



Comparación de la Norma ISO 29110 Parte 5-1-2 y el Framework SCRUM por Medio de un Sistema Web para la Gestión de Hoteles

Rodríguez Solórzano, Verónica Alexandra y Vera Gómez, Karla Cecilia

Departamento de Ciencias de la Computación

Carrera de Ingeniería en Tecnologías de la Información

Trabajo de integración curricular, previo a la obtención del título de Ingeniería en Tecnologías de la Información

Ing. Castillo Salinas, Luis Alberto

01 de marzo 2024

Reporte de Verificación de Contenido



Plagiarism and AI Content Detection Report

Tesina-RodriguezVeronicaVeraKarla....

Scan details

Scan time: March 1th, 2024 at 16:37 UTC Total Pages: 104 Total Words: 25922

Plagiarism Detection



Types of plagiarism		Words
Identical	1%	252
Minor Changes	0.2%	58
Paraphrased	0.9%	223
Omitted Words	0%	0

AI Content Detection



Text coverage		Words
AI text	0%	0
Human text	100%	25922

[Learn more](#)

🔍 Plagiarism Results: (15)

Firma:



Firmado electrónicamente por:
**LUIS ALBERTO
CASTILLO SALINAS**

.....
Ing. Castillo Salinas, Luis Alberto

Director



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Ciencias de la Computación

Carrera de Ingeniería en Tecnologías de la Información

Certificación

Certifico que el trabajo de integración curricular: “**Comparación de la Norma ISO 29110 Parte 5-1-2 y el Marco ágil SCRUM por Medio de un Sistema Web para la Gestión de Hoteles**” fue realizado por **Rodríguez Solórzano, Verónica Alexandra y Vera Gómez, Karla Cecilia** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto, cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustenten públicamente.

Santo Domingo de los Tsáchilas, 01 de marzo del 2024

Firma:



Firmado electrónicamente por:
**LUIS ALBERTO
CASTILLO SALINAS**

.....
Ing. Castillo Salinas, Luis Alberto

C. C. 1716902380



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

RESPONSABILIDAD DE AUTORÍA

Nosotros, Rodríguez Solórzano, Verónica Alexandra y Vera Gómez, Karla Cecilia, con cédulas de ciudadanía n° 1725053555 y 1315754240, declaramos que el contenido, ideas y criterios del trabajo de integración curricular: “Comparación de la Norma ISO 29110 Parte 5-1-2 y el Framework Scrum por Medio de un Sistema Web para la Gestión de Hoteles” es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Santo Domingo de los Tsáchilas, 01 de marzo del 2024

Firmas:

Rodríguez Solórzano Verónica Alexandra

C. C: 1725053555

Vera Gómez Karla Cecilia

C.C:1315754240



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

AUTORIZACIÓN DE PUBLICACIÓN

Nosotros, **Rodríguez Solórzano, Verónica Alexandra y Vera Gómez, Karla Cecilia**, con cédulas de ciudadanía n° 1725053555 y 1315754240, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: “Comparación de la Norma ISO 29110 Parte 5-1-2 y el Framework Scrum por Medio de un Sistema Web para la Gestión de Hoteles” en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Santo Domingo de los Tsáchilas, 01 de marzo del 204

Firmas:

Rodríguez Solórzano Verónica Alexandra

C. C: 1725053555

Vera Gómez Karla Cecilia

C.C:1315754240

Dedicatoria

A Dios por ser mi guía espiritual en este camino y permitirme disfrutar de este logro tan importante en esta nueva etapa de mi vida.

A mi padre, por su amor y apoyo incondicional, por ser el faro en todo este camino, gracias por creer en mí.

A mi madre, por ser esa mujer valiente, por darme apoyo y amor constantemente para seguir adelante y cumplir mis sueños académicos.

A mis padrinos por apoyarme física y emocionalmente en toda mi carrera universitaria y haber depositado su confianza en mí.

A mis hermanos y a mis cuñadas por estar siempre ayudándome en cada situación difícil que se me ha presentado.

A mí misma por ser perseverante ante cualquier dificultad y sobre todo esforzarme para llegar hasta este punto de mi vida.

Karla Cecilia Vera Gómez

Dedicatoria

Dedico este trabajo a mis padres, por ser mis mentores y guías, las personas que me enseñaron a dar mis primeros pasos y siempre me apoyaron en cada decisión.

A mi novio, por ser una nueva luz en mi camino, por apoyarme en mis momentos más difíciles y ser mi fuente de confianza.

A mi bella hija, por existir.

A la Sra. Aurita, mis hermanas y abuelitos, por apoyarme siempre y liberar mis cargas para poder seguir adelante.

A Martín, mi compañero de madrugadas, por su guía y apoyo.

A mí misma, por perseverar, luchar, seguir adelante y siempre querer superar mis objetivos hasta llegar a la meta soñada.

Verónica Alexandra Rodríguez Solórzano

Agradecimiento

Extiendo un profundo agradecimiento a quienes hicieron posible la realización de este proyecto de titulación.

Un agradecimiento especial a mi familia por su apoyo y sacrificio, gracias por ser mi apoyo incondicional en cada momento y permitirme llegar hasta aquí.

A mi tutor de tesis, Ing. Luis Castillo, por ser el eje principal en la ejecución de este proyecto, por su paciencia y su invaluable orientación que ha sido esencial en este proceso académico.

Al director de carrera, Ing. Christian Coronel, por aconsejarnos en aquel momento de oscuridad y asesorarnos ante la situación y permitirnos seguir adelante con el proceso de titulación.

A mis docentes cuya enseñanza y consejos han sido fundamentales en mi vida y desarrollo académico, gracias por ser mi guía.

A mis compañeros de aula por brindarme momentos de felicidad y alegría, sin ustedes esto habría sido diferente.

De manera especial le agradezco a mi amiga y compañera de tesis Verónica Rodríguez, por apoyarme y extenderme la mano en momentos difíciles, por no rendirse y permanecer firme hasta llegar a nuestro objetivo, gracias por estar.

Karla Cecilia Vera Gómez

Agradecimiento

Extiendo un profundo agradecimiento a la Universidad de las Fuerzas Armadas ESPE, sede Santo Domingo, por proporcionarme los instrumentos claves para el desarrollo de mi formación académica.

Al Ing. Christian Coronel, director de carrera, por ser nuestro guía, ya que con sus consejos salimos adelante con el desarrollo del trabajo de investigación.

Al revisor oponente de tesis, la Ing. Verónica Martínez, por darnos las pautas y correcciones necesarias, alentándonos a superarnos y así poder presentar este documento de la mejor manera.

Al Ing. Luis Castillo, tutor de tesis, por su constante orientación, su apoyo y paciencia que hicieron posible el desarrollo y culminación de los objetivos planteados.

A mis docentes, que a lo largo de toda la carrera me brindaron sus conocimientos, su orientación y su disposición para ayudarnos en todo el proceso de formación académica.

A mis estimados compañeros de clase que me brindaron palabras de ánimo durante todo el proceso académico.

A mi compañera de tesis Karla Vera, tú eres mi persona, mi apoyo. Agradezco mucho haberte encontrado para seguir juntas desde el primer día hasta el último.

Verónica Alexandra Rodríguez Solórzano

Tabla de Contenido

Dedicatoria	I
Agradecimiento	III
Resumen	1
Abstract	2
I. Introducción y Estado del Arte.....	3
A. <i>Estado del Arte</i>	4
B. <i>Objetivos</i>	5
1) <i>Objetivo General:</i>	5
2) <i>Objetivos Específicos:</i>	5
II. Marco Teórico	6
A. <i>Norma ISO 29110 parte 5-1-2</i>	6
B. <i>Marco ágil SCRUM</i>	7
C. <i>Angular</i>	8
D. <i>Spring Boot</i>	8
E. <i>SQL Server</i>	9
F. <i>Laravel</i>	9
G. <i>MySQL</i>	9
H. <i>SonarQube</i>	10
I. <i>Postman</i>	10
III. Metodología.....	10
A. <i>Definición de los Requerimientos</i>	10
1) <i>Requerimientos funcionales (RF):</i>	11
2) <i>Requerimientos no funcionales (RNF):</i>	11
B. <i>Implementación de SCRUM</i>	12
1) <i>Roles:</i>	12
2) <i>Product Backlog:</i>	12
3) <i>Arquitectura del SCRUM:</i>	14
4) <i>Sprint 1:</i>	15
a) <i>Planificación del Sprint 1:</i>	15
b) <i>Burndown Chart del Sprint 1:</i>	16

c)	<i>Revisión del Sprint 1:</i>	16
d)	<i>Entregables del Sprint 1:</i>	17
5)	<i>Sprint 2:</i>	19
a)	<i>Planificación del Sprint 2:</i>	19
b)	<i>Burndown Chart del Sprint 2:</i>	20
c)	<i>Revisión del Sprint 2:</i>	21
d)	<i>Entregables del Sprint 2:</i>	22
6)	<i>Sprint 3:</i>	27
a)	<i>Planificación del Sprint 3:</i>	27
b)	<i>Burndown Chart del Sprint 3:</i>	28
c)	<i>Revisión del Sprint 3:</i>	29
d)	<i>Entregables del Sprint 3:</i>	30
7)	<i>Sprint 4:</i>	35
a)	<i>Planificación del Sprint 4:</i>	36
b)	<i>Burndown Chart del Sprint 4:</i>	37
c)	<i>Revisión del Sprint 4:</i>	38
d)	<i>Entregables del Sprint 4:</i>	39
C.	<i>Implementación de la norma ISO 29110 parte 5-1-2</i>	45
1)	<i>Arquitectura MVC:</i>	46
2)	<i>Roles del SI:</i>	46
3)	<i>Evaluación de calidad del SI:</i>	47
4)	<i>Objetivos del SI:</i>	48
5)	<i>Gestión del Trabajo Realizado:</i>	60
6)	<i>Producto Final – sistema ISO</i>	61
D.	<i>Análisis del SCRUM y la norma ISO parte 5-1-2</i>	63
1)	<i>SonarQube:</i>	63
a)	<i>Análisis de SCRUM:</i>	64
b)	<i>Análisis de la norma ISO:</i>	65
2)	<i>Encuesta de Satisfacción:</i>	67
IV.	<i>Resultados</i>	69
A.	<i>Resultados de SonarQube</i>	69
B.	<i>Resultados de la encuesta de satisfacción</i>	71
C.	<i>Resultados sobre roles y responsabilidades</i>	72

D.	<i>Resultados sobre la cantidad de documentación</i>	73
E.	<i>Comparación de Resultados</i>	73
V.	Conclusiones y Recomendaciones	75
A.	<i>Conclusiones</i>	75
B.	<i>Recomendaciones</i>	76
	Referencias	77

Lista de Tablas

TABLA I REQUERIMIENTOS FUNCIONALES.....	11
TABLA II REQUERIMIENTOS NO FUNCIONALES	11
TABLA III ROLES DE SCRUM.....	12
TABLA IV PRODUCT BACKLOG	13
TABLA VI PLANIFICACIÓN DEL SPRINT 1	15
TABLA VII REVISIÓN DEL SPRINT 1	17
TABLA VIII PLANIFICACIÓN DEL SPRINT 2.....	20
TABLA IX REVISIÓN DEL SPRINT 2	22
TABLA X LISTA DE API'S DE LOS CLIENTES Y HABITACIONES	22
TABLA XI (CONTINUACIÓN) LISTA DE API'S DE LOS CLIENTES Y HABITACIONES.....	23
TABLA XII PLANIFICACIÓN DEL SPRINT 3	28
TABLA XIII REVISIÓN DEL SPRINT 3.....	30
TABLA XIV LISTA DE API'S DE LAS RECEPCIONES Y COMPROBANTE DE PAGO	30
TABLA XV PLANIFICACIÓN DEL SPRINT 4.....	36
TABLA XVI (CONTINUACIÓN) PLANIFICACIÓN DEL SPRINT 4	37
TABLA XVII REVISIÓN DEL SPRINT 4	39
TABLA XVIII LISTA DE API'S DE INICIO DE SESIÓN, ESTADÍSTICAS Y REPORTES.....	39
TABLA XIX ASIGNACIÓN DE ROLES DE PROCESOS SI.....	47
TABLA XX ESCALA DE EVALUACIÓN DE PROCESOS SI.....	47
TABLA XXI TAREAS DEL SI.....	48
TABLA XXII (CONTINUACIÓN) TAREAS DEL SI.....	49
TABLA XXIII PROGRESO DESDE EL 16/NOV/2023 HASTA EL 30/NOV/2023.....	50
TABLA XXIV PROGRESO DESDE EL 01/DIC/2023 HASTA EL 22/DIC/2023	51
TABLA XXV REGISTRO DE TRAZABILIDAD DEL SI-003	52
TABLA XXVI (CONTINUACIÓN) REGISTRO DE TRAZABILIDAD DEL SI-003	53
TABLA XXVII PROGRESO DESDE EL 23/DIC/2023 HASTA EL 13/ENE/2024.....	54
TABLA XXVIII PROGRESO DESDE EL 14/ENE/2024 HASTA EL 04/FEB/2024	55
TABLA XXIX REGISTRO DE TRAZABILIDAD DEL SI-004.....	56
TABLA XXX (CONTINUACIÓN) REGISTRO DE TRAZABILIDAD DEL SI-004.....	57
TABLA XXXI REGISTRO DE TRAZABILIDAD DEL SI-005.....	58
TABLA XXXII (CONTINUACIÓN) REGISTRO DE TRAZABILIDAD DEL SI-005	59
TABLA XXXIII PREGUNTAS DE LA ENCUESTA DE SATISFACCIÓN	68
TABLA XXXIV (CONTINUACIÓN) PREGUNTAS DE LA ENCUESTA DE SATISFACCIÓN	69
TABLA XXXV RESULTADOS DE SONARQUE MEDIANTE SQALE.....	70
TABLA XXXVI COMPARACIÓN DE ROLES Y RESPONSABILIDADES	72
TABLA XXXVII COMPARACIÓN DE LA CANTIDAD DE DOCUMENTACIÓN	73
TABLA XXXVIII COMPARACIÓN DE RESULTADOS DE LA ISO 29110 PARTE 5-1-2 Y EL MARCO ÁGIL SCRUM.....	74

Lista de Figuras

Fig. 1. Diagrama del Proceso de Implementación de Software.....	7
Fig. 2. Flujo de Trabajo en el marco ágil SCRUM	8
Fig. 3. Arquitectura Cliente - Servidor.....	14
Fig. 4. Burndown Chart del Sprint 1	16
Fig. 5. Diseño E/R de la base de datos en SQL Server	17
Fig. 6. Implementación de la base de datos en Spring Boot.....	18
Fig. 7. Configuración de las propiedades de la base de datos en Spring Boot.....	18
Fig. 8. Burndown Chart del Sprint 2	21
Fig. 9. Gestión de los clientes en el Back-end.....	23
Fig. 10. Interfaz de la gestión de los clientes en el Front-end.....	24
Fig. 11. Gestión de habitaciones en el Back-end.....	25
Fig. 12. Interfaz de gestión de habitaciones en el Front-end (1).....	26
Fig. 13. Interfaz de gestión de habitaciones en el Front-end (2).....	26
Fig. 14. Burndown Chart del Sprint 3	29
Fig. 15. Listar y guardar recepciones en el Back-end.....	31
Fig. 16. Cambiar estado de una habitación y buscar cliente por cédula en el Back-end.....	31
Fig. 17. Listar habitaciones por primer piso y segundo piso.....	32
Fig. 18 Interfaces de la gestión de recepciones en el Front-End (1)	32
Fig. 19. Interfaces de la gestión de recepciones en el Front-End (2)	33
Fig. 20. Formulario para guardar una recepción	34
Fig. 21. Gestión de los comprobantes de pago en el Back-end.....	34
Fig. 22. Interfaz del comprobante de pago en el Front-end.....	35
Fig. 23. Burndown Chart del Sprint 4	38
Fig. 24. Gestión de las estadísticas claves en el Back-end.....	40
Fig. 25. Interfaz de las estadísticas claves en el Front-end.....	40
Fig. 26. Listar usuario en el Back-end.....	41
Fig. 27. Interfaces de inicio de sesión en el Front-end (1)	41
Fig. 28. Interfaces de inicio de sesión en el Front-end (2)	42
Fig. 29. Gestión de los reportes en el Back-end.....	42
Fig. 30. Interfaz de los reportes en el Front-end.....	43
Fig. 31. Vista previa de las interfaces del sistema (1).....	43
Fig. 32. Vista previa de las interfaces del sistema (2).....	44
Fig. 33. Vista previa de las interfaces del sistema (3).....	44
Fig. 34. Vista previa de las interfaces del sistema (4).....	45
Fig. 35. Vista previa de las interfaces del sistema (5).....	45
Fig. 36.Arquitectura MVC	46
Fig. 37. Gráfico Burn-Up Chart	60
Fig. 38. Inicio de sesión en el sistema web	61
Fig. 39. Módulos funcionales	62
Fig. 40. Módulo de Recepciones	62
Fig. 41. Funcionalidades del sistema.....	63
Fig. 42. Análisis del sistema SCRUM en el Front-end	64
Fig. 43. Análisis del sistema SCRUM en el Back-end.....	65

Fig. 44. Análisis del sistema de la norma ISO en el Front-end 66
Fig. 45. Análisis del sistema de la norma ISO en el Back-end..... 67
Fig. 46. Resultados de Cobertura y Duplicaciones..... 71
Fig. 47. Respuestas de la encuesta de satisfacción..... 71

Siglas, Acrónimos y Abreviaturas

Back-end	Procesar toda la información que alimenta a un front-end
CRUD	Crear, Leer, Actualizar, Eliminar
Front-end	Desarrollo del lado del cliente
HTTP	Protocolo de transferencia de hipertexto
ISO	Organización Internacional de Normalización
MVC	Modelo-Vista-Controlador
PHP	Preprocesador de hipertexto
PM	Project Management (Gestión de Proyectos)
PyMEs	Pequeñas y medianas empresas
RDBMS	Sistema de gestión de bases de datos relacionales
SI	Software Implementation (Implementación de Software)
SPA	Solicitud de una sola página
SQALE	Evaluación de la calidad del software basada en las expectativas del ciclo de vida
SQL	Lenguaje de consulta estructurado
VSE	Entidades muy pequeñas

Resumen

Este trabajo se enfocó en realizar la comparativa de la norma ISO 29110 parte 5-1-2 contra el marco ágil SCRUM mediante el desarrollo de sistemas web para la gestión hotelera. El objetivo fue determinar que metodología se adapta de mejor manera para el desarrollo de software en la gestión hotelera evaluando la calidad del software, la satisfacción del cliente, los roles y documentación generada. El sistema SCRUM utilizó una arquitectura cliente-servidor, utilizando Spring Boot y Angular con una base de datos SQL Server. Por otro lado, el sistema con la norma ISO se desarrolló con una arquitectura MVC, utilizando el framework Laravel con base de datos MySQL. Los resultados muestran que SCRUM supera a la norma ISO en seguridad, fiabilidad y mantenibilidad (valores de A-B frente a A-D). La duplicidad de código es menor en SCRUM (6,8%) que en la norma ISO (8,5%). La satisfacción del usuario es alta en ambos, con un 91,5% para SCRUM y un 94,5% para la norma ISO. En cuanto a la documentación y roles, la norma ISO genera más (10 documentos y 7 roles) que SCRUM (6 documentos y 3 roles). En síntesis, aunque SCRUM y la norma ISO se ajustaron a los requerimientos, SCRUM gestionó de mejor manera su tiempo obteniendo mayor calidad. Sin embargo, no es posible determinar el mejor enfoque debido a que los resultados se dieron a partir de un sistema para un cliente específico. Esta comparativa proporciona un impacto para la elección del marco de trabajo en futuros proyectos.

***Palabras Clave* — Marco ágil SCRUM, ISO 29110 parte 5-1-2, Spring Boot, Angular, Laravel, Sistema web hotelero.**

Abstract

This work focused on comparing the ISO 29110 part 5-1-2 standard against the agile SCRUM framework through the development of web systems for hotel management. The objective was to determine which methodology is best suited for software development in hotel management by evaluating the quality of the software, customer satisfaction, roles and the documentation generated. The SCRUM system used a client-server architecture, using Spring Boot and Angular with a SQL Server database. On the other hand, the system with the ISO standard was developed with an MVC architecture, using the Laravel framework with a MySQL database. The results show that SCRUM exceeds the ISO standard in security, reliability and maintainability (values of A-B versus A-D). Code duplication is lower in SCRUM (6.8%) than in the ISO standard (8.5%). User satisfaction is high in both, at 91.5% for SCRUM and 945 for ISO. Regarding documentation and roles, the ISO standard generates more (10 documents and 7 roles) than SCRUM (6 documents and 3 roles). In summary, although SCRUM and the ISO standard adjusted to the requirements, SCRUM managed its time better, obtaining higher quality. However, it is not possible to determine the best approach because the results were obtained from a system for a specific client. This comparison provides an impact for the choice of framework in future projects.

***Keywords* — SCRUM agile framework, ISO 29110 part 5-1-2, Spring Boot, Angular, Laravel, Hotel web system.**

I. Introducción y Estado del Arte

Hoy en día, la industria del software reconoce el valor de las VSE al aportar productos y servicios de calidad. Una VSE es una entidad (empresa, organización, departamento o proyecto) que tiene hasta 25 personas. Dado que, las VSE desarrollan o mantienen software que más adelante se utiliza en sistemas más grandes, es necesario reconocer a las VSE como proveedores de software de alta calidad.

La norma ISO 29110, parte 5-1-2, y el marco ágil SCRUM son enfoques metodológicos reconocidos por su eficacia y eficiencia en la gestión de proyectos mediante el empleo de métodos productivos. La norma ISO es aplicable a entidades muy pequeñas y la parte 5-1-2 se aplica a proyectos de desarrollo de software y se destina a ser utilizado en cualquier proceso, técnica y método que mejore la satisfacción y productividad del cliente de VSE. Por otro lado, SCRUM es un marco de trabajo liviano que ayuda a personas, equipos y organizaciones a generar valor mediante sus soluciones adaptativas para problemas complejos. Por lo consiguiente, este marco de trabajo se puede emplear para varios procesos, técnicas y métodos haciendo visible su eficacia en cuanto a técnicas de gestión, entorno y trabajo.

El objetivo de la presente investigación es comparar dos sistemas web, creados a partir de los requerimientos específicos del Hotel Puerta de Oro y desarrollados de manera independiente, con distintas arquitecturas, frameworks y base de datos, esto, con el fin de emplear la norma ISO y el marco ágil SCRUM y determinar cuál enfoque es más acertado para realizar sistemas VSE, mediante la cantidad de documentación, la cantidad de roles, calidad del software y satisfacción del usuario.

Por consiguiente, en el sistema a desarrollar con el marco ágil SCRUM, se optará por una arquitectura Cliente-Servidor, en el Back-end se empleará el framework Spring Boot, mientras que en el Front-end se utilizará Angular, simultáneamente, se integrará una base de datos SQL Server. Mientras que, el sistema a desarrollar con la norma ISO, seguirá únicamente el proceso de Implementación de Software, para ello se optará por una arquitectura MVC, mediante el framework Laravel y como base de datos, se utilizará MySQL.

En la sección de introducción y estado del arte, se ofrece una visión general de los objetivos planteados, así como una exploración del estado actual del campo. El marco teórico posterior, aborda en detalle los fundamentos de la norma ISO 29110, parte 5-1-2, el marco ágil SCRUM y las tecnologías esenciales como Angular, Spring Boot, SQL Server, Laravel, MySQL y SonarQube.

La metodología empleada se desglosa en el levantamiento de requerimientos y la implementación de SCRUM y la norma ISO 29110 en dos sistemas específicos. Luego, se analizan los resultados obtenidos en términos de calidad del software, satisfacción del usuario final, cantidad de documentación y cantidad de roles. Finalmente, las conclusiones extraídas y recomendaciones propuestas brindan una visión integral de los hallazgos de esta investigación.

A. Estado del Arte

Según [1] las empresas dependen de su capacidad para proporcionar sistemas de calidad con el fin de demostrar su competitividad en el mercado. Debido a lo expuesto, el estudio se enfoca en aplicar la norma ISO 29110 5-1-2, enfocada en pequeñas empresas, a partir de dos grupos experimentales, con el objetivo de determinar las mejores prácticas en el proceso de implementación del software. El estudio concluye que la implementación de la norma, en el proceso de Implementación de Software, resulta en un desarrollo más destacado, lo que se convierte en una gran ventaja competitiva para la empresa.

Por otro lado, en [2] se identificó un problema en una aplicación web para contabilidad, en la cual se trabajó en dos variables para culminar con éxito el proyecto. Estas variables fueron el alcance de los requerimientos y el cumplimiento del tiempo estimado. Para lograr esto, se implementó SCRUM y los perfiles básicos de la ISO 29110, lo cual permitió alcanzar el objetivo, a pesar de los fallos en el tiempo y en la satisfacción de los requerimientos.

Según [3], en su trabajo de graduación titulado “Uso de la metodología SCRUM en el desarrollo de aplicaciones web”, se analizó cómo SCRUM puede mejorar la eficiencia y calidad en el desarrollo de aplicaciones web. La autora destacó los componentes de SCRUM, como los roles (dueño del producto, equipo de desarrollo y SCRUM master), los artefactos (pila del producto, pila del Sprint e incremento) y los eventos (Sprint, planificación del Sprint, SCRUM diario, revisión del Sprint y retrospectiva). Además, se discutieron las ventajas y desventajas de SCRUM y se abordaron las ventajas de las aplicaciones web en comparación con otros enfoques de desarrollo. El estudio concluyó que la metodología SCRUM puede ser altamente beneficiosa en el desarrollo de aplicaciones web, mejorando la comunicación, la flexibilidad y la adaptabilidad del equipo de desarrollo.

Según [4], se utilizó la metodología SCRUM para el desarrollo del software y se describe cómo se aplicó en el proyecto de reserva de hoteles. También se incluyen detalles sobre el diseño,

la implementación y las pruebas del sistema web. En la sección de anexos, se pueden encontrar diagramas y tablas adicionales que pueden ser relevantes para la investigación.

En conclusión, tanto la norma ISO 29110 como la metodología SCRUM presentaron limitaciones en su implementación. Por ejemplo, en el estudio [2], se mencionan fallos en el cumplimiento del tiempo y la satisfacción de los requerimientos, a pesar de la implementación de SCRUM. Lo anterior sugiere que estas herramientas son útiles, pero no garantizan el éxito absoluto y pueden necesitar ajustes adicionales en el contexto del proyecto. Por otro lado, en [1] se destaca que si bien la norma ISO 29110 parte 5-1-2 consta de dos procesos interrelacionados, la Gestión de Proyectos (PM) y la Implementación de Software (SI), es importante destacar que el SI puede ser implementado de manera independiente, sin necesidad de llevar a cabo el PM. Esto significa que, aunque el PM proporciona el plan de proyecto como producto de entrada para el SI, este último no está obligado a basarse exclusivamente en el plan generado por el PM, lo que permite una mayor flexibilidad en el proceso. Los trabajos expuestos nos brindan perspectivas valiosas acerca de la aplicación práctica y los resultados obtenidos con la implementación de la norma ISO 29110 parte 5-1-2 y el marco ágil SCRUM en el desarrollo de sistemas web, lo cual se demuestra con las comparaciones en roles, documentación, calidad del software y satisfacción del cliente.

B. Objetivos

1) Objetivo General:

Realizar la comparación de la norma ISO 29110 parte 5-1-2 y el marco ágil SCRUM por medio de un sistema web para la gestión de hoteles.

2) Objetivos Específicos:

- Desarrollar un sistema web de gestión de hoteles utilizando el proceso de Implementación de Software (SI) de la norma ISO 29110, parte 5-1-2.
- Desarrollar un sistema web de gestión de hoteles utilizando la guía completa del marco ágil SCRUM.
- Realizar la comparación de la implementación de la norma ISO 29110 parte 5-1-2 y el marco ágil SCRUM para el desarrollo de los dos sistemas.

II. Marco Teórico

A. Norma ISO 29110 parte 5-1-2

La norma ISO 29110 [5] proporciona un modelo fundamental para el desarrollo de software en entidades muy pequeñas (VSE). La norma se enfoca en abordar las limitaciones y necesidades específicas de estas empresas, y está organizada en perfiles, cada uno adaptado a las características particulares de distintos tipos de organizaciones. Esta norma consta de las siguientes partes:

- **Parte 1:** Descripción general [informe Técnico]
- **Parte 2:** Marco y Taxonomía
- **Parte 3:** Guía de evaluación [Informe Técnico]
- **Parte 4-1:** Especificación del perfil: grupo de perfiles genéricos
- **Parte 5-1-2:** Guía de gestión e ingeniería: Grupo de perfiles genéricos: Perfil básico [Informe técnico]

Para cumplir con los requisitos específicos del perfil elegido, se ofrecen guías de procesos detalladas en cada perfil, lo que posibilita la adaptabilidad y escalabilidad del marco. La norma ISO 29110 “Ingeniería de software - Perfiles de ciclo de vida para entidades muy pequeñas”, es una serie de estándares y guías preparadas por los comités: Técnico, de Tecnologías de la Información y de Ingeniería de Software y Sistemas [5].

Este estudio utiliza la norma ISO 29110, parte 5-1-2 como guía. "Aquí, el dígito '5' corresponde al número de sección, el dígito '1' se refiere al grupo de perfil general y el dígito '2' identifica el perfil básico" [6]. Primeramente, nos basamos en la definición de dos procesos fundamentales, esenciales para este último perfil: Gestión de Proyectos (PM) e Implementación de Software (SI). Para el desarrollo de la metodología se hará uso, únicamente, de la sección correspondiente a la Implementación de Software, la cual, como se muestra en la **Fig. 1**, se centra en el desempeño sistemático de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software, de acuerdo con los requisitos especificados. El diagrama presentado a continuación ilustra cómo se intercambia información a lo largo de las diversas actividades involucradas en el proceso de Implementación de Software. Este también destaca los productos de trabajo clave y la forma en que se relacionan entre sí.

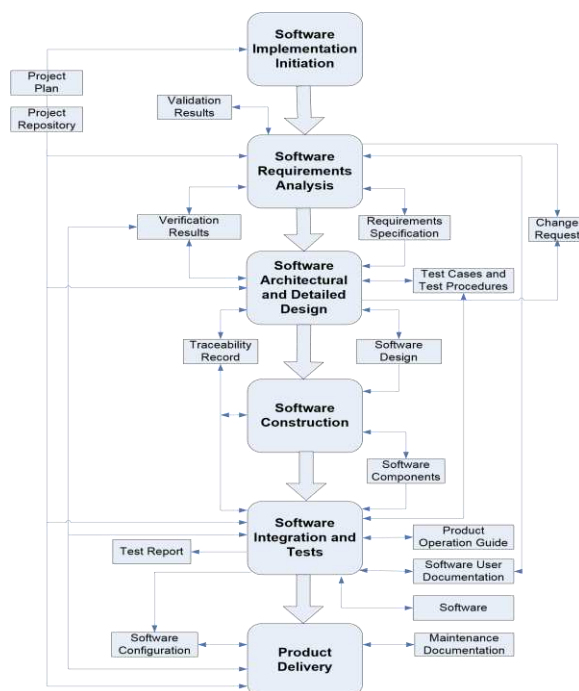


Fig. 1. Diagrama del Proceso de Implementación de Software [5]

B. Marco ágil SCRUM

SCRUM es un marco ágil diseñado para gestionar proyectos complejos de manera eficiente. Busca maximizar el valor para el cliente a través de la colaboración, la transparencia y la adaptación continua. Este marco incorpora partes esenciales como el propietario del producto, el SCRUM Master y el equipo de desarrollo, quienes trabajan en estrecha colaboración para maximizar la entrega de valor al cliente. Los eventos esenciales de SCRUM (Sprint, planificación, reuniones diarias, revisiones y retrospectiva) brindan oportunidades vitales para realizar pruebas y ajustes continuos. Los artefactos, como Product Backlog, Sprint Backlog e Increment, aumentan la transparencia y permiten revisiones y ajustes continuos. Las reglas de SCRUM son fáciles de entender y claras, pero su aplicación práctica puede presentar dificultades. SCRUM es un marco incremental e iterativo que se centra en la entrega oportuna de valor al cliente y en la mejora continua de procesos y productos. Su enfoque ágil promueve la flexibilidad y la eficiencia en el desarrollo de productos complejos [7].

El flujo de trabajo de SCRUM (**Fig. 2**) comienza con la planificación del Sprint, etapa en la que se seleccionan las tareas a realizar. Durante el Sprint, el Equipo de Desarrollo realiza reuniones diarias con el objetivo de coordinar y planificar sus actividades, asegurando la finalización de las tareas definidas para ese Sprint. Al concluir el Sprint, se lleva a cabo la Revisión

del Sprint, donde se evalúa el trabajo realizado (Incremento) y se realizan los ajustes necesarios en el Product Backlog, si es necesario, para seguir avanzando hacia la entrega del producto planificado. Este ciclo fomenta una entrega continua de valor, garantizando que el producto se desarrolle de manera eficiente y que permanezca alineado con las necesidades de los usuarios y los objetivos del negocio.

Flujo de Trabajo en Scrum

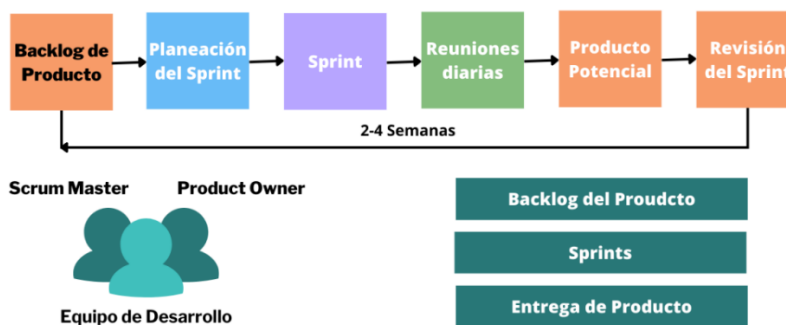


Fig. 2. Flujo de Trabajo en el marco ágil SCRUM [8]

C. Angular

Angular, desarrollado por Google, actualmente en la versión 16.2.11, es un marco de código fuente abierto para crear aplicaciones web potentes y dinámicas. Se basa en componentes reutilizables, utiliza TypeScript para aumentar la capacidad y tiene la ventaja de permitir que las aplicaciones se construyan en una sola página (SPA) que se carga solo una vez, lo que permite interacciones dinámicas sin necesidad de recargar la página. Angular es una opción potente y versátil para el desarrollo eficiente de aplicaciones en línea modernas y escalables, gracias a sus características como el enlace de datos bidireccional, enrutamiento, manejo de formularios y pruebas unitarias integradas [9].

D. Spring Boot

Al proporcionar una configuración predeterminada inteligente centrada en convenciones sobre configuración, Spring Boot, en su versión 3.2.0, simplifica el desarrollo de aplicaciones Java, según la guía de referencia de Spring Boot [10]. Reduce la necesidad de configuración manual, lo que permite a los desarrolladores concentrarse en la lógica empresarial. Destaca por su capacidad de modificar la configuración de acuerdo con las necesidades particulares. Utiliza un servidor web integrado para gestionar las dependencias con Maven o Gradle más fácilmente y eliminar la

necesidad de servidores externos. En resumen, Spring Boot simplifica el desarrollo al combinar funciones útiles, flexibilidad para ajustarse a requisitos particulares y configuración intuitiva.

E. SQL Server

Microsoft SQL Server, el Servidor de Lenguaje de Consulta Estructurado, se destaca como un sistema de gestión de bases de datos relacionales (RDBMS) vital en el mundo empresarial. La versión específica utilizada en este contexto es Microsoft SQL Server 2022 - 16.0.1000.6, ya que ofrece un entorno robusto para el archivo y recuperación eficiente de datos gracias al uso del lenguaje SQL. Sus capacidades incluyen procedimientos almacenados, seguridad avanzada, copia de seguridad y replicación, ofreciendo una solución integral para la gestión de datos.

SQL Server se incluye en discusiones teóricas sobre la importancia de las bases de datos relacionales y la normalización de datos. Su adaptabilidad y escalabilidad les permiten satisfacer una variedad de necesidades comerciales, desde aplicaciones de pequeña escala hasta sistemas de información empresarial de gran escala. La discusión teórica de SQL Server destaca su papel crucial en la organización y gestión eficaz de la información en entornos empresariales.

F. Laravel

El marco PHP de vanguardia Laravel, creado por Taylor Otwell, redefine los estándares para el desarrollo de aplicaciones web. Utilizando específicamente la versión 10.39.0 de Laravel, este método ágil proporciona flexibilidad y eficacia, destacando especialmente por ofrecer una alternativa directa al convencional Modelo-Vista-Controlador (MVC) para manejar las respuestas a solicitudes HTTP [11]. Este enfoque, que se alinea con la filosofía fundamental de Laravel de "hacer las cosas de manera fácil y sencilla", simplifica el código, mejora la legibilidad y acelera el desarrollo de aplicaciones en línea [12]. Dentro del marco teórico y con la implementación de la versión 10.39.0, Laravel se posiciona como una herramienta innovadora que no solo se adhiere a las convenciones establecidas, sino que también sugiere un enfoque más refinado para promover la agilidad y la eficiencia en el desarrollo web en el panorama tecnológico actual.

G. MySQL

MySQL se destaca como uno de los sistemas de gestión de bases de datos más utilizados en el ámbito de las aplicaciones web. [13] menciona que su facilidad de uso y continua actualización con nuevas funcionalidades y medidas de seguridad lo convierten en una opción favorita. En particular, la versión 8.2.0 de MySQL, que se ha elegido para este sistema, refleja este compromiso con la velocidad, seguridad y la incorporación de nuevas características avanzadas,

manteniendo el equilibrio entre las funcionalidades ofrecidas y los recursos necesarios. Aunque existen versiones de pago diseñadas para uso comercial, la versión gratuita, especialmente la 8.2.0, destaca por ofrecer mayor velocidad y seguridad al omitir numerosas funciones adicionales. Dependiendo de las necesidades particulares del administrador de datos, este elemento puede verse como positivo o negativo [14]. La versatilidad de MySQL queda demostrada por su capacidad para seleccionar y administrar datos en una amplia gama de tipos de tablas, ofreciendo un alto nivel de seguridad y generando reputación mediante el uso eficaz de los recursos del servidor. Este sistema se presenta como la opción ideal para desarrolladores independientes que buscan una solución confiable con un presupuesto limitado [15].

H. SonarQube

SonarQube, en su versión 10.4.0, será utilizado específicamente para evaluar los procesos establecidos por la norma ISO 29110 parte 5-1-2 y el marco ágil SCRUM. Esta herramienta estará enfocada en ofrecer métricas de rendimiento, evaluación de calidad y monitoreo de proyectos. La selección de SonarQube es fundamental para asegurar una comparación objetiva y exitosa entre ambos métodos, ya que destaca por su capacidad para evaluar continuamente la calidad del código y su efectiva integración con métodos ágiles. Esta versión de SonarQube se ha seleccionado debido a sus características mejoradas y compatibilidad con las últimas prácticas de desarrollo, lo que la hace especialmente adecuada para nuestros propósitos de análisis de la calidad del software.

I. Postman

Postman, en su versión 10.23.9, se utilizará para probar la comunicación de servicios web a través de APIs creadas en Spring Boot. Postman es una herramienta de desarrollo y colaboración que ayuda a los desarrolladores a interactuar y probar aplicaciones y servicios web, permitiendo enviar solicitudes a servidores web, organizar solicitudes en colecciones, gestionar entornos de desarrollo, automatizar pruebas de rendimiento y mejorar la colaboración en el ciclo de vida de las API. [16]

III. Metodología

A. Definición de los Requerimientos

Previo a la realización de los sistemas de acuerdo con la norma ISO 29110, parte 5-1-2 y el marco ágil SCRUM, se realizó un análisis exhaustivo con el objetivo de identificar y documentar los requisitos tanto funcionales como no funcionales del sistema.

1) *Requerimientos funcionales (RF):*

Los requisitos funcionales se centran en las necesidades específicas del usuario, enfocándose en la gestión de habitaciones, clientes, procesos de recepción y generación de informes, como se puede observar en la **TABLA I**.

TABLA I
REQUERIMIENTOS FUNCIONALES

ID	REQUERIMIENTOS FUNCIONALES
RF1	El sistema solo es utilizado por una persona encargada de la recepción de clientes.
RF2	La página de inicio debe mostrar estadísticas clave, como el total de habitaciones, habitaciones ocupadas y disponibles.
RF3	Se deben permitir operaciones CRUD para la gestión de habitaciones, clientes, estado de habitación, ubicación y categoría de la habitación.
RF4	Las habitaciones deben clasificarse por categorías (simple, doble, triple y suite) y por ubicación (primer y segundo piso). Además, se debe permitir cambiar el estado de las habitaciones (disponible y ocupado).
RF5	El pago se realiza en el momento que se ingresa la recepción y se acepta solo pago en efectivo.
RF6	El sistema debe generar un comprobante de pago para los clientes al finalizar la recepción.
RF7	Deben generarse un informe sobre las recepciones dentro de un rango de fecha establecida con el total de ingresos, descargables en formato PDF.

2) *Requerimientos no funcionales (RNF):*

Los requisitos no funcionales establecen los parámetros de calidad y limitaciones del sistema, incluyendo aspectos como la seguridad, el rendimiento y la usabilidad. Estos se describen de manera más detallada en la **TABLA II**.

TABLA II
REQUERIMIENTOS NO FUNCIONALES

ID	REQUERIMIENTOS NO FUNCIONALES
RNF1	El acceso a funciones administrativas debe ser seguro.
RNF2	Los datos personales de los clientes deben almacenarse de manera segura.
RNF3	La interfaz de usuario debe ser intuitiva y fácil de usar.
RNF4	El sistema debe cargar la información de habitaciones, clientes, estado de las habitaciones, categoría y recepción de manera eficiente.
RNF5	La generación de reportes y comprobantes debe ser rápida y eficiente.
RNF6	El sistema debe ser fácilmente actualizable sin afectar la disponibilidad.
RNF7	El sistema debe ser escalable para manejar un aumento gradual en el número de habitaciones, clientes, ubicaciones, categorías, estado de habitación y recepción del cliente.

B. Implementación de SCRUM

Se describe la implementación del marco ágil SCRUM para la administración del ciclo de vida del SCRUM y se analiza su impacto en el desarrollo del sistema de gestión de hoteles.

1) Roles:

Los roles esenciales en el equipo de desarrollo SCRUM fueron claramente definidos, abarcando al SCRUM Master, el Product Owner y el Equipo de Desarrollo. Cada uno de estos roles asume responsabilidades específicas, contribuyendo así a asegurar una implementación efectiva del marco ágil SCRUM. En la **TABLA III**, se detalla la asignación de roles clave para la implementación del marco ágil SCRUM, indicando las responsabilidades asociadas a cada función y los responsables para desempeñar estos roles.

TABLA III
ROLES DE SCRUM

Roles	Responsabilidad	Responsable
SCRUM Master	Coordinar el equipo y facilitar el proceso.	Ing. Luis Castillo
Product Owner	Representar las necesidades del cliente y definir prioridades.	Vilma Jacho
Equipo de Desarrollo	Desarrollar y entregar incrementos de producto.	Karla Vera

2) Product Backlog:

El Product Backlog desempeñó un papel central en la captura y priorización de los requisitos del sistema, sirviendo como una herramienta clave en la planificación. En colaboración con el Product Owner, se utilizó para recopilar y organizar de manera sistemática las funcionalidades y mejoras deseadas para el sistema de gestión de hoteles. Este proceso facilitó la claridad en las prioridades, permitiendo al equipo de desarrollo abordar los elementos del backlog de manera eficiente y adaptarse a las necesidades cambiantes del sistema a lo largo del ciclo de vida. Como se puede observar en la **TABLA IV**, la planificación de tareas para el desarrollo del sistema se detalla indicando la identificación única (ID), la lista de tareas, una breve descripción de cada tarea, la prioridad asignada, el tiempo estimado en días para completar cada tarea y el orden planificado.

TABLA IV
PRODUCT BACKLOG

ID	Lista de Tareas	Descripción Breve	Prioridad	Tiempo Estimado (días)	Secuencia
PB01	Diseñar la base de datos del sistema	Definir la estructura de la base de datos, identificar relaciones clave y establecer las tablas necesarias para respaldar la gestión hotelera.	1	7	1
PB02	Implementación de la base de datos	Crear la base de datos según el diseño establecido, asegurando la coherencia y la eficiencia en el manejo de datos.	2	8	2
PB06	Gestionar clientes	Desarrollar funciones para agregar, editar y eliminar información de clientes, proporcionando una interfaz fácil de usar para la gestión de la base de datos de clientes.	Alta	8	3
PB05	Gestionar las habitaciones (Incluye operaciones CRUD, clasificación por categorías, detalle y pisos, y el estado de la habitación)	Implementar operaciones CRUD para habitaciones, permitiendo la clasificación por categorías y pisos, y facilitando el cambio de estado de las habitaciones según su disponibilidad.	Alta	12	4
PB07	Gestionar las recepciones	Permitir gestionar las recepciones de las habitaciones, integrando la funcionalidad de facturación al finalizar cada reserva.	Alta	11	5
PB08	Generar comprobante de pago	Integrar la funcionalidad de generación del comprobante de alojamiento, asegurando un proceso eficiente y preciso.	Medio	11	6
PB10	Generar reportes de las reservas en PDF	Implementar la generación de informes detallados sobre las reservas, permitiendo la descarga en formato PDF para su fácil acceso y almacenamiento.	Medio	9	9
PB04	Dashboard con estadísticas clave	Diseñar un panel de control que presente estadísticas esenciales, proporcionando una visión general sobre las habitaciones disponibles, ocupadas y las que se encuentran en limpieza.	Medio	6	7
PB09	Crear la interfaz de inicio de sesión	Diseñar la interfaz de inicio de sesión	Bajo	6	6

Una vez que se hayan definido los requerimientos funcionales y no funcionales, se procedió a establecer el Product Backlog. Los detalles de este proceso están contenidos en el documento adjunto, "Product Backlog".

3) *Arquitectura del SCRUM:*

La arquitectura que se utilizará en el SCRUM es la de cliente-servidor (**Fig. 3**). En el Back-end, se emplea Spring Boot, mientras que para el Front-end se utiliza Angular, que es el encargado de consumir las APIs establecidas en el controlador REST de Spring Boot mediante el cliente HTTP.

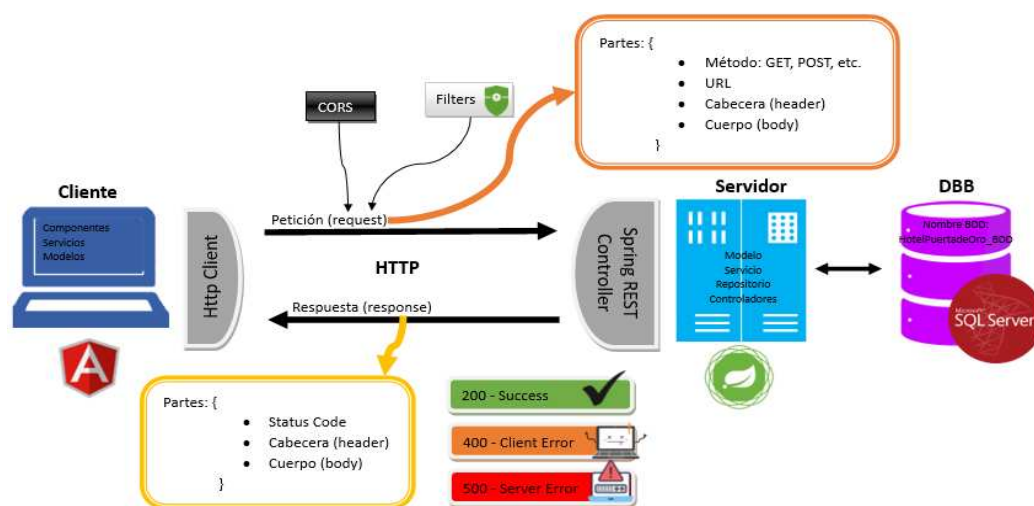


Fig. 3. Arquitectura Cliente - Servidor

La **Fig. 3**, presenta una visión detallada de los diversos componentes y tecnologías implementados en el SCRUM, el cual es una aplicación de microservicios desarrollada usando Spring Boot para el Back-end, Angular para el Front-end y SQL Server como sistema de gestión de base de datos. Este diagrama muestra la interacción entre las capas de cliente y servidor, incluyendo los elementos de la lógica de negocio y de la base de datos. En la capa de cliente, se destacan el HTTP Cliente, CORS y filtros específicos para gestionar los diferentes estados de respuesta HTTP (200 para respuestas exitosas, 400 para errores del cliente y 500 para errores del servidor). La arquitectura del lado del cliente incluye servicios, modelos y componentes, mientras que el lado del servidor está estructurado en controladores, repositorios, servicios y modelos. Finalmente, la base de datos, nombrada "HotelPuertadeOro_BDD" y hospedada en SQL Server, constituye el soporte de datos para la aplicación.

4) *Sprint 1:*

En el Sprint 1, su enfoque principal se centró en el diseño e implementación de la base de datos del SCRUM. Durante este periodo, el equipo de desarrollo se dedicó a la fase inicial, abordando la estructura fundamental de la base de datos. Se llevaron a cabo varias subtareas, como la identificación de entidades, la definición de atributos y relaciones, así como la implementación efectiva de la base de datos. A continuación, se presentan los detalles del Sprint 1:

- **Objetivos:** Diseñar e implementar la base de datos del sistema.
- **Duración del Sprint 1:** 2 semanas (16/11/2023 - 30/11/2023).
- **Responsable:** Karla Vera.

a) *Planificación del Sprint 1:*

Cada tarea se ha desglosado en subtareas, incluyendo la identificación de entidades, definición de atributos, establecimiento de relaciones, la presentación de un diseño E/R y la implementación de la base de datos en SQL Server. En la **TABLA V**, se muestra el detalle de las subtareas de las tareas PB01 y PB02, presentando el tiempo estimado en días para cada una de ellas y el tiempo total estimado para completar el Sprint 1.

TABLA V
PLANIFICACIÓN DEL SPRINT 1

SPRINT PLANNING		SPRINT 1 (2 SEMANAS)	
ID	Ítem	Requerimiento/Tarea	Tiempo Estimado (Días)
PB01		Diseñar la base de datos del sistema:	7
	PB01.1	Identificar las entidades para el sistema	
	PB01.2	Definir los atributos para cada entidad	
	PB01.3	Establecer las relaciones necesarias	
	PB01.4	Crear el diseño de E/R	
	PB01.5	Revisar el diseño	
	PB01.6	Documentar	
PB02		Implementación de la base de datos:	8
	PB02.1	Escoger un motor de base de datos	
	PB02.2	Crear la base de datos para el sistema	
	PB02.3	Crear las entidades en la base de datos con sus atributos	
	PB02.4	Establecer las relaciones entre las entidades	
	PB02.5	Realizar consultas.	
	PB02.6	Documentar	
Duración del sprint (días)			15

b) *Burndown Chart del Sprint 1:*

Para visualizar el progreso a lo largo del Sprint 1, se implementó el Burndown Chart correspondiente, basándonos en el Sprint Backlog que registra el avance de cada actividad específica de cada tarea a medida que avanzamos en el Sprint 1. En la **Fig. 4**, se ilustra la proyección del trabajo realizado durante el Sprint 1 a lo largo de los días. La línea naranja representa las horas planificadas, mientras que la línea azul indica las horas reales completadas cada día.

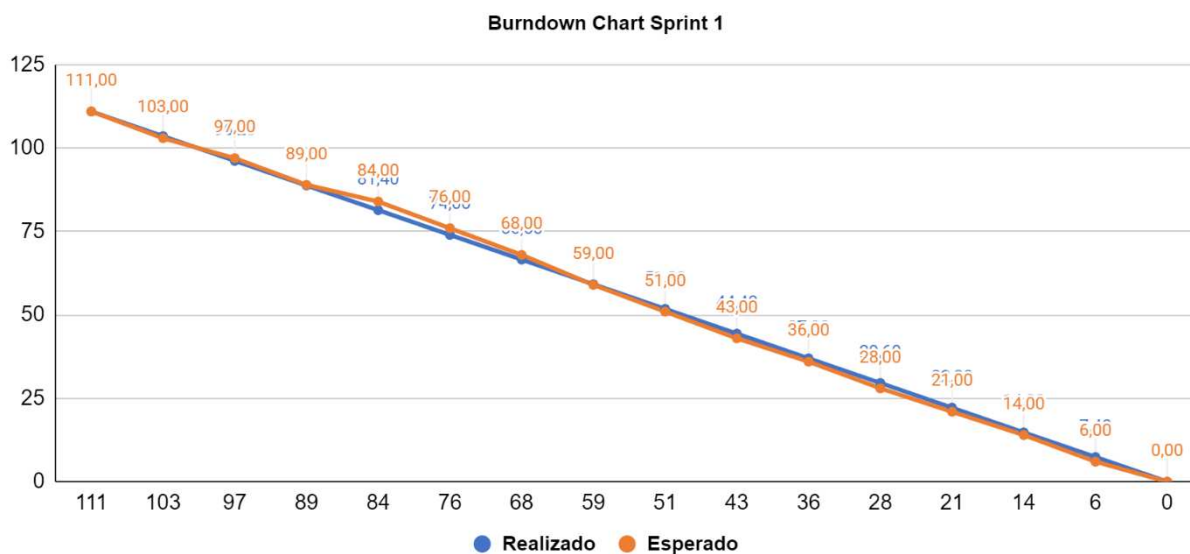


Fig. 4. Burndown Chart del Sprint 1

La **Fig. 4** muestra que, en los primeros 3 días, el equipo progresa a un ritmo que coincide con las expectativas basadas en el plan original. Sin embargo, se percibe un retraso en el trazado del gráfico durante los 3 días siguientes, reflejando un avance más lento de lo previsto, especialmente en las etapas de creación y revisión del modelo Entidad-Relación (E/R). A pesar del retraso, al final del Sprint, durante los últimos 9 días, el equipo de desarrollo consiguió ajustar su enfoque y acelerar el trabajo, culminando todas las tareas programadas con éxito.

c) *Revisión del Sprint 1:*

Al finalizar el Sprint 1, se realizó la revisión de este, constituyendo una evaluación reflexiva tanto de los procesos como de los resultados obtenidos. Durante el Sprint 1, el enfoque estuvo en el diseño e implementación exitosa de la base de datos del sistema (**TABLA VI**), logrando completar todas las historias planificadas. A pesar de alcanzar los objetivos establecidos respecto al diseño e implementación de la base de datos, alojada en SQL Server, se identificaron ciertos fallos al momento de establecer las relaciones entre las entidades del modelo.

TABLA VI
REVISIÓN DEL SPRINT 1

SPRINT REVIEW	SPRINT 1
Ítem	Descripción
Objetivos del Sprint 1	Diseñar e implementar la base de datos del sistema.
Historias Completadas	Todas
Demostración	Presentación de la base de datos implementada.
Feedback	Se recomienda identificar primero que es lo que se necesita, y en base a eso realizar un bosquejo a mano.
Fecha	30/11/2023
Retrospectiva	Lo que salió bien: Diseño E/R completado según lo planeado Que se debe mejorar: Establecer correctamente las relaciones entre las entidades con la finalidad de obtener un modelo bien definido.

d) *Entregables del Sprint 1:*

A continuación, compartimos el resultado del diseño (**Fig. 5**) y la implementación (**Fig. 6**, **Fig. 7**) de la base de datos para el SCRUM, como parte del entregable del Sprint 1. Este diagrama es fundamental para comprender la estructura de la base de datos, resaltando las relaciones cruciales entre las entidades clave. Cabe destacar que este diseño se encuentra implementado en SQL Server, añadiendo una capa adicional de robustez a nuestra arquitectura y la implementación se realizó en Spring Boot con su respectiva configuración hacia la misma.

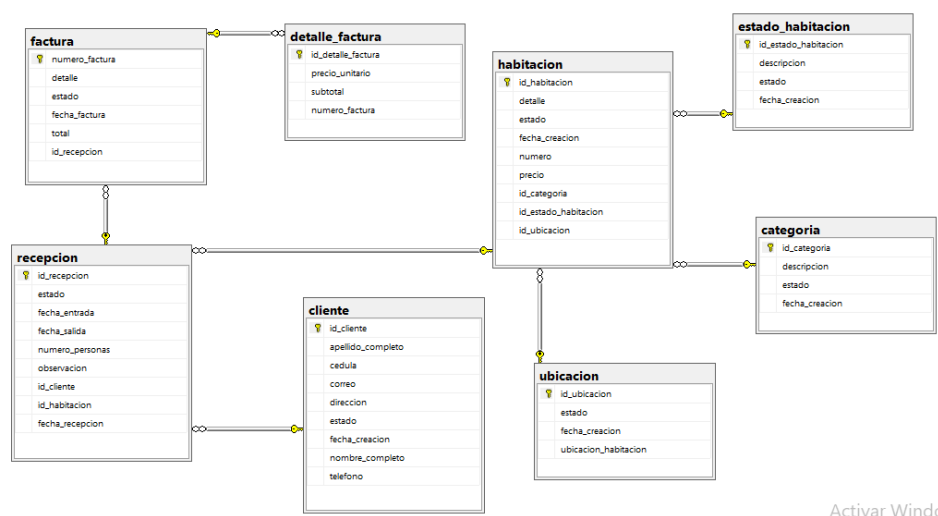


Fig. 5. Diseño E/R de la base de datos en SQL Server

En la **Fig. 5**, se puede observar el diseño de la base de datos que consta de 8 tablas: habitaciones, estado, ubicación, categoría, recepción, cliente, factura y detalle de la factura. Cada tabla representa una entidad fundamental en el SCRUM y se han establecido relaciones estratégicas entre ellas para facilitar la gestión eficiente de la información.

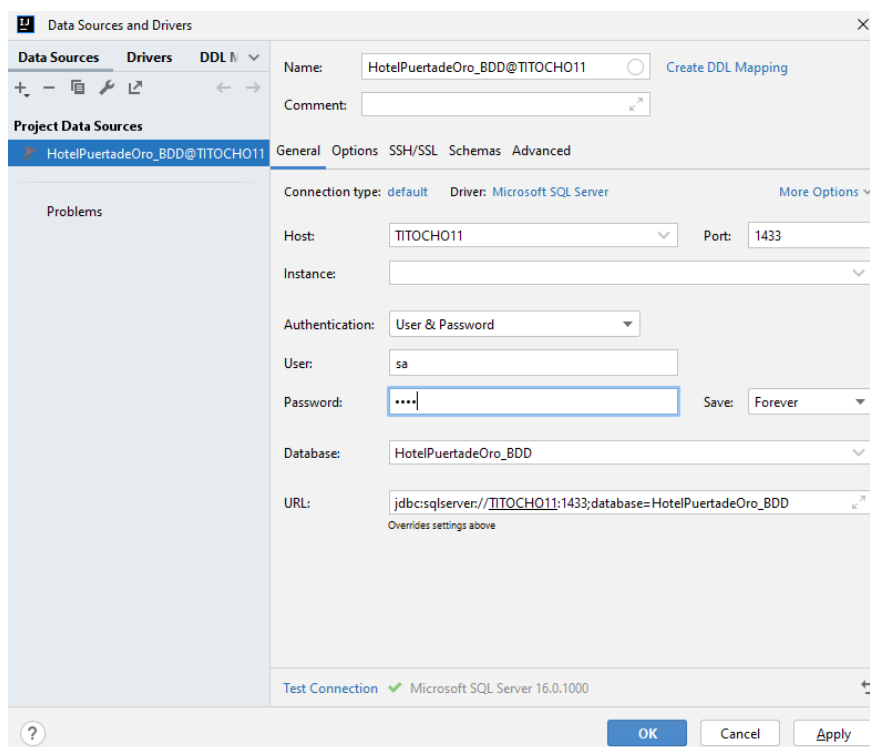


Fig. 6. Implementación de la base de datos en Spring Boot

En la **Fig. 6**, se presentan los elementos clave para establecer con éxito la conexión hacia la base de datos. El proceso inicia configurando el host, que corresponde al nombre del servidor, y el puerto (1433), el cual facilita la conexión con SQL Server. En el proceso de autenticación, se emplea el usuario 'sa' junto con su correspondiente clave. Posteriormente, se lleva a cabo la selección de la base de datos específica que se utilizará. Finalmente, se ejecuta la prueba de conexión, la cual asegura que todos los parámetros necesarios estén correctamente establecidos, garantizando así una interacción fluida entre la aplicación y la base de datos en SQL Server.

```
spring.datasource.url=jdbc:sqlserver://;serverName=TITOCHO11;databaseName=HotelPuertadeOro_BDD
spring.datasource.username=sa
spring.datasource.password=tita
spring.datasource.driverClassName=com.microsoft.sqlserver.jdbc.SQLServerDriver
spring.jpa.hibernate.dialect=org.hibernate.dialect.SQLServerDialect
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=update
spring.datasource.show-sql=true
```

Fig. 7. Configuración de las propiedades de la base de datos en Spring Boot

En la **Fig. 7**, se definen los parámetros clave para conectar la aplicación a una base de datos SQL Server en el entorno de desarrollo. La URL de conexión incluye el nombre del servidor y de la base de datos, mientras que las credenciales de autenticación son el usuario 'sa' con su contraseña. Otras configuraciones abordan detalles como el controlador JDBC, el dialecto de Hibernate y la visualización de consultas SQL en la consola. La opción `spring.jpa.hibernate.ddl-auto=update` permite la actualización automática del esquema de la base de datos.

Completando el primer Sprint hemos conseguido los objetivos marcados. Los detalles de este proceso están contenidos en el documento adjunto, "*Sprint [1]*", el cual describe cada etapa, desde la planificación hasta la implementación, culminando exitosamente con los objetivos planteados en el inicio.

5) *Sprint 2:*

En el Sprint 2, su enfoque principal se centró en la gestión de clientes y habitaciones. Se planificaron las sub tareas, que incluyeron el desarrollo de interfaces intuitivas, validación de información, funciones CRUD, pruebas y documentación. A continuación, se presentan los detalles del Sprint 2:

- Objetivos: Gestionar clientes y habitaciones.
- Duración del Sprint 2: 3 semanas (01/12/2023 - 22/12/2023).
- Responsable: Karla Vera.

a) *Planificación del Sprint 2:*

Cada tarea se ha desglosado en subtareas, incluyendo las operaciones CRUD para la gestión de clientes, habitaciones, ubicaciones, estados y categorías. En la **TABLA VII**, se muestra el análisis detallado de las subtareas de las tareas PB05 y PB04, presentando el tiempo estimado en días para cada una de ellas, así como el tiempo total estimado para completar el Sprint 2.

TABLA VII
PLANIFICACIÓN DEL SPRINT 2

SPRINT PLANNING		SPRINT 2 (3 SEMANAS)	
ID	Ítem	Requerimiento/Tarea	Tiempo Estimado (Días)
PB05		Gestionar los clientes	
	PB05.1	Desarrollar una interfaz intuitiva para la gestión de clientes	
	PB05.2	Validar el ingreso de información en el formulario	
	PB05.3	Visualizar la información de los clientes	
	PB05.4	Desarrollar función para editar clientes	10
	PB05.5	Desarrollar la función de eliminar clientes	
	PB05.6	Desarrollar la función de agregar clientes	
	PB05.7	Realizar pruebas	
	PB05.8	Documentar	
PB04	Gestionar las habitaciones (Incluye operaciones CRUD, clasificación por categorías y pisos, y el estado de la habitación)		
	PB04.1	Desarrollar una interfaz intuitiva para la gestión de habitaciones, la clasificación por categoría, detalle y pisos, y el estado de la habitación	
	PB04.2	Validar el ingreso de información en el formulario	
	PB04.3	Visualizar la información de las habitaciones, categorías, detalle pisos y el estado de la habitación.	
	PB04.4	Desarrollar función para editar las habitaciones, categorías, detalle pisos y el estado de la habitación.	12
	PB04.5	Desarrollar la función de eliminar las habitaciones, categorías, pisos, detalle y el estado de la habitación.	
	PB04.6	Desarrollar la función de agregar las habitaciones, categorías, detalle pisos y el estado de la habitación.	
	PB04.7	Realizar pruebas	
		Documentar	
Duración del sprint (días)			22

b) *Burndown Chart del Sprint 2:*

Para visualizar el progreso a lo largo del Sprint 2, se implementó el Burndown Chart correspondiente, basándonos en el Sprint Backlog que registra el avance de cada actividad específica de cada tarea a medida que avanzamos en el Sprint 2. En la **Fig. 8**, se ilustra la proyección del trabajo realizado durante el Sprint 2 a lo largo de los días. La línea naranja representa las horas planificadas, mientras que la línea azul indica las horas reales completadas cada día.

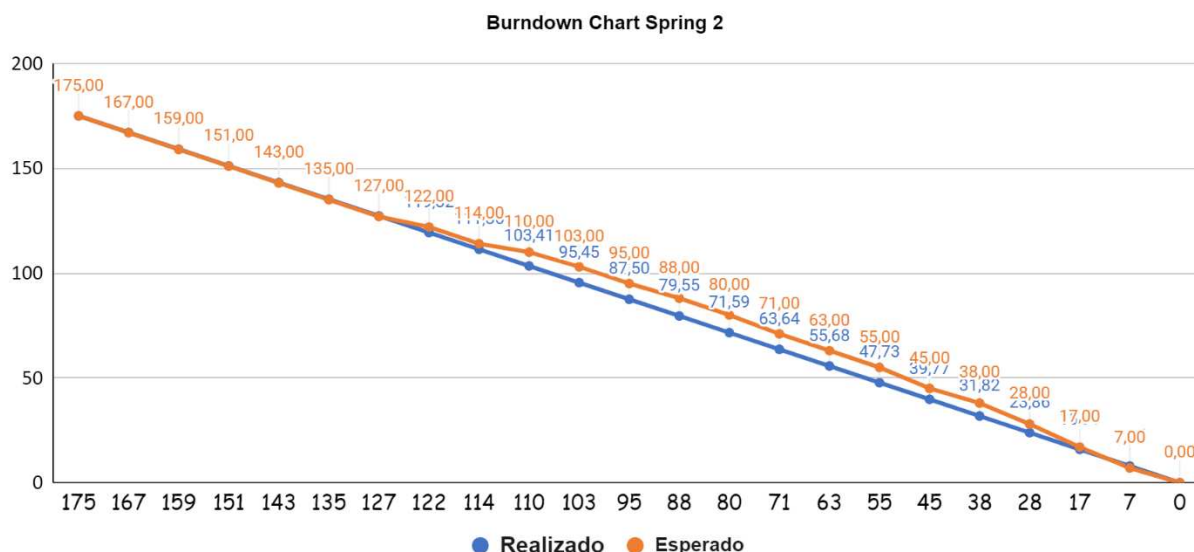


Fig. 8. Burndown Chart del Sprint 2

En la **Fig. 8**, muestra que, los primeros 6 días, el equipo progresa a un ritmo que coincide con las expectativas basadas en el plan original. Sin embargo, se percibe un retraso en el trazado del gráfico durante los 13 días siguientes, reflejando un avance más lento de lo previsto. Este retraso se atribuye principalmente a las dificultades encontradas al finalizar las tareas relacionadas con la gestión de clientes y habitaciones, las cuales se complicaron por problemas al extraer los datos de la base de datos. A pesar del retraso, al final del Sprint, durante los últimos 3 días, el equipo de desarrollo consiguió ajustar su enfoque y acelerar el trabajo, culminando todas las tareas programadas con éxito.

c) Revisión del Sprint 2:

Al finalizar el Sprint 2, se llevó a cabo la revisión, una evaluación reflexiva de los procesos y resultados obtenidos. Esta fase proporcionó la gestión de los clientes y de las habitaciones, donde se demostraron las operaciones CRUD de estas con éxito, sin embargo, se tienen algunas sugerencias sobre validar la información que se ingresa en los formularios y esto se puede observar en la **TABLA VIII**

TABLA VIII
REVISIÓN DEL SPRINT 2

SPRINT REVIEW	SPRINT 2
Ítem	Descripción
Objetivos del Sprint 2	Gestionar los clientes y las habitaciones
Historias Completadas	Fueron completadas las tareas, pero con algunas recomendaciones en las validaciones con los formularios y el control de los modals.
Demostración	Operaciones CRUD para gestionar los clientes y las habitaciones Se sugiere enfocarse en la implementación de una sólida validación de formularios para garantizar la integridad de los datos. Además, mejorar la gestión de modales al editar o agregar nuevos datos para proporcionar una experiencia de usuario más fluida e intuitiva.
Feedback	
Fecha	22/12/2023
Retrospectiva	Lo que salió bien: Demostración de las operaciones CRUD de la gestión de clientes y las habitaciones Que se debe mejorar: Validar el ingreso de información en los formularios y controlar los modales al editar o agregar nuevos datos.

d) Entregables del Sprint 2:

A continuación, presentamos la implementación de la gestión de clientes y habitaciones, abarcando tanto el Back-end como el front-end, como parte del entregable del Sprint 1. En esta sección, se establecen las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) para garantizar una completa funcionalidad en la interacción con la información de clientes y habitaciones en el sistema. En la **TABLA IX**, se encuentran las APIs establecidas tanto para los clientes como para las habitaciones (estados, ubicación, categoría), donde se proporcionan los métodos HTTP permitidos y los códigos de estado HTTP que indican el resultado de cada operación para las APIs relacionadas con clientes y habitaciones (estados, ubicación, categoría) en el sistema.

TABLA IX
LISTA DE API'S DE LOS CLIENTES Y HABITACIONES

Tabla	Descripción	APIs	Tipo	Resultado
Cliente	Listar clientes	/api/HPO/clientes/listar	GET	Status: 200 OK
Cliente	Guardar cliente	/api/HPO/clientes/guardar	POST	Status: 200 OK
Cliente	Actualizar cliente	/api/HPO/clientes/actualizar/{id}	PUT	Status: 200 OK
Cliente	Eliminar cliente	/api/HPO/clientes/eliminar/{id}	DELETE	Status: 200 OK
Estado	Listar estados	/api/HPO/estadoHabitacion/listar	GET	Status: 200 OK
Estado	Guardar estados	/api/HPO/estadoHabitacion/guardar	POST	Status: 200 OK
Estado	Actualizar estados	/api/HPO/estadoHabitacion/actualizar/{id}	PUT	Status: 200 OK

TABLA X
(CONTINUACIÓN) LISTA DE API'S DE LOS CLIENTES Y HABITACIONES

Estado	Eliminar estados	/api/HPO/estadoHabitacion/eliminar/{id}	DELETE	Status: 200 OK
Categoría	Listar categoría	/api/HPO/categorias/listar	GET	Status: 200 OK
Categoría	Guardar categoría	/api/HPO/categorias/guardar	POST	Status: 200 OK
Categoría	Actualizar categoría	/api/HPO/categorias/actualizar/{id}	PUT	Status: 200 OK
Categoría	Eliminar categoría	/api/HPO/categorias/eliminar/{id}	DELETE	Status: 200 OK
Ubicación	Listar ubicación	/api/HPO/ubicaciones/listar	GET	Status: 200 OK
Ubicación	Guardar ubicación	/api/HPO/ubicaciones/guardar	POST	Status: 200 OK
Ubicación	Actualizar ubicación	/api/HPO/ubicaciones/actualizar/{id}	PUT	Status: 200 OK
Ubicación	Eliminar ubicación	/api/HPO/ubicaciones/eliminar/{id}	DELETE	Status: 200 OK
Habitación	Listar habitación	/api/HPO/habitaciones/listar	GET	Status: 200 OK
Habitación	Guardar habitación	/api/HPO/habitaciones/guardar	POST	Status: 200 OK
Habitación	Actualizar habitación	/api/HPO/habitaciones/actualizar/{id}	PUT	Status: 200 OK
Habitación	Eliminar habitación	/api/HPO/habitaciones/eliminar/{id}	DELETE	Status: 200 OK

La gestión de clientes se implementa tanto en el Back-end (Fig. 9) como en el Front-end (Fig. 10), como parte del Sprint 2. Esto abarca las operaciones esenciales para manejar información de clientes desde el servidor y la interfaz de usuario.

(a) **POST** localhost:8080/api/HPO/clientes/guardar

Params Authorization Headers (9) Body Pre-request Script

none form-data x-www-form-urlencoded raw

```

1 {
2   "nombreCompleto": "Karla D",
3   "apellidoCompleto": "Cecilia",
4   "cedula": "1315754240",
5   "correo": "kcovera@espe.edu.ec",
6   "telefono": "0990503575",
7 }

```

(b) **DELETE** localhost:8080/api/HPO/clientes/eliminar/2

Params Authorization Headers (9) Body Pre-request Script

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```

1 {
2   "eliminar": true
3 }

```

(c) **PUT** localhost:8080/api/HPO/clientes/actualizar/1002

Params Authorization Headers (9) Body Pre-request Script

none form-data x-www-form-urlencoded raw binary

```

1 {
2   "nombreCompleto": "Verónica",
3   "apellidoCompleto": "Rodríguez",
4   "cedula": "1725053655",
5   "correo": "varodriguez2@espe.edu.ec",
6   "telefono": "0995384761",
7   "direccion": "dato a actualizar"
8 }

```

(d) **GET** localhost:8080/api/HPO/clientes/listar

Params Authorization Headers (9) Body Pre-request Script

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```















1 [
2   {
3     "idCliente": 1,
4     "nombreCompleto": "Karla D",
5     "apellidoCompleto": "Cecilia",
6     "cedula": "1315754240",
7     "correo": "kcovera@espe.edu.ec",


```

Fig. 9. Gestión de los clientes en el Back-end. (a) Guardar cliente. (b) Eliminar cliente. (c) Actualizar cliente. (d) Listar clientes

En la **Fig. 9**, se muestran las operaciones CRUD utilizadas en la gestión de clientes, las cuales fueron probadas mediante Postman. (Fig. 9a) Método POST, empleado para guardar un nuevo cliente. (Fig. 9b) Método DELETE, utilizado para eliminar un cliente existente. (Fig. 9c) Método PUT, para actualizar la información de un cliente específico. (Fig. 9d) Método GET, que permite listar todos los clientes presentes en la base de datos, la cual está alojada en SQL Server.

(a) **Nuevo Cliente** Generar y descargar reporte

N°	Cliente	Cedula	Correo	Teléfono	Dirección	Acciones
1	Karla D Cecilia	1315754240	kcvera@espe.edu.ec	0990503575	AV. 3 DE JULIO	 
2	Verónica Rodríguez	1725053555	varodriguez2@espe.edu.ec	0995384761	dato a actualizar	 
3	wdw asasa	7852154512	rosa@gmail.com	0981045834	Quito	 
4	Noelia Anchundia	1316408622	noe@gmail.com	0980733471	Santo Domingo	 
5	Juan Perez Perez	1234567890	juan.perez@example.com	0987654321	Calle Falsa 123	 
6	Juan Perez Perez	1234567890	juan.perez@example.com	0987654321	Calle Falsa 123	 
7	Juan Perez Perez	1234567890	juan.perez@example.com	0987654321	Calle Falsa 123	 

(b) 
¿Estas seguro?
 Confirma si deseas eliminar al cliente

(c) **Nuevo Cliente** x

Nombres Ingrese los nombres del cliente	Apellido Ingrese los apellidos del cliente
Cédula Ingrese la cédula del cliente	Teléfono Ingrese el teléfono del cliente
Correo Ingrese el correo del cliente	Dirección Ingrese la dirección del cliente

(d) **Actualizar Cliente** x

Nombres Karla D	Apellido Cecilia
Cédula 1315754240	Teléfono 0990503575
Correo kcvera@espe.edu.ec	Dirección AV. 3 DE JULIO

Fig. 10. Interfaz de la gestión de los clientes en el Front-end. (a) Listar clientes. (b) Eliminar cliente. (c) Nuevo cliente. (d) Actualizar cliente

En la **Fig. 10**, se presenta la interfaz dedicada a la gestión de clientes, la cual facilita la realización de las operaciones CRUD. (Fig. 10a) Se observa una tabla que lista a los clientes almacenados en la base de datos, junto a un área de acciones con dos botones: uno de color amarillo con un icono de pincel para que el usuario actualice la información de un cliente, y otro de color rojo para su eliminación. Además, en la esquina superior izquierda de la tabla, se ubica un botón de color verde etiquetado como ‘Nuevo Cliente’, permitiendo agregar un nuevo cliente a la base de datos y en la parte superior derecha de la tabla se tiene un botón de color verde ‘Generar y descargar reporte’, permitiendo descargar los clientes almacenados en un documento de EXCEL. (Fig. 10b) Cuadro de diálogo para eliminar un cliente, el cual ofrece las opciones ‘Sí, eliminar’ y ‘No, cancelar’. (Fig. 10c) Modal con el formulario para ingresar un nuevo cliente, teniendo en

cuenta que todos los campos están validados para asegurar la precisión y la integridad de la información ingresada por el usuario. (Fig. 10d) Modal con el formulario para actualizar los datos de un cliente existente, permitiendo así que se modifique cualquier atributo del cliente según sea necesario.

La gestión de habitaciones se implementa tanto en el Back-end (**Fig. 11**) como en el Front-end (**Fig. 12, Fig. 13**), como parte del entregable del Sprint 2. Esto incluye funcionalidades clave para administrar las habitaciones.

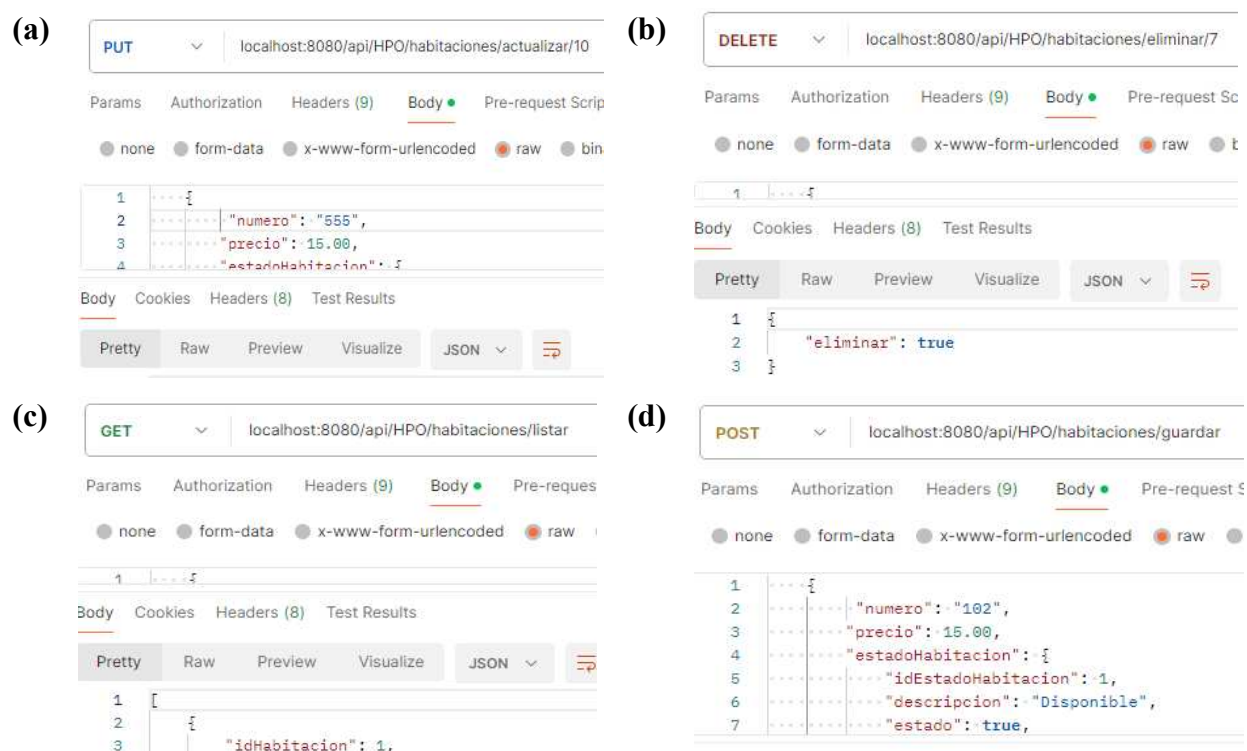


Fig. 11. Gestión de habitaciones en el Back-end. (a) Actualizar habitación. (b) Eliminar habitación. (c) Listar habitaciones. (d) Guardar habitaciones

En la **Fig. 11**, se muestran las operaciones CRUD utilizadas en la gestión de las habitaciones, las cuales fueron probadas mediante Postman. (Fig. 11a) Método POST, empleado para guardar una nueva habitación. (Fig. 11b) Método DELETE, utilizado para eliminar una habitación existente. (Fig. 11c) Método GET, que permite listar todas las habitaciones presentes en la base de datos, la cual está alojada en SQL Server. (Fig. 11d) Método PUT, para actualizar la información de una habitación específica.

(a) Nueva Habitación

N°	Número de Habitación	Precio	Estado	Ubicación	Categoría	Detalle	Acciones
1	102	15	Disponible	Primer Piso	Simple	Aire acondicionado, 2 camas de 2p1/2.	 
2	101	15	Disponible	Primer Piso	Doble	Aire acondicionado, 2 camas de 2p1/2.	 
3	103	15	Ocupado	Primer Piso	Doble	Aire acondicionado, 2 camas de 2p1/2.	 
4	104	10	Disponible	Primer Piso	Triple	Ventilador, 3 camas (2 de 2 plazas y 1 de 1 plaza y media)	 
5	105	10	Ocupado	Primer Piso	Doble	Ventilador, 2 camas de plaza y media.	 

(b)



¿Estas seguro?
Confirma si desea eliminar la habitación

Si, elimínalo **No, cancelar**

Fig. 12. Interfaz de gestión de habitaciones en el Front-end (1). (a) Listar habitaciones. (b) Eliminar habitación

En la **Fig. 12**, presenta la interfaz dedicada a la gestión de habitaciones, especialmente la de listar y eliminar las habitaciones. (Fig. 12a) Se observa una tabla que lista las habitaciones disponibles en la base de datos, junto a un área de acciones con dos botones: uno de color amarillo con un icono de pincel para que el usuario actualice la información de una habitación, y otro de color rojo para su eliminación. Además, en la esquina superior izquierda de la tabla, se ubica un botón de color verde etiquetado como ‘Nueva Habitación’, permitiendo agregar una nueva habitación a la base de datos. (Fig. 12b) Cuadro de diálogo de confirmación para eliminar una habitación, el cual ofrece las opciones ‘Sí, eliminar’ y ‘No, cancelar’.

(a) Nueva Habitación ×

Número de Habitación
Ingrese el número de la habitación

Precio
Ingrese el precio de la habitación

Estado de la Habitación

Ubicación de la Habitación

Categoría de la Habitación

Detalle de la Habitación

Cerrar **Guardar**

(b) Actualizar Habitación ×

Número de Habitación
102

Precio
15

Estado de la Habitación
Disponible

Ubicación de la Habitación
Primer Piso

Categoría de la Habitación
Simple

Detalle de la Habitación
Aire acondicionado, 2 camas de 2p1/2.

Cerrar **Actualizar**

Fig. 13. Interfaz de gestión de habitaciones en el Front-end (2). (a) Nueva habitación. (b) Actualizar habitación

En la **Fig. 13**, presenta la interfaz dedicada a la gestión de habitaciones, especialmente la de listar y eliminar las habitaciones. (Fig. 13a) Modal con el formulario para ingresar una nueva habitación, teniendo en cuenta que todos los campos están validados para asegurar la precisión y la integridad de la información ingresada por el usuario. (Fig. 13d) Modal con el formulario para actualizar los datos de una habitación existente, permitiendo así que se modifique cualquier atributo de la habitación según sea necesario.

Completando el segundo Sprint hemos conseguido los objetivos marcados para esta fase del SCRUM. Los detalles de este proceso están contenidos en el documento adjunto, "*Sprint [2]*", el cual describe cada etapa, desde la planificación hasta la implementación, culminando exitosamente con su finalización.

6) *Sprint 3:*

En el Sprint 3, el enfoque principal está centrado en la gestión eficiente de las recepciones y generar su respectivo comprobante de alojamiento. Se han planificado las correspondientes subtareas para llevar a cabo este Sprint. A continuación, se detallan los aspectos clave del Sprint 3:

- Objetivos: Gestionar las recepciones y el comprobante de pago.
- Duración del Sprint 3: 3 semanas (23/12/2023 - 13/01/2024).
- Responsable: Karla Vera.

a) *Planificación del Sprint 3:*

En esta sección, se detallan las tareas específicas y las subtareas que abordaremos en el Sprint 3. Cada tarea contribuirá al logro de los objetivos establecidos, centrados en la gestión de recepciones y el comprobante de pago. La **TABLA XI** muestra el análisis detallado de las subtareas de las tareas PB06 y PB07, incluyendo el tiempo estimado en días para cada una y el tiempo total estimado para completar el Sprint 3.

TABLA XI
PLANIFICACIÓN DEL SPRINT 3

SPRINT PLANNING		SPRINT 3 (3 SEMANAS)	
ID	Ítem	Requerimiento/Tarea	Tiempo Estimado (Días)
PB06		Gestionar las recepciones	
PB06.1		Realizar una interfaz intuitiva para la gestión de las recepciones	
PB06.2		Mostrar las habitaciones disponibles y ocupada, para así hacer las recepciones	
PB06.3		Permitir observar las habitaciones por la ubicación (Primer Piso, Segundo Piso)	
PB06.4		Permitir observar el total de las recepciones realizadas, con la opción de ver el comprobante.	15
PB06.5		Permitir seleccionar un cliente existente al momento de hacer la recepción	
PB06.7		Agregar la función de guardar las recepciones	
PB06.8		Realizar pruebas	
PB06.9		Documentar	
PB07		Gestionar comprobante de la recepción	
PB07.1		Realizar una interfaz intuitiva para el comprobante	
PB07.2		Integrar la generación automática de los comprobantes al guardar una recepción.	
PB07.3		Incluir un desglose detallado de los costos de la habitación en el comprobante.	7
PB07.6		Configurar la opción para ver y descargar de nuevo los comprobantes de pago de las recepciones realizadas	
PB07.7		Realizar pruebas	
PB07.8		Documentar	
Duración del sprint (días)			22

b) Burndown Chart del Sprint 3:

El Burndown Chart del Sprint 3, represent gráficamente, muestra el progreso del equipo en relación con las tareas planificadas durante el transcurso del Sprint. En la **Fig. 14**, proporciona una visualización clara de la velocidad del equipo en términos de completar las sub tareas asignadas a lo largo del tiempo. La línea naranja representa las horas planificadas, mientras que la línea azul indica las horas reales completadas cada día.

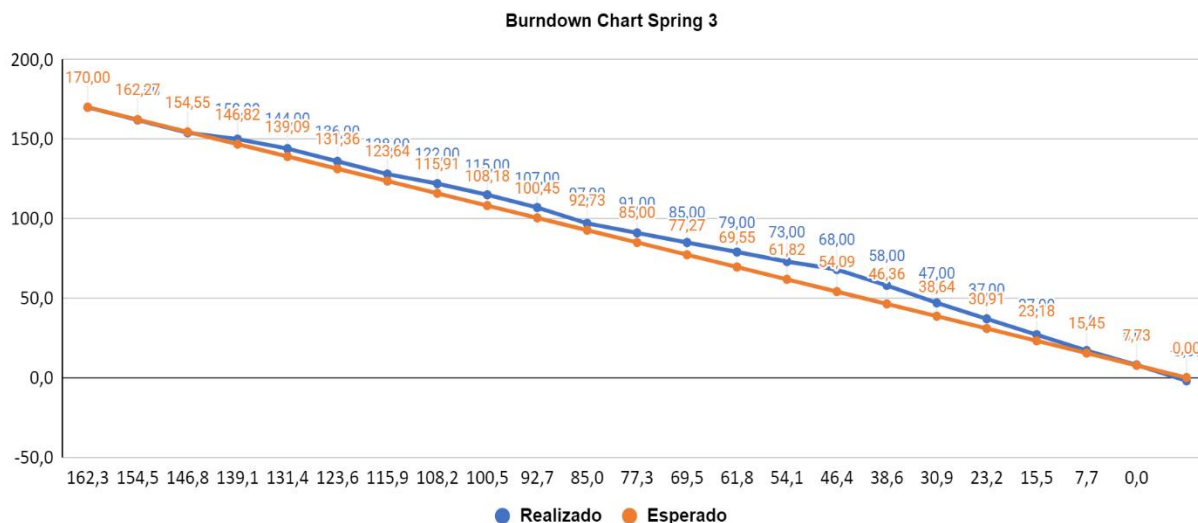


Fig. 14. Burndown Chart del Sprint 3

La **Fig. 14**, muestra que, los primeros 3 días, el equipo progresa a un ritmo que coincide con las expectativas basadas en el plan original. Sin embargo, se percibe un atraso en el trazado del gráfico durante los 15 días siguientes, reflejando un avance más lento de lo previsto. Este retraso se atribuye principalmente a las tareas de gestión de recepciones y la generación automática de comprobantes de alojamiento, específicamente al enviar los datos a través del formato JSON y la generación de los comprobantes al traer los datos. A pesar del retraso, al final del Sprint, durante los últimos 4 días, el equipo de desarrollo consiguió ajustar su enfoque y acelerar el trabajo, culminando todas las tareas programadas con éxito.

c) Revisión del Sprint 3:

Durante la revisión del Sprint 3, se alcanzaron los objetivos previstos de generar las recepciones con su comprobante de alojamiento. En la **TABLA XII**, se presenta un resumen de las actividades realizadas y los resultados obtenidos. Donde se completaron todas las tareas propuestas, se recomendó simplificar la gestión de estados a solo dos: ocupado y disponible. En la demostración, se mostró el correcto almacenamiento de las recepciones con sus comprobantes. Se recogió el feedback sobre la importancia de utilizar mensajes de consola para identificar y solucionar errores más eficientemente. La retrospectiva destacó la eficiencia en la ejecución de tareas y la implementación de las recepciones y cómo mejorar el diseño del comprobante de alojamiento.

TABLA XII
REVISIÓN DEL SPRINT 3

SPRINT REVIEW	SPRINT 3
Ítem	Descripción
Objetivos del Sprint 3	Generar las recepciones con su comprobante de pago.
Historias Completadas	Las tareas han sido completadas, aunque se sugiere eliminar una tabla (detalle) y trabajar únicamente con dos estados: ocupado y disponible.
Demostración	Se guarda correctamente las recepciones con su debido comprobante de pago.
Feedback	Se sugiere la utilización de mensajes de consola para una mejor identificación de errores o aspectos que no funcionan correctamente, facilitando así la depuración y el mantenimiento del código.
Fecha	13/1/2024
Retrospectiva	Lo que salió bien: La eficiencia en completar todas las tareas dentro del sprint y la implementación efectiva de las recepciones con comprobantes. Que se debe mejorar: Mejorar el diseño de los comprobantes de pago.

d) Entregables del Sprint 3:

A continuación, presentamos la implementación de la gestión de la recepción con sus debidas facturas, abarcando tanto el Back-end como el Front-end, como parte del entregable del Sprint 3. En esta sección, se establece la operación de crear una recepción con su debida factura y así mismo poder visualizar el listado de las recepciones. En la **TABLA XIII**, se encuentran las APIs establecidas tanto para las recepciones como para las facturas (detalle Factura). Proporciona los métodos HTTP permitidos y los códigos de estado HTTP que indican el resultado de cada operación para las APIs relacionadas con las recepciones y comprobante de pago.

TABLA XIII
LISTA DE API'S DE LAS RECEPCIONES Y COMPROBANTE DE PAGO

Tabla	Descripción	APIs	Tipo	Resultado
Recepción	Listar recepciones	/api/HPO/recepciones/listar	GET	Status: 200 OK
Recepción	Guardar recepción	/api/HPO/recepciones/guardar	POST	Status: 200 OK
Habitación	Cambiar estado de habitación	/api/HPO/habitaciones/{id}/actualizarEstado	PATCH	Status: 200 OK
Recepción	Listar ubicación (Primer Piso)	/api/HPO/recepciones/primerpiso	GET	Status: 200 OK
Recepción	Listar ubicación (Segundo Piso)	/api/HPO/recepciones/segundopiso	GET	Status: 200 OK
Factura	Crear Factura	/api/HPO/facturas/guardar	POST	Status: 200 OK
Factura	Listar facturas	/api/HPO/facturas/listar	GET	Status: 200 OK
Factura	Descargar PDF por ID	/api/HPO/facturas/download-pdf/{id}	GET	Status: 200 OK
Cliente	Buscar el cliente (cédula)	/api/HPO/clientes/buscarPorCedula	GET	Status: 200 OK

La gestión de las recepciones se implementa tanto en el Back-end (**Fig. 15**, **Fig. 16**, **Fig. 17**) como en el Front-end (**Fig. 18**, **Fig. 19**), como parte de los entregables del Sprint 3. Esto abarca las funcionalidades necesarias para el manejo de recepciones.

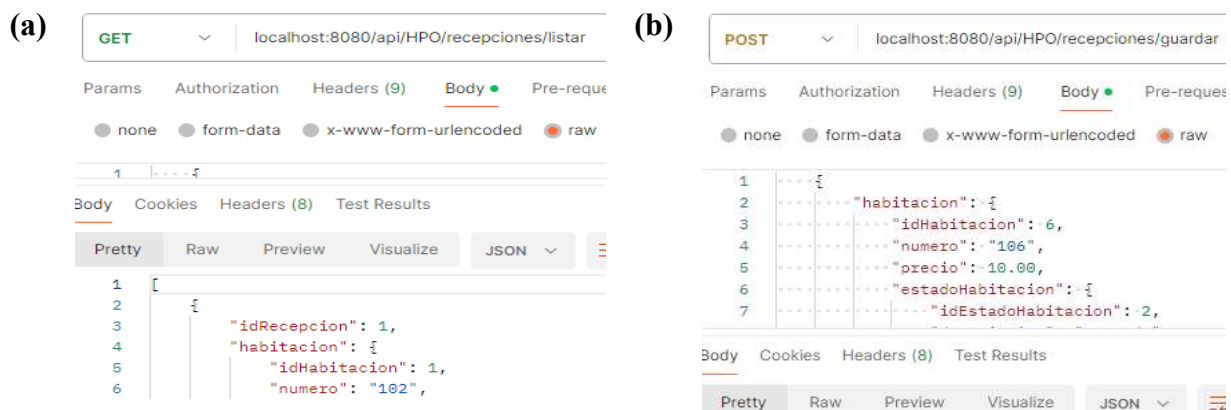


Fig. 15. Listar y guardar recepciones en el Back-end. (a) Listar recepciones. (b) Guardar recepción

En la **Fig. 15**, se muestra el proceso para ejecutar dos operaciones a través de Postman. (Fig. 15a) Método GET, para listar todas las recepciones presentes en la base de datos, la cual está alojada en SQL Server. (Fig. 15b) Método POST, empleado para guardar una nueva recepción.

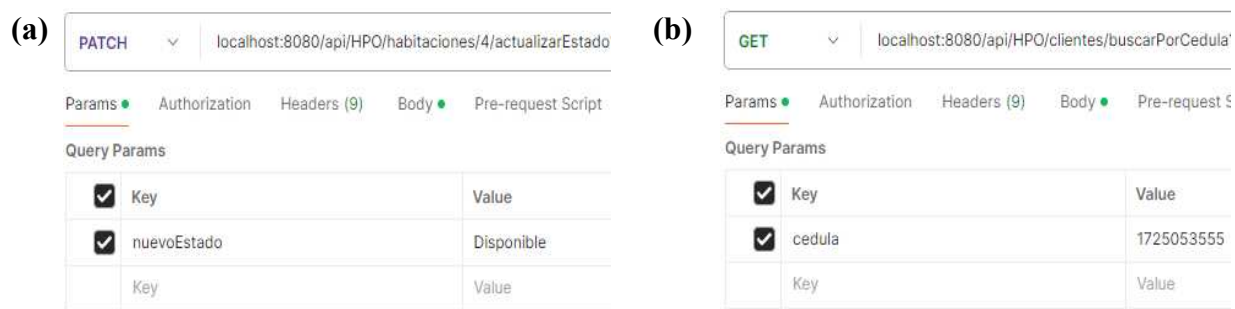


Fig. 16. Cambiar estado de una habitación y buscar cliente por cédula en el Back-end. (a) Cambiar estado de habitación. (a) Buscar cliente por cédula

En la **Fig. 16** se ilustra cómo realizar dos operaciones específicas mediante Postman. (Fig. 16a) Método PATCH, para cambiar el estado a una habitación. (Fig. 16b) Método GET, para buscar un cliente por medio de su cédula.

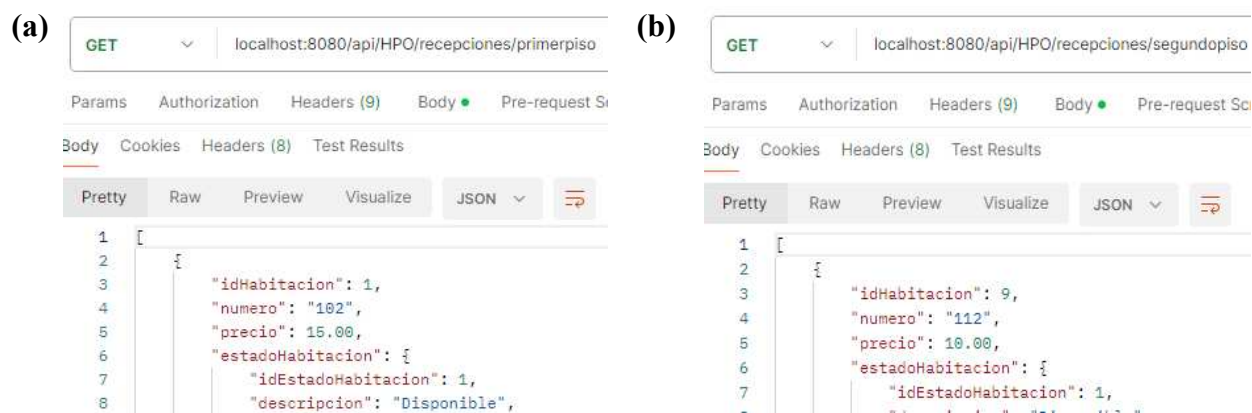


Fig. 17. Listar habitaciones por primer piso y segundo piso

En la **Fig. 17** se ilustra cómo realizar dos operaciones específicas mediante Postman. (Fig. 17a) Método GET, para listar las habitaciones que se encuentran en el primer piso. (Fig. 17b) Método GET, para listar las habitaciones que se encuentran en el segundo piso.

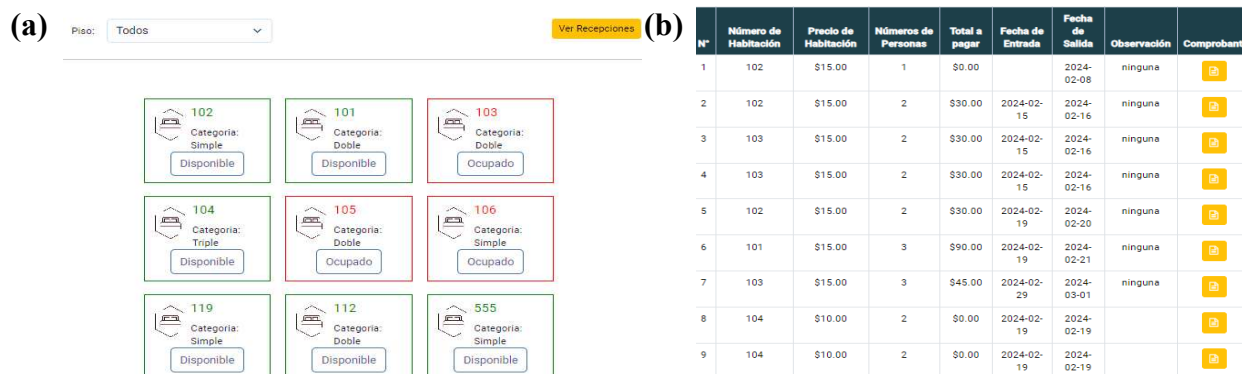


Fig. 18 Interfaces de la gestión de recepciones en el Front-End (1). (a) Visualizar habitaciones disponibles u ocupadas. (b) Visualizar las recepciones realizadas

La **Fig. 18** ilustra la interfaz dedicada a la administración de recepciones en el Hotel Puerta de Oro, enfocándose en dos aspectos principales. (Fig. 18a) Muestran las habitaciones del Hotel Puerta de Oro mediante rectángulos, destacando que, dependiendo del estado de la habitación (Disponible u Ocupado), se asigna un color correspondiente (verde para disponible, rojo para ocupado). Cabe resaltar que al hacer clic en el botón de una habitación marcada como ‘Ocupado’, aparece un cuadro de confirmación para cambiar su estado a ‘Disponible’. Por otro lado, al hacer clic en el botón de una habitación ‘Disponible’, se despliega un modal dividido en tres secciones: la primera con los datos de la habitación, la segunda para buscar un cliente por medio de la cédula, y la última para ingresar los datos de la recepción. Además, en la parte superior derecha, se

encuentra un botón ‘Ver Recepciones’ que muestra una lista de recepciones realizadas, incluyendo un botón para visualizar el comprobante de alojamiento. En la parte superior izquierda, hay una lista que permite visualizar las habitaciones según su ubicación (Primer Piso, Segundo Piso). (Fig. 18b) Modal que contiene la lista de las recepciones realizadas, indicando información relevante de la misma con su respectivo comprobante de pago.

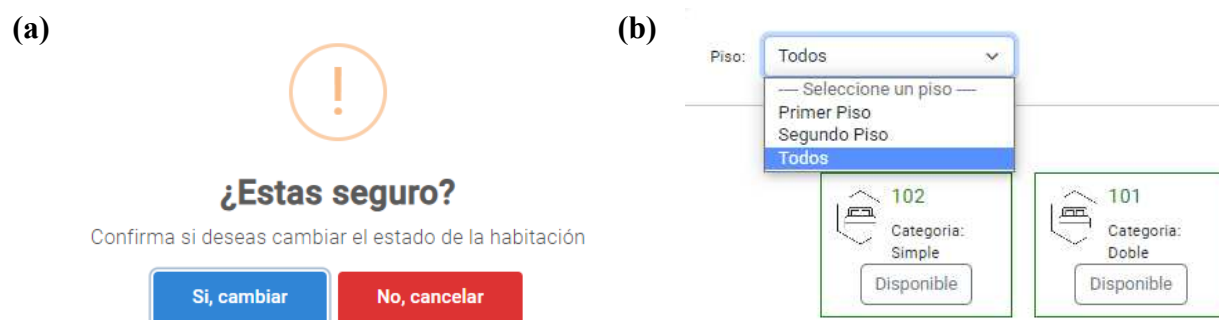


Fig. 19. Interfaces de la gestión de recepciones en el Front-End (2). (a) Cambiar estado de una habitación. (b) Filtrar habitaciones por el piso

La **Fig. 19** muestra la interfaz destinada a la gestión de recepciones, destacando especialmente dos funcionalidades. (Fig. 19a) Cuadro de confirmación para cambiar el estado de la habitación, ofreciendo dos opciones: ‘Sí, cambiar’ y ‘No, cancelar’. (Fig. 19b) Filtro para listar las habitaciones dependiendo de su ubicación.

En la **Fig. 20**, se muestra el modal que contiene un formulario con tres secciones: La primera sección muestra la información de las habitaciones, cuando se ingresa automáticamente al dar clic en el botón de “Disponibile”. La segunda sección se utiliza para ingresar los datos del cliente con el fin de generar el comprobante de pago de la habitación. Si se trata de un cliente recurrente, se puede buscar mediante su cédula y los campos están desactivados. En cambio, si es un nuevo cliente, se ingresan los datos y este se guarda en la lista de clientes. En la tercera sección se tienen los datos específicos para guardar la recepción, teniendo en cuenta que todos los campos están validados.

Detalle de la Recepciones X

Habitación

Número de Habitación 102	Precio 15
Categoría de la Habitación Simple	Ubicación de la Habitación Primer Piso

Detalle de la Habitación
Aire acondicionado, 2 camas de 2p1/2.

Ciente

Ingrese la cédula del cliente Buscar

Nombres Ingrese los nombres del cliente	Apellido Ingrese los apellidos del cliente
Cédula Ingrese la cédula del cliente	Teléfono Ingrese el teléfono del cliente
Correo Ingrese el correo del cliente	Dirección Ingrese la dirección del cliente

Recepción

Fecha de Entrada 06/03/2024	Fecha de Salida dd/mm/aaaa
Número de Personas <input type="text"/>	Alguna observación <input type="text"/>

Cerrar
Guardar

Fig. 20. Formulario para guardar una recepción

La implementación del comprobante de pago en el Back-end se ilustra en la **Fig. 21**, y en el Front-end, en la **Fig. 22**, como parte de los entregables del Sprint 3.

a)

GET localhost:8080/api/HPO/facturas/listar

Params Authorization Headers (9) Body ● Pre-request Scrip

body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON ⌵

```

1 {
2   "numeroFactura": 1,
3   "recepcion": {
4     "idRecepcion": 1,
5     "habitacion": {
6       "idHabitacion": 1,
7       "numero": "102",
8       "precio": 15.00,
9       "estadoHabitacion": {
10        "idEstadoHabitacion": 1,
11        "descripcion": "Disponibile",

```

b)

POST localhost:8080/api/HPO/facturas/guardar

Params Authorization Headers (9) Body ● Pre-req

none form-data x-www-form-urlencoded raw

```

1 {
2   "recepcion": {
3     "idRecepcion": 1,
4     "habitacion": {
5       "idHabitacion": 1,
6       "numero": "102",

```

body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON ⌵

```

1 {
2   "numeroFactura": 2008,
3   "recepcion": {

```

Fig. 21. Gestión de los comprobantes de pago en el Back-end. (a) Listar comprobantes de pago. (b) Guarda comprobante de pago

En la **Fig. 21**, presenta la interfaz para gestionar el comprobante de pago al momento de realizar una recepción, las cuales fueron probadas mediante Postman. (Fig. 21a) Método GET, permite listar todos los comprobantes de pagos presentes en la base de datos, la cual está alojada en SQL Server. (Fig. 21b) Método POST, para guardar el comprobante de pago al realizar una recepción.



Fig. 22. Interfaz del comprobante de pago en el Front-end

En la **Fig. 22**, muestra un mensaje de confirmación con el comprobante de pago generado de la recepción realizada, además cuenta con dos opciones: 'Descargar' y 'Cerrar'.

Completando el tercer Sprint, hemos conseguido los objetivos marcados para esta fase del SCRUM. Los detalles de este proceso están contenidos en el documento adjunto, "*Sprint [3]*", el cual describe cada etapa, desde la planificación hasta la implementación, culminando exitosamente con su finalización.

7) *Sprint 4:*

En el Sprint 4, dirigimos nuestro enfoque hacia la generación de reportes en formato PDF para análisis detallado de las reservas, la implementación de un dashboard con estadísticas clave para una evaluación rápida del estado del hotel, y la mejora de una interfaz de inicio de sesión robusta y personalizable. A continuación, se detallan los aspectos clave del Sprint 4:

- **Objetivos:** Se centra en tres objetivos esenciales: la gestión de reportes en formato PDF, la implementación de un dashboard con estadísticas clave y crear la interfaz de inicio de sesión.

- Duración del Sprint 4: 3 semanas (14/01/2024 - 04/02/2024).
- Responsable: Karla Vera.

a) *Planificación del Sprint 4:*

En esta sección, cada tarea se ha desglosado en subtareas para abordar de manera efectiva las complejidades y asegurar un desarrollo coherente y de alta calidad. En la **TABLA XIV** se muestra el análisis detallado de las subtareas para las tareas PB09, PB03 y PB08, presentando el tiempo estimado en días para cada una, así como el tiempo total estimado para completar el Sprint 4. Aquí se detallan las tareas específicas planificadas para el Sprint 4, centradas en la generación de reportes, el dashboard con estadísticas clave y la interfaz de inicio de sesión. Cada tarea contribuirá al cumplimiento de los objetivos establecidos.

TABLA XIV
PLANIFICACIÓN DEL SPRINT 4

SPRINT PLANNING		SPRINT 4 (3 SEMANAS)	
ID	Ítem	Requerimiento/Tarea	Tiempo Estimado (Días)
PB09		Generar reportes de las reservas en PDF	
	PB09.1	Diseñar una interfaz para los reportes en formato PDF	
	PB09.2	Realizar filtrado de fechas para listar las recepciones, durante ese periodo.	
	PB09.3	Obtener la suma total obtenida de las recepciones realizadas en esas fechas	
	PB09.4	Implementar un proceso automatizado que genere los reportes en formato PDF de manera eficiente	9
	PB09.5	Proporcionar una función que permita a los usuarios descargar los reportes en PDF y almacenarlos localmente	
	PB09.6	Realizar pruebas	
	PB09.7	Documentar	
PB03		Dashboard con estadísticas clave	
	PB03.1	Desarrollar un diseño atractivo y funcional para el dashboard que destaque las estadísticas clave de manera clara	
	PB03.2	Implementar el logotipo distintivo del hotel, brindando una identidad visual única y reconocible del sujeto de prueba.	7
	PB03.3	Implementar un sistema que recopile y actualice en tiempo real las estadísticas de total de habitaciones, habitaciones ocupadas, habitaciones disponibles y habitaciones en limpieza	
	PB03.5	Realizar pruebas	
	PB03.6	Documentar	

TABLA XV
(CONTINUACIÓN) PLANIFICACIÓN DEL SPRINT 4

PB08	Generar la interfaz de inicio de sesión	6
PB08.1	Desarrollar un diseño atractivo y funcional para la interfaz de inicio de sesión que sea intuitiva para los usuarios	
PB08.2	Incluir campos de entrada de datos para el nombre de usuario y la contraseña, asegurando una fácil comprensión y usabilidad.	
PB08.3	Implementar mecanismos de validación para garantizar la introducción correcta de la información requerida.	
PB08.4	Incluir mensajes al momento que las credenciales sean incorrecta y correctas.	
PB08.5	Incorporar un icono o interruptor que permita a los usuarios visualizar la contraseña que están escribiendo.	
PB08.6	Garantizar que la interfaz sea responsive.	
PB08.7	Realizar pruebas	
PB08.8	Documentar	
Duración del sprint (días)		22

b) Burndown Chart del Sprint 4:

El Burndown Chart del Sprint 4, expresado gráficamente, refleja el progreso del equipo en comparación con las tareas planificadas a lo largo de este Sprint. En la **Fig. 23**, proporciona una visualización clara de la velocidad del equipo en la ejecución de las subtareas asignadas a lo largo del tiempo. La línea naranja traza las horas planificadas, mientras que la línea azul ilustra las horas reales completadas cada día, brindando así una evaluación detallada del avance diario del equipo.

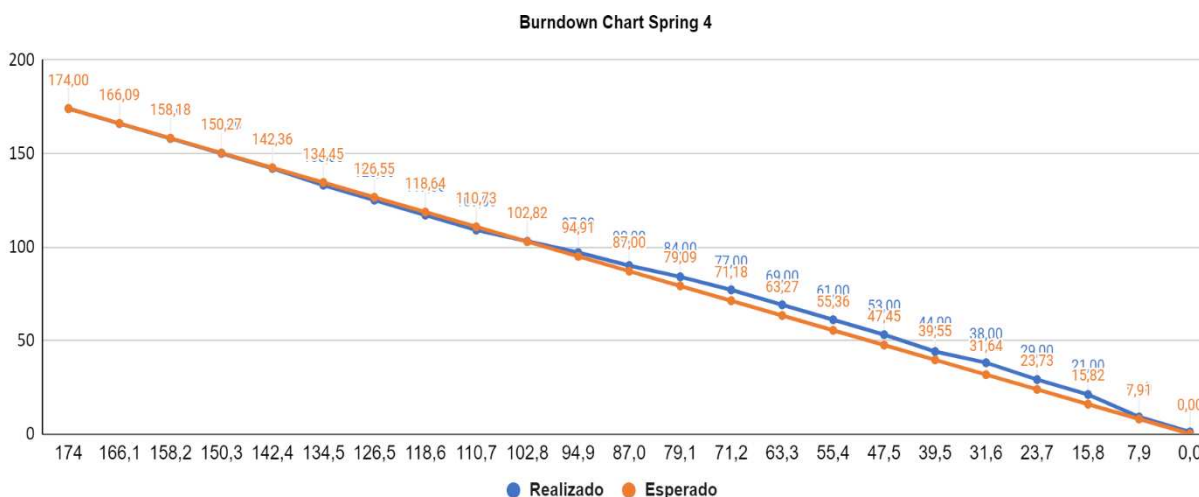


Fig. 23. Burndown Chart del Sprint 4

En la **Fig. 23** se aprecia que durante los primeros 10 días, el equipo de desarrollo progresa a un ritmo que coincide con las expectativas basadas en el plan original. Sin embargo, se percibe un retraso en el trazado del gráfico durante los 10 días siguientes, reflejando un avance más lento de lo previsto, especialmente en traer lo que respecta a las estadísticas clave y al manejo del inicio de sesión. A pesar del retraso, al final del Sprint, durante los últimos 2 días, el equipo de desarrollo consiguió ajustar su enfoque y acelerar el trabajo, culminando todas las tareas programadas con éxito.

c) Revisión del Sprint 4:

La revisión del Sprint 4 consistió en una evaluación reflexiva tanto de los procesos como de los resultados obtenidos. El Sprint 4 se enfocó en la generación de reportes, el dashboard con estadísticas clave y la interfaz de inicio de sesión, completando satisfactoriamente todas las historias planificadas. Durante la demostración, se presentó con éxito el ingreso al sistema mediante el usuario y contraseña. Se visualizó el total de habitaciones, el total de habitaciones ocupadas y las disponibles, además se demostró la descarga de los reportes en formato PDF. La **TABLA XVI** destaca el objetivo general del Sprint 4, incluyendo las historias completadas. Aunque la demostración fue exitosa, el feedback recibido indicó que se debe mejorar, especialmente en controlar el acceso al sistema de manera segura.

TABLA XVI
REVISIÓN DEL SPRINT 4

SPRINT REVIEW	SPRINT 4
ítem	Descripción
Objetivos del Sprint 4	Generar de reportes en formato PDF, implementar un dashboard con estadísticas claves y crear un inicio de sesión.
Historias Completadas	Todas
Demostración	Las recepciones se filtran correctamente, y los reportes se descargan en formato PDF. Además, se visualizan las estadísticas claves y se prueba con éxito el inicio de sesión.
Feedback	Mantener la coherencia con el patrón de colores del sistema.
Fecha	4/2/2024
Retrospectiva	Lo que salió bien: Se accede correctamente al sistema web, se visualizan adecuadamente las estadísticas clave y se generan los reportes en formato PDF. Que se debe mejorar: Controlar el acceso al sistema

d) Entregables del Sprint 4:

A continuación, presentamos la implementación de los reportes en formato PDF, las estadísticas clave (total de habitaciones, habitaciones disponibles, habitaciones ocupadas), inicio de sesión, abarcando tanto el Back-end como el front-end, como parte del entregable del Sprint 4. En la **TABLA XVII**, se encuentran las APIs establecidas para los reportes en PDF, las estadísticas clave y para el inicio de sesión. Esta tabla proporciona los métodos HTTP permitidos y los códigos de estado HTTP que indican el resultado de cada operación para las APIs relacionadas con los reportes en PDF, las estadísticas clave y para el inicio de sesión en el sistema.

TABLA XVII
LISTA DE API'S DE INICIO DE SESIÓN, ESTADÍSTICAS Y REPORTE

Tabla	Descripción	APIs	Tipo	Resultado
Habitación	Listar el total de habitaciones	/api/HPO/clientes/listar/totalHabitaciones	GET	Status: 200 OK
Habitación	Listar habitaciones ocupadas	/api/HPO/clientes/listar/habitacionesOcupadas	GET	Status: 200 OK
Habitación	Listar habitaciones disponibles	/api/HPO/clientes/listar/habitacionesDisponibles	GET	Status: 200 OK
Usuario	Ingresar al sistema web	api/HPO/usuarios/login	POST	Status: 200 OK
Usuario	Listar usuario	api/HPO/usuarios/listar	GET	Status: 200 OK
Usuario	Editar usuario	api/HPO/usuarios/actualizar	PUT	Status: 200 OK
Recepción	Descargar el reporte PDF	/api/HPO/recepciones/download-pdf	GET	Status: 200 OK
Recepción	Reporte clientes en un EXCEL	/api/HPO/recepciones/download	GET	Status: 200 OK
Reportes	Listar los reportes	api/HPO/reportes/listar	GET	Status: 200 OK
Reportes	Guardar el reporte	api/HPO/reportes/guardar	POST	Status: 200 OK

La implementación de estadísticas clave (total de habitaciones, habitaciones disponibles, habitaciones ocupadas) se realiza en el Back-end, como se muestra en la **Fig. 24**, y en el Front-end, en la **Fig. 25**, formando parte de los entregables del Sprint 4.

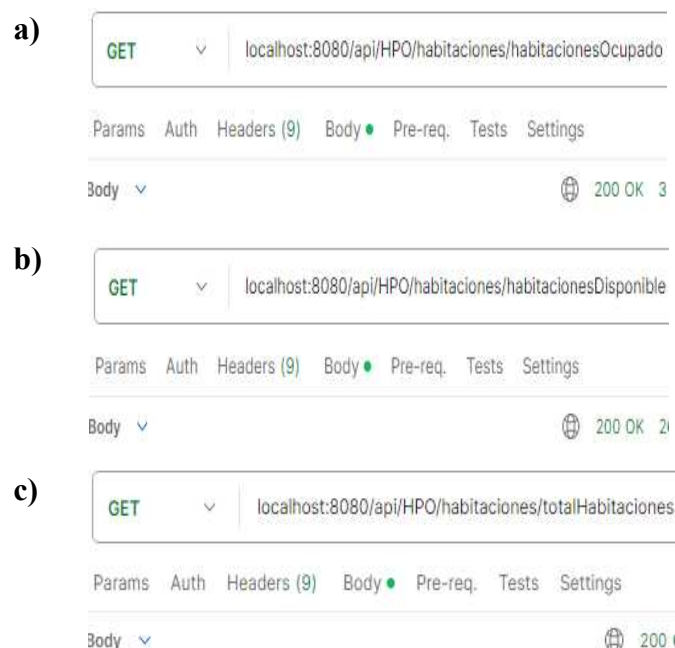


Fig. 24. Gestión de las estadísticas claves en el Back-end. (a) Total de habitaciones ocupadas. (b) Total de habitaciones disponibles. (c) Total de habitaciones

En la **Fig. 24**, se presenta la interfaz para mostrar las estadísticas clave sobre las habitaciones, las cuales fueron probadas mediante Postman. (Fig. 24a) Método GET, para obtener la cantidad de habitaciones ocupadas. (Fig. 24b) Método GET, para obtener la cantidad de habitaciones disponibles. (Fig. 24c) Método GET, para obtener la cantidad total de habitaciones.

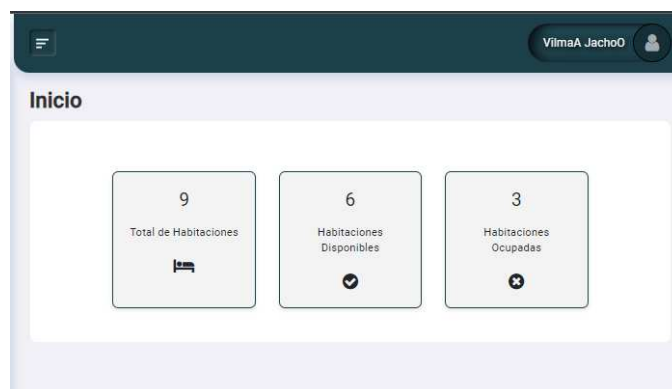


Fig. 25. Interfaz de las estadísticas claves en el Front-end

La **Fig. 25**, muestra la interfaz de las estadísticas clave mediante tres rectángulos, indicando el total de habitaciones que hay en el hotel, el total de habitaciones disponibles y el total de habitaciones ocupadas.

La implementación del inicio de sesión se detalla en el Back-end, como se muestra en la **Fig. 26**, y en el Front-end (**Fig. 27**, **Fig. 28**), siendo parte de los entregables del Sprint 4.

```

1 [
2   {
3     "idUsuario": 1,
4     "nombre": "Vilma",
5     "apellido": "Jacho",
6     "cedula": "1315754240",
7     "telefono": "0999999999",
8     "clave": "12345",
9     "token": null,
10    "username": "vilmaj",
11    "estado": true,
12    "rol": {
13      "idRol": 1,
14      "nombreRol": "admin"
15    }
16  }
17 ]

```

Fig. 26. Listar usuario en el Back-end

La **Fig. 26** se muestra el uso del método GET, el cual se emplea para listar los datos del usuario almacenado en la base de datos alojada en SQL Server. Esta información es utilizada posteriormente para autenticar y permitir el acceso del usuario al sistema.

(a) Ingrese sus credenciales

Usuario

Contraseña

Ingresar

(b) VilmaA JachoO
admin

Nombres: VilmaA

Apellidos: JachoO

Cédula: 1315754240

Teléfono: 0999999999

Username: vilmaj

Contraseña:

Editar

Fig. 27. Interfaces de inicio de sesión en el Front-end (1). (a) Inicio de sesión. (b) Información de usuario

En la **Fig. 27**, muestra la interfaz para gestionar el inicio de sesión. (Fig. 27a) Muestra la interfaz de inicio de sesión, donde se solicita al usuario ingresar sus credenciales (nombre de usuario y contraseña) y es importante destacar que ambos datos son obligatorios. (Fig. 27b) Información del usuario que está ingresando al sistema, con la opción de poder editar su información.

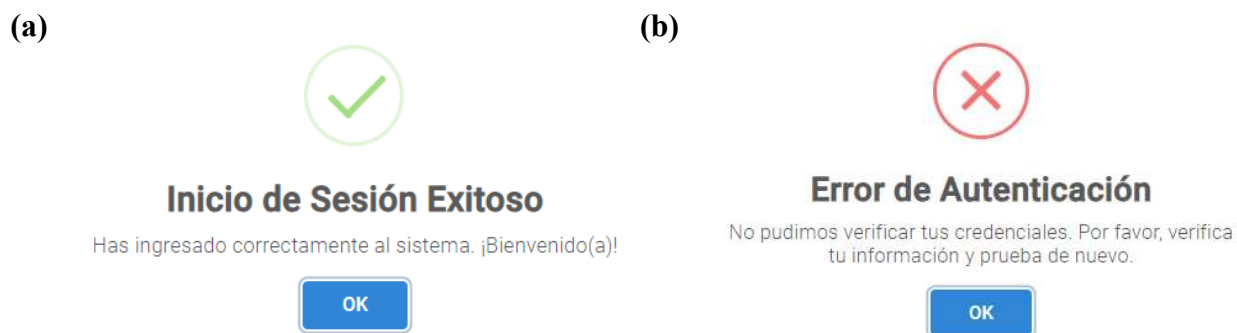


Fig. 28. Interfaces de inicio de sesión en el Front-end (2). (a) Inicio de Sesión Exitoso. (b) Error de Autenticación

En la **Fig. 28**, ilustra los mensajes de ingreso proporcionados durante el proceso de inicio de sesión. (Fig. 28a) Si el usuario ingresa las credenciales correctamente, aparecerá un mensaje de confirmación, permitiendo el acceso al sistema. (Fig. 28b) En caso de que el usuario introduzca de manera incorrecta sus credenciales, se mostrará un mensaje de error.

La implementación de los reportes en PDF se realiza en el Back-end, como se ilustra en la **Fig. 29**, y en el Front-end, en la **Fig. 30**, formando parte de los entregables del Sprint 4.



Fig. 29. Gestión de los reportes en el Back-end. (a) Guardar reportes en PDF. (b) Listar los reportes

En la **Fig. 29**, se presentan las operaciones para gestionar los reportes en formato PDF, las cuales fueron probadas mediante Postman. (Fig. 29a) Método POST, empleado para guardar un nuevo reporte. (Fig. 29b) Método GET, permite listar los reportes generados en la base de datos.

Desde: 19/02/2024 Hasta: 22/02/2024 Generar y Descargar reporte

Recepciones					
N°	Habitación	Cliente	Fecha Entrada	Fecha Salida	Total
1	102	Karla Cecilia	2024-02-19	2024-02-20	\$30.00
2	101	Karla Cecilia	2024-02-19	2024-02-21	\$90.00
3	104	Karla Cecilia	2024-02-19	2024-02-19	\$0.00
4	104	Karla Cecilia	2024-02-19	2024-02-19	\$0.00
5	105	Karla Cecilia	2024-02-19	2024-02-20	\$10.00

Suma Total
\$130.00

Fig. 30. Interfaz de los reportes en el Front-end

En la **Fig. 30**, se muestra en la parte superior izquierda un rango de fechas que permite seleccionar el período deseado para observar las recepciones en los informes. Al hacer clic en el botón ubicado en la parte superior derecha, se generará y descargará el reporte automáticamente. En la parte central, se listan las recepciones realizadas durante el período de tiempo establecido. Por último, se presenta la suma total de todas las recepciones.

Completando el cuarto Sprint hemos conseguido los objetivos marcados para esta fase del SCRUM. Los detalles de este proceso están contenidos en el documento adjunto, "*Sprint [4]*", el cual describe cada etapa, desde la planificación hasta la implementación, culminando exitosamente con su finalización.

Una vez que se cumplió con lo planificado con SCRUM en las fechas establecidas, se generó el respectivo manual de usuario con la finalidad de enseñar el funcionamiento del SCRUM. En el documento adjunto "*MANUAL DE USUARIO SCRUM*", se explica detalladamente cada una de las funcionalidades del SCRUM. En las siguientes figuras (**Fig. 31**, **Fig. 32**, **Fig. 33**, **Fig. 34**, **Fig. 35**) se puede observar una vista previa de las interfaces que contiene el sistema.

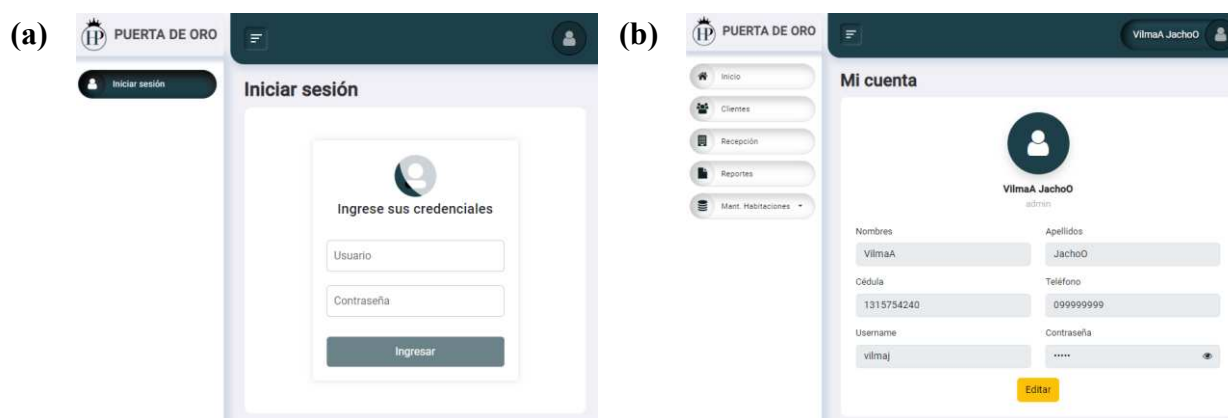


Fig. 31. Vista previa de las interfaces del sistema (1). (a) Interfaz de inicio de sesión. (b) Interfaz de la información del usuario

En la **Fig. 31**, se presentan dos interfaces cruciales del sistema de SCRUM. (Fig. 31a) La interfaz de inicio de sesión, donde se requiere que el usuario introduzca sus credenciales para acceder al sistema. (Fig. 31b) La interfaz para visualizar la información del usuario, proporcionando detalles relevantes y permitiendo editar el perfil.

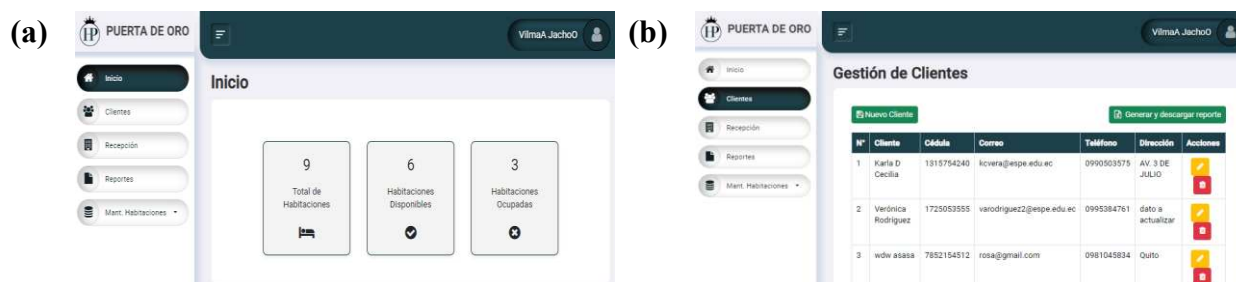


Fig. 32. Vista previa de las interfaces del sistema (2). (a) Interfaz del inicio. (b) Interfaz de la gestión de clientes

En la **Fig. 32**, se presentan dos interfaces cruciales del sistema de SCRUM. (Fig. 32a) La interfaz de estadísticas clave sobre las habitaciones, ofreciendo datos importantes como el total de habitaciones, habitaciones disponibles y ocupadas. (Fig. 32b) La interfaz para la gestión de clientes, facilitando operaciones como el registro, actualización, eliminación y visualización de información de los clientes.

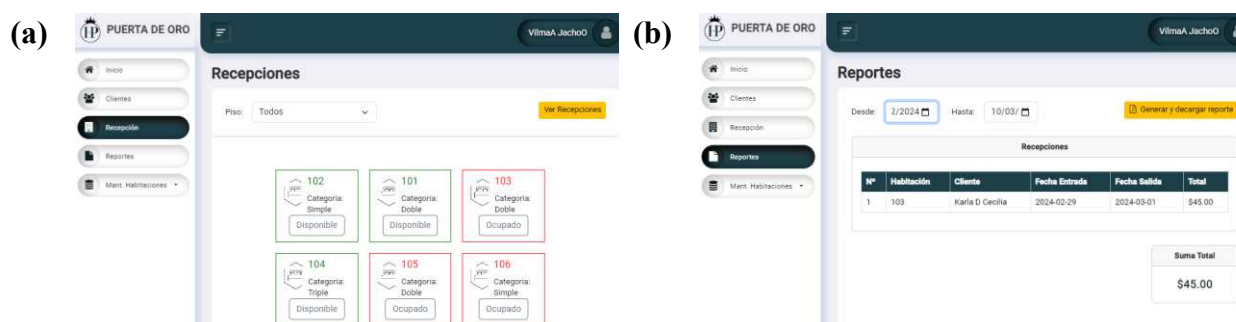


Fig. 33. Vista previa de las interfaces del sistema (3). (a) Interfaz de la gestión de recepciones. (b) Interfaz de la gestión de reportes

En la **Fig. 33**, se presentan dos interfaces cruciales del sistema de SCRUM. (Fig. 33a) La interfaz para la gestión de las recepciones permitiendo realizar operaciones como el registro y la visualización de recepciones. (Fig. 33b) La interfaz para la gestión de reportes, facilitando la creación y visualización de reportes en base a un rango de fechas.

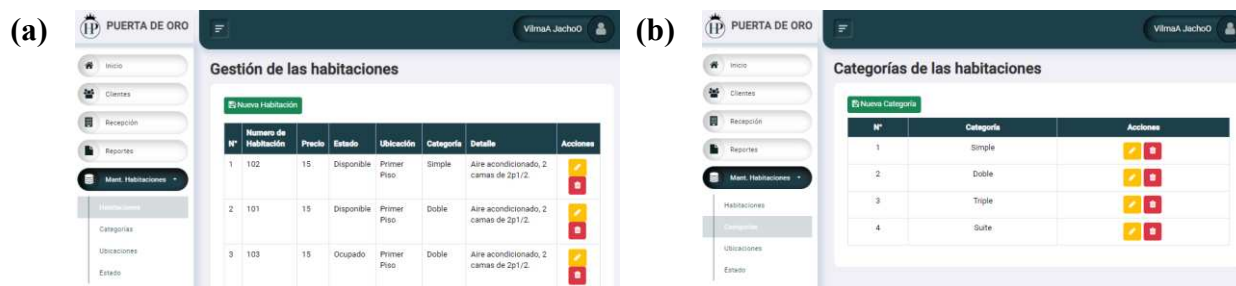


Fig. 34. Vista previa de las interfaces del sistema (4). (a) Interfaz de la gestión de habitaciones. (b) Interfaz de la gestión de categorías

En la Fig. 34, se presentan dos interfaces cruciales del sistema de SCRUM. (Fig. 34a) La interfaz para la gestión de las habitaciones, facilitando operaciones como el registro, actualización, eliminación y visualización de información de las habitaciones. (Fig. 34b) La interfaz para la gestión de categorías de las habitaciones, facilitando operaciones como el registro, actualización, eliminación y visualización de la información.

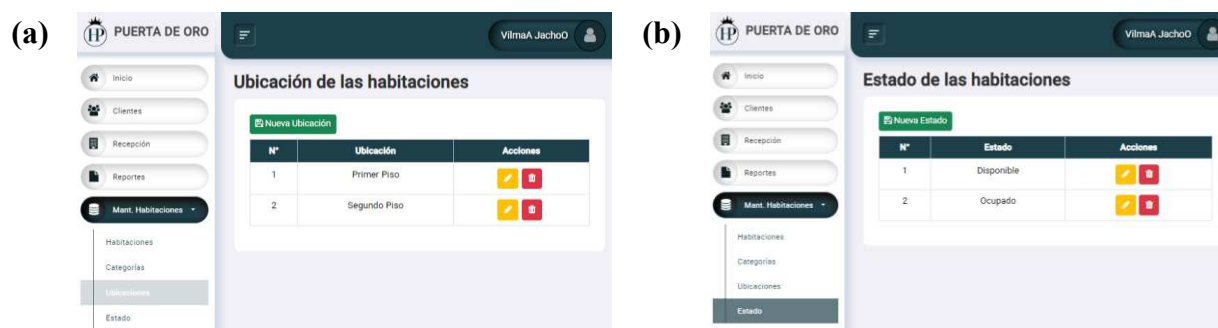


Fig. 35. Vista previa de las interfaces del sistema (5). (a) Interfaz de la gestión de categorías. (b) Interfaz de la gestión de estado

En la Fig. 35, se presentan dos interfaces cruciales del sistema de SCRUM. (Fig. 35a) La interfaz para la gestión de ubicación de las habitaciones, facilitando operaciones como el registro, actualización, eliminación y visualización de la información. (Fig. 35b) La interfaz para la gestión de estado de las habitaciones, facilitando operaciones como el registro, actualización, eliminación y visualización de la información.

C. Implementación de la norma ISO 29110 parte 5-1-2

A continuación, para alcanzar los objetivos establecidos para el desarrollo, se optó por implementar la guía ISO 29110 parte 5-1-2, focalizándose especialmente en el proceso de Implementación del Software (SI). Para llevar a cabo este estudio, se empleó el lenguaje PHP en conjunto con el marco ágil Laravel, y se utilizó MySQL como gestor de base de datos,

conectándose a través de la herramienta Heidi SQL. Esta elección metodológica proporcionó un marco estructurado y eficaz para el desarrollo del experimento, asegurando una implementación coherente y robusta.

1) *Arquitectura MVC:*

El patrón de arquitectura Modelo-Vista-Controlador (MVC) se representa gráficamente como una metodología que fragmenta la lógica de la aplicación y la interfaz de usuario en tres componentes esenciales: el Modelo, la Vista y el Controlador.

En la **Fig. 36**, el modelo se visualiza como la entidad encargada de gestionar datos y comportamientos fundamentales, pudiendo ser una base de datos o cualquier estructura de almacenamiento de datos. La vista, por su parte, se presenta como el componente que proporciona la interfaz de usuario, representando de manera efectiva los datos del modelo. Mientras tanto, el controlador aparece como el elemento que recibe la información del usuario y realiza las llamadas necesarias al modelo y la vista para ejecutar acciones específicas. En conjunto, este modelo gráfico ilustra cómo la arquitectura MVC facilita la separación de preocupaciones y la colaboración eficiente entre los elementos significativos para que el sistema funcione eficazmente.

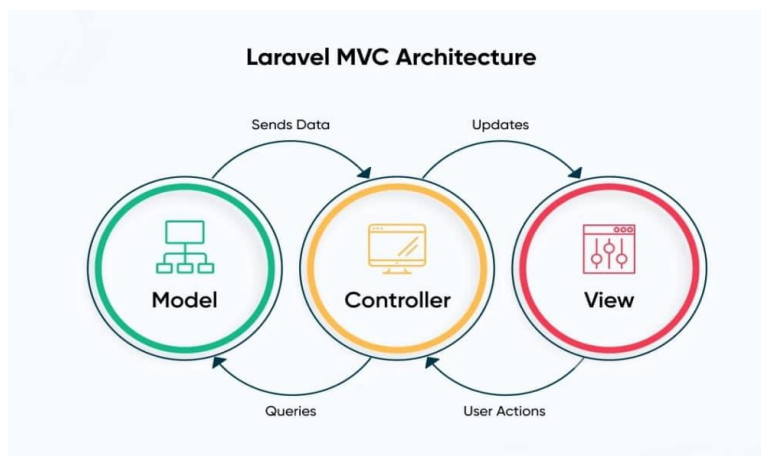


Fig. 36.Arquitectura MVC [17]

2) *Roles del SI:*

Con el fin de gestionar de manera eficaz cada proceso de SI se definieron roles para que los responsables contribuyan a lograr los objetivos del sistema. La **TABLA XVIII** describe las responsabilidades de cada rol, así como las abreviaturas que se utilizarán en las distintas etapas del desarrollo. Los roles establecidos comprenden: cliente, analista, diseñador, programador, gerente del sistema, líder técnico y el grupo de trabajo. Los roles mencionados serán asignados a la Sra.

Vilma Jacho, en calidad de gerente del Hotel Puerta de Oro, a la Srta. Verónica Rodríguez como responsable principal del desarrollo del sistema, y al Ing. Luis Castillo, quien actuará como tutor y líder del sistema.

TABLA XVIII
ASIGNACIÓN DE ROLES DE PROCESOS SI

Rol	Abreviatura	Responsable
Cliente	CUS	Vilma Jacho (Hotel Puerta de Oro)
Analista	AN	Verónica Rodríguez
Diseñador	DES	Verónica Rodríguez
Programador	PR	Verónica Rodríguez
Gerente de Proyecto	PM	Luis Castillo
Líder Técnico	TL	Luis Castillo
Grupo de trabajo	WT	Verónica Rodríguez

3) *Evaluación de calidad del SI:*

Posteriormente, se presenta la escala de evaluación que se utilizará para verificar el progreso de los productos de salida del SI. Esta medición considera el producto final de cada tarea realizada en la semana de trabajo correspondiente. El funcionamiento de la escala se basa en el porcentaje de evolución de cada tarea, clasificando el valor en rangos del 1 al 5 describiéndolos como Muy bajo, Bajo, Medio, Alto y Muy alto, respectivamente. Como se puede visualizar en la **TABLA XIX**, el porcentaje se distribuye del 0 al 100 en intervalos de 20, facilitando la comprensión y evaluación del progreso de manera detallada. Este sistema proporciona una herramienta efectiva para medir y comunicar el nivel de desarrollo alcanzado en cada tarea del sistema.

TABLA XIX
ESCALA DE EVALUACIÓN DE PROCESOS SI

Porcentaje	Valor	Descripción
0-20	1	Muy Bajo
21-40	2	Bajo
41-60	3	Medio
61-80	4	Alto
81-100	5	Muy Alto

4) *Objetivos del SI:*

A continuación, se realizaron las actividades del proceso de desarrollo del Software, como se muestra en la **TABLA XX**. Para la actividad de la Iniciación de Implementación de Software (SI-001), se realizó una exhaustiva revisión del Plan de Proyecto, seguida de la configuración del entorno de implementación. Durante el Análisis de Requisitos del Software (SI-002), se asignaron tareas, se documentaron y aprobaron especificaciones, y se preparó una versión preliminar de la Documentación de Usuario.

En la fase de Diseño Arquitectónico y Detallado del Software (SI-003), se implicó la comprensión y aprobación del diseño, la definición de casos de prueba y la actualización del registro de trazabilidad. En la Construcción de Software (SI-004), se implementó el software, se diseñaron casos de prueba unitarios, se corrigieron errores y se actualizó el registro de trazabilidad. La Integración de Software y Pruebas (SI-005) abarcó la comprensión de casos y procedimientos de prueba, la integración del software, pruebas, correcciones, la documentación y aprobación de guías. Finalmente, en la fase de Entrega del Producto (SI-006), se comprendió la configuración del software, y se documentó y aprobó la documentación de mantenimiento. Todas estas actividades se incorporaron correctamente a la línea base del proyecto, según se iban realizando y antes de la entrega según el plan establecido.

TABLA XX
TAREAS DEL SI

Objetivos del SI		
ID	Actividad	Lista de tareas
SI-001	Iniciación de implementación de software	1.1. Revisión del Plan de Proyecto vigente. 1.2. Establecer o actualizar el entorno de implementación. 2.1. Asignar tareas con base en el Plan de Proyecto vigente. 2.2. Documentar o actualizar la especificación de requisitos. 2.3. Verificar y obtener la aprobación de la especificación de requisitos. 2.4. Validar y obtener aprobación de la especificación de requisitos.
SI-002	Análisis de requisitos del software	2.5. Documentar la versión preliminar de la Documentación de Usuario del Software. 2.6. Verificar y obtener aprobación de la Documentación del Usuario del Software. 2.7. Incorporar la Especificación de Requisitos y la Documentación del Usuario del Software a la Configuración del Software en la línea base.
SI-003	Diseño arquitectónico y detallado del software	3.1. Comprender la especificación de requisitos. 3.2. Documentar o actualizar el diseño del Software.

TABLA XXI
(CONTINUACIÓN) TAREAS DEL SI

SI-003	Diseño Arquitectónico y detallado del software	3.3. Verificar y obtener la aprobación del diseño del software.
		3.4. Establecer o actualizar casos de prueba y procedimientos de requisitos de prueba.
		3.5. Verificar y obtener aprobación de los casos de prueba y procedimientos de prueba.
		3.6. Actualizar el registro de la trazabilidad.
		3.7. Incorporar el diseño y el registro de trazabilidad del software.
		4.1. Comprender el diseño de software.
		4.2. Construir o actualizar el software.
SI-004	Construcción de software	4.3. Diseñar o actualizar casos de prueba unitarios.
		4.4. Corregir los defectos encontrados.
		4.5. Actualizar el registro de trazabilidad.
		4.6. Incorporar componentes de software y registro de trazabilidad a la configuración del software.
		5.1. Comprender casos y procedimientos de prueba.
		5.2. Integrar el software y actualizar casos y procedimientos de prueba.
SI-005	Integración de software y pruebas	5.3. Realizar pruebas de software utilizando casos y procedimientos de prueba.
		5.4. Corregir los defectos encontrados y realizar pruebas.
		5.5. Actualizar el registro de trazabilidad.
		5.6. Documentar o actualizar la guía de operación del producto.
		5.7. Verificar y obtener aprobación de la guía de operación del producto.
		5.8. Documentar o actualizar la documentación de usuario de software.
		5.9. Verificar y obtener aprobación de la documentación del usuario de Software.
		5.10. Incorporar lo anterior a la configuración del software como línea base.
SI-006	Entrega del producto	6.1. Comprender la configuración del software.
		6.2. Documentar la documentación de mantenimiento.
		6.3. Verificar y obtener la aprobación de la documentación de mantenimiento.
		6.4. Incorporar la documentación de mantenimiento como línea base para la configuración del software.
		6.5. Realizar la entrega de acuerdo con las instrucciones del plan de entrega del proyecto.

En la **TABLA XXII**, se presenta el instrumento diseñado para el registro de progreso de los productos de salida entregados al finalizar las actividades correspondientes a la primera etapa, que abarca desde el 16 de noviembre hasta el 30 de noviembre de 2023. Este instrumento se utiliza

para evaluar la calidad de la implementación de cada actividad del proceso SI, abarcando todos los productos de salida especificados para el SI en la guía ISO 29110 parte 5-1-2.

TABLA XXII
PROGRESO DESDE EL 16/NOV/2023 HASTA EL 30/NOV/2023

Objetivos del SI										
ID	Actividad	Productos de Salida	Registro de progreso del 16/nov al 30/nov de 2023							
			16/nov a 23/nov			24/nov a 30/nov				
			%	Valor	Descripción	%	Valor	Descripción		
SI-001	Iniciación de implementación de software	Plan de proyecto [revisado]	70	4	Alto	100	5	Muy Alto		
			100	5	Muy Alto	100	5	Muy Alto		
SI-002	Análisis de requisitos del software	- Especificación de requisitos [validada, línea base]	50	3	Medio	100	5	Muy Alto		
			40	2	Bajo	100	5	Muy Alto		
			0	1	Muy Bajo	100	5	Muy Alto		
			0	1	Muy Bajo	100	5	Muy Alto		
			20	1	Muy Bajo	100	5	Muy Alto		
			0	1	Muy Bajo	100	5	Muy Alto		
SI-003	Diseño arquitectónico y detallado del software	- Casos de prueba y procedimientos de prueba [verificado]	0	1	Muy Bajo	100	5	Muy Alto		
			0	1	Muy Bajo	100	5	Muy Alto		
		- Registro de trazabilidad [verificado, línea base]								
		- Diseño de software [verificado, línea base]								

En la **TABLA XXII**, se pueden visualizar las tareas realizadas en la semana del 16 hasta el 30 de noviembre de 2023, en la cual se entrega el plan de proyecto para la actividad SI-001, en el cual se detallan las actividades por realizar en los tiempos determinados por el grupo de trabajo (consultar <https://cutt.ly/Ww0qUNFy>). Luego, para la actividad SI-002 se realiza la especificación de requisitos, los cuales se revisaron y aprobaron por parte del Analista y el Gerente del Hotel Puerta de Oro. Estos se encuentran en el documento adjunto "*SI-002 ANÁLISIS DE LOS REQUISITOS DEL SOFTWARE*". Por otro lado, en esta actividad también se entrega la Documentación del Usuario de Software o llamado también Manual de Usuario en el que consten las partes más relevantes del sistema y su funcionamiento, esté verificado por el Analista y se encuentra en el documento adjunto "*SI-002 MANUAL DE USUARIO*".

Finalmente, para la actividad SI-003 se entrega el Diseño de Software verificado por el Diseñador, el cual contiene una especificación de los módulos que se van a trabajar en el sistema, los recursos de la implementación y la función de la interfaz de usuario (ver documento adjunto “*SI-003 DISEÑO DEL SOFTWARE*”).

Por otro lado, en la **TABLA XXIII** se presenta el instrumento diseñado para el registro de progreso de los productos de salida entregados al finalizar las actividades correspondientes a la primera etapa, que abarca desde el 01 de diciembre hasta el 22 de diciembre de 2023. Este instrumento se utiliza para evaluar la calidad de la implementación de cada actividad del proceso SI, abarcando todos los productos de salida especificados para el SI en la guía ISO 29110 parte 5-1-2.

TABLA XXIII
PROGRESO DESDE EL 01/DIC/2023 HASTA EL 22/DIC/2023

ID	Actividad	Productos de Salida	Objetivos del SI											
			Registro de progreso del 01/dic al 22/dic de 2023											
			01/dic a 08/dic			09/dic a 15/dic			16/dic a 22/dic					
%	Valor	Descripción	%	Valor	Descripción	%	Valor	Descripción	%	Valor	Descripción			
SI-003	Diseño arquitectónico y detallado del software	- Casos de prueba y procedimientos de prueba [verificado]	100	5	Muy Alto	100	5	Muy Alto	100	5	Muy Alto	100	5	Muy Alto
		- Registro de trazabilidad [verificado, línea base]	50	3	Medio	100	5	Muy Alto	100	5	Muy Alto	100	5	Muy Alto
	Diseño de software [verificado, línea base]	- Diseño de software [verificado, línea base]	0	1	Muy Bajo	0	1	Muy Bajo	100	5	Muy Alto	100	5	Muy Alto
		- Diseño de software [verificado, línea base]	0	1	Muy Bajo	0	1	Muy Bajo	100	5	Muy Alto	100	5	Muy Alto
SI-004	Construcción de software	- Componentes de software [corregido, línea base]	0	1	Muy Bajo	0	1	Muy Bajo	100	5	Muy Alto	100	5	Muy Alto
		- Registro de trazabilidad [actualizado, línea base]	0	1	Muy Bajo	0	1	Muy Bajo	5	1	Muy Bajo	5	1	Muy Bajo

En la **TABLA XXIII**, se pueden visualizar las tareas realizadas en la semana del 01 hasta el 22 de diciembre de 2023 en las cuales se entregan, para la tarea SI-003, los casos de prueba y procedimientos de prueba verificados por el Diseñador y el Analista (ver

documento adjunto “SI-003 CASOS Y PROCEDIMIENTOS DE PRUEBA”). Luego, a partir de los casos de prueba se realizó el Registro de Trazabilidad en su primera versión, verificado por el Líder Técnico. El Registro de Trazabilidad se realizó en base a los requerimientos y contiene información acerca de los casos de prueba y el procedimiento a seguir para verificar si se están cumpliendo los requerimientos del proyecto en el tiempo determinado, esto se define en tres estados según corresponda: Completado, Pendiente y Fallido. En la ~~¡Error! No se encuentra el origen de la referencia.~~, se puede visualizar que el estado de todos los procedimientos cuenta como pendiente, debido a que es la primera versión registrada y aún no se presentan pruebas del sistema.

TABLA XXIV
REGISTRO DE TRAZABILIDAD DEL SI-003

ID	Requerimientos	Casos de Prueba	Procedimientos de prueba	Estado
RF1	El sistema solo es utilizado por una persona encargada de la recepción de clientes.	Iniciar sesión como usuario de recepción.	Ingresar al sistema con las credenciales de la persona encargada de la recepción.	Pendiente
		Intentar acceder con credenciales incorrectas.	Ingresar credenciales incorrectas y verificar el manejo de autenticación.	Pendiente
RF2	La página de inicio debe mostrar estadísticas clave, como el total de habitaciones, habitaciones ocupadas y disponibles.	Visualizar estadísticas en la página de inicio.	Acceder a la página de inicio y verificar la visualización de las estadísticas.	Pendiente
		Crear una nueva habitación	Agregar una nueva habitación a través de la funcionalidad de Crear.	Pendiente
RF3	Se deben permitir operaciones CRUD para la gestión de habitaciones, clientes, estado de habitación, ubicación y categoría de la habitación.	Modificar detalles de una habitación existente.	Editar los detalles de una habitación existente.	Pendiente
		Eliminar una habitación.	Eliminar una habitación a través de la funcionalidad de Eliminar.	Pendiente
		Crear un nuevo cliente	Agregar un nuevo cliente a través de la funcionalidad de Crear.	Pendiente
		Editar un cliente.	Editar los detalles de un cliente.	Pendiente
		Crear una nueva ubicación	Agregar una nueva ubicación a través de la funcionalidad de Crear.	Pendiente

TABLA XXV
(CONTINUACIÓN) REGISTRO DE TRAZABILIDAD DEL SI-003

RF3	Se deben permitir operaciones CRUD para la gestión de habitaciones, clientes, estado de habitación, ubicación y categoría de la habitación.	Eliminar una ubicación.	Eliminar una ubicación a través de la funcionalidad de Eliminar.	Pendiente
		Crear una nueva categoría	Agregar una nueva categoría través de la funcionalidad de Crear.	Pendiente
		Modificar una categoría existente.	Editar los detalles de una categoría.	Pendiente
		Eliminar una categoría.	Eliminar una categoría a través de la funcionalidad de Eliminar.	Pendiente
RF4	Las habitaciones deben clasificarse por categorías (simple, doble, triple y suite) y por ubicación (primer y segundo piso). Además, se debe permitir cambiar el estado de las habitaciones (disponible y ocupado).	Clasificación por categoría y ubicación.	Verificar que las habitaciones se clasifiquen correctamente por categoría y ubicación.	Pendiente
		Cambiar el estado de una habitación.	Modificar el estado de una habitación y verificar la actualización en el sistema	Pendiente
		Detalle de una habitación.	Verificar la información detallada de una habitación.	Pendiente
RF5	El sistema debe generar un comprobante de pago para los clientes al finalizar la recepción.	Generar un comprobante al finalizar una recepción.	Completar una recepción y verificar la generación del comprobante.	Pendiente
RF6	Deben generarse un informe sobre las recepciones dentro de un rango de fecha establecida con el total de ingresos, descargables en formato PDF.	Generar un informe sobre las recepciones.	Ingresar un rango de fecha y verificar que se genere el informe de la recepción.	Pendiente
		Descargar informe en formato PDF.	Verificar la descarga en formato PDF.	Pendiente

Por otra parte, la tarea de Construir el Software que es parte del SI-004 inicia con lentitud y termina con una tarea completa al 5%, por lo que se decide que esta tarea va a tomar el siguiente periodo de fechas correspondientes del 23 de diciembre de 2023 al 13 de enero de 2024.

En la **TABLA XXVI**, se presenta el instrumento diseñado para el registro de progreso de los productos de salida entregados al finalizar las actividades correspondientes a la primera etapa, que abarca desde el 23 de diciembre de 2023 hasta el 13 de enero de 2024. Este instrumento se utiliza para evaluar la calidad de la implementación de cada actividad del proceso SI, abarcando todos los productos de salida especificados para el SI en la guía ISO 29110 parte 5-1-2.

TABLA XXVI
PROGRESO DESDE EL 23/DIC/2023 HASTA EL 13/ENE/2024

Objetivos del SI											
ID	Actividad	Productos de Salida	Registro de progreso del 23/dic de 2023 al 13/ene de 2024								
			23/dic a 30/dic			31/dic a 06/ene			07/ene a 13/ene		
			%	Valor	Descripción	%	Valor	Descripción	%	Valor	Descripción
SI-004	Construcción de software	- Componentes de software [corregido, línea base]	20	1	Muy Bajo	40	2	Bajo	75	4	Alto
		- Registro de trazabilidad [actualizado, línea base]	0	1	Muy Bajo	0	1	Muy Bajo	0	1	Muy Bajo
			0	1	Muy Bajo	0	1	Muy Bajo	0	1	Muy Bajo
			0	1	Muy Bajo	0	1	Muy Bajo	0	1	Muy Bajo
			0	1	Muy Bajo	0	1	Muy Bajo	0	1	Muy Bajo
SI-005	Integración de software y pruebas	- Casos de prueba y procedimientos de prueba [línea base]	0	1	Muy Bajo	0	1	Muy Bajo	0	1	Muy Bajo
		- Software [probado, línea base]	0	1	Muy Bajo	0	1	Muy Bajo	0	1	Muy Bajo
		- Registro de trazabilidad [actualizado, línea base]	0	1	Muy Bajo	0	1	Muy Bajo	0	1	Muy Bajo
		- Informe de prueba [línea base]	0	1	Muy Bajo	0	1	Muy Bajo	0	1	Muy Bajo
		- Guía de operación del producto [verificada, línea base]	0	1	Muy Bajo	0	1	Muy Bajo	0	1	Muy Bajo
	- Documentación del usuario de software [verificada, línea base]	0	1	Muy Bajo	0	1	Muy Bajo	0	1	Muy Bajo	

En la **TABLA XXVI**, se pueden visualizar las tareas que debían ser realizadas en la semana del 23 de diciembre de 2023 hasta el 13 de enero de 2024, pero debido a la ausencia del programador por enfermedad, no se logró cumplir con la planificación de esta etapa, pero sí se nota el avance significativo que tuvo en la construcción del Software que terminó con un 75%.

En la **TABLA XXVII**, se presenta el instrumento diseñado para el registro de progreso de los productos de salida entregados al finalizar las actividades correspondientes a la primera etapa, que abarca desde el 14 de enero hasta el 04 de febrero de 2024. Este instrumento se utiliza para evaluar la calidad de la implementación de cada actividad del proceso SI, abarcando todos los productos de salida especificados para el SI en la guía ISO 29110 parte 5-1-2.

TABLA XXVII
PROGRESO DESDE EL 14/ENE/2024 HASTA EL 04/FEB/2024

Objetivos del SI															
ID	Actividad	Productos de Salida	Registro de progreso del 14/ene al 04/feb de 2024												
			14/ene a 21/ene			22/ene a 28/ene			29/ene a 03/feb			04/feb			
			%	Valor	Descripción	%	Valor	Descripción	%	Valor	Descripción	%	Valor	Descripción	
SI-004	Construcción de software	- Componentes de software [corregido, línea base]	89	5	Muy Alto	100	5	Muy Alto	100	5	Muy Alto	100	5	Muy Alto	
			10	1	Muy Bajo	50	3	Medio	100	5	Muy Alto	100	5	Muy Alto	
		- Registro de trazabilidad [actualizado, línea base]	0	1	Muy Bajo	40	2	Bajo	100	5	Muy Alto	100	5	Muy Alto	
			0	1	Muy Bajo	100	5	Muy Alto	100	5	Muy Alto	100	5	Muy Alto	
			0	1	Muy Bajo	100	5	Muy Alto	100	5	Muy Alto	100	5	Muy Alto	
SI-005	Integración de software y pruebas	- Casos de prueba y procedimientos de prueba [línea base]	0	1	Muy Bajo	60	3	Medio	100	5	Muy Alto	100	5	Muy Alto	
			0	1	Muy Bajo	50	3	Medio	100	5	Muy Alto	100	5	Muy Alto	
		- Software [probado, línea base]	0	1	Muy Bajo	50	3	Medio	100	5	Muy Alto	100	5	Muy Alto	
		- Registro de trazabilidad [actualizado, línea base]	0	1	Muy Bajo	60	3	Medio	100	5	Muy Alto	100	5	Muy Alto	
			0	1	Muy Bajo	100	5	Muy Alto	100	5	Muy Alto	100	5	Muy Alto	
		- Informe de prueba [línea base]	0	1	Muy Bajo	100	5	Muy Alto	100	5	Muy Alto	100	5	Muy Alto	
		- Guía de operación del producto [verificada, línea base]	0	1	Muy Bajo	100	5	Muy Alto	100	5	Muy Alto	100	5	Muy Alto	
			0	1	Muy Bajo	100	5	Muy Alto	100	5	Muy Alto	100	5	Muy Alto	
SI-006	Entrega del producto	- Documentación de mantenimiento [verificada, línea base]	0	1	Muy Bajo	0	1	Muy Bajo	99	5	Muy Alto	100	5	Muy Alto	
			0	1	Muy Bajo	0	1	Muy Bajo	99	5	Muy Alto	100	5	Muy Alto	
		- Configuración del software [entregado]	0	1	Muy Bajo	0	1	Muy Bajo	99	5	Muy Alto	100	5	Muy Alto	
			0	1	Muy Bajo	0	1	Muy Bajo	99	5	Muy Alto	100	5	Muy Alto	
			0	1	Muy Bajo	0	1	Muy Bajo	99	5	Muy Alto	100	5	Muy Alto	
			0	1	Muy Bajo	0	1	Muy Bajo	0	1	Muy Bajo	100	5	Muy Alto	

En la **TABLA XXVII**, se puede observar que la semana del 23 de diciembre de 2023 al 13 de enero de 2024, correspondiente a la tercera etapa (subrayada), se incluye debido a que no se logró completar en el tiempo acordado y se logró finalizar en la semana del 14 al 21 de enero para las actividades del SI-004. Es decir, las tareas que corresponden a resolver los componentes del Software detallados en el documento adjunto “SI-004 COMPONENTES DE SOFTWARE” y actualizar el Registro de Trazabilidad (**TABLA XXVIII**). El Registro de Trazabilidad correspondiente al SI-004 muestra un avance considerable con respecto a los módulos principales del sistema, los cuales hasta este registro se encuentran operando correctamente.

TABLA XXVIII
REGISTRO DE TRAZABILIDAD DEL SI-004

ID	Requerimientos	Casos de Prueba	Procedimientos de prueba	Estado
RF1	El sistema solo es utilizado por una persona encargada de la recepción de clientes.	Iniciar sesión como usuario de recepción.	Ingresar al sistema con las credenciales de la persona encargada de la recepción.	Completado
		Intentar acceder con credenciales incorrectas.	Ingresar credenciales incorrectas y verificar el manejo de autenticación.	Completado
RF2	La página de inicio debe mostrar estadísticas clave, como el total de habitaciones, habitaciones ocupadas y disponibles.	Visualizar estadísticas en la página de inicio.	Acceder a la página de inicio y verificar la visualización de las estadísticas.	Completado
RF3	Se deben permitir operaciones CRUD para la gestión de habitaciones, clientes, estado de habitación, ubicación y categoría de la habitación.	Crear una nueva habitación	Agregar una nueva habitación a través de la funcionalidad de Crear.	Completado
		Modificar detalles de una habitación existente.	Editar los detalles de una habitación existente.	Completado
		Eliminar una habitación.	Eliminar una habitación a través de la funcionalidad de Eliminar.	Completado
		Crear un nuevo cliente	Agregar un nuevo cliente a través de la funcionalidad de Crear.	Completado
		Editar un cliente.	Editar los detalles de un cliente.	Completado

TABLA XXIX
(CONTINUACIÓN) REGISTRO DE TRAZABILIDAD DEL SI-004

		Crear una nueva ubicación	Agregar una nueva ubicación a través de la funcionalidad de Crear.	Completado
		Modificar una ubicación existente.	Editar los detalles de una ubicación.	Completado
FR3	Se deben permitir operaciones CRUD para la gestión de habitaciones, clientes, estado de habitación, ubicación y categoría de la habitación.	Eliminar una ubicación.	Eliminar una ubicación a través de la funcionalidad de Eliminar.	Completado
		Crear una nueva categoría	Agregar una nueva categoría través de la funcionalidad de Crear.	Completado
		Modificar una categoría existente.	Editar los detalles de una categoría.	Completado
		Eliminar una categoría.	Eliminar una categoría a través de la funcionalidad de Eliminar.	Completado
		Clasificación por categoría y ubicación.	Verificar que las habitaciones se clasifiquen correctamente por categoría y ubicación.	Completado
RF4	Las habitaciones deben clasificarse por categorías (simple, doble, triple y suite) y por ubicación (primer y segundo piso). Además, se debe permitir cambiar el estado de las habitaciones (disponible y ocupado).	Cambiar el estado de una habitación.	Modificar el estado de una habitación y verificar la actualización en el sistema	Completado
		Detalle de una habitación.	Verificar la información detallada de una habitación.	Completado
RF5	El sistema debe generar un comprobante de pago para los clientes al finalizar la recepción.	Generar un comprobante al finalizar una recepción.	Completar una recepción y verificar la generación del comprobante.	Pendiente
RF6	Deben generarse un informe sobre las recepciones dentro de un rango de fecha establecida con el total de ingresos, descargables en formato PDF.	Generar un informe sobre las recepciones.	Ingresar un rango de fecha y verificar que se genere el informe de la recepción.	Pendiente
		Descargar informe en formato PDF.	Verificar la descarga en formato PDF.	Pendiente

Después de esto, en la siguiente semana (del 22 al 28 de enero de 2024) se completaron las actividades del SI-005, correspondientes a la etapa 3, las cuales consistieron en entregar los casos y procedimientos de prueba realizados al Software y así determinar y documentar los avances sugeridos (consultar documento adjunto “SI-005 CASOS Y PROCEDIMIENTOS DE PRUEBA”). Además, se entregó el informe de pruebas (consultar documento adjunto “SI-005 INFORME DE PRUEBA”) donde se corrigieron todos los defectos encontrados, y, finalmente se actualizó el registro de trazabilidad mostrado en la **TABLA XXX**. Para esta última versión del registro de trazabilidad se encontró que se completaron todos los procedimientos de prueba de manera exitosa, lo que resultó en un sistema funcional en los módulos planteados a partir de los requerimientos iniciales.

TABLA XXX
REGISTRO DE TRAZABILIDAD DEL SI-005

	Requerimientos	Casos de Prueba	Procedimientos de prueba	Estado
RF1	El sistema solo es utilizado por una persona encargada de la recepción de clientes.	Iniciar sesión como usuario de recepción.	Ingresar al sistema con las credenciales de la persona encargada de la recepción.	Completado
		Intentar acceder con credenciales incorrectas.	Ingresar credenciales incorrectas y verificar el manejo de autenticación.	Completado
RF2	La página de inicio debe mostrar estadísticas clave, como el total de habitaciones, habitaciones ocupadas y disponibles.	Visualizar estadísticas en la página de inicio.	Acceder a la página de inicio y verificar la visualización de las estadísticas.	Completado
		Crear una nueva habitación	Agregar una nueva habitación a través de la funcionalidad de Crear.	Completado
RF3	Se deben permitir operaciones CRUD para la gestión de habitaciones, clientes, estado de habitación, ubicación y categoría de la habitación.	Modificar detalles de una habitación existente.	Editar los detalles de una habitación existente.	Completado
		Eliminar una habitación.	Eliminar una habitación a través de la funcionalidad de Eliminar.	Completado
		Crear un nuevo cliente	Agregar un nuevo cliente a través de la funcionalidad de Crear.	Completado
		Editar un cliente.	Editar los detalles de un cliente.	Completado

TABLA XXXI
(CONTINUACIÓN) REGISTRO DE TRAZABILIDAD DEL SI-005

		Crear una nueva ubicación	Agregar una nueva ubicación a través de la funcionalidad de Crear.	Completado
		Modificar una ubicación existente.	Editar los detalles de una ubicación.	Completado
		Eliminar una ubicación.	Eliminar una ubicación a través de la funcionalidad de Eliminar.	Completado
RF3	Se deben permitir operaciones CRUD para la gestión de habitaciones, clientes, estado de habitación, ubicación y categoría de la habitación.	Crear una nueva categoría	Agregar una nueva categoría través de la funcionalidad de Crear.	Completado
		Modificar una categoría existente.	Editar los detalles de una categoría.	Completado
		Eliminar una categoría.	Eliminar una categoría a través de la funcionalidad de Eliminar.	Completado
		Clasificación por categoría y ubicación.	Verificar que las habitaciones se clasifiquen correctamente por categoría y ubicación.	Completado
RF4	Las habitaciones deben clasificarse por categorías (simple, doble, triple y suite) y por ubicación (primer y segundo piso). Además, se debe permitir cambiar el estado de las habitaciones (disponible y ocupado).	Cambiar el estado de una habitación.	Modificar el estado de una habitación y verificar la actualización en el sistema	Completado
		Detalle de una habitación.	Verificar la información detallada de una habitación.	Completado
RF5	El sistema debe generar un comprobante de pago para los clientes al finalizar la recepción.	Generar un comprobante al finalizar una recepción.	Completar una recepción y verificar la generación del comprobante.	Completado
		Generar un informe sobre las recepciones.	Ingresar un rango de fecha y verificar que se genere el informe de la recepción.	Completado
RF6	Deben generarse un informe sobre las recepciones dentro de un rango de fecha establecida con el total de ingresos, descargables en formato PDF.	Descargar informe en formato PDF.	Verificar la descarga en formato PDF.	Completado

Luego, en el periodo correspondiente del 29 de enero al 03 de febrero se logró obtener la versión final del producto Software, la cual se encuentra guardada en un repositorio de GitHub “*Sys-Hotel-ISO-29110-parte-5-1-2*”. También, se entregó la guía de operación del producto (consultar documento adjunto “*SI-005 GUÍA DE OPERACIÓN DEL PRODUCTO*”) y la versión final del manual de Usuario, la cual se puede verificar en el documento adjunto “*SI-005 MANUAL DE USUARIO*”.

Para finalizar, se entregó el documento de mantenimiento revisado en el documento adjunto “SI-006 DOCUMENTACIÓN DE MANTENIMIENTO” y el día 04 de febrero se realizó la prueba del producto final configurado, para poder llevar a cabo la encuesta de satisfacción del cliente.

5) Gestión del Trabajo Realizado:

Se ha desarrollado un Burn-up Chart enfocado exclusivamente en la “Línea de Trabajo Completado” para visualizar el avance del sistema, desglosado en porcentajes para cada actividad del SI. Este gráfico presenta una única línea que representa la cantidad de trabajo realizado a lo largo del tiempo. La línea comienza en cero al inicio del sistema y progresa hacia arriba a medida que se logran avances hasta llegar al 100%. Este enfoque simplificado permite una visualización clara y directa del progreso del sistema, centrándose exclusivamente en el trabajo completado hasta la fecha, lo cual se visualiza en la **Fig. 37**.

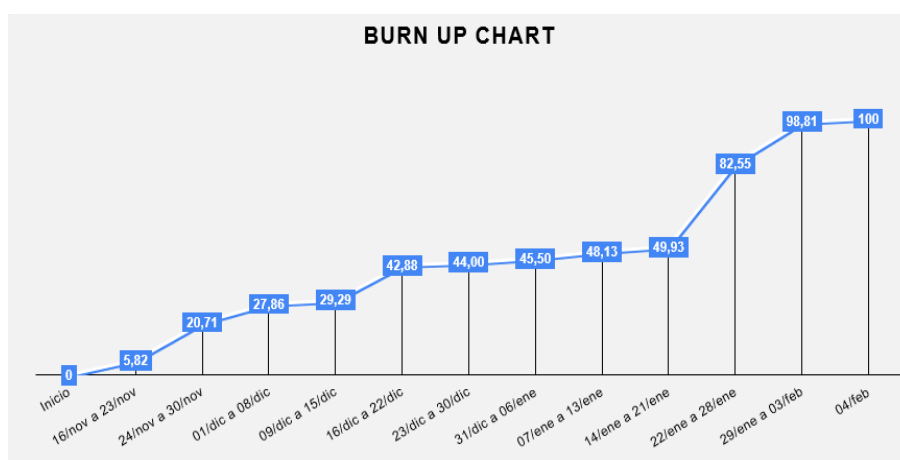


Fig. 37. Gráfico Burn-Up Chart

En la **Fig. 37**, se observa el trabajo realizado utilizando la norma ISO 20110, para tener una visión más clara del progreso de un equipo en la implementación del software. Durante la primera a la cuarta semana (09 al 15 de diciembre de 2023) el equipo progresa y da un salto de un 13% hacia la siguiente semana para compensar los días en los que el programador estuvo ausente por enfermedad. Luego, la semana del 14 al 21 de enero de 2024 se mantuvo constante para luego dar un salto de 3 semanas, del 22 al 03 de febrero, en donde se observa que se realizó el trabajo en menos tiempo debido a que, principalmente, se tenían que actualizar los documentos ya existentes.

6) Producto Final – sistema ISO

El día 04 de febrero se llevó a cabo una sesión de prueba del producto final durante la cual se proporcionó al usuario acceso al producto y a la documentación para su revisión. Es importante destacar que esta entrega no implicó una entrega oficial del producto para su uso continuo, sino que fue exclusivamente para que el usuario probara su funcionalidad. En este apartado se va a describir las funcionalidades principales del sistema web desarrollado con la norma ISO 29110 parte 5-1-2, la arquitectura MVC, en el framework Laravel y con la base de datos MySQL.

En la **Fig. 38**, se visualiza el Login de inicio de sesión (Fig. 38a) que permite la autenticación con el nombre de usuario y contraseña. Luego, al ingresar correctamente, en la página de inicio se muestran estadísticas para el total de habitaciones, las habitaciones disponibles y las ocupadas (Fig. 38b).

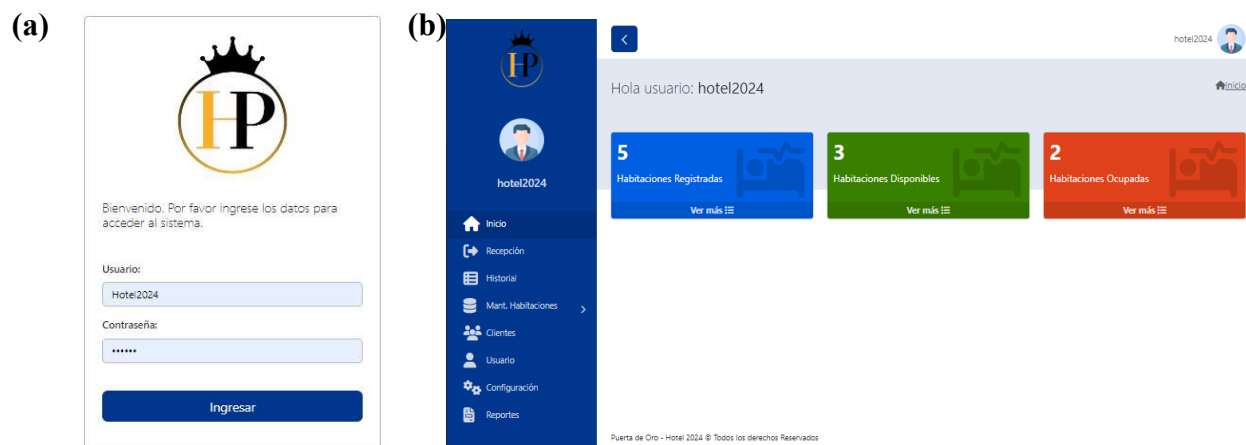


Fig. 38. Inicio de sesión en el sistema web. (a) Login de acceso al sistema. (b) Página principal de inicio con estadísticas

A continuación, en la **Fig. 39** se visualizan los módulos de Habitaciones, Categorías, Ubicaciones y Clientes. En el módulo de clientes (Fig. 39d), se visualizan los clientes agregados al momento de realizar las recepciones y para estos solo se puede realizar la edición de los datos, en caso de que sea necesario. En los módulos de Categorías (Fig. 39b) y Ubicaciones (Fig. 39c) se dispone de una opción para agregar nuevos registros, editarlos, eliminarlos y activar o desactivar según sea conveniente. Finalmente, en el módulo de Habitaciones (Fig. 39a) se puede agregar una nueva habitación con una categoría y una ubicación específica, así mismo, se puede editar y eliminar un registro.

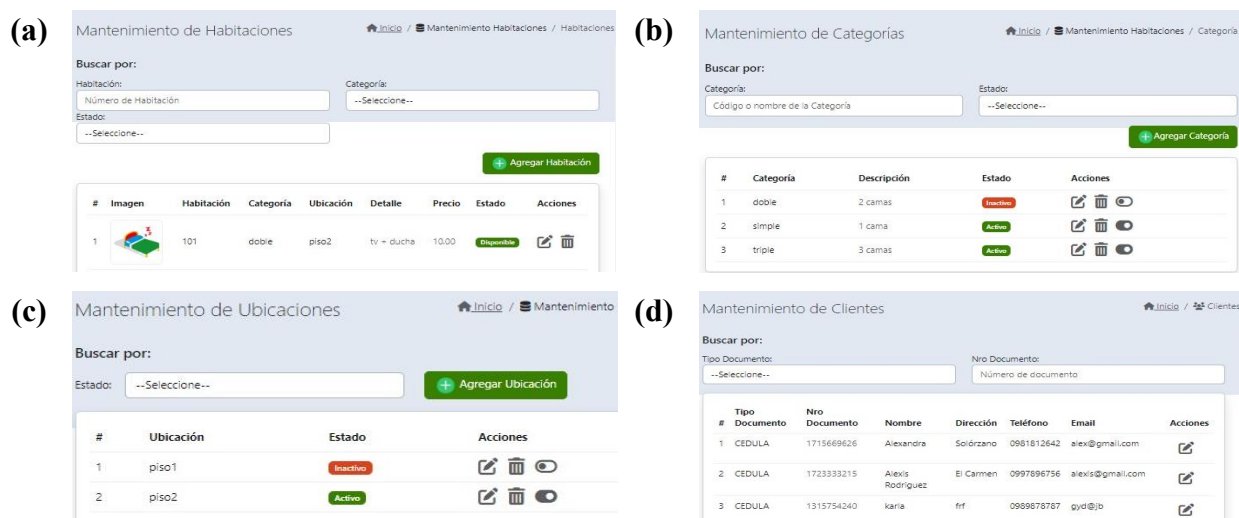


Fig. 39. Módulos funcionales. (a) Habitaciones. (b) Categorías. (c) Ubicaciones. (d) Clientes

En la **Fig. 40**, se visualiza la vista general del módulo de recepciones, aquí se muestran las habitaciones disponibles de color verde y las ocupadas de color rojo las cuales se pueden organizar por la ubicación (Fig. 40a). Una vez elegida la habitación, se registran los datos del cliente (si ya está registrado, aparecerán automáticamente) y el alojamiento (Fig. 40b). Posteriormente, para finalizar la recepción se muestra una ventana con los datos y con la opción de generar un comprobante de pago (Fig. 40c). Finalmente, al elegir la opción para generar un comprobante se muestra una ventana con la vista previa del comprobante para imprimir (Fig. 40d).

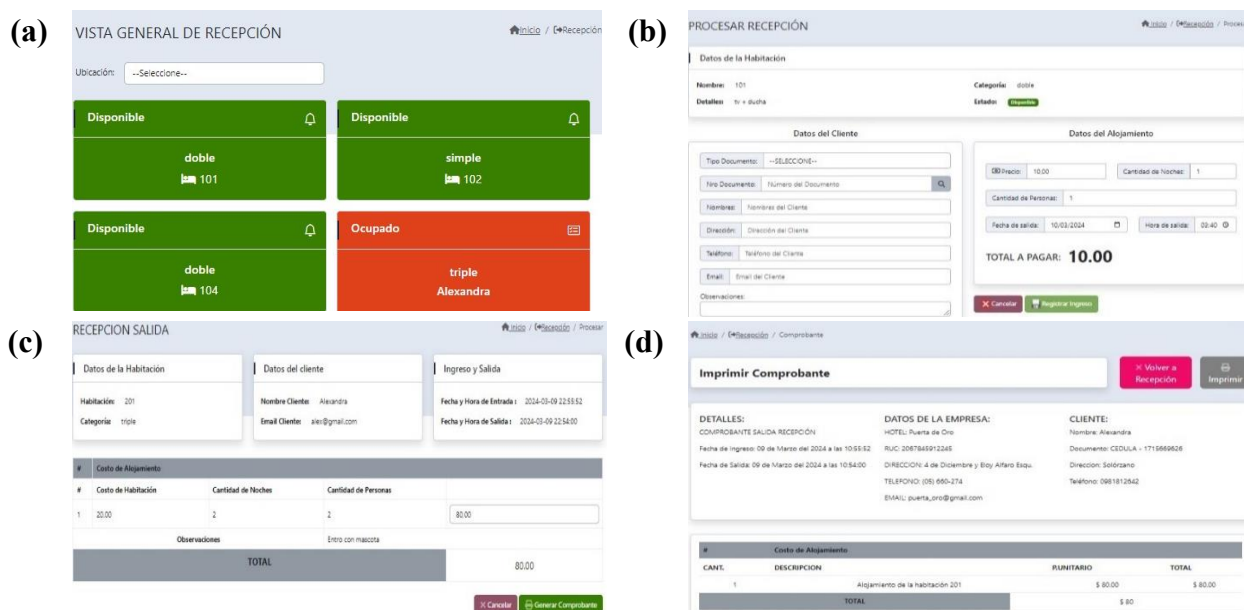


Fig. 40. Módulo de Recepciones. (a) Vista general de las recepciones. (b) Proceso de una recepción. (c) Proceso de salida de la recepción. (d) Imprimir comprobante

A continuación, se presentan las funcionalidades del sistema (**Fig. 41**): Mantenimiento de usuarios que ayuda a editar la información del único usuario registrado (Fig. 41a); Listado de parámetros de configuración que edita la información del hotel para utilizarla posteriormente en los comprobantes (Fig. 41b); Historial de las recepciones realizadas, las cuales se muestran en una tabla (Fig. 41c); Módulo para generar el reporte de ingresos de las recepciones en un tiempo determinado por el usuario (Fig. 41d).

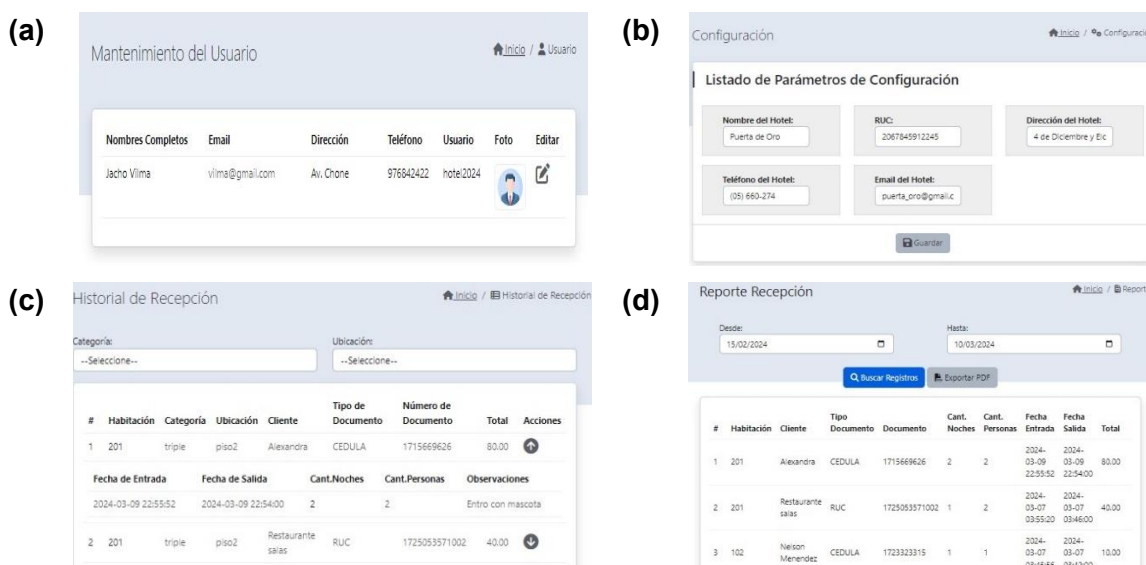


Fig. 41. Funcionalidades del sistema: (a) Mantenimiento del usuario. (b) Información del Hotel. (c) Historial de Recepciones. (d) Reporte de recepciones

D. Análisis del SCRUM y la norma ISO parte 5-1-2

El estudio actual lleva a cabo una evaluación comparativa de la calidad del software entre los dos sistemas: el primero, desarrollado utilizando el marco ágil SCRUM, y el segundo, conforme a la norma ISO/IEC 29110, parte 5-1-2. Para este análisis, se seleccionaron cuatro dimensiones de evaluación: el análisis de calidad de código mediante SonarQube, las encuestas de satisfacción utilizando Google Forms, la comparación de la cantidad de documentación y la comparación de roles. A continuación, se realiza el análisis de los sistemas utilizando SonarQube y la encuesta de satisfacción.

1) SonarQube:

La utilización de SonarQube para el análisis estático del código ha facilitado la identificación y cuantificación de aspectos fundamentales, tales como la seguridad, la fiabilidad, la mantenibilidad, la revisión de puntos de acceso, la cobertura y la duplicación de código.

a) *Análisis de SCRUM:*

El sistema desarrollado utilizando el marco ágil SCRUM, se realizó el análisis de calidad tanto para el Front-end (**Fig. 42**) como para el Back-end (**Fig. 43**). Se presenta a continuación la visualización de los resultados obtenidos.

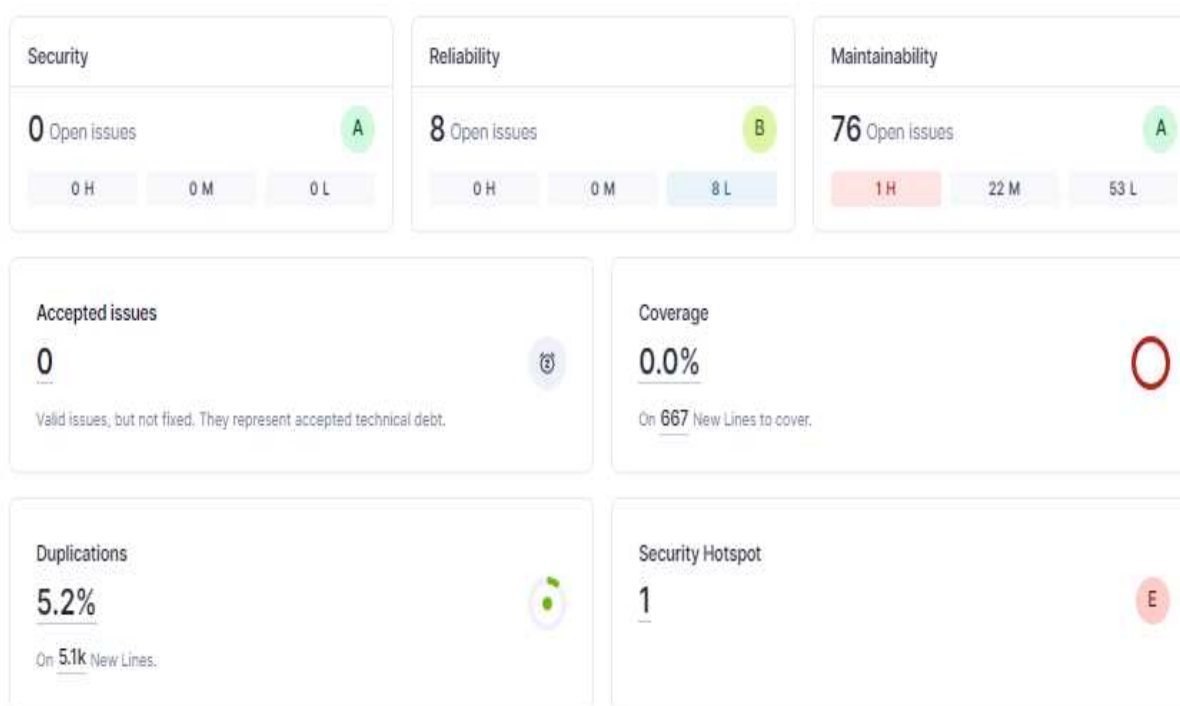


Fig. 42. Análisis del sistema SCRUM en el Front-end

En la **Fig. 42**, el SCRUM en su parte de Front-end ha superado la puerta de calidad de SonarQube, lo que indica que el código cumple con los estándares mínimos de calidad establecidos. Obtuvo una calificación de fiabilidad 'B', señalando la presencia de 8 problemas abiertos que deben ser atendidos para prevenir inconvenientes en el sistema. También se detectó un problema en el punto de acceso de seguridad, que debe ser resuelto para garantizar la protección del sistema. La mantenibilidad se sitúa en 76, con una calificación 'A', lo cual sugiere que, aunque el código es limpio, organizado y bien estructurado, el índice de deuda técnica revela la necesidad de realizar ciertas mejoras menores para alcanzar un nivel óptimo de código. Este sistema presenta un 0,0% de cobertura de pruebas, indicando que varias secciones del código no han sido evaluadas mediante pruebas automáticas, ya sea por la ausencia de pruebas que confirmen su funcionamiento adecuado o porque las pruebas existentes no se ejecutaron correctamente. Además, existe un 5,2% de

duplicidad en el código, lo que indica la presencia de líneas de código repetidas que, de ser optimizadas, podrían elevar aún más la calidad del sistema.

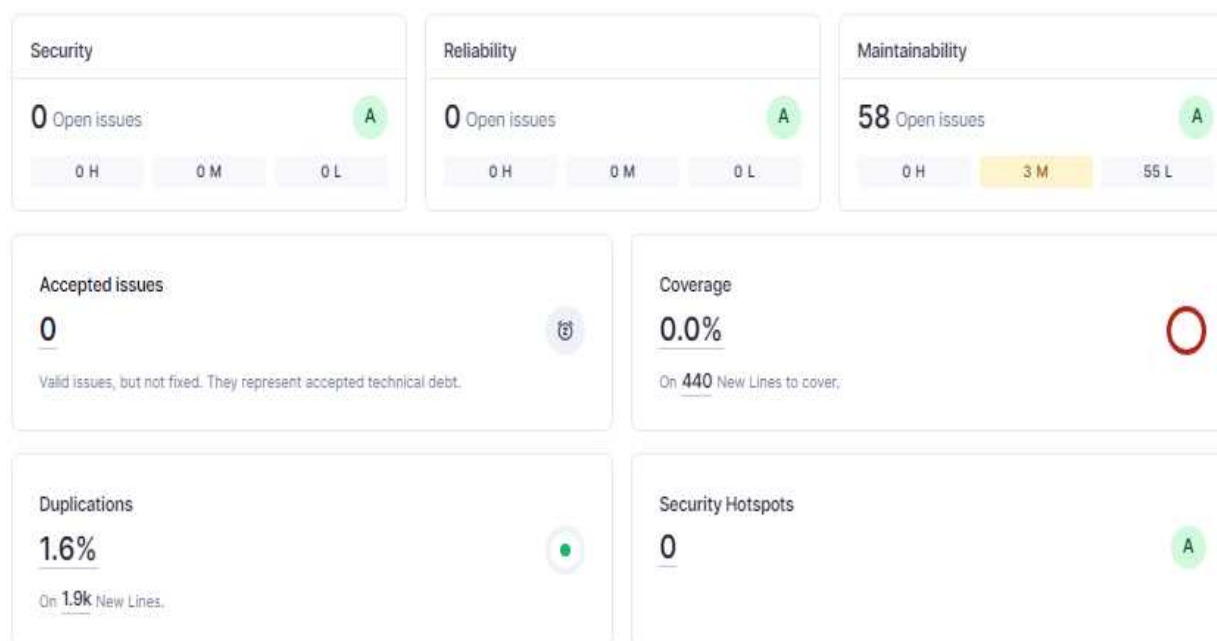


Fig. 43. Análisis del sistema SCRUM en el Back-end

En la **Fig. 43**, el SCRUM en su parte de Back-end ha superado la puerta de calidad de SonarQube, lo que indica que el código cumple con los estándares mínimos de calidad establecidos. Obtuvo una calificación de fiabilidad 'A', indicando la ausencia de problemas abiertos y el bajo riesgo de fallos de ejecución, además no se detectaron problemas en el punto de acceso de seguridad. La mantenibilidad se sitúa en 58, con una calificación 'A', lo cual sugiere que, aunque el código es limpio, organizado y bien estructurado, el índice de deuda técnica revela la necesidad de realizar ciertas mejoras menores para alcanzar un nivel óptimo de código. Este sistema presenta un 0,0% de cobertura de pruebas, indicando que varias secciones del código no han sido evaluadas mediante pruebas automáticas, ya sea por la ausencia de pruebas que confirmen su funcionamiento adecuado o porque las pruebas existentes no se ejecutaron correctamente. Además, existe un 1,6% de duplicidad en el código, lo que indica la presencia de líneas de código repetidas que, de ser optimizadas, podrían elevar aún más la calidad del sistema.

b) Análisis de la norma ISO:

Se realizó el análisis de calidad, para el sistema desarrollado con la norma ISO 29110 parte 5-1-2, tanto para el Front-end (**Fig. 44**) como para el Back-end (**Fig. 45**). Se presenta a continuación la visualización de los resultados obtenidos.

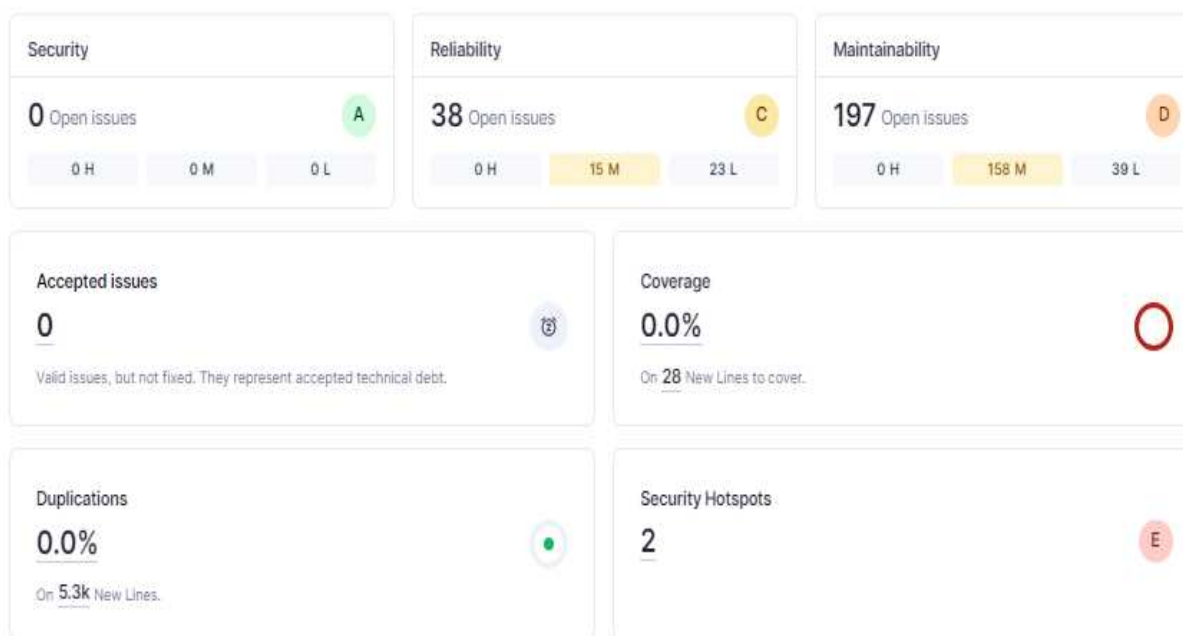


Fig. 44. Análisis del sistema de la norma ISO en el Front-end

En la **Fig. 44**, el sistema desarrollado con la norma ISO en su parte de Front-end ha superado la puerta de calidad de SonarQube, lo que indica que el código cumple con los estándares mínimos de calidad establecidos. Obtuvo una calificación de fiabilidad ‘C’, señalando la presencia de 38 problemas abiertos que deben ser atendidos para evitar tener fallos en el sistema. También detectaron dos problemas en el punto de acceso de seguridad, que deben ser resueltos para garantizar la protección del sistema. La mantenibilidad tiene un índice de 197 y una calificación de ‘D’, lo que señala que el código presenta dificultades para ser comprendido, actualizado o extendido debido a la presencia de “code smells”, a la falta del uso de las buenas prácticas de programación. Este sistema presenta un 0,0% de cobertura de pruebas, indicando que varias secciones del código no han sido evaluadas mediante pruebas automáticas, ya sea por la ausencia de pruebas que confirmen su funcionamiento adecuado o porque las pruebas existentes no se ejecutaron correctamente. Además, existe un 0,0% de duplicidad en el código, lo que indica que no existen líneas de código repetidas en el sistema.

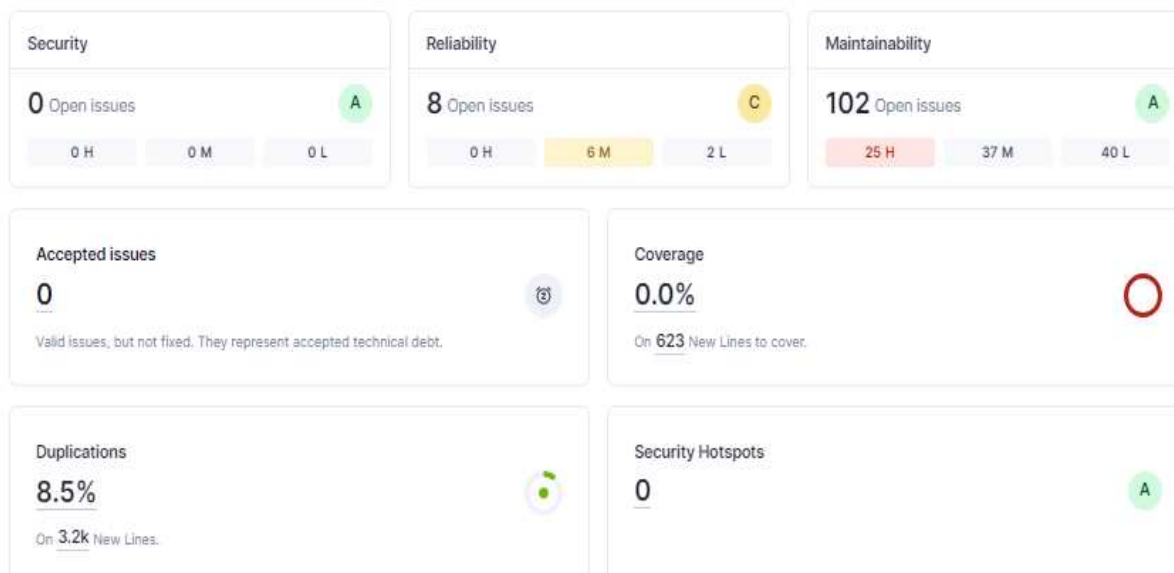


Fig. 45. Análisis del sistema de la norma ISO en el Back-end

En la **Fig. 45**, el sistema desarrollado con la norma ISO en su parte de Back-end ha superado la puerta de calidad de SonarQube, lo que indica que el código cumple con los estándares mínimos de calidad establecidos. Obtuvo una calificación de fiabilidad ‘C’, señalando la presencia de 8 problemas abiertos que deben ser atendidos para evitar tener fallos en el sistema, además no se detectaron problemas en el punto de acceso de seguridad. La mantenibilidad se sitúa en 102, con una calificación ‘A’, lo cual sugiere que, aunque el código es limpio, organizado y bien estructurado, el índice de deuda técnica revela la necesidad de realizar ciertas mejoras menores para alcanzar un nivel óptimo de código. Este sistema presenta un 0,0% de cobertura de pruebas, indicando que varias secciones del código no han sido evaluadas mediante pruebas automáticas, ya sea por la ausencia de pruebas que confirmen su funcionamiento adecuado o porque las pruebas existentes no se ejecutaron correctamente. Además, existe un 8,5% de duplicidad en el código, lo que indica la presencia de líneas de código repetidas que, de ser optimizadas, podrían elevar aún más la calidad del sistema.

2) Encuesta de Satisfacción:

Se realizó una encuesta de satisfacción para evaluar la usabilidad del sistema para el usuario final, la gerente del Hotel Puerta de Oro y sus tres colaboradores, únicos miembros del personal debido a que se comprende como una VSE. Esta encuesta abordó temas como la calidad del Sistema y funcionalidad del sistema, la interfaz de usuario y la satisfacción general y usabilidad del sistema. Cada sección tiene preguntas diseñadas para que el usuario las pueda responder una vez que se

haya entregado el sistema. Para lograr esto, se utilizó la herramienta Google Forms, que, entre otras cosas, realiza encuestas en línea. Esto nos ayudó a que el proceso fuera más analítico al evaluar la calidad de la experiencia del usuario en el sistema web de la recepción del hotel. En la **TABLA XXXII**, se muestran las preguntas diseñadas para abarcar los temas propuestos y las cuales también están disponibles en el siguiente formulario en línea <https://forms.gle/6oQ6YpLpxpNpt1dw9>.

TABLA XXXII
PREGUNTAS DE LA ENCUESTA DE SATISFACCIÓN

SUB-CARACTERÍSTICA DE USABILIDAD	PREGUNTAS	PESO
Calidad del Sistema y Funcionalidad del Sistema	1. En una escala del 1 al 5, donde 1 representa “Muy insatisfecho” y 5 representa “Muy satisfecho”, ¿cómo calificaría su satisfacción con la funcionalidad del sistema? 5 - Muy satisfecho 4 - Satisfecho 3 - Neutral 2 - Insatisfecho 1 - Muy insatisfecho	5
	2. Respecto a la calidad del sistema, ¿considera usted que cumple con tus estándares de excelencia y expectativas? 5 - Si 1- No	5
Interfaz de Usuario	3. ¿Detectó algún problema de error mientras se utilizó el sistema? 5 - No experimenté ningún problema ni error 4 - Encontré 1 a 2 problemas que afectaron moderadamente la experiencia 3 - Experimenté hasta 5 problemas que dificultaron el uso del sistema 2 - Enfrenté hasta 10 problemas y errores críticos que impidieron el uso efectivo del sistema 1 - No pude utilizar el sistema	5
	4. ¿Considera usted que la interfaz de usuario es estéticamente agradable y fácil de usar? 5 - Si 1- No	5
Satisfacción General y Usabilidad del Sistema	5. En una escala del 1 al 10, donde 1 representa una apariencia visual muy deficiente y 10 representa una apariencia visual excepcional, por favor califique la apariencia de la interfaz de usuario del sistema	5
	6. En una escala del 1 al 5, ¿Cómo calificarías tu grado de satisfacción global con el sistema, considerando tu experiencia completa desde la calidad hasta el servicio recibido? 5 - (Muy satisfecho) 4 - (Satisfecho) 3 - (Neutral) 2 - (Insatisfecho) 1 - (Muy insatisfecho)	5
	7. ¿El sistema le permite completar todas las tareas necesarias para la gestión del hotel de manera satisfactoria? 5 - Si 1- No	5

TABLA XXXIII
(CONTINUACIÓN) PREGUNTAS DE LA ENCUESTA DE SATISFACCIÓN

	8. Califique la capacidad del sistema en cuanto a su operabilidad:	
	5 - Muy Alta (El sistema es altamente operable)	
	4 - Alta (El sistema es fácil de operar en la mayoría de los módulos)	5
	3 - Moderada (El sistema es operable en general, pero podría mejorar)	
	2 - Baja (El sistema tiene dificultades significativas en su operación)	
	1 - Muy Baja (El sistema es extremadamente difícil de operar)	
	9. El sistema gestiona adecuadamente la entrada de nueva información y la actualización de datos existentes.	
	5 - El sistema maneja la información correctamente durante mi uso	
Satisfacción General y Usabilidad del Sistema	4 - El sistema tuvo algunos problemas menores al manejar la información	5
	3 - Tuve dificultades moderadas con el sistema al intentar ingresar información	
	2 - Encontré numerosos problemas que afectaron significativamente mi capacidad para ingresar información al sistema	
	1 - El sistema fue incapaz de manejar la información adecuadamente durante su uso	
	10. ¿Quedaste satisfecho con la atención a la cliente proporcionada durante el sistema?	
	5 - Muy Satisfecho	
	4 - Satisfecho	5
	3 - Neutral	
	2 - Insatisfecho	
	1 -Muy Insatisfecho	

IV. Resultados

A. Resultados de SonarQube

Para la evaluación de la calidad de los sistemas web mediante SonarQube, se incluyeron métricas como la Seguridad, la Fiabilidad, la Mantenibilidad, la Cobertura, los Puntos de Acceso y las Duplicaciones. En la **TABLA XXXIV** se observa que, para la seguridad, fiabilidad, mantenibilidad y puntos de acceso, se puntúa mediante SQALE, el cual es un modelo de calidad generado para brindar soporte a la evaluación de software. Este rating se mide con las letras: A (Calidad Excelente), B (Calidad notable), C (Calidad insuficiente), D (Calidad baja) y E (Calidad muy baja).

TABLA XXXIV
RESULTADOS DE SONARQUBE MEDIANTE SQALE

Sistema	Seguridad					Fiabilidad					Mantenibilidad					Puntos de acceso				
	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
Rating																				
SCRUM en el Front-end	x						x				x									
SCRUM en el Back-end	x					x					x					x				
ISO en el Front-end	x							x						x						
ISO en el Back-end	x							x			x					x				

En la **TABLA XXXIV** se muestra el primer parámetro correspondiente a la seguridad, el cual tuvo una puntuación de A, tanto en el Front-end como en el Back-end, para los dos sistemas, lo cual indica que el sistema cumple con los estándares mínimos de seguridad. Luego, se muestra que el sistema con SCRUM tiene una alta calidad, en el Front-end y en el Back.end con una puntuación de fiabilidad de B y A, respectivamente, además se detectó un problema en el punto de acceso de seguridad y en la mantenibilidad tiene una calificación de A lo que sugiere la necesidad de mejoras menores. Por otro lado, el sistema desarrollado con la norma ISO tuvo una puntuación de fiabilidad de C en el Front-end y el Back.end, y se detectaron dos problemas en el punto de acceso de seguridad lo que sugiere dificultades significativas en la mantenibilidad con una calificación de D. Por último, no se detectaron problemas en el punto de acceso de seguridad y la mantenibilidad se encuentra en A, lo que sugiere la necesidad de mejoras menores.

En la **Fig. 46**, se indica que la cobertura de las pruebas es del 0,0%, para el sistema con SCRUM lo que manifiesta la ausencia de evaluación de varias secciones del código y un 5,2% de duplicación en el código. Para el sistema realizado con la ISO la cobertura de las pruebas es del 0,0%, lo que indica la ausencia de evaluación de varias secciones del código y un 8,5% de duplicidad en el código, lo que sugiere la presencia de líneas de código repetidas que deben ser optimizadas

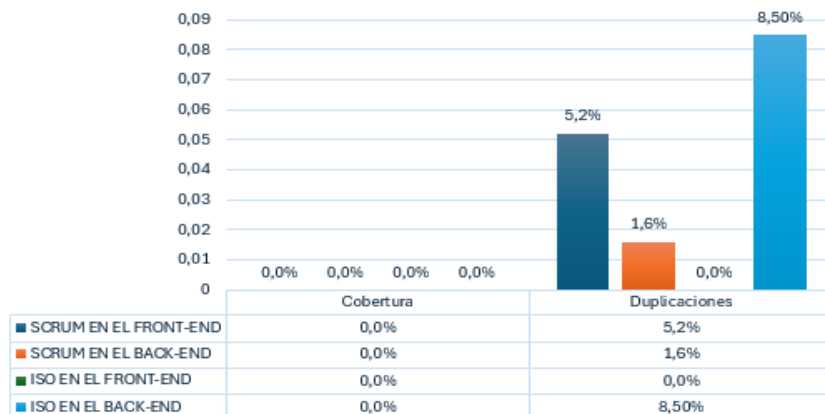


Fig. 46. Resultados de Cobertura y Duplicaciones

B. Resultados de la encuesta de satisfacción

La encuesta de satisfacción fue realizada para el usuario final, la gerente del Hotel Puerta de Oro y sus tres colaboradores. Esta encuesta se compone de 10 preguntas, agrupadas en tres categorías: Calidad del Sistema y Funcionalidad del Sistema, Interfaz de Usuario, y Satisfacción General con la Usabilidad del Sistema. Los resultados de esta encuesta se presentan en la **Fig. 47**, donde se visualizan las respuestas del usuario final para cada sistema. La línea de color rojo representa al SCRUM (Fig. 47-a), mientras que la línea de color verde corresponde al ISO (Fig. 47-b). Además, se incluyen los puntajes obtenidos en cada una de las preguntas y una gráfica comparativa que se puede visualizar en la Fig. 47-c.

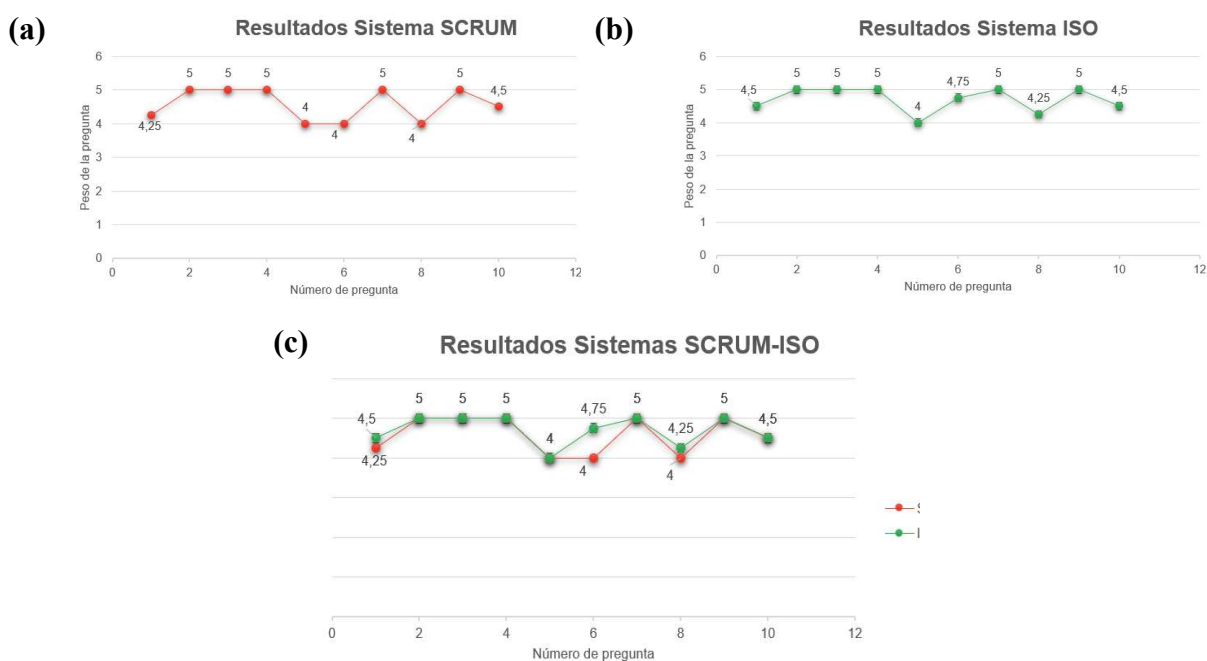


Fig. 47. Respuestas de la encuesta de satisfacción

En la **Fig. 47**, se visualizan los resultados de las respuestas obtenidas a través del formulario de Google Forms, completado por los usuarios finales al evaluar ambos sistemas. En este se indica que el sistema desarrollado con la norma ISO obtuvo un puntaje superior al SCRUM, específicamente en la pregunta 6 y 8, correspondiente a la Satisfacción General y Usabilidad del Sistema. En la Figura 38-c, se demuestra que, en la mayoría de las preguntas, ambos sistemas satisfacen los requisitos establecidos inicialmente, a excepción de la pregunta 6 y la pregunta 8, lo que nos da a entender que el sistema ISO, con un puntaje total de 94%, fue calificado como más operable que el SCRUM que obtuvo un puntaje total de 91,50%.

C. Resultados sobre roles y responsabilidades

La asignación de roles y responsabilidades en la norma ISO 29110 parte 5-1-2 y en SCRUM se establecieron de manera diferente, reflejando sus distintos enfoques hacia la gestión y ejecución de sistemas, especialmente en el desarrollo de software. Por un lado, como se puede observar en la **TABLA III**, los roles de SCRUM se enfocan más en las funciones dentro del proceso ágil de desarrollo de software, donde el Product Owner representa los intereses del cliente y el SCRUM Master facilita el proceso. Por otro lado, la norma ISO establece roles más específicos, como el Cliente, el Analista, el Diseñador y el Programador (**TABLA XVIII**), que reflejan una división más detallada de responsabilidades dentro del equipo de trabajo. Algunos roles pueden superponerse, como el Gerente de Proyecto en la norma ISO y el SCRUM Master en el marco ágil SCRUM, otro ejemplo es el Cliente en la norma ISO.

TABLA XXXV
COMPARACIÓN DE ROLES Y RESPONSABILIDADES

Clasificación	ISO	SCRUM
Gestión y Dirección	1	2
Análisis y Diseño	3	0
Desarrollo e Implementación	2	1
Usuarios	1	0

En la **TABLA XXXV**, se muestran los resultados del conteo de roles que tuvieron lugar en la ejecución de cada sistema, estos se clasificaron en 4 grupos que fueron la Gestión y Dirección, el Análisis y Diseño, el Desarrollo e Implementación y los Usuarios. Esto da como resultado 7 roles para el sistema desarrollado con la norma ISO y 3 roles para el marco ágil SCRUM, y debido a esto, la norma ISO 29110 parte 5-1-2 es más recomendable para sistemas de gran magnitud, en donde se requieran varios profesionales.

D. Resultados sobre la cantidad de documentación

La evaluación de la documentación para el marco SCRUM y la norma ISO 29110 parte 5-1-2 se realizó mediante los productos entregados al final de cada planificación. Debido a las diferencias significativas entre ambos enfoques, se tuvo en cuenta, para el marco SCRUM, el Product Backlog, el resumen de cada Sprint generado por cada reunión realizada al final del Sprint y el manual de usuario. Por otro lado, en la documentación de la ISO, se menciona que se deben de entregar productos de salida en cada objetivo completado. Estos documentos se generan con el fin de actualizar la información frecuentemente y así contactar con los roles involucrados para evidenciar los cambios realizados.

TABLA XXXVI
COMPARACIÓN DE LA CANTIDAD DE DOCUMENTACIÓN

Categorías	ISO	SCRUM
Documentación de Requisitos	1	0
Documentación de Diseño	2	0
Documentación de Pruebas	3	0
Documentación de Usuario	3	1
Documentación de Mantenimiento	1	0
Documentación de Proyecto	0	5

En la **TABLA XXXVI**, se puede observar que el SCRUM generó un total de 6 documentos para evidenciar el trabajo realizado, mientras que la norma ISO generó un total de 10 documentos, los cuales se deben realizar y entregar como producto de salida. Con respecto a esto, se observa que el marco ágil SCRUM genera menos documentación, priorizando el producto final. Por otro lado, la norma ISO 29110 parte-5-1-2 enfatiza en una documentación exhaustiva y detallada a lo largo del proceso, lo que resulta en un marco más estructurado. Por lo tanto, si se quisiera realizar un sistema pequeño en poco tiempo, la opción recomendada sería el marco ágil SCRUM debido a su poca cantidad de documentación, lo que da más tiempo a la construcción del software.

E. Comparación de Resultados

El análisis comparativo entre los sistemas realizados con el marco SCRUM y la norma ISO 29110 parte 5-1-2, como se muestra en la **TABLA XXXVII** revela distintas fortalezas y consideraciones importantes para la selección del enfoque más adecuado en un contexto determinado. El sistema ISO, caracterizado por la amplia variedad de roles y una extensa documentación, exhibe una estructura más detallada y orientada hacia la gestión exhaustiva de

requisitos y procesos. Esto se refleja en su alta calificación de satisfacción del usuario con un 94%, lo que sugiere una entrega de producto altamente satisfactoria. Sin embargo, el SCRUM, con una estructura de equipo más simplificada y un enfoque ágil, presenta una calificación ligeramente más baja en la satisfacción del usuario (91,50%), pero aun así indica un nivel aceptable de satisfacción.

TABLA XXXVII
COMPARACIÓN DE RESULTADOS DE LA ISO 29110 PARTE 5-1-2 Y EL MARCO ÁGIL SCRUM

Categorías	Instrumento Evaluado	ISO	SCRUM
Roles	Cantidad	7	3
Documentación	Cantidad	10	6
Satisfacción del Usuario	Calificación total	94%	91,5%
	Seguridad	A / A	A / A
	Fiabilidad	C / C	B / A
Calidad del Sistema (Front-end / Back-end)	Mantenibilidad	D / A	A / A
	Punto de Acceso	A / E	A / E
	Cobertura	0% / 0%	0% / 0%
	Duplicaciones	0% / 8,5%	5,2% / 1,6%

En términos de calidad del sistema, aunque ambos sistemas muestran métricas positivas en fiabilidad y mantenibilidad, el sistema desarrollado con la norma ISO exhibe un mayor detalle en su evaluación de calidad, mientras que el SCRUM proporciona una visión más ágil y flexible. Esto sugiere que la elección entre ambos sistemas dependerá del contexto específico del cliente y del sistema. Esto quiere decir que, aunque el sistema web del Hotel Puerta de Oro tuvo una mejor calificación utilizando la norma ISO 29110 parte 5-1-2, es preferible utilizar el marco ágil SCRUM debido a que, en base a los requisitos expuestos, el sistema no requiere una gestión exhaustiva de procesos, estándares rigurosos, y no es un sistema de gran magnitud, considerando el espacio físico y los servicios que ofrece el hotel.

V. Conclusiones y Recomendaciones

A. Conclusiones

El desarrollo del sistema web hotelero, utilizando el marco ágil SCRUM, se empezó con la creación del Product Backlog, que contiene una lista priorizada con los requisitos del sistema, en colaboración del Product Owner, la cual sirvió como base para establecer los 4 Sprints. Cada Sprint tuvo un objetivo específico con sus respectivos entregables. Durante el Sprint Planning se desglosaron las tareas en subtareas y, mediante el Sprint Length, se calculó el tiempo necesario para completar cada Sprint. En el Sprint Backlog, se llevó el registro de cada tarea con su respectivo Burndown Chart. Al final de cada Sprint, se realizó el Sprint Review que consistió en una evaluación reflexiva tanto de los procesos como de los resultados obtenidos. El proceso culminó con la entrega del producto final con él un sistema web de gestión hotelero cumpliendo con todos los requisitos especificados al inicio. La implementación del marco ágil SCRUM fue de gran ayuda para facilitar la entrega del producto final, logrando un menor esfuerzo y permitiendo una mayor disponibilidad del equipo para analizar posibles mejoras sustanciales.

Se desarrolló un sistema web para el Hotel Puerta de Oro utilizando la norma ISO 29110 parte 5-1-2 iniciando con el plan de proyecto en el cual se definió el tiempo de desarrollo, luego se realizó el levantamiento de requisitos y el diseño arquitectónico del software. En cuanto a la construcción del software se definió un tiempo de ejecución mayormente largo comparado con el resto de los objetivos y así se logró un sistema completo que implementó todos los requerimientos. El proceso culminó con la entrega del sistema web realizado, implementando la norma ISO 29110, que requirió 10 documentos detallados y validados, además de la aplicación de 7 roles con responsabilidades claras de ejecución en el proyecto logrando un proceso más demorado y costoso.

De acuerdo con los resultados de la comparación, en la calidad del software, se determinó que el sistema hecho con SCRUM tiene mayor fiabilidad, en comparación con el sistema hecho con la norma ISO, con una calificación de A en el Back-end y B en el Front-end para el Sistema SCRUM y una calificación de C para el Back-end y Front-end en el sistema ISO. En términos de mantenibilidad, SCRUM tuvo una calificación de A en el Back-end y Front-end y la ISO una calificación de A en el Back-end y D en el Front-end. La duplicidad de código en el Back-end también se señaló como un área que requiere atención, con un puntaje de 1,6% en el Back-end y 5,2% en el Front-end para el sistema SCRUM y 8,5% en el Back-end del sistema con la ISO. Además, se obtuvo que en la seguridad ambos sistemas tuvieron una calificación de A, y en la

cobertura, tuvieron 0% respectivamente. Para los puntos de acceso ambos sistemas por parte del Front-end se tuvo una calificación del A y en el Back-end una de E.

En cuanto a la documentación, se observó que la metodología SCRUM genera menos documentación (6 documentos), ya que prioriza el producto final, mientras que la norma ISO 29110 parte 5-1-2 requiere documentación exhaustiva y detallada (10 documentos) a lo largo de todo el proceso. Por otro lado, en cuanto a la asignación de roles, se obtuvo que la norma ISO 29110, parte 5-1-2 tiene un mayor número de roles responsables (7 roles) para los procesos, en comparación con el marco ágil SCRUM (3 roles), y en este caso específico, SCRUM es el enfoque que más se adaptaría al sistema web del Hotel Puerta de Oro, debido a que no es un sistema de mayor magnitud, y no requiere una documentación exhaustiva y un equipo de desarrollo comprendido por 7 roles.

En base a los resultados de la encuesta de satisfacción llevada a cabo por las 4 personas que trabajan en el Hotel Puerta de Oro, se concluyó que el sistema con la ISO obtuvo un puntaje mayor que el sistema SCRUM, en cuanto a la Satisfacción General y Usabilidad del Sistema, sin embargo, ambos proyectos cumplieron los requisitos inicialmente establecidos evaluados en la encuesta.

B. Recomendaciones

Dado que tanto SCRUM como la norma ISO 29110 tienen fortalezas y debilidades específicas, se recomienda considerar la posibilidad de combinar elementos de ambas metodologías en futuros proyectos, por ejemplo, podría explorarse la posibilidad de utilizar SCRUM para el desarrollo iterativo y ágil del producto, mientras se sigue un marco de referencia ISO para garantizar la documentación y calidad del proceso.

Es esencial proporcionar capacitación y formación adecuadas al equipo de desarrollo en las metodologías seleccionadas, esto incluye, no solo la comprensión de los principios y prácticas de SCRUM y la norma ISO 29110, sino también su aplicación efectiva en el contexto específico del sistema de gestión hotelera.

Teniendo en consideración que la norma ISO 29110 requiere una documentación más exhaustiva, se sugiere implementar herramientas y procesos adecuados para gestionar y mantener esta documentación de manera eficiente, esto incluye el uso de sistemas de gestión de documentos y la asignación clara de responsabilidades para su creación y actualización.

Basándonos en los resultados de este estudio, para futuros proyectos de similar magnitud al del Hotel Puerta de Oro, se recomienda el uso del marco ágil SCRUM. Esto se debe a que la

norma ISO 29110 parte 5-1-2, aunque esta diseñada para VSE requiere una documentación más exhaustiva y el manejo de más personal con diferentes roles lo que la convierte en una alternativa más costosa.

Referencias

- [1] L. A. Castillo Salinas, S. Sanchez Gordon, J. Villarroel Ramos, Sanchez Gordon Ma y M. Sanchez Gordon, «Evaluation of the implementation of a subset of ISO/IEC 29110 Software Implementation process in four teams of undergraduate students of Ecuador. An empirical software engineering experiment,» *Computer Standards & Interfaces*, vol. 70, p. 103430, 2020.
- [2] V. J. Aliaga Cruz, «Implementación del módulo de compras del sistema “lamb-financial” de la asociación iglesia adventista del séptimo día peruana del norte para el cumplimiento del tiempo y alcance, basado en la ISO 29110 y el marco de trabajo SCRUM,» *Universidad Peruana Unión*, 2019.
- [3] L. M. Gaitán gonzalez, «Uso de la metodología SCRUM en el desarrollo de aplicaciones web,» *Universidad de San Carlos de Guatemala*, 2017.
- [4] E. J. Coaquira Vasco, «Sistema web de reserva de hoteles caso: Agencia de viajes definitely bolivia».
- [5] ISO/IEC, «Software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile,» *ISO/IEC TR 29110-5-1-2:2011*, 2011.
- [6] M. Morales Trujillo, H. Oktaba, T. Ventura y R. Torres, «From moprosoft level 2 to ISO 29110 basic profile: Bridging the gap,» *CLEI Electronic Journal*, vol. 16, n° 1, pp. 3--3, 2013.
- [7] K. Schwaber y J. Sutherland, «La guía definitiva de SCRUM: Las reglas del juego,» 2020.
- [8] «Productiviza,» 2023. [En línea]. Available: <https://www.productiviza.com/wp-content/uploads/2023/06/ejemplo-de-Scrum-1024x576.png>. [Último acceso: 06 03 2024].
- [9] M. Harmerník, «Development of modern user interfaces in angular framework,» *Brno*, 2020.
- [10] P. Webb, D. Syer, J. Long, S. Nicoll, R. Wich, A. Wilkinson, M. Overdijk, C. Dupuis y S. Deleuze, «Spring boot reference guide,» *Part IV. Spring Boot features*, vol. 24, 2013.
- [11] M. Bean, *Laravel 5 essentials*, Packt Publishing Ltd, 2015.
- [12] D. J. Ortega, O. Asesor y M. A. Pérez, «Bootstrap y laravel, herramientas para el desarrollo de aplicaciones web,» *UNIVERSIDAD POLITÉCNICA DE SINALOA, Mexico*, 2019.

- [13] J. M. Lozano Banqueri, «Creación y gestión de una base de datos con mysql y phpmyadmin,» *Jaén: Universidad de Jaén*, 2018.
- [14] H. A. Baca, F. d. L. P. Valdivia y I. S. Solís, *Introducción a las Bases de Datos con MySQL*, Herwin Alayn Huillcen Baca, 2022.
- [15] P. López Herrera, «Comparación del desempeño de los sistemas gestores de bases de datos mysql y postgresql,» *Universidad Autónoma del Estado de México*, 2016.
- [16] Javier, «Formadores IT,» 24 08 2023. [En línea]. Available: <https://formadoresit.es/que-es-postman-cuales-son-sus-principales-ventajas/>.
- [17] Z. Powell, «CodeIgniter vs Laravel: Una Comparación Detallada,» Kinsta, 23 08 2023. [En línea]. Available: <https://kinsta.com/wp-content/uploads/2023/01/laravel-mvc-architecture.jpg>. [Último acceso: 06 03 2024].