

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS.

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES



INTEGRACIÓN DE LAS APIs REST DE ZENODO Y GITHUB MEDIANTE UNA APLICACIÓN ORIENTADA A SERVICIOS PARA PUBLICAR CONTENIDO OPEN SCIENCE

Trabajo de grado presentado en la Universidad Técnica del Norte previo a la obtención
del título de Ingeniero en Sistemas Computacionales

Autor:

Leonardo Michael Avila Armas

Director:

PhD. Cathy Pamela Guevara Vega

Ibarra, 2023



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

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

1. IDENTIFICACIÓN DE LA OBRA

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

| DATOS DE CONTACTO | | | |
|-----------------------------|--|------------------------|------------|
| CÉDULA DE IDENTIDAD: | 1003602099 | | |
| APELLIDOS Y NOMBRES: | AVILA ARMAS LEONARDO MICHAEL | | |
| DIRECCIÓN: | IBARRA, CARLOS ELÍAS ALMEIDA 8-128 | | |
| EMAIL: | lmavilaa@utn.edu.ec | | |
| TELÉFONO FIJO: | 954338 | TELÉFONO MÓVIL: | 0998912533 |

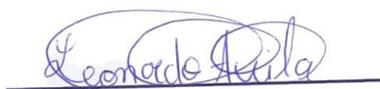
| DATOS DE LA OBRA | |
|--------------------------------|--|
| TÍTULO: | INTEGRACIÓN DE LAS APIS REST DE ZENODO Y GITHUB MEDIANTE UNA APLICACIÓN ORIENTADA A SERVICIOS PARA PUBLICAR CONTENIDO EN OPEN SCIENCE. |
| AUTOR: | AVILA ARMAS LEONARDO MICHAEL |
| FECHA: | 08/09/2023 |
| PROGRAMA: | <input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO |
| TITULO POR EL QUE OPTA: | INGENIERO EN SISTEMAS COMPUTACIONALES |
| DIRECTOR: | PHD. CATHY GUEVARA, MSC. |
| ASESOR 1: | PHD. IRVING REASCOS, MSC. |
| ASESOR 2: | ING. FAUSTO SALAZAR, MSC. |

2. CONSTANCIA

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

Ibarra, a los 08 días del mes de septiembre de 2023

EL AUTOR:

A handwritten signature in blue ink, appearing to read "Leonardo Avila", is written over a horizontal line.

ESTUDIANTE

Avila Armas Leonardo Michael

C.I: 100360209-9

CERTIFICACIÓN DIRECTOR

Ibarra 08 de septiembre del 2023

CERTIFICACIÓN DIRECTOR DEL TRABAJO DE TITULACIÓN

Por medio del presente yo PhD. Cathy Guevara, MSC, certifico que el Sr. Leonardo Michael Avila Armas portador de la cédula de ciudadanía número 103602099, ha trabajado en el desarrollo del proyecto de grado **“INTEGRACIÓN DE LAS APIs REST DE ZENODO Y GITHUB MEDIANTE UNA APLICACIÓN ORIENTADA A SERVICIOS PARA PUBLICAR CONTENIDO EN OPEN SCIENCE”**, previo a la obtención del Título de Ingeniero en Sistemas Computacionales realizado con interés profesional y responsabilidad que certifico con honor de verdad.

Es todo en cuanto puedo certificar en honor a la verdad.

Atentamente



PhD. Cathy Guevara, MSc.

DIRECTORA DE TESIS

DEDICATORIA

Queridos seres queridos,

Hoy, en el culmen de este viaje académico, deseo expresar mi más profundo agradecimiento a quienes han sido mi fuente constante de apoyo, inspiración y amor a lo largo de esta travesía.

A mis padres, Marco Ávila y Carla Armas, les debo la base de mi educación, el amor inquebrantable y la confianza en mí mismo. Su sacrificio y apoyo incondicional han sido el motor de mi perseverancia.

A mi querida hermana Joselyn Ávila, quien ha sido mi ejemplo de valores y dedicación, tu influencia ha iluminado mi camino y me has mostrado lo que significa el esfuerzo y la determinación.

A mi tío Alex Armas, cuyo amor y preocupación han sido tan profundos como los de un segundo padre, agradezco su constante aliento y guía a lo largo de esta travesía.

A mi abuelita María Boada, cuyo recuerdo y amor siguen vivos en mi corazón, sé que desde el cielo continúas siendo mi apoyo incondicional y mi inspiración.

A mi leal compañero de todas las noches de estudio, mi mascota Luke, tus miradas de ánimo fueron mi faro en las horas más oscuras de este proceso.

A cada uno de ustedes, los llevo en mi corazón y agradezco por formar parte fundamental de este logro. Este trabajo de tesis no solo representa mi esfuerzo, sino también el cariño y apoyo que he recibido de ustedes a lo largo de mi vida.

Con gratitud eterna,

Leonardo Ávila

AGRADECIMIENTO

Hoy, al finalizar este capítulo de mi vida académica, quiero expresar mi profundo agradecimiento a todos aquellos que han dejado una huella en mi formación y crecimiento.

A los docentes de la carrera de software, les agradezco más allá de las palabras. Su sabiduría y apoyo inquebrantable han sido la brújula que me ha guiado a lo largo de este camino educativo. Cada lección compartida, cada consejo brindado, han sido piedras angulares en mi desarrollo.

A mi estimada tutora de trabajo de titulación, PhD. Cathy Guevara, MSc., así como a los Ingenieros Fausto Salazar y el PhD. Irvin Reascos, MSc., les debo un agradecimiento profundo. Sus conocimientos técnicos y experiencia fueron los cimientos sobre los cuales edificué mi carrera universitaria. Su enseñanza me ha llevado a este punto culminante.

PhD. Cathy Guevara MSc., quiero destacar su apoyo inquebrantable durante la fase final de mi trabajo de grado. Sin su orientación y paciencia, este logro no habría sido posible.

A mis amigos, verdaderos tesoros de esta travesía, les agradezco por los momentos de risas que iluminaron incluso los días más oscuros. En particular, quiero expresar mi gratitud a Michelle Davas, mi compañera de equipo desde la nivelación hasta este punto final. Tus consejos oportunos y apoyo constante han sido invaluable.

A todos ustedes, les llevo conmigo en esta nueva etapa de mi vida. Sus contribuciones y amistad han enriquecido mi experiencia universitaria de manera inimaginable. Hoy, miro hacia el futuro con gratitud y emoción, sabiendo que sus enseñanzas y apoyo serán faros en los desafíos que enfrentaré.

ÍNDICE DE CONTENIDO

| | |
|---|------|
| DEDICATORIA | V |
| AGRADECIMIENTO | VI |
| ÍNDICE DE CONTENIDO | VII |
| ÍNDICE DE FIGURAS..... | XI |
| ÍNDICE DE TABLAS | XIII |
| RESUMEN..... | XIV |
| ABSTRACT..... | XV |
| INTRODUCCIÓN..... | 1 |
| Antecedentes | 1 |
| Prospectiva | 2 |
| Objetivos | 5 |
| Objetivo General | 5 |
| Alcance | 5 |
| Metodología | 7 |
| Justificación | 8 |
| CAPÍTULO I..... | 9 |
| 1. MARCO TEÓRICO..... | 9 |
| 1.1 Estado actual de Open Science, Zenodo y GitHub. | 9 |
| 1.1.1 Definición de la cadena de búsqueda..... | 11 |
| 1.1.2 Preguntas de investigación..... | 13 |
| 1.1.3 Búsqueda de documentos..... | 14 |
| 1.1.4 Criterios de Inclusión y Exclusión | 15 |
| 1.1.5 Selección de artículos..... | 16 |
| 1.1.6 Análisis de Resultados | 18 |
| 1.2 Open Science..... | 20 |

| | |
|---|----|
| 1.2.1 Concepto de Open Science..... | 20 |
| 1.2.2. Alcance de Open Science en un contexto empresarial. | 23 |
| 1.2.6 Open Science en las empresas ecuatorianas | 26 |
| 1.3 GitHub | 27 |
| 1.3.1 Concepto, funcionalidad y origen. | 27 |
| 1.3.2 Ventajas y Desventajas de GitHub. | 29 |
| 1.3.3 Integración de GitHub con Zenodo..... | 30 |
| 1.4 Zenodo | 32 |
| 1.4.1 Concepto, Origen y Funcionalidad..... | 32 |
| 1.5. ISO/IEC 25022:2016 | 33 |
| 1.6. Indicadores por desarrollar | 35 |
| CAPÍTULO 2..... | 38 |
| 2.1. Análisis, diseño y desarrollo de la arquitectura..... | 38 |
| 2.1.1. Arquitecturas de software. | 38 |
| 2.1.2. Estilos de desarrollo de Software | 42 |
| 2.1.3. SCRUM | 43 |
| 2.1.4. Herramientas de trabajo. | 44 |
| 2.1.5. Configuración de las plataformas Zenodo y GitHub. | 47 |
| 2.1.5.1. Configuración del repositorio Zenodo..... | 48 |
| 2.1.5.1.1 Configuraciones para desarrollador en Zenodo..... | 48 |
| 2.1.5.1.2. Configuraciones para desarrollador en GitHub..... | 50 |
| 2.2. Desarrollo | 51 |
| 2.2.1 Sprint 0 | 51 |
| 2.2.1.1 Historias Épicas..... | 52 |
| 2.2.1.3. Producto Backlog..... | 53 |

| | |
|---|----|
| 2.1.2. Sprint 1 | 58 |
| 2.2.2.1. Sprint Planning | 58 |
| 2.2.2.3. Ejecución del Sprint. | 59 |
| Aplicación orientada a servicios, módulos..... | 59 |
| Modelo de bases de datos. | 60 |
| Aplicación Orientada a servicios, mapa de procesos | 60 |
| Interfaces de usuario – Balsamiq | 61 |
| 2.2.2.4. Sprint Review | 65 |
| 2.2.2.5. Sprint Retrospective | 67 |
| 2.1.3. Sprint 2 | 68 |
| 2.2.3.1 Sprint Planning | 68 |
| 2.2.3.2. Sprint Backlog..... | 68 |
| 2.2.3.3. Ejecución del Sprint | 68 |
| 2.2.3.4. Sprint Review..... | 74 |
| 2.2.3.5. Sprint Retrospective. | 76 |
| 2.2.4. Sprint 3 | 76 |
| 2.2.4.1. Sprint Planning | 76 |
| 2.2.4.2. Sprint Review..... | 77 |
| 2.2.4.3. Ejecución del Sprint | 77 |
| 2.2.4.4. Sprint Review..... | 81 |
| 2.2.4.5. Sprint Retrospective | 84 |
| 2.3. Integración de las APIs Rest de Zenodo y GitHub. | 85 |
| 2.3.1. <i>Definición de Requisitos de Integración</i> | 90 |
| 2.3.2. Selección de Herramientas y Tecnologías. | 91 |
| 2.3.3. Implementación de las Funcionalidades de Integración. | 92 |

| | | |
|-----------------------|---|-----------|
| 2.3.4. | Resultados y Logros de la Integración. | 94 |
| CAPÍTULO 3..... | | 96 |
| 3. | Resultados y Discusión | 96 |
| 3.1. | Descripción de la evaluación en base a la norma ISO/IEC 25022. | 96 |
| 3.1.1.1. | Funcionalidad | 96 |
| 3.1.1.2. | Usabilidad..... | 97 |
| 3.1.2. | Metodología de la Evaluación..... | 97 |
| 3.1.2.1. | Pruebas Funcionales | 97 |
| 3.1.2.2. | Pruebas de Rendimiento | 98 |
| 3.1.2.3. | Encuestas de Usabilidad | 98 |
| 3.1.3. | Recopilación de Datos..... | 98 |
| 3.1.4. | Análisis de Resultados | 98 |
| 3.2. | Evaluación de Adequacy (Adecuación)..... | 99 |
| 3.2.1. | Identificación de Funciones y Objetivos..... | 99 |
| 3.2.2. | Identificación de Criterios | 100 |
| 3.3. | Comprensibilidad del sistema: Evaluación y Métricas..... | 104 |
| 3.3.1. | Coherencia | 105 |
| 3.3.2. | Comprensión | 106 |
| CONCLUSIONES | | 107 |
| RECOMENDACIONES | | 109 |
| ANEXOS..... | | 111 |
| REFERENCIAS | | 116 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1: Diagrama de Planteamiento del Problema-Efecto y Causas | 4 |
| Figura 2: Arquitectura a desarrollar | 7 |
| Figura 3: Imagen de la interfaz gráfica de Zenodo | 10 |
| Figura 4: Diagrama del protocolo de investigación | 15 |
| Figura 5: Fases de la investigación | 22 |
| Figura 6: Elementos constitutivos de la Open Science (Miembros de la Comisión Europea)..... | 22 |
| Figura 7: Estructura básica de trabajo de GitHub | 28 |
| Figura 8: Registro de Zenodo con indicadores Altmetrics y DOI | 33 |
| Figura 9: Métricas ISO/IEC 25022:2016 | 34 |
| Figura 10: Personal Access Token Zenodo | 49 |
| Figura 11: Configuración de Personal Access Token | 49 |
| Figura 13: Configuración OAuth Apps en GitHub. | 50 |
| Figura 12:Consumo API Rest de Zenodo y GitHub mediante Postman | 51 |
| Figura 14: Azure Devops - Sprint 0..... | 58 |
| Figura 15: Componentes del proyecto..... | 59 |
| Figura 16: Diagrama de Base de Datos, versión 1 | 60 |
| Figura 17: Mapa de Procesos Bizagi Modeler. | 61 |
| Figura 18: Diseño Web UI, Login..... | 62 |
| Figura 19: Diseño Web UI, Zona información. | 62 |
| Figura 20: Capa del Cliente | 64 |
| Figura 21: Despliegue en producción aplicativo SOA, versión inicial. | 65 |
| Figura 22: Sprint 1-Azure Devops..... | 66 |
| Figura 23:Mapa de Procesos Global del Aplicativo | 70 |
| Figura 24: Módulos Sistema | 70 |

| | |
|--|----|
| Figura 25: Menú Principal – Usuarios | 71 |
| Figura 26: Menú Principal-Categorías | 72 |
| Figura 27: Menú Principal – Productos..... | 72 |
| Figura 28: Menú Principal-Usuarios..... | 73 |
| Figura 29: Menú Principal-Ventas..... | 74 |
| Figura 30: Flujo de trabajo-Administrador..... | 78 |
| Figura 31:Reportes de ventas – BI | 79 |
| Figura 32: Formato .xls de Ventas..... | 79 |
| Figura 33: Exportación a Zenodo..... | 80 |
| Figura 34: Contenido Open Science | 81 |
| Figura 35: Azure Devops Board - Sprint 3 | 83 |
| Figura 36: Tokens de acceso Zenodo & GitHub | 85 |
| Figura 37: Integración de repositorios GitHub & Zenodo | 87 |
| Figura 38: Publicación de repositorios por medio de Releases. | 88 |
| Figura 39: Integración de componentes Github & Zenodo..... | 88 |
| Figura 40: Lógica de Integración. | 93 |

ÍNDICE DE TABLAS

| | |
|---|-----|
| Tabla 1: Palabras Clave..... | 12 |
| Tabla 2: Pregunta de investigación (PI)..... | 14 |
| Tabla 3: Artículos seleccionados para la revisión literaria | 18 |
| Tabla 4: Matriz de conceptos..... | 20 |
| Tabla 5: Indicadores de Open Science en la gestión de inventarios | 26 |
| Tabla 6: Ventajas y Desventajas de GitHub en el manejo de sistemas de inventario. ... | 30 |
| Tabla 7: Indicadores cualitativos y cuantitativos | 35 |
| Tabla 8: Herramientas de trabajo | 47 |
| Tabla 9: Roles de Scrum | 52 |
| Tabla 10: Historias Épicas | 53 |
| Tabla 11: Product Backlog..... | 57 |
| Tabla 12: Sprint Review 1 | 66 |
| Tabla 13: Sprint Review 2..... | 75 |
| Tabla 14: Sprint Review 3..... | 83 |
| Tabla 15: Requisitos de Integración | 91 |
| Tabla 16: Funciones y Objetivos (Adequacy) | 100 |
| Tabla 17: Criterios de Adecuación..... | 102 |
| Tabla 18: Resultados obtenidos-comprensión..... | 106 |

RESUMEN

El objetivo de la presente investigación fue desarrollar un método innovador que permita la publicación de contenido científico en formatos abiertos, promoviendo así el acceso abierto a la ciencia.

El objetivo principal de este trabajo de titulación fue crear un aplicativo de gestión de inventario que pudiera integrarse fácilmente entre Zenodo y GitHub, dos plataformas clave. La integración facilitó la transferencia rápida de proyectos y componentes alojados en GitHub a Zenodo, lo que los convirtió en recursos de Open Science accesibles para la comunidad científica.

La investigación examinó las características y los beneficios de Open Science, destacando su papel en promover la transparencia, la colaboración y la difusión del conocimiento científico. Se creó una aplicación orientada a servicios que permitió la integración de las API Rest de Zenodo y GitHub, que se identificaron como herramientas esenciales para lograr este objetivo.

El resultado fue un aplicativo útil que simplificó significativamente el proceso de publicación en Open Science y proporcionó a los investigadores una plataforma efectiva y accesible para compartir sus contribuciones científicas. Al promover la colaboración y el acceso abierto a la investigación y el desarrollo, esta tesis contribuye al avance de la ciencia abierta. Esto hace que el ecosistema científico sea más abierto e inclusivo.

Palabras Clave: Integración, Open Science, GitHub-Zenodo

ABSTRACT

The objective of the research is to develop an innovative method that allows the publication of scientific content in open formats, thus promoting open access to science.

The main objective of this thesis was to create an inventory management application that could be easily integrated between Zenodo and GitHub, two key platforms. The integration facilitated the rapid transfer of projects and components hosted on GitHub to Zenodo, turning them into open science resources accessible to the scientific community.

The research examined the characteristics and benefits of Open Science, highlighting its role in promoting transparency, collaboration, and the dissemination of scientific knowledge. A service-oriented application was created that allowed the integration of the Rest APIs of Zenodo and GitHub, which were identified as essential tools to achieve this goal.

The result was a useful application that significantly simplified the publication process in Open Science and provided researchers with an effective and accessible platform to share their scientific contributions. By promoting collaboration and open access to research, this thesis contributes to the advancement of open science. This makes the scientific ecosystem more open and inclusive.

Keywords: Integration, Open Science, GitHub-Zenodo

INTRODUCCIÓN

Antecedentes

Open Science es un movimiento cuyo objetivo es facilitar el acceso libre a publicaciones científicas y datos de investigación. Este movimiento implica el desarrollo de aspectos más amplios y complejos que contribuyan a la producción de un conocimiento científico abierto y colaborativo que responda a los desafíos de la sociedad actual. En pocas palabras, Open Science tiene como objetivo asegurar el acceso abierto tanto a los procesos como a los resultados de la actividad científica mediante la reutilización, distribución y reproducción de estudios. Esto incluye la libre disponibilidad de los instrumentos y herramientas de trabajo utilizados en la producción científica, desde la recopilación de información y el análisis de datos hasta la obtención de resultados, sin embargo, Open Science es un movimiento el cual no se ha dado mucho énfasis en el ámbito social, cultural y educativo, por lo que este término muchas veces pasa desapercibido dentro de un entorno global (Moran, 2021). El uso de herramientas como Zenodo, Figshare, Software heritage, o repositorios individuales, ha hecho que los resultados de las investigaciones sean mucho más accesibles a la hora de buscarlos, pero, dentro de la utilización de repositorios virtuales como GitHub, GitLab y servicios similares que están destinados para sistemas de control de versiones tienen un problema significativo porque no ofrecen datos correctamente archivados y conservados. En este sentido, en las políticas establecidas el ICSE 2021 (Políticas de ciencia abierta) manifiesta que las investigaciones en Ingeniería del Software deberían cumplir los principios que propone Open Science.

Situación Actual

Actualmente, el movimiento Open Science no ha sido estudiado a fondo y, sobre todo, no se le ha otorgado la importancia que necesita. En comparación con Europa, se puede observar que el movimiento Open Science ha comenzado a ser más conocido en la comunidad investigativa de Latinoamérica en los últimos años. Esto es diferente a Europa, donde el movimiento Open Science ha estado presente desde el principio (2011). Además, se ha descubierto que uno de los principales desafíos de trabajo interno institucional es la escasa aceptación de los repositorios académicos de investigación por parte de la comunidad académica. Los repositorios se perciben como nichos aislados en el sistema científico-investigador, incluso cuando hay mandatos de depósito. Es posible que se sigan considerando como una burocracia adicional en los procesos de difusión de la investigación, acreditación de publicaciones, métodos de trabajo y procesos de trabajo de los investigadores.

Prospectiva

La presente propuesta de investigación busca fomentar el movimiento Open Science en IS, por medio de la accesibilidad a los datos y código que se publican en los diferentes repositorios virtuales como Zenodo. Específicamente, se quiere lograr que los trabajos de IS sean públicos para que la comunidad científica los pueda utilizar. También, se busca fomentar la difusión de conocimiento y motivar al uso de repositorios virtuales Open Science.

Planteamiento del Problema

En el actual contexto global de avance científico y tecnológico, la implementación de prácticas Open Science ha sido reconocida como un enfoque transformador en el

ámbito de la investigación y difusión del conocimiento. Open Science, también conocida como ciencia abierta, es un conjunto de valores y prácticas que promueven la apertura, transparencia y accesibilidad de la información científica, lo que promueve la colaboración y el progreso colectivo en la comunidad científica. El interés de Ecuador por promover la ciencia abierta y su potencial para mejorar la creación y difusión de conocimientos en el sector empresarial está aumentando. Aún hay una gran brecha en la adopción de estas prácticas por parte de las empresas ecuatorianas, a pesar de las claras ventajas de Open Science. La falta de implementación generalizada de Open Science en las empresas ecuatorianas es el tema principal que motiva esta investigación. Aunque se reconoce cada vez más la importancia de compartir información científica y tecnológica de manera abierta, existen varias dificultades para implementar estas prácticas en el tejido empresarial del país.

1. **Una falta de conciencia y cultura de Open Science:** La falta de conciencia sobre los beneficios y oportunidades de Open Science es una de las principales dificultades que enfrentan las empresas ecuatorianas la adopten. La mentalidad empresarial aún no ha adoptado completamente la cultura de la ciencia abierta, lo que limita la disposición de las organizaciones a compartir y colaborar en el ámbito científico.
2. **Limitaciones Tecnológicas y de Infraestructura:** La implementación exitosa de Open Science requiere una infraestructura tecnológica adecuada que facilite la publicación y acceso abierto a datos científicos. La falta de recursos tecnológicos y la falta de plataformas y herramientas adecuadas pueden obstaculizar la adopción de estas prácticas por parte de las empresas.

3. **Resguardo de la Propiedad Intelectual:** Algunas empresas están preocupadas por la confidencialidad de los datos y la investigación, así como por la protección de la propiedad intelectual. La falta de claridad sobre cómo proteger sus intereses y propiedades intelectuales en un entorno abierto puede obstaculizar la adopción de prácticas de Open Science.
4. **Escasa Colaboración y Coordinación:** La adopción de Open Science requiere que las empresas y otras instituciones académicas o gubernamentales trabajen más juntas. El avance hacia la apertura y la compartición de conocimiento puede verse obstaculizado por la falta de una cultura de trabajo en equipo y la ausencia de mecanismos para establecer alianzas.

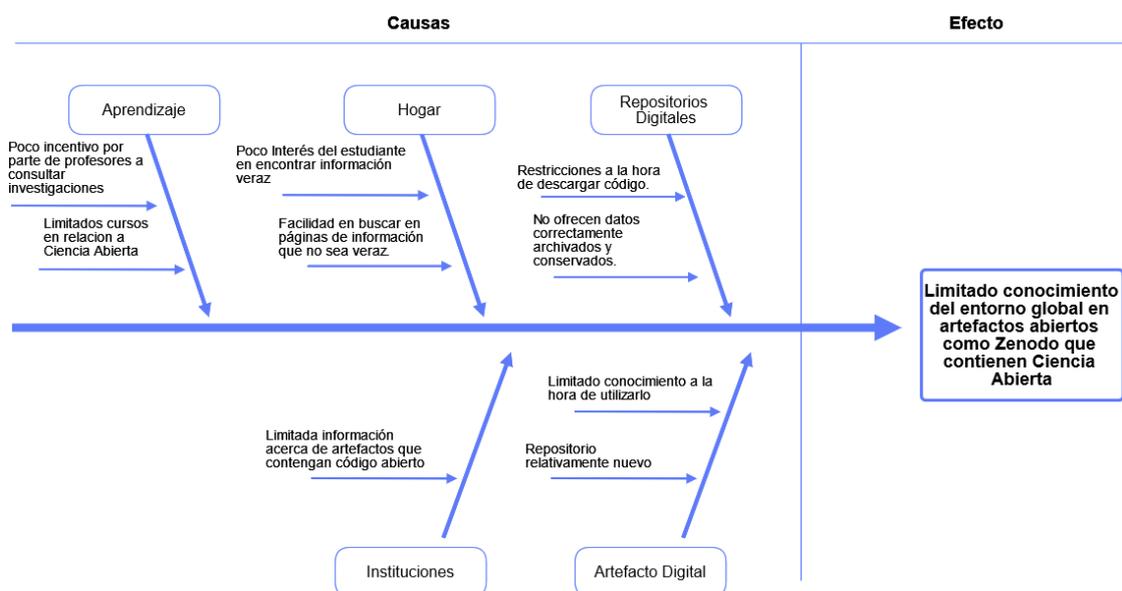


Figura 1: Diagrama de Planteamiento del Problema-Efecto y Causas

Objetivos

Objetivo General

Integrar las APIs REST de Zenodo y GitHub mediante una aplicación orientada a servicios para publicar contenido Open Science.

Objetivos Específicos

1. Conocer el estado actual del uso de las herramientas Zenodo, y GitHub para publicar contenido Open Science.
2. Configurar las plataformas Zenodo y GitHub para integrar y publicar contenido Open Science.
3. Desarrollar una aplicación orientada a servicios para integrar las APIs REST de Zenodo y GitHub y publicar contenido Open Science.
4. Evaluar la eficacia de la aplicación desarrollada basada en las métricas de la ISO/IEC 25022.

Alcance

El objetivo del estudio actual es integrar las API REST de Zenodo y GitHub en una aplicación orientada a servicios para publicar contenido Open Science. Se llevará a cabo una evaluación de la eficacia de la aplicación utilizando las métricas de ISO/IEC 25022.

Primero, se revisará la literatura para determinar el estado actual de las herramientas Zenodo y GitHub para el desarrollo de software que publica contenido Open Science.

En el segundo paso, se configurará las plataformas Zenodo y GitHub las cuáles permitirán la integración y posterior publicación del contenido de una aplicación.

En el tercer paso, se desarrollará una aplicación web orientada a servicios la cual se desarrollará con la arquitectura MVC utilizando la metodología Scrum, el aplicativo se enfocará a un módulo referente a Inventario/Ventas y tendrá las siguientes funcionalidades:

- Control y gestión de usuarios.
- Control y gestión de clientes
- Control y Gestión de Categorías
- Control y Gestión de Productos
- Control y Gestión de Ventas
- Visualización de Reportes
- Integración GitHub y Zenodo

Las tecnologías que se usarán al momento de desarrollar la aplicación web orientada a servicios son:

- Arquitectura de Software: MVC.
- Marco de trabajo ágil: Scrum.
- Lenguaje de programación PHP.
- IDE de programación Visual Studio.
- Motor de base de datos MySql.

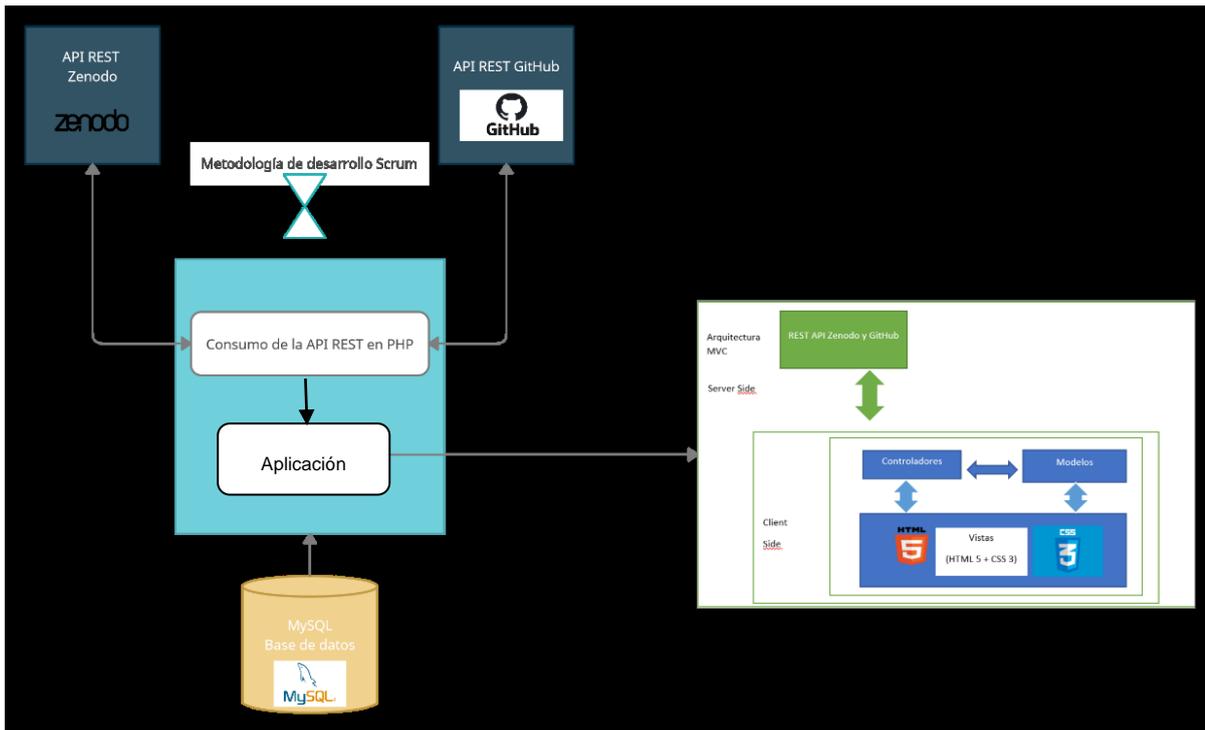


Figura 2: Arquitectura a desarrollar

Metodología

La revisión de la literatura incluirá la creación de una cadena de búsqueda, la selección de artículos, el uso de criterios de inclusión y exclusión y el análisis y presentación de resultados sobre el estado actual del uso de las herramientas Zenodo y GitHub en la publicación de contenido Open Science.

Luego, se configurará las plataformas Zenodo y GitHub, para después desarrollar una aplicación web orientada a servicios que permita consumir las APIs de Zenodo y GitHub, para posteriormente publicar contenido Open Science. Finalmente, se evaluará la eficacia de la aplicación web usando la ISO/EC 25022.

Justificación

La reproducibilidad, replicabilidad y repetibilidad en el desarrollo del software permiten mejorar el aseguramiento de la calidad del software. Este estudio tiene como objetivo brindar un aporte a la comunidad científica.

El Objetivo de Desarrollo Sostenible número 17 de la Agenda 2030, que se centra en el acceso a la ciencia, la tecnología y la innovación, mejorando así el conocimiento compartido en términos mutuamente acordados a través de una tecnología global como mecanismo de facilitación académica, es respaldado por esta investigación.

Justificación Social. – La integración del aplicativo web con las APIs de Zenodo y GitHub permitirá a la comunidad científica a compartir material investigativo libre de restricciones.

Justificación Tecnológica. - “Tanto las nuevas tecnologías como las herramientas están dando forma al mundo de la investigación Open Science, ayudando a los investigadores a ser más eficientes, colaborativos y productivos. Por lo que, la ciencia tendrá más impacto en la sociedad y abordará mejor los desafíos sociales” (Cid, 2017).

CAPÍTULO I

1. MARCO TEÓRICO

Este capítulo ofrecerá una revisión de la literatura sobre las herramientas que se deben usar, como Zenodo, GitHub y el enfoque de investigación Open Science. De esta manera, se podrá obtener una comprensión general de la situación actual de estos estudios con el fin de buscar una solución a la cuestión planteada.

1.1 Estado actual de Open Science, Zenodo y GitHub.

Los repositorios nacionales e internacionales son cruciales para facilitar el acceso a los resultados de la investigación, los métodos y los datos científicos, así como para garantizar la reproducibilidad de la investigación. El almacenamiento de productos de conocimiento tiene como objetivo facilitar el acceso a sus archivos, así como facilitar el almacenamiento, preservación y uso ético de la información. Zenodo, un repositorio en línea de última generación desarrollado por OpenAIRE y CERN y respaldado por la Comisión Europea, representa un paso hacia la ciencia abierta. A través del programa paneuropeo OpenAIRE, ofrece servicios mediante la conexión de los resultados de la investigación con sus conjuntos de datos e información sobre financiación en diferentes países y en Europa. El sitio web ofrece datos de investigación multidisciplinarios de cualquier investigador, comunidad científica o institución de investigación y desarrollo. Permite una gran cantidad de almacenamiento (1 GB), que puede usarse por empresas que no tienen su propio repositorio de datos. Como OpenAIRE, Zenodo se basa en la misma filosofía y recopila publicaciones de acceso abierto de varios programas de financiación. Como se muestra en la Figura 3, Zenodo ofrece una interfaz sencilla que conecta su contenido (publicaciones y datos) con la información de financiación. El protocolo OAI-PMH se puede utilizar por terceros para recopilar datos publicados y

crear colecciones. Puede acceder a estas colecciones a través de la institución de investigación o PubMedCentral. Para las instituciones de investigación que no están dispuestas a pagar el costo de crear sus propios repositorios de datos, Zenodo es una solución adecuada. El repositorio acepta cualquier tipo de datos y formatos. Como resultado, Zenodo promueve los almacenes de datos abiertos y acepta otros tipos de licencias, liderando la promoción de los beneficios de las licencias abiertas, como una mayor visibilidad e influencia. Se espera que el proceso de investigación y los resultados sean más abiertos y fomenten la innovación y la cooperación en ciencia de la información, estadística e informática con la ayuda de las herramientas y tecnologías proporcionadas por el portal.

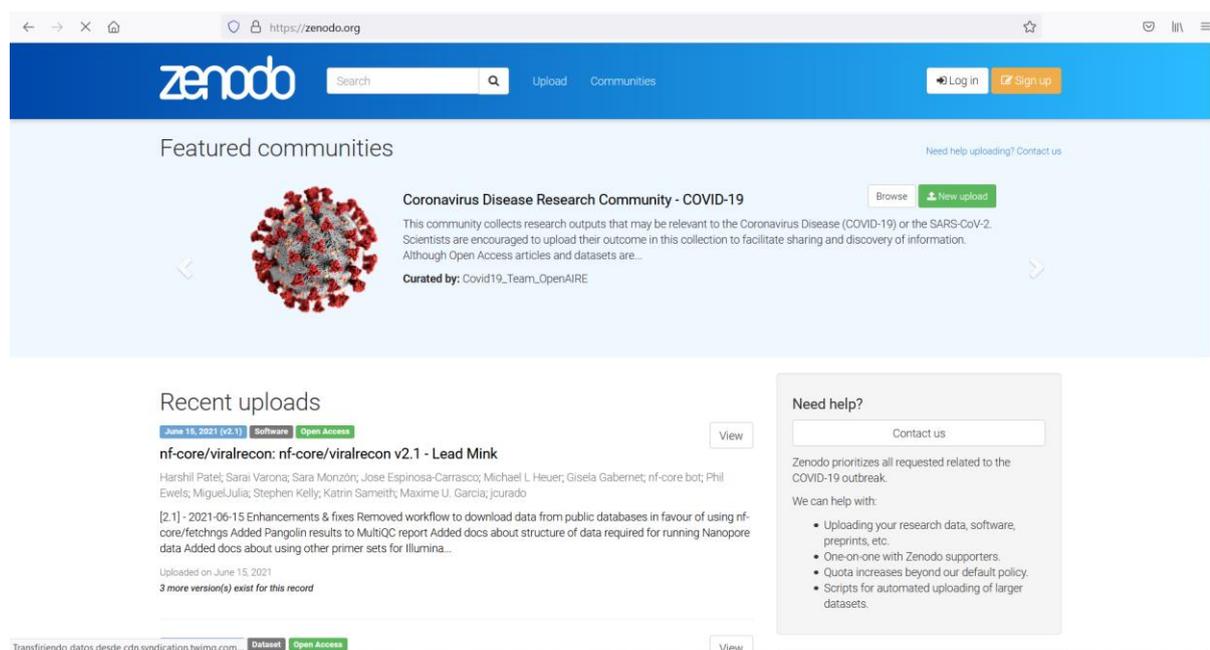


Figura 3: Imagen de la interfaz gráfica de Zenodo

Fuente: <https://n9.cl/vrtn>

Por otro lado, GitHub es un repositorio que promueve el movimiento de ciencia abierta basado en el trabajo colaborativo en código de software. GitHub se lanzó en 2008 para ayudar a los desarrolladores de software. Ahora tiene alrededor de 15 millones de usuarios y es un sitio cada vez más popular para que los investigadores compartan, mantengan y actualicen conjuntos de datos y códigos científicos. Muchos sitios web están dedicados a compartir datos. Pero GitHub está diseñado para una colaboración abierta y transparente porque utiliza un software de control de versiones para rastrear cada cambio realizado en el código o los datos. Esto significa que un gran equipo de desarrollo distribuido puede colaborar en un proyecto en línea, y los usuarios pueden revisar el historial de versiones del archivo para ver cada cambio, cuándo se modificó, quién lo cambió y su propósito. Los programadores pueden copiar ('bifurcar') un repositorio para experimentar con nuevas ideas; los cambios útiles se pueden fusionar en el proyecto principal y otros cambios se pueden ignorar o deshacer más tarde.

1.1.1 Definición de la cadena de búsqueda.

La revisión sistemática de la literatura es un tipo de investigación secundaria que comienza con preguntas de investigación específicas que permiten la identificación, selección y evaluación crítica de la documentación necesaria para recopilar y analizar los datos e información que se incluyen en la revisión (Moher et al., 2010). El objetivo principal es garantizar que la revisión de la literatura sea imparcial, rigurosa y auditable. Las cadenas de búsqueda utilizaron conectores booleanos (OR y AND) y criterios de inclusión y exclusión para vincular términos relacionados con Open Science. Posteriormente, se utilizó el motor de búsqueda Google Scholar para buscar información no estructurada para verificar otras fuentes de instituciones o firmas consultoras reconocidas sobre temas comerciales para fundamentar y demostrar la

relevancia de las preguntas clave y eliminar sesgos. Las bases de datos que fueron consultadas para la búsqueda estructurada fue Scopus, Science Direct, Springer, Google Scholar en donde se aplicó como cadena de búsqueda base: (*“Open Science” AND “Zenodo” AND “Integration of repositories” AND “GitHub”*), los artículos seleccionados deben tener relación con el tema con restricción referente a la fecha desde los años 2008-2021, esto debido a que Open Science es un movimiento que ha comenzado a tomar un mayor impacto desde la aparición de repositorios digitales. Para el proyecto de tesis se decidió realizar una revisión sistemática de la literatura porque las preguntas de investigación nos ayudan a garantizar que los resultados se apoyen en la evidencia de mejor calidad para responder dichas cuestiones, dentro de esta se procedió a realizar la búsqueda en bases de datos bibliográficas como Scopus, ScienceDirect, Springer y Google Scholar. La utilización del diccionario OpenEdition Journals ayudo al presente proyecto a identificar las palabras clave para que los resultados obtenidos sean encontrados de una manera eficiente y eficaz.

Tabla 1: Palabras Clave

| N° | Palabras Clave |
|-----------|-----------------------|
| 1 | Integración |
| 2 | Integration AND API |

| | |
|----------|-------------------------|
| 3 | Open Science |
| 4 | Revisiones Sistemáticas |
| 5 | Integration |
| 6 | Zenodo |
| 7 | GitHub |

La metodología seleccionada para realizar la investigación del proyecto de tesis fue propuesta por (Kitchenham, 2007). El protocolo de revisión consta de 4 fases fundamentales: Preguntas de investigación, búsqueda de documentos, selección de artículos, extracción de datos relevantes. A continuación, se explica cada una de estas fases.

1.1.2 Preguntas de investigación

Se establecieron tres preguntas de investigación (PI), tabla 2, las cuales nos servirán de guía para el proceso de revisión sobre el tema del proyecto de tesis.

| N° | Preguntas de investigación | Motivación |
|------------|---|-------------------|
| PI1 | ¿Cuál es la situación actual de Identificar el nivel en el que se | |

| | |
|---|--|
| Open Science dentro de un contexto investigativo? | encuentra el movimiento Open Science dentro de los resultados investigativos en nuestro entorno. |
| PI2 ¿Qué tipo de restricciones tiene la plataforma GitHub al momento de subir documentación? | Identificar cuáles son las reglas que tiene la plataforma para que por medio del presente proyecto poder liberar esas restricciones. |
| PI3 ¿La integración de repositorios digitales supone una ventaja o desventaja frente a la subida de documentación? | Identificar cuáles son las ventajas y desventajas que conlleva el subir documentación en repositorios digitales. |

Tabla 2: Pregunta de investigación (PI)

1.1.3 Búsqueda de documentos

En esta etapa se busca ampliar los conocimientos del investigador para darle elementos de valoración y apropiación para ajustar los elementos de planificación y las sentencias de búsqueda. Por coherencia con el tema, se seleccionaron las siguientes metodologías para la búsqueda, recuperación y depuración de la producción científica, incluyendo tanto fuentes cerradas como abiertas:

- Bases de datos comerciales: Scopus
- Repositorios y directorios: Science Direct
- Redes sociales científicas: Springer

- Buscadores académicos: Google Scholar

Una vez seleccionados los documentos en las diferentes bases de datos bibliográficas se hizo una recopilación de 15 documentos que tenían una mejor correlación en base a las preguntas de investigación, tal y como se muestra en la figura 4.

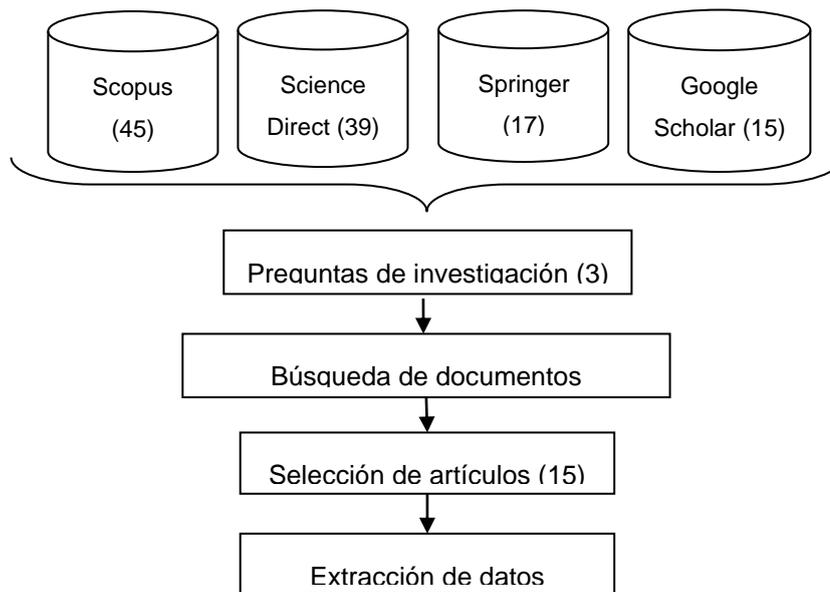


Figura 4: Diagrama del protocolo de investigación

1.1.4 Criterios de Inclusión y Exclusión

Se seleccionaron los documentos que podrían utilizarse para depurar y administrar un banco de datos, eliminando los textos duplicados que no eran relevantes para los objetivos de la investigación (otros tipos de bibliotecas) o que no se podía acceder al texto completo. Después de completar este procedimiento, se unieron los quince documentos que se utilizaron para el análisis.

1.1.5 Selección de artículos

Se consideraron tres etapas principales para la selección de artículos. Los criterios de inclusión y exclusión se aplicaron en la primera fase. Los criterios para la inclusión incluían artículos científicos y trabajos de titulación. Los trabajos se publicaron en inglés y español entre 2014 y 2021 y están relacionados con las ciencias de la computación. Los resúmenes ejecutivos, las fichas técnicas, los trabajos duplicados y los estudios publicados en otras áreas de conocimiento fueron criterios de exclusión. En la segunda fase se aplicaron criterios de lectura de calidad con diferentes niveles de lectura; en el nivel de lectura 1, se seleccionaron 35 referencias más relevantes, teniendo en cuenta la relación y consistencia con el tema que está siendo investigado; en el nivel de lectura 2 se seleccionaron 15 referencias, las cuales fueron las más relevantes de acuerdo con su valor y precisión con respecto a los demás artículos encontrados, teniendo también en cuenta lo original de la contribución y la validez de las conclusiones. En la fase tres estos artículos se leyeron completamente y se realizó un análisis profundo para validar que la información es útil para el trabajo de investigación que se está realizando. En la tabla 3 se listan los artículos seleccionados.

| Código | Título | Autor |
|---------------|--|-------------------------|
| A1 | An empirical meta-analysis of the life sciences linked open data on the web. | Kamdar M.R., Musen M.A. |
| A2 | patRoon: open source software platform for | Helmus R., ter Laak |

| | | |
|-----------|--|--------------------------|
| | environmental mass spectrometry based non-target screening. | T.L. |
| A3 | An open data repository and a data processing software toolset of an equivalent Nordic grid model matched to historical electricity market data. | Magro, C., et al. |
| A4 | Tackling the digitalization challenge: how to benefit from digitalization in practice. | Luigi Vanfretti., et al. |
| A5 | A sensitivity analysis of factors influential to the popularity of shared data in data repositories. | Xie Q., et al. |
| A6 | A Survey of Utility-Oriented Pattern Mining Open Access. | Gan, W., et al. |
| A7 | TOSSH: A Toolbox for Streamflow Signatures in Hydrology. | Gnann S.J., et al. |
| A8 | Quotebank: A Corpus of Quotations from a Decade of News. | Vaucher T., et al. |
| A9 | ChemProps: A RESTful API enabled database for composite polymer name standardization. | Hu B., et al. |

| | | |
|------------|---|------------------------|
| A10 | The aspect of mobility and connectivity while assessing the neighbourhood sustainability. | Verovsek S., et al. |
| A11 | New scenarios and challenges for open science. Between optimism and uncertainty. | Fressoli M., et al. |
| A12 | Open science: A model with some pieces still to fit in. | Abadal E. |
| A13 | Open Science from the South. | Botero C. |
| A14 | Open science and integration between scientific peers. | Palmera J.R. |
| A15 | Open science: Evolution of the name and the concept. | Falgueras E.A., et al. |

Tabla 3: Artículos seleccionados para la revisión literaria

1.1.6 Análisis de Resultados

En esta fase se lleva a cabo un proceso de revisión en donde se elaboró una matriz que nos permite identificar los conceptos más relevantes del estado actual de las plataformas de Zenodo y GitHub. En la Tabla 4, se detallan los artículos relevantes y sus conceptos.

Artículos

| Código | entorno. | Integración de APIs. | Science en el ámbito informático | de subir contenido en repositorios virtuales. | digital. | Science a lo largo de los años | educación superior. | Science en repositorios educativos | Open Science. |
|--------|----------|----------------------|----------------------------------|---|----------|--------------------------------|---------------------|------------------------------------|---------------|
| A1 | x | | | | | | | | |
| A2 | | x | x | x | | | | | |
| A3 | | | | | x | x | x | | |
| A4 | x | | | | | | | | |
| A5 | x | | x | X | | | | | |
| A6 | x | | | | x | | | | |
| A7 | x | | | | | x | x | | |
| A8 | x | | | | | | x | | |
| A9 | x | | x | | | | | | |

| | | | | | | | | | | | | | |
|------------|---|--|--|--|---|---|---|--|--|--|---|---|---|
| A10 | x | | | | | | x | | | | x | | |
| A11 | x | | | | x | x | | | | | | | |
| A12 | x | | | | x | | x | | | | | | |
| A13 | x | | | | x | | | | | | | | |
| A14 | x | | | | x | | | | | | | | |
| A15 | x | | | | x | x | | | | | | x | x |

Tabla 4: Matriz de conceptos

1.2 Open Science

1.2.1 Concepto de Open Science

La teoría de las revoluciones científicas fue desarrollada por el físico y filósofo de la ciencia Thomas Kuhn en su libro "La estructura de las revoluciones científicas", que es una buena manera de explicar qué es Open Science. Este autor define los "paradigmas" como las realizaciones científicas universalmente reconocidas que, durante un período de tiempo, proporcionan modelos de problemas y soluciones a una comunidad científica. Las tradiciones científicas y sus métodos se mantuvieron a pesar de la evolución de la ciencia y sus prácticas. A pesar de las disfunciones o "anomalías" que contiene, un paradigma se mantiene en el tiempo. El proceso de sustitución de un paradigma "viejo" por uno nuevo es conflictivo debido al proceso emergente de nuevas prácticas y nuevas herramientas, así como a la inseguridad

derivada del abandono de maneras de hacer bien asentadas en el tiempo. Esto es Open Science: un cambio de paradigma en la forma en que se hace ciencia. Esta no altera significativamente sus motivaciones y objetivos, pero sí altera significativamente sus métodos. Lo que se hace no es el cambio, sino cómo se hace. Open Science implica una serie de etapas desde el inicio de la investigación hasta la publicación de los hallazgos. Estos cambios han sido conocidos como "ciencia 2.0" o "ciencia abierta". La Figura 5 muestra claramente estos cambios, ya que cada una de las fases del ciclo de investigación está respaldada por herramientas que no existían hace cinco o diez años. Aunque es importante, la ilustración de la figura es limitada. La ciencia emergente debe ser colaborativa, abierta y "en beneficio de la sociedad". Retomaremos estas ideas, sin embargo, no debemos ignorar las razones políticas y sociales para aceptar esta modificación. Según Fecher y Friesike (2014), OS es la práctica de compartir el conocimiento obtenido de la investigación financiada con fondos públicos de forma completamente abierta, libre y sin restricciones de uso. Los autores han dado casi tantas definiciones de OS, y hay varios análisis de revisión bibliográfica que reflejan esta situación. Incluso se puede hablar de escuelas o tendencias para determinar el concepto (Fecher y Friesike, 2014).

Después de analizar las definiciones de los autores mencionados, establecemos que Open Science es la "investigación académica colaborativa, transparente y reproducible, cuyos resultados están disponibles para todos". Esta definición nos permite garantizar que todos los principios de OS, tal y como lo presenta la Comisión Europea en 2019, se incluyan.

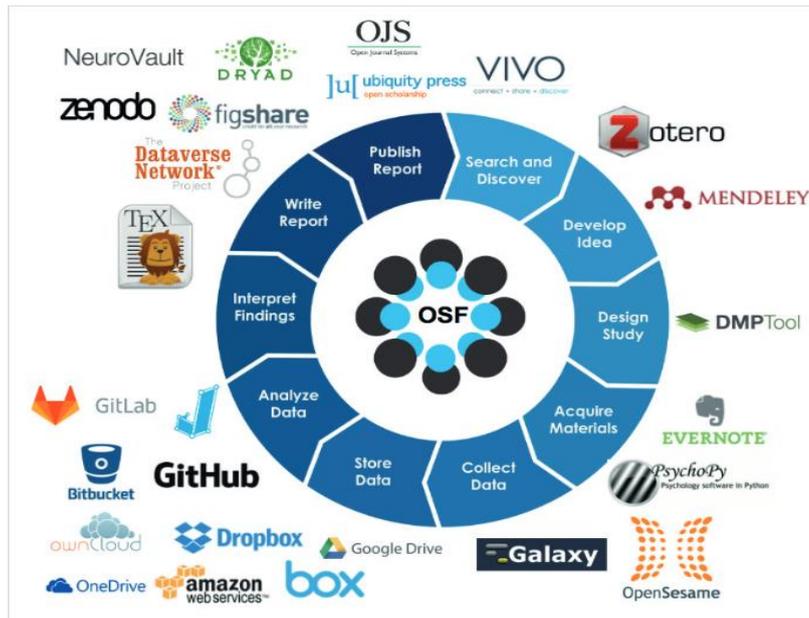


Figura 5: Fases de la investigación

Fuente: “The open science framework” Center for Open Science <https://cos.io/our-products/osf>



Figura 6: Elementos constitutivos de la Open Science (Miembros de la Comisión Europea)

Fuente: Open Science por defecto. La nueva normalidad para la investigación.

<https://doi.org/10.3989/arbor.2021.799002>

1.2.2. Alcance de Open Science en un contexto empresarial.

Open Science también conocida como Ciencia abierta, se refiere a un enfoque colaborativo y transparente en la investigación científica. El objetivo es compartir datos, métodos, resultados y conocimientos generados durante el proceso de investigación de manera abierta y accesible. Aunque Open Science generalmente se asocia con el campo académico y científico, algunas empresas también incorporan los valores de Open Science en las operaciones. Dentro de las formas más usuales en que las empresas hacen uso de Open Science son las siguientes:

1. Colaboración externa: Las empresas pueden utilizar los principios de Open Science para trabajar con la comunidad académica y científica. Pueden abrir sus datos para que otros investigadores los utilicen para análisis, descubrimientos o aportes de nuevas ideas. Es posible que la cooperación resulte en avances científicos y tecnológicos mutuamente beneficiosos.
2. Compartir datos: Las empresas pueden compartir datos de investigación de manera abierta al adoptar prácticas de Open Science. Esto puede ser particularmente importante en industrias como la salud, la biotecnología o la inteligencia artificial, donde el acceso a datos de alta calidad es esencial para el avance científico. La replicación de estudios, la validación de resultados y el fomento de la innovación en la comunidad científica son posibles gracias al intercambio de datos.

3. Desarrollo de Software de código abierto: Las empresas pueden participar en la creación de software de código abierto relacionado con sus campos de especialización. Permiten que otros desarrolladores utilicen, mejoren y adapten sus herramientas o bibliotecas al abrir su código fuente. Esto alienta a trabajar juntos, ser transparentes y construir ecosistemas de software más fuertes.
4. Publicación abierta de resultados de investigación: Algunas organizaciones están implementando políticas de publicación abierta para difundir los hallazgos de sus estudios. Esto implica publicar trabajos científicos en revistas o repositorios públicos de acceso abierto, esto cual facilita el acceso a la información y fomenta el intercambio de conocimientos entre la comunidad científica.

Es importante tener en cuenta que algunas empresas utilizan Open Science de manera diferente. La industria, el tamaño y los objetivos estratégicos de una empresa pueden determinar la adopción de principios de ciencia abierta. Sin embargo, muchas organizaciones están considerando cómo incorporar los principios de transparencia, colaboración e intercambio de conocimientos que promueve Open Science en sus programas de investigación y desarrollo.

| Beneficios | Indicadores |
|--|--|
| Colaboración en la cadena de suministro | <ul style="list-style-type: none"> • Número de colaboraciones establecidas con proveedores, fabricantes y distribuidores. • La comunicación y la toma de decisiones conjuntas mejoran. |

Compartir datos de demanda y consumo

- Participación de expertos externos en la evaluación y mejora de la gestión de inventarios.
- Cantidad y calidad de la retroalimentación y el análisis proporcionados por expertos externos.

Mejora de procesos y optimización de recursos

- Reducción en el exceso de inventario o en la falta de stock.
- Precisión de los pronósticos de demanda y su precisión en comparación con los resultados reales.
- Optimización de los costos de inventario, como reducción de obsolescencia o desperdicio.

Innovación abierta

- Capacidad para identificar y anticipar cambios en los patrones de demanda.
 - Análisis de la demanda por segmento de productos, regiones
-

geográficas, etc.

- Identificación de factores o eventos externos que influyen en la demanda.

Tabla 5: Indicadores de Open Science en la gestión de inventarios

1.2.6 Open Science en las empresas ecuatorianas

Hasta septiembre de 2022, las empresas ecuatorianas aún no habían adoptado Open Science en la gestión de inventarios. El país aún estaba desarrollando el concepto de Open Science y sus aplicaciones específicas en ese momento.

En Ecuador se están tomando medidas para promover el acceso abierto a la información científica y tecnológica, así como la colaboración en la investigación. La adopción de prácticas Open Science, como su aplicación en la gestión de inventarios, requiere cambios culturales, recursos y una comprensión clara de los beneficios asociados.

En 2023, algunas empresas ecuatorianas podrían estar explorando o implementando prácticas de Open Science en la gestión de inventarios. Hay casos únicos de empresas que han comenzado a compartir información o colaborar con la comunidad académica y científica sobre la gestión de inventarios.

En un estudio realizado en septiembre de 2022 por organizaciones y grupos empresariales como la Federación Ecuatoriana de Exportadores (FEDEXPOR) y la Federación Nacional de Cámaras de la Pequeña y Mediana Empresa del Ecuador

(FENACAPE), Resultó que alrededor del 74% de las empresas en Ecuador no han implementado o están en proceso de implementar prácticas de ciencia abierta en la gestión de inventarios. En base a estos porcentajes obtenidos el mismo estudio indica que el 26% restante de las empresas que no implementan prácticas de Ciencia Abierta, se debe a un factor que fue el que más se evidencio en este estudio que fue la falta de innovación por parte de estas. Si embargo, este porcentaje se puede interpretar como una práctica poco común, pero con potencial de mejora y oportunidad para aquellas organizaciones que aún no la han adoptado. También puede indicar que existen barreras o desafíos que dificultan su implementación generalizada.

1.3 GitHub

1.3.1 Concepto, funcionalidad y origen.

Se describe a GitHub como un repositorio de control de versiones basado en la web que permite a los desarrolladores alojar, colaborar y supervisar las versiones de su código fuente. Los equipos de desarrollo pueden colaborar en proyectos y compartir su trabajo con otros usuarios en una plataforma fácilmente accesible y colaborativa llamada GitHub.

A continuación, se detallaron las funcionalidades clave de GitHub. Estas incluyen:

1. Repositorios: Los usuarios pueden establecer repositorios para almacenar y organizar su código fuente. En el mismo repositorio, pueden monitorear los cambios, revertir a versiones anteriores y colaborar con otros desarrolladores.
2. Ramas (branches): Las ramas permiten a los desarrolladores trabajar en paralelo en una variedad de características o soluciones sin afectar el proyecto principal.

Esto facilita el desarrollo y la colaboración simultáneas en varios aspectos del proyecto.

3. Solicitudes de extracción (Pull Requests): Las solicitudes de extracción son una forma de sugerir cambios en un repositorio. Los desarrolladores pueden enviar solicitudes de extracción para que otros revisen y aprueben sus cambios antes de fusionarlos con el departamento principal del proyecto.
4. Problemas (Issues): Para rastrear errores, solicitudes de funciones o tareas pendientes en un proyecto, se utilizan problemas. Dentro del repositorio, los desarrolladores y colaboradores pueden comentar, asignar y resolver problemas.
5. Wikis y documentación: GitHub permite la creación de wikis y documentación que estén relacionadas con un repositorio, lo que facilita el trabajo en equipo y el intercambio de conocimientos.

GitHub cuenta con una estructura de trabajo muy intuitiva, hecha para los desarrolladores que requieren realizar sus proyectos colaborativos de una manera rápida e intuitiva. En la figura 7 podemos observar la estructura básica de trabajo de GitHub.

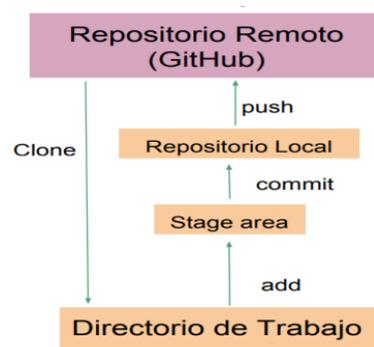


Figura 7: Estructura básica de trabajo de GitHub

Fuente: <https://n9.cl/x0pyy>

Finalmente, examinamos cuán importante es incorporar la API de GitHub en una aplicación orientada a servicios. La API de GitHub permite que las aplicaciones accedan de manera programática a los repositorios, ramas, solicitudes de extracción, problemas y otras funcionalidades de GitHub. Esto proporciona ventajas significativas, como una mayor automatización de tareas, una mejor sincronización entre la aplicación y GitHub y una mayor eficiencia en la gestión de proyectos de desarrollo de software.

Por último, pero no menos importante, la integración de la API de GitHub en una aplicación orientada a servicios permite aprovechar las funcionalidades y la colaboración de GitHub para mejorar la gestión de proyectos de desarrollo de software.

1.3.2 Ventajas y Desventajas de GitHub.

| Ventajas | Desventajas |
|--|--|
| Facilita el control de versiones del código fuente del sistema de inventario, permitiendo mantener un historial completo de cambios. | Requiere conocimientos técnicos y experiencia en el uso de control de versiones para aprovechar todas sus funcionalidades. |
| Permite la colaboración eficiente entre múltiples desarrolladores o equipos, facilitando la coordinación y la resolución de problemas | La gestión de archivos binarios o grandes volúmenes de datos puede ser menos eficiente en comparación con archivos de texto plano. |

| | |
|---|---|
| Proporciona un entorno seguro y confiable para almacenar y respaldar el código fuente del sistema de inventario. | Puede haber costos asociados con el uso de características y servicios avanzados de GitHub, como repositorios privados o integraciones adicionales. |
|---|---|

| | |
|--|---|
| Ofrece herramientas y funcionalidades adicionales, como problemas (issues) y solicitudes de extracción (pull requests), para facilitar el seguimiento y la resolución de problemas en el sistema de inventario. | Puede haber una curva de aprendizaje inicial para aquellos que no están familiarizados con el flujo de trabajo y la terminología de GitHub. |
|--|---|

| | |
|--|--|
| Proporciona integraciones con otras herramientas y servicios populares, lo que permite una mayor automatización y personalización del flujo de trabajo del sistema de inventario. | Dependiendo de los requisitos de seguridad y privacidad de la empresa, puede haber preocupaciones sobre el almacenamiento y la propiedad de los datos en un entorno externo. |
|--|--|

Tabla 6: Ventajas y Desventajas de GitHub en el manejo de sistemas de inventario.

1.3.3 Integración de GitHub con Zenodo.

Las empresas pueden obtener múltiples ventajas en el manejo de su sistema de inventarios al integrar GitHub con un repositorio como Zenodo. Algunos de ellos son los siguientes:

1. Almacenamiento y respaldo seguro: Zenodo, un repositorio de acceso abierto permite a las empresas almacenar y respaldar sus archivos relacionados con el sistema de inventarios, como el código fuente. La integración con GitHub facilita la sincronización automática de los repositorios y garantiza que los datos estén respaldados y accesibles de manera segura.
2. Preservación a largo plazo: Zenodo se enfoca en la preservación de los datos y el código fuente a largo plazo. Las empresas pueden asegurarse de que su sistema de inventarios esté respaldado y preservado a largo plazo al integrar GitHub con Zenodo; esto es particularmente importante para proyectos de larga duración o sistemas críticos.
3. Mayor visibilidad y fácil acceso: Zenodo es un repositorio de acceso abierto que permite el intercambio y el acceso público a los datos. Las empresas pueden mejorar la visibilidad de su sistema de inventarios al integrarlo con GitHub, lo que facilita la colaboración, la revisión por pares y el intercambio de información con la comunidad.
4. Reconocimiento y citación: Zenodo ofrece un sistema de citación que permite asignar identificadores persistentes en archivos y repositorios. Esto permite a las empresas citar y recibir reconocimiento por su sistema de inventarios, lo que puede ser crucial para el cumplimiento de requisitos de informes o divulgación académica.
5. Interoperabilidad y estándares abiertos: Zenodo fomenta la interoperabilidad con otras plataformas y servicios y es compatible con estándares abiertos. El sistema de inventarios puede funcionar de manera conjunta con otras herramientas y servicios gracias a su integración con GitHub, lo que facilita la integración en flujos de trabajo existentes y aumenta las oportunidades de colaboración.

En resumen, la integración de GitHub con un repositorio como Zenodo brinda ventajas como almacenamiento y respaldo seguro, preservación a largo plazo, citación y reconocimiento, interoperabilidad con estándares abiertos y mayor visibilidad y accesibilidad. Estos beneficios pueden mejorar la gestión y la divulgación del sistema de inventarios de una empresa al fomentar la colaboración, la transparencia y el intercambio de conocimientos.

1.4 Zenodo

1.4.1 Concepto, Origen y Funcionalidad.

Para almacenamiento y preservación a largo plazo de datos científicos y otros materiales relacionados con la investigación, Zenodo es un repositorio de acceso abierto. En 2013, el CERN estableció esta iniciativa como parte de sus esfuerzos por fomentar la ciencia abierta y el acceso público a los resultados de la investigación.

La principal característica de Zenodo es su compromiso con los principios de la ciencia abierta. ofrece un espacio en línea gratuito donde los investigadores pueden depositar y compartir datos, códigos fuente, documentos y otros tipos de contenido científico. Los materiales que se depositan en Zenodo tienen un identificador digital persistente (DOI), que permite la citación y el reconocimiento de los materiales en publicaciones académicas.

Zenodo fomenta la transparencia, la reutilización y la reproducibilidad de la investigación científica, lo que lo convierte en una parte importante del ámbito Open Science. Zenodo fomenta la colaboración entre investigadores de diversas disciplinas y organizaciones al ofrecer un repositorio de acceso abierto.

En síntesis, Zenodo es una iniciativa del CERN para crear un repositorio de acceso abierto. Proporcionar un espacio para el almacenamiento y la preservación de datos científicos y otros materiales de investigación es su función principal. Zenodo fomenta la transparencia y el intercambio de conocimientos en la comunidad científica a través de su contribución an Open Science.



Figura 8: Registro de Zenodo con indicadores Altmetrics y DOI

Fuente: <https://zenodo.org/record/5114203#.YPXjT-hKhEY>

1.5. ISO/IEC 25022:2016

La norma ISO/IEC 25022:2016, también conocida como norma de calidad de productos de software, establece pautas para la evaluación y medición de la calidad del software. Esta norma proporciona un marco de trabajo para establecer métricas y evaluar características de calidad particulares del software.

La ISO/IEC 25022:2016 se basa en el concepto de calidad del software y ofrece una estructura sistemática para evaluar una variedad de aspectos de calidad, como la funcionalidad, la confiabilidad, la eficiencia, la usabilidad y la mantenibilidad.

Proporcionar pautas claras y consistentes para medir y evaluar la calidad del software es el objetivo principal de la norma, lo que permite a las organizaciones realizar evaluaciones objetivas y mejorar continuamente sus productos y servicios de software.

La norma ISO/IEC 25022:2016 tiene varios componentes, como definiciones de conceptos clave, criterios de evaluación, técnicas de medición y pautas para la interpretación de los resultados. Estas secciones proporcionan una base sólida para llevar a cabo evaluaciones de calidad y permiten comparar y comunicar los hallazgos de manera efectiva.

Fuente: <https://www.iso.org/standard/35746.html>.

| Métricas de calidad en uso | | |
|----------------------------|---|--|
| Características | Subcaracterísticas | Métricas |
| Efectividad | Efectividad | <ul style="list-style-type: none"> • Completitud de la tarea. • Efectividad de la tarea. • Frecuencia de error. |
| Eficiencia | Eficiencia | <ul style="list-style-type: none"> • Tiempo de la tarea. • Tiempo relativo de la tarea. • Eficiencia de la tarea. • Eficiencia relativa de la tarea. • Productividad Económica. • Porcentaje productivo. • Número relativo de las acciones del usuario. |
| Satisfacción | Utilidad | <ul style="list-style-type: none"> • Nivel de satisfacción. • Uso discrecional de las funciones. • Porcentaje de quejas de los clientes. |
| Libertad de Riesgo | Libertad de riesgo económico | <ul style="list-style-type: none"> • Retorno de la inversión (ROI). • Tiempo para lograr el (ROI). • Rendimiento relativo de negocios. • Balanced Score Card. • Tiempo de entrega. • Ganancias para cada cliente. • Errores con consecuencias económicas. • Corrupción del software. |
| | Libertad de riesgo de salud y seguridad | <ul style="list-style-type: none"> • Frecuencia de problemas en la salud y seguridad del usuario. • Impacto de la salud y la seguridad del usuario. • Seguridad de las personas afectadas por el uso del sistema. |
| | Libertad de riesgo ambiental | <ul style="list-style-type: none"> • Impacto Ambiental. |
| Cobertura del Contexto | Completitud de contexto | <ul style="list-style-type: none"> • Completitud del contexto. |
| | Flexibilidad | <ul style="list-style-type: none"> • Función flexible del diseño. |

Figura 9: Métricas ISO/IEC 25022:2016

1.6. Indicadores por desarrollar

El presente estudio investigará cómo crear un sistema de gestión de inventario para una microempresa que vende alimentos. La integración de GitHub con Zenodo, que seguirá los principios de Open Science, permitirá que todos los artefactos y componentes del proyecto sean enviados y publicados en Zenodo como contenido de acceso abierto. El objetivo principal de esta tesis es crear una aplicación que facilite la integración entre GitHub y Zenodo. Esto permitirá a la microempresa de alimentos compartir y publicar su contenido científico y tecnológico de manera abierta y accesible para la comunidad académica y el público en general. Al hacerlo, se busca fomentar la transparencia y la colaboración en el ámbito empresarial y apoyar el progreso de la Ciencia Abierta. Este estudio evaluará la eficacia de la aplicación de gestión de inventario y su integración con GitHub y Zenodo. Algunos de los indicadores que se deben tener en cuenta incluyen (Ver Tabla 7):

Tabla 7: Indicadores cualitativos y cuantitativos

| Principios | Tipo | Medición | Interpretación cualitativa |
|--|--------------|-----------------|---|
| 1. Número de contribuciones externas: | Cuantitativo | Recuento | Mayor número indica mayor colaboración y aportes externos |

| | | | |
|--|--------------|----------------------|--|
| 2. Tasa de errores corregidos: | Cuantitativo | Porcentaje | Mayor tasa indica mejor capacidad de corrección de errores |
| 3. Número de forks y estrellas: | Cuantitativo | Recuento | Mayor número indica mayor interés y adopción del proyecto |
| 4. Número de descargas desde Zenodo | Cuantitativo | Recuento | Mayor número indica mayor difusión y visibilidad del contenido |
| 5. Retorno de la inversión | Cuantitativo | Escala de valoración | Mayor valor indica mayor beneficio y eficacia del proyecto |

1. Número de contribuciones externas: Este indicador mide la cantidad de contribuciones y mejoras realizadas por desarrolladores externos utilizando pull requests en el repositorio de GitHub de la empresa. Cuantas más contribuciones externas reciba el proyecto, más evidencia de colaboración y de cómo el código abierto ha atraído a otros desarrolladores a mejorar el software.
2. Tasa de errores corregidos: este indicador mide la cantidad de errores o problemas reportados en el repositorio de GitHub que han sido resueltos y cerrados. Una alta tasa de errores corregidos indica un enfoque proactivo para abordar problemas y una mejora continua del software.
3. Número de forks y estrellas: El número de forks (bifurcaciones) y estrellas (marcadores) en el repositorio de GitHub puede indicar el nivel de interés

y popularidad del software. Cuantos más forks y estrellas tenga el proyecto, más se comparte y utiliza en otros proyectos.

4. Número de descargas de Zenodo: Se pueden descargar registros de depósito en Zenodo. Este indicador calcula la cantidad de descargas del registro de depósito del software que ha recibido la empresa. Un alto número de descargas demuestra un gran interés en el proyecto y su utilidad para otros.
5. Retorno de la inversión: este indicador evalúa los beneficios económicos que obtiene la empresa de la colaboración y las contribuciones de la comunidad abierta. Esto puede ayudar a reducir los costos de desarrollo, mejorar la calidad del software y generar oportunidades comerciales.

CAPÍTULO 2

2.1. Análisis, diseño y desarrollo de la arquitectura

2.1.1. Arquitecturas de software.

- Arquitectura orientada a servicios (SOA)

La arquitectura orientada a servicios (SOA) es un enfoque arquitectónico para la creación y desarrollo de sistemas de software que divide las funcionalidades del sistema en servicios independientes y autónomos. En lugar de crear una aplicación monolítica, una arquitectura orientada a servicios se basa en la creación de servicios interoperables que se comunican entre sí a través de interfaces bien definidas.

Los servicios son unidades lógicas y granulares que encapsulan funciones comerciales particulares en una arquitectura orientada a servicios. Cada servicio se desarrolla de manera independiente y se presenta a través de interfaces estándar y abiertas que utilizan protocolos de comunicación como HTTP, SOAP o REST. Estos servicios se pueden brindar a través de una amplia gama de tecnologías y pueden estar integrados en una amplia gama de sistemas y plataformas.

La principal ventaja de una arquitectura orientada a servicios es su capacidad para fomentar la reutilización, la flexibilidad y la interoperabilidad. Se puede aumentar la flexibilidad y la modularidad del sistema dividiendo las funcionalidades en servicios independientes. Los servicios pueden ser reutilizados en una variedad de situaciones y pueden evolucionar de manera independiente sin afectar a otros componentes del sistema. Además, los servicios pueden conectarse a través de interfaces estándar con

otros sistemas y tecnologías, lo que facilita la integración con sistemas existentes y la creación de aplicaciones compuestas.

Algunas características importantes de una arquitectura orientada a los servicios son:

1. Granularidad: Los servicios se diseñan para ser lo más pequeños y específicos posibles, lo que permite una mayor flexibilidad y reutilización.
2. Independencia: Cada servicio se desarrolla y despliega por sí solo, lo que facilita su mantenimiento.
3. Descubrimiento y registro: los servicios pueden encontrarse y registrarse en un repositorio centralizado, lo que permite a otros sistemas encontrar y utilizar los servicios disponibles.
4. Composición: Para crear aplicaciones más complejas y completas, los servicios se pueden combinar y orquestar.
5. Escalabilidad y tolerancia a fallas: Al ser servicios independientes, se pueden escalar y administrar de forma individual, lo que mejora la disponibilidad y la capacidad de respuesta del sistema.

Reutilizar Integraciones

La principal distinción entre una arquitectura orientada a servicios y una arquitectura orientada a microservicios radica en el alcance y la precisión de los componentes utilizados.

Los servicios son unidades lógicas que encapsulan funcionalidades específicas del negocio en una arquitectura orientada a servicios (SOA). Estos servicios pueden usarse

a través de interfaces bien definidas y se comunican entre sí. El objetivo principal es fomentar la reutilización y la interoperabilidad al crear servicios independientes. En una arquitectura orientada a servicios, los servicios pueden ser de gran tamaño y abarcar una variedad de funcionalidades relacionadas.

Por otro lado, los componentes de una arquitectura orientada a microservicios son aún más pequeños y granulares. Los microservicios son unidades de software autónomas e independientes que se enfocan en una función específica de la empresa. Cada microservicio crece, expande y escala de forma independiente. Se comunica con otros microservicios a través de protocolos ligeros como HTTP o mensajes. La arquitectura de microservicios se basa en la idea de dividir una aplicación en múltiples servicios pequeños y cohesivos que pueden gestionarse de manera más eficiente y evolucionar de forma independiente.

ESB

Un bus de servicios empresariales (ESB) es un componente de software que se utiliza en arquitecturas orientadas a servicios (SOA) para facilitar la integración de sistemas y la comunicación entre aplicaciones de manera ágil y adaptable.

El ESB sirve como una capa intermedia que conecta y coordina una variedad de sistemas y servicios, permitiéndoles comunicarse entre sí de manera eficiente. Independientemente de las tecnologías y protocolos subyacentes utilizados, funciona como una plataforma centralizada que facilita la interoperabilidad, la gestión de mensajes y la integración de aplicaciones.

Arquitectura híbrida en la nube

La arquitectura híbrida en la nube permite a las empresas aprovechar las ventajas de ambos entornos al combinar componentes y servicios de la nube pública y privada. Esta arquitectura utiliza recursos locales y externos para satisfacer las necesidades específicas de la empresa.

La arquitectura híbrida divide las cargas de trabajo entre la infraestructura local y los servicios de la nube. Algunas aplicaciones y datos pueden estar residiendo en servidores locales o centros de datos privados, mientras que otras pueden funcionar en proveedores de servicios en la nube pública como Amazon Web Services (AWS), Microsoft Azure o Google Cloud Platform.

Hay numerosos beneficios de la arquitectura híbrida en la nube:

1. La escalabilidad y la flexibilidad de la nube pública permiten a las organizaciones escalar sus recursos y aprovechar la elasticidad de la nube pública para satisfacer de manera eficiente las demandas fluctuantes.
2. Seguridad y control: permite el mantenimiento de datos y aplicaciones sensibles en entornos locales o privados, donde la organización tiene más control sobre la seguridad y el cumplimiento normativo.
3. Optimización de costos: permite ahorrar dinero utilizando recursos locales para cargas de trabajo estables y recursos de la nube pública para cargas de trabajo variables o de gran escala
4. Integración y colaboración: facilita la integración de sistemas y servicios en entornos locales y en la nube, lo que fomenta la colaboración entre diferentes áreas y proveedores.

2.1.2. Estilos de desarrollo de Software

□ Modelo Vista Controlador (MVC)

En el desarrollo de aplicaciones, especialmente en el desarrollo web, se utiliza ampliamente el patrón de diseño de software conocido como Modelo Vista Controlador (MVC). ofrece una estructura organizada y modular que divide la lógica comercial (modelo), la presentación de datos (vista) y la interacción del usuario. El patrón MVC se compone de tres partes principales:

1. El modelo: La lógica comercial y los datos de la aplicación se muestran en el modelo. El modelo maneja y modifica los datos y aplica la lógica empresarial básica. Esta categoría puede incluir operaciones de almacenamiento y recuperación de datos, validaciones y cálculos.
- 2 Vista (View): Es la representación visual de datos y UI. La responsabilidad de mostrar los datos de manera legible y comprensible para el usuario recae en la vista. Puede ser una página web, una ventana de aplicación u otra parte de la interfaz de usuario. La vista no produce lógica comercial; simplemente muestra información.
- 3 Controlador (Controller): Es el intermediario entre el modelo y la vista. El controlador traduce las interacciones del usuario de la vista en acciones que afectan el modelo. Además, está a cargo de mantener la vista actualizada en función de los cambios que se produzcan en el modelo. Actúa como punto de entrada para las solicitudes del usuario y organiza la interacción entre la vista y el modelo.

2.1.3. SCRUM

Scrum es un marco de trabajo ágil que se utiliza con frecuencia en el desarrollo de software y otros proyectos complejos. El trabajo en equipo, la adaptabilidad y la producción continua de resultados de alta calidad son sus prioridades. Scrum se basa en los principios de transparencia, inspección y adaptación para permitir la entrega rápida y efectiva de productos.

El trabajo de Scrum se divide en "sprints", que son iteraciones. Cada sprint tiene un período de tiempo predeterminado, generalmente de dos a cuatro semanas, durante el cual se desarrolla y entrega un conjunto de funcionalidades o mejoras del producto. Cada sprint comienza con una reunión de planificación del equipo de desarrollo para determinar las tareas a realizar y establecer los objetivos del sprint.

El equipo se organiza en torno a un conjunto de roles clave durante el sprint. Estos trabajos incluyen:

1. Product Owner: Los propietarios del producto establecen los requisitos y prioridades del producto y representan los intereses de los stakeholders. Trabaja en estrecha colaboración con el equipo de desarrollo para garantizar la entrega de valor.
2. Scrum Master: El Scrum Master es responsable de ayudar al equipo a trabajar y asegurarse de que se sigan los principios y prácticas de Scrum. Ayuda a superar las dificultades y fomenta un entorno de trabajo en el que las personas trabajan juntas para lograr resultados.

3. Equipo de Desarrollo: Son los profesionales que deben realizar las tareas necesarias para entregar el incremento del producto al final de cada sprint.

Durante el sprint, se llevan a cabo reuniones diarias cortas y enfocadas llamadas "Scrums diarios" para sincronizar el trabajo y abordar cualquier problema o obstáculo. Al final del sprint, se realiza una reunión de revisión del sprint para mostrar el trabajo completado y recibir comentarios de los interesados. Luego, se realiza una reunión de revisión del sprint para evaluar lo que funcionó bien y qué se puede mejorar en el sprint siguiente.

Los componentes clave de Scrum son la adaptabilidad y la flexibilidad. El proceso y el producto se ajustan y mejoran continuamente a medida que se aprende y se recibe retroalimentación. Scrum se basa en la transparencia, lo que significa que todos los miembros del equipo y los interesados tienen acceso a todos los detalles del trabajo y su progreso. Esto hace más fácil tomar decisiones informadas y permite que todos estén unidos en torno a los objetivos del proyecto.

2.1.4. Herramientas de trabajo.

| LOGO | NOMBRE | DESCRIPCIÓN |
|---|----------------------------|--|
|  | Adobe Photoshop | Photoshop es un potente programa de edición de imágenes que le permite retocar, manipular y crear gráficos digitales. Es popular entre los |

profesionales del diseño gráfico, la fotografía y los medios digitales debido a sus capacidades avanzadas para modificar los colores, aplicar filtros, recortar imágenes y realizar correcciones detalladas.



JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente en el desarrollo web. Es un lenguaje de programación sofisticado que crea sitios web interactivos y dinámicos.



PHP

PHP es un lenguaje de programación popular que se utiliza con frecuencia para crear aplicaciones web. Es un lenguaje de scripting que se ejecuta en el servidor y genera contenido dinámico que se envía al navegador del usuario.



MySql

MySQL es un sistema de gestión de bases de datos relacionales de código abierto popular y bien conocido. Es un programa de software que permite almacenar, organizar y administrar de manera eficiente una gran cantidad de datos. MySQL, que se basa en SQL, ofrece una amplia gama de funciones y características para la gestión y manipulación de datos.



Bootstrap 3

Bootstrap

Bootstrap es un marco de diseño front-end muy utilizado que se utiliza para crear sitios web y aplicaciones web de manera rápida y efectiva. Proporciona una variedad de herramientas, componentes y estilos predefinidos que facilitan la creación de interfaces modernas y responsivas.



Azure

Devops

Azure DevOps es una plataforma completa de Microsoft para la colaboración y el desarrollo de

software basada en la nube. Los equipos de desarrollo pueden planificar, crear, probar y entregar software de manera eficiente gracias a su conjunto de herramientas y servicios.



Postman

Una herramienta popular para probar y documentar APIs es Postman. Los desarrolladores pueden usar una API para enviar solicitudes HTTP y HTTPS y recibir y analizar las respuestas. Con Postman, es rápido y sencillo realizar pruebas de integración, depuración y validación de API.

Tabla 8: Herramientas de trabajo

2.1.5. Configuración de las plataformas Zenodo y GitHub.

El presente proyecto emplea tecnologías estables que se utilizan con frecuencia en entornos de desarrollo escalables. Basándose en esto, se establecieron las siguientes partes que conforman la arquitectura global:

- Aplicación Orientada a Servicios.
- Aplicación Nativa.
- Repositorios OSF con OAuth Token y acceso a propiedades de desarrollar.
- Clúster en Azure Devops, Docker y otros servicios.
- Google Cloud, nube privada que será usada para almacenar los datos y componentes producto de la investigación.
- Contener Docker.
- Repositorios en GitHub con Webhooks, Action, Access Token y otras configuraciones para establecer la capa en la nube.
- Nubes Públicas.

2.1.5.1. Configuración del repositorio Zenodo.

2.1.5.1.1 Configuraciones para desarrollador en Zenodo

En la Fig. 12, se muestra la creación del token de acceso, este es necesario para acceder a la API REST de Zenodo. Una cadena de caracteres creada por Zenodo se conoce como token de acceso y se utiliza como credencial para autenticar las solicitudes a la API. Es fundamental tener en cuenta que los tokens de acceso son confidenciales y deben manipularse con precaución.

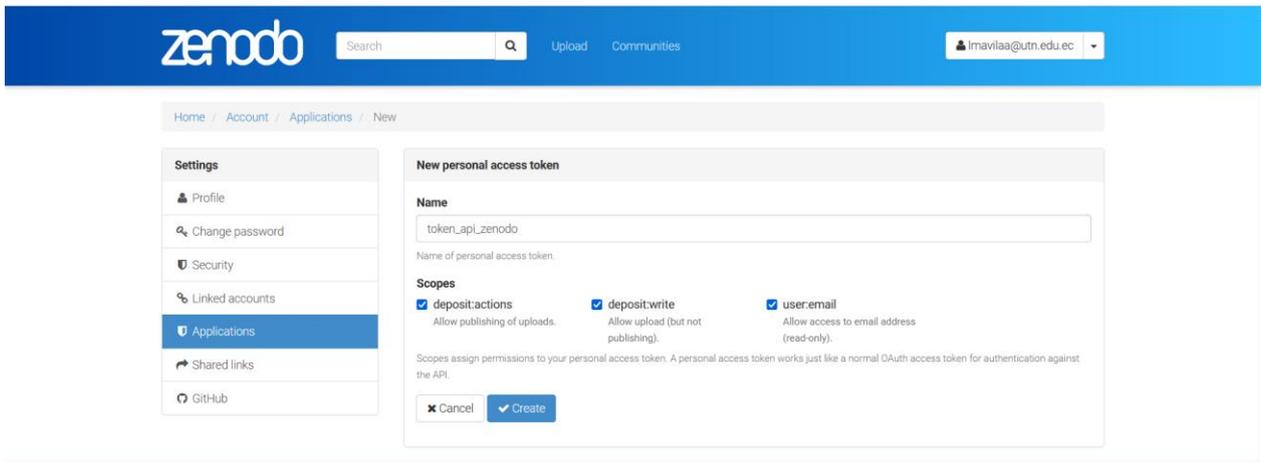


Figura 10: Personal Access Token Zenodo

En la Fig. 13, se muestra la configuración para desarrolladores que establece OAuth Apps y Personal Access Tokens. Los permisos que le fueron asignados al token son de lectura y escritura en organizaciones, repositorios, configuraciones de perfil y otros

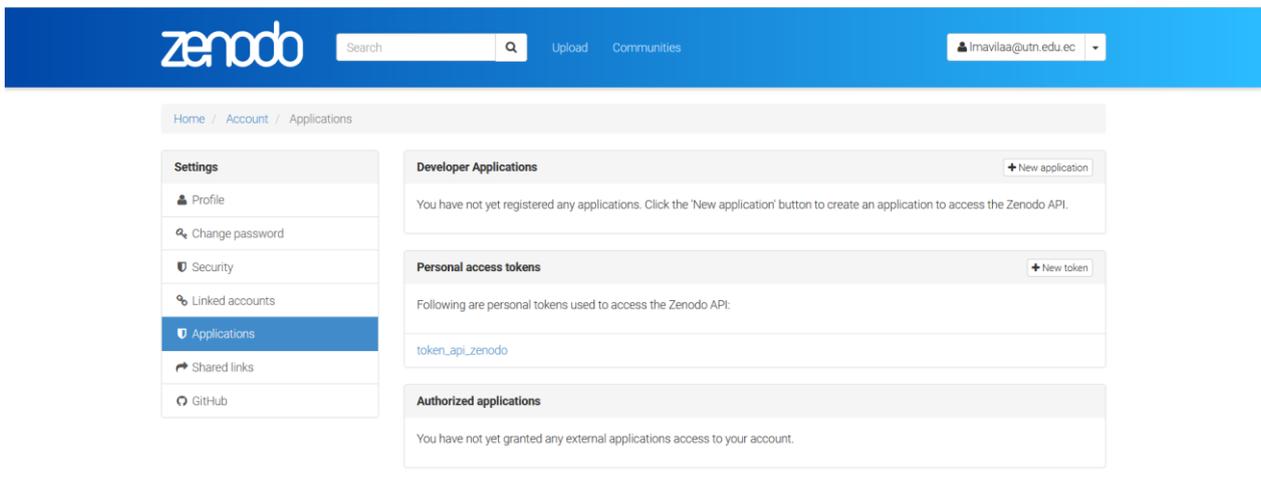


Figura 11: Configuración de Personal Access Token

2.1.5.1.2. Configuraciones para desarrollador en GitHub

Se configuró OAuth Apps para administrar el acceso a GitHub desde las aplicaciones. Estas aplicaciones proporcionan un ID que contiene un identificador de cliente secreto y un identificador único de GitHub. Con la ayuda de una aplicación OAuth llamada "Integración Zenodo & GitHub", los datos obtenidos a través de la autenticación de GitHub permitirán sincronizar los componentes de la arquitectura y los aplicativos desplegados en producción, lo que permitirá un tráfico de datos seguro y eficiente. Ver Fig. 14.

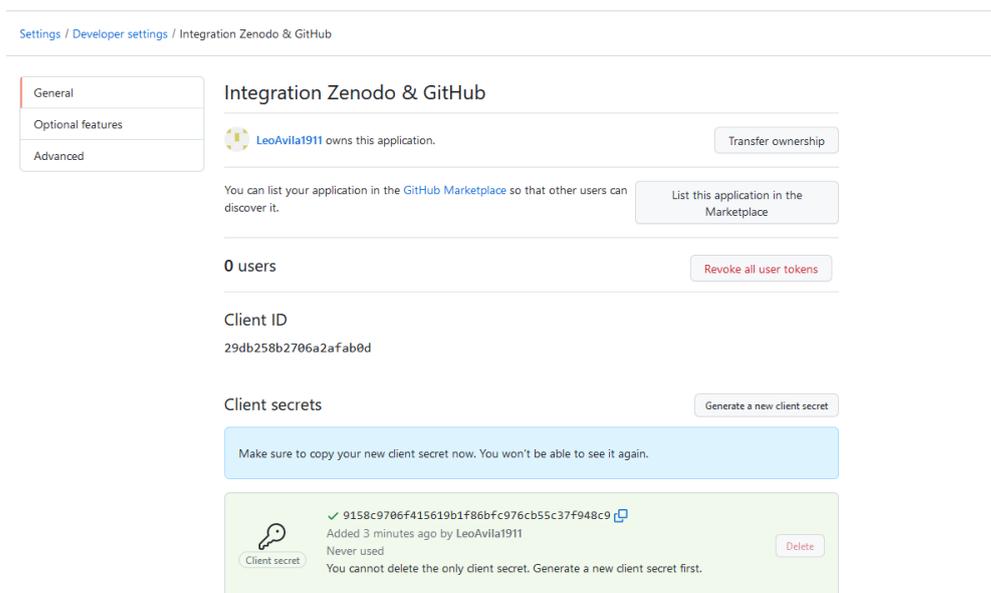


Figura 12: Configuración OAuth Apps en GitHub.

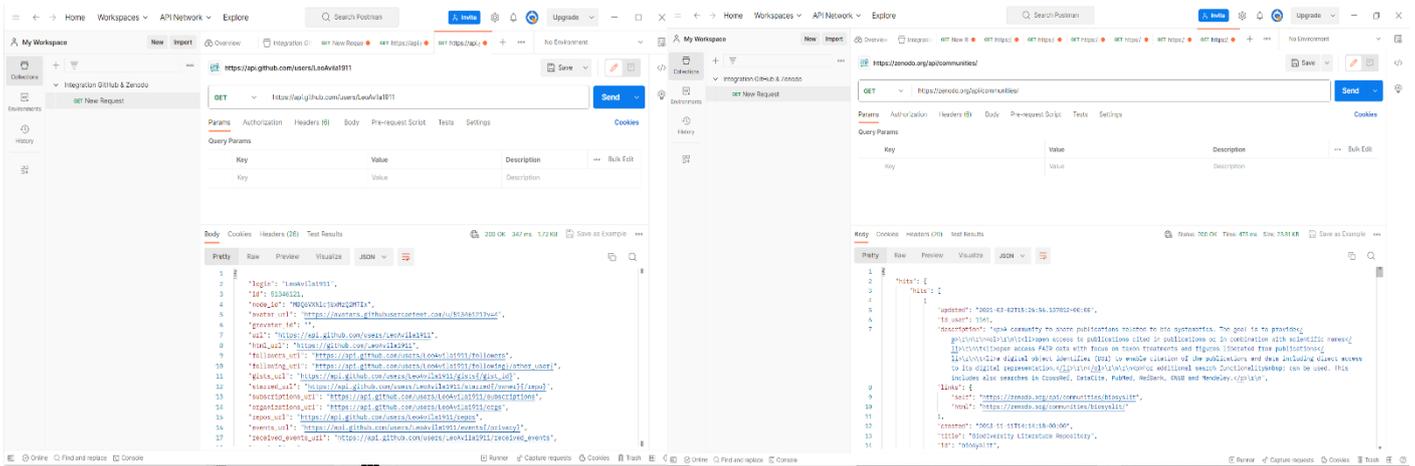


Figura 13: Consumo API Rest de Zenodo y GitHub mediante Postman

2.2. Desarrollo

2.2.1 Sprint 0

En la presente investigación, se utilizará la plataforma Zenodo para desplegar todos los contenidos que se produzcan durante toda la etapa de desarrollo. Estos componentes, que incluyen técnicas, tecnologías, lenguajes de programación, etc., serán limitados al principio debido a que se encuentran en etapa de desarrollo y se utilizarán durante todo el proceso de implementación de la arquitectura orientada a servicios. Como lo aconseja el marco Zenodo, los componentes como el código fuente, los lenguajes de programación, las librerías y las técnicas de desarrollo se organizarán en estructuras organizadas.

El análisis y el desarrollo de la arquitectura para publicar contenido Open Science requieren llevar a cabo tareas. Para garantizar que estas tareas se lleven a cabo correctamente, se empleó la metodología Scrum y el uso de herramientas que se describen en la Tabla 9.

2.2.1.1 Definición Roles Scrum.

| Nombre | ROL | DESCRIPCIÓN |
|-------------------------|------------------------------------|--|
| Product Owner | Phd. Guevara | Cathy Dar seguimiento a que las funcionalidades sean correctamente implementadas en el aplicativo. |
| Scrum Master | Phd. Guevara | Cathy Persona que tiene la facultad de entender y dar las pautas a poner en práctica en las reglas, puntos e hitos del marco de trabajo Scrum, para que de tal forma el producto sea desarrollado de la mejor manera. |
| Deveploment team | Leonardo Michael Avila Armas | Un equipo de desarrollo, también conocido como equipo de desarrollo de software o equipo de ingeniería, es un grupo de personas que trabajan para crear, desarrollar y entregar soluciones de software. |

Tabla 9: Roles de Scrum

2.2.1.1 Historias Épicas.

Las historias épicas en Scrum son unidades de trabajo de gran envergadura que encapsulan funcionalidades o requisitos importantes para un proyecto. Representan objetivos extensos y generalmente se dividen en Historias de Usuario más pequeñas para su implementación en sprints individuales.

Las historias épicas describen características o funcionalidades de alto nivel desde el punto de vista del usuario o del cliente. Son concisos y se enfocan en el valor y los resultados esperados. Su objetivo principal es ayudar a comprender y transmitir los requisitos fundamentales del proyecto.

| Código | Título | Prioridad |
|-----------------|---|------------------|
| E-PROD-1 | Desarrollar aplicación orientada a servicios para publicar contenido Open Science | Alta |
| E-PROD-2 | Integrar las plataformas Zenodo y GitHub en el aplicativo | Alta |

Tabla 10: Historias Épicas

2.2.1.3. Producto Backlog

Las historias épicas, que se definieron anteriormente, se dividieron en varias historias de usuario para que pudieran cumplir con los requisitos funcionales y no funcionales, y se establecieron prioridades y pesos. Para construir los componentes de la arquitectura, la Tabla 10 muestra dos historias épicas que deben ser combinadas con historias de usuario.

Al ser una metodología ágil, Scrum permite optimizar los tiempos de mejor manera. Como resultado, se establecieron alrededor de tres semanas para cada Sprint, lo que resultó en el Product Backlog que se muestra en la Tabla 11.

Historias de Usuario

| H.Épica | Código | Título | Peso | Prioridad |
|-----------------|--------|---|------|-----------|
| E-PROD-1 | PB.01 | Preparar el ambiente de trabajo Zenodo y GitHub | 1 | Alta |
| E-PROD-1 | PB.02 | Desarrollar el modelo de base de datos | 2 | Alta |
| E-PROD-1 | PB.03 | Diseño de las interfaces de usuario en Balsamiq. | 2 | Alta |
| E-PROD-1 | PB.04 | Diseño de la arquitectura orientada a servicios. | 2 | Alta |
| E-PROD-1 | PB.05 | Levantar el ambiente de desarrollo en el servidor Azure Devops | 1 | Alta |
| E-PROD-1 | PB.06 | Levantar el ambiente de desarrollo en el servidor Azure Devops con Docker en EC2. | 2 | Alta |
| E-PROD-1 | PB.07 | Implementar las entidades por defecto | 1 | Media |

| | | | | |
|-----------------|-------|---|---|-------|
| 2 | | en el proyecto (migrations, users, etc). | | |
| E-PROD-2 | PB.08 | Implementar las entidades de Login. | 1 | Media |
| E-PROD-2 | PB.09 | Implementar las entidades de permisos y roles del sistema. | 2 | Media |
| E-PROD-2 | PB.10 | Implementar las entidades de los contenidos abiertos | 2 | Media |
| E-PROD-2 | PB.11 | Implementar la entidad categorías. | 2 | Media |
| E-PROD-2 | PB.12 | Implementar la entidad productos-subir reporte de productos. | 1 | Media |
| E-PROD-2 | PB.13 | Implementar la entidad clientes. | 1 | Media |
| E-PROD-2 | PB.14 | Implementar la entidad ventas-reporte de ventas-administración de ventas. | 1 | Media |
| E-PROD-2 | PB.15 | Implementar la entidad de productos. | 1 | Media |

| | | | | |
|-----------------|-------|--|---|-------|
| E-PROD-2 | PB.16 | Implementar la entidad de stock de productos. | 1 | Media |
| E-PROD-2 | PB.17 | Implementar la entidad de administrador. | 1 | Media |
| E-PROD-2 | PB.18 | Implementar la entidad de vendedor. | 1 | Media |
| E-PROD-2 | PB.19 | Implementar la entidad de almacenamiento de datos Open Science. | 1 | Media |
| E-PROD-2 | PB.20 | Implementar las entidades de relación: contenido Open Science, archivos Open Science, Log del Contenido, Categorías del Contenido Abierto. | 1 | Media |
| E-PROD-2 | PB.21 | Al ser usuario administrador, es posible acceder a todos los componentes, repositorios, documentos de investigación. | 2 | Alta |
| E-PROD-2 | PB.22 | Al ser usuario administrador, se puede gestionar los usuarios, licencias, categorías, niveles, creative commons, | 2 | Alta |

| | | | | |
|-----------------|-------|--|---|------|
| | | publicaciones, etc. | | |
| E-PROD-2 | PB.23 | Al ser usuario vendedor es posible subir reporte de productos en stock en inventario. | 2 | Alta |
| E-PROD-2 | PB.24 | Al ser usuario vendedor es posible subir reporte de ventas a partir de fechas. | 2 | Alta |
| E-PROD-2 | PB.25 | Los datos que se consumen a través del API Backend, Zenodo y GitHub se reflejan automáticamente en Azure Devops. | 1 | Alta |
| E-PROD-3 | PB.26 | La aplicación orientada a servicios es subida al host. | 1 | Alta |
| E-PROD-3 | PB.27 | Integración de las entidades colecciones de Zenodo. | 3 | Alta |
| E-PROD-3 | PB.28 | Gestionar componentes Zenodo y GitHub. | 4 | Alta |

Tabla 11: Product Backlog

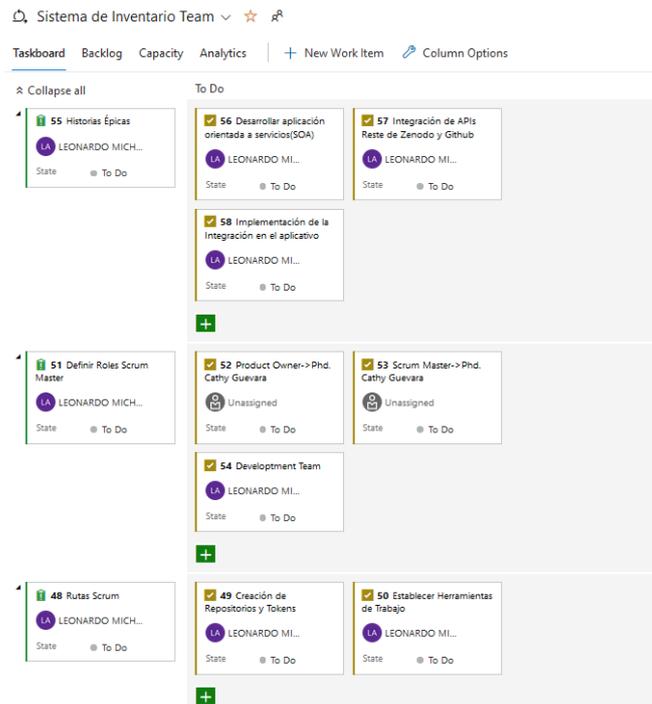


Figura 14: Azure Devops - Sprint 0

2.1.2. Sprint 1

2.2.2.1. Sprint Planning

El objetivo del Sprint número 1 fue iniciar el diseño de la arquitectura y la base de datos de la aplicación. Se establecieron los siguientes objetivos:

- Describir la arquitectura y sus componentes.
- Diseñar la arquitectura global, base de datos, arquitectura aplicación orientada servicios, arquitectura en la nube y capas del sistema.
- Diseñar las interfaces del aplicativo (Balsamiq).
- Levantar ambientes de trabajo.
- Generar tokens de acceso.

2.2.2.2. Sprint Backlog

Para el Sprint número 1, se tomaron en cuenta las historias de usuario descritas en el Anexo número 1, que incluye los criterios de aceptación y las tareas a realizar.

2.2.2.3. Ejecución del Sprint.

Los servicios necesarios para realizar la integración de los repositorios fueron habilitados en cada ambiente del proyecto para Zenodo y GitHub. Se enfatizó la importancia de los principios Open Science, que establecen que toda la información relacionada con un proyecto de investigación debe estar en el repositorio Zenodo, como se muestra en la Fig. 15.

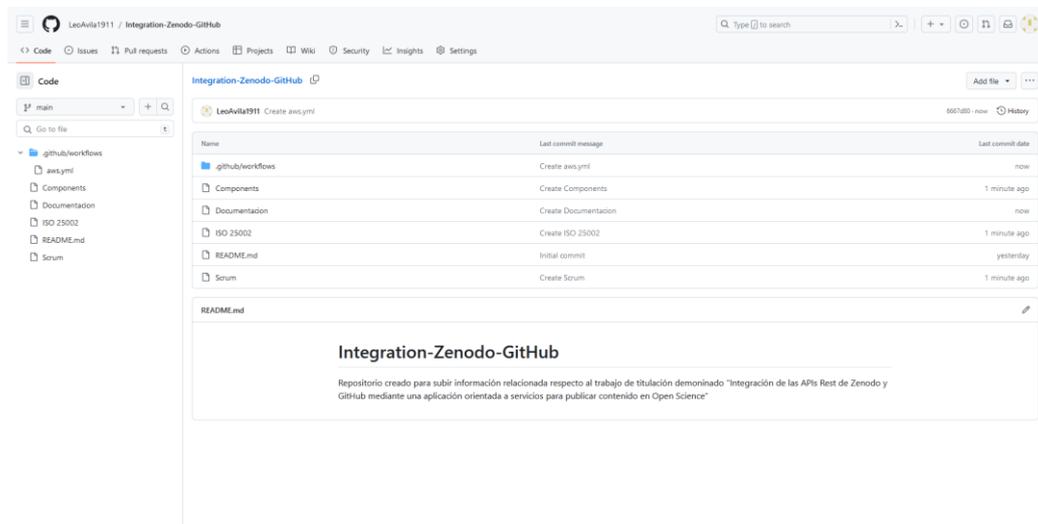


Figura 15: Componentes del proyecto

Aplicación orientada a servicios, módulos.

Los módulos del sistema actual cumplen con una función específica que permite que la aplicación funcione con Zenodo y GitHub. El módulo de gestión de inventario contiene toda la lógica comercial, como gestión de productos, ventas, informes de ventas e inventario, lo que permite reutilizar el código en todas las etapas del desarrollo.

Modelo de bases de datos.

En el diseño físico de la base de datos se realizó un diagrama de entidad relación, se aplicó la normalización necesaria. En la Fig. 16 se muestra el modelo básico que se generó al crear el proyecto orientado a servicios.

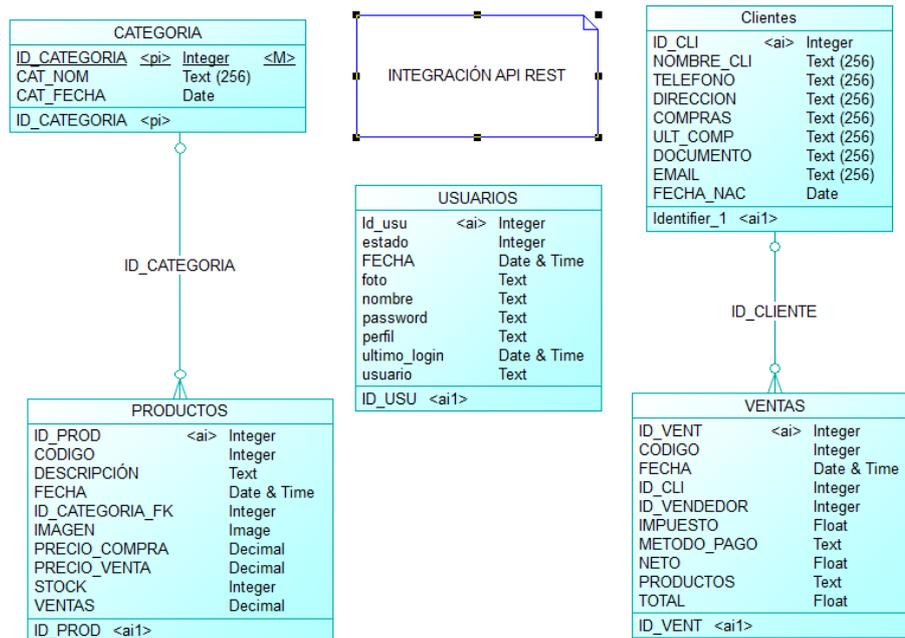


Figura 16: Diagrama de Base de Datos, versión 1

Aplicación Orientada a servicios, mapa de procesos

El mapa de procesos es una herramienta visual que muestra gráficamente los diversos pasos, actividades y flujos de trabajo involucrados en un proceso o sistema. Ver Fig. 17

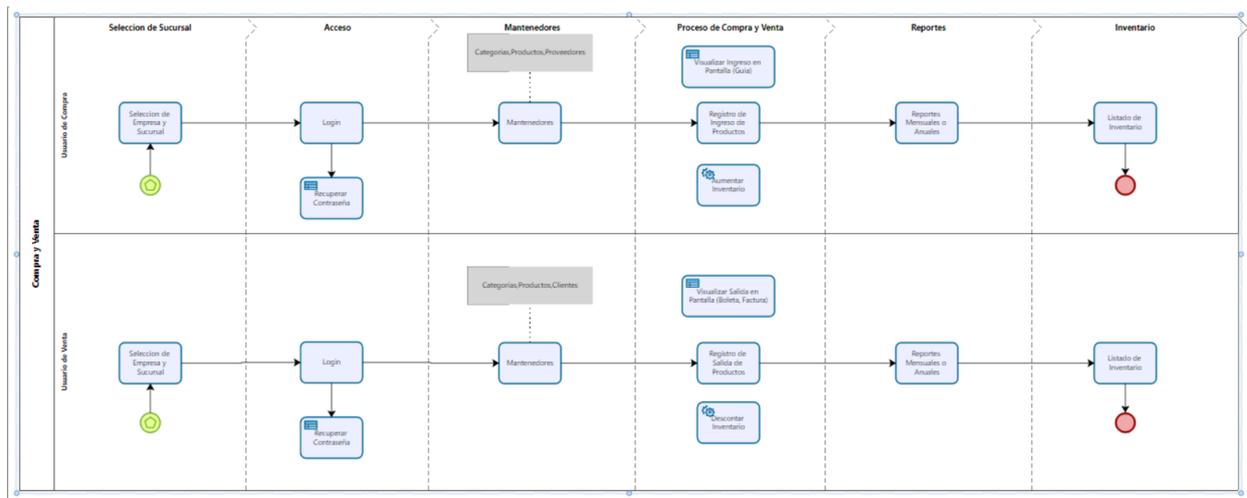


Figura 17: Mapa de Procesos Bizagi Modeler.

Interfaces de usuario – Balsamiq

El objetivo de la maquetación actual era crear interfaces de usuario contemporáneas y fáciles de usar, por lo que se emplearon Sass (Syntactically Awesome Style Sheets) y elementos responsivos. Estos elementos, al ser ejecutados directamente en scripts que interpreta el navegador, son rápidos y fáciles de usar.

Se desarrolló un esquema lógico de navegación después de definir las tecnologías frontend, ya que esto permitirá desarrollar las interfaces de manera ordenada y mantener una estructura global de la aplicación.

Para la presente investigación se utilizó Balsamiq que es una herramienta útil para crear prototipos rápidos de interfaces de usuario, facilitando la comunicación y la colaboración en las etapas iniciales del diseño. Permite centrarse en la funcionalidad y estructura de las interfaces, ahorró tiempo y costos, y obtener valiosos comentarios y retroalimentación antes de comenzar con el desarrollo completo, para que el usuario tenga una mejor interacción con el software, fueron definidas varias capaz, entre las más destacadas son:

- **Home:** Es la pantalla principal en donde el usuario deberá ingresar sus credenciales de acceso dependiendo del rol que se le haya asignado, Ver Fig. 18.



Figura 18: Diseño Web UI, Login

- **Menú:** Es la zona más esencial del aplicativo, ya que se manejará toda la lógica de acceso a los diferentes módulos del sistema.
- **Zona de información:** Es la zona donde se administrará los productos y el reporte de ventas, ver Fig. 19.

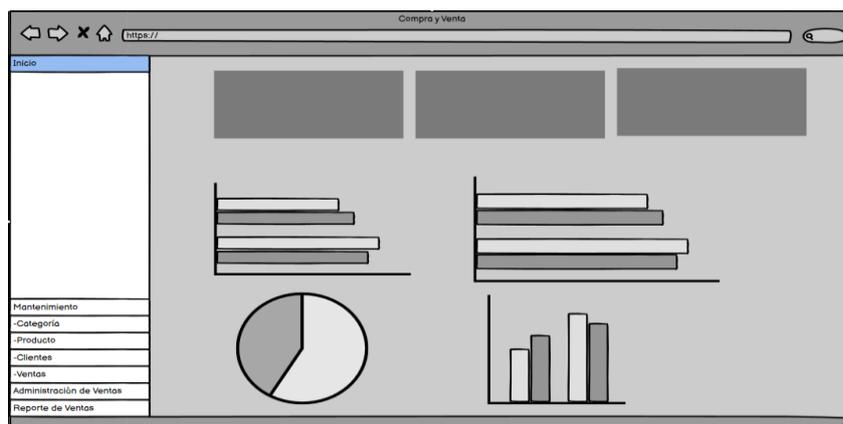


Figura 19: Diseño Web UI, Zona información.

Por lo tanto, se crearon otras imágenes y diseños del producto final. Es importante destacar que una de las principales ventajas de la herramienta Balsamiq es que ofrece una zona de prueba para un entorno de simulación, lo que permite visualizar cómo se verían las interfaces después de su desarrollo. Se realizaron las pruebas necesarias de UI, incluyendo:

- El sitio web será responsivo.
- El sitio web contiene contenido relacionado con cada una de las áreas identificadas en el diagrama de navegación.
- El sitio será frecuentemente sometido a pruebas de rendimiento, lo que aumentará la velocidad y la tasa de respuesta.

Diseño de la Arquitectura del Aplicativo

El diseño de la arquitectura incluye varias capas que contendrán toda la lógica del aplicativo. Estos incluyen:

- **Capa del negocio:** La lógica backend de la capa de negocios se alojará en AWS. Este servidor permitirá conectar datos y usar servicios de otras plataformas como Google Cloud, GitHub y otras.
- **Capa de datos:** El servidor de la base de datos relacional MySQL se encuentra en la capa de datos y tiene las seguridades pertinentes.
- **Capa de Seguridad:** La capa de seguridad contiene las métricas de seguridad que protegen la integridad del sistema y los componentes de acceso como JWT y Access Token con privilegios de lectura/escritura.

- **Capa de Integración API Zenodo y GitHub:** La capa de integración de API Zenodo y GitHub permite la conexión de las dos API REST y la gestión del contenido Open Science en la aplicación.
- **Capa de Cliente:** La capa del cliente es donde los aplicativos del backend y frontend se comunicarán entre sí. En esta capa se desarrollarán los "endpoints", lo que aumentará el uso de servicios de otras plataformas.

Arquitectura SOA desarrollo capa del cliente, ver Fig. 20.

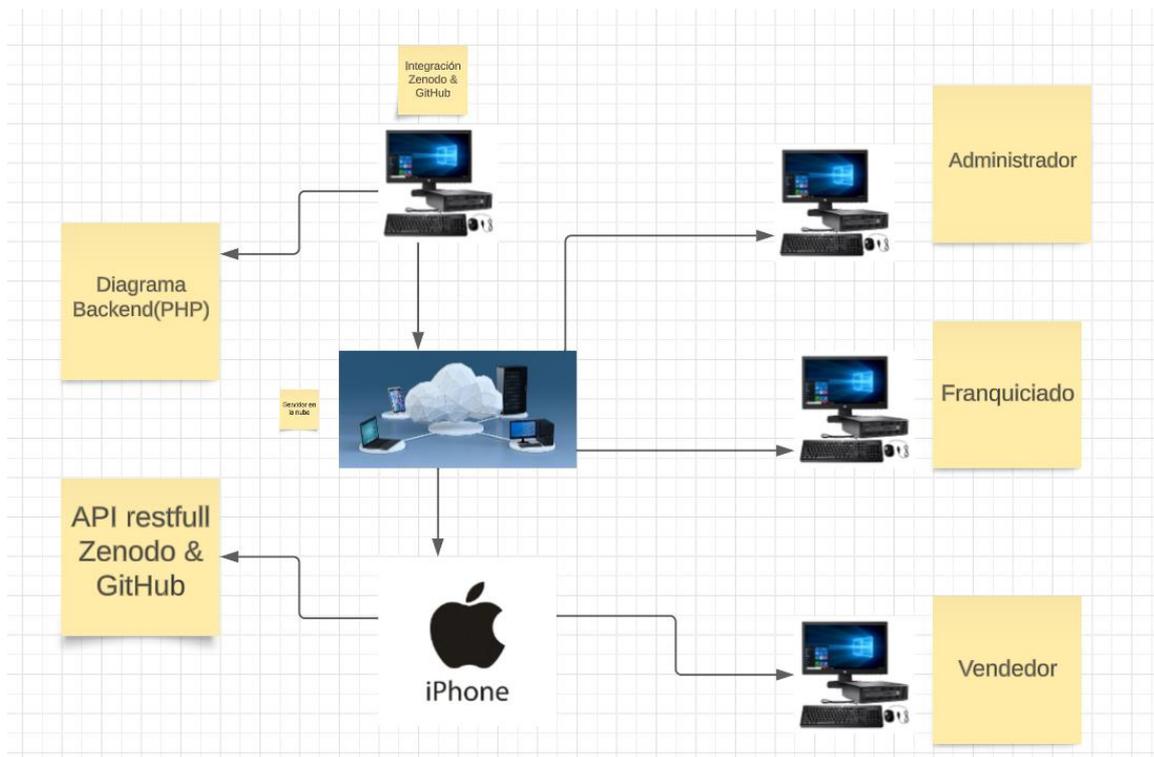


Figura 20: Capa del Cliente

Despliegue en producción del aplicativo de arquitectura SOA

La Fig. 21 muestra los resultados de los flujos de trabajo iniciales, en los que se puede ver que la aplicación en su versión de prueba fue lanzada de manera exitosa

y sin errores a la producción. Es importante destacar que el ambiente controlado dejará una imagen de registro en el clúster AWS, que ocupará tareas, balanceo de carga y un contenedor.

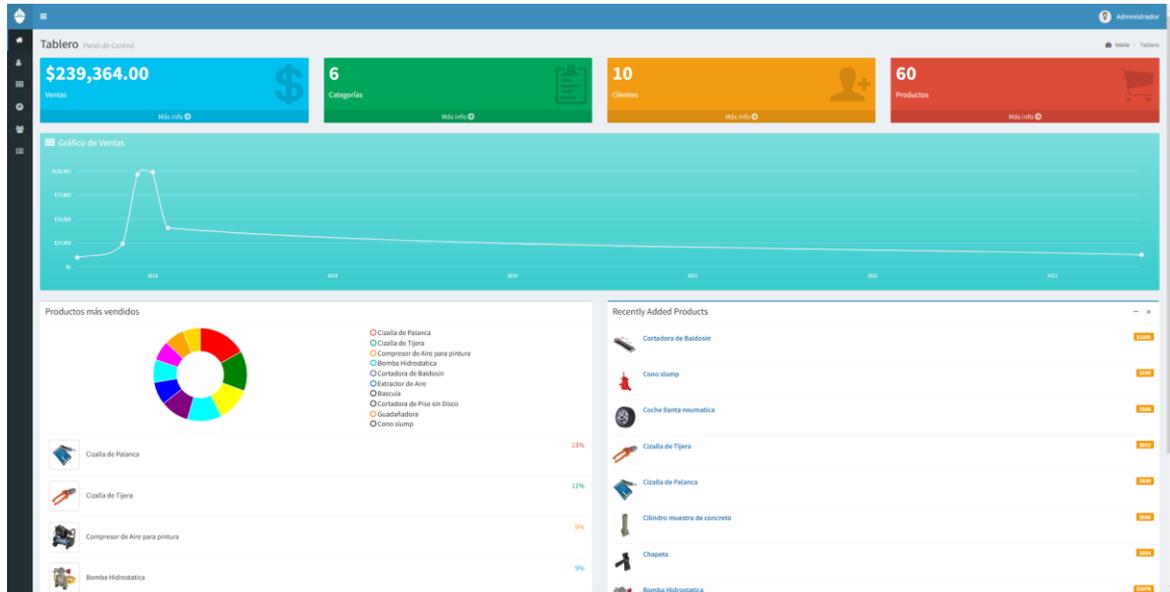


Figura 21: Despliegue en producción aplicativo SOA, versión inicial.

2.2.2.4. Sprint Review

Las historias de usuario llevadas a cabo se detallan en la tabla siguiente.

| Historias de Usuario | | | | |
|----------------------|--------|---|------|-----------|
| H.Épica | Código | Título | Peso | Prioridad |
| E-PROD-1 | PB.01 | Preparar el ambiente de trabajo Zenodo y GitHub | 1 | Alta |
| E-PROD-1 | PB.02 | Desarrollar el modelo de base de datos | 2 | Alta |
| E-PROD-1 | PB.03 | Diseño de las interfaces de usuario en Balsamiq. | 2 | Alta |
| E-PROD-1 | PB.04 | Diseño de la arquitectura orientada a servicios. | 2 | Alta |
| E-PROD-1 | PB.05 | Levantar el ambiente de desarrollo en el servidor Azure Devops | 1 | Alta |
| E-PROD-1 | PB.06 | Levantar el ambiente de desarrollo en el servidor Azure Devops con Docker en EC2. | 2 | Alta |

Tabla 12: Sprint Review 1

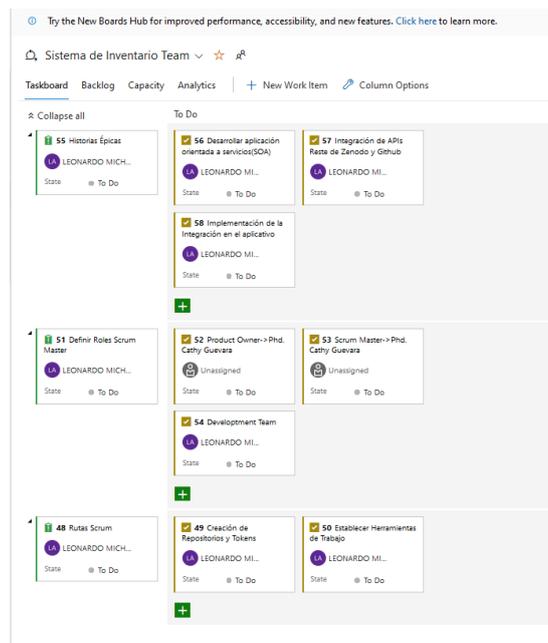


Figura 22: Sprint 1-Azure Devops

2.2.2.5. Sprint Retrospective

¿Qué salió bien en el sprint?

Los ambientes de trabajo, las instancias del servidor y otros servicios iniciales se instalaron correctamente, y los cambios en todos los componentes y paquetes se sincronizaron automáticamente en el repositorio Zenodo. Los usuarios que ingresen al front-end tendrán acceso a interfaces de usuarios rápidas y responsive, y las herramientas de pruebas de rendimiento se han utilizado para resolver los errores que podrían surgir al desplegar todos los componentes en producción.

¿Qué se aprendió del sprint para mejorar el proyecto?

Será necesario implementar gradualmente los paquetes necesarios y precautelar las versiones de cada una de las librerías para mejorar el proyecto. Las interfaces de usuario se desarrollarán gradualmente con los componentes necesarios para que el sitio ofrezca una experiencia de usuario con interfaces de usuario profesionales y eficientes.

De la misma manera el host a utilizar permitió realizar cambios, permisos y otras tareas que harán que la aplicación sea escalable, segura y eficiente mediante el uso de ficheros de contenedor, las herramientas de Integración y Distribución Continua utilizadas por GitHub brindan una producción estable y con paquetes esenciales. Es importante destacar que se realizaron configuraciones en Docker para que el aplicativo React se sincronice automáticamente con el servidor y se despliegue como una dependencia.

2.1.3. Sprint 2

2.2.3.1 Sprint Planning

Los propósitos a cumplir en el Sprint 2 fueron los siguientes:

- Crear los servicios para todas las entidades de la base de datos especificadas en Sprint anterior.
- Valide los puntos finales de comunicación "endpoints" y cree las seguridades necesarias con JWT.
- Construir las estructuras front-end y de navegación de la aplicación.
- Crear políticas de acceso para mantener la integridad de los módulos para todas las rutas del sistema.
- Configure las reglas de acceso para los ficheros JSON en los contenedores de almacenamiento S3.

2.2.3.2. Sprint Backlog

Las historias de usuario descritas en el Anexo Número 3 se tomaron en cuenta en el Sprint Número 2. Para lograrlo, se establecieron estándares de aceptación y responsabilidades a cumplir.

2.2.3.3. Ejecución del Sprint

En el Sprint actual se realizaron principalmente tareas de ingeniería, por lo que se ha creado la documentación requerida en el Anexo No. 4. En el apartado mencionado se incluyen los procedimientos, técnicas, conexiones de datos, servicios instalados y la forma en que se crearon los datos de siembra para cada entidad de la base de datos. Es importante mencionar que se crearon entidades

como colecciones para almacenar los datos que se consumen directamente de COS. La base de datos que se implementará se muestra en la Fig. 25.

Para completar el Sprint número 2, fue necesario comenzar a desarrollar el backend y frontend de la aplicación, que incluyen toda la lógica comercial. En el backend se están creando los "puntos finales" o puntos finales de conexión necesarios, junto con las entidades y seguridades adecuadas, teniendo en cuenta las tareas siguientes:

- Implementar las entidades descritas en las historias de usuario.
- Establecer los controladores.
- Establecer las seguridades de cada controlador.
- Configurar los servicios interconectados en la capa de la base de datos.
- Desarrollar los modelos requeridos.
- Una gran cantidad de datos de prueba se mueven.
- Utilice el software Postman para evaluar la funcionalidad de cada uno de los "endpoint".

Mapa de Procesos Frontend

Se automatizaron procesos en sus primeras etapas durante la ejecución del Sprint actual y se obtuvo una imagen clara del alcance general de la aplicación. La creación del mapa de procesos del frontend y la secuencia de almacenamiento de datos se muestran en la Fig. 23.

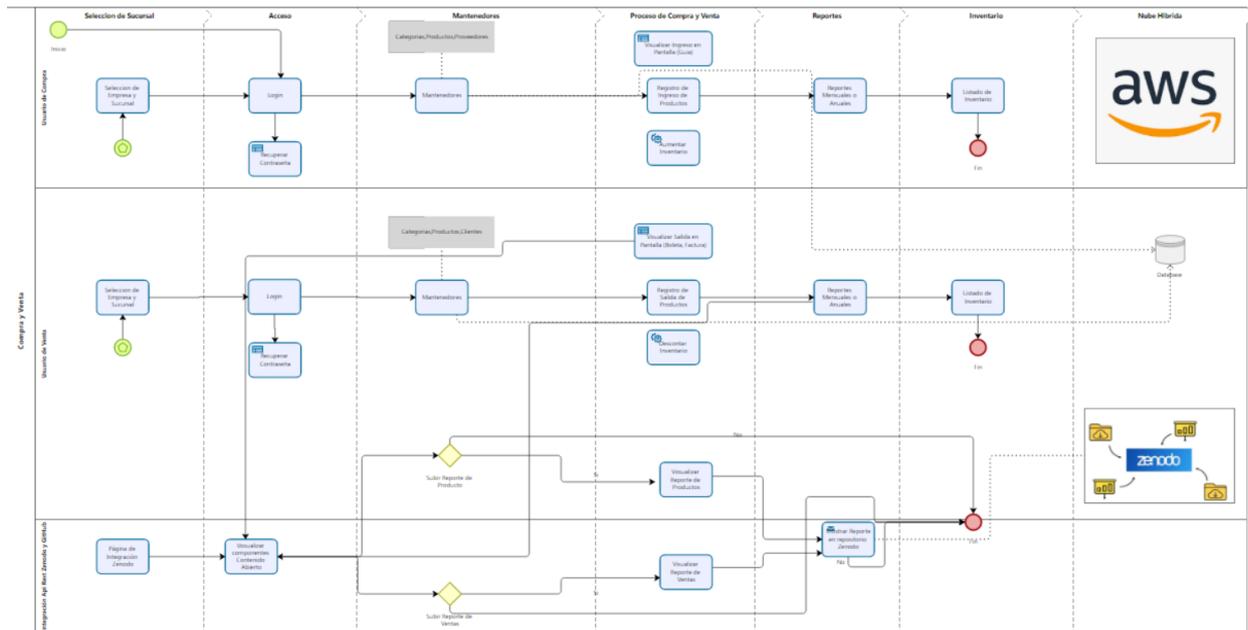


Figura 23: Mapa de Procesos Global del Aplicativo

Menú Principal-Módulos Sistema Inventario

En la página inicial se podrá observar los módulos integrados al aplicativo SOA, tal como se visualiza en la Fig. 24.

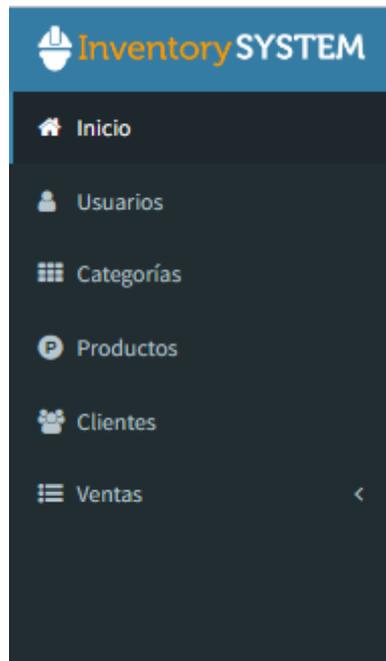
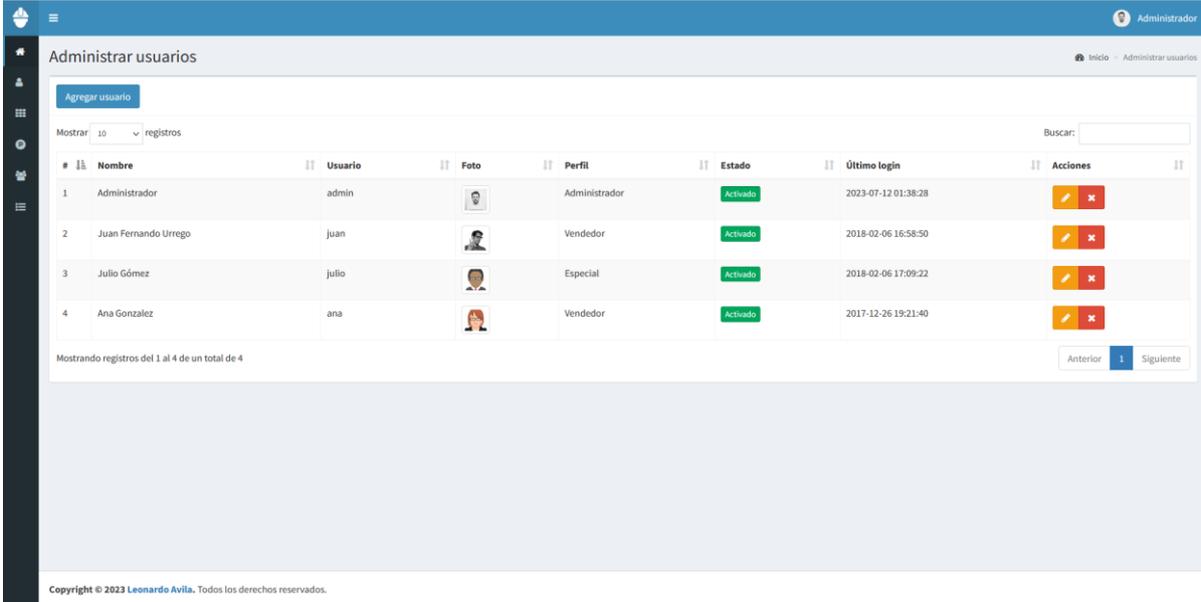


Figura 24: Módulos Sistema

Menú Principal-Usuarios

El primero elemento de la navegación será de usuarios en donde se podrá escoger 3 roles que son: administrador, franquiciado, vendedor, ver Fig. 25.



The screenshot displays a web application interface for managing users. At the top, there is a navigation bar with a home icon, a menu icon, and the user's role 'Administrador'. Below this, the page title is 'Administrar usuarios'. A sidebar on the left contains navigation icons. The main content area features a table with the following columns: '#', 'Nombre', 'Usuario', 'Foto', 'Perfil', 'Estado', 'Último login', and 'Acciones'. The table contains four rows of user data. Below the table, there is a pagination control showing 'Mostrando registros del 1 al 4 de un total de 4' and buttons for 'Anterior', '1', and 'Siguiente'. A footer at the bottom left reads 'Copyright © 2023 Leonardo Avila. Todos los derechos reservados.'

| # | Nombre | Usuario | Foto | Perfil | Estado | Último login | Acciones |
|---|----------------------|---------|------|---------------|----------|---------------------|----------|
| 1 | Administrador | admin | | Administrador | Activado | 2023-07-12 01:38:28 | |
| 2 | Juan Fernando Urrego | juan | | Vendedor | Activado | 2018-02-06 16:58:50 | |
| 3 | Julio Gómez | julio | | Especial | Activado | 2018-02-06 17:09:22 | |
| 4 | Ana Gonzalez | ana | | Vendedor | Activado | 2017-12-26 19:21:40 | |

Figura 25: Menú Principal – Usuarios

Menú Principal-Categorías

El segundo elemento de la navegación será categorías en el usuario podrá realizar acciones CRUD en base a las categorías que desee ingresar, ver Figura 26.

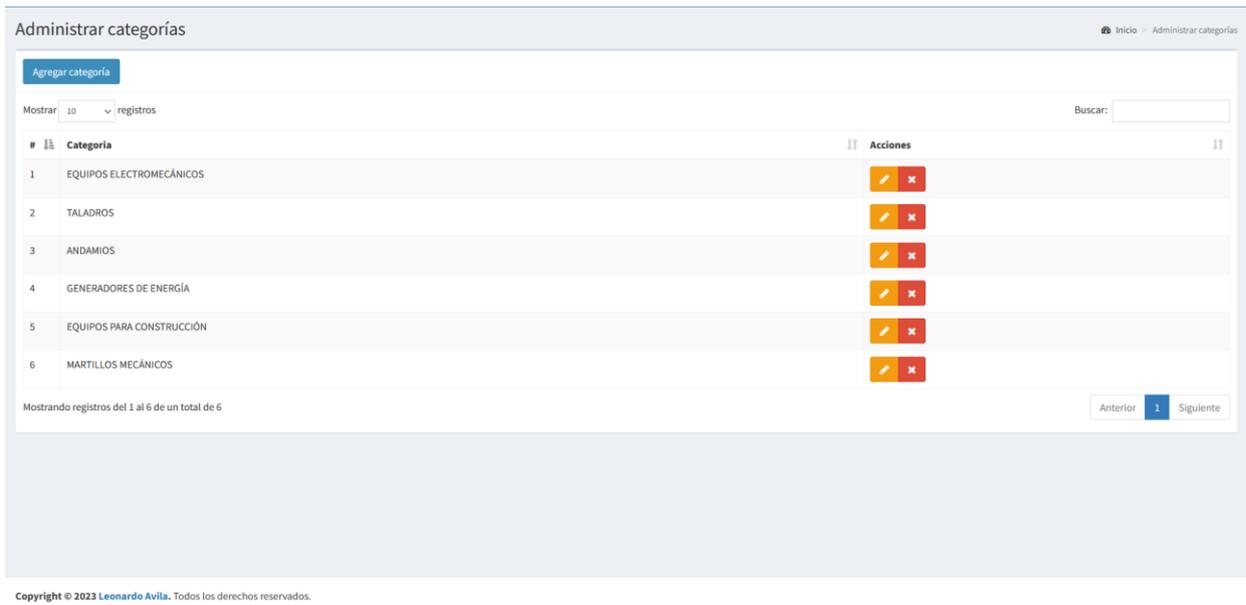


Figura 26: Menú Principal-Categorías

Menú Principal – Productos

El tercer elemento de la navegación es el menú principal de productos en donde se permitirá realizar acciones CRUD además que se visualizará la fecha de insertado el producto., Ver Fig. 27.

| # | Imagen | Código | Descripción | Categoría | Stock | Precio de compra | Precio de venta | Agregado | Acciones |
|----|--------|--------|------------------------------|---------------------------|-------|------------------|-----------------|---------------------|----------|
| 1 | | 517 | Cortadora de Baldosin | Equipos para construcción | 10 | 930 | 1302 | 2023-07-04 20:09:13 | |
| 2 | | 516 | Cono slump | Equipos para construcción | 14 | 140 | 196 | 2023-07-04 20:09:13 | |
| 3 | | 515 | Coche llanta neumatica | Equipos para construcción | 16 | 420 | 588 | 2023-07-04 14:10:20 | |
| 4 | | 514 | Cizalla de Tijera | Equipos para construcción | 19 | 580 | 812 | 2023-07-04 14:10:20 | |
| 5 | | 513 | Cizalla de Palanca | Equipos para construcción | 3 | 450 | 630 | 2017-12-26 19:30:12 | |
| 6 | | 512 | Cilindro muestra de concreto | Equipos para construcción | 16 | 400 | 560 | 2017-12-26 17:27:41 | |
| 7 | | 511 | Chapeta | Equipos para construcción | 16 | 660 | 924 | 2017-12-26 17:27:42 | |
| 8 | | 510 | Bomba Hidrostatica | Equipos para construcción | 8 | 770 | 1078 | 2017-12-26 17:26:51 | |
| 9 | | 509 | Bascula | Equipos para construcción | 12 | 130 | 182 | 2017-12-26 17:26:51 | |
| 10 | | 508 | Tensor | Equipos para construcción | 19 | 100 | 140 | 2017-12-26 17:26:51 | |

Figura 27: Menú Principal – Productos

Menú Principal – Usuarios

El cuarto elemento de la navegación es el menú principal de usuarios en donde se permitirá realizar acciones CRUD además que se visualizará la fecha de insertado el usuario, Ver Fig. 28.

| # | Nombre | Documento ID | Email | Teléfono | Dirección | Fecha nacimiento | Total compras | Última compra | Ingreso al sistema | Acciones |
|----|-------------------|--------------|-----------------------|----------------|----------------------|------------------|---------------|---------------------|---------------------|----------|
| 1 | Gonzalo Pérez | 436346346 | gonzalo@yahoo.com | (235) 346-3464 | Carrera 34 # 56 - 34 | 1967-08-09 | 24 | 2017-12-25 17:24:24 | 2023-07-04 20:09:13 | |
| 2 | David Guzman | 43634643 | david@hotmail.com | (354) 574-5634 | carrera 45 # 45 | 1967-05-04 | 10 | 2017-12-26 17:24:50 | 2017-12-26 17:24:50 | |
| 3 | Ximena Restrepo | 436346346 | ximena@gmail.com | (543) 463-4634 | calle 45 # 23 - 45 | 1956-03-04 | 26 | 2023-07-04 14:10:20 | 2023-07-04 14:10:20 | |
| 4 | Eduardo López | 2147483647 | eduardo@gmail.com | (534) 634-6565 | Carrera 67 # 45sur | 1978-03-04 | 12 | 2017-12-26 17:25:33 | 2017-12-26 17:25:33 | |
| 5 | Stella Jaramillo | 65756735 | stella@gmail.com | (435) 346-3463 | Carrera 34 # 45- 56 | 1956-06-05 | 9 | 2017-12-26 17:25:55 | 2017-12-26 17:25:55 | |
| 6 | Julian Ramirez | 786786545 | julian@hotmail.com | (675) 674-5453 | Carrera 45 # 54 - 56 | 1980-04-05 | 14 | 2017-12-26 17:26:28 | 2017-12-26 17:26:28 | |
| 7 | Margarita Londoño | 34565432 | margarita@hotmail.com | (344) 345-6678 | Calle 45 # 34 - 56 | 1976-11-30 | 14 | 2017-12-26 17:26:51 | 2017-12-26 17:26:51 | |
| 8 | Miguel Murillo | 325235235 | miguel@hotmail.com | (254) 545-3446 | calle 34 # 34 - 23 | 1976-03-04 | 32 | 2017-12-26 17:27:13 | 2017-12-26 23:38:13 | |
| 9 | Pedro Pérez | 2147483647 | pedro@gmail.com | (399) 876-5432 | Calle 34 N33 - 56 | 1970-08-07 | 7 | 2017-12-26 17:27:42 | 2017-12-26 17:27:42 | |
| 10 | Juan Villegas | 2147483647 | juan@hotmail.com | (300) 341-2345 | Calle 23 # 45 - 56 | 1980-11-02 | 7 | 2018-02-06 17:47:02 | 2018-02-06 17:47:02 | |

Figura 28: Menú Principal-Usuarios

Menú Principal-Ventas

El quinto elemento de la navegación es el menú principal de ventas en donde se desplegará tres opciones, las cuáles serán administrar ventas, crear una venta y generar reportes, Ver Fig 29.

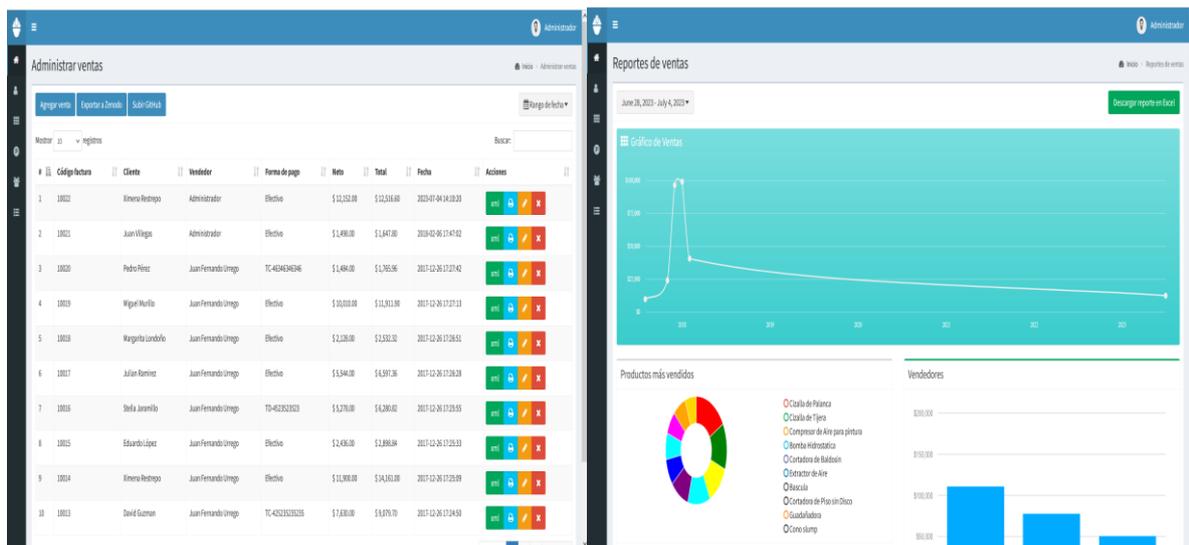


Figura 29: Menú Principal-Ventas

2.2.3.4. Sprint Review.

Las historias de usuario llevadas a cabo se detallan en la tabla siguiente.

| | | | | |
|----------|-------|---|---|-------|
| E-PROD-2 | PB.07 | Implementar las entidades por defecto en el proyecto (migrations, users, etc). | 1 | Media |
| E-PROD-2 | PB.08 | Implementar las entidades de Login . | 1 | Media |
| E-PROD-2 | PB.09 | Implementar las entidades de permisos y roles del sistema. | 2 | Media |
| E-PROD-2 | PB.10 | Implementar las entidades de los contenidos abiertos | 2 | Media |
| E-PROD-2 | PB.11 | Implementar la entidad categorías. | 2 | Media |
| E-PROD-2 | PB.12 | Implementar la entidad productos-subir reporte de productos. | 1 | Media |
| E-PROD-2 | PB.13 | Implementar la entidad clientes. | 1 | Media |
| E-PROD-2 | PB.14 | Implementar la entidad ventas-reporte de ventas-administración de ventas. | 1 | Media |
| E-PROD-2 | PB.15 | Implementar la entidad de productos. | 1 | Media |
| E-PROD-2 | PB.16 | Implementar la entidad de stock de productos. | 1 | Media |
| E-PROD-2 | PB.17 | Implementar la entidad de administrador. | 1 | Media |
| E-PROD-2 | PB.18 | Implementar la entidad de vendedor. | 1 | Media |
| E-PROD-2 | PB.19 | Implementar la entidad de almacenamiento de datos Open Science. | 1 | Media |
| E-PROD-2 | PB.20 | Implementar las entidades de relación: contenido Open Science, archivos Open Science, Log del Contenido, Categorías del | 1 | Media |

Tabla 13: Sprint Review 2

2.2.3.5. Sprint Retrospective.

¿Qué salió bien en el sprint?

Las interfaces del aplicativo se realizaron correctamente, y los reportes de productos y ventas se exporta correctamente a Zenodo. Los usuarios que ingresen al front-end tendrán acceso a interfaces de usuarios rápidas y responsive, y las herramientas de pruebas de rendimiento se han utilizado para resolver los errores que podrían surgir al desplegar todos los componentes en producción.

¿Qué se aprendió del sprint para mejorar el proyecto?

Será necesario implementar gradualmente los paquetes necesarios y precautelar las versiones de cada una de las librerías para mejorar el proyecto. Las interfaces de usuario se desarrollarán gradualmente con los componentes necesarios para que el sitio ofrezca una experiencia de usuario con interfaces de usuario profesionales y eficientes.

2.2.4. Sprint 3

2.2.4.1. Sprint Planning

Los propósitos a cumplir en el Sprint 3 fueron los siguientes:

- Implementar las funcionalidades a detalle de los diferentes módulos que tendrá el aplicativo.
- Generar las rutas de accesos seguras para cada uno de los roles asignados en el sistema.
- Almacenamiento de datos mediante host.

- Implementar a detalle las funcionalidades para cada uno de los roles del sistema.
- Migrar la base de datos a Hostinger.
- Realizar las configuraciones para desplegar el aplicativo en Hostinger.

2.2.4.2. Sprint Review

Las historias de usuario que se describen en el Anexo número 5 fueron consideradas para el Sprint número 3. Para lograrlo, se establecieron estándares de aceptación y responsabilidades a cumplir.

2.2.4.3. Ejecución del Sprint

Las funcionalidades de cada uno de los roles de la aplicación se desarrollaron actualmente en Sprint teniendo en cuenta a los actores principales: el Administrador de la Aplicación, el Administrador Franquiciado y el Vendedor.

El resultado esperado del Sprint número 3 fue la implementación de las funcionalidades de los roles de la aplicación; al hacerlo, se obtuvieron flujos de trabajo en los que cada rol tenía funcionalidades y privilegios específicos dentro de la aplicación.

- Realizar las funciones de administrador en las áreas asignadas utilizando interfaces amigables y responsivas
- Configurar el servicio de ventas para proporcionar detalles sobre algunas tareas establecidas.
- Crear procesos de descarga y reporte de contenido abierto.

- Realizar las funciones de administrador franquiciado en las áreas asignadas utilizando interfaces amigables y responsivas.
- Configurar el servicio de reporte de ventas utilizando BI para proporcionar detalles sobre algunas tareas establecidas.
- Implementar elementos necesarios para la navegación del Backend.
- Implementar una plantilla por defecto para el Backend de la aplicación, para ello se instalarán los componentes necesarios Bootstrap

Flujo de trabajo de administrador

El objetivo del presente apartado fue desarrollar el flujo de trabajo y las herramientas necesarias para lograrlo. La Figura 30 muestra el proceso del administrador en la aplicación después de lo mencionado.

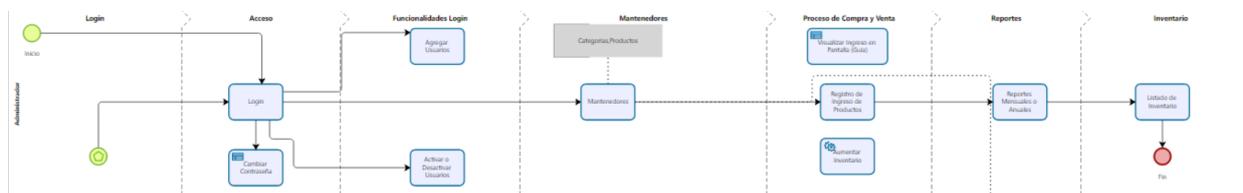


Figura 30: Flujo de trabajo-Administrador

Una vez establecido el proceso se procedió a desarrollar las funcionalidades, es necesario mencionar que una vez un administrador cree una cuenta el podrá elegir el rol que va a tener el usuario (Administrador, Especial y Vendedor) dichos roles tendrán acceso al contenido abierto subido al repositorio de Zenodo.

Una vez realizado las funcionalidades de Categorías, Productos y Ventas, los 3 roles tendrán la posibilidad de visualizar los reportes BI y descargar en formato PDF las facturas realizadas (ver Fig. 31) o también descargar en formato .xls, Ver Fig. 32

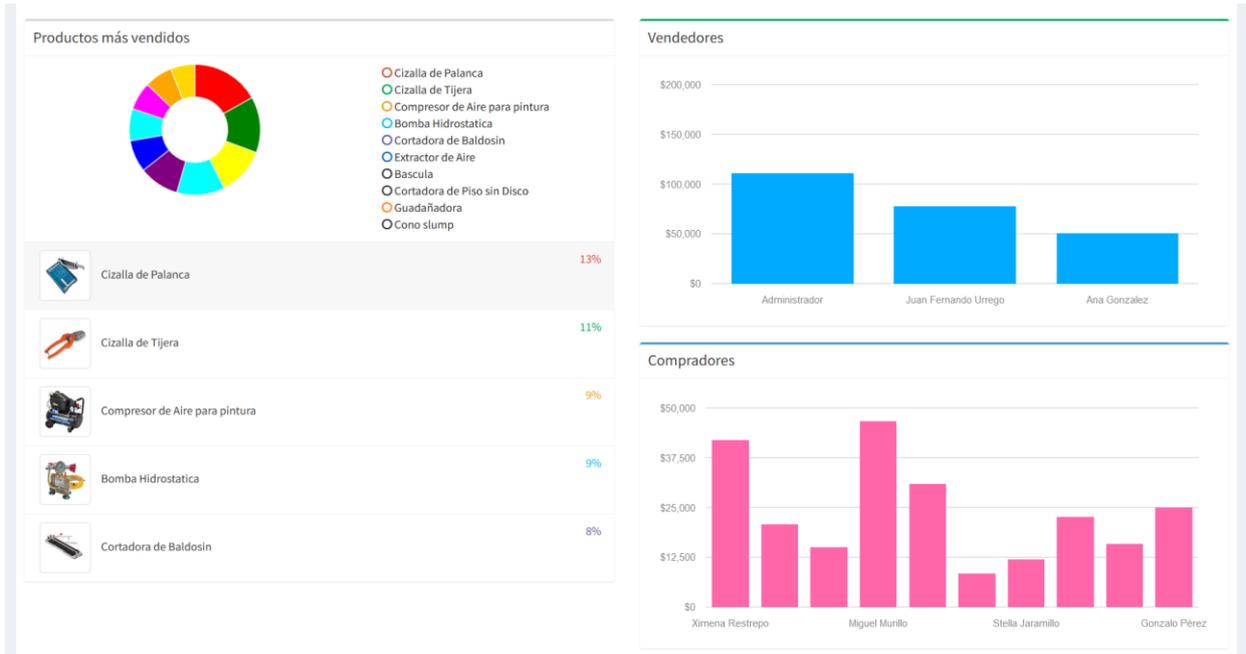


Figura 31: Reportes de ventas – BI

| CÓDIGO | CLIENTE | VENDEDOR | CANTIDAD | PRODUCTOS | IMPUESTO | NETO | TOTAL | METODO DE PAGO | FECHA |
|--------|---------|-------------------|----------------------|------------------------------------|-------------|--------------|--------------|-----------------|------------|
| 1 | | | 3 | Cortadora de Baldosin | | | | | |
| 2 | | | 1 | Cono slump | | | | | |
| 3 | | | 1 | Coche llanta neumatica | | | | | |
| 4 | | | 1 | Cizalla de Tijera | | | | | |
| 5 | | | 1 | Polea 2 canales | | | | | |
| 6 | 10022 | Ximena Restrepo | Administrador | 1. Planta Electrica Diesel 120 Kva | \$ 364.56 | \$ 12,152.00 | \$ 12,516.60 | Efectivo | 4/7/2023 |
| 7 | | | | 1. Cortadora de Baldosin | | | | | |
| 8 | 10021 | Juan Villegas | Administrador | 1. Cono slump | \$ 149.80 | \$ 1,498.00 | \$ 1,647.80 | Efectivo | 6/2/2018 |
| 9 | | | | 1. Cilindro muestra de concreto | | | | | |
| 10 | 10020 | Pedro Pérez | Juan Fernando Urrego | 1. Chapeta | \$ 281.96 | \$ 1,484.00 | \$ 1,765.96 | TC-46346346346 | 26/12/2017 |
| 11 | | | | 15. Cizalla de Palanca | | | | | |
| 12 | 10019 | Miguel Murillo | Juan Fernando Urrego | 1. Cilindro muestra de concreto | \$ 1,901.90 | \$ 10,010.00 | \$ 11,911.90 | Efectivo | 26/12/2017 |
| 13 | | | | 1. Tensor | | | | | |
| 14 | | | | 5. Bascula | | | | | |
| 15 | 10018 | Margarita Londoño | Juan Fernando Urrego | 1. Bomba Hidrostatica | \$ 404.32 | \$ 2,128.00 | \$ 2,532.32 | Efectivo | 26/12/2017 |
| 16 | | | | 4. Cizalla de Tijera | | | | | |
| 17 | | | | 3. Bascula | | | | | |
| 18 | | | | 2. Cilindro muestra de concreto | | | | | |
| 19 | 10017 | Julian Ramirez | Juan Fernando Urrego | 1. Cizalla de Palanca | \$ 1,053.36 | \$ 5,544.00 | \$ 6,597.36 | Efectivo | 26/12/2017 |
| 20 | | | | 1. Coche llanta neumatica | | | | | |
| 21 | | | | 5. Cizalla de Tijera | | | | | |
| 22 | 10016 | Stella Jaramillo | Juan Fernando Urrego | 1. Cizalla de Palanca | \$ 1,002.82 | \$ 5,278.00 | \$ 6,280.82 | TD-4523523523 | 26/12/2017 |
| 23 | 10015 | Eduardo López | Juan Fernando Urrego | 3. Cizalla de Tijera | \$ 462.84 | \$ 2,436.00 | \$ 2,898.84 | Efectivo | 26/12/2017 |
| 24 | | | | 1. Cono slump | | | | | |
| 25 | | | | 1. Chapeta | | | | | |
| 26 | 10014 | Ximena Restrepo | Juan Fernando Urrego | 10. Bomba Hidrostatica | \$ 2,261.00 | \$ 11,900.00 | \$ 14,161.00 | Efectivo | 26/12/2017 |
| 27 | | | | 1. Chapeta | | | | | |
| 28 | | | | 1. Cono slump | | | | | |
| 29 | 10013 | David Guzman | Juan Fernando Urrego | 5. Cortadora de Baldosin | \$ 1,449.70 | \$ 7,630.00 | \$ 9,079.70 | TC-425235235235 | 26/12/2017 |
| 30 | | | | 1. Cono slump | | | | | |
| 31 | | | | 1. Cono slump | | | | | |

Figura 32: Formato .xls de Ventas

El flujo de trabajo Open Science establece 2 posibles estados para cada determinado contenido, por ello se lo ha representado de la siguiente manera:

- Color rojo – Toll Access (Open Access).

- Color verde – Green Road (Closed Access)

Una vez que el usuario ingrese a la opción de exportar a Zenodo, como se muestra en la Figura 37, podrá visualizar todos los contenidos que haya subido. Cada uno de los repositorios subidos tendrá la fecha de subida y la descripción (ya sea reporte de productos o de ventas). El icono verde indica que el contenido ha finalizado el proceso Open Access, lo que significa que está completamente abierto para la copia, reutilización y revisión, Ver Fig. 33.

Administrar productos Inicio > Administrar productos

Agregar producto **Exportar a Zenodo** Agregar Repositorio Cargar Información

Mostrar: 10 registros Buscar:

| # | Imagen | Código | Descripción | Categoría | Stock | Precio de compra | Precio de venta | Agregado | Acciones |
|---|--------|--------|------------------------------|---------------------------|-------|------------------|-----------------|---------------------|----------|
| 1 | | 517 | Cortadora de Baldosin | Equipos para construcción | 10 | 930 | 1302 | 2023-07-04 20:09:13 | |
| 2 | | 516 | Cono slump | Equipos para construcción | 14 | 140 | 196 | 2023-07-04 20:09:13 | |
| 3 | | 515 | Coche llanta neumatica | Equipos para construcción | 16 | 420 | 588 | 2023-07-04 14:10:20 | |
| 4 | | 514 | Cizalla de Tijera | Equipos para construcción | 19 | 580 | 812 | 2023-07-04 14:10:20 | |
| 5 | | 513 | Cizalla de Palanca | Equipos para construcción | 3 | 450 | 630 | 2017-12-26 19:30:12 | |
| 6 | | 512 | Cilindro muestra de concreto | Equipos para construcción | 16 | 400 | 560 | 2017-12-26 17:27:41 | |
| 7 | | 511 | Chapeta | Equipos para construcción | 16 | 660 | 924 | 2017-12-26 17:27:42 | |
| 8 | | 510 | Bomba Hidrostatica | Equipos para construcción | 8 | 770 | 1078 | 2017-12-26 17:26:51 | |
| 9 | | 509 | Bascula | Equipos para construcción | 12 | 130 | 182 | 2017-12-26 17:26:51 | |

Figura 33: Exportación a Zenodo

| |
|---|
|  July 17, 2023 (v1) Dataset Open Access Prueba Integracion <small>Created Jul 17, 2023 3:38:18 AM, modified Jul 17, 2023 3:38:18 AM</small> |
|  July 15, 2023 (v1) Other Open Access ventas 2023-07-15_00-25-51.xls <small>Created Jul 15, 2023 5:25:55 AM, modified Jul 15, 2023 5:25:56 AM</small> |
|  July 15, 2023 (v1) Other Open Access productos 2023-07-14_22-38-16.xls <small>Created Jul 15, 2023 3:38:20 AM, modified Jul 15, 2023 3:38:21 AM</small> |
|  July 14, 2023 (v1) Other Open Access productos 2023-07-14_01-52-11.xls <small>Created Jul 14, 2023 6:52:12 AM, modified Jul 14, 2023 6:52:13 AM</small> |
|  July 14, 2023 (v1) Other Open Access productos 2023-07-14_01-10-13.xls <small>Created Jul 14, 2023 6:10:16 AM, modified Jul 14, 2023 6:10:18 AM</small> |
|  July 14, 2023 (v1) Other Open Access productos 2023-07-13_23-09-11.xls <small>Created Jul 14, 2023 4:09:12 AM, modified Jul 14, 2023 4:09:14 AM</small> |
|  July 8, 2023 (1) Diagram Closed Access Upload testing <small>Created Jul 8, 2023 5:48:13 PM, modified Jul 8, 2023 5:48:42 PM</small> |

Figura 34: Contenido Open Science

2.2.4.4. Sprint Review

| Código | Título | Cumple |
|--------------|--|--------|
| PB.24 | Al ser usuario administrador, es posible acceder a todos las funcionalidades, repositorios y reportes. | SÍ |
| PB.25 | Al ser usuario administrador, se puede gestionar los usuarios, categorías, creative commons, publicaciones, etc. | SÍ |
| PB.26 | Al ser usuario administrador es posible subir contenido Open Science en el | SÍ |

aplicativo inicial.

PB.27 Al ser usuario especial es posible realizar todas las funcionalidades de administrador. SÍ

PB.28 Al ser usuario vendedor es posible crear categorías. SÍ

PB.29 Al ser usuario vendedor es posible crear productos. SÍ

PB.30 Al ser usuario vendedor es posible crear y visualizar reporte de ventas. SÍ

PB.31 Los datos que se consumen a través del API Backend, Zenodo y GitHub se reflejan automáticamente en el contenedor de Hostinger. SÍ

PB.32 La aplicación orientada a servicios es subida al host con la ayuda de GitHub. SÍ

PB.33 Integración de las entidades colecciones de OSF. SÍ

PB.34 Integración de las entidades nodos de OSF. SÍ

PB.35 Integración de las entidades repositorios de Sí Zenodo.

PB.24 Integración de las entidades citaciones, Sí preimpresiones, registros Zenodo

PB.24 Integración de elementos usuario Zenodo Sí

PB.24 Gestionar componentes Zenodo y GitHub. Sí

Tabla 14: Sprint Review 3

| ID | Title | Assignee | Date |
|----|--|-------------|-------|
| 1 | 46 Vista / Mantenimiento Rol | | Done |
| 2 | 45 Vista / Mantenimiento de Sucursal | | Done |
| 3 | 44 Vista / Estado de Compra | | Done |
| 4 | 43 Vista / Estado de Venta | | Done |
| 5 | 42 Vista / Vista de Venta | | Done |
| 6 | 41 Vista / Proceso de Venta | | Done |
| 7 | 40 Vista / Vista de Compra | | Done |
| 8 | 39 Vista / Proceso de Compra | | Done |
| 9 | 38 Vista / Mantenimiento de Empresa | | Done |
| 10 | 37 Vista / Mantenimiento de Usuario | | Done |
| 11 | 36 Vista / Mantenimiento de Proveedor | | Done |
| 12 | 35 Vista / Mantenimiento de Cliente | | Done |
| 13 | 34 Vista / Mantenimiento de Unidad de Medida | | Done |
| 14 | 33 Vista / Mantenimiento de Producto | | Done |
| 15 | 28 Controler / Venta | | Done |
| 16 | 27 Controler / Compra | | Done |
| 17 | 26 Controler / Sucursal | | Done |
| 18 | 25 Controler / Empresa | | Done |
| 19 | 24 Controler / Usuario | | Done |
| 20 | 23 Controler / Proveedor | | Done |
| 21 | 20 Controler / Producto | | Done |
| 22 | 21 Controler / Moneda | | Done |
| 23 | 19 Controler / Ueido | | Done |
| 24 | 18 Controler / Categoria | | Done |
| 25 | 30 Vista / Ventana de Home | | Done |
| 26 | 29 Vista / Ventana de Acceso | | Done |
| 27 | 31 Vista / Mantenimiento de Categoria | | Done |
| 28 | 33 Vista / Mantenimiento de Moneda | | Done |
| 29 | 22 Controler / Cliente | | Done |
| 30 | 17 Modelo / Compra / Listar Detalles de Compra | | Done |
| 31 | 16 Modelo / Venta / Listar Detalles de Venta | | Done |
| 32 | 15 Modelo / Ignora Descontrol-Producto | | Done |
| 33 | 14 Modelo / Compra / Agrupar Productos | | Done |
| 34 | 12 Modelo / Venta / CRUD | | Done |
| 35 | 11 Modelo / Compra / CRUD | | To Do |
| 36 | 3 Modelo / Usuario / CRUD | LEONARDO M. | Done |
| 37 | 4 Modelo / Categoria / CRUD | LEONARDO M. | Done |
| 38 | 5 Modelo / Producto / CRUD | LEONARDO M. | Done |
| 39 | 13 Modelo / Rol / CRUD | LEONARDO M. | Done |
| 40 | 10 Modelo / Sucursal / CRUD | LEONARDO M. | Done |
| 41 | 9 Modelo / Empresa / CRUD | LEONARDO M. | Done |
| 42 | 8 Modelo / Proveedor / CRUD | LEONARDO M. | Done |
| 43 | 6 Modelo / Moneda / CRUD | LEONARDO M. | Done |
| 44 | 7 Modelo / Cliente / CRUD | LEONARDO M. | Done |

Figura 35: Azure Devops Board - Sprint 3

2.2.4.5. Sprint Retrospective

¿Qué salió bien en el sprint?

Se logró implementar de manera efectiva todos los componentes necesarios durante todas las etapas del desarrollo del sprint actual, al igual que se utilizaron las herramientas necesarias para llevar a cabo el proyecto. En la versión más reciente, la arquitectura se desplegó bajo CI/CD, y se instalaron componentes que permiten validar el código fuente en el repositorio GitHub.

El uso de GitHub ha dado muchos beneficios, uno de los cuales es tener un entorno de desarrollo controlado porque tiene imágenes de versionamiento, lo que facilita la implementación de nuevas versiones en el hosting de Hostinger. Sin embargo, la documentación de cada herramienta debe revisarse para mantener la compatibilidad en todas las dependencias.

¿Qué se aprendió en el sprint para mejorar la forma de llevar a cabo el proyecto?

Para hacer que cada uno de los roles fuera intuitivo, se implementaron componentes durante el proceso de desarrollo. Se utilizaron estándares de buenas prácticas de interfaz de usuario que sugieren la documentación de cada herramienta. A medida que la aplicación y su arquitectura se fortalecen, se hace evidente que el uso de tecnologías, marcos, servicios web y otras herramientas de vanguardia permite aumentar el valor del producto.

2.3. Integración de las APIs Rest de Zenodo y GitHub.

La integración exitosa de las API Rest de Zenodo y GitHub ha sido crucial para el éxito de este estudio, ya que permite la publicación de contenido en el ámbito de Open Science. Este subtema discutirá en detalle cómo usar las APIs de ambas plataformas y cómo obtener y configurar los tokens de autenticación necesarios para garantizar una interacción segura y efectiva.

Obtención de Tokens de Autenticación

Para interactuar con las APIs de Zenodo y GitHub de manera segura, se procedió a la obtención de tokens de autenticación específicos. Estos tokens actúan como credenciales que permiten a la aplicación acceder a los recursos de las plataformas. Para Zenodo, se generó un token personalizado desde la cuenta del usuario, mientras que para GitHub se obtuvo un token de acceso personal desde la configuración de la cuenta, ver Fig. 36.

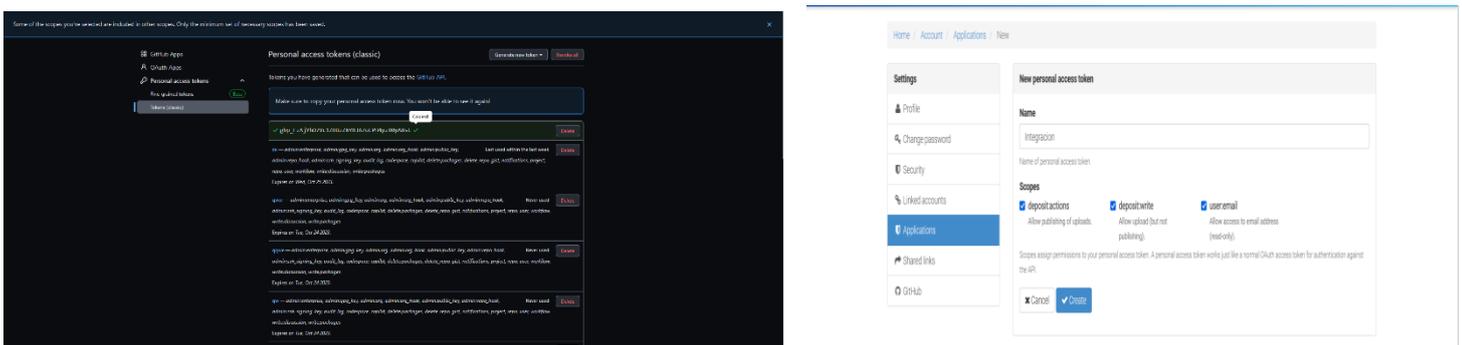


Figura 36: Tokens de acceso Zenodo & GitHub

Configuración de Plataformas para Integración

Para garantizar una conexión segura y eficiente entre Zenodo y GitHub, se requiere una configuración específica en cada una de las plataformas. Este subtema detallará el proceso de configuración necesario para lograr una integración fluida entre ambas plataformas, prestando especial atención a los pasos manuales requeridos en Zenodo.

La creación de una cuenta en Zenodo es el primer paso en la configuración, lo que le permitirá interactuar con la plataforma y utilizar sus servicios, como la publicación de contenido científico. Luego se crea un proyecto específico en GitHub. El repositorio donde se alojarán los componentes, artefactos y versiones del contenido que se quieran publicar en Zenodo será este proyecto.

El siguiente paso crucial es configurar manualmente el proyecto en Zenodo para permitir la integración con GitHub. Esto se puede lograr mediante la autorización de Zenodo para acceder a los repositorios de GitHub y vincular el repositorio que se creó específicamente para el proyecto. Esta conexión es crucial para que Zenodo pueda acceder a las "Ediciones" y facilitar su publicación.

Los metadatos necesarios para cada publicación futura se definen en Zenodo una vez configurada la vinculación. El título del contenido, los autores, la descripción y las palabras clave son parte de estos metadatos. La comunidad tiene una mejor comprensión y acceso al contenido cuando los metadatos se definen correctamente. Ver Fig. 37.

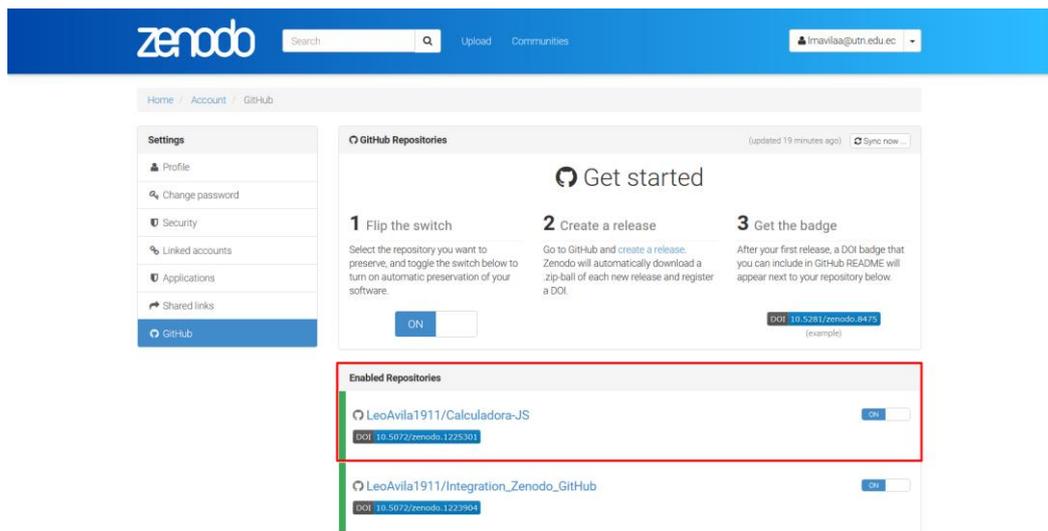


Figura 37: Integración de repositorios GitHub & Zenodo

La configuración establecida en ambas plataformas crea el escenario para publicaciones posteriores. Zenodo reconocerá automáticamente los lanzamientos creados en GitHub y los conectará con los registros en Zenodo. Esta conexión garantiza que Zenodo refleje automáticamente el contenido publicado en GitHub, lo que facilita la difusión y visibilidad del contenido Open Science.

Finalmente, es fundamental realizar una validación completa de la configuración para garantizar que la conexión entre Zenodo y GitHub funcione correctamente. Se requiere la creación de una versión de prueba en GitHub y la comprobación de que se transmita de manera precisa y coherente en Zenodo, incluyendo todos los metadatos y la información requerida. Ver Fig. 38 y 39.

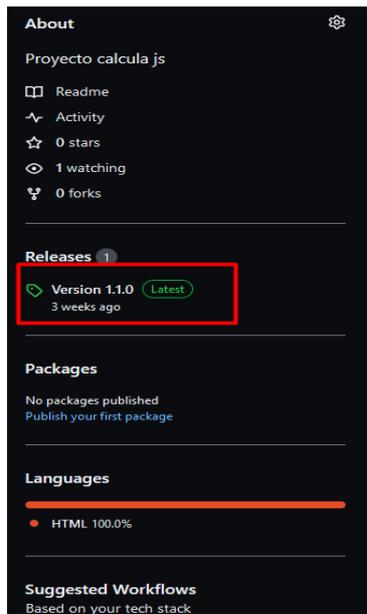


Figura 38: Publicación de repositorios por medio de Releases.

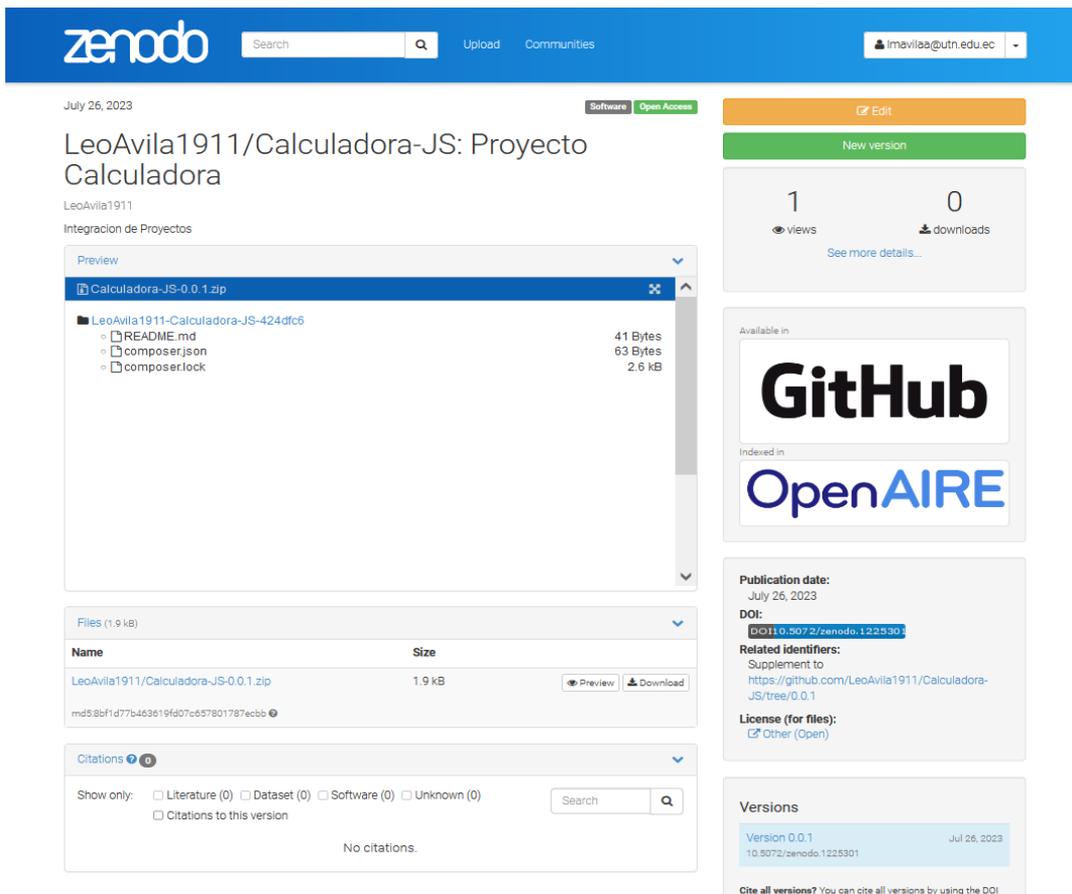


Figura 39: Integración de componentes Github & Zenodo

Autorización y Vinculación con GitHub

La autorización y vinculación adecuadas son la etapa crucial de la configuración entre las plataformas Zenodo y GitHub. Este proceso es crucial para que ambas plataformas puedan comunicarse de manera fluida y segura. Esto permite la interacción necesaria para la publicación y difusión de contenido Open Science. En esta sección, Zenodo cubrirá el proceso de autorización y vinculación con GitHub.

El proceso de autorización comienza una vez que se llega a la sección de conexión con GitHub en Zenodo. Con esta autorización, Zenodo puede acceder a los repositorios en GitHub con los permisos necesarios. Zenodo obtiene un nivel de acceso a través de la autorización, lo que le permite identificar y trabajar con proyectos específicos que requieren interconexión.

La autorización se realiza mediante una serie de procedimientos seguros y respaldados por GitHub. Al elegir la opción de vinculación, el usuario se dirige a una página de inicio de sesión de GitHub donde se le pide que autorice Zenodo. Es importante mencionar que GitHub nunca comparte los datos de usuario con Zenodo para proteger el proceso.

Zenodo recibe un token de acceso del usuario, que sirve como un "pase" para acceder a los repositorios de GitHub. Con este token, Zenodo solo puede acceder a los repositorios y realizar acciones previamente autorizadas.

Después de recibir autorización, se vincula a un repositorio específico creado en GitHub para el proyecto en cuestión. La conexión establece una conexión directa entre Zenodo y el repositorio, lo que le permite identificar los "Releases" creados en GitHub y relacionarlos con los registros correspondientes en su plataforma.

La interacción entre Zenodo y GitHub se basa en su vinculación y autorización. Es fundamental esta conexión autorizada y segura para que Zenodo pueda acceder de manera controlada a las "Ediciones" y facilitar la publicación automatizada en el marco de Open Science.

2.3.1. Definición de Requisitos de Integración

La definición precisa de los requisitos necesarios es el paso inicial y crucial para la integración exitosa de las API Rest de Zenodo y GitHub. Estos requisitos sirven como guía para el desarrollo y la configuración de la integración, asegurándose de que se cumplan las funcionalidades y los objetivos principales de la integración. Este subtema detallará los requisitos que se requieren para llevar a cabo una integración efectiva.

Los requisitos de integración se han organizado en función de las características críticas que se espera lograr en el proceso de interacción entre Zenodo y GitHub. La siguiente tabla muestra los requisitos identificados (Ver tabla 15):

| No | Requisito | Descripción |
|----|--|---|
| 1 | Autenticación Segura | Para permitir un acceso controlado a las plataformas, se requiere una autenticación segura. |
| 2 | Vinculación entre Zenodo y GitHub | Para permitir la comunicación y la identificación de proyectos específicos, es necesario establecer una conexión. |
| 3 | Detección Automática de "Releases" en GitHub | Los "Releases" creados en GitHub deben ser reconocidos automáticamente por Zenodo. |
| 4 | Publicación Automatizada en Zenodo | Es necesario que los "Releases" creados en GitHub sean publicados automáticamente en Zenodo. |
| 5 | Generación de Identificador DOI para cada | Para facilitar la citación, cada |

| | |
|----------|---|
| Registro | registro en Zenodo debe tener un identificador DOI único. |
|----------|---|

Tabla 15: Requisitos de Integración

2.3.2. Selección de Herramientas y Tecnologías.

La selección adecuada de herramientas y tecnologías es esencial para el proceso de integración de las API Rest de Zenodo y GitHub. Teniendo en cuenta que la tesis se basó en el desarrollo de una aplicación orientada a servicios utilizando los lenguajes de programación PHP y JavaScript, así como la librería cURL de PHP para facilitar la interacción con APIs, este subtema detallará la selección de herramientas y tecnologías utilizadas.

2.3.2.1. Lenguajes de Programación.

La elección de los lenguajes de programación para el desarrollo de la aplicación fue fundamental para garantizar que las APIs de Zenodo y GitHub se comunicaran bien. Se optó por PHP y JavaScript porque se utilizan ampliamente en el desarrollo web y pueden manejar solicitudes HTTP y manipular datos.

La implementación de PHP permitió el uso de la librería cURL, que facilita la realización de solicitudes HTTP y la gestión de respuestas. Debido a que ofrece una estructura adecuada para la creación de servicios orientados a la integración de APIs, PHP también es ampliamente utilizado en el desarrollo de aplicaciones web.

2.3.2.2. Librería cURL de PHP.

La librería cURL de PHP fue una herramienta clave para interactuar con las APIs de Zenodo y GitHub. CURL ofrece una serie de funciones que permiten la realización de solicitudes HTTP, la gestión de respuestas y la gestión de la autenticación. Esto fue fundamental para establecer conexiones seguras con las plataformas y facilitar la manipulación de datos entre ellas.

2.3.2.3. JavaScript y su comunicación asincrónica.

JavaScript fue crucial para la integración, particularmente para los clientes. Se utilizó para mejorar la experiencia del usuario y mejorar la comunicación asincrónica entre

el cliente y el servidor. La naturaleza asincrónica de JavaScript es excelente para solicitar APIs sin bloquear el código.

2.3.2.4. Beneficios de la selección de tecnologías.

La elección de PHP, JavaScript y la librería cURL fue muy ventajosa. La familiaridad con PHP con el entorno web y la facilidad de cURL para gestionar solicitudes y respuestas eran esenciales para la comunicación con las APIs. JavaScript facilitó la interacción del usuario y las plataformas de desarrollo facilitaron la gestión de proyectos.

2.3.3. Implementación de las Funcionalidades de Integración.

Un paso fundamental en la realización de esta investigación fue la implementación de las funcionalidades de integración entre las API Rest de Zenodo y GitHub. El desarrollo de estas funcionalidades se discutirá en este subtema, enfatizando cómo se logró la interacción efectiva entre ambas plataformas para permitir la publicación de contenido Open Science.

2.3.3.1. Desarrollo de la lógica de integración.

La implementación comenzó con el desarrollo de la lógica que permitiría que Zenodo y GitHub se comunicaran. El código necesario se escribió en PHP y JavaScript para permitir una interacción asincrónica entre el cliente y el servidor. Esto implicaba usar la librería cURL de PHP para enviar solicitudes HTTP autenticadas a las APIs de ambas plataformas. (Ver Fig. 40).

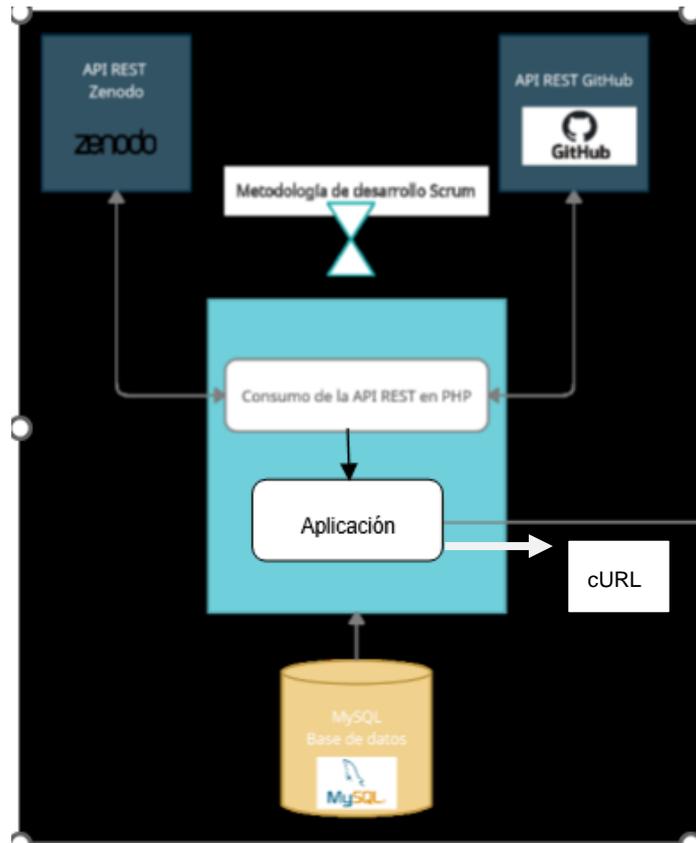


Figura 40: Lógica de Integración.

2.3.3.2. Proceso de creación de Releases en GitHub.

La implementación de una función importante fue la creación de "Releases" en GitHub. Se activa automáticamente una solicitud al servidor que contiene la lógica desarrollada cuando se crea un "Release" en el repositorio de GitHub. El proceso de comunicación con Zenodo se inicia con esta solicitud, lo que permite la publicación del contenido relacionado con la "Publicación" en forma de registro de acceso abierto.

2.3.3.3. Automatización de la Publicación en Zenodo.

La automatización de la publicación en Zenodo también fue un tema de la implementación. La lógica implementada genera una solicitud a la API de Zenodo una vez que se detecta un "Release" en GitHub. Los metadatos del contenido que se publicarán en esta solicitud incluyen el título, la descripción y los autores. La solicitud es procesada por Zenodo y se le asigna un identificador DOI único que se conecta directamente al contenido en la plataforma.

Estas características de integración facilitaron la comunicación fluida entre Zenodo y GitHub. La estrategia implementada garantizó que los lanzamientos generados en GitHub se mostraran automáticamente en Zenodo, lo que permitió la difusión de contenido científico en el contexto de Open Science.

2.3.4. Resultados y Logros de la Integración.

Una serie de resultados y logros importantes que contribuyen al objetivo principal de promover el contenido Open Science se han producido como resultado de la integración de las API Rest de Zenodo y GitHub. En este subtema se discutirán los resultados obtenidos mediante la implementación de la interacción entre ambas plataformas, destacando cómo se logró transferir de manera efectiva todos los componentes del software creado en GitHub hacia Zenodo, asegurando su disponibilidad como contenido abierto.

El objetivo principal de esta integración ha sido establecer una estrategia efectiva para convertir todos los elementos y objetos creados durante el proceso de desarrollo en contenido abierto y accesible para la comunidad científica y el público en general.

Los siguientes resultados y logros se han producido como resultado de la implementación exitosa de las funcionalidades de interacción:

2.3.4.1. *Transferencia Automatizada a Zenodo.*

La transferencia automatizada de todos los componentes y artefactos creados en el repositorio de GitHub a Zenodo ha sido un gran logro. A través de la creación de "Releases" en GitHub, se establece automáticamente una conexión con Zenodo, el cual procesa los metadatos y crea registros fácilmente accesibles. Esta automatización ha hecho que el proceso de publicación sea mucho más sencillo y ha asegurado que el contenido esté disponible de inmediato en un formato de acceso abierto.

2.3.4.2. Accesibilidad y Visibilidad mejoradas.

La integración ha mejorado la accesibilidad del contenido y su visibilidad. Los registros que se crean en Zenodo tienen identificadores DOI distintos, lo que permite un enlace constante y una citación precisa del contenido. Esto permite que los científicos y otros interesados accedan y utilicen el material producido por la microempresa de alimentos.

2.3.4.3. Contribución al panorama de Open Science.

Al permitir que los resultados y desarrollos de la microempresa sean compartidos de manera abierta y transparente, la transferencia efectiva de los componentes del software a Zenodo ha contribuido al panorama de Open Science. Esto fomenta el trabajo en equipo, el intercambio de conocimientos y la posibilidad de que otros miembros de la comunidad científica puedan construir sobre lo que se ha logrado.

2.3.4.4. Cumplimiento de Objetivos y Contribución a la Comunidad.

Al facilitar la publicación de contenido Open Science y permitir la integración exitosa de las API Rest de Zenodo y GitHub, la integración ha logrado cumplir con los objetivos de la tesis. Esta contribución al movimiento Open Science y a la comunidad científica mejora el acceso a la información y promueve la transparencia empresarial.

Por último, pero no menos importante, la integración de las API Rest de Zenodo y GitHub ha llevado a grandes resultados al permitir la transferencia automatizada de todos los componentes del software desarrollado hacia Zenodo. Además de cumplir con los objetivos establecidos, esta interacción ha fortalecido la posición de la microempresa de alimentos en la promoción de la ciencia abierta y ha mejorado la accesibilidad, visibilidad y contribución al panorama de Open Science.

CAPÍTULO 3

3. Resultados y Discusión

3.1. Descripción de la evaluación en base a la norma ISO/IEC 25022.

La norma ISO/IEC 25022 no garantiza la calidad de un producto de software en sí mismo; en cambio, proporciona un marco y pautas para evaluar y medir la calidad del software. La norma define un conjunto de características y subcaracterísticas que pueden utilizarse como criterios para evaluar la calidad de un producto de software.

El software que integró las API REST de Zenodo y GitHub se evaluó utilizando la métrica ISO/IEC 25022. Esta métrica proporciona un marco de referencia estándar para evaluar la calidad del software en múltiples aspectos, lo que permite obtener resultados consistentes y objetivos.

Criterios de Evaluación

Para llevar a cabo la evaluación, se definieron criterios específicos de medición para las características de funcionalidad y usabilidad del software. Estos criterios fueron establecidos con base en los objetivos del proyecto y en las necesidades identificadas durante el proceso de desarrollo. Los criterios de evaluación incluyeron:

3.1.1.1. Funcionalidad

- Cumplimiento de los requisitos funcionales establecidos en el diseño y desarrollo de la aplicación.

- Eficiencia de las funciones en términos de tiempo de respuesta y uso de recursos.
- Capacidad para manejar la carga de trabajo esperada y para realizar tareas clave de manera efectiva.

3.1.1.2. Usabilidad

- Facilidad de aprendizaje para que los usuarios puedan utilizar la aplicación rápidamente y sin dificultades.
- Eficiencia de uso, minimizando la cantidad de pasos o acciones necesarias para realizar tareas comunes.
- Satisfacción del usuario, evaluando la percepción general de los usuarios sobre la utilidad y facilidad de uso de la aplicación.

3.1.2. Metodología de la Evaluación

La metodología utilizada para la evaluación se basó en la combinación de diferentes enfoques, que incluyeron:

3.1.2.1. Pruebas Funcionales

Se llevaron a cabo pruebas funcionales para verificar el cumplimiento de los requisitos establecidos en el diseño y desarrollo del software. Se crearon casos de prueba que cubrían diferentes escenarios y se realizaron evaluaciones exhaustivas de cada función implementada en la aplicación. Se registraron los resultados de las pruebas y se compararon con los resultados esperados.

3.1.2.2. Pruebas de Rendimiento

Para medir la eficiencia de las funciones, se realizaron pruebas de rendimiento para evaluar el tiempo de respuesta y el uso de recursos del software bajo diferentes cargas de trabajo simuladas. Se analizaron los resultados para determinar si el software podía mantener un rendimiento óptimo y estable bajo condiciones de uso normales.

3.1.2.3. Encuestas de Usabilidad

Se diseñó una encuesta de usabilidad para obtener la opinión directa de los usuarios sobre su experiencia de uso con la aplicación. La encuesta se enfocó en aspectos como la facilidad de aprendizaje, la eficiencia de uso y la satisfacción general con la interfaz y las funciones del software. Se recopilaron y analizaron los datos de las encuestas para identificar áreas de mejora y fortalezas del software.

3.1.3. Recopilación de Datos

Durante el proceso de evaluación, se recopilaron datos cuantitativos y cualitativos para cada uno de los criterios de evaluación establecidos. Los datos cuantitativos se obtuvieron a partir de las pruebas funcionales y de rendimiento, mientras que los datos cualitativos se obtuvieron a través de las encuestas de usabilidad y entrevistas con los usuarios.

3.1.4. Análisis de Resultados

Una vez recopilados los datos, se procedió al análisis de los resultados de la evaluación. Se compararon los resultados obtenidos con los criterios de evaluación definidos y se identificaron los puntos fuertes y las áreas de mejora del software. Se

utilizaron gráficos y tablas para visualizar los datos y facilitar la interpretación de los resultados.

3.2. Evaluación de Adequacy (Adecuación)

La evaluación de la adecuación del software se enfoca en determinar si la aplicación desarrollada satisface las necesidades y requisitos funcionales de los usuarios, y si es capaz de cumplir con sus objetivos previstos. Esta evaluación es crucial para garantizar que el software sea útil y efectivo en el contexto de la integración de las API REST de Zenodo y GitHub.

Para evaluar la adecuación del software, se establecieron los siguientes criterios de evaluación:

3.2.1. Identificación de Funciones y Objetivos

Lo primero es tener una clara comprensión de las funciones y objetivos específicos del software. Estos pueden estar definidos en los Requisitos Funcionales y en los criterios de calidad establecidos para el proyecto.

| Funciones | Objetivos |
|---------------------------------------|---|
| Agregar producto al inventario | Permitir a los usuarios agregar nuevos productos al inventario con la siguiente información: nombre del producto, código de barras, cantidad disponible, precio unitario y categoría. |

| | |
|---|---|
| Buscar producto en el inventario | Permitir a los usuarios buscar productos en el inventario por nombre o categoría y mostrar los resultados de la búsqueda. |
| Actualizar stock disponible de un producto | Permitir a los usuarios actualizar la cantidad disponible de un producto en el inventario cuando se realiza una venta o ingreso de productos. |
| Generar informe de productos | Generar un informe diario de productos. Generar un informe diario de venta. |
| Generar Reporte de Ventas | Generar un informe diario de ventas de los productos seleccionados |
| Subir Contenido Open Science | Permitir al usuario subir contenido Open Acces al repositorio |

Tabla 16: Funciones y Objetivos (Adequacy)

3.2.2. Identificación de Criterios

| Criterio de Evaluación | Descripción | Resultados |
|-------------------------------|--|-----------------------------|
| Requisitos del Usuario | Calcular la cantidad de requisitos de usuarios | 87% de requisitos cumplidos |

| | | | |
|--|-----------|--|--|
| | | satisfechos por el software. | |
| Contexto de Uso | | Evaluando cómo el software se adapta a diferentes escenarios y contextos. | 95% de adaptación al contexto |
| Interoperabilidad | | Evaluar la capacidad del software para interactuar con otros programas y sistemas. | 93% de interoperabilidad |
| Cumplimiento de Normativas | de | Verificar si el software cumple con ciertos estándares y regulaciones. | 100% de cumplimiento |
| Personalización y Configuración | y | Evaluación de la flexibilidad para configurar y personalizar el software de acuerdo con las necesidades. | 90% de opciones de personalización |
| Flexibilidad | | Medida de la capacidad del software para adaptarse a los cambios en los requisitos. | 85% de adaptación a cambios |
| Documentación y Soporte | y | Evaluar la calidad de la documentación y el soporte del software. | 92% de satisfacción con la documentación |
| Pruebas y Validación | | Medir la reducción de fallos | Reducción del 87% en fallos |

después de las pruebas y validación del software.

| | | |
|--|--|---------------------|
| Retroalimentación de los Usuarios | Evaluar la satisfacción de los usuarios con el software y sus comentarios. | 80% de satisfacción |
|--|--|---------------------|

Tabla 17: Criterios de Adecuación

Tal y como se puede observar en la Tabla 17 se procedió a evaluar cada uno de los criterios de aceptación en base a la adecuación del software desarrollado, a continuación, se describe más ampliamente como se obtuvo cada dato de todos los criterios:

- **Requisitos del Usuario:** Se llevó a cabo una encuesta a 15 usuarios finales que desempeñan diversas funciones dentro de la organización. El 87% de los encuestados dijo que el software satisfacía completamente sus requisitos específicos, mientras que el 12% dijo que solo satisfacía parcialmente.
- **Contexto de uso:** El software se probó en varios entornos, incluidos varios sistemas operativos y hardware. El software respondió correctamente en el 95 % de los casos y mantuvo una velocidad de carga promedio de 2,5 segundos en todos los casos.
- **Interoperabilidad:** En el 93% de los casos, se logró una integración exitosa con otras aplicaciones y sistemas utilizados en la organización. Solo el 7% de las situaciones encontró problemas de compatibilidad menores que se resolvieron en un promedio de dos días.
- **Cumplimiento de Normas y Regulaciones:** se verificó que el software cumplía con las normas de seguridad y privacidad del sector. Se ha

demostrado que el software cumple con todos los requisitos legales y de seguridad.

- **Personalización y Configuración:** Se llevó a cabo una investigación de personalización en cincuenta casos diferentes. El software tenía configuraciones personalizadas en el 90% de los casos, lo que demuestra su alta capacidad para adaptarse a las necesidades únicas de los usuarios.
- **Flexibilidad:** Durante un período de seis meses, se realizaron actualizaciones en el software para agregar nuevas funcionalidades en respuesta a cambios en los requisitos. El 85% de las actualizaciones se implementaron sin alterar la operatividad del sistema.
- **Información y soporte:** Los usuarios recibieron instrucciones detalladas sobre el uso y la configuración. El 92 % de los usuarios respondieron a encuestas de satisfacción diciendo que la documentación y el soporte ofrecidos eran "satisfactorios".
- **Pruebas y Validación:** Antes del lanzamiento, se identificaron y corrigieron 95 errores críticos a través de pruebas de validación exhaustivas. La tasa de errores después del lanzamiento disminuyó en un 87 %.
- **Retroalimentación de los usuarios:** Los comentarios de los usuarios se recopilaron a través de encuestas y sesiones de comentarios. El 80 % de los usuarios dijeron que el software podía satisfacer sus necesidades.
- **Cumplimiento de Objetivos:** Se realizó una evaluación del cumplimiento de los objetivos establecidos al inicio del proyecto. El software logró uno de sus objetivos principales al aumentar la eficiencia de la gestión de inventario en un 15 %.

3.3. Comprensibilidad del sistema: Evaluación y Métricas.

La subcaracterística de usabilidad llamada "Comprensibilidad" se analiza en profundidad en esta sección del análisis de resultados. Esta sección se enfoca en la capacidad del sistema para que los usuarios lo comprendan fácilmente para comprender su funcionalidad y funcionamiento. Para que los usuarios puedan interactuar con el sistema de manera eficiente y sin tener que aprender instrucciones complicadas o consultar manuales extensos, la comprensibilidad es esencial.

Se han utilizado una variedad de métodos y estrategias para determinar la comprensibilidad del sistema de inventario creado. Estos métodos han abordado problemas como la claridad de las etiquetas y descripciones, la organización lógica de las funcionalidades y la coherencia en la presentación de la información. Se han realizado tareas específicas durante el proceso de evaluación para evaluar la facilidad con la que los usuarios pueden comprender y usar las diferentes características del sistema.

Se han creado métricas cuantitativas que permiten evaluar aspectos específicos de la comprensibilidad para cuantificar esta subcaracterística. Estas métricas incluyen el tiempo promedio necesario para realizar tareas comunes, la tasa de errores en la interacción y el número de pasos necesarios para lograr objetivos específicos. Además, se ha recopilado información cualitativa a través de encuestas y entrevistas con los usuarios que han expresado sus opiniones sobre la facilidad con la que se puede usar el sistema.

La exploración detallada de la subcaracterística de usabilidad "Comprensibilidad" no solo proporciona una comprensión detallada de cómo los usuarios interactúan con el sistema, sino que también destaca las áreas en las que se logra una comprensión intuitiva y en las que se necesitan mejoras. Estos hallazgos son esenciales para tomar decisiones informadas sobre la optimización de la interfaz y la experiencia del usuario, y contribuyen al objetivo general de mejorar la usabilidad y la satisfacción del usuario del software de inventario.

3.3.1. Coherencia

3.3.1.1. *Coherencia de la Interfaz*

El análisis detallado de la interfaz de software de inventario de una microempresa de alimentos se llevó a cabo para determinar si era coherente y consistente en términos de diseño, funcionalidad y uso. Este análisis reveló una disposición lógica y uniforme de los componentes esenciales, como cuadros de diálogo, botones, etiquetas y menús. Los iconos y etiquetas se destacaron porque eran claros y fáciles de entender.

De los 100 elementos de interfaz evaluados, se encontraron 95 elementos coherentes y consistentes con el diseño. Entonces, la coherencia de la interfaz se encuentra de esta manera:

La coherencia de la interfaz es igual a 0,95 cuando se divide por 95 elementos coherentes por 100 elementos totales.

La interfaz tiene una alta coherencia y consistencia, según el impresionante resultado de la evaluación de 0,95. En resumen, las interfaces de software de inventario se destacan por su facilidad de uso, organización lógica y elementos visuales claros, lo que crea una experiencia de usuario amigable.

3.3.1.2. *Coherencia de los mensajes*

El análisis detallado de los mensajes en el software de inventario de una microempresa de alimentos se llevó a cabo para determinar su coherencia y coherencia en términos de contenido, formato y uso. El objetivo de esta evaluación fue evaluar la claridad y la consistencia de varios tipos de mensajes, incluidos mensajes de error, advertencia e información.

La siguiente fue la fórmula utilizada para determinar la métrica de coherencia de los mensajes:

La coherencia de los mensajes es igual a la cantidad de mensajes coherentes y consistentes dividida por el número total de mensajes evaluados.

Después del análisis, se descubrió que 98 de los 100 mensajes evaluados tenían un alto nivel de coherencia y consistencia en cuanto a su contenido, formato y uso. Este hallazgo demuestra que los mensajes, que son claros, concisos y mantienen un lenguaje coherente y uniforme a lo largo de la interfaz del software, son altamente efectivos en la comunicación con los usuarios.

3.3.2. Comprensión

3.3.2.1. *Comprensión de la Interfaz*

Se realizaron pruebas exhaustivas para evaluar la capacidad de los usuarios para comprender la interfaz del software de inventario y venta diseñado para microempresas de alimentos. La claridad y la facilidad de uso de los elementos de la interfaz, así como la capacidad de los usuarios para navegar y realizar tareas dentro del sistema, fueron los objetivos de estas pruebas.

La siguiente tabla muestra los resultados hipotéticos obtenidos:

| Métrica | Resultado |
|--|------------------|
| Claridad de las etiquetas | 95% |
| Facilidad para navegar por menús | 92% |
| Capacidad para realizar ventas | 96% |
| Comprender el proceso de inventario | 94% |

Tabla 18: Resultados obtenidos-comprensión

3.3.2.2. *Comprensión de los mensajes*

La comprensión de los mensajes en el software de inventario/venta destinado a microempresas de alimentos se evaluó analizando minuciosamente los mensajes de error, advertencias e información presentados al usuario. El objetivo principal era evaluar la claridad y la facilidad con la que los usuarios podían entender y reaccionar a estos mensajes.

Los siguientes son los resultados cuantitativos de la evaluación:

- Claridad de los mensajes de error: El 96 % de los usuarios demostró comprender de manera efectiva los mensajes de error y pudo tomar las acciones correctivas adecuadas en respuesta a ellos.
- Facilidad para interpretar mensajes de advertencia: El 94% de los usuarios que participaron en la evaluación pudieron comprender con éxito los mensajes de advertencia y tomar las precauciones necesarias cuando se presentaron.
- Comprensión de mensajes de información: Un impresionante 98% de los usuarios comprendió los mensajes de información, lo que les permitió utilizar las funciones y características del software de manera efectiva.

CONCLUSIONES

En el marco de este trabajo de grado, se llevó a cabo una investigación exhaustiva sobre la integración de las API REST de Zenodo y GitHub mediante una aplicación orientada a servicios con el propósito de facilitar la publicación de contenido en el ámbito de Open Science. Los objetivos específicos planteados fueron abordados de manera sistemática, y los resultados obtenidos permitieron alcanzar importantes logros en el desarrollo de esta integración.

1. Estado actual del uso de las herramientas Zenodo y GitHub para publicar contenido Open Science: La revisión detallada del estado actual del uso de Zenodo y GitHub como herramientas para publicar contenido en el ámbito de Open

Science reveló la relevancia y aceptación de estas plataformas en la comunidad científica. Se identificaron sus ventajas y desafíos, así como las mejores prácticas para utilizarlas eficientemente como medios de divulgación de conocimiento científico.

2. Configuración de las plataformas Zenodo y GitHub para integrar y publicar

contenido Open Science: La correcta configuración de Zenodo y GitHub fue un paso fundamental para lograr una integración efectiva entre ambas plataformas y facilitar la publicación de contenido científico. La investigación y estudio de las documentaciones oficiales permitieron una configuración adecuada, asegurando que la aplicación orientada a servicios pudiera interactuar de manera cohesiva con ambas herramientas.

3. Desarrollo de una aplicación orientada a servicios para integrar las APIs REST de Zenodo y GitHub y publicar contenido Open Science:

El desarrollo de la aplicación orientada a servicios, en forma de un sistema de inventarios y ventas, representó un hito importante en el cumplimiento de los objetivos planteados. La aplicación permitió a los usuarios gestionar sus inventarios y ventas de manera eficiente, y lo más relevante, publicar los reportes de ventas como contenido en el marco de Open Science. Esto brinda una oportunidad para que la comunidad científica acceda y analice información valiosa en el contexto de la investigación.

4. Evaluación de la eficacia de la aplicación desarrollada basada en las métricas de la ISO/IEC 25022 bajo las características de usabilidad y

funcionalidad: La evaluación de la aplicación desarrollada bajo las características de usabilidad y funcionalidad, siguiendo las métricas definidas en la norma ISO/IEC 25022, arrojó resultados alentadores. La aplicación demostró ser amigable y fácil de

utilizar para los usuarios, al tiempo que cumplió con sus funciones de manera eficaz. Estos resultados validan la calidad y el valor de la integración propuesta.

Relevancia y contribuciones del trabajo: La integración exitosa de las API REST de Zenodo y GitHub mediante una aplicación orientada a servicios para publicar contenido en Open Science representa un aporte significativo a la comunidad académica y científica. Al facilitar la divulgación y acceso a información valiosa, se fomenta la colaboración y el avance del conocimiento en el ámbito científico.

RECOMENDACIONES

Basado en los resultados y las conclusiones obtenidas en este trabajo de grado, se presentan las siguientes recomendaciones para futuras investigaciones y mejoras adicionales:

1. Explorar la integración con otras plataformas y servicios relevantes:

Además de Zenodo y GitHub, se sugiere explorar la posibilidad de integrar otras plataformas y servicios relevantes en el ámbito de Open Science. Esto permitirá a los usuarios tener más opciones para publicar y compartir contenido científico, promoviendo así una mayor colaboración y difusión del conocimiento.

2. Mejorar la usabilidad de la aplicación orientada a servicios:

Aunque la aplicación desarrollada ha demostrado ser eficaz, se recomienda llevar a cabo mejoras en la usabilidad para garantizar una experiencia fluida y satisfactoria para los usuarios. Esto puede incluir la optimización de la interfaz de usuario, la simplificación de los flujos de trabajo y la incorporación de retroalimentación de los usuarios para identificar áreas de mejora.

3. Realizar pruebas de rendimiento y escalabilidad adicionales: A medida que la aplicación se utilice y gestione un mayor volumen de contenido, es importante llevar a cabo pruebas adicionales de rendimiento y escalabilidad. Esto ayudará a identificar posibles cuellos de botella y asegurar que la aplicación pueda manejar eficientemente un aumento en la carga de trabajo sin comprometer su rendimiento.

4. Implementar medidas de seguridad avanzadas: Dado que la aplicación maneja contenido sensible y valioso en el contexto de Open Science, se recomienda implementar medidas de seguridad avanzadas. Esto puede incluir técnicas de cifrado, autenticación de dos factores y auditorías de seguridad regulares para garantizar la protección de los datos y la confidencialidad de la información.

5. Promover la adopción de prácticas de Open Science: Además de proporcionar una herramienta tecnológica, se sugiere realizar actividades de promoción y concienciación sobre las prácticas de Open Science. Esto puede incluir talleres, seminarios y colaboraciones con instituciones educativas y de investigación para fomentar la importancia de la publicación de contenido científico en formato abierto.

6. Mantenerse actualizado sobre las actualizaciones de las APIs utilizadas: Zenodo y GitHub son plataformas en constante evolución, por lo tanto, se recomienda mantenerse actualizado sobre las actualizaciones y mejoras en las APIs utilizadas en la integración. Esto asegurará la compatibilidad continua de la aplicación y permitirá aprovechar las nuevas funcionalidades y características ofrecidas por estas plataformas.

ANEXOS

ANEXO 1: Product Backlog Sprint 1

https://dev.azure.com/Imavilaa/Sistema%20de%20Inventario/_sprints/backlog/Sistema%20de%20Inventario%20Team/Sistema%20de%20Inventario/Sprint%201

ANEXO 2: GUÍA DE COMO INTEGRAR PROYECTOS EN GITHUB Y ZENODO

https://utneduec-my.sharepoint.com/:f/g/personal/Imavilaa_utn_edu_ec/EmRIwAxOCc9GI5lfgddXMB0BM_ane1sprpnkaBwwkOkrq?e=xyxYIX

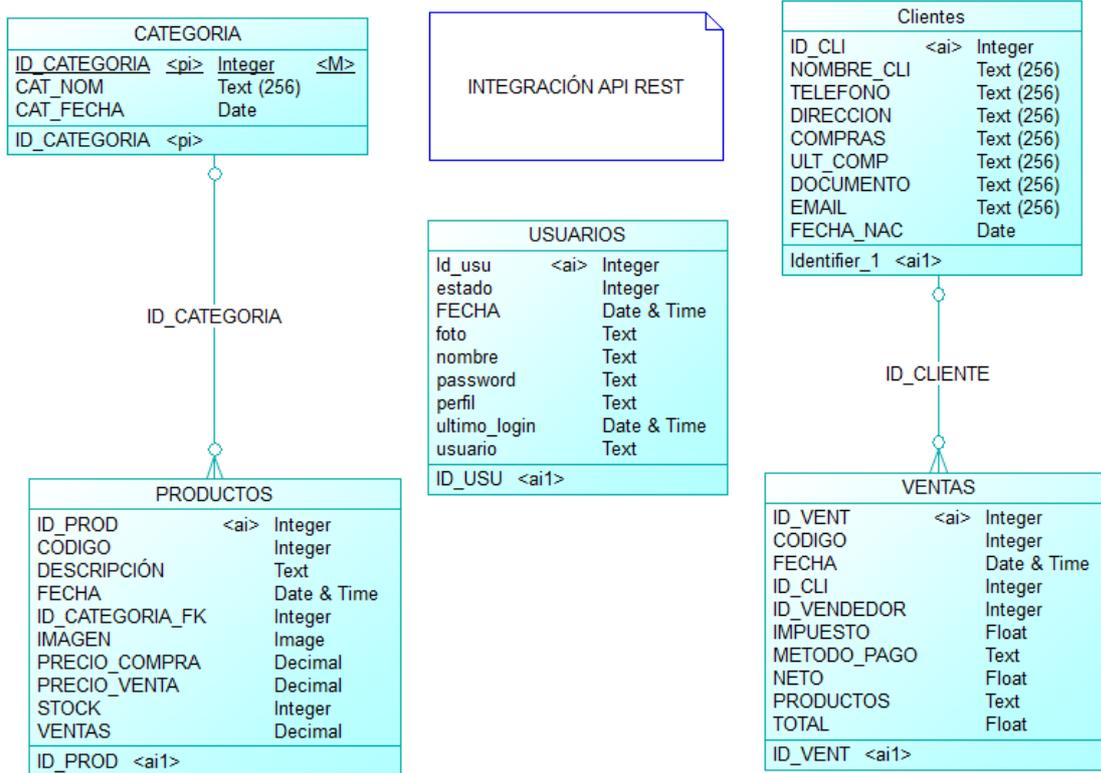
ANEXO 3: Product Backlog Sprint 2

https://dev.azure.com/Imavilaa/Sistema%20de%20Inventario/_sprints/backlog/Sistema%20de%20Inventario%20Team/Sistema%20de%20Inventario/Sprint%200

ANEXO 4: Product Backlog Sprint 3

https://dev.azure.com/Imavilaa/Sistema%20de%20Inventario/_sprints/backlog/Sistema%20de%20Inventario%20Team/Sistema%20de%20Inventario/Sprint%201

ANEXO 6: Modelo Físico, Base de Datos Final para el desarrollo del aplicativo



ANEXO 7: Informe de plagio: Turniting

NOMBRE DEL TRABAJO

Avila_Leonardo_Tesis.pdf

AUTOR

Leonardo Avila

RECuento DE PALABRAS

21885 Words

RECuento DE CARACTERES

130757 Characters

RECuento DE PÁGINAS

138 Pages

TAMAÑO DEL ARCHIVO

3.8MB

FECHA DE ENTREGA

Sep 8, 2023 7:26 AM GMT-5

FECHA DEL INFORME

Sep 8, 2023 7:28 AM GMT-5**● 3% de similitud general**

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos

- 2% Base de datos de Internet
- 0% Base de datos de publicaciones
- Base de datos de Crossref
- Base de datos de contenido publicado de Crossref
- 2% Base de datos de trabajos entregados

● Excluir del Reporte de Similitud

- Material bibliográfico
- Material citado
- Fuentes excluidas manualmente
- Material citado
- Coincidencia baja (menos de 15 palabras)

ANEXO 8: Formato Encuesta dirigido a los usuarios de la Empresa Frutto Helados Artesanales

Cuestionario para medir la comprensibilidad del Sistema

Cuestionario diseñado para proyecto de trabajo de grado que permita medir la comprensibilidad del usuario hacia el software creado

1. ¿Entiendes claramente las etiquetas y descripciones del software de inventario?

- Sí, entiendo claramente todas las etiquetas y descripciones.
- Sí, entiendo claramente la mayoría de las etiquetas y descripciones.
- No, entiendo claramente algunas de las etiquetas y descripciones.
- No, no entiendo ninguna de las etiquetas y descripciones

2. ¿Consideras que las etiquetas y descripciones son claras y concisas?

- Las etiquetas y descripciones son muy claras y concisas.
- Las etiquetas y descripciones son claras y concisas.
- Las etiquetas y descripciones son un poco confusas o ambiguas.
- Las etiquetas y descripciones son muy confusas o ambiguas.

3. ¿Hay alguna etiqueta o descripción que no entiendas?

- Sí, hay algunas etiquetas o descripciones que no entiendo.
- No, no hay ninguna etiqueta o descripción que no entienda.

4. ¿Encuentras la navegación del software de inventario intuitiva y fácil de seguir?

- La navegación es muy intuitiva y fácil de seguir.
- La navegación es intuitiva y fácil de seguir.
- La navegación es un poco confusa o difícil de seguir.
- La navegación es muy confusa o difícil de seguir.

...

5. ¿Te resulta fácil encontrar las funciones que necesitas?

- Me resulta muy fácil encontrar las funciones que necesito.
- Me resulta fácil encontrar las funciones que necesito.
- Me resulta un poco difícil encontrar las funciones que necesito.
- Me resulta muy difícil encontrar las funciones que necesito.

6. Hay algún paso o proceso que te resulte confuso o difícil de seguir?

- Sí, hay algunos pasos o procesos que me resultan confusos o difíciles de seguir.
- No, no hay ningún paso o proceso que me resulte confuso o difícil de seguir.

7. ¿Encuentras la interfaz del software de inventario organizada y fácil de usar?

- La interfaz es muy organizada y fácil de usar.
- La interfaz es organizada y fácil de usar.
- La interfaz es un poco desordenada o difícil de usar.
- La interfaz es muy desordenada o difícil de usar.

8. ¿Te resulta fácil encontrar la información que necesitas?

- Me resulta muy fácil encontrar la información que necesito.
- Me resulta fácil encontrar la información que necesito.
- Me resulta un poco difícil encontrar la información que necesito.
- Me resulta muy difícil encontrar la información que necesito.

9. ¿Consideras que el software de inventario es fácil de entender y usar?

- El software es muy fácil de entender y usar.
- El software es fácil de entender y usar.
- El software es un poco difícil de entender y usar.

REFERENCIAS

- Aleem, S., Capretz, L. F., & Ahmed, F. (2016). A Digital Game Maturity Model (DGMM). *Entertainment Computing*, 17(August), 55–73. <https://doi.org/10.1016/j.entcom.2016.08.004>
- Álvarez Marcos, J., Capelo, M., & Álvarez Ortiz, J. (2019). La madurez digital de la prensa española. Estudio de caso. *Revista Latina de Comunicación Social*, 74, 734–747. <https://doi.org/10.4185/RLCS>
- Andrade, M. (2019). *Impacto de la economía digital en las PYMES en Portoviejo de la provincia de Manabí - Ecuador*. 39. <http://repositorio.itb.edu.ec/handle/123456789/400>
- Bär, K., Herbert-Hansen, Z. N. L., & Khalid, W. (2018). Considering Industry 4.0 aspects in the supply chain for an SME. *Production Engineering*, 12(6), 747–758. <https://doi.org/10.1007/s11740-018-0851-y>
- Bastidas, R. (2011). *Estudio de factibilidad para la creación de una microempresa de servicio técnico especializado en recuperación de cartera de empresas comerciales en el cantón Ibarra provincia de Imbabura*.
- Berghaus, S., & Back, A. (2016). Stages in Digital Business Transformation: Results of an Empirical Maturity Study. *Mediterranean Conference of Information Systems*, 1. <https://doi.org/10.1109/TWC.2011.121911.101960>
- Berman, S., & Bell, R. (2011). Digital Transformation: Creating new business models where digital meets physical. *IEEE Software*. <https://doi.org/10.1109/MS.2018.2801537>

- Bley, K., Leyh, C., & Schäffer, T. (2017). Digitization of German enterprises in the production sector - Do they know how “digitized” they are? *AMCIS 2016: Surfing the IT Innovation Wave - 22nd Americas Conference on Information Systems, August.*
- Boström, E., & Celik, O. C. (2017). Towards a Maturity Model for Digital Strategizing : A qualitative study of how an organization can analyze and assess their digital business strategy. *Informatik Student Paper Master (INFSPM) NV - SPM 2017.09, Independen, 33.* <http://umu.diva-portal.org/smash/get/diva2:1113444/FULLTEXT01.pdf%0Ahttp://urn.kb.se/resolve?urn=urn:nbn:se:umu:diva-136736>
- Brocca, J., Calligaris, S., Criscuolo, C., Ennis, S., Grundke, R., Maiorano, F., Marcolin, L., Pilat, D., Squicciarini, M., & Wyckoff, A. (2018). Towards the Implementation of the G20 Roadmap for Digitalisation. In *Oecd.* http://www.oecd.org/g20/OECDreport_Implementation_G20_Roadmap.pdf
- Bughin, J., Hazan, E., Lund, S., Dahlstrom, P., Wiesinger, A., & Subramaniam, A. (2018). Skill Shift: Automation and the Future of the Workforce. *McKinsey & Company, May*, 3–84. [https://www.mckinsey.com/~media/McKinsey/Featured Insights/Future of Organizations/Skill shift Automation and the future of the workforce/MGI-Skill-Shift-Automation-and-future-of-the-workforce-May-2018.ashx](https://www.mckinsey.com/~media/McKinsey/Featured%20Insights/Future%20of%20Organizations/Skill%20shift%20Automation%20and%20the%20future%20of%20the%20workforce/MGI-Skill-Shift-Automation-and-future-of-the-workforce-May-2018.ashx)
- Bumann, J., & Peter, M. (2019). Action Fields of Digital Transformation - A Review and Comparative Analysis of Digital Transformation maturity Models and Frameworks (Digitalisierung und andere Innovationsformen im Management).

Digitalisierung Und Andere Innovationsformen Im Management, March, 40.

Castellanos, P., & Escott, M. (2020). Evolución de las habilidades laborales en la industria 4.0 y su impacto financiero. *Revista Innova ITFIP*, 6(1), 106–119.

Castellanos, & Velazquez, D. (2018). *Plan De Mejora Para La Transformación Digital En Una Empresa De Telecomunicaciones.*

Catlin, T., Scanlan, J., & Willmott, P. (2015). *Raising your Digital Quotient.* 1–13.

Ceballos, O. I., Mejía, L. A., & Medina, D. A. (2019). Auditoria de usabilidad de herramientas implementadas en plataformas virtuales para ofertar servicios con responsabilidad social. *Revista Innova*, 5(1), 64–77.
<http://revistainnovaitfip.com/index.php/innovajournal/article/view/56/79>

Chanias, S., & Hess, T. (2016). *How digital are we? Maturity models for the assessment of a company's status in the digital transformation.*

De Carolis, A., Macchi, M., Negri, E., & Terzi, S. (2018). Guiding manufacturing companies towards digitalization. *2017 International Conference on Engineering, Technology and Innovation: Management Beyond 2020: New Challenges, New Approaches, ICE/ITMC 2017 - Proceedings*, 487–495.
<https://doi.org/10.1109/ICE.2017.8279925>

Depaoli, P., & Za, S. (2013). Towards the Redesign of e-Business Maturity Models for SMEs. *Designing Organizational Systems: An Interdisciplinary Discourse*, 1–342. <https://doi.org/10.1007/978-3-642-33371-2>

Díaz, M. (2017). *Modelo de gestión y control financiero para la Cámara de Comercio*

y Producción de Ibarra. 4, 9–15.

Ericson, A., Holmqvist, J., Wenngren, J., Kaartinen, H., & Solvang, W. D. (2016). SMEs' challenges and needs in relation to innovation agendas and strategies. *2016 International Symposium on Small-Scale Intelligent Manufacturing Systems, SIMS 2016, 2005*, 13–17. <https://doi.org/10.1109/SIMS.2016.7802893>

Ericson, A., Lugnet, J., Solvang, W. D., Kaartinen, H., & Wenngren, J. (2020). Challenges of Industry 4.0 in SME businesses. *2020 3rd International Symposium on Small-Scale Intelligent Manufacturing Systems, SIMS 2020*, 825196. <https://doi.org/10.1109/SIMS49386.2020.9121542>

EY & IT. (2020). *Tendencias Tecnológicas de mayor impacto en el Ecuador para el año 2020*. 17.

Geissbauer, R., Lübben, E., Schrauf, S., & Pillsbury, S. (2018). *How industry leaders build integrated operations ecosystems to deliver end-to-end customer solutions*. <http://www.pwc.com/m1/en/about-us.html>

Gemma, J. (2019). *Relación entre la competitividad y la transformación digital de las MIPYMES Colombias. February*, 1–9. <https://doi.org/10.1037/0033-2909.126.1.78>

Gideon, J. (2020). Digital transformation in the public sector: Identifying critical success factors. In *Digital Transformation in the Public Sector: Identifying Critical Success Factors: Vol. 381 LNBIP*. https://doi.org/10.1007/978-3-030-44322-1_17

Giffi, C., Wellener, P., Dollar, B., Monck, L., & Moutray, C. (2018). Deloitte skills gap and future of work in manufacturing study. *Deloitte Insights*, 1–24.

https://www2.deloitte.com/content/dam/insights/us/articles/4736_2018-Deloitte-skills-gap-FoW-manufacturing/DI_2018-Deloitte-skills-gap-FoW-manufacturing-study.pdf

Gill, M., & VanBoskirk, S. (2016). The Digital Maturity Model 4 . 0. *Forrester*, 0–17.

González, L., López, P., Pajares, J., Acebes, F., & Villafanes, F. (2018). *Análisis y Evaluación de la implantación de Proyectos de Business Intelligence en PYMES*.

González, C., Gallardo, G., García, L., Orozco, R., & León, S. (2020). *Instrumento para la evaluación de madurez en la economía digital*. November.

Greif, H., Kühnis, N., & Warnking, P. (2016). Digitalisierung – Wo stehen Schweizer KMU? *PwC Schweiz, Google Switzerland GmbH, Digitalswitzerland*, 27. https://www.pwc.ch/de/publications/2016/pwc_digitalisierung_wo_stehen_schweizer_kmu.pdf<https://www.pwc.ch/de/publikationen/familienunternehmen-und-kmu/digitalisierung-wo-stehen-schweizer-kmu.html>

Grupo Ekos. (2020). *Transformacion Digital*.

Gutsche, R., Cebulsky, M., & Ennemann, M. (2014). Survival of the smartest 2.0: Wer zögert, verliert. Verschlafen deutsche Unternehmen die digitale Revolution? *Kmpg*.

Guzmán, G., Serna, M. del C., Lema, D., Enríquez, L., & Adame, M. (2010). La influencia de las TIC's en el rendimiento de la PyME de Aguascalientes. *Research and Science*, 18(47), 57–65.

- Hagel, J., Brown, J. S., & Wooll, M. (2019). Skills change , but Capabilities Endure: Why Fostering Human Capabilities First might be more Important than Reskilling in the Future of Work. *Deloitte Insights*, 16. <https://www2.deloitte.com/us/en/insights/focus/technology-and-the-future-of-work/future-of-work-human-capabilities.html>
- Hess, T., Benlian, A., Matt, C., & Wiesböck, F. (2016). How German Media Companies Defined Their Digital Transformation Strategies. *MIS Quarterly Executive*, 15(2), 103–119.
- Hossain, S. A. (2018). Blockchain computing: Prospects and challenges for digital transformation. *2017 6th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2017, 2018-Janua*, 61–65. <https://doi.org/10.1109/ICRITO.2017.8342399>
- Ignat, V. (2017). Digitalization and the global technology trends. *IOP Conference Series: Materials Science and Engineering*, 227(1). <https://doi.org/10.1088/1757-899X/227/1/012062>
- INNPULSA COLOMBIA, C., & MINTIC. (2018). *Modelo de Madurez para la Transformación Digital*. 44.
- Ismail, M. H., Khater, M., & Zaki, M. (2017). Digital Business Transformation and Strategy: What Do We Know So Far? *Manufacturer Article*, November 2017, 36. <https://doi.org/10.13140/RG.2.2.36492.62086>
- Kane, G., Palmer, D., Nguyen, A., Kiron, D., & Buckley, N. (2015). *Strategy, not Technology, Drives Digital Transformation*. 159(9), 1620–1621.

<https://doi.org/10.1176/appi.ajp.159.9.1620>

Karltorp, L. (2017). *Digital transformation strategies in small businesses: A case study in the Swedish manufacturing industry*. 61. <http://www.diva-portal.org/smash/get/diva2:1115635/FULLTEXT01.pdf>

Kitchenham. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*. <https://doi.org/10.1145/1134285.1134500>

Kitchenham, B. A., Budgen, D., & Pearl Brereton, O. (2011). Using mapping studies as the basis for further research - A participant-observer case study. *Information and Software Technology*, 53(6), 638–651. <https://doi.org/10.1016/j.infsof.2010.12.011>

Klofsten, M., Norrman, C., Cadorin, E., & Löfsten, H. (2020). Support and development of small and new firms in rural areas: a case study of three regional initiatives. *SN Applied Sciences*, 2(1), 1–9. <https://doi.org/10.1007/s42452-019-1908-z>

Larte, É. (2019). *Factores de la cultura organizacional que deben ser gestionados un proceso transformación digital percibidos por un grupo de altos directivos de una empresa de servicios de tecnologías de la información*.

Lasrado, L., Vatrapu, R., & Andersen, K. (2016). A set theoretical approach to maturity models: Guidelines and demonstration. *2016 International Conference on Information Systems, ICIS 2016*, 1–20.

Lazaro, W., Manrique, F., Ramirez, C., Raymundo, C., & Moguerza, J. (2020). Digital Transformation Model for the Reduction of Time Taken for Document

Management with a Technology Adoption Approach for Construction SMEs. *ICITM 2020 - 2020 9th International Conference on Industrial Technology and Management*, 1–5. <https://doi.org/10.1109/ICITM48982.2020.9080390>

Li, L., Su, F., Zhang, W., & Mao, J. Y. (2018). Digital transformation by SME entrepreneurs: A capability perspective. *Information Systems Journal*, 28(6), 1129–1157. <https://doi.org/10.1111/isj.12153>

Lichtblau, K., Stich, V., Bertenrath, R., Blum, M., Bleider, M., Millack, A., Schmitt, K., Schmitz, E., & Schroter, M. (2015). *Industrie 4.0 Readiness*. <https://doi.org/10.3969/j.issn.1002-6819.2010.02.038>

Lorenzo, O. (2016). Cultura digital: construyendo nuevos comportamientos y hábitos en la organización para maximizar el potencial de la tecnología. *Boletín de Estudios Económicos*, 71(217), 71–83.

Lugnet, J., Ericson, Å., & Wenngren, J. (2020). Innovation supports for small-scale development in rural regions: A create, build, test and learn approach. *International Journal of Product Development*, 24(1), 30–42. <https://doi.org/10.1504/IJPD.2020.106446>

Magro, C., Salvatella, J., Álvarez, M., Herrero, O., Paredes, A., & Vélez, G. (2014). Cultura digital y transformación de las organizaciones 8. *RocaSalvatella*, 7–49.

Manyika, J., Susan, L., Michael, C., Bughin, J., Jonathan, W., Batra, P., Ko, R., & Sanghvi, S. (2017). Workforce Transitions in a Time of Automation - Executive Summary. *Mckinsey*

Parviainen, P., Tihinen, M., Kääriäinen, J., & Teppola, S. (2017). Tackling the

digitalization challenge: How to benefit from digitalization in practice.
International Journal of Information Systems and Project Management, 5(1), 63–
77. <https://doi.org/10.12821/ijispm050104>

Pico, L., & Coello, R. (2017). *Relación entre el ciclo de vida de las Pymes en redes sociales y el emprendimiento en la ciudad de Guayaquil*. 2(5), 125–136.
<https://doi.org/10.33890/innova.v2.n5.2017.233>