 2020. <https://spacy.io/usage/linguistic-features#vectors-similarity> (accessed Feb. 15, 2022).
- [33] S. Newman, *Building microservices - Designing Fine-Grained Systems.*, 2nd ed. 2021.

- [34] L. Walker, “IBM business transformation enabled by service-oriented architecture,” *BM Syst. J.*, vol. 46, pp. 651–667, 2007, doi: 10.1147/sj.464.0651.
- [35] S. Dharmendra, M. Rezai, and R. Hill, “Microservices: Granularity vs. Performance,” 2017, p. 2, doi: arXiv:1709.09242v1.
- [36] H. S. Roberto, F. C. Carlos, and del P. B. L. María, *Metodología de la investigación*, Sexta Edic. México: elosopanda, 2014.
- [37] F. . G. Arias, “El proyecto de investigación,” 6ta Edicio., 2012.
- [38] QuestionPro, “Definición de encuesta.” <https://www.questionpro.com/es/una-encuesta.html> (accessed Feb. 15, 2022).
- [39] A. C.-A. Heidi Celina Oviedo1, “Aproximación al uso del coeficiente alfa de Cronbach,” *Scielo*, no. vol.34, 2005, doi: 0034-7450.
- [40] C. J. B. Abundis, “Metodologías para desarrollar software seguro,” *ReCIBE. Rev. electrónica Comput. Informática Biomédica y Electrónica*, vol. 3, no. Universidad de Guadalajara, p. 4,3, 2013.
- [41] D. C. y A. Alfaro, “Método de aseguramiento de la calidad en una metodología de desarrollo de software,” *Ingeniare. Rev. Chil. Ing.*, vol. 26, p. 1, 2018.
- [42] J. D. F. Martines, “Metodologías ágiles para el desarrollo de software,” *Prospect. Univ. Autónoma del Caribe Colomb.*, vol. 11, p. 30, 2013.
- [43] Gitbook, “Desarrollo de Software Ágil,” 2018. <https://10pines.gitbook.io/desarrollo-de-software-agil-en-10pines/estimaciones> (accessed Feb. 11, 2022).
- [44] A. A. Coach, “Prácticas Ágiles de Scrum,” 2018. <https://www.atlassian.com/es/agile/scrum> (accessed Feb. 11, 2022).
- [45] Institute Scrum, “Release_Planning,” 2018. https://www.scrum-institute.org/Release_Planning.php (accessed Feb. 11, 2022).
- [46] H. Kniberg, *Scrum y XP desde las trincheras Como hacemos Scrum*. editor de InfoQ.com., 2007.

8. ANEXOS

Anexo A: Datos informativos estudiante 1.

Hoja de Vida



DATOS PERSONALES

APELLIDOS: Tarco Taípe

NOMBRES: Bryan Jhoel

NACIONALIDAD: Ecuatoriana

FECHA DE NACIMIENTO: 22 de marzo de 1999

ESTADO CIVIL: Soltero

CEDULA: 0504100918

DIRECCIÓN: Parroquia Juan Montalvo, calle Quijano y Ordoñez

TELÉFONO: 0984725989

EMAIL: bryan.tarco0918@utc.edu.ec

ESTUDIOS REALIZADOS

INSTRUCCIÓN PRIMARIA: Escuela “Simón Bolívar”

INSTRUCCIÓN SECUNDARIA: Colegio “Ramón Barba Naranjo”

ESTUDIOS ACTUALES: Universidad Técnica de Cotopaxi

Anexo B: Datos informativos estudiante 2.

Hoja de Vida



DATOS PERSONALES

APELLIDOS: Alcasiga Quilumbaquin

NOMBRES: Carlos Stalin

NACIONALIDAD: Ecuatoriana

FECHA DE NACIMIENTO: 13 de diciembre de 1997

ESTADO CIVIL: Soltero

CÉDULA: 1727236026

DIRECCIÓN: Barrio San Felipe, Avenida Simón Rodríguez y México S/N

TELÉFONO: 0967459735

EMAIL: carlos.alcasig6026@utc.edu.ec

ESTUDIOS REALIZADOS

INSTRUCCIÓN PRIMARIA: Escuela “Jose Acosta Vallejo”

INSTRUCCIÓN SECUNDARIA: Colegio “Cesar Augusto Tamayo Medina

ESTUDIOS ACTUALES: Universidad Técnica de Cotopaxi

Anexo C: Historias de usuario

Historia de usuario 1.

HISTORIA DE USUARIO (HU)			
Código HU:	HU001	Fecha:	25/10/2021
Sprint:	S001	Prioridad:	Alta
Actor(es):	Desarrollador de Software	Puntos:	22
Nombre:	Ingresar datos al Prototipo Inteligente.		
Descripción:			
Como: Desarrollador de Software “Usuario”			
Quiero: Ingresar una solicitud o relato al prototipo de forma oral o por archivo de audio.			
Para: Transformar el relato hablado del usuario a texto para facilitar el procesamiento de la solicitud.			
Restricciones:			
Criterios de aceptación:			
Escenario 1			
Dado: Formulario para ingresar solicitud del usuario al prototipo de forma oral o por archivo de audio.			
Cuando: El prototipo esté funcionando de manera correcta y guarde el relato sin palabras ambiguas.			
Entonces: El prototipo podrá realizar la transformación de relato a texto de manera correcta.			
Escenario 2			
Dado: Formulario para ingresar solicitud del usuario al prototipo.			
Cuando: Datos recogidos por el micrófono no son correctos.			
Entonces: El prototipo mostrará los errores de ortografía en el formulario.			
DoD (Definition of Done):			
<ul style="list-style-type: none"> - Las funcionalidades son correctas. - Se cumplen con los criterios de aceptación. - Las pruebas unitarias se completaron correctamente. - El código funciona en armonía con las demás partes. 			

Historia de usuario 2.

HISTORIA DE USUARIO (HU)			
Código HU:	HU002	Fecha:	08/11/2021
Sprint:	S001	Prioridad:	Alta
Actor(es):	Desarrollador de Software	Puntos:	15
Nombre:	Ingresar el tipo de usuario.		
Descripción:			

<p>Como: Desarrollador de Software “Usuario”</p> <p>Quiero: Poder escribir cual es el usuario responsable.</p> <p>Para: Identificar las historias de usuario que corresponde a ese usuario.</p>
<p>Restricciones:</p>
<p>Criterios de aceptación:</p> <p>Escenario 1 Dado: Cuadro de edición. Cuando: Digita un usuario responsable. Entonces: El prototipo permite el ingreso de la solicitud del usuario.</p> <p>Escenario 2 Dado: Cuadro de edición. Cuando: No ingresa ningún usuario en el cuadro de edición. Entonces: El prototipo mostrará mensaje emergente “Rellene este campo”.</p>
<p>DoD (Definition of Done):</p> <ul style="list-style-type: none"> - Las funcionalidades son correctas. - Cumple con los criterios de aceptación. - La prueba unitaria se completó satisfactoriamente. - El código funciona en armonía con las demás partes.

Historia de usuario 3.

HISTORIA DE USUARIO (HU)			
Código HU:	HU003	Fecha:	22/11/2021
Sprint:	S002	Prioridad:	Alta
Actor(es):	Desarrollador de Software	Puntos:	32
Nombre:	Reconocer historias de usuario.		
Descripción:	<p>Como: Desarrollador de Software “Usuario”</p> <p>Quiero: encontrar las actividades en la solicitud ingresada</p> <p>Para: Observar las historias de usuario en su plantilla correspondiente.</p>		
Restricciones:			
Criterios de aceptación:	<p>Escenario 1 Dado: La solicitud del usuario. Cuando: La solicitud contiene acciones a realizar. Entonces: El prototipo procesa la solicitud y muestra en historias de usuario en una nueva pantalla.</p> <p>Escenario 2 Dado: La solicitud del usuario.</p>		

Cuando: La solicitud no contiene ninguna acción.
 Entonces: El prototipo muestra una pantalla con una lista vacía.

DoD (Definition of Done):

- Las funcionalidades son correctas.
- Los criterios de aceptación se cumplieron.
- Las pruebas unitarias se completaron satisfactoriamente.
- El código funciona en armonía con las demás partes.

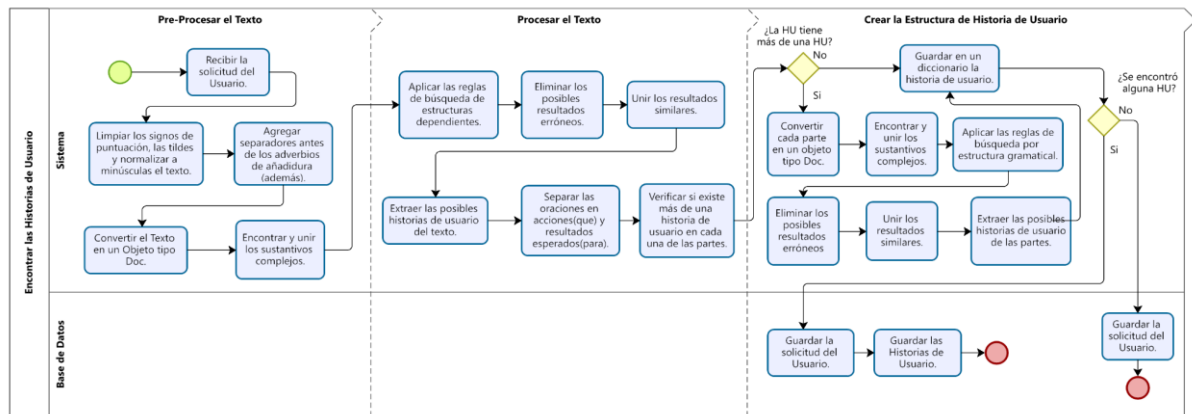


Diagrama BPMN del proceso para encontrar las historias de usuario.

La Historia de Usuario 3 “Encontrar historias de usuario” es una de las más complejas del sistema, en base a esta se generan los procesos posteriores, se divide en tres etapas, 1) Preprocesar el texto, donde se limpian los signos de puntuación, las tildes, se normaliza a minúsculas el texto, se agregan separadores (puntos) en sitios clave, se convierte el texto en un objeto de tipo Doc para unir los sustantivos complejos. 2) Procesar el texto, aquí se aplican reglas de búsqueda de estructuras dependientes, se limpian los resultados erróneos, y se encuentran las posibles historias de usuario. 3) Crear la estructura de historia de usuario, se verifica que cada historia de usuario solo tenga una acción y un propósito, luego se dividen en partes y se guarda la información necesaria para cada historia de usuario.

Historia de usuario 4.

HISTORIA DE USUARIO (HU)			
Código HU:	HU004	Fecha:	06/12/2021
Sprint:	S002	Prioridad:	Alta
Actor(es):	Desarrollador de Software	Puntos:	28
Nombre:	Generar Product Backlog.		
Descripción:			

<p>Como: Desarrollador de Software “Usuario”</p> <p>Quiero: Generar Product Backlog.</p> <p>Para: Visualizar el conjunto de historias de usuario.</p> <p>Restricciones:</p>
<p>Criterios de aceptación:</p> <p>Escenario 1 Dado: Interfaz con historias de usuario. Cuando: Presiona el botón “Continuar”. Entonces: El prototipo procesa solicitud y muestra “Product Backlog” en una nueva pantalla.</p> <p>Escenario 2 Dado: 0 Historias de Usuario Encontradas. Cuando: Presiona el botón “Continuar”. Entonces: El prototipo muestra una lista vacía en el Product Backlog.</p> <p>DoD (Definition of Done):</p> <ul style="list-style-type: none"> - Las funcionalidades son correctas. - Los criterios de aceptación se cumplieron. - Las pruebas unitarias se completaron satisfactoriamente. - El código funciona en armonía con las demás partes.

Historia de usuario 5.

HISTORIA DE USUARIO (HU)			
Código HU:	HU005	Fecha:	20/12/2021
Sprint:	S003	Prioridad:	Media
Actor(es):	Desarrollador de Software	Puntos:	20
Nombre:	Ordenar prioridad de las historias de usuario.		
Descripción:	<p>Como: Desarrollador de Software “Usuario”</p> <p>Quiero: Tener la opción de ordenar manualmente la prioridad de las historias de usuario.</p> <p>Para: Poder priorizar el orden de desarrollo.</p>		
Restricciones:			
Criterios de aceptación:	<p>Escenario 1 Dado: Interfaz con Product Backlog. Cuando: El usuario desea mover el orden de prioridad. Entonces: Elige la Historia de usuario que desea mover y arrastra dando clic sostenido.</p>		
DoD (Definition of Done):	<ul style="list-style-type: none"> - Las funcionalidades son correctas. - Se cumplieron con los criterios de aceptación. - La prueba unitaria se cumplió satisfactoriamente. 		

Historia de usuario 6.

HISTORIA DE USUARIO (HU)			
Código HU:	HU006	Fecha:	03/01/2022
Sprint:	S003	Prioridad:	Alta
Actor(es):	Desarrollador de Software	Puntos:	32
Nombre:	Mejorar el algoritmo para el Procesamiento de Lenguaje Natural.		
Descripción:	Como: Desarrollador de Software “Usuario” Quiero: Mejorar el algoritmo para el Procesamiento de Lenguaje Natural. Para: Obtener historias de usuario de una manera correcta.		
Restricciones:			
Criterios de aceptación:	Escenario 1 Dado: Formulario de solicitud completo. Cuando: Presiona el botón “ Ingresar y procesar ”. Entonces: El prototipo procesa solicitud y muestra en historias de usuario de manera correcta en una nueva pantalla.		
DoD (Definition of Done):	<ul style="list-style-type: none">- Las funcionalidades son correctas.- Se cumplieron con los criterios de aceptación.- La prueba unitaria se completó satisfactoriamente.- El código funciona en armonía con las demás partes.		

Historia de usuario 7.

HISTORIA DE USUARIO (HU)			
Código HU:	HU007	Fecha:	27/01/2022
Sprint:	S004	Prioridad:	Alta
Actor(es):	Desarrollador de Software	Puntos:	40
Nombre:	Graficar un gráfico de los microservicios y la relación de dependencia en los mismos.		
Descripción:	Como: Desarrollador de Software “Usuario” Quiero: Graficar cada una de las historias de usuario que serán reflejados en microservicios. Para: Ver las relaciones de dependencia.		
Restricciones:			
Criterios de aceptación:			

Escenario 1

Dado: Interfaz gráfica de Product Backlog.

Cuando: Presiona el botón “Continuar”.

Entonces: El prototipo procesa solicitud y muestra el gráfico de microservicios y dependencia en una nueva pantalla.

Escenario 2

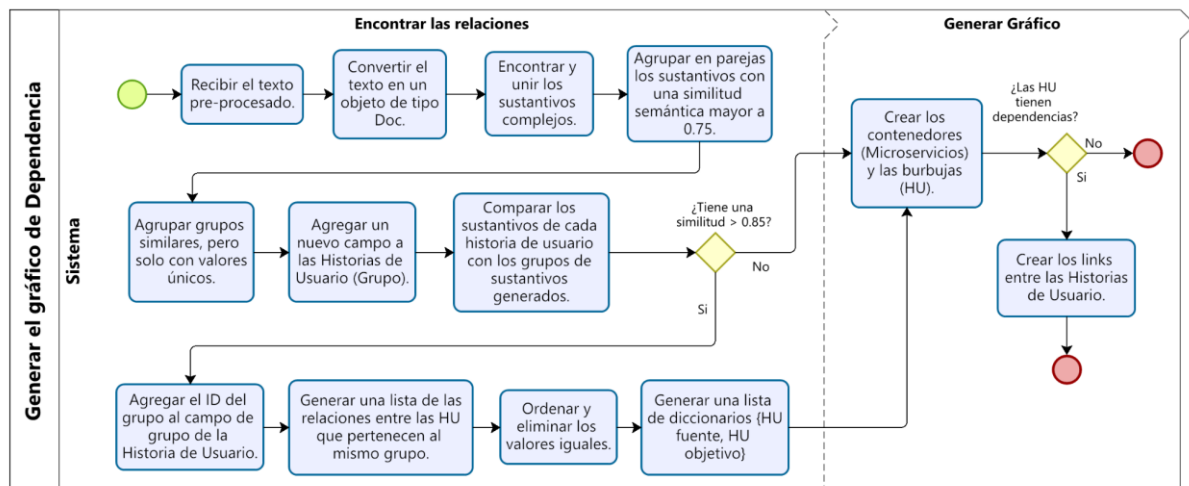
Dado: Interfaz gráfica de microservicios.

Cuando: Presiona el sobre el botón “Microservicio”.

Entonces: El prototipo procesa solicitud y muestra una ventana emergente en donde se muestra el detalle del microservicio.

DoD (Definition of Done):

- Las funcionalidades son correctas.
- Se cumplen con los criterios de aceptación.
- Las pruebas unitarias se completaron satisfactoriamente.
- El código funciona en armonía con las demás partes.

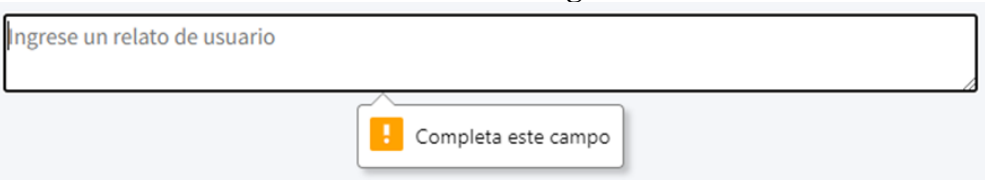


BPMN para el proceso de generación del gráfico de microservicios.

La Historia de Usuario 7, donde se genera el gráfico de dependencia entre los microservicios cuenta con dos etapas, 1) Encontrar las relaciones, se agrupan los sustantivos o entidades con una alta similitud semántica (> 0.75), estos grupos se comparan para cada sustantivo de la acción (que) de las historias de usuario y se marca el ID del grupo en la historia de usuario si la similitud semántica es mayor a 0.85. La segunda etapa es graficar esta información, cada historia de usuario se convierte en un microservicio y las dependencias encontradas.

Anexo D: Pruebas unitarias.

Prueba unitaria 1.

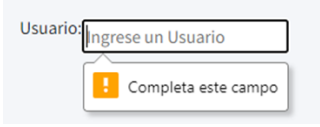
Código Prueba Unitaria	PU 001	Fecha de ejecución	25/10/2021
Historia de Usuario	HU 001		
Responsable:	Bryan Tarco, Carlos Alcasiga		
Descripción del caso de prueba	Comprobar la validación del campo de la solicitud o relato del usuario al momento de ingresar datos en el formulario principal de la aplicación.		
Prerrequisitos	- Ingresar al Sistema.		
Prueba Unitaria:	Validar que se ingresen datos a la solicitud del usuario		
Fragmento de Código	<pre> <div class="form-group contenedor"> {{ form.relatoUsuario }} <button type="submit" class="btn btn-primary btn_ingresar"> Ingresar y Procesar </button> </div> </pre>		
Resultados Esperados:	El sistema muestra un mensaje de advertencia, informando que el campo del formulario debe estar lleno.		
Datos Ingresados:	 <p>The screenshot shows a text input field with the placeholder text "Ingrese un relato de usuario". Below the field, a yellow warning icon is displayed with the message "Completa este campo".</p>		
Observaciones	Los campos principales del formulario solo están validados para que no sean enviados vacíos, debido a que el usuario puede ingresar cualquier tipo de usuario y cualquier tipo de solicitud.		

Prueba unitaria 2.

Código Prueba Unitaria	PU 002	Fecha de ejecución	25/10/2021
Historia de Usuario	HU 001		
Responsable:	Bryan Tarco, Carlos Alcasiga		
Descripción del caso de prueba	Comprobar el reconocimiento automático de voz, para transformar audio en texto.		
Prerrequisitos	- Ingresar al Sistema.		

Prueba Unitaria	Validar que el reconocedor automático de voz detecte de manera correcta los datos ingresados por voz o por un archivo de audio (.wav).
Fragmento de Código	<pre> r = sr.Recognizer() r.dynamic_energy_threshold = True r.operation_timeout = 5 with sr.Microphone() as source: print('Habla') r.adjust_for_ambient_noise(source, duration=1) audio = r.listen(source, timeout=5) try: text = r.recognize_google(audio, language='es-ES') print(text) except: print("Perdon, pero no entiendo") data['error'] = 'No se entiende' data['voz'] = text form1 = self.get_form() audio = request.FILES['subirAudio'] r = sr.Recognizer() r.energy_threshold = 300 with sr.AudioFile(audio) as source: r.adjust_for_ambient_noise(source, duration=1) sonido = r.record(source) try: print('Reading the audio') text = r.recognize_google(sonido, language='es-ES') print(text) except: print('Losiento no se puede entender...') data['audio'] = text </pre>
Resultados Esperados	El sistema transforma un audio ya sea ingresado utilizando un micrófono o un archivo de audio y lo transforma a texto.
<p>Datos Ingresados: Datos Recogidos por el Micrófono.</p> <pre> voz Habla la historia de usuario es una recolección de los requisitos de usuario </pre> <p>Datos Ingresados: Datos de un archivo de audio (.wav).</p> <pre> Reading the audio Cómo ejecutivo de cuenta Quiero poder abrir una nueva cuenta de cheques a un cliente seleccionando el código de cliente y una nueva solicitud de crédito hipotecario ambiente luego quiero y que indique Cuáles son los documentos que debo restar dito hipotecario </pre> <p>Datos Ingresados: []</p> <pre> voz Habla Ingreso a la excepcion {'error': 'listening timed out while waiting for phrase to start'} </pre>	
Observaciones	La calidad del resultado de la transformación de voz a texto depende de varios factores, uno de los más importantes es la calidad del micrófono del equipo que se esté utilizando.

Prueba unitaria 3.

Código Prueba Unitaria	PU 003	Fecha de ejecución	08/11/2021
Historia de Usuario	HU 002		
Responsable:	Bryan Tarco, Carlos Alcasiga		
Descripción del caso de prueba	Comprobar la validación de los campos al momento de ingresar datos en el formulario principal de la aplicación.		
Prerrequisitos	- Ingresar al Sistema.		
Prueba Unitaria	Validar el campo de usuario responsable.		
Fragmento de Código	<pre><div class="form-group contenedor"> <p>Usuario: </p> <input type="text" id="nombreUsuario" name="nombreUsuario" placeholder="Ingrese un Usuario" requi </div></pre>		
Resultados Esperados	El sistema muestra un mensaje de advertencia, informando que este campo del formulario debe estar lleno.		
Datos Ingresados: []	 <p>The screenshot shows a text input field with the placeholder text 'Ingrese un Usuario'. Below the field, there is a yellow warning icon and the message 'Completa este campo'.</p>		
Observaciones	El campo de Usuario responsable puede aceptar cualquier tipo de usuario.		

Prueba unitaria 4.

Código Prueba Unitaria	PU 004	Fecha de ejecución	22/11/2021
Historia de Usuario	HU 003		
Responsable:	Bryan Tarco, Carlos Alcasiga		

Descripción del caso de prueba	Comprobar que se detecten las historias de usuario con una estructura coherente.
Prerrequisitos	<ul style="list-style-type: none"> - Ingresar al Sistema. - Ingresar una solicitud y el tipo de usuario responsable.
Prueba Unitaria	Detectar y unir las entidades o sustantivos complejos que existan en el relato del usuario.
Fragmento de Código	<pre> matches = matcher_sustantivos(doc) matches = limpiar_encontrar_sustantivos(matches, doc) resultado = [] l = matches l = sorted(l) for index, span in enumerate(matches): match = [(np.array(matchi) - resultado) for matchi in l] comienzo = len(doc) if (len(match) != 0): start = min(match[0]) end = max(match[0]) + 1 act = Span(doc, start, end, label="JUNTAR OPCIONALES") # print(act) with doc.retokenize() as retokenizer: retokenizer.merge(act) final = len(doc) resultado = (comienzo - final) match.remove(match[0]) l = match </pre>
Resultados Esperados	Encontrar las entidades o sustantivos complejos y unir las partes de cada uno para crear un único token.
<p>Datos Ingresados: “usuario quiero crear un sistema que me permita procesar el texto hablado y convertirlo a texto plano además de esto requiero guardar el registro de la partida de ajedrez para conseguir los mejores resultados”</p> <pre> [[4, 3], [10, 9], [15, 16], [23, 24, 26, 27, 29], [32, 33, 34]] [un sistema, el texto, texto plano, el registro de la partida de ajedrez, los mejores resultado] </pre>	
Observaciones	Encontrar y unir las entidades o sustantivos complejas facilita la tarea de encontrar las historias de usuario.

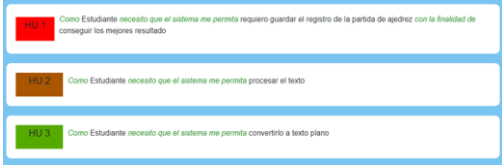
Prueba unitaria 5.

Código Prueba Unitaria	PU 005	Fecha de ejecución	22/11/2021
Historia de Usuario	HU 003		
Responsable:	Bryan Tarco, Carlos Alcasiga		
Descripción del caso de prueba	Comprobar que se detecten las historias de usuario con una estructura coherente.		

Prerrequisitos	<ul style="list-style-type: none"> - Ingresar al Sistema. - Ingresar una solicitud y el tipo de usuario responsable.
Prueba Unitaria:	
Prueba Unitaria	Encontrar los índices de comienzo y final de las historias de usuario dentro del relato o solicitud del usuario.
Fragmento de Código	<pre> matcher_dependencia = DependencyMatcher(nlp.vocab, validate=True) matcher_dependencia.add("OBJECTO_COMPUESTO", [pattern6]) matcher_dependencia.add("CASO_IR", [pattern8]) matcher_dependencia.add("CASO_SIMILAR_AL_IR", [pattern9]) matches = matcher_dependencia(doc) </pre>
Resultados Esperados	Los índices de inicio y final de las posibles historias de usuario para extraer las historias de usuario del objeto Doc.
<p>Datos Ingresados: “usuario quiero crear un sistema que me permita procesar el texto hablado y convertirlo a texto plano además de esto requiero guardar el registro de la partida de ajedrez para conseguir los mejores resultados”</p> <pre> [[2, 3, 5], [18, 19, 20, 21, 22, 23], [7, 8], [11, 13]] HISTORIA DE USUARIO: crear un sistema que me HISTORIA DE USUARIO: requiero guardar el registro de la partida de ajedrez para conseguir los mejores resultado HISTORIA DE USUARIO: procesar el texto HISTORIA DE USUARIO: convertirlo a texto plano </pre>	
Observaciones	Las historias de usuario encontradas dentro del relato del usuario pueden tener algunas ambigüedades o palabras extra.

Prueba unitaria 6.

Código Prueba Unitaria	PU 006	Fecha de ejecución	06/12/2021
Historia de Usuario	HU 004		
Responsable:	Bryan Tarco, Carlos Alcasiga		
Descripción del caso de prueba	Verificar que la estructura del Product Backlog sea la correcta.		
Prerrequisitos	<ul style="list-style-type: none"> - Ingresar al Sistema. - Solicitud del Usuario procesada. 		

Prueba Unitaria	Crear una estructura correcta para visualizar la información de las historias de usuario del Product Backlog.
Fragmento de Código	<pre> variable = Auxiliar.objects.all().last() acciones = Accion.objects.filter(aux_id=variable.id) data = [] for frase in acciones: item = {} item['id'] = frase.id item['usuario'] = frase.actor item['que'] = frase.que item['para_que'] = frase.para_que item['posicion'] = frase.posicion data.append(item) function imprimir(data){ color = generar_color(data.length); numero_colores = color; for (i = 0; i < data.length; i++){ var nuevo = '<div class="accion" id="'+data[i].id+' " data-id="'+(i + 1)+'"> ' + '<div class="div_numero" id="numero'+data[i].id+' "hp class="numero"> HU '+i+1+'</div>' + '<div class="relato"><em class="relato"> Como '+data[i].usuario + '<em class="relato"> necesito que el sistema me permita ' + data[i].que; if (data[i].para_que != 'sin proposito'){ nuevo += '<em class="relato"> con la finalidad de ' + data[i].para_que; } nuevo += '</div>' + '</div>' + '</div>'; \$('#lista-acciones').append(nuevo); ids[i] = data[i].id; } colocar_c(color); } </pre>
Resultados Esperados	Estructura del Product Backlog: ID de la historia de usuario con un color que represente el nivel de prioridad y la historia de usuario.
Datos Ingresados: Estructura de las Historias de usuario encontradas.	
	
Observaciones	La prioridad de las historias de usuario se detona por el color y por la posición donde se encuentra.

Prueba unitaria 7.

Código Prueba Unitaria	PU 007	Fecha de ejecución	20/12/2021
Historia de Usuario	HU 005		
Responsable:	Bryan Tarco, Carlos Alcasiga		
Descripción del caso de prueba	La prioridad de las historias de usuario se actualiza automáticamente cuando se altera el orden de forma manual.		

Prerrequisitos	<ul style="list-style-type: none"> - Ingresar al Sistema. - Solicitud del Usuario procesada.
Prueba Unitaria	Registrar el orden de las historias de usuario cuando se arrastra un elemento a una nueva posición.
Fragmento de Código	<pre> function ordenar(){ const lista = document.getElementById('lista-acciones'); Sortable.create(lista, { animation: 150, chosenClass: "seleccionado", dragClass: "drag", onEnd: (e) => { var item = e.item; colocar_c(numero_colores); } }); group: "lista-backlog", store: { // Guardamos el orden set: (sortable) => { const orden = sortable.toArray(); localStorage.setItem(sortable.options.group.name, orden.join('-')); console.log('se guardo el orden', sortable) guardarOrden(orden); }, //Obtener el orden get: (sortable) => { const orden = localStorage.getItem(sortable.options.group.name); return orden ? orden.split('-') : []; } } }); colocar_c(numero_colores); </pre>
Resultados Esperados	Registro del orden de prioridad de las historias de usuario.
Datos Ingresados: Arrastrar un Historia de Usuario.	
Posicion anterior: 2	backlog.js:161
Nueva Posicion: 0	backlog.js:170
Se guardo el orden ▶ (3) ['3', '2', '1']	backlog.js:179
Observaciones	La prioridad se guarda en la base de datos y de manera local en el navegador web.

Prueba unitaria 8.

Código Prueba Unitaria	PU 008	Fecha de ejecución	03/01/2022
Historia de Usuario	HU 006		
Responsable:	Bryan Tarco, Carlos Alcasiga		
Descripción del caso de prueba	Ingresar un relato más complejo para probar el reconocimiento de historias de usuario.		

<p>Prerrequisitos</p>	<ul style="list-style-type: none"> - Ingresar al Sistema. - Ingresar la solicitud del usuario y el usuario responsable.
<p>Prueba Unitaria</p>	<p>Encontrar historias de usuario en un relato o solicitud de usuario con un nivel de complejidad mayor.</p>
<p>Fragmento de Código</p>	<pre> matches.sort(key=lambda x: x[1]) preview = [] tokens_ids = [(start, end) for _, start, end in matches] for index, i in enumerate(tokens_ids): preview = i for oracion in (tokens_ids[tokens_ids.index(preview) + 1] + tokens_ids[tokens_ids.index(preview)]): if oracion[0] == preview[0] or oracion[0] > preview[0] and oracion[0] < preview[1]: valor = list(set(oracion + preview)) tokens_ids[tokens_ids.index(preview)] = [min(valor), max(valor)] if oracion in tokens_ids: tokens_ids.remove(oracion) preview = [min(valor), max(valor)] for match in tokens_ids: oracion = doc[match[0]:match[1]] limpiar = oracion[2:] if limpiar[0].text in ("que") and limpiar[1].pos == "NOUN": oracion = doc[match[0]:match[1] - 2] if limpiar[1].pos_in ("ADP", "AUX") and limpiar[1].lemma_not_in ("poder"): oracion = doc[match[0]:match[1] + 1] if limpiar[1].lemma_in ("poder") and limpiar[1].pos_in ("ADP", "AUX"): oracion = doc[match[0]:match[1] - 1] oraciones_encontradas.append(oracion) return oraciones_encontradas </pre>
<p>Resultados Esperados</p>	<p>Historias de usuario encontradas dentro de un texto con una complejidad mayor.</p>
<p>Datos Ingresados: “Como ejecutivo de cuenta quiero abrir una nueva cuenta de cheques a un cliente seleccionando el código de cliente y producto bancario, registrar una nueva solicitud de crédito hipotecario a mi cliente indique cuales son los documentos que debe solicitar al cliente para procesar su solicitud de crédito hipotecario aprobar una solicitud de crédito hipotecario para que sea evaluada por el comité de crédito que el sistema requiera de mi aprobación para todo crédito a otorgar cuyo monto exceda los USD 10 MM poder ejecutar tu producto en todas las versiones de windows desde windows 95 en adelante que el sitio web responda a toda transacción o funcionalidad de negocio en menos de 5 segundos que el sitio web esté disponible el 99,999% de las veces que intente accederlo”</p> <pre> [[4, 5, 6, 7, 8, 9, 10, 11, 13, 16, 18, 19, 20, 21, 22, 23, 24, 27, 29, 32, 34, 35, 36, 38, 39, 40, 41, 42, 44, 45, HISTORIA DE USUARIO: abrir una nueva cuenta de cheques a un cliente seleccionando el codigo de cliente y producto HISTORIA DE USUARIO: registrar una nueva solicitud de credito hipotecario a mi cliente HISTORIA DE USUARIO: indique cuales son los documentos que debo solicitar al cliente HISTORIA DE USUARIO: aprobar una solicitud de credito hipotecario HISTORIA DE USUARIO: requiera de mi aprobacion para todo credito a otorgar cuyo monto HISTORIA DE USUARIO: exceda los usd 10 mm HISTORIA DE USUARIO: ejecutar tu producto en todas las versiones de windows desde windows 95 en adelante HISTORIA DE USUARIO: utilice la base de datos de pedidos existente en lugar de crear una nueva HISTORIA DE USUARIO: responda a toda transaccion o funcionalidad de negocio en menos </pre>	
<p>Observaciones</p>	<p>Debido a que el prototipo acepta cualquier tipo de solicitud de usuario, existen una enorme variedad de variables que pueden influir la calidad del resultado final sea bueno, malo o regular.</p>

Prueba unitaria 9.

Código Prueba Unitaria	PU 009	Fecha de ejecución	27/01/2022
Historia de Usuario	HU 007		
Responsable:	Bryan Tarco, Carlos Alcasiga		
Descripción del caso de prueba	Comprobar la similitud semántica entre las historias de usuario.		
Prerrequisitos	<ul style="list-style-type: none"> - Ingresar al Sistema. - Product Backlog 		
Prueba Unitaria	Calcular la similitud semántica.		
Fragmento de Código	<pre> # Encontrar los grupos de los sustantivos relacionados resultado = 0 contador = 0 grupos = [] sustantivos = [token for token in doc if token.pos_ == "NOUN"] for index, token in enumerate(sustantivos): if token.pos_ == "NOUN": for i in sustantivos[index + 1:]: resultado = token.similarity(i) if (resultado > 0.75): grupos.append([token, i]) contador += 1 preview = [] for index, grupo in enumerate(grupos): preview = grupo for sus in (grupos[grupos.index(preview)] + grupos[grupos.index(preview) + 1:]): if not set(sus).isdisjoint(preview): valor = list(set(preview + sus)) if sus in grupos: grupos.remove(sus) grupos[grupos.index(preview)] = valor preview = valor </pre>		
Resultados Esperados	Asignar un ID de grupo a cada historia de usuario cuyos objetos tengan una similitud mayor a 0.85 con los objetos de un grupo de objetos en específico.		
Datos Ingresados: Lista de sustantivos Encontrados.			
<pre> [[los usuarios, los usuarios]] HISTORIA DE USUARIO: {'usuario': 'usuario', 'que': 'consultar a los usuarios', 'para_que': 'sin proposito', 'grupo': 1, 'nombre': 'H1'} HISTORIA DE USUARIO: {'usuario': 'usuario', 'que': 'registrar los usuarios', 'para_que': 'sin proposito', 'grupo': 1, 'nombre': 'H2'} HISTORIA DE USUARIO: {'usuario': 'usuario', 'que': 'crear una ruta de trabajo', 'para_que': 'sin proposito', 'grupo': 0, 'nombre': 'H3'} </pre>			
Observaciones	Se agrupan las entidades o sustantivos con una similitud semántica mayor a 0.75, en base a los grupos se determina cuáles historias de usuario afectan a un mismo objeto.		

Prueba unitaria 10.

Código Prueba Unitaria	PU 010	Fecha de ejecución	27/01/2022
Historia de Usuario	HU 007		
Responsable:	Bryan Tarco, Carlos Alcasiga		
Descripción del caso de prueba	Comprobar la similitud semántica entre las historias de usuario.		
Prerrequisitos	<ul style="list-style-type: none"> - Ingresar al Sistema. - Product Backlog 		
Prueba Unitaria	Las historias de usuario se colocan dentro de cada uno de los microservicios.		
Fragmento de Código	<pre> if (n_rh > 1) { altura = ((rh * n_rh) - (rh / 2)) + padding + 15; } return altura; }).strength(0.5) .force("link", ds.forceLink() .links(datos[1]) .id(function(d) { return d.id; }).strength(0)) .on("tick", ticked); </pre>		
Resultados Esperados	Gráfico de los microservicios y su historia de usuario correspondiente.		
Datos Ingresados: Diccionario de Source y Target.			
<p>HU 1 coordenada X: 97.5</p> <p>HU 2 coordenada X: 272.5</p> <p>HU 3 coordenada X: 447.5</p> <p>HU 1 coordenada Y: 447.5</p> <p>HU 2 coordenada Y: 447.5</p> <p>HU 3 coordenada Y: 447.5</p>			
Observaciones	La relación de las historias de usuario se genera gracias a un diccionario que contiene las historias de usuario (Source) y su objetivo (Target).		

Anexo E: Estimación del proyecto a través de Puntos de Historia “SP”



UNIVERSIDAD TÉCNICA DE COTOPAXI

“Desarrollo de un prototipo inteligente para determinar problemas de dependencia en aplicaciones basadas en microservicios”

En la tabla de a continuación se da a conocer cada una de las tareas definidas a través de historias de usuario con su respectiva puntuación.

ID	TAREA	SP
HU001	Ingresar datos al prototipo de forma oral y por archivo de audio.	22
HU002	Crear un campo para ingresar el usuario responsable de la solicitud.	15
HU003	Utilizar spaCy para desarrollar un algoritmo capaz de obtener historias de usuario.	32
HU004	Resumir historias de usuario y mostrar.	28
HU005	Permitir al usuario ordenar de forma manual las historias de usuario.	20
HU006	Mejorar el algoritmo utilizando spaCy para obtener historias de usuario de una manera correcta	32
HU007	Generar un gráfico de los microservicios y las relaciones de dependencia entre los mismos.	40
PUNTOS DE HISTORIA TOTALES:		189

Para realizar las tareas descritas en la tabla anterior se procede a planificar tres sprints conformados por dos tareas y un sprint conformado por una tarea, por lo tanto, los puntos de historia por sprint quedarían de la siguiente manera.

SPRINT	N° DE PUNTOS DE HISTORIA
1	37
2	60
3	52
4	40

En el desarrollo del Prototipo Inteligente se considera realizar 5 puntos de historia al día, en ese sentido se requiere de 38 días para completar los 189 SP del proyecto, por lo cual el presupuesto del proyecto es el siguiente:

Total, de Puntos de Historia = 189 SP

Puntos de Historia Diarios= 5 SP

Días Requeridos = 38 días (Total de Puntos de Historia / Puntos de Historia Diarios)

Salario Día por Programador = \$60.00 (Considerando un salario mensual de \$1.200.00)

Costo Estimado del Proyecto = \$4.560,00 (Días Requeridos * Salario Día por Programador)

Una vez aplicado la estimación de costos a por medio de los puntos de historia se obtiene que el costo para el “Desarrollo de un prototipo inteligente para determinar problemas de dependencia en aplicaciones basadas en microservicios” es de \$4.560,00 (cuatro mil quinientos sesenta dólares).

Anexo F: Formulario de validación de expertos



UNIVERSIDAD TÉCNICA DE COTOPAXI

Formulario de Juicio de Expertos

1. Título del Proyecto

Desarrollo de un prototipo inteligente para determinar problemas de dependencia en aplicaciones basadas en microservicios.

2. Datos del Experto

Nombres y Apellidos:	
Grado Académico:	
Lugar de Trabajo:	
Cargo que Desempeña:	
Años de Experiencia:	
E-mail:	
Teléfono o Celular:	
Fecha de Ejecución:	

3. Validación correspondiente a Procesamiento de Lenguaje Natural, Problemas de Dependencia y Utilidad del Prototipo.

Ítem	Componente a evaluar	<p>Valore en una escala de 1 a 6 el grado de relevancia que otorgan los siguientes ítems, señale con una X su respuesta en la casilla correspondiente.</p> <p>(1 = Totalmente en desacuerdo; 2 = en desacuerdo; 3 = en desacuerdo más que en acuerdo; 4 = de acuerdo más que en desacuerdo; 5 = de acuerdo; 6 = Totalmente de acuerdo)</p>	Grado de acuerdo					
			1	2	3	4	5	6
1	Procesamiento de lenguaje natural	Considera usted que la transformación de forma oral a texto se realiza de manera correcta.						
2		Considera usted que la transformación por medio de archivo de audio a texto se realiza de manera correcta.						
3		Considera usted que las historias de usuario generadas por el prototipo tienen una estructura lógica y coherente.						
4	Problemas de dependencia	Considera usted que el prototipo permite determinar problemas de dependencia.						
5		Considera usted que el gráfico permite evidenciar los microservicios sus relaciones y dependencia de manera efectiva.						
6	Utilidad del prototipo	Considera usted que este prototipo va ayudar a los equipos de desarrollo a mejorar la arquitectura de las aplicaciones basadas en microservicios.						
7		Considera usted que el prototipo presentado se convierte en una herramienta de apoyo para los equipos de desarrollo.						

4. Observaciones y Recomendaciones

.....
.....

5. Firma

Nombre :.....

C.I.:

Anexo G: Perfil de los expertos que validaron la propuesta



UNIVERSIDAD TÉCNICA DE COTOPAXI

“Desarrollo de un prototipo inteligente para determinar problemas de dependencia en aplicaciones basadas en microservicios”

No.	Grado Académico	Cargo que Desempeña	Años de Experiencia
Experto 1	<ul style="list-style-type: none"> Magíster en Ingeniería de Software. Ingeniero en Informática y Sistemas Computacionales 	Docente Universitario	6 años
Experto 2	<ul style="list-style-type: none"> Ingeniero en Sistemas Computacionales 	Desarrollador Web	5 años
Experto 3	<ul style="list-style-type: none"> Ingeniero de Computación y Sistemas (orientado a las Tecnologías de Información). Programador y Desarrollador Web. 	Desarrollador web	7 años

Anexo H: Respuestas de Juicio de Expertos

Experto 1

Ítem	Componente a evaluar	<p style="text-align: center;">Valore en una escala de 1 a 6 el grado de relevancia que otorgan los siguientes ítems, señale con una X su respuesta en la casilla correspondiente.</p> <p style="text-align: center;">(1 = Totalmente en desacuerdo; 2 = en desacuerdo; 3 = en desacuerdo más que en acuerdo; 4 = de acuerdo más que en desacuerdo; 5 = de acuerdo; 6 = Totalmente de acuerdo)</p>	Grado de acuerdo					
			1	2	3	4	5	6
1	Procesamiento de lenguaje natural	Considera usted que la transformación de forma oral a texto se realiza de manera correcta.					X	
2		Considera usted que la transformación por medio de archivo de audio a texto se realiza de manera correcta.						X
3		Considera usted que las historias de usuario generadas por el prototipo tienen una estructura lógica y coherente.					X	
4	Problemas de dependencia	Considera usted que el prototipo permite determinar problemas de dependencia.					X	
5		Considera usted que el gráfico permite evidenciar los microservicios sus relaciones y dependencia de manera efectiva.					X	
6	Utilidad del prototipo	Considera usted que este prototipo va ayudar a los equipos de desarrollo a mejorar la arquitectura de las aplicaciones basadas en microservicios.						X
7		Considera usted que el prototipo presentado se convierte en una herramienta de apoyo para los equipos de desarrollo.						X

Observaciones y recomendaciones:

Afinar el reconocimiento por voz en tiempo real.

Experto 2

Ítem	Componente a evaluar	<p>Valore en una escala de 1 a 6 el grado de relevancia que otorgan los siguientes ítems, señale con una X su respuesta en la casilla correspondiente.</p> <p>(1 = Totalmente en desacuerdo; 2 = en desacuerdo; 3 = en desacuerdo más que en acuerdo; 4 = de acuerdo más que en desacuerdo; 5 = de acuerdo; 6 = Totalmente de acuerdo)</p>	Grado de acuerdo					
			1	2	3	4	5	6
1	Procesamiento de lenguaje natural	Considera usted que la transformación de forma oral a texto se realiza de manera correcta.						X
2		Considera usted que la transformación por medio de archivo de audio a texto se realiza de manera correcta.						X
3		Considera usted que las historias de usuario generadas por el prototipo tienen una estructura lógica y coherente.					X	
4	Problemas de dependencia	Considera usted que el prototipo permite determinar problemas de dependencia.						X
5		Considera usted que el gráfico permite evidenciar los microservicios sus relaciones y dependencia de manera efectiva.					X	
6	Utilidad del prototipo	Considera usted que este prototipo va ayudar a los equipos de desarrollo a mejorar la arquitectura de las aplicaciones basadas en microservicios.						X
7		Considera usted que el prototipo presentado se convierte en una herramienta de apoyo para los equipos de desarrollo.						X

Observaciones y recomendaciones:

Recomiendo realizar un plan de pruebas para garantizar un % mínimo de aciertos en cuanto a la conversión entre voz/texto y viceversa.

Experto 3

Ítem	Componente a evaluar	<p>Valore en una escala de 1 a 6 el grado de relevancia que otorgan los siguientes ítems, señale con una X su respuesta en la casilla correspondiente.</p> <p>(1 = Totalmente en desacuerdo; 2 = en desacuerdo; 3 = en desacuerdo más que en acuerdo; 4 = de acuerdo más que en desacuerdo; 5 = de acuerdo; 6 = Totalmente de acuerdo)</p>	Grado de acuerdo					
			1	2	3	4	5	6
1	Procesamiento de lenguaje natural	Considera usted que la transformación de forma oral a texto se realiza de manera correcta.					X	
2		Considera usted que la transformación por medio de archivo de audio a texto se realiza de manera correcta.					X	
3		Considera usted que las historias de usuario generadas por el prototipo tienen una estructura lógica y coherente.				X		
4	Problemas de dependencia	Considera usted que el prototipo permite determinar problemas de dependencia.					X	
5		Considera usted que el gráfico permite evidenciar los microservicios sus relaciones y dependencia de manera efectiva.					X	
6	Utilidad del prototipo	Considera usted que este prototipo va ayudar a los equipos de desarrollo a mejorar la arquitectura de las aplicaciones basadas en microservicios.					X	
7		Considera usted que el prototipo presentado se convierte en una herramienta de apoyo para los equipos de desarrollo.					X	

Observaciones y recomendaciones:

Deberían validar cual sería el tamaño máximo que puede soportar su audio para la transformación porque entre más pesado se puede demorar o dar un error de transformación, siempre es bueno poner restricciones. El resto lo veo bien.

Anexo I: Hoja de vida de Expertos.

Hoja de vida Experto1

HOJA DE VIDA

DATOS PERSONALES

NOMBRE: Luis René Quisaguano Collaguazo.
CÉDULA: 172189518-1
DIRECCIÓN: Calle Tiberio N° 78, Cantón Mejía, Provincia de Pichincha.
TELÉFONOS: 0998820095 – 022 309164
ESTADO CIVIL: Soltero.
E-MAIL: renequisaguano@gmail.com
lquisaguano@gmail.com

NACIMIENTO: Machachi, Febrero 07 de 1992.
EDAD: 30 Años.



FORMACIÓN ACADÉMICA

Cuarto Nivel: **Maestría en Sistemas de Información**
Universidad Técnica de Cotopaxi
Latacunga, 2020
Tercer Nivel: **Ingeniería en Informática y Sistemas Computacionales**
Universidad Técnica de Cotopaxi
Latacunga, 2016
Idiomas Extranjeros: **Inglés**
Certificación B1
Latacunga, 2019

SEMINARIOS Y CURSOS

FINDER DE TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

Fundación Carlos Slim – 40 horas
Febrero, 2021

II ENCUENTRO DE INVESTIGACIÓN EN LA EDUCACIÓN SUPERIOR

Instituto Superior Tecnológico Pelileo – 20 horas
Diciembre, 2020

USO DE HERRAMIENTAS DIGITALES PARA FACILITAR LA EDUCACIÓN EN LÍNEA

Alcaldía de Mejía – 48 horas
Octubre, 2020

HUBSPOT SALES SOFTWARE CERTIFIED

HubSpot Academy – 40 horas
Septiembre, 2019

INBOUND CERTIFIED

HubSpot Academy – 40 horas
Septiembre, 2019

CURSO DE PROGRAMACIÓN DE DISPOSITIVOS MÓVILES

Platzi – 40 horas

Julio, 2019

CURSO DE PLATAFORMAS DE INTEGRACIÓN, DESPLIEGUE CONTINUOS E INFRAESTRUCTURA COMO CÓDIGO

BDOT Automation Tools – 8 horas

Mayo, 2018

PUBLICACIONES, PONENCIAS Y PARTICIPACIONES

VI JORNADAS INFORMÁTICAS – APLICACIONES WEB PROGRESIVAS VS APLICACIONES WEB TRADICIONALES

BTD Capacitaciones

La Maná, Julio 2021 - PONENCIA

ADOPCIÓN DE ESTRATEGIAS DE MARKETING DIGITAL EN LAS MICROEMPRESAS DEL CANTÓN MEJÍA

Revista Académica y Científica VICTEC

ISSN: 2737 -6214, Junio 2021 – ARTÍCULO

DESARROLLO DE APLICACIONES WEB PROGRESIVAS MEDIANTE LA IMPLEMENTACIÓN DE SERVICES WORKERS

Instituto Superior Tecnológico Vicente León

Latacunga, Marzo 2021 – PONENCIA

V SEMINARIO INTERNACIONAL UNIVERSIDAD – SOCIEDAD “TEJIENDO REDES PARA EL DESARROLLO SOCIAL”

Universidad Técnica de Cotopaxi

Latacunga, Febrero 2020 – PONENCIA

DESARROLLO DE APLICACIONES WEB CON ARQUITECTURA MVC MEDIANTE YII FRAMEWORK

Dirección de la Industria Aeronáutica del Ecuador

Latacunga, Diciembre 2019 - FACILITADOR

IV JORNADAS INFORMÁTICAS – ANALITICA DE DATOS CON QLIK SENSE CLOUD

Universidad Técnica de Cotopaxi

La Maná, Diciembre 2019 - PONENCIA

FORTALECIMIENTO DEL PERFIL DE LOS PROFESIONALES EN SISTEMAS INFORMATICOS PARA LA CREACIÓN DE UNA CULTURA DATA DRIVEN

Universidad Técnica de Cotopaxi

Latacunga, Octubre 2019 – PONENCIA

SISTEMA INFORMÁTICO PARA LA GESTIÓN DE INFORMACIÓN INSTITUCIONAL DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI

Revista de Ciencias de la Ingeniería y Aplicadas

ISSN: 2602-8255, Octubre 2019 – ARTÍCULO

IV SEMINARIO INTERNACIONAL UNIVERSIDAD – SOCIEDAD “TEJIENDO REDES PARA EL DESARROLLO SOCIAL”

Universidad Técnica de Cotopaxi

Latacunga, Enero 2019 – PONENCIA

DESARROLLO DE APLICACIONES WEB CON PLAY FRAMEWORK Y POSTGRESQL

Universidad Regional Autónoma de los Andes

Ambato, Septiembre 2018 - FACILITADOR

RECONOCIMIENTOS

BECA INSTITUCIONAL POR EXCELENCIA ACADÉMICA

Universidad Técnica de Cotopaxi
Programa de Maestría en Sistemas de Información, Mayo 2019

III CONCURSO NACIONAL DE DESARROLLO DE SOFTWARE, IBEE 2015

Escuela Superior Politécnica de Chimborazo.
Primer Lugar, Julio 2015.

BECA AL MERITO ACADÉMICO

Universidad Técnica de Cotopaxi, Carrera de Ingeniería en Informática y Sistemas Computacionales
Periodos Académicos: Marzo – Julio 2012, Septiembre – Febrero 2013, Marzo – Agosto 2013, Abril – Agosto 2014, Septiembre – Febrero 2015

MEJOR EGRESADO

Instituto Tecnológico Superior "Aloasi"
Machachi, 2009

EXPERIENCIA PROFESIONAL

PROFESOR A TIEMPO COMPLETO

Universidad Técnica de Cotopaxi. Desde: Abril 2021 – Hasta: Actualidad

ANALISTA DE TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

GAD Municipal del Cantón Mejía. Desde: Marzo 2020 – Hasta: Marzo 2021

PROFESOR A TIEMPO PARCIAL

Universidad Técnica de Cotopaxi. Desde: Octubre 2018 – Hasta: Febrero 2019

ANALISTA Y CONSULTOR DE TI

Evolution Soft. Desde: Enero 2018 – Hasta: Febrero 2020

DESARROLLADOR DE SOFTWARE

Rebian Software. Desde: Marzo 2016 – Hasta: Diciembre 2017

REFERENCIAS PERSONALES

MG. RAÚL JIMÉNEZ ROMÁN

Asesor de Alcaldía, GAD Municipal del Cantón Mejía
Cel: 0998484870

ING. GLADYS ESQUIVEL

Gerente de Evolution Soft
Cel: 0979259973

Ing. Luis René Quisaguano Mg.
C.I.: 172189518-1

Hoja de vida experto 2

Contactar

www.linkedin.com/in/uskokrum2010 (LinkedIn)

Aptitudes principales

Informática
Programación
Computación

Languages

Español
Inglés

Certifications

Scrum Foundation Professional Certificate

Oscar Alejandro F. García Fuentes

Ingeniero de Software en Nagnoi, LLC
Perú

Extracto

Soy Ingeniero de Computación y Sistemas (orientado a las Tecnologías de Información), Programador y Desarrollador Web, con más de 7 años de experiencia en diversos lenguajes de programación, administración de bases de datos y aplicación de tecnologías informáticas en diversos campos.

Me encanta conocer todo tipo de tecnologías nuevas y poder afianzar estos conocimientos para poder transmitirlos más adelante a personas interesadas en iniciarse en el mundo de la programación y todo lo referente a las tecnologías de información.

Experiencia

Nagnoi, LLC
Ingeniero de software
octubre de 2021 - Present (6 meses)
San Juan, Puerto Rico

Desarrollo de software utilizando el lenguaje de programación C# y .NET Core.

Udemy
Instructor
mayo de 2017 - Present (4 años 11 meses)
San Francisco, California, Estados Unidos

Soy instructor y desarrollo cursos para Udemy.

Me encanta conocer todo tipo de tecnologías nuevas y poder afianzar estos conocimientos para poder transmitirlos más adelante a personas interesadas en iniciarse en el mundo de la programación y todo lo referente a las tecnologías de información.

García Solutions

Representante
enero de 2014 - Present (8 años 3 meses)
Provincia de Chiclayo, Peru

Empresa dedicada a reparar equipos de cómputo, clases de computación y programación, consultoría y soluciones informáticas.

Creador de Videos, Reparador de Computadoras
2010 - Present (12 años)
Gerente General (Administrador).

YouTube
Partner
octubre de 2010 - Present (11 años 6 meses)
San Bruno, California, Estados Unidos

Me dedico a la creación, edición y publicación de videos en YouTube, de diversas categorías como computación, tutoriales, programación, entre otros.

www.youtube.com/UskoKruM2010

Universidad de San Martín de Porres
Analista programador
marzo de 2015 - octubre de 2021 (6 años 8 meses)
Urb. La Pradera, Pimentel, Perú

Me dedico a diseñar e implementar soluciones informáticas en el Departamento de Sistemas de la Universidad San Martín de Porres - Filial Norte.

Autocentro Ferreñafe S.A.
Auxiliar Contable
agosto de 2011 - abril de 2012 (9 meses)

Educación

Universidad San Martín de Porres
Ingeniero de Computación & Sistemas, Ingeniería de software · (2010 - 2016)

Universidad Peruana de Ciencias Aplicadas
Ing. de Software · (2009 - 2009)

Colegio San Agustín

Hoja de vida experto 3

Nombres : William Jair Apellidos : Dávila Vargas Fecha de nacimiento: 19 de Octubre de 1994 Domicilio: Cdla. Dager, Calle Rio Zamora entre Av.Tumbes y Av. Tarqui Teléfono de domicilio: 2977557 E-mail 1: davilawilliam94@gmail.com	C.I.: 0928363993 Tipo de sangre : O+ Edad: 25 años Ciudad: Milagro, Guayas Celular: 0979014551 E-mail 2: williamjair94@hotmail.com	
---	---	---

FORMACIÓN ACADÉMICA

2017 <i>Milagro</i>	Ingeniero en Sistemas Computacionales Superior: Universidad de Estatal de Milagro (UNEMI)
2012 <i>Milagro</i>	Aplicaciones Informáticas Secundaria: Colegio Particular Mixto "Madre de Dios"
2006 <i>Milagro</i>	Educación Básica Primaria: Escuela Particular "Victoria Concha de Valdez"

PERFIL PROFESIONAL

Especializado en dar soluciones informáticas a medida en múltiples plataformas de programación, responsable, dinámico, innovador, siempre con ideas nuevas, y con la capacidad de adaptarme a cualquier proyecto así sea en condiciones de alta presión.

CONOCIMIENTO ADQUIRIDO

N°	Descripción	Programas
1	Base de datos	▪ SQLServer, PostgreSQL, MySQL, SQLITE y Access
2	Lenguajes de programación de escritorio	▪ Visual.Net, C#, Java y Python
3	Aplicaciones Móviles	▪ Android
4	Lenguaje de programación para la web	▪ PHP, Django y ASP.NET
5	Sistemas operativos, programas utilitarios, entre otros	▪ Windows 7,8.1,10, Linux Ubuntu, Microsoft Word, Excel, PowerPoint, Publisher, Project, Visio, Prezi, Lucidchart
6	Conocimiento adicional para el desarrollo de aplicaciones	▪ POO, XHTML, HTML5, CSS, Javascript, JQuery, Ajax, XML, JSON, Bootstrap y GIT.

EXPERIENCIA LABORAL

N°	Empresa	Área	Posición	De mes-año	A mes-año
1	UNEMI	Departamento de Ensamblaje y reparación de computadoras	Técnico	Abril, 2015	Junio, 2015
2	UNEMI	Facultad Ciencias de la Ingeniería	Ayudante de Catedra	Febrero, 2013	Octubre, 2016
3	Cavnet S.A.	Helpdesk	Analista	Abril, 2016	Junio, 2016
4	Pc & Servicios	Desarrollo	Programador	Enero, 2016	Octubre, 2016
5	IN.PLANET S.A.	Departamento de Proyectos Informáticos	Desarrollador Web	Abril, 2018	Mayo, 2019

6	IN.PLANET S.A.	Departamento de Proyectos Informáticos	Ingeniero de Proyectos	Junio, 2019	Actualidad
---	----------------	--	------------------------	-------------	------------

CURSOS, JORNADAS, TALLERES Y SEMINARIOS

N°	Nombre	Institución	Horas	Año
1	Curso de operador, programador y diseñador gráfico	ICD Computer Center	140	2012
2	V Jornadas Tecnológicas e Industriales	UNEMI	20	2012
3	Jornadas Tecnológicas Calidad en la Web	UNEMI	5	2014
4	Curso Online "Analítica Web"	Gobierno de España- Ministerio de industria, energía y turismo-Escuela de organización industrial	40	2016
5	Curso Online "Productividad Personal en la Era Digital"	Fundación Santa María la Real del Patrimonio Histórico – España	40	2016
6	Curso de Desarrollo de Aplicaciones Web usando Tecnología Open Source	Academia Cisco Espol	50	2018
7	Curso Online "Internet of Everything"	Cisco Networking Academy	30	2018
8	Security Day 2018	Escuela Superior Politécnica del Litoral	8	2018
9	Aplicaciones Web Responsivas con ASP.NET, WCF y Entity Framework 6	SIPECOM	24	2019

MERITOS Y GALARDONES

N°	Nombre	Institución	Año
1	Mención honorífica por haber sido el segundo escolta del pabellón de Milagro	Escuela Particular "Victoria Concha de Valdez"	2005 - 2006
2	Medalla de reconocimiento por haber sido el segundo escolta del pabellón de Milagro	Escuela Particular "Victoria Concha de Valdez"	2005 - 2006
3	Reconocimiento por excelencia académica de buen desempeño como estudiante y abanderado del pabellón nacional	Club de Leones de Milagro y su Comité de Damas	2011
4	Mención honorífica por haber sido el abanderado del pabellón nacional del Ecuador	Colegio Particular Mixto "Madre de Dios"	2012
5	Placa de reconocimiento por la elaboración del mejor proyecto de informática	Colegio Particular Mixto "Madre de Dios"	2012
6	Medalla de oro del mejor bachiller	Colegio Particular Mixto "Madre de Dios"	2012
7	Reconocimiento por haber realizado y concluido con éxito las ayudantías de cátedra de "ESTRUCTURA DE DATOS" Y "REALIDAD NACIONAL"	Universidad Estatal de Milagro	2014
8	Beca de Alto promedio y distinción académica	Universidad Estatal de Milagro	2013 - 2016
9	Participación en el concurso de Galardones Nacionales 2016 con el tema "Diseño e implementación de una aplicación"	Secretaría de Educación Superior, Ciencias, Tecnología	2016

tecnológica que permita detectar el maltrato y acoso en niños, niñas y adolescentes en las instituciones educativas"	(SENESCYT)-Investigaciones Científicas
--	--

HABILIDADES PROFESIONALES

- Manejo de diferentes herramientas de programación y administración de bases de datos.
- Experiencia en el desarrollo de sistemas informáticos tanto web y escritorio.
- Facilidad y eficacia en la resolución de problemas así sea bajo presión.
- Creatividad e innovación en cada trabajo realizado.
- Colaboración, argumentación y toma de decisiones.
- Capacidad de adaptación al grupo de trabajo.
- Comunicación eficaz y liderazgo.

LOGROS PERSONALES

- Creador y fundador de AlgoriSoft, sitio web dedicado a la comercialización de fragmentos de sistemas informáticos que he ido desarrollando a lo largo de mi carrera profesional como programador para aquellas personas que quieran incursionar en el mundo de la programación y quieran tener una base ideal para realizar cualquier tipo de proyecto tecnológico a través de un gestor de contenido que desarrolle para la subida dinámica de la información. Enlace del sitio web: <https://algorisoft.com/>
- Instructor en la plataforma UDEMY. Curso denominado "Implementa tu sitio web Django en un VPS Ubuntu Español 2019". Enlace del curso: <https://www.udemy.com/implementa-tu-sitio-web-django-en-un-vps-ubuntu-espanol-2019/>
- Creador del canal de YouTube AlgoriSoft para dictar cursos de programación, Curso de Python con Django de 0 a Máster | Español. Enlace del canal <https://www.youtube.com/c/AlgoriSoft>

REFERENCIAS PERSONALES

MSC. William Bazán Vera Universidad Agraria del Ecuador Docente de la Facultad de Ciencias de la Informática william_bazan@hotmail.com 0996852974	MSc. Oscar Berneo Almeida Universidad Estatal de Milagro Docente de la Facultad de Ciencias de la Ingeniería oscar_majic@hotmail.com 0960458705
Ing. Marcos Naranjo Pino Papelera Nacional S.A. Auditor de tecnologías de la información marcos_naranjo_pino@hotmail.com 0991324716	MgTI. Rufina Narcisca Bravo Universidad Estatal de Milagro Docente de la Facultad de Ciencias de la Educación rbravo@unemi.edu.ec 0996460826

REFERENCIAS LABORALES

Ing. Jordy Malo IN.PLANET S.A. Desarrollador de aplicaciones móviles jmalo@in-planet.net 0982467563	Ing. Freddy Valenzuela Ortega Cavnet S.A. Gerente Técnico del departamento Helpdesk fvalenzu@hotmail.com 0994657465
--	--