



UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E
INFORMÁTICA



**DESARROLLO DE UN SISTEMA WEB CON CUADRO DE
MANDO, COMBINANDO TECNOLOGÍAS JAVA Y R PARA LA
ADMINISTRACIÓN DE SERVICIOS DEL CITE TEXTIL
CAMÉLIDOS PUNO - 2019**

TESIS

PRESENTADA POR:

Bach. ADALIDT ANCACHI LIMACHI

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ESTADÍSTICO E INFORMÁTICO

PUNO – PERÚ

2021



DEDICATORIA

A Braulia, mi querida madre, que me crío, me llenó de amor y cariño y ahora que no está con nosotros aún sigue cuidando y guiando el camino de sus hijos desde el cielo.

A Alipio, mi padre y amigo, que me enseñó a valirme por mi mismo desde pequeño y a hacerme responsable de mis acciones y hoy desde el cielo vela por el bienestar de sus hijos.

A Yesenia, mi compañera de vida, que ha estado presente siempre en mi vida, acompañándome en momentos felices y tristes y me ha ayudado a sobrellevar mi tristeza.

A Efraín y Sandro, mis hermanos, con los que compartí una infancia llena de alegrías y con los que supe afrontar y sobrellevar momentos tristes de mi vida.

A Jabel, el hermano que yo elegí, con quien compartí gratos momentos universitarios y que siempre me animó a culminar este trabajo.

Adalidt Ancachi Limachi



AGRADECIMIENTOS

A mis padres Alipio y Braulia que siempre me inculcaron valores, por haberme enseñado que en la vida siempre se debe ser honesto, por haberme brindado las oportunidades que ellos no tuvieron y siempre apoyarme hasta el día su partida.

Agradezco también a mis hermanos Efraín y Sandro por tenerme paciencia y brindarme siempre su apoyo consejo y amistad.

A los docentes de la Escuela Profesional de Ingeniería Estadística e Informática por brindarme sus conocimientos y experiencias durante mi carrera universitaria y haber inculcado en mi la pasión por la tecnología y la resolución de problemas mediante su uso y aplicación.

Agradezco también al Ing., Fredy por darme la oportunidad de aplicar mis conocimientos por primera vez y darme ese empujón y esa confianza para el inicio de mi vida profesional y haber aceptado ser mi asesor.

A los Ing. Remo, Ing. Charles e Ing. Ángel por haberse tomado el tiempo y darse la molestia de revisar el presente trabajo de investigación.

Adalidt Ancachi Limachi



ÍNDICE GENERAL

DEDICATORIA

AGRADECIMIENTOS

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ACRÓNIMOS

RESUMEN..... 13

ABSTRACT..... 14

CAPÍTULO I

INTRODUCCIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA..... 16

1.2 FORMULACIÓN DEL PROBLEMA 18

1.3 OBJETIVOS DE INVESTIGACIÓN 18

 1.3.1 Objetivo General 18

 1.3.2 Objetivos Específicos..... 18

1.4 HIPÓTESIS DE INVESTIGACIÓN..... 18

1.5 JUSTIFICACIÓN DE LA INVESTIGACIÓN 19

1.6 DELIMITACIÓN DE LA INVESTIGACIÓN..... 19

CAPÍTULO II

REVISIÓN DE LITERATURA

2.1 ANTECEDENTES DE LA INVESTIGACIÓN 21

2.2 BASE TEÓRICA..... 25

 2.2.1 Lenguaje de programación Java..... 25

 2.2.2 Kit de Desarrollo de Java (JDK) 26

 2.2.3 Amazon Corretto 27



2.2.4 Java VM (Máquina Virtual de Java)	28
2.2.5 Spring Framework.....	29
2.2.6 Spring Boot	29
2.2.7 Base de Datos Relacional.....	30
2.2.8 Maria DB Server	31
2.2.9 R.....	32
2.2.10 Shiny	33
2.2.11 Patrón de diseño MVC.....	34
2.2.12 Sistema web	35
2.2.13 Cuadro de mando	36
2.2.14 Servidor de Aplicaciones	38
2.2.15 API Rest	39
2.2.16 Metodología de desarrollo ágil.....	40
2.2.17 Metodología de desarrollo XP	41
2.2.18 Desarrollo de software	43
2.2.19 Pruebas unitarias	44
2.2.20 JUnit.....	45
2.2.21 Contenedores de Software.....	46
2.2.22 Docker.....	47
2.2.23 Sistema de control de versiones	48
2.2.24 Git.....	49
2.2.25 Git Hub.....	51
2.2.26 Estadístico de Prueba U de Mann Whitney.....	51

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1 UBICACIÓN GEOGRÁFICA DEL ESTUDIO	52
---	-----------



3.2 POBLACIÓN	52
3.3 DISEÑO DE LA MUESTRA	52
3.4 MÉTODO DE RECOPIACIÓN DE INFORMACIÓN	52
3.5 METODOLOGÍA DE DESARROLLO DE SOFTWARE	53
3.6 MATERIAL EXPERIMENTAL	54
3.6.1 Hardware	55
3.6.2 Software	55

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1 ANÁLISIS.....	56
4.1.1 Viabilidad.....	56
4.1.2 Historia del Usuario	57
4.1.3 Roles.....	58
4.1.4 Planificación.....	59
4.1.5 Análisis de requerimientos del sistema	60
4.1.6 Análisis de requerimientos técnicos.....	62
4.2 DISEÑO	63
4.2.1 Documentación mediante UML.....	64
4.2.2 Diagramas de caso de uso	64
4.2.3 Diagramas de secuencia	66
4.2.4 Diagrama de flujo.....	71
4.3 DESARROLLO DEL SISTEMA.....	72
4.3.1 Diseño de base de datos	73
4.3.2 Interfaces del sistema	76
4.4 DESARROLLO DEL CUADRO DE MANDO (DASHBOARD) CON R.....	81
4.4.1 Audiencia y Propósito	82
4.4.2 Definición las métricas.....	82



4.4.3 Catalogación las fuentes de datos.....	83
4.4.5 Definición de dimensiones y filtros	83
4.4.6 Implementación del Cuadro de Mando (Dashboard)	84
4.5 PRUEBAS UNITARIAS.....	87
4.6 VALIDACIÓN DEL SISTEMA	91
4.6.1 ISO 25010	91
4.6.2 Eficiencia de desempeño	92
4.6.3 Comportamiento temporal	93
4.6.4 Capacidad.....	97
4.6.5 Adecuación funcional y Usabilidad	99
4.6.6 Contraste de hipótesis	110
V. CONCLUSIONES.....	116
VI. RECOMENDACIONES	118
VII. REFERENCIAS BIBLIOGRÁFICAS	119
ANEXOS	127
Anexo 1. Cuestionario.....	128

Área: Informática

Tema: Desarrollo de Software

FECHA DE SUSTENTACIÓN: 28 de diciembre 2021



ÍNDICE DE TABLAS

Tabla 1 Historias de usuario recopiladas para el desarrollo del sistema	57
Tabla 2 Cronograma de actividades para el desarrollo del sistema	59
Tabla 3 Componentes de hardware mínimos requeridos	62
Tabla 4 Software utilizado para el desarrollo del sistema	62
Tabla 5 Descripción de las subcategorías de eficiencia de desempeño	92
Tabla 6 Resumen de los resultados de la medición del comportamiento temporal	95
Tabla 7 Resumen del informe agregado de la medición de rendimiento con ocho usuarios	97
Tabla 8 Resumen de la medición de capacidad con cincuenta usuarios	98
Tabla 9 Tabla de distribución de frecuencia de la pregunta N° 1	100
Tabla 10 Tabla de frecuencias de la pregunta N° 2	101
Tabla 11 Tabla de frecuencia de la pregunta N° 3	102
Tabla 12 Tabla de frecuencias de la pregunta N° 4	103
Tabla 13 Tabla de frecuencias de la pregunta N° 5	104
Tabla 14 Tabla de frecuencias de la pregunta N° 6	105
Tabla 15 Tabla de frecuencias de la pregunta N° 7	106
Tabla 16 Tabla de frecuencias de la pregunta N° 8	107
Tabla 17 Valorización de respuestas del cuestionario	108
Tabla 18 Intervalos del puntaje acumulado	108
Tabla 19 Acumulado de valores obtenidos en el cuestionario	109
Tabla 20 Tabla de mediciones del tiempo empleado para la elaboración de reportes..	110



ÍNDICE DE FIGURAS

Figura 1 Logo MaríaDB Foundation	32
Figura 2 Estructura básica de un proyecto con Shiny	34
Figura 3 Patrón de diseño MVC (Modelo, Vista, Controlador)	35
Figura 4 Funcionamiento de las API Rest	40
Figura 5 Proceso ágil de desarrollo de software	41
Figura 6 Logo de JUnit 5	46
Figura 7 Arquitectura de una imagen Docker	47
Figura 8 Comparación entre una máquina virtual y un contenedor Docker	48
Figura 9 Funcionamiento de un sistema de control de versiones	49
Figura 10 Representación gráfica de ramas y commits	50
Figura 11 Fórmula U de Mann-Whitney	51
Figura 12 Diagrama de caso de uso general del sistema	66
Figura 13 Diagrama de secuencia de acceso al sistema	67
Figura 14 Diagrama de secuencia de registro de unidad productiva	68
Figura 15 Diagrama de secuencia de registro de personal	69
Figura 16 Diagrama de secuencia de registro de evento	70
Figura 17 Diagrama de secuencia de consulta cuadro de mando	71
Figura 18 Diagrama de flujo del proceso de elaboración de reportes mensuales	72
Figura 19 Diagrama de entidad relación de la base de datos	74
Figura 20 Inicio de sesión fallido en el sistema	78
Figura 21 Cierre de sesión correcto en el sistema	78
Figura 22 Página de inicio del sistema para el usuario administrador	79
Figura 23 Página de inicio del sistema para el usuario digitador	79
Figura 24 Pantalla de registro de unidades productivas en el sistema	80



Figura 25 Pantalla de registro de evento realizado en el sistema	80
Figura 26 Pantalla de registro de servicios brindados en el sistema.....	81
Figura 27 Código de implementación del cuadro de mando	84
Figura 28 Interfaz de servicios brindados en el cuadro de mando.....	85
Figura 29 Interfaz de unidades productivas del cuadro de mando	85
Figura 30 Interfaz de personal en el cuadro de mando	86
Figura 31 Interfaz de eventos realizados en el cuadro de mando	87
Figura 32 Configuración de JUnit 5 en el archivo POM.xml del proyecto	88
Figura 33 Clases creadas para el testeo de los servicios.....	88
Figura 34 Prueba unitaria para los servicios de tareas.....	89
Figura 35 Resultado correcto de la prueba unitaria	90
Figura 36 Resultados incorrectos de la prueba unitaria.....	90
Figura 37 Categorías de calidad de producto del ISO/IEC 25010.....	91
Figura 38 Categorías consideradas para la calidad del sistema	92
Figura 39 Configuración de JMeter para medir el comportamiento temporal del sistema	93
Figura 40 Árbol de resultados de la prueba de comportamiento temporal.....	94
Figura 41 Informe agregado de la medición del comportamiento temporal.....	94
Figura 42 Configuración de JMeter para las pruebas de comportamiento temporal con ocho usuarios	96
Figura 43 Informe agregado del resultado de la medición de rendimiento con ocho usuarios	96
Figura 44 Configuración de JMeter con cincuenta usuarios.....	98
Figura 45 Gráfico de barras de la pregunta N° 1	100
Figura 46 Gráfico de barras de la pregunta N° 2	101



Figura 47 Gráfico de barras de la pregunta N° 3	102
Figura 48 Gráfico de barras de la pregunta N° 4	103
Figura 49 Gráfico de barras de la pregunta N° 5	104
Figura 50 Gráfico de barras de la pregunta N° 6	105
Figura 51 Gráfico de barras de la pregunta N° 7	106
Figura 52 Gráfico de barras de la pregunta N° 8	107
Figura 53 Gráficos de densidad de las mediciones de tiempo	111
Figura 54 Prueba de normalidad de Shapiro-Wilk en R.....	112
Figura 55 Formulas de la prueba Mann-Withney	112
Figura 56 Medianas del tiempo con sistema y sin sistema	113
Figura 57 Prueba de Mann Withney en R.....	113



ÍNDICE DE ACRÓNIMOS

MVC: Modelo Vista Controlador

API: Aplicacion Programming Interface (Interfaz de Programación de Aplicaciones)

REST:Representational State Transfer (Transferencia de Estado Representacional)

HTTP: Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto)

ISO: International Organization for Standardization(Organización Internacional de Normalización)

IEC: International Electrotechnical Commission (Comisión Electrotécnica Internacional)



RESUMEN

El CITE textil camélidos Puno brinda servicios a unidades productivas de la cadena productiva textil camélidos, la información que se recopila al desarrollar servicios debe ser informada, digitalizada, almacenada y reportada periódicamente, esto se realizaba de forma manual dificultando la consolidación de información, reporte y búsqueda de información. La investigación titulada “Desarrollo de un sistema web con cuadro de mando, combinando tecnologías Java y R para la administración de servicios del CITE textil camélidos Puno -2019” desarrollado en la ciudad de Puno, Región Puno en el periodo 2019, cuyo objetivo principal fue desarrollar un sistema web con cuadro de mando combinando tecnologías Java y R para el almacenamiento y procesamiento de información brindando una interfaz amigable para reducir tiempos combinando tecnologías Java y R para mostrar un cuadro de mando. Para el desarrollo del sistema se utilizaron las pautas de la metodología de desarrollo ágil XP, la validación se hizo con la norma ISO/IEC 25010, se finalizó la investigación comparando medias del empleado con ayuda y sin ayuda del sistema con la prueba U de Mann Withney. Tras el desarrollo del sistema web se logró su implementación brindando interfaces de usuario amigables para su uso, se redujo el tiempo en el registro y generación de reportes de servicios, se logró interoperar tecnologías Java y R mediante la utilización de API’s REST y finalmente se logró mostrar un cuadro de mando interactivo para la toma de decisiones. Del desarrollo de la investigación podemos concluir que el desarrollo e implementación de un sistema web con cuadro de mando reduce significativamente el tiempo empleado en el registro de información y la generación de reportes, además la interoperabilidad de tecnologías distintas como Java y R facilitan el proceso de creación de un cuadro de mando interactivo para la toma de decisiones.

Palabras Clave: Sistema web, cuadro de mando, toma de decisiones, planificación de actividades.



ABSTRACT

The CITE textile camelid Puno provides services to productive units of the camelid textile production chain, the information that is collected to develop services must be reported, digitized, stored and reported periodically, this was done manually making it difficult to consolidate information, reporting and search for information. The research entitled "Development of a web system with dashboard, combining Java and R technologies for the administration of services of the camelid textile CITE Puno -2019" developed in the city of Puno, Puno Region in the period 2019, whose main objective was to develop a web system with dashboard combining Java and R technologies for the storage and processing of information providing a friendly interface to reduce time by combining Java and R technologies to display a scorecard. For the development of the system the guidelines of the XP agile development methodology were used, the validation was done with the ISO/IEC 25010 standard, the research was finalized by comparing averages of the employee with and without the help of the system with the Mann Withney U test. After the development of the web system, its implementation was achieved by providing user-friendly interfaces for its use, the time in the registration and generation of service reports was reduced, Java and R technologies were interoperated through the use of REST API's and finally an interactive scorecard for decision making was displayed. From the development of the research we can conclude that the development and implementation of a web system with scorecard significantly reduces the time spent in recording information and generating reports, and the interoperability of different technologies such as Java and R facilitates the process of creating an interactive scorecard for decision making.

Keywords: Web system, scorecard, decision making, activity planning.



CAPÍTULO I

INTRODUCCIÓN

El centro de Innovación Productiva y Transferencia Tecnológica textil camélidos de Puno brinda servicios de capacitación, proyectos de I+D+I, información especializada y asistencia técnica a todas la unidades productivas de la cadena productiva de transformación de fibra de camélidos sudamericanos en la región Puno, la información recabada en los eventos generan información como servicios brindados y clientes atendidos que deben de reportarse mensualmente para medir el cumplimiento de metas, este proceso se venía realizando de forma manual esto ocasiona en muchos casos se registre información redundante y en muchos casos repetida, se invierta mucho tiempo en su registro y el análisis de esta información lleve tiempo además de que generaba dependencia en personal dedicado a ello, por lo que se plantea el desarrollo e implementación de un sistema web con cuadro de mando combinando tecnologías Java y R para la administración de servicios y toma de decisiones.

Para el desarrollo del sistema web se utilizó la tecnología Java para el desarrollo del back-end e interfaces de interacción con los usuarios para el registro de la información para esto se empleó diagramas UML para documentar el desarrollo del sistema, la metodología ágil XP nos ayudó a hacer más flexible el desarrollo del sistema con pruebas unitarias, utilizamos Junit5 y logramos testear los módulos del sistema, los diagramas entidad relación para la implementación de la lógica de la base de datos y para agilizar el desarrollo con tecnología Java Spring Boot proporcionó todo lo necesario para este fin. Para el desarrollo del cuadro de mando se utilizó el lenguaje de programación R con su librería Shiny que proporciona todos los componentes para la implementación de un cuadro de mando reactivo y amigable con el usuario. El sistema también se validó



utilizando lo propuesto en la norma ISO/IEC 25010, la presente investigación está conformada por los siguientes capítulos.

En el capítulo I describimos el plan de la investigación aquí se encuentran plasmados nuestro Problema, el Planteamiento del Problema, la Hipótesis de investigación. En el Capítulo II tenemos la Revisión de la Literatura, Antecedentes y definiciones conceptuales. En el Capítulo III describimos los materiales y métodos empleados en la investigación. En el Capítulo IV mostramos los resultados obtenidos y finalizamos con las conclusiones y recomendaciones.

1.1 PLANTEAMIENTO DEL PROBLEMA

Perú es uno de los países con la mayor cantidad de población de alpacas en el mundo y según (MINAGRI, 2019) actualmente la población de alpacas supera los 6 millones de ejemplares, los cuales están distribuidos en más de 60 países de los 5 continentes del mundo; siendo Perú (71.7%) el país con la mayor población, seguido de Bolivia, Australia, Estados Unidos y los países europeos.

El Centro de Innovación Productiva y Transferencia Tecnológica textil Puno – CITE textil camélidos Puno tiene como objeto contribuir a la mejora de la productividad y competitividad de la empresas, organizaciones y pequeños productores de este sector productivo a través de actividades de capacitación, asistencia técnica, asesoría especializada para la adopción de nuevas tecnologías, transferencia tecnológica, investigación, desarrollo e innovación productiva y servicios tecnológicos, difusión de información; interrelación de actores estratégicos y generación de sinergias bajo un enfoque de demanda.



Para llevar un control de los servicios brindados por el CITE textil camélidos Puno se elaboran reportes de servicio mensual; con la aplicación de tecnologías de la información se optimiza los procesos de registro, almacenamiento de información, automatizando la elaboración de reportes, evitando la redundancia de información, manteniendo una sola base de datos con todas las medidas de seguridad y respaldo para salvaguardarla; así mismo nos permite procesar información y presentarla de manera gráfica para una fácil interpretación y toma de decisiones de intervención.

Para el proceso de registro de servicios se debe de tomar en cuenta ciertas indicaciones y criterios que se repiten cada mes, la información de los clientes debe ser registrada con cuidado y utilizada cada vez que este sea atendido en cualquier evento o servicio a medida que brinde el CITE textil camélidos Puno, esta información también debe de ser accedida por los colaboradores del CITE textil camélidos Puno para contactar algún cliente, buscar productores de una línea de producción, programar eventos, generar reportes. Un sistema web basado en arquitectura Modelo Vista Controlador (MVC) facilita todas estas actividades aminorando el tiempo dedicado y centralizando información vital para la toma de decisiones.

EL CITE textil camélidos Puno cuenta con el equipamiento informático necesario para la implementación de un sistema de este tipo y arquitectura de manera local. Con la presente investigación se pretende implementar un sistema web con un cuadro de mando para brindar una interfaz fácil de usar para la inspección de datos, reducir los tiempos dedicados al registro de información combinando tecnologías Java para el desarrollo del sistema web y la potencia de R para el análisis de la información.



1.2 FORMULACIÓN DEL PROBLEMA

¿El desarrollo e implementación de un sistema web con cuadro de mando combinado tecnologías Java y R reducirá el tiempo empleado en el almacenamiento y procesamiento de datos y ayudará a la toma de decisiones a los colaboradores del CITE textil camélidos Puno - 2019?

1.3 OBJETIVOS DE INVESTIGACIÓN

1.3.1 Objetivo General

Desarrollar un sistema web con cuadro de mando combinando tecnologías Java y R para el almacenamiento y procesamiento de datos.

1.3.2 Objetivos Específicos

- Brindar una interfaz amigable para la inspección de los datos por parte de los colaboradores del CITE textil camélidos Puno.
- Reducir el tiempo de registro de servicios mensuales realizados por el CITE textil camélidos Puno – 2019.
- Interoperar tecnologías Java y R mediante la publicación y consumo de APIs por la transferencia de datos entre ambos lenguajes.
- Mostrar un cuadro de mando para la toma de decisiones por parte de los colaboradores del CITE textil camélidos Puno – 2019.

1.4 HIPÓTESIS DE INVESTIGACIÓN

El desarrollo del sistema web con control de mando combinando tecnologías Java y R reduce del tiempo promedio invertido en el proceso de registro de servicios brindados por el CITE textil camélidos Puno



1.5 JUSTIFICACIÓN DE LA INVESTIGACIÓN

Perú es el país que posee la mayor cantidad de alpacas a nivel mundial y Puno es el departamento con la mayor población de alpacas en Perú, por tal motivo el Ministerio de la Producción mediante el Instituto Tecnológico de la Producción y su Red de CITE promueven la innovación productiva y transferencia tecnológica a la cadena de producción de la fibra de camélidos en la Región Puno.

El CITE textil camélidos Puno tiene como objetivo brindar servicios de promoción de la investigación, capacitación, asistencias técnicas especializadas, asesorías técnicas a todas las unidades productivas dedicadas a la producción y transformación de la fibra de camélidos sudamericanos. Su trabajo está orientado al incremento de la competitividad innovando procesos para el desarrollo de productos con valor agregado. Es por este motivo que se requiere contar con una herramienta que permita el almacenamiento de la información de los servicios brindados y clientes atendidos, procese los datos obtenidos y finalmente brinde información procesada de estos servicios, unidades productivas y personal atendido en las diferentes provincias de su ámbito de intervención esto con el fin de planificar futuros servicios y reportar información mensualmente a las entidades pertinentes dentro de su organización.

Se espera que con el desarrollo que con el desarrollo e implementación del sistema web con cuadro de mando se pueda tener una visión más amplia de los servicios brindados y esto ayude en la planificación de servicios posteriores.

1.6 DELIMITACIÓN DE LA INVESTIGACIÓN

El sistema web con cuadro de mando combinando tecnologías Java y R para la administración de servicios del CITE textil camélidos Puno – 2019 se implementó y se



puso a disposición de todos los colaboradores del CITE textil camélidos Puno sobre la infraestructura de una red local dentro de las instalaciones de sus oficinas.



CAPÍTULO II

REVISIÓN DE LITERATURA

2.1 ANTECEDENTES DE LA INVESTIGACIÓN

Nautino (2015) con su sistema de apoyo en la toma de decisiones. Los sistemas de soporte a la toma de decisiones, pertenecen al paraguas de aplicaciones de Inteligencia de negocio, que pueden ser utilizados para obtener información que ayude a lograr los objetivos organizacionales. También que, el rol de este tipo de sistemas está en proveer información relevante y de manera oportuna a las personas adecuadas, para que estos puedan responder rápidamente a los cambios en el mercado o los procesos mediante la toma de decisiones de calidad.

Chambilla (2017) en su implementación y desarrollo de sistemas automatizados para mejorar la programación y ejecución de procesos en la gestión municipal, basados en arquitectura Java ee-Spring. Un sistema que sirve como apoyo para la administración de los procesos de la gestión municipal en cada fase de formulación, programación, aprobación, evaluación y ejecución del Plan Operativo Institucional de la Municipalidad Provincial de Arequipa, que permitió la mejora de los indicadores de desempeño de la municipalidad de un 40.76% en el año 2021 a un 91.25% en el año 2017.

Tapia (2017) con su análisis y diseño de un sistema de gestión de servicios para el área de servicios corporativos de la agencia de viajes BCD travel. Esta investigación resalta lo importante que es un buen análisis del negocio para que se pueda construir un sistema de gestión de servicios que se adecue a los requerimientos, necesidades de la organización, también nos dice que un sistema de gestión de servicios el área de servicios corporativos mejoró notablemente en el ejercicio de sus funciones, donde se puede afirmar la calidad



de los mismos, asimismo el área mejoró un 66% en el tiempo de los procesos involucrados.

Andragón & Palomino (2015) en su implementación de un sistema de información ejecutiva utilizando inteligencia de negocios para la eficaz interpretación de indicadores de atención y afiliación en el seguro integral de salud para la administración de la red de salud Huarochirí. La implementación de un sistema de inteligencia de negocios ha proporcionado una eficaz interpretación de indicadores en base a encuestas realizadas obteniendo un 90% de satisfacción por parte de la administración en el uso del sistema, que utilizaron la metodología híbrida de kimball y hefeosto para la generación de reportes.

Fernández (2020), en su investigación sobre las ventajas de R como herramienta para el análisis y visualización de datos en ciencias sociales. R es un lenguaje de programación empleado primordialmente para efectuar análisis estadístico de datos y construcción de gráficos. Actualmente R es considerado la lengua franca de la estadística, debido a algunas de sus características que lo sitúan muy por encima de prácticamente todos sus competidores: R es gratuito y libre, es muy versátil, permite realizar una cantidad insospechada de procedimientos estadísticos y gráficos, permite construir gráficos de calidad inmejorable.

Haro, Guarda, Zambrano & Ninahualpa (2019), en el desarrollo backend para aplicaciones web, servicios web restfull: Node.js Vs Spring Boot. Spring tiene un nivel de madurez de varios años más sobre Node.js. Ya que el desarrollo del framework se ha basado en Java; Spring es más robusto al momento de hablar de seguridad, ya que posee su propio conjunto de aplicaciones para seguridad como es Spring Security; Spring Boot tiene integración con bases de datos transaccionales con la cuales se puede realizar



aplicaciones empresariales debido a la posibilidad de utilizar las transacciones de las bases de datos para asegurar la integridad de la información.

López (2019), con su análisis, diseño e implementación de una aplicación web Java MVC para puntuación y recomendación de películas. El desarrollo utilizando Java y la tecnología Spring Boot ha permitido minimizar mucho el tiempo de puesta en marcha del proyecto y en general ha facilitado el desarrollo.

Condado (2020), con su aplicación Shiny para explorar datos de resultados electorales. El desarrollo de una aplicación web con parámetros y variables dinámicas permite dar una mayor accesibilidad de los datos, así como facilitar al usuario a que se centre en las dudas y curiosidades que tenga sobre el tema presentado en la aplicación web. Gracias a Shiny cualquier persona con el único conocimiento de programación en R puede desarrollar aplicaciones web interactivas. Una vez que se han desarrollado estas aplicaciones web cualquier usuario final puede interactuar con los datos sin necesidad de manipular el código.

Salazar, Tovar, Linares, Lozano & Valbuena (2018), en su comparación de Scrum versus XP: similitudes y diferencias. El uso de Scrum y XP en el desarrollo de proyectos introduce un conjunto de prácticas y métodos que ayudan a mejorar el rendimiento de un equipo, además de mostrar de forma notable la evolución de un producto software y su calidad; sin embargo, es necesario siempre analizar el contexto bajo el cual se desenvuelve el desarrollo del proyecto con el fin de aplicar la metodología que más se ajuste a las condiciones del medio. El uso de Scrum/XP, y en general de las metodologías ágiles, puede llegar a traer grandes beneficios, pero como todo siempre dependerá del contexto en el que se apliquen y si sus principios son bien aplicados.



Suárez (2018), en su desarrollo de un sistema informática web con la metodología ágil XP para el control de información del proceso de evaporación y batido de la panela en la productora Aprocana Norandino, 2017. La metodología Extreme Programming XP es una buena alternativa para desarrollar sistemas informáticos, ya que brinda la posibilidad de definir o actualizar los requisitos a medida que se avanza en el proyecto en base a las historias de usuarios y las pruebas dentro de cada iteración.

Villanueva & Reque (2018), con el desarrollo de un Dashboard para la toma de decisiones estratégicas en la cadena de farmacias “Belén Farma” – Ancash. El tiempo de promedio de acceso a la información fue de 132.22 segundos antes del uso del dashboard, después de la implementación del dashboard es de 32.16 segundos por lo que se obtiene una reducción de 100.07 segundos, que significa que se disminuyó en un 75.68 % el tiempo de acceso a la información. El tiempo promedio de elaboración de reportes es de 49 minutos antes del uso del dashboard, después de la implementación es de 4.29 minutos por lo que se obtiene reducción de 44.71 minutos, que significa que se redujo en un 91.24 % el tiempo de elaboración de reportes y que concluye que el desarrollo de un dashboard mejora la toma de decisiones estratégicas en la cadena de farmacias “Belén Farma”.

Elías, Rojas & Segura (2019), con la Implementación de los dashboard para mejorar la productividad de la obra: Rehabilitación de la protección del perímetro del terreno de la Universidad Nacional de Piura colindante con la margen izquierda del río Piura. La implementación de los Dashboard influye significativamente en la mejora de la productividad de la obra “Rehabilitación de la protección del perímetro del Terreno de la Universidad Nacional de Piura colindante con la margen izquierda del Río Piura. El uso de los DASHBOARD mejoró la eficiencia en la asignación de los recursos y permitió realizar el seguimiento respectivo de los costos de producción durante la ejecución de la



obra, al analizar los recursos y las partidas más incidentes de la obra nos ayudó a tomar acciones que repercutieron significativamente en la mejora de la productividad y por ende en la disminución del 10% en el costo total de la obra, ello significa una reducción de S/ 3'284,097.91 a S/ 2'997,275.46 soles.

2.2 BASE TEÓRICA

2.2.1 Lenguaje de programación Java

En el mundo de la programación y desarrollo de software tenemos varios lenguajes de programación y dentro de ellos tenemos Java que es un lenguaje de programación que tiene una gran ventaja sobre los lenguajes de programación tradicionales, la permitir crear programas para una plataforma universal. La independencia de la plataforma es un factor importante de éxito, sus programas se compilan en código de bytes y pueden ejecutarse en cualquier computador con un intérprete o Máquina Virtual Java (Vanegas & Pinzón, 2002).

Para entender mejor cómo funciona java tenemos que conocer los lenguajes compilados e interpretados.

- **Lenguajes Compilados:** son los de que requieren de un programa especial llamado compilador, que se encarga de traducir el programa fuente a su equivalente en código máquina.
- **Lenguajes Interpretados:** son aquellos que por medio de un programa llamado intérprete se convierte en lenguaje máquina y se ejecuta (Vanegas & Pinzón, 2002).



2.2.2 Kit de Desarrollo de Java (JDK)

El Kit de Desarrollo de Java o JDK es un software para los desarrolladores de Java, este incluye el intérprete de Java, clases y herramientas de desarrollo Java. Este software permite a los desarrolladores escribir aplicaciones que se desarrollan una vez y se ejecutan en cualquier lugar de cualquier máquina virtual sin la necesidad de cambiar o recompilar el código haciendo que los archivos de clase Java sean portables a cualquier máquina virtual (IBM, 2015).

Se pueden tener varios JDK de varios proveedores instaladas en un mismo ordenador, pero todos los JDK constan de una serie de componentes para realizar las tareas encargadas (Sáez Romero, 2016):

- **JRE – Intérprete en tiempo de ejecución:** hace posible que las aplicaciones Java puedan ser ejecutadas.
- **Javac – Compilador:** se encarga de compilar los archivos de código fuente Java en archivos ejecutables. Se crea un archivo de clase para cada archivo fuente.
- **Appletviewer – Visualizador de applets:** herramienta que sirva como campo de prueba para la visualización de applets, permite abrir una ventana en la que se muestran los applets.
- **Jdb – Depurador:** utilidad de línea de comandos que permite depurar aplicaciones Java, permite encontrar y eliminar errores en nuestros programas Java con mucha exactitud.
- **Javap – Desensamblador de archivos de clase:** muestra atributos y métodos públicos de una clase desensamblada, pero agregándole otras opciones desensambla código de bytes, esto es útil cuando queremos saber cómo se codificó una clase sin tener el código fuente.



- **Javah – Generador de cabeceras y archivos apéndice:** utilizado para generar archivos fuente y cabeceras en el lenguaje C para implementar métodos Java en C. Esto mediante la generación de una estructura C que coincide con la clase Java. Este generador de cabeceras crea ficheros C/C++ para implementar métodos nativos que presenta un programa Java.
- **Javadoc – Generador de documentación:** se encarga de la generación de documentación sobre la interfaz de programación al mismo estilo que la documentación que se obtiene con el JDK.
- **Applets de demostración:** código fuente de demostración que se instala de forma automática junto con el JDK que se localiza en el directorio Java que se crea durante la instalación.

2.2.3 Amazon Corretto

Amazon Corretto es una distribución sin costo, multiplataforma y lista para producción de Open Java Development Kit (Open JDK). Corretto cuenta con soporte a largo plazo que incluye mejoras de rendimiento y correcciones de seguridad y tiene varios beneficios con respecto al tradicional JDK de Oracle el Open JDK (Amazon Web Services Inc, 2021):

- **Corretto está respaldado por Amazon:** cuenta con soporte a largo plazo lo que implica que se puede actualizar a nuevas versiones cuando sea necesario. Además, Amazon ejecuta internamente en sus procesos.
- **Listo para Producción:** Corretto está certificado para cumplir con los estándares Java SE y puede ser utilizado como reemplazo a muchas distribuciones Java SE.



- **Soporte multiplataforma:** permite ejecutar el mismo entorno en la nube, en las instalaciones en su máquina local y es compatible con Linux, Windows, macOS, y Docker en su versión 8 y con Linux, Windows y macOS en su versión 11.
- **No tiene costo alguno:** a diferencia del JDK de Oracle y otras distribuciones Corretto se puede descargar y utilizar en producción sin costo alguno.

2.2.4 Java VM (Máquina Virtual de Java)

Es el núcleo del lenguaje de programación Java y es clave para muchas características principales de Java, como la portabilidad, la eficiencia y la seguridad. Siempre que se corre un programa las instrucciones que lo componen no son ejecutadas directamente por el hardware que subyace, sino que son pasadas a un elemento de software intermedio, que es el encargado de que las instrucciones sean ejecutadas por el hardware. Es decir, el código Java no se ejecuta directamente sobre un procesador físico, sino sobre un procesador virtual Java, precisamente el software intermedio del que habíamos hablado anteriormente (Méndez & Carballeira, 2000).

Debido a que en Java se pueden crear programas de tipo applet y programas independientes, la máquina virtual de Java puede encontrarse en dos formas. Si se desea ejecutar applets, ella está implícita en el navegador que se usa para visualizar, por ejemplo Netscape Navigator o Internet Explorer, para el segundo caso se requiere instalar la máquina virtual de Java que se encuentra en el Kit de Desarrollo de Java JDK (Vanegas & Pinzón, 2002). Pero actualmente la primera opción de uso está descontinuada por fallos en la seguridad que este presentaba y también porque aparecieron mejores opciones a los applet.

Las tareas principales de la máquina virtual Java son las Siguietes:



- Reservar espacio en memoria para los objetos
- Liberar la memoria no usada (Garbage Collection)
- Asignar variables a registros y pilas.
- Cargar y enlazar los archivos de clases que sean necesarios para la ejecución del programa.
- Vigilar el cumplimiento de las normas de seguridad de las aplicaciones Java.

2.2.5 Spring Framework

Spring framework es un marco de desarrollo de código libre para la plataforma Java, por lo tanto, cualquier sistema operativo con una máquina virtual de Java puede ejecutar aplicaciones desarrolladas en este framework. Su aspecto modular lo hace flexible y configurable para cualquier tipo de aplicación (Gutierrez Faraoni, 2015). Spring Framework se encarga del manejo de toda la infraestructura que actúa de soporte para desarrollar aplicaciones Java, uniendo los componentes de la aplicación, manejando sus ciclos de vida y encargándose en la interacción entre ellos(Sandoval Unapucha, 2019).

Spring maneja como dos conceptos fundamentales:

- **Inyección de Dependencia (DI):** este patrón de diseño permite suministrar objetos a una clase (POJO) que tiene dependencias.
- **Programación Orientada a Aspectos (AOP):** es un paradigma de programación que permite modularizar las aplicaciones y mejorar la separación de responsabilidades entre módulos (Chambilla Quispe, 2017).

2.2.6 Spring Boot

Spring Boot es un marco de desarrollo para Java basado en el patrón Modelo Vista Controlador que gracias a los componentes y librerías que brinda hace más fácil el



desarrollo y despliegue. Una de sus principales características es que nos permite crear aplicaciones Spring independientes, al poseer servidores de aplicaciones embebidos como Tomcat, Jetty o Undertown, así mismo ya no es necesario la configuración de archivos XML, esto facilita y hace más rápida la integración con otros proyectos de Spring Framework (Haro et al., 2019).

Las características que hacen a Spring Boot tan importante y exitoso son las siguientes (Martín Escabia, 2018):

- Módulos de configuración y validación que facilitan y aportan configuraciones necesarias al desarrollador frente a distintas situaciones, esto se conoce como Convención sobre Configuración.
- Los desarrolladores solo indicaran las dependencias necesarias para el proyecto incluso sin la necesidad de especificar la versión de estos, Spring Boot se encargará de hacer el análisis de las dependencias del proyecto esto gracias a su módulo de Gestión de Dependencias.
- Spring Boot posea cuenta con Auto-configuración que realizar configuraciones de manera autónoma ahorrando tiempo de esta tarea a los desarrolladores.
- Cuenta con un módulo que ofrece soporte para tomcat, Jetty, Undertown como servidores embebidos.
- Gracias a sus actuadores tenemos accesos a una serie de endpoints que brindan información relevante sobre la infraestructura interna de la aplicación como los beans, metrics y health, etc.

2.2.7 Base de Datos Relacional

Una base de datos relacional es una colección de tablas que cuenta con un conjunto de atributos (columnas y campos) que pueden contener un montón de tuplas que representa



un objeto, que se caracteriza por poseer una clave única o primaria que lo identifica (Yapuchura Platero, 2011). En las bases de datos relacionales se comparten datos, es decir se crean relaciones entre las tablas y a cada relación se le asigna un nombre único (Gil et al., 2018).

Entre las principales ventajas de las bases de datos relacionales tenemos (Cordova Espinoza & Cuzco Sarango, 2013):

- Se obvia la utilización de papel como una fuente de almacenamiento primario de información.
- La actualización de la información en este tipo de base de datos es más rápida
- Se dispone de la información de manera precisa.
- Se limita bastante la redundancia de información durante el ingreso de datos.

2.2.8 Maria DB Server

MariaDB Server es uno de los servidores de base de datos más populares del mundo desarrollado como software de código abierto y destinado a bases de datos relacionales que proporciona una interfaz SQL para acceder a los datos que este almacena que en sus últimas versiones incluye funciones GIS y JSON (MariaDB Foundation, 2021).

MariaDB no solamente heredó todo lo bueno de MySQL, sino que también con el pasar de los años ha obtenido funciones adicionales, mejoras en su rendimiento, mejores pruebas y menores fallos, además de una creciente comunidad por lo que se convierte en una buena opción por sobre MySQL (Bartholomew, 2012).



Figura 1: Logo MaríaDB Foundation

Fuente: <https://mariadb.org/>

2.2.9 R

Dentro de los lenguajes de programación tenemos a R que según (Muños Alonso, 2019) también es un entorno de programación más usados en la comunidad científica para la computación estadística y gráficos que es distribuida bajo la licencia GNU que lo convierte en un producto de código abierto.

Este poderoso lenguaje es interpretado ya que los comandos son ejecutados sin la necesidad de construir un ejecutable y esto facilita enormemente el trabajo de análisis complejos. Desde un punto de vista de programación R es un lenguaje orientado a objetos (Condado Gómez, 2020).

Este software posee varias ventajas para el análisis de datos (Muños Alonso, 2019)

- Es de libre acceso ya que al ser de código libre se puede utilizar efectuar ningún pago.
- Al tener una comunidad amplia posee paquetes que te permiten realizar técnicas estadísticas básica, avanzadas y robustas.
- Posee paquetes que permiten generar gráficos de alta calidad y exportables.
- Posee buena documentación sobre sus funciones y paquetes.



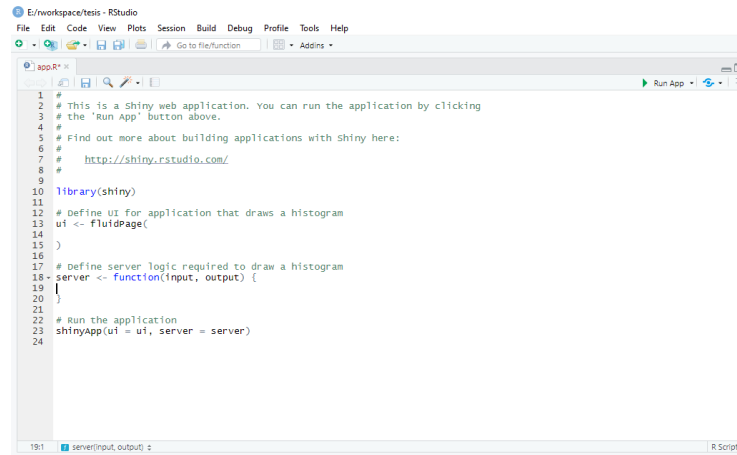
- Es multiplataforma ya que existen instaladores para varias plataformas.
- Tiene la capacidad de ser interoperable con otros lenguajes de programación.
- Cuenta con el paquete Shiny que permite la creación de aplicaciones web interactivas.
- Disponibilidad de miles de paquetes en su repositorio CRAN que extienden sus funciones básicas.

2.2.10 Shiny

Dentro de los múltiples paquetes que nos ofrece R tenemos Shiny que facilita el desarrollo de aplicaciones web reactivas e interactivas utilizando el lenguaje de programación R, que nos posibilita crear aplicaciones web independientes o incrustarlas dentro de documentos R Markdown o crear paneles de control, también permite ampliar sus posibilidades agregando temas CSS, htmlwidgets y acciones Javascript (RStudio, 2020).

Este paquete se encuentra en el repositorio CRAN de R desde el año 2012, para instalar este paquete se utiliza una sintaxis similar a la del resto de paquetes a través de los menús de RStudio o ejecutando las ordenes correspondientes en la consola (Muños Alonso, 2019).

Según (Condado Gómez, 2020) los analistas o desarrolladores de una aplicación Shiny solo deben concentrarse en lo que se mostrará en la pantalla y los resultados cada que los usuarios realicen algún cambio y no en generar código HTML o Javascript. La estructura básica de una aplicación Shiny se muestra en la siguiente imagen.



```
1 #
2 # This is a shiny web application. You can run the application by clicking
3 # the 'Run App' button above.
4 # Find out more about building applications with Shiny here:
5 #
6 # http://shiny.rstudio.com/
7 #
8 #
9
10 library(shiny)
11
12 # Define UI for application that draws a histogram
13 ui <- fluidPage(
14   )
15 )
16
17 # Define server logic required to draw a histogram
18 server <- function(input, output) {
19   }
20 }
21
22 # Run the application
23 shinyApp(ui = ui, server = server)
24
```

Figura 2: Estructura básica de un proyecto con Shiny

2.2.11 Patrón de diseño MVC

Un patrón de diseño describe problemas que ocurrieron una y otra vez en un entorno y luego describe el set de soluciones de estos, de esta forma estas soluciones pueden utilizarse las veces que sean necesarias sin la necesidad de que tenga que hacerse de la misma manera cada vez que se usan (Jaramillo, 2016).

Modelo Vista Controlador es un patrón de arquitectura de software que plantea separa los datos y lógica de la presentación en una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. MVC propone la construcción de 3 componentes (modelo, vista y controlador) que definen la representación de la información y por otro lado la interacción con el usuario. Este patrón se basa en la idea de reutilizar el código y separar conceptos para facilitar el desarrollo de aplicaciones y su mantenimiento (Arcos-Medina et al., 2018).

- **Modelo:** conjunto de clases que representa la información con la que trabaja el usuario.

- **Vista:** representan gráficamente el modelo y brindan operaciones de los controladores a los usuarios para que puedan interactuar con el sistema. En el caso de sistemas web la vista es una página HTML con contenido dinámico.
- **Controlador:** organiza las interacciones del modelo y la vista. Recibe las peticiones, realiza las operaciones necesarias para hacer consultas y modificaciones y decide la vista que mostrará los resultados al usuario o también delega tareas a otro controlador.

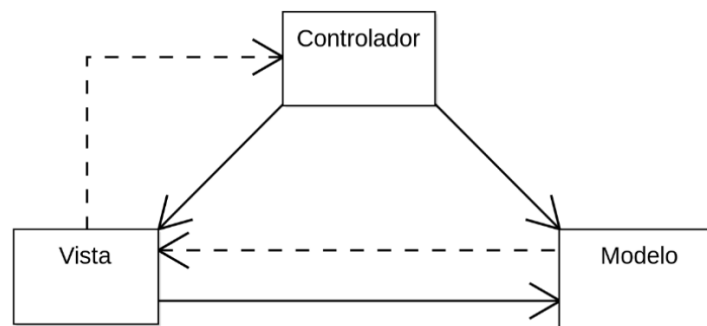


Figura 3: Patrón de diseño MVC (Modelo, Vista, Controlador)

Fuente: Wikipedia

2.2.12 Sistema web

También conocidos como aplicaciones web, son aquellos que están creados y no se instalan sobre un sistema operativo específico para su ejecución, sino que se están alojados en un servidor en internet o dentro de una red local (intranet), su aspecto es similar a una página web pero que ofrece mayor funcionalidad y respuestas a casos muy particulares. Este tipo de sistemas puede ser ejecutado desde un navegador web sin que sea relevante el sistema operativo en que este se haya instalado. Entre sus ventajas sobre otro tipo de software podemos mencionar las siguientes (Castillo Peña, 2018):



- Compatibilidad multiplataforma ya que no es necesario instalarla en alguna plataforma y por ende no es necesario crear instaladores para cada sistema operativo.
- Las actualizaciones sobre la aplicación se ven reflejadas para todos los usuarios sin requerir que estos tomen acciones pro-activas y sin interferir en sus hábitos de trabajo.
- Su acceso es inmediato sin la necesidad de compilar para un sistema operativo específico, también no es necesario descarga ni instalar algún ejecutable en los equipos de los usuarios pues estos solo deben acceder con sus cuantas mediante una URL.
- Reduce la cantidad de requisitos de hardware en los equipos de los usuarios finales ya que toda la infraestructura necesaria está implementada en los centros de datos que albergan al servidores que ejecuta la aplicación web.
- Reduce los errores y fallas por problemas técnicos o de compatibilidad con otras aplicaciones.
- Las aplicaciones web pueden atender a múltiples usuarios de manera concurrente al mismo tiempo sin la necesidad de crear instantáneas cuando varios usuarios están viendo o editando el mismo documento de manera conjunta.
- La información está más segura ya que los centros de datos donde se encuentran los servidores cuentan con infraestructura y herramientas que ayudan a evitar riesgos de perder la información por fallos de disco, virus u otras amenazas.

2.2.13 Cuadro de mando

También conocidos como Tableros de Control y Dashboard en Inglés (Haro Valle, 2018), es una representación visual de la información más importante y necesaria para lograr



uno o más objetivos, de forma consolidada y organizada en una sola pantalla, estos son utilizados cada vez con más frecuencia en la inteligencia de negocios (BI) para mejorar la logística en servicios de atención a clientes, analizar impacto de ventas, monitorear el comportamiento del personal en relación a la productividad (Robalino, 2017).

Según (Haro Valle, 2018), al momento de diseñar un cuadro de mando debemos evitar los siguientes errores:

- Exceder los límites de una sola pantalla.
- Proveer el contexto inadecuado de los datos.
- Visualizar con excesivo detalle o precisión.
- Elegir una medida deficiente.
- La elección de gráficos inapropiados.
- Incluir alguna complejidad sin sentido.
- Uso de medios de visualización mal diseñados.
- Mostrar datos cuantitativos inexactos
- La organización pobre de los datos.
- Destacar los datos importantes erróneamente o no resaltarlos en absoluto.
- Saturación o demasiada decoración.
- El mal uso o el uso excesivo del color.
- Diseñar una visualización poco atractiva.

El mismo autor también destaca los siguientes beneficios de la implementación de un cuadro de mando.

- Brinda información global de la organización para la toma de decisiones.



- Permite detectar desviaciones de los objetivos del plan estratégico e incluso descubrir la causa y así poder corregirla.
- Clarifica las acciones de corto y largo plazo.
- Comunica los planes de la organización para encaminar los recursos y así evitar su dispersión.

Finalmente podemos agregar que existen tres tipos de cuadros de mando: los cuadros de mando operacional, cuadros de mando Estratégico/Ejecutivo y el cuadro de mando analítico (Robalino, 2017).

2.2.14 Servidor de Aplicaciones

Servidor de aplicaciones viene relacionado al concepto de sistemas distribuidos que permiten 3 aspectos fundamentales en una aplicación con respecto a sistemas monolíticos como son alta disponibilidad, escalabilidad y mantenimiento (Balbuca Ramones & Ortíz Ramírez, 2017), estos están especializados en la distribución de cierto tipo de aplicaciones que se ejecutan sobre un navegador web pero a diferenciad estas estas aplicaciones son distribuidas en varias capas (Vera Zegarra, 2018):

- **Presentación:** es la interfaz gráfica que se muestra al usuario por medio de un navegador web.
- **Acceso a datos:** pare especializada encargada del acceso a los datos ejecutando procedimientos coordinados con el sistema de gestión de base de datos.
- **Lógica de negocio:** es la parte de la aplicación encargada de ejecutar par que la información sea distribuida a cada cliente de forma eficaz.

Según (Muñoz Muñoz, 2015), las características de los servidores de aplicaciones son los siguientes:



- Cliente web: que responde a las solicitudes HTTP que en su mayoría vienen de un navegador web y responde HTML, XML.
- Aplicación Cliente: similares a un navegador web que actúan como cliente, pero estas no se ejecutan en un navegador y la tecnología que utilizan varía según el contenedor web o la base de datos.
- Contenedor Web: es el mismo servidor web que utiliza protocolos HTTP y SSL para poder comunicarse.
- Servidor de Aplicaciones: encargado de proveer servicios que soportan la disponibilidad y ejecución de las aplicaciones.

2.2.15 API Rest

REST (Representational State Transfer) es un estilo de arquitectura de software para sistemas hipermedia distribuidos, este término fue introducido en la tesis doctoral de Roy Fielding en el año 2000. REST es una colección de recursos definidos y diseccionados que a menudo es utilizado para describir a cualquier interfaz que transmite datos específicos de un dominio sobre los protocolos HTTP (Monago Ruiz, 2019).

La API REST son Interfaces de programación de aplicaciones que se basan en REST, estos no son un estándar, pero se basan también en algunos estándares HTTP y URL y que sumado a JSON como formato ligero de intercambio de datos han acelerado considerablemente el crecimiento de las API REST que en la actualidad es soportado por todos los lenguajes de programación existentes y cuenta con las siguientes características (López Mollinedo, 2019):

- Protocolo cliente/servidor sin estado
- Operaciones Importantes
- URI de identificación único de errores

- Interfaz uniforme
- Sistema de capas
- Uso de hipermedios

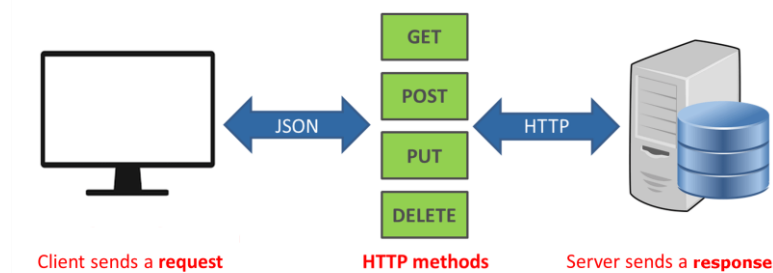


Figura 4: Funcionamiento de las API Rest

Fuente: <https://aprendiendoarduino.wordpress.com/2019/10/27/api-rest/>

2.2.16 Metodología de desarrollo ágil

En febrero del 2021 en Utah - EEUU luego de la reunión de 17 expertos de la industria del software nace el término ágil. Su objetivo fue esbozar los valores y principios que debería permitir a los equipos de desarrollar software rápidamente y responder a los cambios que puedan surgir a lo largo del proyecto (Penadés & Letelier Torres, 2006). Estos valores y principios se plasman en el manifiesto ágil.

Manifiesto ágil

Según (López Menéndez de Jiménez, 2015), este es un documento que engloba principios y valores de un proyecto de desarrollo de software ágil de uno en su forma tradicional y donde se valora:

- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.

- Desarrollar software que funcione más que la documentación del mismo.
- La colaboración con el cliente más que la negociación de su contrato.
- Responde a los cambios más que seguir con el plan establecido.

Esta metodología también está regida por doce principios que ayudan en que el proceso de desarrollo sea menos complejo y responda más oportunamente a los cambios que puedan surgir siempre contando con punto de vista del cliente.



Figura 5: Proceso ágil de desarrollo de software

Fuente: Metodologías Ágiles de Desarrollo de Software Aplicadas a la Gestión de Proyectos Empresariales (2015).

2.2.17 Metodología de desarrollo XP

XP es una metodología ágil que se centra en las relaciones interpersonales como parte del éxito del desarrollo de software, promoviendo el trabajo en equipo preocupándose por el aprendizaje de los desarrolladores propiciando un buen clima de trabajo (Penadés & Letelier Torres, 2006). XP fue introducida por Kent Beck también es especialmente



adecuada para proyectos con requisitos imprecisos y muy cambiantes y donde existe un alto riesgo técnico, las características principales de XP son las siguientes (Salazar, 2007):

- **Las historias de los usuarios:** técnicas utilizadas para especificar los requisitos del software, en estas se describe brevemente las características funcionales y no funcionales del software. Las historias deben ser comprensibles y delimitadas para que los desarrolladores puedan implementarla en poco tiempo.
- **Roles XP:** de acuerdo con la propuesta de Beck:
 - **Programador:** escribe le código del sistema y las pruebas unitarias.
 - **Cliente:** escribe las historias de usuario además de hacer las pruebas funcionales y no funcionales además de dar prioridad a las historias de usuarios y decide cuales se implementan en cada iteración.
 - **Tester:** ayuda al cliente a escribir las pruebas funcionales, ejecuta pruebas regularmente y difunde los resultados.
 - **Tracker:** proporciona realimentación al equipo, verifica el grado de acierto entre las estimaciones realizadas y el tiempo dedicado para futuras estimaciones y realiza el seguimiento en cada iteración.
 - **Coach:** encargado del proceso global, provee guías al equipo de forma que apliquen prácticas de XP y sigan el proceso correctamente.
 - **Consultor:** miembro externo al equipo de desarrollo que tiene conocimientos específicos en temas necesarios para el proyecto en el que puedan surgir problemas.
 - **Gestor:** vínculo entre clientes y desarrolladores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas para el trabajo.
- **Procesos XP:** a grandes rasgos consiste en los siguientes pasos:
 - El cliente define los valores de negocio a implementar



- El programador estima el esfuerzo necesario para su implementación.
 - El cliente selecciona qué construir de acuerdo a sus prioridades y restricciones de tiempo.
 - El desarrollador construye ese valor de negocio
 - Vuele al paso 1.
- **Prácticas XP:** la posibilidad de disminuir la curva exponencial del costo del cambio a lo largo del proyecto se consigue gracias a las tecnologías para ayudar en el desarrollo y a la aplicación disciplinaria de las siguientes prácticas:
 - El juego de la planificación
 - Entregas pequeñas
 - Metáfora
 - Diseño simple
 - Pruebas
 - Refactorización
 - Programación en parejas
 - Propiedad colectiva del código
 - Integración continua
 - 40 horas por semana
 - Cliente in-situ
 - Estándares de programación

2.2.18 Desarrollo de software

El desarrollo de software tiene como objetivo la elaboración de un producto que reúna los requisitos del cliente, el desarrollo de software es un proceso intelectual, determinado



por la creatividad y juicio del equipo multidisciplinario que participa en la planificación y ejecución (Esteller & Medina, 2012), y los modelos más conocidos tenemos:

- Diseñar y corregir
- Modelo de cascada
- Desarrollo evolutivo
- Desarrollo basado en reutilización
- Desarrollo incremental
- Desarrollo en espiral

2.2.19 Pruebas unitarias

Las pruebas unitarias un fragmento de código que prueba la funcionalidad deseada de una unidad de código que comprende un módulo más complejo, función, procedimiento completo o una interfaz completa como una clase de un objeto, dentro de sus características podemos mencionar las siguientes (Asimbaya, 2019):

- Debe probar solo pequeñas cantidades de código
- No debe ser dependiente ni afectar el funcionamiento de otras pruebas unitarias
- Debe probar métodos públicos de lo contrario sería frágil a los cambios en la implementación.
- Debe dar la posibilidad de ser automatizada.
- Debe ser repetible y predecible ya que todas las veces que se repita la prueba el resultado siempre debe ser el mismo.
- Deben ser consideradas igual que el código, con la misma profesionalidad, documentación.



Según (Chela LLumiguano, 2019), el propósito de utilizar las pruebas unitarias es aislar cada sección de código y verificar si devuelven resultados correctos, y las pruebas unitarias nos pueden ofrecer los siguientes beneficios si los utilizamos en nuestro proceso de desarrollo de software:

- Determinar las especificaciones.
- Provee una temprana detección de errores
- Mantenimiento de soporte.
- Mejorar el diseño.
- Documentación del producto.

2.2.20 JUnit

Según (Peñarrocha Planells, 2016), JUnit es un conjunto de bibliotecas que es empleada al momento de programar para realizar pruebas unitarias en aplicaciones Java. Este ejecuta clases para comprobar si funcionan correctamente, JUnit toma un valor de entrada y evalúa el valor de retorno, si la clase cumple con la especificación indicada devolverá que la prueba se realizó con éxito, en caso contrario, devolverá el fallo de esta.

Actualmente tenemos JUnit 5 cuyo objetivo es crear una base actualizada para las pruebas del lado del desarrollador de la Java Virtual Machine (JVM), que incluye centrarse en Java 8 y versiones posteriores y habilitar muchos estilos de prueba (The JUnit Team, 2019).



Figura 6: Logo de JUnit 5

Fuente: <https://junit.org/junit5/>

2.2.21 Contenedores de Software

Puede considerarse a un contenedor como un entorno virtual aislado que incluye un conjunto de dependencias específicas necesarias para la ejecución de una aplicación, los contenedores aparte de ser entornos aislados comparte el kernel del sistema que lo aloja y dependiendo de la configuración también puede estar aislado de otros contenedores (Hurtado Chichande, 2018).

Hablando de escalabilidad y rendimiento comparando los contenedores con las máquinas virtuales tradicionales, los contenedores son mejores pues pueden ejecutar varios sistemas operativos sin generar una sobrecarga sobre los recursos disponibles. Algunos de sus principales beneficios son los siguientes (Iñiguez, 2017):

- **Menos consumo de recursos:** los contenedores comporten los recursos con el sistema anfitrión y por eso las acciones de arranque y parada se producen en segundos, esto también aumenta la velocidad de las aplicaciones que se ejecutan en contenedores.
- **Portabilidad:** los contenedores eliminan errores causados por los cambios de entornos de ejecución y la dependencia de proveedores de estos.

- **Ligero:** los contenedores son más livianos y permiten que los desarrolladores ejecuten varios contenedores haciendo posible emular entornos de producción además de que los ingenieros de operación pueden correr más contenedores en un solo anfitrión que utilizando máquinas virtuales.

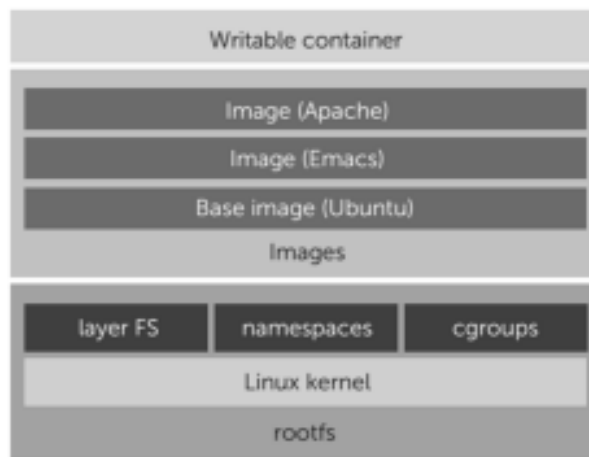


Figura 7: Arquitectura de una imagen Docker

Fuente: Pahl (2015).

2.2.22 Docker

Según lo expuesto por (Pol Ponsico, 2017), Docker es un proyecto open source que permite crear contenedores de software ligeros, portátiles y autosuficientes obviando la utilización de máquinas virtuales e hipervisores además de que el código que corre sobre estos contenedores está totalmente aislado y comparten los recursos de la máquina con otros contenedores sin la sobrecarga de los hipervisores. Docker se compone de los siguientes elementos:

- Docker daemon
- Docker CLI
- Docker Hub

- Imágenes
- Contenedores.

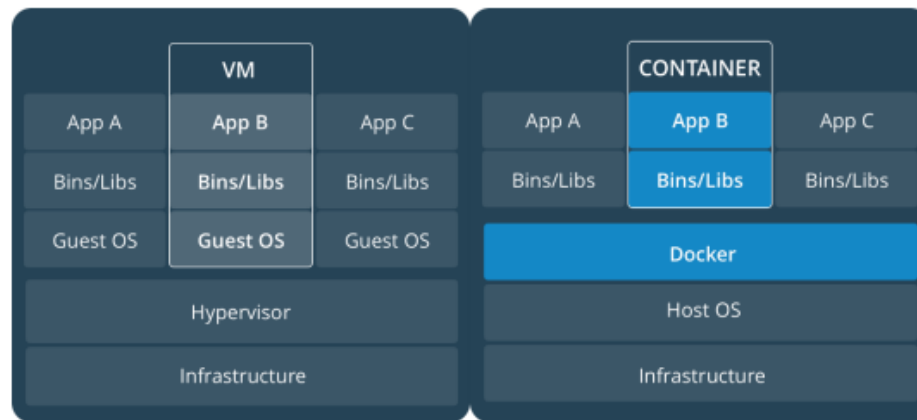


Figura 8: Comparación entre una máquina virtual y un contenedor Docker

Fuente: Pol Ponsico (2017)

También podemos acotar que según (Fornés de Juan, 2021) esta tecnología facilita la automatización del despliegue de aplicaciones en contenedores donde estas se virtualizan y ejecutan, además de añadir la característica de poder probar el código antes de ponerlo en producción.

2.2.23 Sistema de control de versiones

Se conoce como control de versiones a la gestión de cambios que se realizan entre elementos de algún producto o configuración del mismo. Se denomina versión, revisión o edición al estado en el que se encuentra el producto en un momento dado de su ciclo de vida. La gestión de versiones puede realizarse de forma manual pero lo recomendable es utilizar herramientas que faciliten esta tarea, estas herramientas son llamadas sistema de control de versiones (Guerrero Higuera et al., 2017).

Los sistemas de control de versiones se encargan de guardar en repositorios las versiones de un software generadas durante su desarrollo, esto representa una ventaja para los

desarrolladores pues estas versiones pueden estar disponibles para más participantes puedan participar en el desarrollo. También se encarga de almacenar la fecha y hora de los cambios y datos del desarrollador que realizó las modificaciones(Tello Leal et al., 2012). A continuación, se muestra un gráfico del funcionamiento de los sistemas de control de versiones:



Figura 9: Funcionamiento de un sistema de control de versiones

Fuente: Tello Leal et al (2012)

2.2.24 Git

Los sistemas de control de versiones ayudan en el proceso de desarrollo de software pues nos permiten almacenar, recuperar, compartir y fusionar versiones de código fuente, Git es uno de estos sistemas de control de versiones que permite a cada colaborador tener una copia local del proyecto y poder trabajar independientemente nuevas características (Aniceto Caba, 2020). Además de ser uno de los sistemas de control de versiones Git sigue un sistema de trabajo distribuido y sigue un modelo cliente servidor y cada copia de un repositorio guarda información completa del proyecto y sus cambios (Guerrero Higuera et al., 2017).

Debemos considerar algunos conceptos básicos a la hora de trabajar con Git (Aniceto Caba, 2020):

- **Repositorios:** es la base desde donde los desarrolladores pueden descargar copias del proyecto, esta puede estar en la nube o servidores propios. También almacena todas las versiones del proyecto en el que se trabaja.
- **Commit:** es un comando que confirma cambios en los ficheros, este comando también crea una nueva versión del código del proyecto.
- **Rama:** es una versión del proyecto que está etiquetada que hace referencia a cambios y características nuevas del proyecto, una rama comienza con un commit, las ramas se encuentran aisladas de otras y estas pueden contener varios commit. La ventaja de las ramas es que permiten el trabajo paralelo de los desarrolladores.
- **Merge:** es el comando en Git que nos permite fusionar ramas ya que es recomendado tener una versión principal a la que se le van añadiendo nuevas funcionalidades desarrolladas en otras ramas.
- **Push:** es el comando que nos permite enviar cambios en nuestro código a un repositorio remoto.

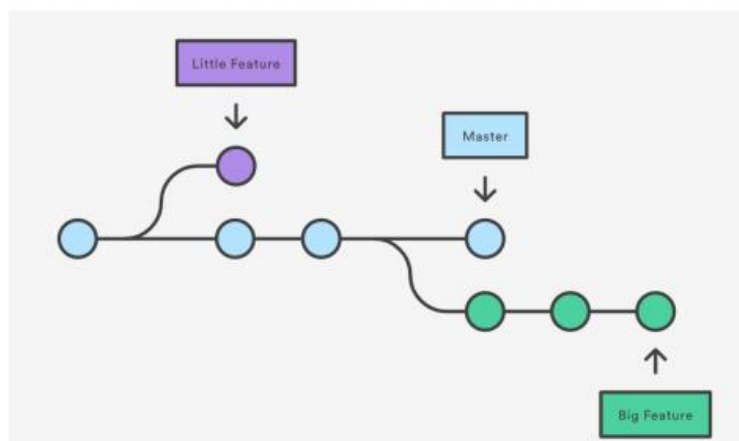


Figura 10: Representación gráfica de ramas y commits

Fuente: Aniceto Caba (2020)

2.2.25 Git Hub

GitHub es una plataforma de desarrollo colaborativo de software que aloja los proyectos que utiliza el sistema de control de versiones Git. Github fue concebida como una herramienta que proporciona por un lado el repositorio para el código de los proyectos y por las otras herramientas para el trabajo colaborativo (Roig-Vila et al., 2017).

Además según (Lagüera Cabrera, 2020), GitHub llena el vacío de tener los repositorios de software almacenados de manera local y los almacena en sus servidores facilitando la colaboración y así muchas máquinas pueden tener acceso a versiones anteriores del proyecto y a nuevas características que quieran implementarse.

2.2.26 Estadístico de Prueba U de Mann Whitney

En muchas ocasiones los datos obtenidos en una investigación no se pueden comparar utilizando pruebas paramétricas por lo que es necesario hacerlo utilizando pruebas no paramétricas para comparar las medias o medianas, la prueba U de Mann Whitney sería la prueba equivalente a la prueba paramétrica T de Student (Romero, 2013) para datos no paramétricos.

$$U_1 = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - R_1$$
$$U_2 = n_1 n_2 + \frac{n_2(n_2 + 1)}{2} - R_2$$

Figura 11: Fórmula U de Mann-Whitney

Fuente: Wikipedia



CAPÍTULO III

MATERIALES Y MÉTODOS

3.1 UBICACIÓN GEOGRÁFICA DEL ESTUDIO

El desarrollo de la presente investigación se ubica en el Departamento de Puno, Provincia de Puno y Distrito de Puno donde se ubica la oficina del Centro de Innovación Productiva y Transferencia Tecnológica textil camélidos Puno.

3.2 POBLACIÓN

Nuestra población de estudio está integrada por todos los colaboradores del Centro de Innovación Productiva y Transferencia Tecnológica textil camélidos Puno en el año 2019 que en este periodo son 8 colaboradores.

3.3 DISEÑO DE LA MUESTRA

Para el desarrollo de esta investigación no fue necesario realizar un proceso de muestreo ya que se tuvo la posibilidad de trabajar con toda la población objetivo que se beneficiará con la implementación del sistema por lo que para esta investigación contamos con una población de 8 colaboradores.

3.4 MÉTODO DE RECOPIACIÓN DE INFORMACIÓN

Para la recopilación de información de los requerimientos que debe de cumplir el trabajo de investigación se realizaron entrevistas con todos los colaboradores del Centro de Innovación Productiva y Transferencia Tecnológica textil camélidos Puno.

Para validad la reducción del tiempo antes y después de la investigación se realizaron paralelamente el registro de información de forma manual y empleado el sistema, esto con el fin de comparar el tiempo medio empleado en cada caso.



Contrastación de Hipótesis

La contrastación de hipótesis se realizó con la prueba no paramétrica U de Mann-Whitney que nos permitirá rechazar o aceptar nuestra hipótesis de investigación, esto se realizó a los tiempos empleados en el registro y generación de reportes antes y después de la investigación, nuestros enunciados son los siguientes:

Hipótesis nula (H₀)

El desarrollo del sistema web con control de mando combinando tecnologías Java y R no reduce del tiempo promedio invertido en la administración de servicios brindados por el CITE textil camélidos Puno

Hipótesis alternativa (H_a)

El desarrollo del sistema web con control de mando combinando tecnologías Java y R reduce del tiempo promedio invertido en la administración de servicios brindados por el CITE textil camélidos Puno

3.5 METODOLOGÍA DE DESARROLLO DE SOFTWARE

Metodología de Desarrollo

En la implementación y desarrollo del sistema web con cuadro de mando combinando tecnologías Java y R para la administración de servicios del CITE textil camélidos Puno – 2019 se empleó la metodología ágil XP pues esta se adecúa a las necesidades y requisitos de desarrollo del sistema y permite hacer cambios pequeños durante el desarrollo y así responder de forma rápida a cualquier cambio que pueda surgir. Las fases seguidas para el desarrollo del sistema fueron las siguientes:



Planificación

Tomamos referencia de las historias de los colaboradores que participan en los procesos de la organización para esto se realizó entrevistas con cada uno de ellos para que nos describan sus necesidades esto se realizó varias veces ya que se revisó estos requisitos en varios periodos.

Diseño

Se diseñó los módulos y funciones que cubran los requerimientos de los usuarios, esto con código orientado a objetos y designando los recursos necesarios para que estos funcionen.

Codificación

Según el diseño se procedió a codificar los módulos y funciones, siempre haciendo consulta a los colaboradores para que este pueda cumplir con sus requerimientos de forma más precisa.

Pruebas

En esta fase se probaron los módulos y las funciones escritas en código para validar los resultados que estos arrojan, para este fin se utilizó JUnit 5 por su buena integración con proyectos Java y su amplia gama de herramientas disponibles para el testeo.

3.6 MATERIAL EXPERIMENTAL

Para el desarrollo de esta investigación los materiales fueron los siguientes:



3.6.1 Hardware

Por el tiempo y cambio de ubicaciones geográficas se utilizaron una PC de escritorio con CPU Ryzen 2700X de 8 núcleos y una laptop con procesador Intel Core I5 de segunda generación.

3.6.2 Software

Se aplicaron varias herramientas de software que puedan cumplir una independencia de plataforma los que se listan a continuación:

- Amazon Corretto
- Xampp
- MaríaDB
- Navicat
- Spring Tool Suite
- RStudio
- R Software
- Git y GitHub
- Docker
- Shiny
- Google Chrome



CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

Para el desarrollo del sistema web con cuadro de mando combinado tecnologías Java y R para el almacenamiento y procesamiento de datos se realizó el siguiente proceso considerando la literatura revisada.

4.1 ANÁLISIS

En el desarrollo del sistema web con cuadro de mando, combinando tecnologías Java y R para la administración de servicios del CITE textil camélidos Puno – 2019. Se desarrolló siguiendo las pautas que nos da la metodología de programación XP, se complementó esta metodología con el lenguaje unificado de modelamiento (UML) para visualizar, especificar, construir y documentar el sistema.

A continuación, detallamos los análisis que se realizaron para el desarrollo del sistema web con cuadro de mando, combinando tecnologías Java y R para la administración de servicios del CITE textil camélidos Puno – 2019.

4.1.1 Viabilidad

El desarrollador cuenta con las habilidades para desarrollar el sistema además de combinar ambas tecnologías, también se contó con las herramientas y equipos necesarios para el desarrollo, el administrador del sistema contará con acceso total al sistema, podrá crear usuarios y asignar roles ya que se pretende separar las tareas de cada rol por seguridad, usabilidad del sistema y según la responsabilidad que se asignó a cada colaborador de la institución.

4.1.2 Historia del Usuario

En esta etapa se recogieron las historias de los usuarios del sistema que colaboran dentro de la institución, esta etapa fue muy flexible ya que se permitió a los usuarios expresar sus necesidades de la manera más comprensible que fueron plasmadas en requerimientos para la interpretación del desarrollador y delimitar los alcances de estas necesidades.

De esta recolección de historias de usuarios se resumió el siguiente cuadro con los requisitos del sistema:

Tabla 1

Historias de usuario recopiladas para el desarrollo del sistema

N°	Historia	Tarea a ejecutar
1	Base de datos centralizada de todos los clientes atendidos y servicios brindados para evitar duplicidades.	Diseño de la base de datos
2	El sistema debe ser accedido por todos los colaboradores según su rol en la institución	Definición de roles y accesos al sistema
3	Registrar información de las unidades productivas en tiempo más cortos.	Desarrollar módulo de registro de unidades productivas
4	Registrar al personal de las unidades productivas atendidas.	Desarrollar el módulo de registro de personal
5	Registrar los servicios brindados a las unidades productivas clientes y su respectivo personal evitando duplicidad consultando información ya registrada.	Desarrollar el módulo de digitación de servicios
6	Registrar los eventos realizados en las diferentes tareas asignadas a la institución en menos tiempo.	Desarrollar el módulo de registro de eventos
7	Generar reportes de servicios brindados mensualmente de manera rápida.	Desarrollar módulo para generar reportes requeridos
8	No debe haber duplicidad de información registrada por el sistema	Implementar validaciones en la base de datos.
9	No se debe perder información en el momento del registro de la información	Implementar transacciones en la base de datos y el sistema



10	Debe estar disponible para todos los colaboradores	Elegir herramientas de desarrollo web
11	Mostrar gráficamente información almacenada en la base de datos	Implementar un cuadro de mando (dashboard)
12	Se debe tener una interfaz de usuario amigable, fácil de entender y que sea intuitiva.	Desarrollo de interfaz con plantillas validadas.

4.1.3 Roles

De acuerdo a la metodología de desarrollo ágil XP se establecieron los siguientes roles para el desarrollo del sistema.

Programador: Este rol fue asumido por el investigador encargado de la implementación del sistema web que tiene conocimientos en la plataforma Java y entorno R Shiny.

Cliente: Fue asignado a los colaboradores de la institución encargados de programar actividades, consultar información y que requieren información procesada.

Tester: Fue asignado al investigador para escribir las pruebas unitarias para cada módulo y testear el funcionamiento en conjunto de los módulos.

Tracker: Este rol lo asumió el Asesor de la tesis quién supervisa del desarrollo del desarrollo e implementación del sistema.

Coach: fue asumido por el investigador que desarrolla el sistema siguiendo las mejores prácticas de la metodología XP y conoce el proyecto en su totalidad.

Consultor: También fue asumido por el investigador, ya que conoce todos los procesos de la institución que puedan influir en el desarrollo del sistema.



Gestor: Este rol también fue asumido por el investigador ya que recolectó información de los colaboradores de la institución (clientes) y plasma esto en mejores prácticas para el desarrollo del sistema.

4.1.4 Planificación

De acuerdo a las tareas establecidas por las historias de los usuarios se realizó la siguiente planificación de tareas:

Tabla 2

Cronograma de actividades para el desarrollo del sistema

N°	Tarea	Semanas							
		1-2	3-4	5-6	7-8	9-10	11-12	13-14	15-16
1	Diseño de la base de datos	X	X						
2	Definición de roles y accesos al sistema		X						
3	Desarrollar módulo de registro de unidades productivas		X						
4	Desarrollar el módulo de registro de personal			X					
5	Desarrollar el módulo de digitación de servicios				X				
6	Desarrollar el módulo de registro de eventos					X			
7	Desarrollar módulo para generar reportes requeridos					X	X		
8	Implementar validaciones en la base de datos.		X						
9	Implementar transacciones en		X	X					



	la base de datos y el sistema			
10	Elegir herramientas de desarrollo web	X		
11	Implementar un cuadro de mando (dashboard)		X	X

4.1.5 Análisis de requerimientos del sistema

Requerimientos funcionales

- **Autenticación de usuarios** con usuario y contraseña y los roles asignados a cada usuario.
- **Registro de nuevos usuarios** permite al administrador del sistema crear y asignar roles a nuevos usuarios del sistema.
- **Registro de unidades productivas** para que los usuarios encargados de este módulo puedan registrar información de las unidades productivas atendidas.
- **Registro de personal por cada unidad productiva** que permitirá al personal encargado del módulo registrar al personal de las unidades productivas atendidas.
- **Registro de eventos desarrollados** permite a los usuarios encargados del módulo registrar información de los eventos realizados por la institución.
- **Registro de servicios a unidades productivas** permite a los usuarios encargados de este módulo registrar los servicios brindados a cada unidad productiva.
- **Registro de servicios a personal de unidades productivas** permite a los usuarios encargados del módulo registrar información de los servicios que recibe el personal de cada unidad productiva atendida.



- **Generar reporte mensual de servicios** permite a los usuarios encargados del módulo descargar el reporte mensual de servicios en el formato correspondiente con la información registrada.
- **Consultar información de unidad productiva** permite a todos los usuarios del sistema consultar información de las unidades productivas atendidas sin la posibilidad de modificar esta información.
- **Consultar información del personal de las unidades productivas atendidas** permite a todos los usuarios del sistema consultar información del personal de las unidades productivas atendidas.
- **Mostrar información procesada** muestra a todos los usuarios del sistema información procesada en gráficos para su interpretación.

Requerimientos no funcionales

- Las contraseñas de los usuarios deben estar encriptadas.
- El sistema debe estar disponible solo en la red local de la institución.
- El sistema debe ejecutarse en un entorno aislado en el servidor.
- Debe ser escalable para abarcar nuevas solicitudes de cambio y ampliación
- Bajo acoplamiento entre los módulos del sistema
- El sistema debe de ser concurrente y poder atender a todos los usuarios a la vez
- Debe recolectar la mayor información de eventos, unidades productivas y personal.
- Debe ser seguro al momento de registrar la información y prevenir la pérdida de esta por fallas externas.

4.1.6 Análisis de requerimientos técnicos

Hardware

Tabla 3

Componentes de hardware mínimos requeridos

Componente	Mínimo requerido
Procesador	3 GHz 4 núcleos
Memoria RAM	8 GB
Almacenamiento	100 GB

Software

Para el desarrollo de la investigación se optó por utilizar software de distribución libre.

Tabla 4

Software utilizado para el desarrollo del sistema

Software	Versión	Uso
Java	8	Lenguaje de programación libre que se ejecuta sobre la JVM que nos permitirá implementar los requisitos de software.
JDK	8	Kid de Desarrollo que contiene las bibliotecas que nos facilitarán el desarrollo de con el lenguaje Java.
JRE	8	Entorno de ejecución para el lenguaje de programación Java.
WildFly	20	Servidor de aplicaciones que implementa especificaciones JEE para la ejecución de aplicaciones Java Empresarial.
MariaDB	10.6	Sistema de gestión de base de datos de uso para el almacenamiento de bases de datos relacionales.



AdminLTE	2	Panel de administración para Bootstrap de código abierto que permite la construcción de aplicaciones web.
Spring Tools Suite (STS)	4	Entorno de desarrollo integrado basado en Eclipse IDE para el desarrollo de aplicaciones Java basadas en Spring Framework y Spring Boot.
R	4	Lenguaje de programación enfocado en el análisis estadístico.
R Shiny	1.6.0	Paquete del lenguaje R para la construcción de cuadros de mando web interactivos.
R Studio	1.4	Entorno de desarrollo integrado para el lenguaje de programación R.

En el análisis para el desarrollo del sistema vimos que se estaba en la posibilidad y se contaba con los recursos para cumplir con los requerimientos establecidos y que se disponía de las herramientas software y hardware para desarrollar e implementar un sistema web con cuadro de mando combinando tecnologías Java y R para el almacenamiento de y procesamiento de datos.

4.2 DISEÑO

Siguiendo las recomendaciones de la metodología XP para el desarrollo de software y de las iteraciones realizadas que se realizaron tomando en cuenta las historias de los usuarios se logró establecer una arquitectura para el sistema que fue utilizada durante todo el proyecto de desarrollo.

Para la documentación de la arquitectura que se fue definiendo durante las iteraciones se empleó UML y se presentan los siguientes gráficos.



4.2.1 Documentación mediante UML

El Lenguaje Unificado de Modelado, UML por sus siglas en inglés, es un lenguaje gráfico que nos permitió modelar el sistema en sus diferentes procesos y así poder documentar cada uno de estos, para el sistema se realizaron los siguientes gráficos.

4.2.2 Diagramas de caso de uso

Se utilizaron los diagramas de caso de uso para describir los escenarios que tienen cada uno de los usuarios, a especificar la utilización del sistema, describir los estímulos a los que debe responder y a describir los procesos del sistema en general.

Casos de uso del sistema

En el sistema el rol más importante es el del Administrador del Sistema que tiene por tarea registrar a nuevos usuarios, y actualizar información pertinente para el correcto funcionamiento de los módulos del sistema y tiene acceso también a estos.

El coordinador es el que se encarga de programar las actividades a realizar y también es que revisa que los servicios e información de las unidades productivas atendidas en los eventos programados estén bien registradas y no haya duplicidades ni errores en la información, consulta información de unidades productivas, servicios y personal atendido. También generar el reporte mensual de servicios para validar en enviar a las oficinas correspondientes.

El digitador es el encargado de centralizar la información de las unidades productivas y su personal respectivo recabada en los eventos realizados, así como



también la de registrar eventos, MYPES y personal. También genera los reportes de servicios mensuales.

El especialista para programar sus servicios y ver la viabilidad de estos debe hacer consultas a las unidades productivas y en muchos casos conversar con el personal de estas unidades productivas, también debe consultar los resultados de los servicios brindados para hacer una mejor planificación y no duplicar servicios por zona geográfica.

El Director es quien toma las decisiones, por lo que interesa verificar reportes ya analizados de la información registrada por los demás roles y con esta información prever, hacer proyecciones y prevenir situaciones que afecten los objetivos de la institución.

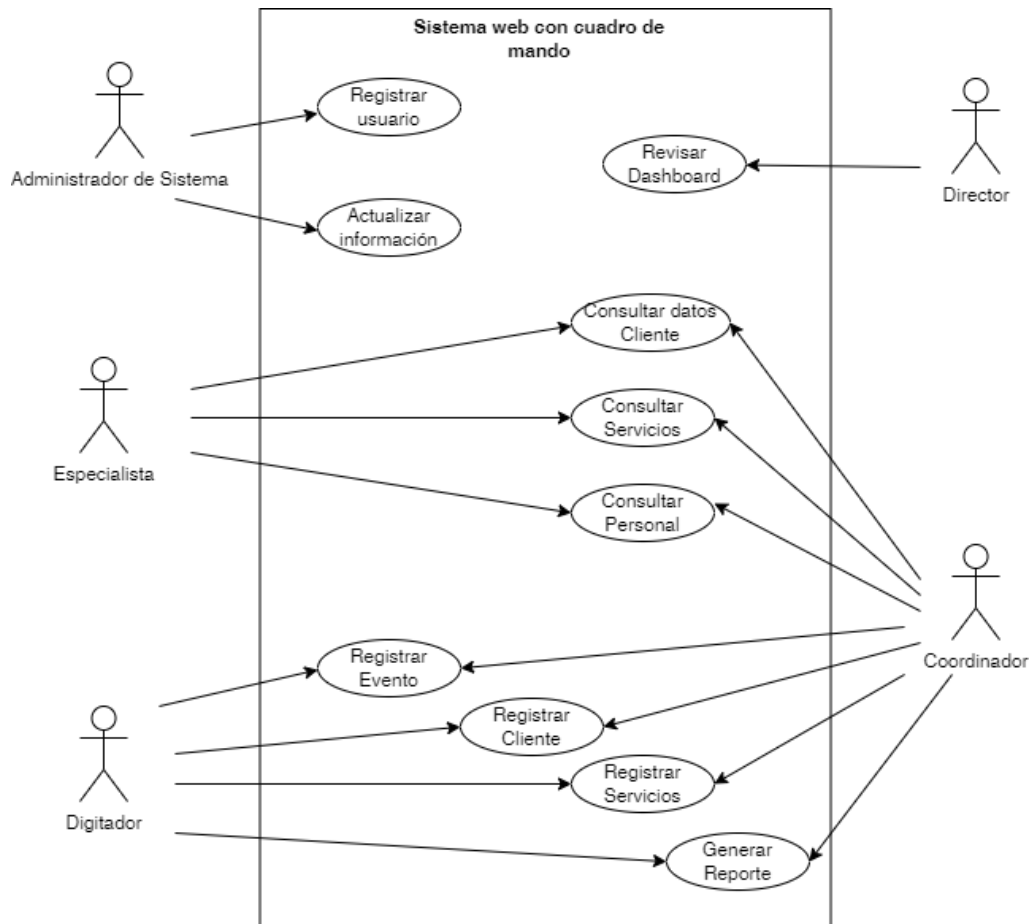


Figura 12: Diagrama de caso de uso general del sistema

4.2.3 Diagramas de secuencia

Acceso al sistema

Todos los usuarios del sistema sean cuales seas sus roles dentro de la organización deben de registrar su acceso al sistema con un usuario o contraseña proporcionada por el administrador del sistema, el sistema se encarga de validar esta información haciendo consultas a la base de datos y descriptando la contraseña del usuario para devolver una sesión, en caso los datos sean erróneos devuelve un mensaje de fallo en la autenticación.

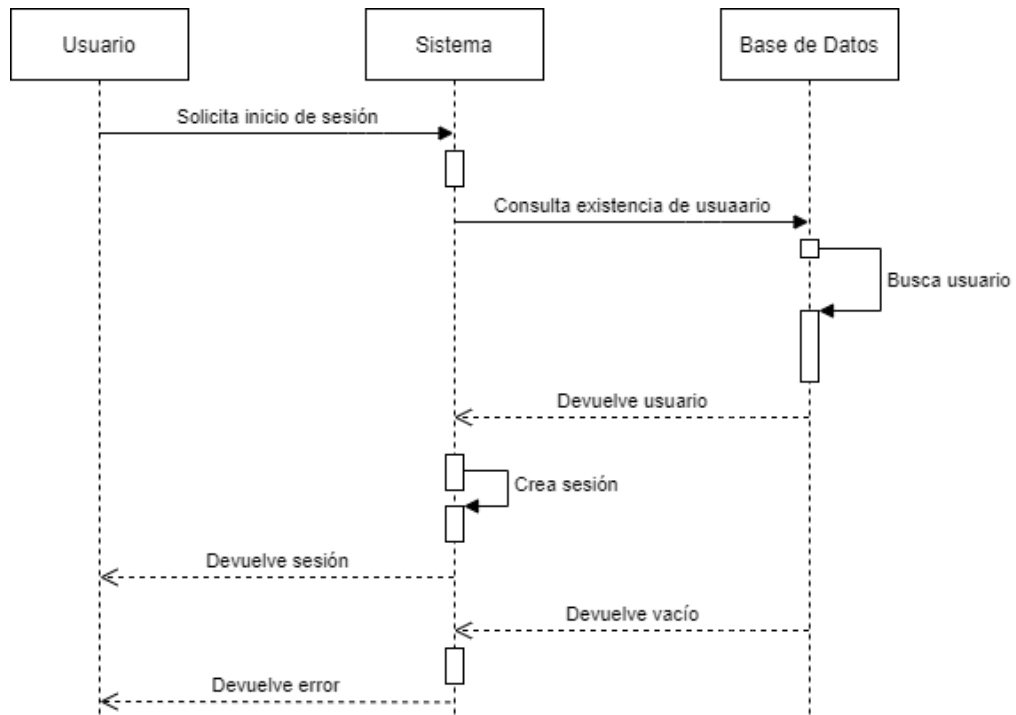


Figura 13: Diagrama de secuencia de acceso al sistema

Registro de Unidad productiva

Son varios roles de usuarios los que tienen acceso al módulo de registro de MYPE cliente pero la lógica de registro es la misma, el usuario luego de iniciar sesión en el sistema ingresa la información en un formulario del sistema, el sistema valida la información y envía la información a la base de datos mediante una transacción, la base de datos dará un commit o un rollback según haya sido el resultado de la operación de persistencia.

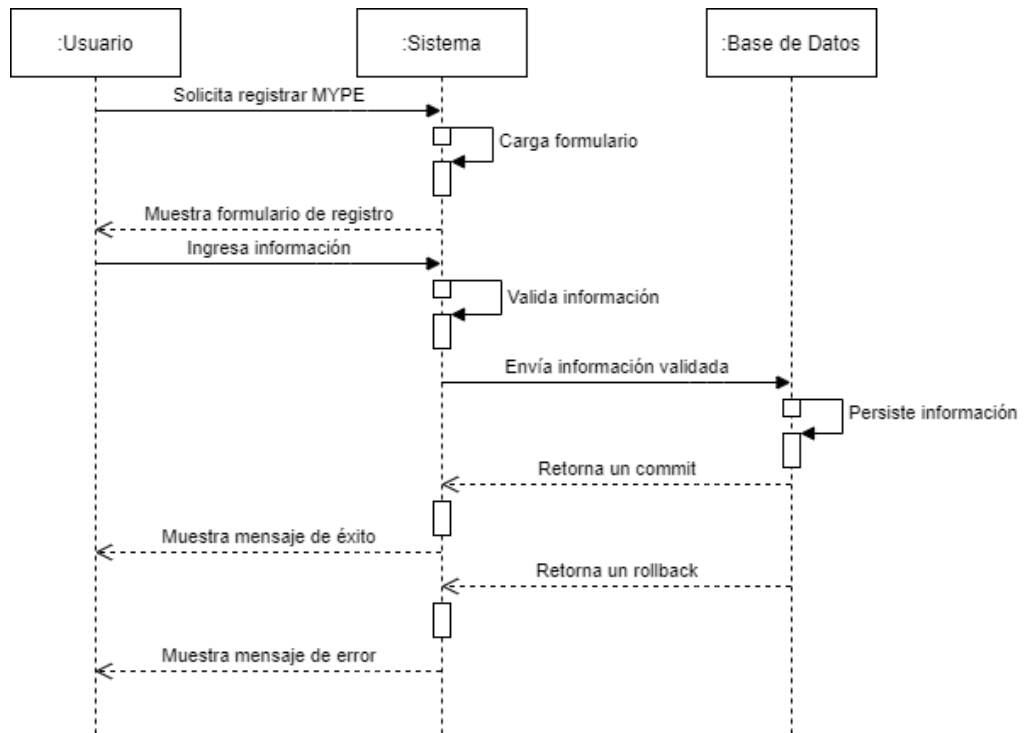


Figura 14: Diagrama de secuencia de registro de unidad productiva

Registro de personal

Para el registro del personal primero se debe de consultar si la unidad productiva a la que pertenece ya está registrada en la base de datos de clientes, en caso esta esté registrada se procede a seleccionarla, cargar la información del personal y se persiste la información en la base de datos previa validación del formato de estos datos.

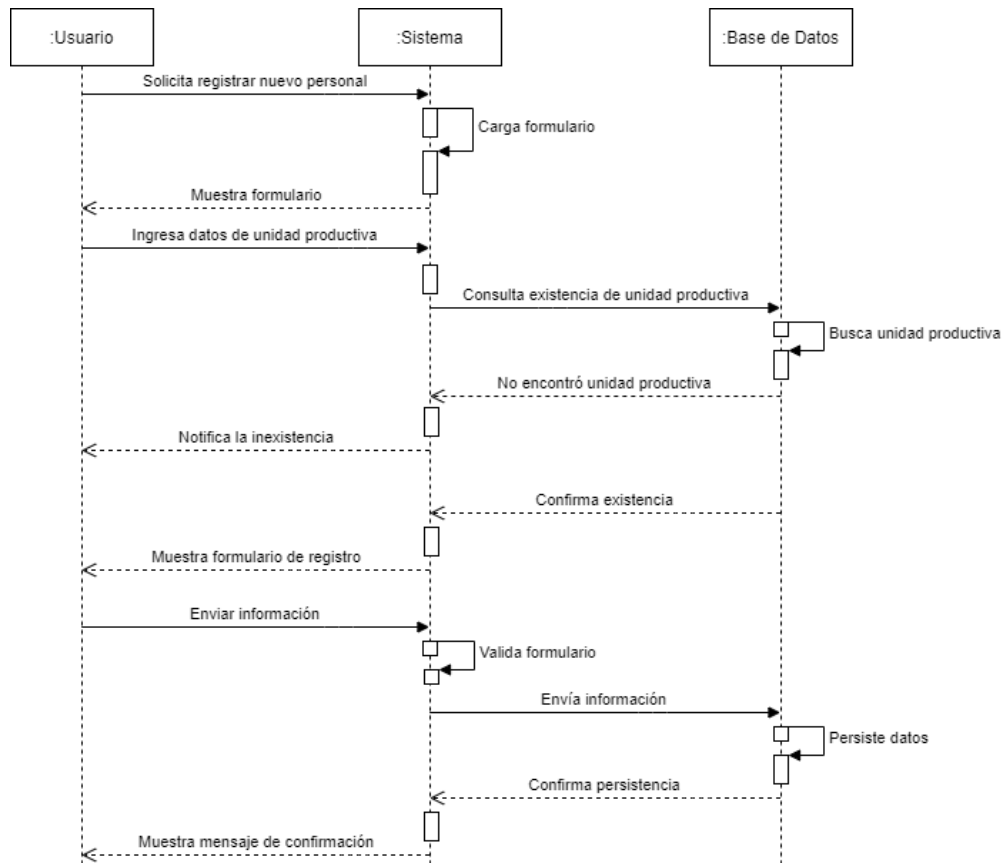


Figura 15: Diagrama de secuencia de registro de personal

Registro de evento

El usuario solicita al sistema el formulario para el registro de un nuevo evento, el sistema muestra el listado de eventos registrados y las opciones agrega nuevo y también la posibilidad de eliminar y editar los ya existentes, el usuario registra la información en el formulario y la manda al sistema para que sea validada y luego persistida en la base de datos.

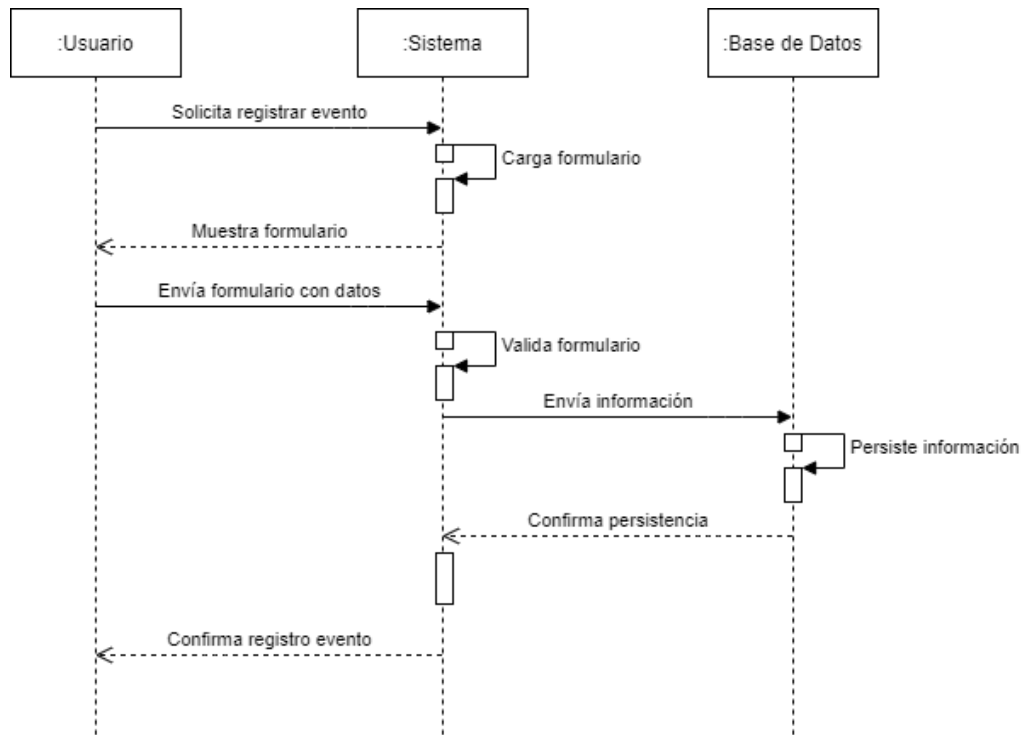


Figura 16: Diagrama de secuencia de registro de evento

Consulta cuadro de mando

El usuario accede al cuadro de mando, el cuadro de mando solicita información al sistema, el sistema proporciona la información solicitada por el cuadro de mando, el cuadro de procesa la información según los parámetros seleccionados por el usuario y muestra el resumen de la información de forma gráfica.

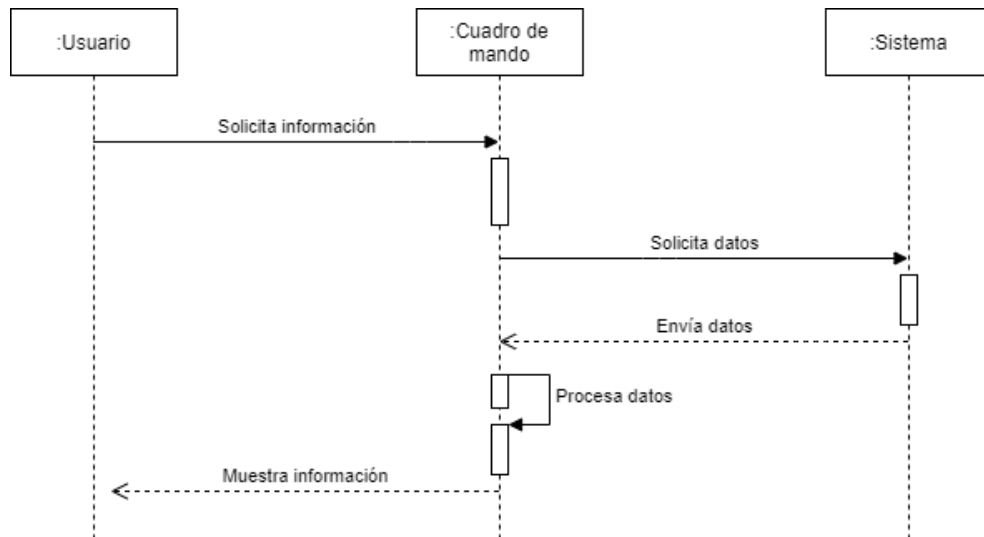


Figura 17: Diagrama de secuencia de consulta cuadro de mando

4.2.4 Diagrama de flujo

Se describe las actividades que desarrolla cada uno de los roles de usuario dentro del sistema y la interacción que tienen entre ellos, para esto se utilizó un diagrama de flujo para describir los procesos en general.

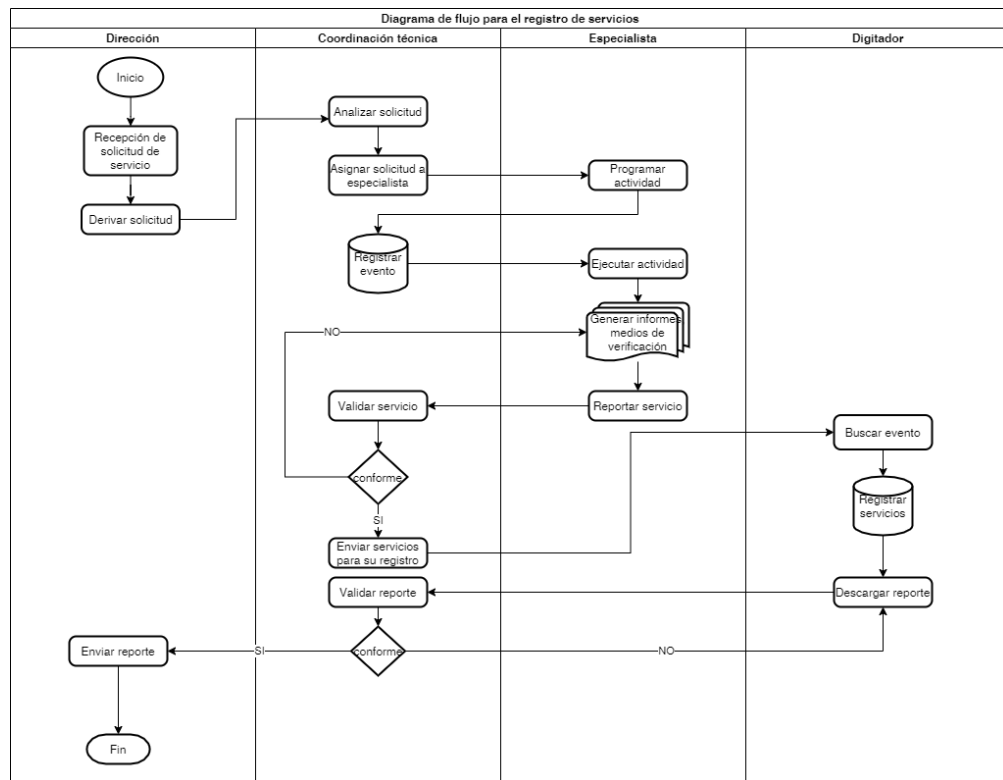


Figura 18: Diagrama de flujo del proceso de elaboración de reportes mensuales

En el diseño del sistema se tomaron en cuenta todos los requisitos a cubrir además de considerar los procesos para reducir el tiempo y la manera de comunicar el sistema desarrollado con el Framework Spring Boot con el cuadro de mando que se desarrolló empleando tecnologías R con su librería Shiny.

4.3 DESARROLLO DEL SISTEMA

Continuando con el proceso de desarrollo una vez definidos los requerimientos y los procesos necesarios para entender la lógica y las actividades de cada uno de los roles de los colaboradores se procedió a diseñar la base de datos del sistema, las interfaces que se mostrarán a los usuarios y también se desarrollaron las pruebas unitarias.



4.3.1 Diseño de base de datos

La base de datos es un componente fundamental en cualquier sistema que tenga como requisitos persistir información en el tiempo, para el diseño de la base de datos se utilizó como gestor de base de datos MariaBD.

Luego de las iteraciones según el método de desarrollo ágil XP y habiendo tomando los requerimientos de los usuarios en cada iteración se obtuvo el siguiente modelo de entidad relación de 46 tablas. Para el diseño de la base de datos se tomaron en cuenta los procesos de normalización para un mejor diseño.



De todas las tablas requeridas para el funcionamiento del sistema y para cumplir con los requerimientos de los usuarios describiremos las más importantes:

usuario: esta tabla almacena la información correspondiente a los usuarios, su estado en el sistema, y sus credenciales para acceso al sistema.

evento: esta tabla almacena la información sobre los eventos que realiza la institución, contiene información como el nombre del evento, la fecha en que se inicia y finaliza, la ubicación del evento, la tarea y servicios al que pertenece, complejidad de los servicios que se brindan en el evento, modalidad del evento, herramientas que se utilizan, financiamiento del evento y el especialista encargado de desarrollar el evento y que presentará el informe de este. En los eventos realizados se generan los servicios.

mype: esta tabla almacena información sobre las unidades productivas que son atendidas en los eventos, RUC, razón social, y toda la información que se pueda adquirir de la SUNAT sobre la unidad productiva, información de contacto, organización jurídica, régimen tributario.

personal: en esta tabla se almacena la información correspondiente al personal de cada una de las unidades productivas que son atendidas como información de identificación, genero, edad, datos para poder contactarlos, nivel educativo, si cuenta con registro nacional del artesano (RNA), si cuentan con alguna discapacidad, años de experiencia y lengua materna.



rol: esta tabla almacena la información de los roles que se asignan a los usuarios, esto con el fin de solo dar acceso a los módulos que corresponden.

tipo_servicio: esta tabla almacena información del tipo de servicio con el que se clasifican, el número que se le asigna según el Plan Operativo Institucional (POI) y el nombre se le da.

tarea_poi: esta tabla almacena información sobre las tareas que se ejecutan por cada servicio, está relacionada a la tala tipo_servicio.

servicio_cite: esta tabla relaciona las tablas evento, mype y personal esto para registrar los servicios que se dan a las unidades productivas en los eventos realizados, se agrega la tabla personal ya que es necesario poner a alguien de contacto en cada servicio.

servicio_personal: en esta tabla se almacena información de los servicios que recibe cada una de las personas que asisten a los eventos, está relacionada con la tabla mype, servicio_cite y evento esto con el fin de crear un perfil del personal sobre los servicios en los eventos en los que participó.

4.3.2 Interfaces del sistema

Para la implementación de las interfaces de usuario se utilizó el motor de plantillas Thymeleaft que es una biblioteca Java que puede interpretar plantillas XML, XHTML, HTML5 que se acopla bastante bien para trabajar con la arquitectura MVC que proporciona Spring Boot.



También se utilizaron hojas de estilo en cascada (CSS) y archivos con funciones JavaScript para añadir funcionalidades a la vista y consumir algunos servicios web necesarios para el correcto registro de servicios y consulta de información.

Para hacer que la interfaz de usuario sea amigable se tomó en cuenta lo descrito por (García Marco, 1995) que nos dice que una interfaz amigable se refiere a la facilidad de uso y a la capacidad que esta tiene para adaptarse al nivel de usuario. También menciona que para medir la amigabilidad de una interfaz podemos recurrir a cuestionarios de escalas estandarizadas.

La técnica utilizada para el proceso de desarrollo de las interfaces fue la organización y la estructuración de las interfaces en niveles jerárquicos esto con el fin de superar las limitaciones de la memoria de trabajo del espacio perceptivo de los usuarios, por lo que se utilizaron menús desplegables, pop-up, estructuras gráficas y mensajes de alerta.

Pantalla de inicio de sesión

Es la primera interfaz que se muestra a los usuarios para ingresar al sistema, en esta interfaz se registran los usuarios con el usuario y contraseña proporcionados por el administrador del sistema. La interfaz avisa al usuario en caso haya introducido sus credenciales de forma incorrecta y muestra un mensaje cuando se cierra la sesión.



Figura 20: Inicio de sesión fallido en el sistema



Figura 21: Cierre de sesión correcto en el sistema

Pantalla principal del sistema

Luego del acceso al sistema se muestra la página principal del sistema que muestra información resumida de la cantidad de unidades productivas, servicios brindados, personas atendidas y eventos realizados. Esta página se adapta al tamaño de pantalla de los dispositivos ya que se utilizó Bootstrap para su implementación.

Las opciones que se muestran en el menú de opciones varía según el rol que cumple el usuario dentro de la organización.

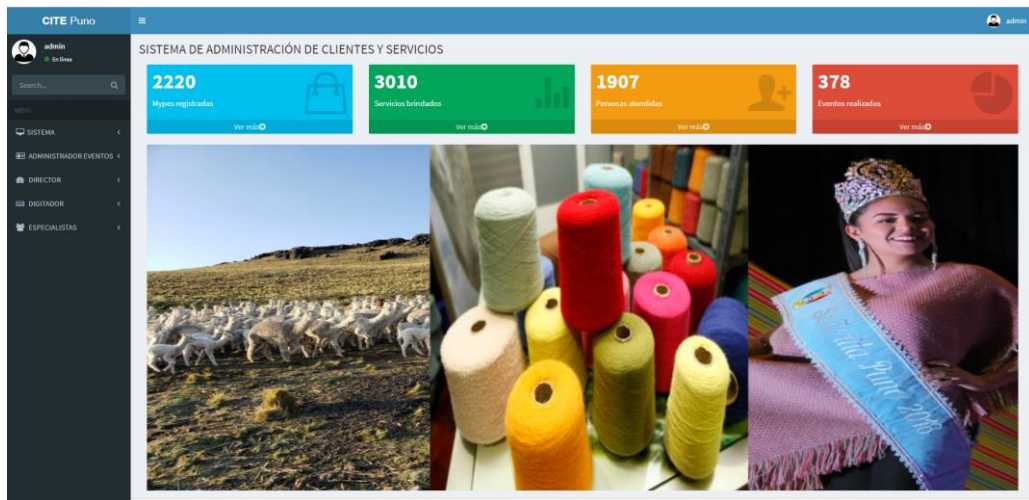


Figura 22: Página de inicio del sistema para el usuario administrador

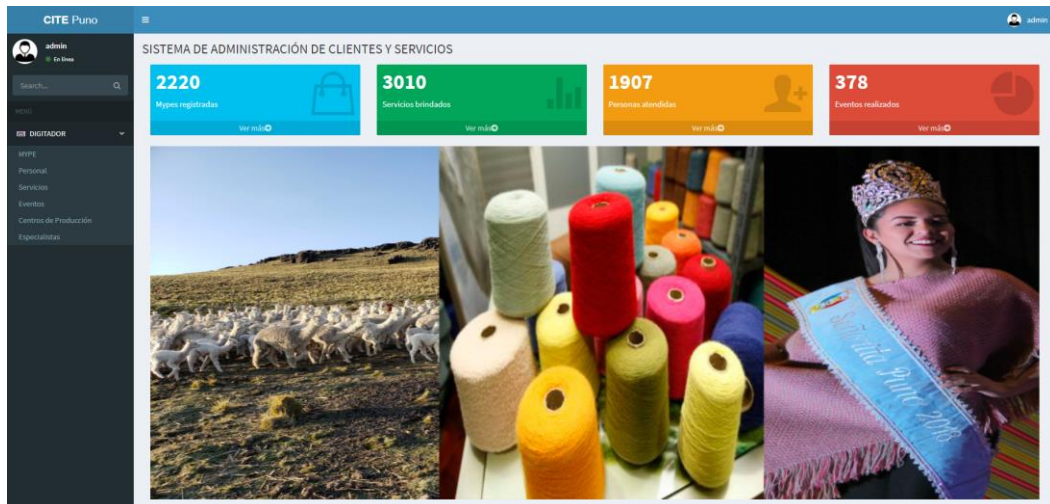


Figura 23: Página de inicio del sistema para el usuario digitador

Pantallas de registro

En todas se muestran formularios de registro para la información requerida, la lógica para el registro es similar.

CITE Puno

admin

Registrar Mype

RUC:

Razón Social:

Información Reporte

Dirección: Cursos: Teléfono:

Departamento: PUNO Provincia: --Selecione-- Distrito: --Selecione--

Tipo Empresa: --Selecione-- Org. Jurídica: --Selecione-- Regimen: --Selecione--

Art. Económico: --Selecione-- RNA: Tipo Entidad:

Web: Facebook: Observación:

Información Adicional Requerida SUNAT

Fecha de Actividad: Condición: Tipo:

Estado: Nombre Comercial: Fecha Inscripción:

Actividad Exterior: Fecha de Baja: Rubro:

Figura 24: Pantalla de registro de unidades productivas en el sistema

CITE Puno

admin

Registrar Evento Realizado

Nombre: Año del evento: Lugar:

Horas: Mes evento: Acceso: --Selecione--

Fecha Inicio: dd/mm/aaaa Fecha Reporte: dd/mm/aaaa Fecha Fin: dd/mm/aaaa

Departamento: PUNO Provincia: --Selecione-- Distrito: --Selecione--

Cursos: --Selecione-- Tarifa PDI: --Selecione-- Reporte: --Selecione--

Estado Servicio: --Selecione-- Completitud Servicio: --Selecione-- Participación CITE Proyectal: --Selecione--

Subvención Proyectal: --Selecione-- Modalidad Servicio: --Selecione-- Herramientas Utilizadas: --Selecione--

Motivo Baja Completitud(Curso): --Selecione-- Sustento (curso de baja): Nombre Facturación:

Financiamiento Servicio: --Selecione-- Informe Evento: Especialista: --Selecione--

Figura 25: Pantalla de registro de evento realizado en el sistema

Registro de servicios

Se desarrolló esta interfaz con las funcionalidades de validar información de las unidades productivas para que no se duplique el servicio brindado a esta. También agregar a los participantes que participan en estos eventos. Se utilizó JavaScript para validar la información antes de enviar el formulario.

The screenshot shows a web interface for registering clients in a service. The title is 'REGISTRAR CLIENTES EN SERVICIO'. Below the title, it specifies the service: 'REGISTRAR SERVICIO EN: CURSO VIRTUAL INTRODUCCION AL CONTROL DE CALIDAD EN EL SECTOR TEXTIL CAMELIDOS' and the report period: 'REPORTE: AGOSTO - 2021'. The form contains several input fields: 'RUC', 'RAZON SOCIAL', 'DISTRITO', 'FECHA' (pre-filled with '11/08/2021'), 'DNI CONTACTO', 'NOMBRE CONTACTO', 'ACCESO AL SERVICIO' (a dropdown menu with '2. Gratuito' selected), 'INGRESO DEL SERVICIO', 'SERIE COMPROBANTE', and 'NUMERO COMPROBANTE'. Below these, there is a section for 'Ingrese los participantes por esta MYPE' with fields for 'DNI ASISTENTE' and 'NOMBRE ASISTENTE'. At the bottom left, there are 'Aceptar' and 'Cancelar' buttons. At the bottom right, there is an 'Agregar' button. The interface has a dark sidebar on the left with navigation options like 'SISTEMA', 'ADMINISTRADOR EVENTOS', 'DIRECTOR', 'DIGITADOR', and 'ESPECIALISTAS'.

Figura 26: Pantalla de registro de servicios brindados en el sistema

Se logró desarrollar una interfaz siguiendo las pautas establecidas, los resultados del cuestionario utilizado para la medición de la amigabilidad de la interfaz se muestran en la sección de la validación del sistema.

4.4 DESARROLLO DEL CUADRO DE MANDO (DASHBOARD) CON R

Para el desarrollo del cuadro de mando (Dashboard) se siguieron los pasos mencionados por (Freddy et al., 2018) en su investigación los cuales se mencionan y describen a continuación.

4.4.1 Audiencia y Propósito

El cuadro de mando está dirigido a todos los colaboradores del Centro de Innovación Productiva y Transferencia Tecnológica Textil Camélidos Puno que apoya o intervienen en el cumplimiento de las metas físicas y objetivos establecidos en el Plan Operativo Institucional.

El propósito del cuadro de mando es mostrar información sobre el avance el avance de la ejecución de las metas físicas programadas.

4.4.2 Definición las métricas

Las métricas establecidas para ser mostradas en el cuadro de mando según la información más requerida y solicitada constantemente que se muestra en el siguiente cuadro.

Tabla 5

Definición de métricas del cuadro de mando

PANTALLA	METRICA ESTABLECIDA
Servicios	Cantidad de Servicios Brindados Cantidad por Tipo de Servicio Cantidad por Tipo de Tarea Histórico de Servicios por año
Unidades productivas	Cantidad de unidades productivas por Región Cantidad de unidades productivas por Provincia Cantidad de unidades productivas por Distrito Unidades productivas con RUC y Sin RUC
Personal	Cantidad de personas atendidas por Región Cantidad de personas atendidas por Provincia Cantidad de personas atendidas por Distrito Género de las personas atendidas
Lugares Eventos	Regiones donde se realizaron eventos Cantidad de eventos realizados Cantidad de Eventos realizados por Tipo de servicio Cantidad de Eventos realizados por Tipo de Tarea Porcentajes del tipo de acceso a los Eventos

Porcentaje de la complejidad de los Eventos
Modalidad del Evento realizado
Histórico de Eventos durante el año

4.4.3 Catalogación las fuentes de datos

Como única fuente de datos para la elaboración del cuadro de mando se utilizaron las API's que se diseñaron justamente para este fin. La información que brindan estas API's es mantenida y actualizada por el personal encargado de la digitalización de la información recabada en los eventos desarrollados.

La comunicación entre el cuadro de mando y el sistema quedó como se muestra en la siguiente imagen.

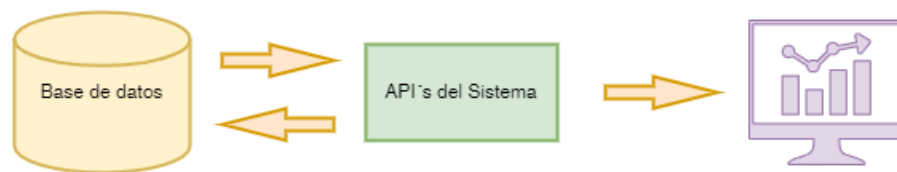


Figura 27: Interacción del cuadro de mando y sistema mediante API's

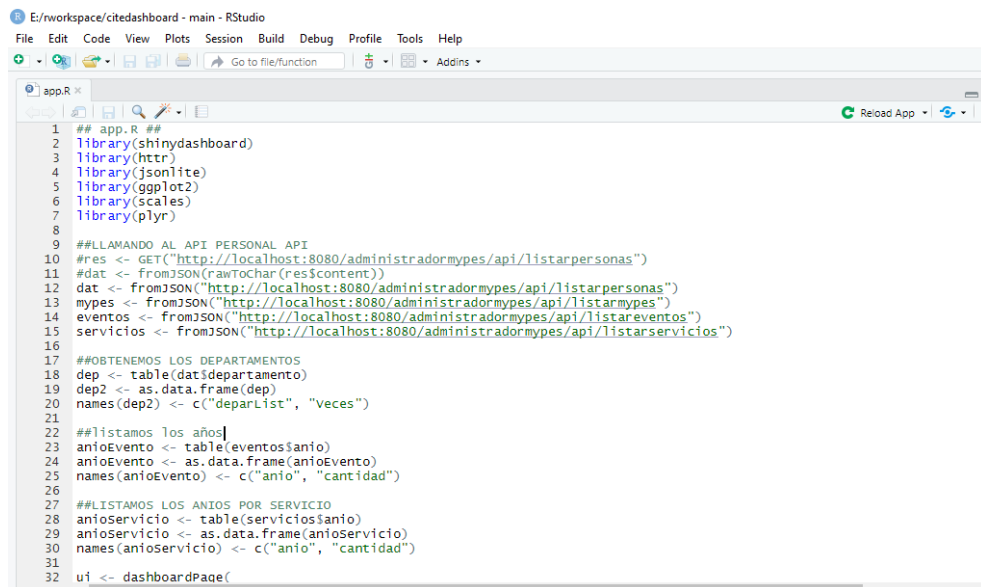
4.4.5 Definición de dimensiones y filtros

El cuadro de mando se orientó a mostrar información procesada en gráficos de todos los servicios brindados por lo que recibe toda esta información de las API's. Se determinó que el cuadro de mando pueda cambiar la información por años por lo que se le agregaron selectores para hacer el cambio.

4.4.6 Implementación del Cuadro de Mando (Dashboard)

Luego de la revisión y definición de la características y requisitos que debe de tener el cuadro de mando se empezó su desarrollo utilizando R con el paquete Shiny que nos permite y facilita la creación de aplicaciones web interactivas directamente con el lenguaje de programación R, esto con el fin de interactuar con los usuarios finales, analizar la información y comunicar los resultados obtenidos. Para el manejo de la solicitudes y procesamiento la información en formato JSON se agregaron las librerías “jsonlite” y “httr”.

Para acelerar el desarrollo se utilizó el paquete “shinydashboard” que provee una estructura y widgets para reducir el tiempo en el desarrollo de un cuadro de mando utilizando R. Se crearon varias páginas para mostrar la información por separado.



```
E:/workspace/citedashboard - main - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to File/function Addins
app.R
1 ## app.R ##
2 library(shinydashboard)
3 library(httr)
4 library(jsonlite)
5 library(ggplot2)
6 library(scales)
7 library(plyr)
8
9 ##LLAMANDO AL API PERSONAL API
10 #res <- GET("http://localhost:8080/administradormypes/api/listarpersonas")
11 #dat <- fromJSON(rawToChar(res$content))
12 dat <- fromJSON("http://localhost:8080/administradormypes/api/listarpersonas")
13 mypes <- fromJSON("http://localhost:8080/administradormypes/api/listarmypes")
14 eventos <- fromJSON("http://localhost:8080/administradormypes/api/listareventos")
15 servicios <- fromJSON("http://localhost:8080/administradormypes/api/listarservicios")
16
17 ##OBTENEMOS LOS DEPARTAMENTOS
18 dep <- table(dat$departamento)
19 dep2 <- as.data.frame(dep)
20 names(dep2) <- c("deparList", "Veces")
21
22 ##listamos los años|
23 anioEvento <- table(eventos$anio)
24 anioEvento <- as.data.frame(anioEvento)
25 names(anioEvento) <- c("anio", "cantidad")
26
27 ##LISTAMOS LOS ANIOS POR SERVICIO
28 anioServicio <- table(servicios$anio)
29 anioServicio <- as.data.frame(anioServicio)
30 names(anioServicio) <- c("anio", "cantidad")
31
32 ui <- dashboardPage(
33
```

Figura 28: Código de implementación del cuadro de mando

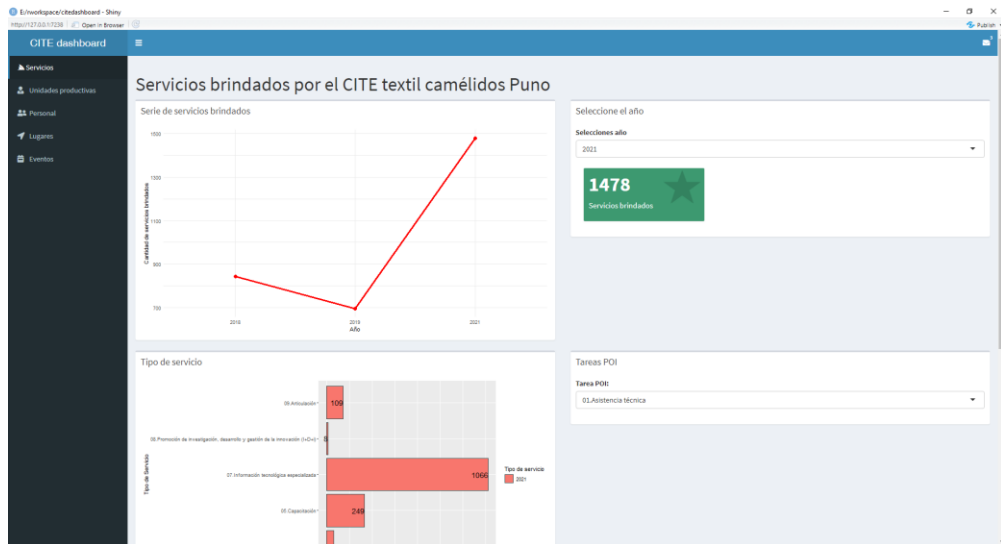


Figura 29: Interfaz de servicios brindados en el cuadro de mando

Página Unidades productivas

En esta página se muestra información resumida sobre todas las unidades productivas que se llegaron a atender por región, provincias por región y distritos por provincia, esto puede ser seleccionado, además de mostrar el porcentaje de unidades productivas con RUC y sin RUC que fueron atendidas.

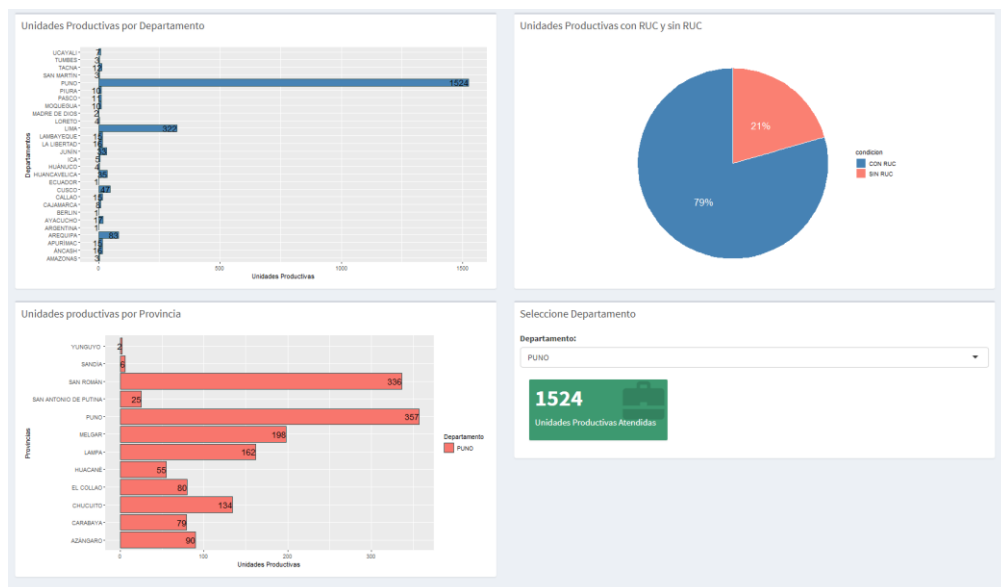


Figura 30: Interfaz de unidades productivas del cuadro de mando

Página personal

Esta página muestra información resumida sobre el personal de las unidades productivas que se atendieron, la información que destaca es la cantidad de personas atendidas por región, provincias por región, distritos por provincia que se puede seleccionar y el porcentaje del género de personal atendido.

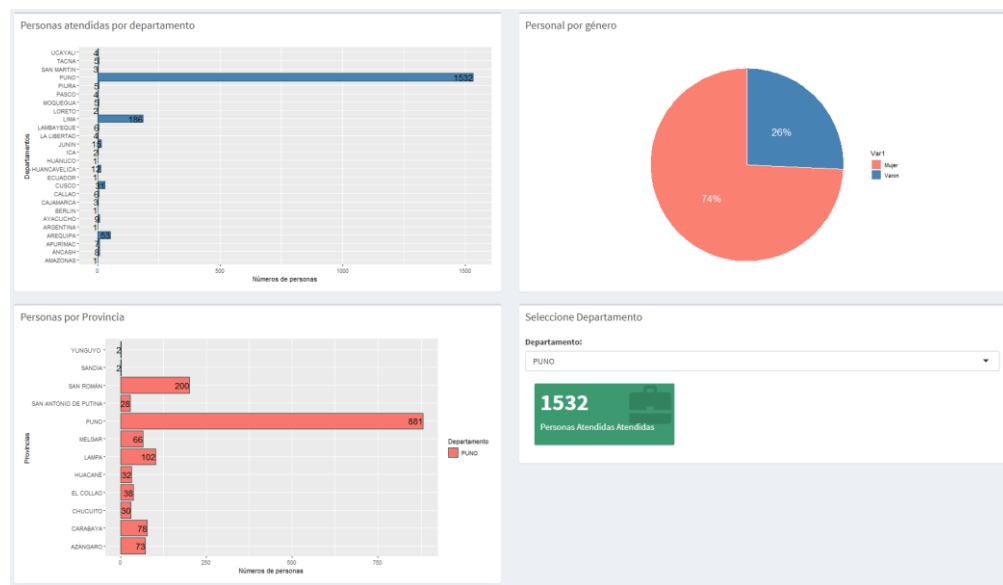


Figura 31: Interfaz de personal en el cuadro de mando

Página de Eventos

Esta página muestra el resumen de los eventos realizados por año, destacando información como el histórico de eventos por año, tipo de servicio brindado en el evento, la tarea del Plan Operativo Institucional al que pertenece, el tipo de acceso al servicio, complejidad del servicio y las herramientas utilizadas para la ejecución de estos, la modalidad en la que se desarrolló el evento y el histórico por mes.

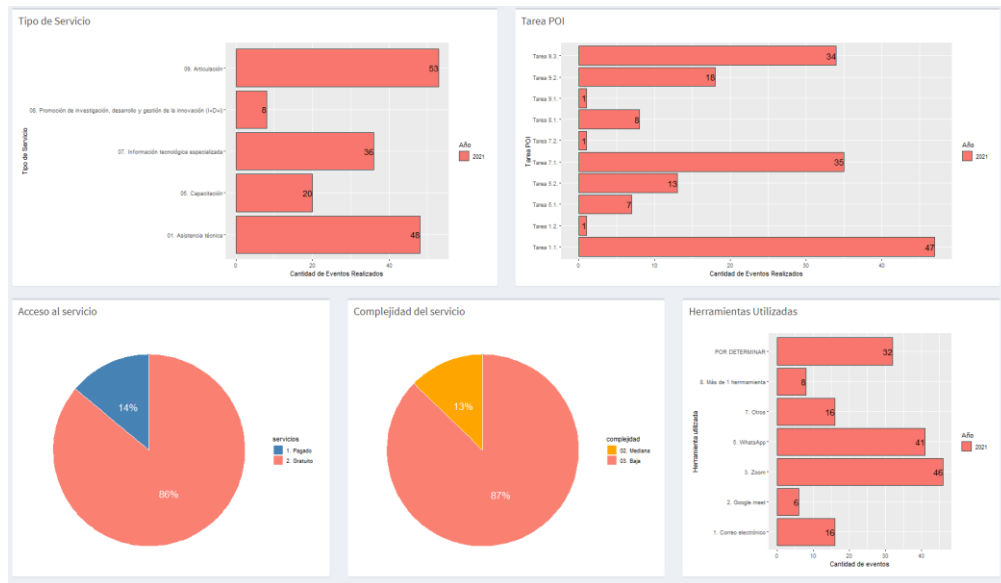


Figura 32: Interfaz de eventos realizados en el cuadro de mando

El cuadro de mando consume información del sistema que es proporcionada mediante un API. Esta información se procesa con el lenguaje de programación R y el sistema se limita proporcionar la información sin procesar en formato JSON.

4.5 PRUEBAS UNITARIAS

Se realizaron pruebas a los métodos implementados en los servicios que contienen la lógica que se encarga de validar la información e interactúa con los repositorios de todas las entidades de las bases de datos. Para este fin se empleó Junit 5 que proporciona anotaciones y métodos para poder testear cada una de las funciones de las clases.

Adicionalmente se empleó MOCK para imitar el comportamiento de los servicios esto con el fin de no alterar la base de datos en el testeo ya que los MOCK nos permiten simular su comportamiento y así mantener la integridad de los datos.

Para poder utilizar JUnit 5 en nuestro proyecto fue necesario agregar las dependencias en el archivo pom.xml de nuestro proyecto, además de excluir las dependencias de JUnit 4 que están instalado por defecto en nuestro proyecto.

```
118  
119     <!-- junit 5 -->  
120     <dependency>  
121         <groupId>org.junit.jupiter</groupId>  
122         <artifactId>junit-jupiter-api</artifactId>  
123         <scope>test</scope>  
124     </dependency>  
125  
126     <dependency>  
127         <groupId>org.junit.jupiter</groupId>  
128         <artifactId>junit-jupiter-engine</artifactId>  
129         <scope>test</scope>  
130     </dependency>  
131
```

Figura 33: Configuración de JUnit 5 en el archivo POM.xml del proyecto

Luego creamos las clases de prueba que contendrán los métodos y acciones que probarán el comportamiento y verificarán los resultados de nuestros métodos, estas clases se crearon en el directorio de test que se crea por defecto en nuestro proyecto.

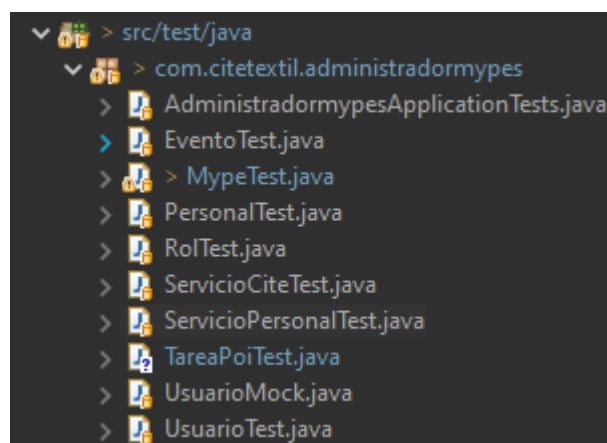


Figura 34: Clases creadas para el testeo de los servicios

Luego de configurar las dependencias necesarias y crear las clases se crearon las funciones que prueban el comportamiento de los servicios desarrollados.

```
TareaPoiTest.java x
19 @SpringBootTest
20 public class TareaPoiTest {
21
22     @MockBean
23     private TareaPoiService tps;
24
25     @MockBean
26     private TareaPoiRepository tpr;
27
28     @Test
29     @DisplayName("Test findById")
30     void testFindById() {
31         TareaPoiModel tpoim = new TareaPoiModel(10, "Tarea 8.1", "Proyectos de I+D+I", "2019", 1);
32
33         doReturn(tpoim).when(tps).buscarPorModelId(10);
34
35         Optional<TareaPoiModel> returnTpoi = Optional.of(tps.buscarPorModelId(10));
36
37         Assertions.assertTrue(returnTpoi.isPresent(), "La Tarea no fue encontrada");
38         Assertions.assertSame(returnTpoi.get(), tpoim, "La tarea devuelta es distinta a la del test");
39     }
40
41
42     @Test
43     @DisplayName("Test Save TareaPoi")
44     void testSave() {
45         TareaPoiModel tpoim = new TareaPoiModel(10, "Tarea 8.1", "Proyectos de I+D+I", "2019", 1);
46         doReturn(tpoim).when(tps).agregar(tpoim);
47
48         TareaPoiModel returnTpoi = tps.agregar(tpoim);
49
50         Assertions.assertNotNull(returnTpoi, "El objeto guardado debe ser nulo");
51         Assertions.assertEquals(10, returnTpoi.getId(), "el id no es el mismo");
52     }
53
54
55     @Test
56     @DisplayName("Test findAll")
57     void testFindAll() {
58         TareaPoiModel tpoim1 = new TareaPoiModel(1, "Tarea 8.1", "Proyectos de I+D+I", "2019", 1);
59         TareaPoiModel tpoim2 = new TareaPoiModel(2, "Tarea 8.2", "Articulos Cientificos", "2019", 1);
60         doReturn(Arrays.asList(tpoim1, tpoim2)).when(tps).listar();
61
62         List<TareaPoiModel> tpoil = tps.listar();
```

Figura 35: Prueba unitaria para los servicios de tareas

JUnit devolverá mensajes según sea el resultado que arrojen los métodos testeados, en nuestro desarrollo se fueron corrigiendo los métodos para que los resultados sean los que esperamos y así tener controlado su comportamiento dentro del sistema.

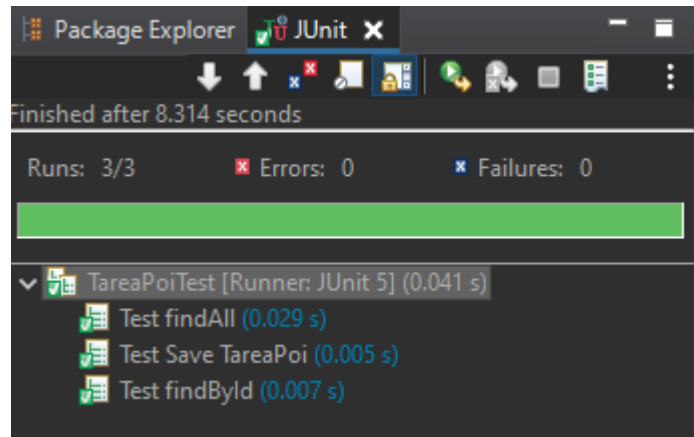


Figura 36: Resultado correcto de la prueba unitaria

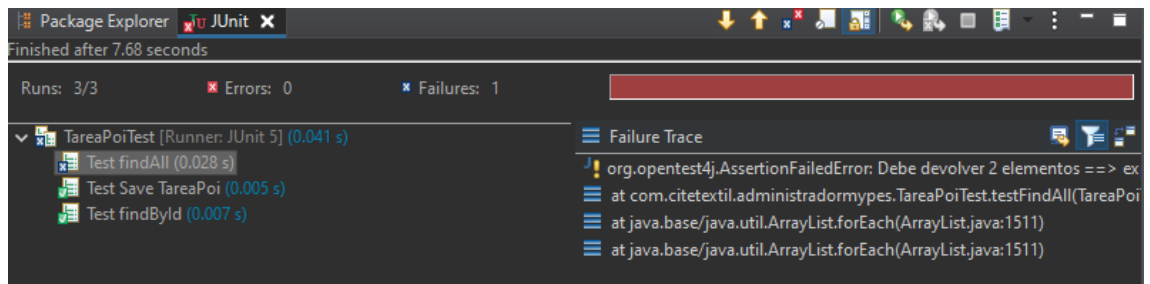


Figura 37: Resultados incorrectos de la prueba unitaria

En la etapa de desarrollo del sistema se logró plasmar todos los requerimientos solicitados en funciones y métodos que permitan reducir el tiempo empleado en las actividades de digitalización de información, considerando brindar interfaces amigables a los usuarios para que su interacción con el sistema sea más fácil además de implementar API's para la comunicación del sistema y el cuadro de mando. En la siguiente sección se muestra el proceso de validación del sistema incluido el cuadro de mando porque este pasa a ser parte del sistema en general.

4.6 VALIDACIÓN DEL SISTEMA

4.6.1 ISO 25010

La evaluación de la calidad del sistema web con cuadro de mando que combina tecnologías Java y R se empleó el modelo de calidad definido por la ISO/IEC 25010 que propone 8 categorías de calidad que a su vez se dividen en subcategorías. Para establecer las categorías de calidad que debe de cumplir nuestro sistema se hizo una evaluación de los requerimientos del sistema y de esta manera seleccionar los módulos de evaluación, métricas y técnicas de medición para definir los criterios de decisión vamos a aplicar en nuestra evaluación del sistema.



Figura 38: Categorías de calidad de producto del ISO/IEC 25010

Fuente: iso25000.com

Luego de realizar un análisis exhaustivo de los requerimientos de las historias de los usuarios se definió que el sistema debe de cumplir tres categorías para cumplir la calidad exigida por los usuarios del sistema. Además dentro de estas categorías el sistema debe de cumplir con 10 subcategorías. Las categorías y subcategorías que debemos de cumplir se muestran en la siguiente imagen.



Figura 39: Categorías consideradas para la calidad del sistema

4.6.2 Eficiencia de desempeño

Las subcategorías que el sistema web con cuadro de mando que combina tecnologías Java y R en esta categoría fueron comportamiento temporal. Utilización de recursos y capacidad y su descripción de muestra en la siguiente tabla.

Tabla 6

Descripción de las subcategorías de eficiencia de desempeño

Eficiencia de desempeño	Comportamiento temporal	Tiempo de respuesta y procesamiento empleado por el software, además de la capacidad de rendimiento del mismo bajo condiciones determinadas
	Capacidad	Capacidad del sistema para funcionar bajo ciertas circunstancias extremas en relación a los requisitos

Las pruebas de eficiencia de desempeño se realizaron utilizando pruebas de carga para el comportamiento temporal y pruebas de estrés para la capacidad, esto se realizó utilizando JMeter como herramienta.

4.6.3 Comportamiento temporal

En la medición de esta subcategoría se estableció que la respuesta del software debe ser menor a 5 segundos por cada función que se realiza y esta cantidad de tiempo se multiplica por la cantidad total de usuarios simultáneos en total que pueda tener el sistema.

Para la primera medición se crearon scripts en JMeter para que estos sean reproducidos con un solo hilo simulando las solicitudes de un solo usuario, la configuración y el árbol de resultados se muestran en las imágenes siguientes.

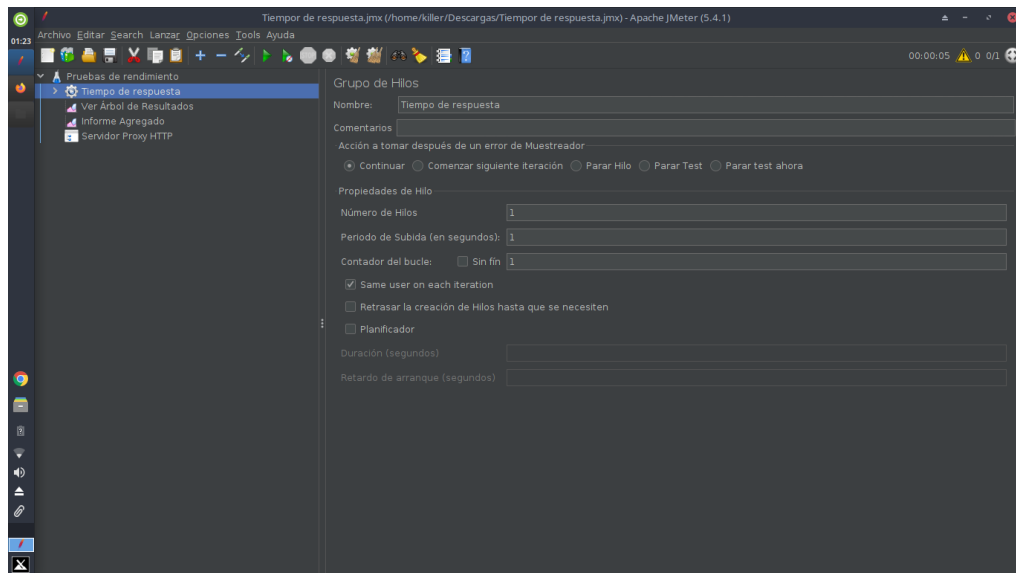


Figura 40: Configuración de JMeter para medir el comportamiento temporal del sistema

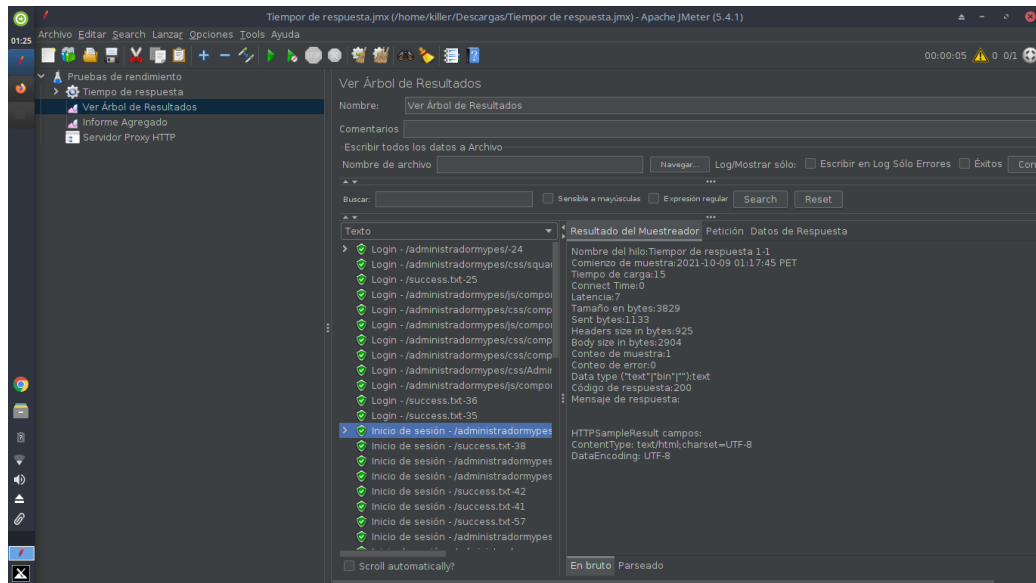


Figura 41: Árbol de resultados de la prueba de comportamiento temporal

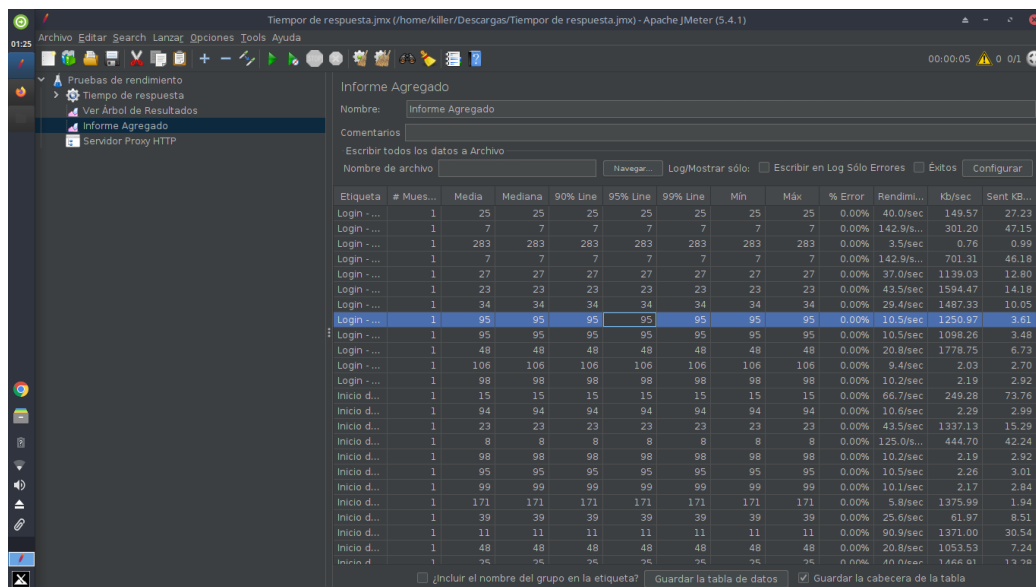


Figura 42: Informe agregado de la medición del comportamiento temporal



Los resultados proporcionados por JMeter en su informe agregado está dados en milisegundos, el resultado para la medición del proceso de generación de reportes para un solo usuario se muestra en la siguiente tabla.

Tabla 7

Resumen de los resultados de la medición del comportamiento temporal

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mínimo	Máximo	% Error
Total	55	96	48	152	302	350	7	1349	0.00%

De la tabla anterior podemos ver que el tiempo de respuesta promedio por proceso que implica la generación del reporte es de 0.096 segundos, teniendo tiempos de respuesta mínimo de 0.007 segundos y picos de 1.349 segundos en algunos procesos por lo que podemos decir que el sistema cumple con la métrica establecida.

Para medir el rendimiento en caso tenga a todos los colaboradores de la institución como usuarios concurrentes por lo que en el peor de los casos el sistema según lo establecido en las métricas descritas párrafos más arriba debería de responder en menos de 40 segundos por proceso para ser considerado de calidad, en esta medición se configuró JMeter para que ejecute los scripts desde 8 hilos que simularan los 8 usuarios en simultaneo. La configuración de JMeter se muestra en las siguientes imágenes.

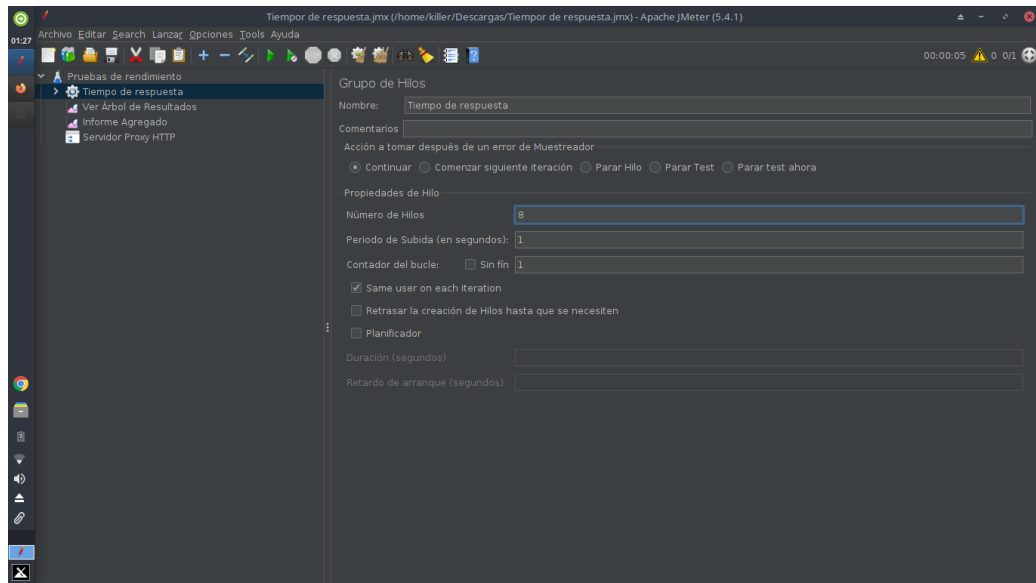


Figura 43: Configuración de JMeter para las pruebas de comportamiento temporal con ocho usuarios

Etiqueta	# Mue...	Media	Mediana	90% Line	95% Line	99% Line	Min	Máx	% Error	Rendim...	Kb/sec	Sent KB...
Inicio d...	8	826	614	1412	1457	1457	263	1457	0.00%	52.1/min	1.39	0.29
Inicio d...	8	728	579	939	1647	1647	388	1647	0.00%	51.1/min	4.01	0.29
Inicio d...	8	1965	625	2522	9549	9549	516	9549	0.00%	25.5/min	97.60	0.20
Inicio d...	8	6999	6469	8169	11043	11043	4652	11043	0.00%	19.7/min	581.98	0.11
Listar r...	8	1027	1143	1448	1723	1723	354	1723	0.00%	26.7/min	1.66	0.37
Listar r...	8	288	159	515	994	994	12	994	0.00%	28.6/min	1.70	0.17
Listar r...	8	430	130	725	2055	2055	14	2055	0.00%	29.5/min	7.41	0.17
Listar r...	8	377	141	992	1320	1320	7	1320	0.00%	30.9/min	1.24	0.18
Listar r...	8	485	155	1143	1942	1942	19	1942	0.00%	32.9/min	16.87	0.20
Listar r...	8	334	176	319	1182	1182	37	1182	0.00%	35.6/min	30.02	0.21
Listar r...	8	376	152	460	1317	1317	56	1317	0.00%	38.9/min	9.45	0.21
Listar r...	8	465	376	536	1119	1119	23	1119	0.00%	43.4/min	26.54	0.25
Listar r...	8	521	513	556	1007	1007	91	1007	0.00%	47.8/min	83.12	0.28
Listar r...	8	395	480	513	539	539	44	539	0.00%	50.1/min	42.47	0.28
Listar r...	8	480	521	568	972	972	86	972	0.00%	50.5/min	99.95	0.30
Listar r...	8	382	450	504	547	547	74	547	0.00%	51.0/min	72.51	0.29
Listar r...	8	383	422	518	520	520	65	520	0.00%	51.2/min	64.34	0.29
Listar r...	8	438	443	706	792	792	15	792	0.00%	51.9/min	4.07	0.31
Listar r...	8	1077	674	1806	2601	2601	189	2601	0.00%	51.7/min	202.67	0.29
Listar r...	8	1595	1367	1828	3842	3842	400	3842	0.00%	51.2/min	384.49	0.29
Descar...	8	1109	781	2371	2829	2829	49	2829	0.00%	59.6/min	3.71	0.85
Descar...	8	546	277	535	2028	2028	206	2028	0.00%	1.1/sec	0.25	0.40
Total	440	617	378	1236	1647	6977	7	11043	0.00%	12.2/sec	1104.30	4.52

Figura 44: Informe agregado del resultado de la medición de rendimiento con ocho usuarios

Los resultados obtenidos en esta medición se muestran en la siguiente tabla de resumen.



Tabla 8

Resumen del informe agregado de la medición de rendimiento con ocho usuarios

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mínimo	Máximo	% Error
Total	440	617	378	1236	1647	6977	7	11043	0.00%

En el cuadro anterior podemos observar que el tiempo promedio de respuesta de los procesos es de 0.617 segundos teniendo bajos mínimos de 0.007 segundos y picos altos de 11.043 segundos, por lo que estos valores están dentro de los establecidos en la métrica para esta medición por lo que cumple con la métrica establecida.

4.6.4 Capacidad

Revisando las especificaciones el caso extremo establecido fue una cantidad mayor de usuarios y se estableció que el sistema debe de ser capaz de responder en mismo tiempo en que responde a ocho usuarios concurrentes (40 segundos) en el peor de los casos, para efectuar esta prueba se aumentó la cantidad de hilos en la configuración de JMeter a 50 para que cada uno simule un usuario en el proceso de generación de reportes de servicios mensuales. La configuración realizada se muestra en la siguiente imagen.

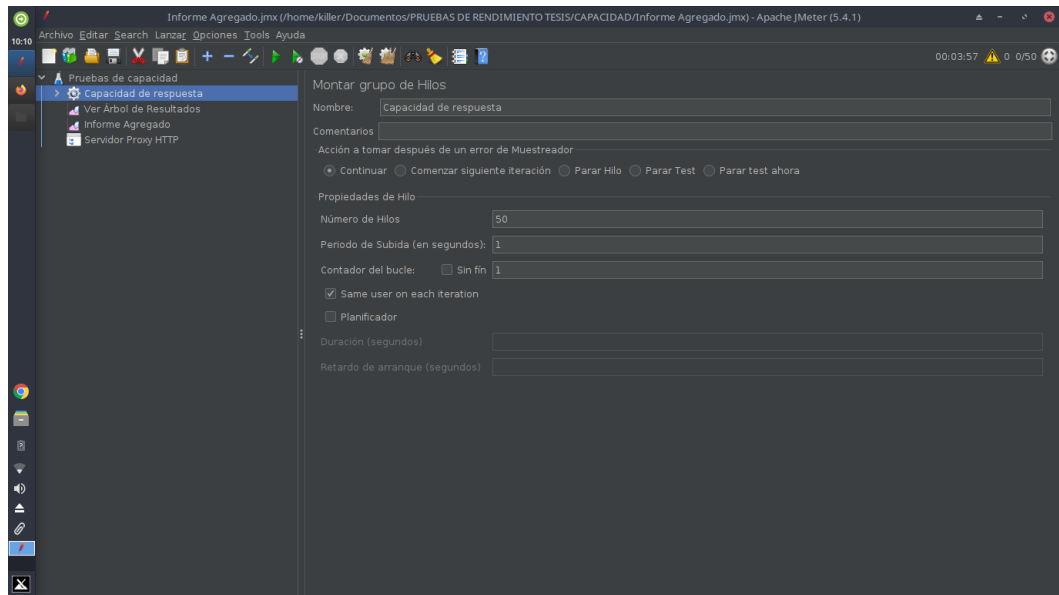


Figura 45: Configuración de JMeter con cincuenta usuarios

El resumen de esta prueba se muestra en la siguiente tabla.

Tabla 9

Resumen de la medición de capacidad con cincuenta usuarios

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mínimo	Máximo	% Error
Total	4400	1847	532	3632	5935	24247	5	158663	1.18%

De los resultados anteriores podemos observar que el tiempo promedio de respuesta de los procesos es de 1.847 segundos, que el tiempo mínimo es de 0.005 segundos y el tiempo máximo es de 158.663, por lo que podemos decir que en promedio el sistema aún cumple con la métrica establecida y que se debe de revisar los procesos que presentan los tiempos más altos.



Debemos considerar que las pruebas se realizaron utilizando una conexión inalámbrica que puede que haya aumentado el tiempo de respuesta del sistema en sus procesos.

4.6.5 Adecuación funcional y Usabilidad

Para la validación de la calidad externa del sistema concerniente a la adecuación funcional y usabilidad se elaboró un cuestionario en donde se plantean una serie de preguntas que se consideran dentro del modelo de calidad de la norma ISO/IEC 25000. Este cuestionario se aplicó a todos los colaboradores implicados en el uso de la información que es manejada por el sistema y de esta manera validar el sistema web con cuadro de mando, combinando tecnologías Java y R para la administración de servicios.

Los resultados y análisis de la aplicación del cuestionario luego de que los usuarios prueben el sistema se muestran a continuación.

Pregunta 1: ¿El sistema le permite realizar todas sus actividades en cuanto a la información manejada en la institución?

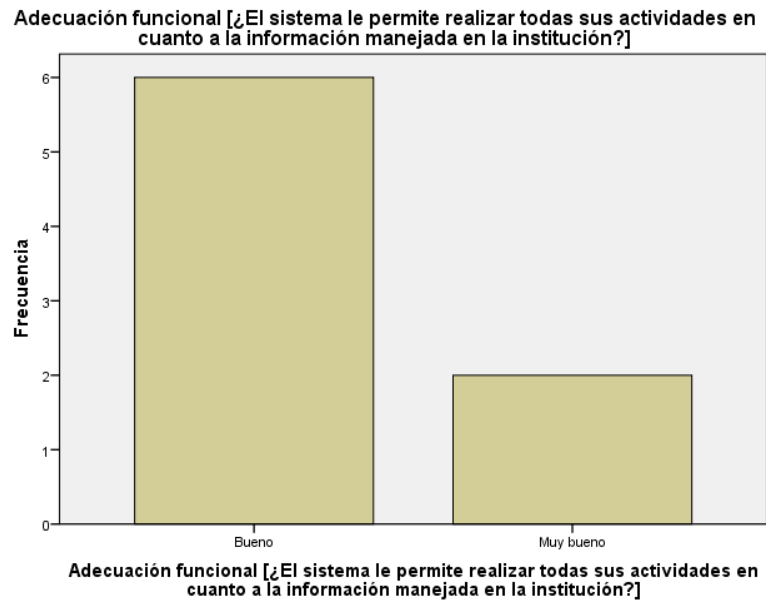


Figura 46: Gráfico de barras de la pregunta N° 1

Tabla 10

Tabla de distribución de frecuencia de la pregunta N° 1

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Bueno	6	75.0	75.0	75.0
Muy bueno	2	25.0	25.0	100.0
Total	8	100.0	100.0	

El 75% de los usuarios opinó que es bueno y el otro 25% de usuarios opinó que es muy bueno, con estas repuesta podemos concluir que el sistema les permite realizar todas sus actividades y el sistema cubre todas las necesidades del usuario.

Pregunta 2: ¿Los resultados que le brinda el resultado son los esperados para sus actividades dentro de la institución?

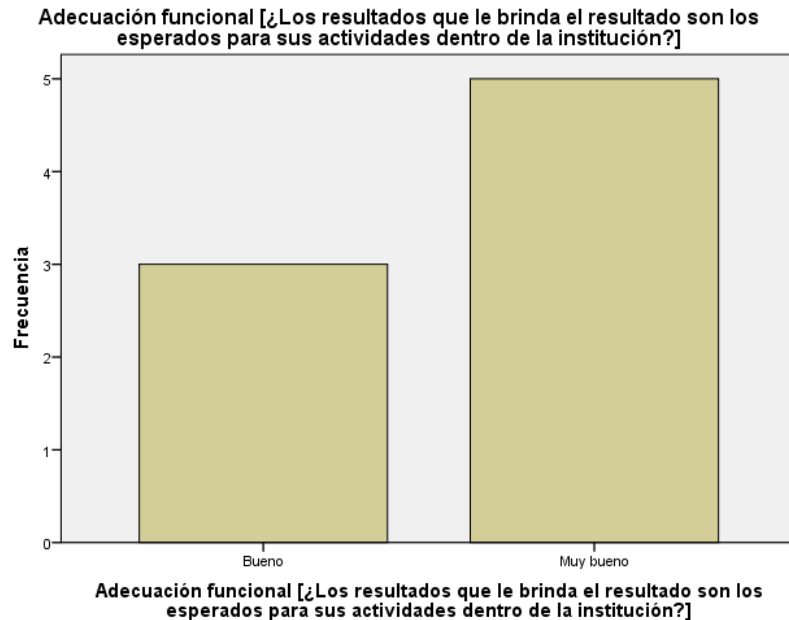


Figura 47: Gráfico de barras de la pregunta N° 2

Tabla 11

Tabla de frecuencias de la pregunta N° 2

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Bueno	3	37.5	37.5	37.5
Muy bueno	5	62.5	62.5	100.0
Total	8	100.0	100.0	

El 37.5% de usuarios respondieron que el sistema es bueno y complementariamente el 62.5% dijo que el sistema es muy bueno en este aspecto por lo que podemos concluir que el sistema es capaz de brindar resultados correctos y coherentes con los que se espera.

Pregunta 3: ¿El sistema le brinda funciones que le permiten ejecutar sus tareas dentro de la institución?

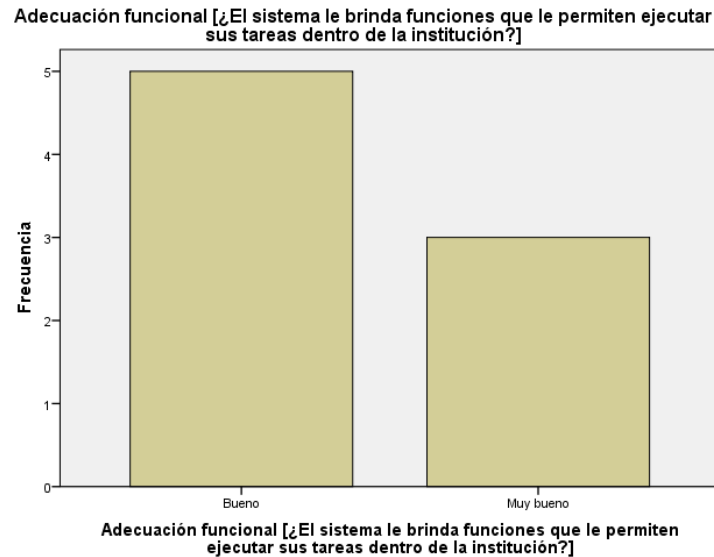


Figura 48: Gráfico de barras de la pregunta N° 3

Tabla 12

Tabla de frecuencia de la pregunta N° 3

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Bueno	5	62.5	62.5	62.5
Muy bueno	3	37.5	37.5	100.0
Total	8	100.0	100.0	

El 62.5% de los usuarios encuestados respondieron que el sistema es bueno y el otro 37.5% indicó que es muy bueno, esto nos lleva a concluir que el sistema es pertinentemente funcional y proporciona las funciones necesarias para que los usuarios realicen sus tareas y cumplan sus objetivos.

Pregunta 4: ¿Las funciones y opciones que le muestra el sistema le ayuda a entender el funcionamiento del sistema?

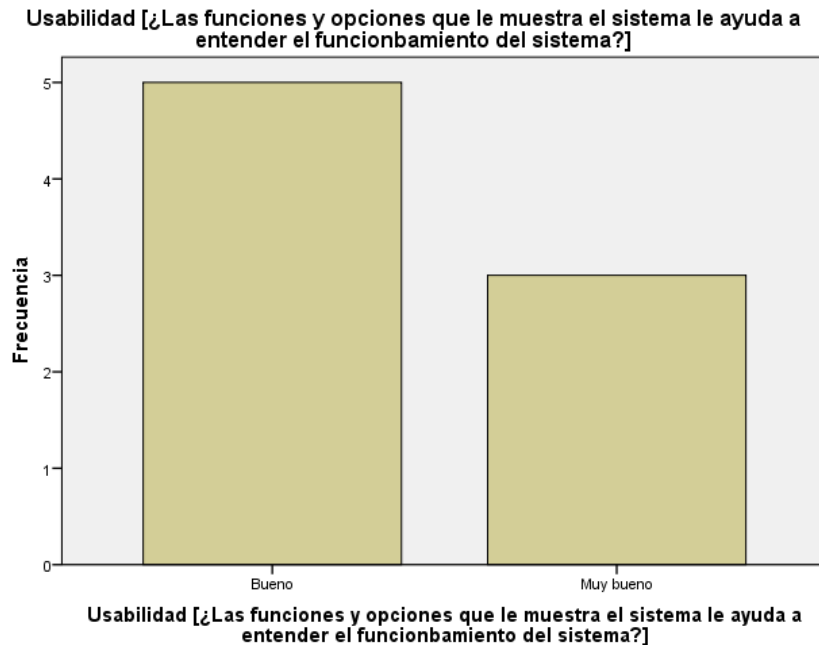


Figura 49: Gráfico de barras de la pregunta N° 4

Tabla 13

Tabla de frecuencias de la pregunta N° 4

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Bueno	5	62.5	62.5	62.5
Muy bueno	3	37.5	37.5	100.0
Total	8	100.0	100.0	

El 62.5% de los usuarios respondió que el sistema es bueno y el otro 37.5% restante indicó que el sistema es muy bueno es este aspecto, por esto podemos concluir que el sistema es capaz de ser entendido por un usuario y cubre sus necesidades.

Pregunta 5: ¿El tiempo empleado en aprender a usar el sistema?

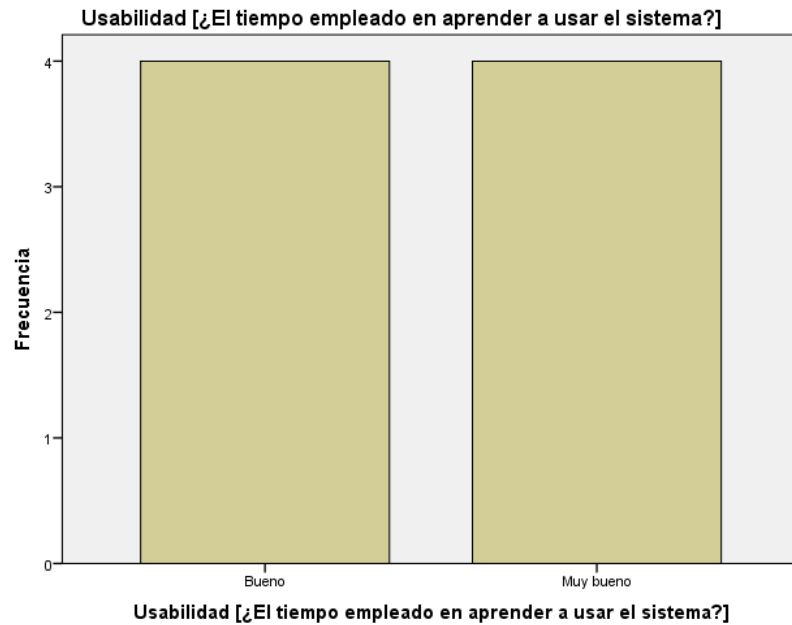


Figura 50: Gráfico de barras de la pregunta N° 5

Tabla 14

Tabla de frecuencias de la pregunta N° 5

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Bueno	4	50.0	50.0	50.0
Muy bueno	4	50.0	50.0	100.0
Total	8	100.0	100.0	

El 50% de los usuarios respondieron que el sistema es bueno, el otro 50% de los usuarios en cambio respondió que el sistema es muy bueno en esta característica por lo que podemos concluir que el sistema tiene la capacidad de ser aprendido en un corto tiempo por los usuarios.

Pregunta 6: ¿El sistema es intuitivo y fácil de manejar?

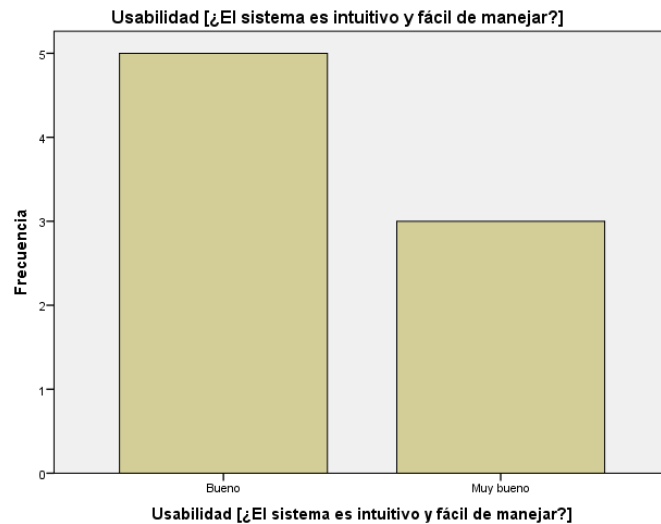


Figura 51: Gráfico de barras de la pregunta N° 6

Tabla 15

Tabla de frecuencias de la pregunta N° 6

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Bueno	5	62.5	62.5	62.5
Muy bueno	3	37.5	37.5	100.0
Total	8	100.0	100.0	

El 62.5% de los usuarios respondió que el sistema es bueno y el otro 37.5% respondió que el sistema es muy bueno en esta característica por lo que podemos concluir que el sistema tiene la capacidad de ser usado, manejado y controlado por el usuario.

Pregunta 7: ¿El sistema tiene mecanismos que le alertan sobre posibles errores en su uso?

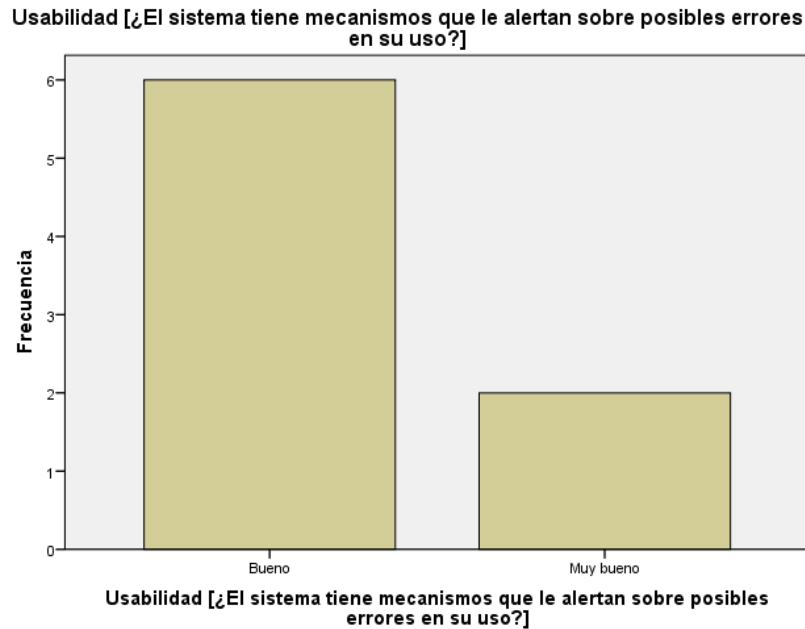


Figura 52: Gráfico de barras de la pregunta N° 7

Tabla 16

Tabla de frecuencias de la pregunta N° 7

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Buena	6	75.0	75.0	75.0
Muy buena	2	25.0	25.0	100.0
Total	8	100.0	100.0	

El 75% de usuarios respondió que el sistema es bueno y el 25% restante respondió que el sistema es muy bueno con respecto a esta característica, por lo que podemos concluir que el sistema es capaz de gestionar el comportamiento no esperado de los usuarios.

Pregunta 8: ¿La interfaz mostrada en el sistema le parece amigable y facilita su uso?

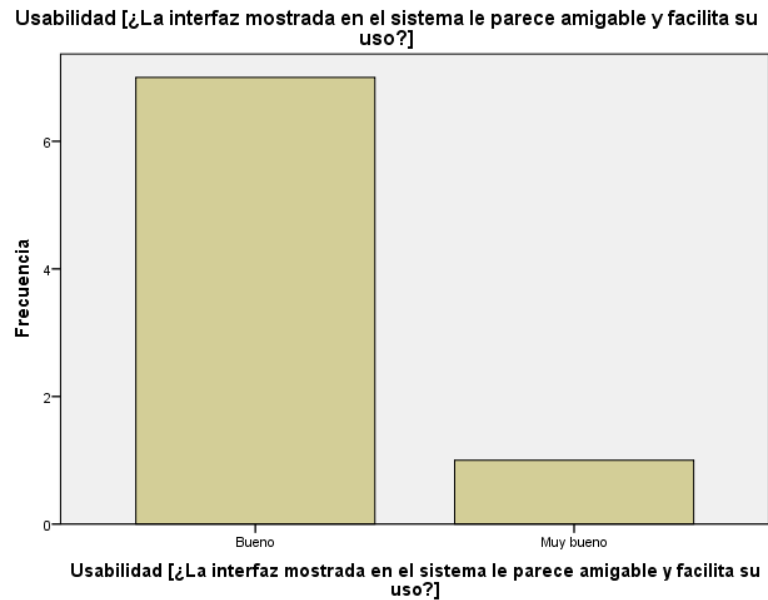


Figura 53: Gráfico de barras de la pregunta N° 8

Tabla 17

Tabla de frecuencias de la pregunta N° 8

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Bueno	7	87.5	87.5	87.5
Muy bueno	1	12.5	12.5	100.0
Total	8	100.0	100.0	

El 87.5% de los usuarios respondió que el sistema es bueno y el 12.5% restante respondió que es muy bueno con respecto a esta característica, por lo que podemos concluir que el sistema presenta una interfaz estéticamente buena y amigable para el usuario.

Adicionalmente para hacer un análisis mejor sobre el conteo de las respuestas de los usuarios del sistema se realizó una valorización de las respuestas en datos numéricos que se muestran en la tabla siguiente.

Tabla 18

Valorización de respuestas del cuestionario

RESPUESTA USUARIO	VALOR ASIGNADO
Muy bueno	5
Bueno	4
Regular	3
Mala	2
Muy mala	1

Los puntajes asignados a las respuestas de los usuarios se clasificaron en intervalos al ser estos acumulados, en la siguiente tabla se muestra los intervalos considerados para el puntaje obtenido en el cuestionario.

Tabla 19

Intervalos del puntaje acumulado

INTERVALO	CALIFICACIÓN ASIGNADA
[36-44]	Muy bueno
[27-35]	Bueno
[18-26]	Regular
[9-17]	Mala
[0-8]	Muy mala



Tabla 20

Acumulado de valores obtenidos en el cuestionario

Usuario	Número de pregunta del cuestionario								Total
	P1	P2	P3	P4	P5	P6	P7	P8	
1	4	5	4	4	5	5	4	4	35
2	4	4	4	4	4	4	4	4	32
3	4	4	5	5	5	4	4	4	35
4	5	5	5	4	4	4	4	4	35
5	4	5	4	4	4	4	4	4	33
6	4	5	5	5	5	5	5	4	38
7	4	4	4	4	4	4	4	5	33
8	5	5	4	5	5	5	5	4	38
Total									279

Del cuadro anterior luego de analizar las respuestas obtenidas en el cuestionario y procesos sus datos numéricos asignados podemos ver que el sistema web con cuadro de mando, combinando tecnologías Java y R para la administración de servicios ha logrado obtener un puntaje promedio de 35 por lo según los intervalos establecidos párrafos arriba podemos concluir que el sistema tiene una calificación de “Bueno” y que ayuda a los usuarios a realizar sus tareas y a cumplir con sus objetivos.

En este capítulo validamos el sistema incluido el cuadro de mando utilizando el ISO 25010 que en la mayoría de las situaciones consideradas para la validación nos dieron buenos resultados en cuando a la reducción de tiempos, brindar interfaces amigables con los usuarios y la interoperabilidad entre el sistema y el cuadro de mando mediante API's.

4.6.6 Contraste de hipótesis

Hipótesis nula (H_0)

El desarrollo del sistema web con control de mando combinando tecnologías Java y R no reduce del tiempo promedio invertido en la administración de servicios brindados por el CITE textil camélidos Puno

Hipótesis alternativa (H_a)

El desarrollo del sistema web con control de mando combinando tecnologías Java y R reduce del tiempo promedio invertido en la administración de servicios brindados por el CITE textil camélidos Puno

Prueba estadística

Los datos obtenidos de las mediciones del tiempo empleado en todo el proceso de registro de información y generación de reportes en los años 2018 y 2019 con ayuda y sin la ayuda del sistema hacen un total de 24 datos y para determinar el estadístico de prueba estos datos deben de cumplir ciertas condiciones, los tiempos medidos se muestran en la siguiente tabla.

Tabla 21

Tabla de mediciones del tiempo empleado para la elaboración de reportes

AÑO	MES	TIEMPO EMPLEADO CON AYUDA DEL SISTEMA (Hrs.)	TIEMPO EMPLEADO SIN AYUDA DEL SISTEMA (Hrs.)
2018	Enero	6	16
2018	Febrero	11	24
2018	Marzo	13	21
2018	Abril	13	22
2018	Mayo	12	20
2018	Junio	15	23

2018	Julio	16	27
2018	Agosto	14	26
2018	Setiembre	15	22
2018	Octubre	16	27
2018	Noviembre	17	26
2018	Diciembre	15	22
2019	Enero	5	15
2019	Febrero	12	20
2019	Marzo	13	22
2019	Abril	14	23
2019	Mayo	11	25
2019	Junio	10	20
2019	Julio	9	21
2019	Agosto	13	24
2019	Setiembre	14	25
2019	Octubre	15	27
2019	Noviembre	17	25
2019	Diciembre	16	25

Utilizando R y RStudio con la librería tidyverse para hacer un gráfico de densidad de nuestros datos de tiempo medidos separados por la condición en las que se realizaron (con sistema y sin sistema), el gráfico se muestra a continuación.

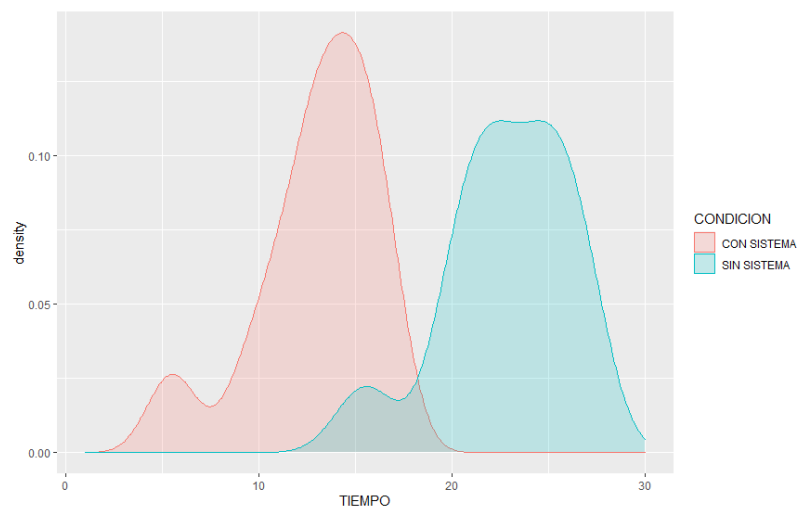


Figura 54: Gráficos de densidad de las mediciones de tiempo empleado

Del gráfico anterior podemos ver indicios de que los datos no tienen una distribución normal por lo que fue necesario validar esto realizando una prueba de normalidad a la distribución de nuestros datos. Como la cantidad de nuestros registros no superan los 30 datos se utilizó la prueba de Shapiro-Wilk para probar la hipótesis de que nuestros datos tienen una distribución normal, de igual manera utilizando el software RStudio se realizó la prueba con un valor de significancia de 0.05.

```
> datos <- read.xlsx("datos.xlsx","Hoja1")  
  
> shapiro.test(datos$TIEMPO)  
  
      Shapiro-Wilk normality test  
  
data:  datos$TIEMPO  
W = 0.95285, p-value = 0.05198
```

Figura 55: Prueba de normalidad de Shapiro-Wilk en R

Como el valor de $p > 0.05$ rechazamos la hipótesis de que nuestros datos tienen una distribución normal.

Ya que nuestros datos no presentan una distribución normal optamos por utilizar la Prueba U de Mann Withney para contrastar nuestra hipótesis de trabajo.

$$U_1 = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - R_1$$
$$U_2 = n_1 n_2 + \frac{n_2(n_2 + 1)}{2} - R_2$$

Figura 56: Formulas de la prueba Mann-Withney

Fuente: <https://www.youtube.com/watch?v=H39uMzl7le4>

Primero revisamos las medianas de las variables ya que estas son más robustas que la media que se deja influenciar por datos atípicos extremos.

```
> Tapply(TIEMPO ~ CONDICION, median, data = datos)
CON SISTEMA SIN SISTEMA
      13.5      23.0
```

Figura 57: Medianas del tiempo con sistema y sin sistema

Posterior a esto aplicamos la prueba utilizando RStudio, los resultados obtenidos se muestran en la siguiente imagen.

```
> wilcox.test(TIEMPO ~ CONDICION, data = datos)

wilcoxon rank sum test with continuity correction

data: TIEMPO by CONDICION
W = 10.5, p-value = 1.041e-08
alternative hypothesis: true location shift is not equal to 0
```

Figura 58: Prueba de Mann Withney en R

Interpretación

Utilizando la prueba de U de Mann Withney para examinar la diferencia significativa en las medianas del tiempo empleado para la elaboración de reportes con uso del sistema (Mediana = 13.5) y sin el uso del sistema (Mediana = 23.0) se encontró que el tiempo empleado utilizando el sistema es significativamente menor que el tiempo empleado sin utilizar el sistema. El valor de $p < 0.05$.



DISCUSIÓN

En este apartado realizaremos la discusión de los resultados obtenidos teniendo en cuenta los objetivos de la investigación y las investigaciones de nuestro marco teórico revisando la información obtenida en el proceso de desarrollo y validación del sistema.

Como objetivo general de la investigación se logró desarrollar e implementar un sistema web con cuadro de mando combinado tecnologías Java y R para el almacenamiento y procesamiento de datos alcanzando buena aceptación y calificación por parte de los usuarios reduciendo bastante los tiempos en el procesamiento de la información coincidiendo con la investigación de (Villanueva Castillo & Reque Valqui, 2018) que concluye que el desarrollo de un dashboard mejora la toma de decisiones estratégicas.

El uso de R con su librería Shiny para la creación del cuadro de mando permitió crear una interfaz amigable para la inspección de datos que permite la interacción con los usuarios esto coincide con lo que concluye (Condado Gómez, 2020) que nos dice que el desarrollo de una aplicación web con parámetro y variables dinámicas permite dar una mayor accesibilidad de los dato, así como facilitar al usuario que se centre en sus dudas y curiosidades.

Con la implementación del sistema se optimizó considerablemente el tiempo empleado en el registro de información y la elaboración de reportes mensuales de servicios brindados lo que concuerda con lo investigado por (López Guntiñas, 2019) que concluye que el análisis, diseño e implementación de una aplicación web Java MVC utilizando Spring Boot permitió minimizar mucho el tiempo de puesta en marcha y facilita el desarrollo de proyectos.



La interoperabilidad entre el sistema web desarrollado en Java y el cuadro de mando desarrollado en R se logró mediando la utilización de API's REST, gracias a la robustez que brinda Spring Boot y la integración que ofrece; esto coincide con (Haro et al., 2019) que concluyeron en su investigación que Spring Boot es más maduro al desarrollar API's por el tiempo que lleva en el mercado y que posee su propio conjunto de aplicaciones de seguridad e integración con base de datos para realizar aplicaciones empresariales utilizando transacciones asegurando la integridad de la información.

El cuadro de mando desarrollado muestra información de forma gráfica lo que posibilita la toma de decisiones de forma rápida ayudando a cumplir los objetivos establecidos por lo que coincidimos con lo concluido por (Nautino, 2019) que dice que los sistemas de soporte en la toma de decisiones pueden ser empleados para obtener información que ayuda a lograr los objetivos organizacionales.



V. CONCLUSIONES

PRIMERA: Se ha logrado desarrollar un sistema web con cuadro de mando, combinando tecnologías Java y R para la administración de servicios, utilizando tecnología Java para la implementación del backend y servir datos al cuadro de mando desarrollado con el lenguaje R utilizando su librería Shiny para el procesamiento de datos y presentación de información de forma gráfica a los usuarios.

SEGUNDA: La utilización de R con su librería Shiny para el desarrollo del cuadro de mando y thymeleaf con la plantilla AdminLTE para el sistema web posibilitó el desarrollo de una interfaz amigable fácil de usar para el registro e inspección de datos permitiendo la interacción de los usuarios con la información brindada en el cuadro de mando.

TERCERA: Se redujo significativamente el tiempo empleado en la elaboración de reportes mensuales con la implementación del sistema web con cuadro de mando, combinando tecnologías Java y R para la administración de servicios.

CUARTA: Se logró la interoperabilidad de un sistema desarrollado con tecnología Java utilizando el framework Spring Boot con un cuadro de mando desarrollado con el lenguaje R utilizando su librería Shiny mediante la implementación de API's que sirven



información al cuadro de mando para que este pueda procesarlas y mostrar información en forma de gráficos.

QUINTA: La implementación de un cuadro de mando permitió visualizar información de forma gráfica en tiempo real lo que permitió la toma de decisiones oportunas para el cumplimiento de objetivos institucionales.



VI. RECOMENDACIONES

PRIMERA: Se recomienda la implementación de sistemas con cuadro de mando para facilitar el proceso de almacenamiento y procesamiento de datos para ver un mejor panorama y apoyar en la toma de decisiones que lleven a los objetivos en las organizaciones que los implementen.

SEGUNDA: Es recomendable brindar interfaces amigables e intuitivas a los usuarios finales de sistemas para mejorar su experiencia y hacer más productivo su trabajo y reducir el tiempo en que aprenden a operar dicho sistema.

TERCERA: Se recomienda interoperar tecnologías en el desarrollo de sistemas para el almacenamiento y procesamiento de información ya que se puede usar las características principales y más destacadas de cada una de ellas.

CUARTA: Se recomienda mostrar cuadros de mando interactivos que presenten información pertinente e importante para la toma de decisiones en organizaciones que los utilicen como herramientas de gestión dentro de sus procesos para reducir tiempos de procesamiento y presentación.



VII. REFERENCIAS BIBLIOGRÁFICAS

Amazon Web Services Inc. (2021). *Amazon Corretto*. 2021.

<https://aws.amazon.com/es/corretto/>

Aniceto Caba, M. L. (2020). *Visualización del rendimiento en equipos de desarrollo*

Software a través de sistemas de control de versiones Git [Universidad de Sevilla].

<https://idus.us.es/handle/11441/104987>

Arcos-Medina, G., Menéndez, J., & Vallejo, J. (2018). Comparative Study of

Performance and Productivity of MVC and MVVM design patterns. *KnE*

Engineering, 1(2), 241. <https://doi.org/10.18502/keg.v1i2.1498>

Asimbaya, A. (2019). *Automatización de Pruebas Unitarias en el Marco de Desarrollo*

Ágil para Cartera de Tickets Electrónicos de Parquadero (Vol. 1, Issue 1)

[Pontificia Universidad Católica del Ecuador].

<http://201.159.222.35/handle/22000/17770>

Balbuca Ramones, D. M., & Ortíz Ramírez, J. M. (2017). *Análisis comparativos de*

servidores de aplicaciones open source para la plataforma Java EE. Caso

práctico: Módulo de Gestión de Juntas Administradoras de Agua Potable y Riego

para la Dirección Provincial de Chimborazo de la Senagua [Universidad Nacional

de Chimborazo]. <http://dspace.unach.edu.ec/bitstream/51000/3397/1/UNACH-EC->

[ING-SIS-COM-2017-0001.pdf](http://dspace.unach.edu.ec/bitstream/51000/3397/1/UNACH-EC-ING-SIS-COM-2017-0001.pdf)

Bartholomew, D. (2012). MariaDB vs. MySQL. 2, 6.

<https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/mariadb-vs-mysql/>

Castillo Peña, G. E. (2018). Implementación De Un Sistema Web De Gestión



- Documentaria En La Municipalidad Distrital De Pararin- Provincia Recuay- departamento de Ancash; 2017. [Universidad Católica los Ángeles de Chimbote].
In *Universidad Católica Los Ángeles de Chimbote*.
<http://repositorio.uladech.edu.pe/handle/123456789/2513>
- Chambilla Quispe, V. C. (2017). *Implementación y Desarrollo de Sistemas Automatizados para Mejorar la Programación y Ejecución de Procesos ed Gestión Municipal, Basados en Arquitectura Java EE - Spring* [Universidad Nacional de San Agustín]. <http://repositorio.unsa.edu.pe/handle/UNSA/4666>
- Chela LLumiguano, J. P. (2019). *Desarrollo Basado en Pruebas Unitarias de un Sistema para la Gestión de Guiones en la Radio Crisólito Celestial 93.1 FM de la Ciudad de Guaranda* [Escuela Superior Poletécnica de Chimborazo].
<http://dspace.esPOCH.edu.ec/handle/123456789/13006>
- Condado Gómez, A. (2020). *Aplicaciones Shiny para explorar datos de resultados electorales* [Universidad de Valladolid].
<https://uvadoc.uva.es/bitstream/handle/10324/43809/TFG-G4606.pdf?sequence=1&isAllowed=y>
- Cordova Espinoza, F., & Cuzco Sarango, E. (2013). *Análisis comparativo entre bases de datos relacionales con bases de datos no relacionales* (Vol. 1) [Universidad Politécnica Salesiana]. <http://dspace.ups.edu.ec/bitstream/123456789/6977/1/UPS-CT003639.pdf>
- Esteller, V., & Medina, E. (2012). Procesos De Desarrollo De Software Y Materiales Educativos Computarizados. *Revista de Tecnología de Información y Comunicación En Educación*, 6(1), 85–99.



<https://revistaeduweb.org/index.php/eduweb/article/view/229>

Fornés de Juan, J. (2021). *Arquitectura de Servicios con Docker Swarm*. Universidad Politécnica de Catalunya.

Freddy, R. F., Guido, D., Javier, C. M., & Juan, G. (2018). *Propuesta de Implementación de un Dashboard para el seguimiento de proyectos en la constructora JEMUR* [Institución Universitaria Politécnico Grancolombiano].
[https://alejandria.poligran.edu.co/bitstream/handle/10823/1737/PROPUESTA DE IMPLEMENTACION DE UN DASHBOARD PARA EL SEGUIMIENTO DE PROYECTOS EN LA CONSTRUCTORA JEMUR.pdf?sequence=1&isAllowed=y](https://alejandria.poligran.edu.co/bitstream/handle/10823/1737/PROPUESTA_DE_IMPLEMENTACION_DE_UN_DASHBOARD_PARA_EL_SEGUIMIENTO_DE_PROYECTOS_EN_LA_CONSTRUCTORA_JEMUR.pdf?sequence=1&isAllowed=y)

García Marco, F. J. (1995). Interfaces amigables para la recuperación de la información bibliográfica. *Scire: Representación y Organización Del Conocimiento*, 1(1), 127–148.

Gil, S., Ramírez, G., Muñoz, M., & González, S. (2018). Implementación de un modelo de datos para el almacenamiento de información climatológica en el estado de Querétaro. *Revista NTHE*, 24, 16–19.
http://nthe.mx/NTHE_v2/pdfArticulos/PDF_Articulo20200226192102.pdf

Guerrero Higuera, Á. M., Conde, M. Á., & Matellan, V. (2017). *Uso de sistemas de control de versiones para aplicar estrategias de evaluación por pares en contextos tecnológicos - [Using Version Control Systems to apply peer review techniques in engineering education]*. *Cinaic*, 1–6.
https://doi.org/10.26754/cinaic.2017.000001_122

Gutierrez Faraoni, F. J. (2015). *Desarrollo De Una Aplicación Web Con Spring*



- Framework Para Un Gestor De Un Recetario* [Universidad Politécnica de Madrid]. http://oa.upm.es/38731/1/TFG_Federico_Gutierrez_Faraoni.pdf
- Haro, E., Guarda, T., Zambrano Peñaherrera, A. R., & Ninahualpa Qiña, G. (2019). Desarrollo backend para aplicaciones web, Servicios Web Restful: Node.js vs Spring Boot. *Revista Ibérica de Sistemas e Tecnologías de Informação*, E17, 309–321.
<https://search.proquest.com/openview/a78cfaa62708fd24f38ac8d1025050eb/1?pq-origsite=gscholar&cbl=1006393>
- Haro Valle, V. A. (2018). *Diseño e implementación de un dashboard de soporte académico basado en datos de entornos virtuales de aprendizaje* [Universidad Politécnica de Valencia]. <https://riunet.upv.es/handle/10251/111761>
- Hurtado Chichande, D. A. (2018). *Tecnología de Contenedores de Software en Entornos de Pruebas* [Pontificia Universidad Católica del Ecuador Esmeraldas]. <https://repositorio.pucese.edu.ec/handle/123456789/1473>
- IBM. (2015). *Java Development Kit*.
<https://www.ibm.com/docs/es/i/7.3?topic=platform-java-development-kit>
- Iñiguez, L. A. (2017). *Arquitectura Tecnológica Para La Entrega Continua De Software Con Despliegue En Contenedores* [Universidad de Cuenca].
<http://dspace.ucuenca.edu.ec/handle/123456789/28529>
- Jaramillo, W. (2016). *Aplicación de la metodología RUP y el patrón de diseño MVC en la construcción de un sistema de gestión académica para la Unidad Educativa Ángel De La Guarda* (Issue 593) [Pontificia Universidad Católica del Ecuador].
<http://repositorio.puce.edu.ec/handle/22000/11264>



- Lagüera Cabrera, P. M. (2020). *Sistema de comentarios basados en GitHub Issues* [Universidad de La Laguna]. <https://riull.ull.es/xmlui/handle/915/23492>
- López Guntiñas, R. (2019). *Análisis, Diseño e Implementación de una Aplicación Web Java MVC para la Puntuación y Recomendación de Películas* [Universidad de Coruña]. <https://ruc.udc.es/dspace/handle/2183/24910>
- López Menéndez de Jiménez, R. (2015). Metodologías ágiles de desarrollo de Software aplicadas a la gestión de proyectos empresariales. *Resvisa Tecnológica* N° 8, 8, 6–11. <http://redicces.org.sv/jspui/handle/10972/2905>
- López Mollinedo, D. (2019). *Api Rest Para El Reconocimiento Facial De Emociones (Fer Rest Api)* [Universidad Central “Martha Abreu” de las Villas]. <https://dspace.uclv.edu.cu/handle/123456789/12159>
- MariaDB Foundation. (2021). *About MariaDB Server*. <https://mariadb.org/about/>
- Martín Escabia, C. J. (2018). *SportMeet: Aplicación Web para la gestión de partidas deportivas usando Spring* [Universidad de Sevilla]. <https://idus.us.es/handle/11441/83188>
- Méndez, R., & Carballeira, F. (2000). Arquitectura de la Máquina Virtual Java. *Revista Digital Universitaria*, 1. <http://www.revista.unam.mx/vol.1/num2/art4/>
- MINAGRI. (2019). *Potencial Productivo y Comercial de la Alpaca, setiembre 2019*.
- Monago Ruiz, A. (2019). *Servicio Web API REST sobre el Framework Spring, Hibernate, JSON Web Token y BBDD Oracle* [Universidad de Sevilla]. <https://idus.us.es/handle/11441/89487>



- Muños Alonso, E. (2019). *Aplicaciones Shiny como herramienta para la estadística en enseñanzas medias* [Universidad de Valladolid].
<https://core.ac.uk/download/pdf/232122847.pdf>
- Muñoz Muñoz, J. G. (2015). *Estudio comparativo de capacidad transaccional entre servidores de aplicaciones web* [Universidad de Guayaquil].
<http://repositorio.ug.edu.ec/handle/redug/11737>
- Penadés, M., & Letelier Torres, P. (2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). *Técnica Administrativa*, 5(26), 1.
- Peñarrocha Planells, M. (2016). *Una experiencia aplicando Test-Driven Development (TDD) usando una herramienta JUnit* [Universidad Politécnica de Valencia].
<https://riunet.upv.es/handle/10251/76346>
- Pol Ponsico, M. (2017). *Tecnología de Contenedores Docker* [Universidad Politécnica de Catalunya]. <https://upcommons.upc.edu/handle/2117/113040>
- Robalino, A. D. (2017). *Metodología para el diseño de Dashboards orientado hacia el registro de evidencias en el proceso de evaluaciones institucionales* [Universidad Internacional de la Rioja]. <https://reunir.unir.net/handle/123456789/6171>
- Roig-Vila, R., Antolí Martínez, J. M., Mira, J. B., Lledo, A., & Pellin, N. (2017). Uso de la herramienta GitHub en la gestión y monitorización de proyectos ABP en cuarto curso del Grado en Ingeniería Multimedia. *Redes Colaborativas En Torno a La Docencia Universitaria*, 302–308. <http://hdl.handle.net/10045/71060>
- Romero, M. (2013). Contraste de Hipótesis Comparación de dos medias independientes mediante pruebas no paramétricas : Prueba U de Mann-Whitney. *Enfermería Del*



- Trabajo, III*, 77–84. file:///Users/Yessica/Downloads/Dialnet-JornadaTecnicaDeEnfermeriaDelTrabajoCordobaFebrero-4327647.pdf
- RStudio. (2020). *Shiny*. <https://shiny.rstudio.com/>
- Sáez Romero, J. R. (2016). “ *Diseño y desarrollo de una aplicación para operadora de Telecomunicaciones para monitorizar la conexión a Internet .* ” [Universidad Politecnica de Madrid]. <https://riunet.upv.es/handle/10251/70240>
- Salazar, P. (2007). Metodologías Ágiles. *ACADEMIA Accelerating the World’s Research*, 38.
https://d1wqtxts1xzle7.cloudfront.net/53222887/Metodologias_Agiles-with-cover-page-v2.pdf?Expires=1629687671&Signature=BTHnYdDhtVN6PKMtDR3I~hOxFbzWLESj8zlWmXQFwDEkO9E7UklbWX1nT0nQJzaYfOeqOsm7euvLvRHnvQ~r18R8sD5vi3hH814vYJPTZbfC071a2XrQX8QcQJ0bYfniaYgfh58E
- Sandoval Unapucha, M. D. (2019). Sistema De Lotería De Apuestas Deportivas En El Fútbol Con Spring Framework Para La Empresa Alquimiasoft S.A. [Universidad Técnica de Ambato]. In *Repositorio Universidad Técnica de Ambato* (Vol. 53, Issue 9). <http://repositorio.uta.edu.ec/handle/123456789/30691>
- Tello Leal, E., Sosa R., C. M., & Tello Leal, D. A. (2012). Revisión de los sistemas de control de versiones utilizados en el desarrollo de software. *Ingenierías USBMed*, 3(1), 74–81. <https://doi.org/10.21500/20275846.267>
- The JUnit Team. (2019). *JUnit 5*. <https://junit.org/junit5/>
- Vanegas, C., & Pinzón, S. (2002). Carlos alberto vanegas*. *Tecnura 11*, 6, 27–34.



<https://doi.org/https://doi.org/10.14483/22487638.6129>

Vera Zegarra, R. M. (2018). *Diseño e implementación de un clúster usando JBoss EAP para aumentar la disponibilidad de los servidores de aplicaciones en una entidad del Estado* [Universidad Nacional Mayor de San Marcos].

<https://cybertesis.unmsm.edu.pe/handle/20.500.12672/10074>

Villanueva Castillo, D. H., & Reque Valqui, P. B. (2018). *Desarrollo de un Dashboard para la toma de decisiones estratégicas en la cadena de farmacias “Belén Farma” - Ancash* (Vol. 6) [Universidad Nacional del Santa].

<http://repositorio.uns.edu.pe/bitstream/handle/UNS/3227/48924.pdf?sequence=1&isAllowed=y>

Yapuchura Platero, V. (2011). *Modelo de datos multidimensionales para el diseño óptimo de base de datos* [Universidad Nacional Jorge Basadre Grohmann - Tacna].

<http://repositorio.unjbg.edu.pe/handle/UNJBG/739>



ANEXOS



Anexo 1. Cuestionario

Validación del Sistema web con cuadro de mando combinando tecnologías Java y R para la administración de servicios

*Obligatorio

1. Adecuación funcional *

Marca solo un óvalo por fila.

	Muy bueno	Bueno	Regular	Mala	Muy mala
¿El sistema le permite realizar todas sus actividades en cuanto a la información manejada en la institución?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿Los resultados que le brinda el resultado son los esperados para sus actividades dentro de la institución?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿El sistema le brinda funciones que le permiten ejecutar sus tareas dentro de la institución?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2. Usabilidad *

Marca solo un óvalo por fila.

	Muy bueno	Bueno	Regular	Mala	Muy mala
¿Las funciones y opciones que le muestra el sistema le ayuda a entender el funcionamiento del sistema?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿El tiempo empleado en aprender a usar el sistema?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿El sistema es intuitivo y fácil de manejar?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿El sistema tiene mecanismos que le alertan sobre posibles errores en su uso?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿La interfaz mostrada en el sistema le parece amigable y facilita su uso?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

https://docs.google.com/forms/d/1ZOue001o19n8zQn6s58KLHCiwmj_KqYKMclFYBKePE/viewform?edit_requested=true