



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE SOFTWARE

Tema:

**APLICACIÓN WEB PARA LA GESTIÓN Y CONTROL DE LA
INFORMACIÓN LABORAL DEL EQUIPO DE DESARROLLO DE LA
EMPRESA SOFTWARE EVOLUTIVO**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la
obtención del Título de Ingeniero de Software

ÁREA: Software

LÍNEA DE INVESTIGACIÓN: Desarrollo de Software

AUTOR: Joshua Mateo Martínez Urbina

TUTOR: Ing. Oscar Fernando Ibarra Torres Mg.

Ambato – Ecuador

febrero – 2024

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: **APLICACIÓN WEB PARA LA GESTIÓN Y CONTROL DE LA INFORMACIÓN LABORAL DEL EQUIPO DE DESARROLLO DE LA EMPRESA SOFTWARE EVOLUTIVO**, desarrollado bajo la modalidad Proyecto de Investigación por el señor Joshua Mateo Martínez Urbina, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, febrero 2024.

Ing. Oscar Fernando Ibarra Torres, Mg.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: APLICACIÓN WEB PARA LA GESTIÓN Y CONTROL DE LA INFORMACIÓN LABORAL DEL EQUIPO DE DESARROLLO DE LA EMPRESA SOFTWARE EVOLUTIVO, es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2024.

Joshua Mateo Martínez Urbina

C.C. 1805776208

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, febrero 2024.

Joshua Mateo Martínez Urbina
C.C. 1805776208
AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Joshua Mateo Martínez Urbina, estudiante de la Carrera de Software de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado APLICACIÓN WEB PARA LA GESTIÓN Y CONTROL DE LA INFORMACIÓN LABORAL DEL EQUIPO DE DESARROLLO DE LA EMPRESA SOFTWARE EVOLUTIVO, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, febrero 2024.

Ing. Elsa Pilar Urrutia Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL

Ing. Hernán Fabricio Naranjo Ávalos
PROFESOR CALIFICADOR

Ing. Rubén Eduardo Nogales Portero
PROFESOR CALIFICADOR

DEDICATORIA

Este proyecto de investigación está dedicado a mi madre y mi orgullo, Silvana. Incontables ocasiones he caído, pero su amor incondicional me ha levantado en todas y cada una de ellas.

A mi hermana Daniela y a mi abuelo Polito, quienes han sido mi motor y mi apoyo durante este difícil proceso.

Una mención especial a mi abuela Fanny, sé que desde el cielo estará orgullosa de mí.

Lo logramos.

Joshua Mateo Martínez Urbina

AGRADECIMIENTO

Quiero agradecer primero a Dios, por ponerme en este camino junto a las personas correctas, por darme la sabiduría y la fuerza para sobrellevar todas las adversidades.

A mi madre Silvana, a mi hermana Daniela y a mi abuelo Polito gracias por nunca dudar de mí, por siempre ayudarme con todo lo que está a su alcance y por ir de mi mano en todo este camino.

Agradezco infinitamente a mi novia Nicole y a toda mi familia, por todo su amor y por estar a mi lado cuando más les he necesitado.

Muchas gracias a todos mis profesores por cada valiosa enseñanza, en especial a mi tutor y guía Ing. Oscar Ibarra por su tiempo y paciencia durante el desarrollo de este proyecto de investigación.

Un agradecimiento enorme al Ing. Jorge Aguilar e Ing. Fernando Paz, quienes han sido más que mis jefes de trabajo. Por haberme dado su voto de confianza y recibirme cuando aún estaba formándome como profesional. Sin duda su orientación y sus enseñanzas me ayudan a crecer día tras día.

A mis amigos de la vida, de la universidad y del trabajo con quienes he compartido muchos momentos inolvidables.

Joshua Mateo Martínez Urbina

ÍNDICE GENERAL DE CONTENIDOS

| | |
|---|--------------|
| PORTADA | i |
| APROBACIÓN DEL TUTOR | ii |
| AUTORÍA | iii |
| DERECHOS DE AUTOR | iv |
| APROBACIÓN TRIBUNAL DE GRADO | v |
| DEDICATORIA | vi |
| AGRADECIMIENTO | vii |
| ÍNDICE GENERAL DE CONTENIDOS | viii |
| ÍNDICE DE FIGURAS | xi |
| ÍNDICE DE TABLAS | xvi |
| ÍNDICE DE ANEXOS | xviii |
| RESUMEN EJECUTIVO | xix |
| ABSTRACT | xx |
| CAPÍTULO I. MARCO TEÓRICO | 21 |
| 1.1. Tema de investigación | 21 |
| 1.1.1. Planteamiento del problema | 21 |
| 1.2. Antecedentes investigativos | 22 |
| 1.3. Fundamentación teórica | 24 |
| 1.3.1. Operación | 24 |
| 1.3.2. Gestión de operaciones | 24 |
| 1.3.3. Equipo de desarrollo | 25 |
| 1.3.4. Líder de proyecto | 25 |
| 1.3.5. Sistema de información | 26 |

| | |
|---|-----------|
| 1.3.6. Aplicación web..... | 26 |
| 1.3.7. Framework | 28 |
| 1.3.8. Frameworks para aplicaciones web | 29 |
| 1.3.9. Sistema de gestión y control | 31 |
| 1.3.10. Base de datos..... | 32 |
| 1.3.11. Base de datos relacional | 32 |
| 1.3.12. Base de datos no relacional | 33 |
| 1.3.13. Sistema de Gestión de Base de Datos | 33 |
| 1.3.14. JHipster | 33 |
| 1.3.15. Metodologías ágiles | 34 |
| 1.3.16. Metodologías híbridas | 34 |
| 1.4. Objetivos | 35 |
| 1.4.1. Objetivo general | 35 |
| 1.4.2. Objetivos específicos | 35 |
| CAPÍTULO II. METODOLOGÍA..... | 36 |
| 2.1. Materiales | 36 |
| 2.2. Métodos..... | 36 |
| 2.2.1. Modalidad de la investigación | 36 |
| 2.2.2. Población y muestra | 37 |
| 2.2.3. Recolección de información..... | 37 |
| 2.2.4. Procesamiento y análisis de datos | 52 |
| CAPÍTULO III. RESULTADOS Y DISCUSIÓN..... | 55 |
| 3.1. Análisis y discusión de los resultados..... | 55 |
| 3.1.1. Diagrama del proceso de gestión con el sistema..... | 55 |
| 3.1.2. Análisis y selección de la metodología de desarrollo | 56 |

| | |
|--|------------|
| 3.1.3. Análisis y selección del framework para el desarrollo de la aplicación web... | 58 |
| 3.2. Desarrollo de la propuesta..... | 61 |
| 3.2.1. Definición de roles | 62 |
| 3.2.2. Recolección de requisitos..... | 62 |
| 3.2.3. Fase I: Planificación..... | 63 |
| 3.2.4. Fase II: Diseño | 69 |
| 3.2.5. Fase III: Desarrollo | 92 |
| 3.2.6. Fase IV: Pruebas | 125 |
| 3.3. Resultados | 134 |
| 3.3.1. Modelo TAM | 134 |
| CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES | 139 |
| 4.1. Conclusiones | 139 |
| 4.2. Recomendaciones..... | 141 |
| REFERENCIAS BIBLIOGRÁFICAS | 142 |
| ANEXOS | 147 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1. Diagrama del proceso de gestión con la aplicación web | 56 |
| Figura 2. Diagrama Entidad Relación | 79 |
| Figura 3. Arquitectura de la aplicación web | 81 |
| Figura 4. Prototipo de la página de inicio de la aplicación web | 82 |
| Figura 5. Prototipo de la interfaz de autenticación de usuarios | 82 |
| Figura 6. Prototipo de la interfaz de gestión de usuarios | 83 |
| Figura 7. Prototipo de la interfaz de creación o edición de usuarios | 83 |
| Figura 8. Prototipo de la interfaz de actualización de datos personales de usuario ... | 84 |
| Figura 9. Prototipo de la interfaz de actualización de contraseña de usuarios..... | 84 |
| Figura 10. Prototipo de la interfaz de gestión de clientes | 85 |
| Figura 11. Prototipo de la interfaz de creación o edición de clientes | 85 |
| Figura 12. Prototipo de la interfaz de gestión de proyectos..... | 86 |
| Figura 13. Prototipo de la interfaz de creación o edición de proyectos | 86 |
| Figura 14. Prototipo de la interfaz de gestión de Actividades | 87 |
| Figura 15. Prototipo de la interfaz de creación o edición de Actividades..... | 87 |
| Figura 16. Prototipo de la interfaz de generación de reportes para Desarrollador..... | 88 |
| Figura 17. Prototipo de la interfaz de vista de Actividades para Líder de Proyecto.. | 89 |
| Figura 18. Prototipo de la interfaz de registro de observaciones | 89 |
| Figura 19. Prototipo de la interfaz de gestión de las observaciones | 90 |

| | |
|--|----|
| Figura 20. Prototipo de la interfaz de edición de observaciones..... | 91 |
| Figura 21. Prototipo de la interfaz de generación de reportes..... | 91 |
| Figura 22. Comando para crear el proyecto con Khipster | 92 |
| Figura 23. Asistente de KHipster para la creación del proyecto base..... | 92 |
| Figura 24. Pantalla de bienvenida a la aplicación web | 93 |
| Figura 25. Interfaz de inicio de sesión | 93 |
| Figura 26. Método para el inicio de sesión | 94 |
| Figura 27. Método de validación de rol Desarrollador | 94 |
| Figura 28. Método de validación de rol Líder de Proyecto..... | 95 |
| Figura 29. Método de validación de rol Gerente de Operaciones..... | 95 |
| Figura 30. Método de validación de rol Administrador | 96 |
| Figura 31. Método POST para la creación de un nuevo registro de una entidad..... | 96 |
| Figura 32. Capa de lógica de negocio para crear un nuevo registro de una entidad.. | 97 |
| Figura 33. Método PUT para la actualización de un registro de una entidad. | 97 |
| Figura 34. Capa de lógica de negocio para actualizar un registro de una entidad. | 98 |
| Figura 35. Método GET para la obtención de los registros de la entidad..... | 98 |
| Figura 36. Capa de lógica de negocio para agregar criterios de búsqueda de registros. | 98 |
| Figura 37. Capa de lógica de negocio para la obtención de registros de una entidad. | 99 |
| Figura 38. Método GET para la obtención de un registro de una entidad. | 99 |

| | |
|--|-----|
| Figura 39. Capa de lógica de negocio para obtener un registro de una entidad..... | 99 |
| Figura 40. Método DELETE para la eliminación de un registro de una entidad..... | 100 |
| Figura 41. Capa de lógica de negocio para la eliminación de un registro de una entidad. | 100 |
| Figura 42. Interfaz de visualización general de los usuarios | 101 |
| Figura 43. Opción de usuario desactivado | 101 |
| Figura 44. Interfaz de creación o edición de usuarios..... | 102 |
| Figura 45. Interfaz de confirmación de eliminación de usuarios | 102 |
| Figura 46. Interfaz de actualización de datos personales para usuarios..... | 103 |
| Figura 47. Interfaz de actualización de contraseña para usuarios..... | 103 |
| Figura 48. Método de la actualización de contraseña | 104 |
| Figura 49. Interfaz de visualización general de Clientes para Gerente de Operaciones | 105 |
| Figura 50. Interfaz del detalle de Clientes para Gerente de Operaciones | 105 |
| Figura 51. Interfaz de creación o edición de Clientes para Gerente de Operaciones | 106 |
| Figura 52. Interfaz de confirmación de borrado de un Cliente | 106 |
| Figura 53. Interfaz de visualización general de Proyectos para Líderes de Proyecto | 107 |
| Figura 54. Interfaz del detalle del Proyecto para Líderes de Proyecto | 107 |
| Figura 55. Interfaz de creación o edición de Proyectos para Líderes de Proyecto .. | 108 |
| Figura 56. Interfaz de confirmación de eliminación de Proyecto | 108 |

| | |
|--|-----|
| Figura 57. Interfaz de visualización general de Actividades para Desarrolladores . | 109 |
| Figura 58. Interfaz del detalle de Actividades para Desarrolladores | 109 |
| Figura 59. Interfaz de creación o edición de Actividades para Desarrolladores..... | 110 |
| Figura 60. Interfaz de confirmación de borrado de una Actividad | 110 |
| Figura 61. Interfaz del detalle del proyecto al que pertenece la Actividad..... | 111 |
| Figura 62. Interfaz de visualización general de Actividades para Líderes de Proyecto | 111 |
| Figura 63. Método GET para obtener las actividades de interés de un Líder de Proyecto | 112 |
| Figura 64. Capa de lógica de negocio para obtener las actividades de interés de un Líder de Proyecto | 112 |
| Figura 65. Interfaz del detalle de la Actividad para Líderes de Proyecto | 113 |
| Figura 66. Interfaz de creación o edición de Observaciones para Líderes de Proyecto | 114 |
| Figura 67. Notificación mediante correo electrónico de nueva Observación | 114 |
| Figura 68. Método para el envío de correo electrónico al crear una Observación... | 115 |
| Figura 69. Interfaz de visualización de Observaciones y edición de Actividad para Desarrollador..... | 116 |
| Figura 70. Interfaz de confirmación de resolución de Observación para Desarrollador | 116 |
| Figura 71. Interfaz de seguimiento de Actividades para Desarrolladores | 117 |
| Figura 72. Método GET para obtener las actividades con Observaciones de un Desarrollador..... | 117 |

| | |
|--|-----|
| Figura 73. Capa de lógica de negocio para obtener las actividades con Observación de un Desarrollador..... | 118 |
| Figura 74. Interfaz de visualización general de Observaciones para Líder de Proyecto | 118 |
| Figura 75. Interfaz de confirmación de Proyecto listo para procesar | 119 |
| Figura 76. Notificación mediante correo electrónico de Proyecto listo para procesar | 119 |
| Figura 77. Método para el envío de correo electrónico de Proyecto listo para procesar | 120 |
| Figura 78. Interfaz para la generación de reportes para Gerente de Operaciones.... | 121 |
| Figura 79. Interfaz de reportes al no encontrar datos para el Gerente de Operaciones | 121 |
| Figura 80. Método para la generación de reporte con JasperReport para Gerente de Operaciones..... | 122 |
| Figura 81. Consulta en PostgreSQL para el reporte para Gerente de Operaciones . | 122 |
| Figura 82. Interfaz para la generación de reportes para Desarrolladores..... | 123 |
| Figura 83. Interfaz de reportes al no encontrar datos para el Desarrollador | 123 |
| Figura 84. Método para la generación de reporte con JasperReport para Desarrollador | 124 |
| Figura 85. Consulta en PostgreSQL para el reporte para Desarrollador..... | 125 |
| Figura 86. Resultados de TAM - Utilidad percibida..... | 136 |
| Figura 87. Resultados de TAM - Facilidad de uso..... | 138 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 1. Población de estudio | 37 |
| Tabla 2. Resultados de la entrevista al Gerente de Operaciones..... | 38 |
| Tabla 3. Resultados de la entrevista al Líder de Proyecto N° 1 | 40 |
| Tabla 4. Resultados de la entrevista al Líder de Proyecto N° 2 | 42 |
| Tabla 5. Resultados de la entrevista al Desarrollador de Software N° 1..... | 44 |
| Tabla 6. Resultados de la entrevista al Desarrollador de Software N° 2..... | 45 |
| Tabla 7. Resultados de la entrevista al Desarrollador de Software N° 3..... | 48 |
| Tabla 8. Resultados de la entrevista al Desarrollador de Software N° 4..... | 49 |
| Tabla 9. Resultados de la entrevista al Desarrollador de Software N° 5..... | 51 |
| Tabla 10. Tabla comparativa de las metodologías de desarrollo | 56 |
| Tabla 11. Tabla comparativa de frameworks para aplicaciones web..... | 59 |
| Tabla 12. Definición de roles | 62 |
| Tabla 13. Tarjetas CRC - Autenticación de usuario | 69 |
| Tabla 14. Tarjetas CRC - Crear usuario..... | 70 |
| Tabla 15. Tarjetas CRC - Editar datos de usuario..... | 70 |
| Tabla 16. Tarjetas CRC - Cambiar contraseña..... | 71 |
| Tabla 17. Tarjetas CRC - Gestionar cliente | 71 |
| Tabla 18. Tarjetas CRC - Gestionar proyecto..... | 71 |
| Tabla 19. Tarjetas CRC - Gestionar actividad | 72 |

| | |
|---|-----|
| Tabla 20. Tarjetas CRC - Generar reportes de desarrollador..... | 72 |
| Tabla 21. Tarjetas CRC - Visualizar actividades | 73 |
| Tabla 22. Tarjetas CRC - Registrar observación | 73 |
| Tabla 23. Tarjetas CRC - Editar observación | 73 |
| Tabla 24. Tarjetas CRC - Notificar información de proyecto..... | 74 |
| Tabla 25. Tarjetas CRC - Generar reportes de cliente | 74 |
| Tabla 26. Tabla de requerimientos funcionales | 75 |
| Tabla 27. Tabla de requerimientos no funcionales | 77 |
| Tabla 28. Tabla de pruebas de aceptación de la aplicación web..... | 125 |
| Tabla 29. Tabla de pruebas de usabilidad de la aplicación web | 133 |
| Tabla 30. Cuestionario de evaluación TAM | 134 |
| Tabla 31. Tabla de frecuencias de TAM - Utilidad percibida..... | 136 |
| Tabla 32. Tabla de frecuencias de TAM - Facilidad de uso | 137 |

ÍNDICE DE ANEXOS

| | |
|---|-----|
| Anexo A. Guía de entrevista para Gerente de Operaciones..... | 147 |
| Anexo B. Guía de entrevista para Líderes de Proyecto | 149 |
| Anexo C. Guía de entrevista para Desarrolladores de Software | 151 |
| Anexo D. Test de integración del método para Inicio de Sesión..... | 153 |
| Anexo E. Test de integración del método de validación de rol Desarrollador | 156 |
| Anexo F. Test de integración del método de validación de rol Líder de Proyecto .. | 157 |
| Anexo G. Test de integración del método de validación de rol Gerente de Operaciones | 158 |
| Anexo H. Test de integración del método de validación de rol Administrador..... | 159 |
| Anexo I. Tests de integración del proceso de creación de un registro de una entidad | 160 |
| Anexo J. Tests de integración del proceso de actualización de un registro de una entidad..... | 162 |
| Anexo K. Test de integración del proceso de obtención de los registros de una entidad | 164 |
| Anexo L. Tests de integración del proceso de obtención de un registro de una entidad | 165 |
| Anexo M. Test de integración del proceso de eliminación de un registro..... | 166 |
| Anexo N. Template del envío de correo de Nueva Observación..... | 167 |
| Anexo O. Template del envío de correo de Proyecto listo para procesar | 168 |
| Anexo P. Generación de reportes para Gerente de Operaciones en formato PDF... | 169 |
| Anexo Q. Generación de reportes para Desarrolladores en formato PDF | 170 |
| Anexo R. Manual de usuario..... | 171 |

RESUMEN EJECUTIVO

El proceso de Gestión y Control de la Información Laboral del equipo de desarrollo de la empresa Software Evolutivo se trata de un proceso interno de operaciones en el cual se registra y recopila la información de las actividades diarias de los desarrolladores de la empresa. Esta información es manipulada de manera manual, haciendo que este proceso sea tedioso e ineficiente, consumiendo recursos de tiempo de los colaboradores de la empresa. El propósito de este proyecto de investigación fue proponer la implantación de una aplicación web que reemplace este proceso manual. Se usaron técnicas de recolección de datos como las entrevistas, para identificar los problemas y a su vez conocer las necesidades puntuales del cliente en cuanto a este proceso interno. Como metodología ágil de desarrollo de software Extreme Programming que permite la constante comunicación y retroalimentación con el cliente y a la vez un eficiente desarrollo de software basado en la entrega frecuente de valor. Para el desarrollo de la aplicación web se usó el framework JHipster que permite desarrollar aplicaciones monolíticas con las tecnologías en tendencia más actuales, en este caso se integró React con Spring Boot en lenguaje Kotlin en una aplicación contenerizada para facilitar su escalabilidad. Para garantizar el correcto funcionamiento de la aplicación web se implantaron tests de integración y para el cumplimiento de los requerimientos funcionales y no funcionales definidos con el cliente se realizaron pruebas de aceptación y de usabilidad de la aplicación web.

Palabras clave: Información laboral, operaciones, aplicación web, Extreme Programming, JHipster.

ABSTRACT

The process of Labor Information Management and Control of the Software Evolutivo development team is an internal operations procedure that records and compiles daily activities of the company's developers. This manual manipulation of information makes the process tedious and inefficient, consuming valuable time resources of the company's collaborators. The purpose of this research project was to propose the implementation of a web application to replace this manual process. Data collection techniques, such as interviews, were applied to identify issues and collect the specific needs of the client regarding this internal process. The chosen agile software development methodology was Extreme Programming, facilitating constant communication and feedback with the client and ensuring efficient software development based on frequent value delivery. The web application was developed using the JHipster framework, enabling the creation of monolithic applications with the latest trending technologies. In this case, React was integrated with Spring Boot using Kotlin language in a containerized application for scalability. To ensure the proper functioning of the web application, integration tests were implemented. Additionally, acceptance and usability tests were conducted to validate compliance with the defined functional and non-functional requirements with the client.

Keywords: Labor Information, operations, web application, Extreme Programming, JHipster.

CAPÍTULO I. MARCO TEÓRICO

1.1. Tema de investigación

APLICACIÓN WEB PARA LA GESTIÓN Y CONTROL DE LA INFORMACIÓN LABORAL DEL EQUIPO DE DESARROLLO DE LA EMPRESA SOFTWARE EVOLUTIVO.

1.1.1. Planteamiento del problema

La gestión de la información laboral es uno de los aspectos fundamentales para la organización operacional de las empresas de consultoría de software. Por este motivo las organizaciones buscan herramientas que les permitan manejar dicha información de la manera más eficaz y eficiente [1].

Mediante una buena gestión de la información laboral se puede lograr optimizar recursos, mejorar la eficiencia operativa y tomar decisiones informadas. Esta gestión eficaz no solo impacta positivamente en el área de operaciones, sino que fortalece la capacidad de la empresa de ejecutar procesos administrativos correctamente organizados.

Actualmente, los encargados de gestionar y controlar la información laboral dentro de una empresa de software se denominan gerentes de operaciones, el proceso de validación de la información laboral está respaldado por los líderes de proyectos de software quienes supervisan los datos de las actividades que diariamente se llevan a cabo por los desarrolladores. Debido a la falta de herramientas dedicadas para este proceso, la información se registra de manera manual y dispersa, dificultando el cumplimiento de las operaciones e impidiendo la supervisión y visibilidad sobre el progreso de los proyectos existentes [2].

Teniendo en cuenta esta problemática, podemos observar que existen empresas de consultoría de software que, pese a estar en contacto continuo con tecnologías de información y comunicación no tienen un manejo adecuado de la información laboral,

es decir, de las actividades realizadas, horas invertidas a clientes y a proyectos. La información laboral se encuentra registrada de manera manual en herramientas que no ofrecen a las empresas las características adecuadas que permitan el manejo de la información de manera integral, la asignación óptima de recursos, la generación de reportes precisos sobre la productividad del equipo de desarrollo.

En la empresa Software Evolutivo el registro, análisis, y extracción de la información laboral es tediosa, ineficiente y es la causa del retraso de los procesos del área de operaciones, esto genera que no se brinde a los clientes potenciales de la empresa la mejor calidad del servicio.

1.2. Antecedentes investigativos

Revisando fuentes bibliográficas sobre proyectos y tesis en repositorios de universidades del Ecuador con ayuda de buscadores adecuados como DSpace, se encontraron tesis relacionadas a la gestión y control de diferentes tipos de información laboral del personal de las empresas.

Evelyn Moya y Lia Callejas [3] en su tesis ‘Análisis diseño e implementación de un sistema de gestión de Talento Humano en la comunidad Salesiana San Juan Bosco’, señalan que antes de la implementación del sistema, el registro de datos personales de los empleados se realizaba mediante hojas de Excel o registros físicos, lo cual significa correr riesgos de extravío o eliminación. Las autoras destacan que la introducción de un software que centraliza toda la información permite mantener un registro confiable, actualizado y seguro.

Anthony Tacuri y Elvis Pérez [4] en su tesis ‘Desarrollo de aplicación web y móvil para control de operaciones eléctricas orientado a trabajadores de empresa Rimaelec’, mencionan que las metodologías de desarrollo de software ágiles permiten alta flexibilidad y agilidad a los equipos de trabajo, permitiéndoles ser más productivos y eficientes. Los autores justifican la elección de una metodología ágil para el desarrollo de su proyecto de investigación debido a la necesidad de entregar valor y ofrecer resultados de calidad en un tiempo determinado.

Luis Taco [5] en su tesis ‘Desarrollo de un sistema móvil de gestión del talento humano para empresas de seguridad’, para la construcción del sistema propone la metodología XP, que se adapta a las necesidades de los proyectos que requieran una rápida implementación. Sobre la programación en parejas que sugiere XP, el autor destaca la adaptabilidad de esta herramienta al permitir que la programación puede ser llevada por un desarrollador. Esta flexibilidad se consigue gracias a la capacidad de adaptación de XP, que permite que un programador realice las actividades del ciclo del software, aprovechando los beneficios y las prácticas ágiles que promueve la metodología. Además, el autor propone el uso de las historias de usuario para recoger adecuadamente los requerimientos del usuario final.

Cindy Padilla y César Quispe [6] en su tesis titulada ‘Sistema web para la gestión y retención de talento de una empresa outsourcing de TI basado en el aprendizaje automático’, concluyen que las buenas prácticas en los dominios de gestión de fuerza laboral y arquitectura de retención del talento son de alta importancia y representan un grado crítico para la organización. Por esta razón, desarrollaron un modelo predictivo de renuncia con algoritmo de regresión logística que redujo significativamente el tiempo de análisis de renuncias e incrementó la cantidad de retenciones en la empresa. Para lograrlo, optaron por el framework Spring Boot para agilizar las etapas de construcción del sistema, sin incurrir en costos adicionales en la implementación de la solución.

Francisco Cerna [7] en su tesis titulada ‘Sistema WebVR para el proceso de evaluación psicológica en una consultora de talento de recursos humanos en Lima 2019’, menciona que la implementación del sistema ha llevado a una reducción del 100% en las horas/hombre que el reclutador dedicaba a la corrección de pruebas psicológicas. Este cambio ha disminuido el tiempo empleado por candidato de una hora a fracciones de segundo, equivalente a un ahorro de 3 horas por proceso de evaluación. Es importante destacar que la implementación se llevó a cabo utilizando React en conjunto con Spring Boot, resaltando la separación entre el frontend y el backend, lo cual ha facilitado el desarrollo y la mantenibilidad del sistema.

Cristian Gómez y Erick Pulido [8] en su tesis titulada ‘Diseño e implementación de un sistema web para categorización docente de la institución universitaria Antonio José Camacho (CATEDOC)’, optaron por una arquitectura mixta orientada a microservicios y multicapa, donde cada aplicación se encuentra en su propio contenedor. La arquitectura se compone de las siguientes capas: presentación, lógica, servicios REST y datos. Además, el autor destaca que la capa de servicios es versátil, ya que puede ser consumida por diferentes tecnologías como Angular, React para web y tecnologías móviles Android o IOS en la capa de presentación.

Marco Bermeo, Francis Buele y Galo Hurtado [9] en su tesis ‘Estudio comparativo para la toma de requerimientos de software para el desarrollo de una aplicación web en investigación psicológica del bullying educativo’, después de un análisis exhaustivo entre tecnologías backend como: Python, JavaScript, Java y Kotlin, optaron por el lenguaje Kotlin implementado en el framework Spring Boot. Eligieron esta combinación por sus características destacadas en rendimiento, facilidad de aprendizaje, capacidad de escalar a gran número de clientes con requisitos de hardware moderno y garantías de seguridad para el tratamiento de la información almacenada.

1.3. Fundamentación teórica

1.3.1. Operación

Una Operación empresarial es el conjunto de todas las actividades y procesos que realizan las diferentes áreas con el objetivo de generar bienes, productos y servicios que posteriormente se prestarán a los usuarios. Estas operaciones pueden ser tanto internas como externas, varían dependiendo del tipo de negocio, pero todas y cada una de ellas son fundamentales para poder mantener a las organizaciones funcionando correctamente [10].

1.3.2. Gestión de operaciones

Dentro de las empresas, la Gestión de las Operaciones son un conjunto de actividades que abarcan la planificación, ejecución y supervisión de las acciones que se lleven a

cabo internamente. El propósito de la Gestión de las Operaciones es potenciar el rendimiento del personal, administrar de manera eficiente los recursos y aumentar la productividad en cada área de la empresa. La mejora continua de los procesos y de las capacidades de los empleados se ve reflejada en la calidad de los servicios que la empresa ofrece a sus consumidores [11].

1.3.3. Equipo de desarrollo

Al hablar de un Equipo de Desarrollo de Software, nos referimos a las personas que trabajan en la parte técnica de un proyecto o producto. Cada uno de los miembros tiene su rol y sus responsabilidades, por lo tanto, es crucial que la organización del equipo sea la adecuada y que exista colaboración y comunicación activa, así, el rendimiento individual en cada actividad diaria de los desarrolladores contribuirá al rendimiento global del equipo [12].

Entre las características de un buen Equipo de Desarrollo encontramos:

- Colaboración continua entre los miembros del equipo.
- Cada miembro del equipo cuenta con los conocimientos y habilidades adecuadas para cumplir con sus responsabilidades.
- La moral y el ánimo del equipo es fundamental, los buenos equipos celebran cada pequeño hito.
- Los líderes de proyecto supervisan continuamente el progreso del proyecto y brindan ayuda a quienes así lo requieran.

1.3.4. Líder de proyecto

Es la persona encargada de asegurar que se cumplan cada uno de los objetivos del desarrollo de un proyecto de software. Desempeña un papel importante en el proceso de la gestión de cambios con el cliente [13].

Las responsabilidades del Líder de Proyecto son:

- Comunicación continua con el cliente con el fin de brindarle apoyo.
- Gestionar los recursos necesarios para el proyecto.
- Controlar y dar seguimiento al cumplimiento de los objetivos del proyecto.
- Fomentar la colaboración y trabajo en equipo.
- Planificar las distintas fases del proyecto.
- Gestionar las restricciones, los riesgos y desarrollar planes de acción durante la ejecución de los proyectos.

1.3.5. Sistema de información

Un sistema de información es un conjunto de componentes, tecnologías (software-hardware), procesos que interconectados buscan un fin común. Dentro de las organizaciones y empresas, estos sistemas adquieren datos, los procesan, guardan y comparten información más elaborada, brindando a las organizaciones la capacidad de mejorar la toma de decisiones, aumentar la eficiencia operativa, proporcionar acceso rápido a información clave, gestionar recursos de manera efectiva, facilitar la comunicación interna y ganar competitividad al adaptarse ágilmente a cualquier tipo de cambio del entorno empresarial [14].

1.3.6. Aplicación web

Las Aplicaciones Web son programas que no requieren ser instalados en tu dispositivo para poder utilizarlo, sino que operan a través de Internet. Esto implica que los datos y archivos con los que trabajas son procesados y almacenados en la web. Las Aplicaciones Web están estrechamente vinculadas al almacenamiento en la nube

debido a que la información se guarda de manera permanente en servidores de Internet y se envían los datos necesarios a tu dispositivo en el momento requerido.

Los servidores de Internet que brindan el alojamiento de estas aplicaciones se encuentran distribuidos alrededor del mundo, lo cual permite que el servicio sea accesible en todo momento y desde cualquier lugar, solo se requiere de una conexión a Internet [15].

Ventajas

- Actualizaciones automáticas (realizadas en el servidor).
- No se requiere de instalación para usarla.
- Acceso desde cualquier dispositivo y sistema operativo.
- Almacenamiento en la nube.
- Colaboración y envío de datos en tiempo real.
- Facilidad de implementación y mantenimiento.
- Escalabilidad.

Desventajas

- Limitación de acceso a recursos del dispositivo.
- Rendimiento ligeramente menor en comparación a aplicaciones nativas.
- Alta dependencia con la conexión a Internet y el proveedor de servicios.
- Son más vulnerables a ciberataques.

Arquitectura de las aplicaciones web

Se basan en una arquitectura de cliente/servidor. Cliente es quien interactúa con el usuario final recibiendo sus solicitudes, las envía al servidor y muestra las respuestas recibidas.

Servidor es el componente que se encarga de recibir, procesar y responder a las solicitudes recibidas del cliente. Si es necesario, el servidor puede acceder a la base de datos.

Esta arquitectura establece comunicación entre las dos partes mediante un protocolo de comunicación, ya sea HTTP (HyperText Transfer Protocol) o HTTPS (HyperText Transfer Protocol Secure) [16].

1.3.7. Framework

Es un marco de trabajo (framework) usado comúnmente por desarrolladores y programadores para cumplir con los objetivos específicos de un proyecto de software. Un Framework ofrece al desarrollador una colección de herramientas, bibliotecas, plugins y plantillas reutilizables, además que cada uno cuenta con una estructura ya definida [17].

El uso de un Framework permite:

- Simplificar tareas y procesos.
- Acelera y facilita el trabajo de desarrollo ya que todo se programa sobre una estructura ya definida por el framework.
- Reduce los errores en el código.
- Aumenta la calidad del producto de software que se obtiene.

1.3.8. Frameworks para aplicaciones web

Las tecnologías están cambiando constantemente y por consiguiente los frameworks para el desarrollo también lo hacen. Actualmente podemos encontrar en tendencia los siguientes frameworks para el desarrollo web [18].

React

React es una librería de JavaScript que brinda las herramientas necesarias para abordar los desafíos de construir interfaces de usuario interactivas y eficientes. Fue desarrollada y utilizada por Facebook para su aplicación web. Se enfoca en la creación de componentes reutilizables que se actualizan de manera eficiente cuando cambian los datos [19].

React se introdujo por primera vez en 2013 y se ha vuelto popular en la comunidad de desarrollo web desde entonces. Ha ganado una gran adopción debido a su eficiencia, su enfoque en la creación de interfaces de usuario declarativas y su capacidad para trabajar con grandes conjuntos de datos en tiempo real. Actualmente es usado en el software de empresas grandes de la industria a nivel mundial, como Facebook, Instagram, Netflix y Airbnb. También cuenta con una comunidad de desarrollo activa que contribuye con bibliotecas y herramientas adicionales para extender su funcionalidad [20].

Angular

Creado por Google en el año 2010 para el desarrollo web. Fue lanzado inicialmente bajo el nombre de AngularJS y luego se lanzó una versión completamente nueva llamada Angular en 2016. Especialmente utilizado para construir aplicaciones web de una sola página (Single Page Applications) y aplicaciones web escalables.

Se basa en TypeScript, un lenguaje de programación que extiende JavaScript y agrega características de tipado estático y programación orientada a objetos. Actualmente, su uso aún es amplio en la industria y es respaldado por una gran comunidad de desarrolladores. Es utilizado por empresas reconocidas como Google, Microsoft, IBM

y muchas otras. Angular ofrece una amplia gama de características y herramientas para el desarrollo web, como enrutamiento, manejo de formularios, pruebas unitarias y más [21].

Vue.js

Framework de JavaScript open source utilizado para desarrollar interfaces de usuario interactivas y dinámicas. Vue.js fue creado por Evan You y su primer release fue en 2014. Se destaca por su simplicidad y su enfoque gradual, lo que significa que puede ser fácilmente adoptado en proyectos existentes sin requerir una reescritura completa.

Utiliza una sintaxis de plantilla declarativa que combina HTML y JavaScript, lo que facilita la creación de componentes reutilizables. Se basa en el paradigma de reactividad, lo que significa que los cambios en los datos se reflejan automáticamente en la interfaz de usuario. El desarrollo de Vue.js fue inspirado por otros frameworks como Angular y React, buscando combinar lo mejor de ambos en una sola herramienta [22].

Django

Django es un framework de desarrollo web de alto nivel basado en Python. Su historia se remonta a la necesidad de construir aplicaciones web de manera rápida y eficiente. Este framework sigue el principio de "baterías incluidas", lo que significa que ofrece una amplia gama de funcionalidades incorporadas y bibliotecas que facilitan el desarrollo de aplicaciones web complejas.

En la actualidad, es ampliamente utilizado en la industria y es elegido por desarrolladores y empresas para construir aplicaciones web de diversos tamaños y complejidades. Es utilizado por empresas grandes como Instagram, Pinterest y Disqus, entre otras [23].

Spring

Framework de desarrollo de aplicaciones web creado por Rod Johnson en 2004 que ha evolucionado hasta convertirse en uno de los frameworks más populares para el

desarrollo de aplicaciones Java. Spring se basa en principios de Inversión de Control (IoC – Inversion of Control) y Programación Orientada a Aspectos (AOP – Aspect-Oriented Programming), lo que permite una mayor modularidad y flexibilidad en el desarrollo de aplicaciones [24].

Se desarrolla principalmente en el lenguaje de programación Java, el cual es un lenguaje ampliamente utilizado en la industria, conocido por su portabilidad, seguridad y capacidad de manejar aplicaciones empresariales escalables [25].

Spring es ampliamente utilizado en el desarrollo de aplicaciones empresariales y es elegido por muchas empresas y desarrolladores. Proporciona una amplia gama de características y módulos que facilitan el desarrollo de aplicaciones robustas y escalables [26].

1.3.9. Sistema de gestión y control

Un Sistema de Gestión y Control es una herramienta ampliamente utilizada por las empresas debido a las características que tiene y a los beneficios que trae a las organizaciones. Estos sistemas permiten administrar y supervisar los recursos, actividades y procesos de las empresas [27]. Puede abarcar diferentes áreas y en base a este criterio podemos encontrar los siguientes sistemas de gestión:

- CRM: Customer Relationship Management (Gestión de Relaciones con Clientes), sistema que ayuda a mejorar la satisfacción del cliente, aumentar la retención y facilitar el seguimiento de ventas y marketing.
- HRM: Human Resource Management (Gestión de Recursos Humanos), permite gestionar y controlar las actividades relacionadas con la gestión de personal, como reclutamiento, contratación, capacitación, evaluación del desempeño, nóminas y beneficios.
- ERP: Enterprise Resource Planning (Planificación de Recursos Empresariales), sistema integral que integra y administra los diferentes

procesos y recursos de una organización, como ventas, compras, inventario, finanzas y recursos humanos.

1.3.10. Base de datos

Una base de datos se define como un repositorio estructurado de datos, organizados de manera lógica y almacenados digitalmente en el centro de un sistema de información. Este conjunto de datos se caracteriza por su capacidad para ser gestionado, accedido y actualizado eficientemente, proporcionando así un medio para el almacenamiento y recuperación de información relevante.

Existen 2 tipos de bases de datos por su estructura:

- Relacional (SQL – Structured Query Language).
- No Relacional (NoSQL – Not Only SQL).

1.3.11. Base de datos relacional

Este tipo de base de datos se fundamenta en la teoría de conjuntos y relaciones. Utilizan el lenguaje SQL que permite almacenar y organizar los datos en tablas que contienen filas y columnas, a su vez, las tablas pueden mantener relaciones entre sí [28].

Entre los SGBD Relacionales más importantes tenemos:

- MySQL.
- Microsoft SQL Server.
- PostgreSQL.
- Oracle.

1.3.12. Base de datos no relacional

El surgimiento de las bases de datos NoSQL a fines de los años 90 fue motivado por la necesidad de gestionar eficientemente grandes volúmenes de datos. Estas tecnologías de almacenamiento estructurado, a diferencia de las bases de datos relacionales, abordan una gran cantidad de datos generada por comunidades en línea, redes sociales y proyectos asociados a la Web 2.0. Las bases de datos NoSQL se destacan por su alta escalabilidad, rendimiento excepcional y distribución eficiente [29].

Entre las bases de datos no relacionales más importantes tenemos:

- MongoDB.
- Amazon DynamoDB.
- Apache Cassandra.

1.3.13. Sistema de Gestión de Base de Datos

Los Sistemas de Gestión de Bases de Datos (SGBD) son componentes esenciales en entornos informáticos y de desarrollo de software, actúan como intermediarios cruciales, permitiendo a los usuarios interactuar con las bases de datos de manera efectiva. Proporcionan interfaces que facilitan la creación, consulta, modificación y eliminación de datos, contribuyendo a la integridad, seguridad y eficiencia de la gestión de la información [30].

1.3.14. JHipster

JHipster es una plataforma para el desarrollo de aplicaciones web que ayuda a la creación y generación de código de aplicaciones empresariales escalables y de alta calidad. Proporciona una estructura de proyecto preconfigurada, una arquitectura modular y una serie de generadores de código que permiten generar rápidamente características comunes, utilizando el lenguaje Java.

La alternativa de JHipster llamada Kotlin Hipster, es una adaptación que utiliza el lenguaje de programación Kotlin en lugar de Java. Al igual que JHipster, Kotlin Hipster combina tecnologías populares como Spring Boot y Angular o React para generar rápidamente aplicaciones modernas [31].

1.3.15. Metodologías ágiles

En el área del desarrollo de software, una metodología ágil es un conjunto de prácticas y marcos de trabajo que se basan en el enfoque iterativo incremental en búsqueda de la calidad de sus productos. Su enfoque nace como respuesta a las metodologías tradicionales poco eficientes que causaban problemas principalmente de falta de flexibilidad y adaptación a cambios en cuanto a los requisitos del cliente. Entre las metodologías ágiles más conocidas se encuentra Scrum, Lean, Extreme Programming, Kanban, entre otras [32].

1.3.16. Metodologías híbridas

La combinación de elementos entre las metodologías tradicionales y las metodologías ágiles, o combinaciones entre sí, da como resultado una metodología híbrida, extrae las fortalezas de cada metodología y las adapta a las necesidades de cada equipo [33]. dando como resultado un conjunto de prácticas con las siguientes características:

- Enfoque en la calidad y mejora continua
- Flexibles
- Adaptables
- De seguimiento
- Basadas en la entrega de valor
- Iterativos e incrementales

1.4. Objetivos

1.4.1. Objetivo general

Implantar una aplicación web para la Gestión y Control de la Información Laboral del equipo de desarrollo de la empresa Software Evolutivo.

1.4.2. Objetivos específicos

- Analizar los procesos que intervienen en la Gestión y Control de la Información Laboral de la empresa Software Evolutivo.
- Determinar la metodología y herramientas necesarias para un correcto desarrollo de la aplicación web.
- Desarrollar los módulos informáticos para la Gestión y Control de la Información Laboral.

CAPÍTULO II. METODOLOGÍA

2.1. Materiales

Para poder comprender el proceso de gestión y control de la información laboral dentro de la empresa Software Evolutivo se aplicaron entrevistas al Gerente de Operaciones, Líderes de Proyecto, Desarrolladores.

El documento guía de las 3 diferentes entrevistas se encuentran en el *Anexo A*, *Anexo B* y *Anexo C*, respectivamente.

2.2. Métodos

2.2.1. Modalidad de la investigación

La investigación de este proyecto se llevó a cabo en base a la modalidad bibliográfica, modalidad de campo y modalidad aplicada, como se detalla a continuación:

Modalidad bibliográfica

Esta modalidad se utilizó con el fin de recolectar información en artículos científicos, libros, tesis del área de informática, revistas científicas y documentos técnicos sobre el aspecto técnico y de operaciones que son los dos ejes fundamentales de este proyecto de investigación.

Modalidad de campo

La modalidad de campo fue considerada con el fin de recolectar información de manera directa de los profesionales encargados del área de operaciones de la empresa en cuestión mediante entrevistas sobre los procesos de operaciones, necesidades y requisitos con relación a la Gestión y Control de la Información Laboral.

Modalidad aplicada

Esta modalidad aplicada permitió llevar el conocimiento teórico a la práctica resolviendo una problemática de la vida real, en este proyecto de investigación se utilizaron los conocimientos adquiridos durante estos años de estudio para solventar una necesidad del área de operaciones de la empresa en cuestión.

2.2.2. Población y muestra

La población de estudio para el presente proyecto corresponde a todos los miembros de la empresa Software Evolutivo, es decir, 8 personas.

Tabla 1. Población de estudio

| Población | Número | Porcentaje |
|-----------------------------|---------------|-------------------|
| Gerente de operaciones | 1 | 12,5% |
| Desarrolladores de Software | 5 | 62,5% |
| Líderes de Proyectos | 2 | 25% |
| Total | 8 | 100% |

Debido a que la población es pequeña se consideró recolectar la información mediante entrevistas a todos los miembros de la población, por lo tanto, no fue necesario realizar un cálculo de la muestra.

2.2.3. Recolección de información

Los resultados de la entrevista aplicada al Gerente de Operaciones de la empresa Software Evolutivo se presentan a continuación:

Tabla 2. Resultados de la entrevista al Gerente de Operaciones

| N° | Pregunta | Respuesta | Observación |
|----|--|--|---|
| 1 | ¿Cuáles son sus principales responsabilidades y objetivos como gerente de operaciones en la empresa? | Que los equipos de trabajo puedan estar activos apoyando, ya sea a proyectos con clientes o a iniciativas propias de la empresa. También está bajo mi cargo la responsabilidad de emitir los reportes de trabajo requeridos para iniciar los procesos de facturación con los clientes. | El Gerente de Operaciones es el actor final en el proceso de Gestión y Control de la Información Laboral. |
| 2 | ¿Qué tipo de información y datos necesita para tomar decisiones estratégicas en la empresa? | Es importante poder contar con información resumida sobre el tiempo invertido en los proyectos con clientes o iniciativas internas, así como también el registro de las tecnologías, herramientas y prácticas que estamos aplicando en los mismos. | Se requiere el registro de cada actividad con horas y detalle de tecnologías y herramientas. |
| 3 | ¿Qué problemas ha encontrado en los datos para tomar decisiones? | Que no siempre se encuentra la información necesaria para poder tomar decisiones. | Se requiere más datos para generar información que permita la toma de decisiones. |
| 4 | ¿Cómo evalúa actualmente el rendimiento de los proyectos y el equipo de desarrollo? | Ha sido algo más subjetivo, por los comentarios de satisfacción o quejas de los clientes. | No se tiene una métrica definida. |
| 5 | ¿En base a la información que dispone, considera que la evaluación de los proyectos ha sido la | Se puede mejorar mucho, por ejemplo, se podría cruzar la información financiera para conocer si ha sido exitoso en esos términos. | Se busca contar con una métrica para la medición de objetivos financieros. |

| N° | Pregunta | Respuesta | Observación |
|---|--|---|---|
| | correcta, si su respuesta es negativa, por qué? | | |
| 6 | ¿Qué tipo de informes o reportes le serían más útiles en la gestión de operaciones? | Un reporte general sobre la ocupación de los desarrolladores, considerando el proyecto y la cantidad de horas. Además, un reporte que detalle el trabajo realizado en los proyectos con cada cliente ya que esta información es utilizada como soporte para los procesos de facturación. | Emitir reportes en base a las actividades laborales de los desarrolladores, en base a proyectos y clientes. |
| 7 | ¿Cuál sería el tipo de aplicación informática en la que se sentiría más a gusto para revisar informes o analizar informes del registro de actividades? (Aplicación web, móvil, etc.) | Una aplicación Web sería los más adecuado por la cantidad de información presente y también porque la emisión y revisión de esta información se la hace una vez al mes y se puede planificar. | Aplicación Web para manejar de mejor manera la información. |
| 8 | ¿Hay algún otro aspecto importante que desee agregar sobre la gestión de información laboral en la empresa "Software Evolutivo"? | Algunos clientes nos solicitan un reporte de trabajo, indicando las horas invertidas para cada uno de los proyectos que tenemos con ellos, entonces es importante poder contar con una herramienta que facilite este registro. Esta información también es importante para uso interno, ya que podremos evaluar en qué iniciativas estamos invirtiendo el tiempo. | Implementar reportes será de utilidad para temas administrativos, financieros y de evaluación interna. |
| Conclusión: Después de haber realizado esta entrevista se concluye que actualmente el proceso manual de gestión y control de la información laboral es ineficiente e incompleto. | | | |

| N° | Pregunta | Respuesta | Observación |
|----|--|-----------|-------------|
| | Se busca la implantación de una aplicación web que permita solventar estos problemas, permitiendo generar al final de cada ciclo, reportes que agreguen valor en el aspecto organizacional y de procesos internos de la empresa. | | |

Los resultados de las 2 entrevistas aplicadas a los Líderes de Proyectos de la empresa Software Evolutivo se presentan a continuación:

Tabla 3. Resultados de la entrevista al Líder de Proyecto N° 1

| N° | Pregunta | Respuesta | Observación |
|----|---|--|---|
| 1 | ¿Cuáles son sus responsabilidades como líder de proyecto en "Software Evolutivo"? | La implementación de un producto o servicio que cumpla con las expectativas del cliente y más que todo que agregue valor al negocio. | Se encarga de validar que los productos de software obtenidos sean de calidad. |
| 2 | ¿Cómo supervisa y gestiona actualmente las actividades laborales de los desarrolladores en sus proyectos? | Se utilizan marcos de trabajo ágiles como Scrum y Extreme Programming y técnicas como Kanban. Para la supervisión, se busca y revisa las hojas de cálculo donde los desarrolladores registran sus actividades diarias. | El proceso de supervisión es un proceso manual y descentralizado, |
| 3 | ¿Qué problemas ha encontrado al supervisar y gestionar las actividades laborales de los desarrolladores? | Los desarrolladores suelen ingresar registros con errores, asignando actividades a proyectos incorrectos o sin asignarlos. | Es común encontrar información de las actividades registrada con errores. |
| 4 | ¿Cómo se comunica con los desarrolladores sobre novedades o cambios en las actividades laborales? | Nuestra primera opción es la comunicación cara a cara, o lo equivalente para ambientes de trabajo remoto (correo, mensajería instantánea), de manera tradicional. Sin embargo, sería importante tener un | La comunicación se da por medios digitales varios. Se requiere tener un medio único para llevar un registro |

| N° | Pregunta | Respuesta | Observación |
|--|--|--|--|
| | | sitio solo para registrar los cambios requeridos en las actividades laborales. | de las novedades encontradas. |
| 5 | ¿Qué problemas ha encontrado el momento de comunicar a los desarrolladores sobre novedades o cambios en las actividades laborales? | Quizás el mayor problema puede ser la falsa idea de que estamos claros sobre los acuerdos o sobre los detalles de la situación que estamos enfrentando, ya que en algunas ocasiones no nos damos el tiempo de confirmar lo conversado. | Se requiere confirmación de que las novedades encontradas han sido comprendidas y resueltas. |
| 6 | ¿Qué información o funcionalidades específicas le ayudarían a ser más efectivo en su rol? | Por la cantidad de frentes que podemos tener abiertos, sería importante tener una visión amplia y priorizada de todos los temas en los cuales estamos trabajando. | Se requiere organización para los frentes o proyectos que se encuentren en desarrollo. |
| 7 | ¿Cuál sería el tipo de aplicación informática en la que se sentiría más a gusto para supervisar y gestionar actividades? (Aplicación web, móvil, etc.) | Principalmente una interfaz Web de escritorio, ya que para las reuniones de trabajo necesitamos contar con otras herramientas que son mejor aprovechadas en estas plataformas. | Una aplicación web, por el aprovechamiento de las herramientas que puedan incluirse. |
| 8 | ¿Hay algún otro aspecto importante que desee agregar sobre la gestión de información laboral en la empresa "Software Evolutivo"? | Es probable, más que todo con desarrolladores nuevos, que se tenga que solicitar correcciones en el detalle del registro de actividades. Entonces, se requiere un mecanismo eficiente y efectivo de comunicación para expresar los problemas que pueda tener la información ingresada. | Los problemas de comunicación sobre las novedades de la información ingresada vuelven a ser mencionados. |
| <p>Conclusión: Al finalizar esta entrevista se puede concluir que para el líder de proyecto es importante la notificación de las novedades encontradas en los registros de actividades de los desarrolladores, debido a que está encargado de validar que dicha información sea correcta.</p> | | | |

| N° | Pregunta | Respuesta | Observación |
|-----------|---|------------------|--------------------|
| | Se evidencia la necesidad de implementar una aplicación web que le permita tener el control centralizado de esta información. | | |

Tabla 4. Resultados de la entrevista al Líder de Proyecto N° 2

| N° | Pregunta | Respuesta | Observación |
|-----------|---|--|---|
| 1 | ¿Cuáles son sus responsabilidades como líder de proyecto en "Software Evolutivo"? | Trabajar en estimaciones de tiempo y costo de proyectos, buscando y priorizando siempre lo que más aporte al negocio basado en sus expectativas de resultados finales. Luego hacer seguimiento de los avances de la implementación, removiendo impedimentos en el camino, despejando dudas funcionales en conjunto con el cliente y asegurando un camino de implementación orientado a resultados en función del tiempo y valor. | Tiene como responsabilidad el seguimiento, control y aseguramiento de la calidad de los proyectos de los cuales está a cargo. |
| 2 | ¿Cómo supervisa y gestiona actualmente las actividades laborales de los desarrolladores en sus proyectos? | Reuniones diarias con el equipo, para asegurar el enfoque en valor y la solución de impedimentos; revisiones a final de cada mes de sus actividades en los proyectos. | Supervisa de manera manual la información de las actividades de los proyectos. |
| 3 | ¿Qué problemas ha encontrado al supervisar y gestionar las actividades laborales de los desarrolladores? | Debido al uso de herramientas manuales para este tipo de gestión, los problemas más comunes se dan en la asignación de actividades a los leads, o la falta de información sobre cada actividad realizada. | Se requiere que la información registrada tenga detalle que agregue valor a cada registro. |

| N° | Pregunta | Respuesta | Observación |
|----|--|--|---|
| 4 | ¿Cómo se comunica con los desarrolladores sobre novedades o cambios en las actividades laborales? | Reuniones remotas o presenciales lo antes posible; mediante correo o canales digitales. | Se puede evidenciar que la comunicación no está definida. |
| 5 | ¿Qué problemas ha encontrado el momento de comunicar a los desarrolladores sobre novedades o cambios en las actividades laborales? | Normalmente ninguno. | |
| 6 | ¿Qué información o funcionalidades específicas le ayudarían a ser más efectivo en su rol? | Buscaría maximizar el valor y cuidar el presupuesto de la empresa con la ayuda de herramientas más efectivas para el cumplimiento de mis responsabilidades como Líder de Proyecto en Software Evolutivo. | Se requiere contar con mejores herramientas para los procesos internos de la empresa. |
| 7 | ¿Cuál sería el tipo de aplicación informática en la que se sentiría más a gusto para supervisar y gestionar actividades? (Aplicación web, móvil, etc.) | Las computadoras de escritorio o laptops se han convertido en nuestra mano derecha para el trabajo. Y es cómodo poder realizarlo todo desde ahí, entonces, considero adecuada una aplicación web. | Se considera una aplicación web para poder manejarlo todo desde la misma estación de trabajo. |
| 8 | ¿Hay algún otro aspecto importante que desee agregar sobre la gestión de información laboral en la empresa "Software Evolutivo"? | Establecer buenos mecanismos de comunicación es clave para cualquier trabajo en equipo, además de crear una cultura de colaboración, empatía y responsabilidad. | La buena comunicación con el equipo es considerada un factor clave. |

| N° | Pregunta | Respuesta | Observación |
|---|----------|-----------|-------------|
| Conclusión: Al entrevistar a este segundo Líder de Proyecto se ha concluido que las herramientas utilizadas actualmente hacen que el proceso de control de la información laboral sea manual e ineficiente. Se requiere una herramienta cómoda y funcional que permita una comunicación adecuada con el equipo, como una aplicación web. | | | |

Los resultados de las 5 entrevistas aplicadas a los Desarrolladores de Software de la empresa Software Evolutivo se presentan a continuación:

Tabla 5. Resultados de la entrevista al Desarrollador de Software N° 1

| N° | Pregunta | Respuesta | Observación |
|----|---|--|--|
| 1 | ¿Cuáles son las principales tareas y actividades que realiza como desarrollador en "Software Evolutivo"? | Desarrollo de software, gestión y planificación de proyectos. | |
| 2 | ¿Registra sus actividades laborales diarias? | Sí. | |
| 3 | Si su respuesta fue Sí, ¿Cómo las registra actualmente? | En una hoja de cálculo dividido por días y por proyectos. | Se registran las actividades de manera manual en hojas de cálculo. |
| 4 | ¿Qué problemas ha enfrentado al registrar sus actividades laborales o al comunicarse con sus líderes de proyecto? | Que no esté claro el proyecto al cual se asigna las horas, o se tome muchos frentes en una misma hora. | Se evidencia un problema de desorganización en el registro. |
| 5 | ¿Cómo han solucionado los problemas encontrados al registrar sus actividades laborales? | Registrando las horas a la par como se esté trabajando en un proyecto y preguntando al líder del proyecto. | Para resolver problemas se establece comunicación directa |

| N° | Pregunta | Respuesta | Observación |
|---|---|---|---|
| | | | con el Líder de Proyecto. |
| 6 | ¿Qué características o funcionalidades le resultarían más útiles en una aplicación de gestión de las actividades laborales? | Poder escoger el proyecto previo al registro de horas, que no sea tan sensible al cambio (en las horas de inicio y fin de la actividad), es decir que yo pueda simplemente elegir o ingresar mi actividad y en la aplicación poder escoger los proyectos disponibles. | Se busca implementar una solución que permita registrar las actividades de manera más sencilla y automática. |
| 7 | ¿Cuál sería el tipo de aplicación informática en la que se sentiría más a gusto para registrar sus actividades? (Aplicación web, móvil, etc.) | Las dos me parecerían muy útiles, pero al estar en el horario laboral trabajando en una PC, tendría preferencia por una aplicación web. | Se busca implementar una aplicación web, por las comodidades que brindaría. |
| 8 | ¿Hay algún otro aspecto importante que desee agregar sobre la gestión de información laboral en la empresa "Software Evolutivo"? | Quizá tener un árbol de aprendizaje automatizado para cualquier colaborador nuevo en la empresa, una especie de tour laboral, con eso se automatizaría capacitaciones e inversión de tiempo en este aspecto. | Se evidencia que actualmente es necesario capacitar a los nuevos desarrolladores para que sepan cómo registrar sus actividades. |
| <p>Conclusión: Al finalizar la primera entrevista a un Desarrollador, se puede ver el motivo por el cual la información es registrada de manera errónea en ocasiones y que se requiere de la intervención del Líder de Proyecto para resolver estas novedades.</p> <p>Se evidencia la necesidad de una solución que automatice y permita que este proceso no sea tan tedioso para los desarrolladores y sería cómodo realizarlo en una aplicación web.</p> | | | |

Tabla 6. Resultados de la entrevista al Desarrollador de Software N° 2

| N° | Pregunta | Respuesta | Observación |
|----|---|---|--|
| 1 | ¿Cuáles son las principales tareas y actividades que realiza como desarrollador en "Software Evolutivo"? | Participo en el despliegue, actualización y corrección de software. También colaboro en el diseño y desarrollo de programas, ya sea creando nuevos sistemas o migrando los existentes. Además, contribuyo activamente en la resolución de problemas técnicos. Para garantizar la calidad del producto, llevo a cabo pruebas tanto funcionales como de rendimiento. Por último, mi labor incluye la depuración del código, identificando y solucionando cualquier problema que pueda surgir durante el desarrollo. | |
| 2 | ¿Registra sus actividades laborales diarias? | Sí. | |
| 3 | Si su respuesta fue Sí, ¿Cómo las registra actualmente? | Registro mis actividades al finalizar el día o al iniciar el día siguiente. Coloco un borrador de las actividades y después de 10 o 15 días, reviso todos esos borradores para asegurarme de que estén bien escritos y no tengan ninguna falta ortográfica. Finalmente, adjunto el registro de actividades como versión final. | Registra las actividades de manera manual y en un proceso que lleva más tiempo y esfuerzo de lo esperado. |
| 4 | ¿Qué problemas ha enfrentado al registrar sus actividades laborales o al comunicarse con sus líderes de proyecto? | Uno de los problemas es que, al realizar alguna de las actividades asignadas, a veces se extienden más allá de la jornada laboral. Debido a que me concentro en la actividad, a veces olvido registrar mis actividades, especialmente si se realizó una actividad en grupo o surgen otras aparte de la que se extendió. | Se evidencia la complejidad y el tiempo que requiere registrar manualmente las actividades diarias de los desarrolladores. |

| N° | Pregunta | Respuesta | Observación |
|----|--|---|---|
| 5 | ¿Cómo han solucionado los problemas encontrados al registrar sus actividades laborales? | Reviso las tareas asignadas en el tablero de actividades del equipo para recordar qué hicimos en esa actividad. También reviso los registros de mis compañeros para recordar las actividades que realizamos. Por último, si registro mal mis actividades, el líder de proyecto se comunica conmigo por las novedades encontradas y las soluciono. | El desarrollador establece un canal de comunicación con el Líder de Proyecto para dar solución a los problemas. |
| 6 | ¿Qué características o funcionalidades le resultarían más útiles en una aplicación de gestión de las actividades laborales? | Debe contar con una interfaz intuitiva y de fácil uso. Además, enviar notificaciones para notificar las novedades encontradas en mi registro de actividades. | Se requiere notificaciones para estar al tanto de las novedades encontradas. |
| 7 | ¿Cuál sería el tipo de aplicación informática en la que se sentiría más a gusto para registrar sus actividades? (Aplicación web, móvil, etc.) | Aplicación web o móvil. | Flexibilidad en estas dos alternativas. |
| 8 | ¿Hay algún otro aspecto importante que desee agregar sobre la gestión de información laboral en la empresa "Software Evolutivo"? | No. | |

| N° | Pregunta | Respuesta | Observación |
|--|----------|-----------|-------------|
| <p>Conclusión: Después de la entrevista realizada a este segundo Desarrollador de Software podemos notar que, para él, el proceso de registro de actividades es tedioso por la manera en la que se lo hace actualmente.</p> <p>Se da a notar que a más de necesitar otra interfaz u otro lugar en dónde registrar las actividades, se necesita que el proceso de registro sea sencillo e intuitivo.</p> | | | |

Tabla 7. Resultados de la entrevista al Desarrollador de Software N° 3

| N° | Pregunta | Respuesta | Observación |
|----|---|--|--|
| 1 | ¿Cuáles son las principales tareas y actividades que realiza como desarrollador en "Software Evolutivo"? | Desarrollo de software y mantenimiento de software existente. | |
| 2 | ¿Registra sus actividades laborales diarias? | Sí. | |
| 3 | Si su respuesta fue Sí, ¿Cómo las registra actualmente? | En una hoja de cálculo en línea. | Registro manual de las actividades diarias. |
| 4 | ¿Qué problemas ha enfrentado al registrar sus actividades laborales o al comunicarse con sus líderes de proyecto? | Confundir clientes o proyectos a los cuales le asigno mis actividades. | La cantidad de registros puede llegar a confundir a los desarrolladores. |
| 5 | ¿Cómo han solucionado los problemas encontrados al registrar sus actividades laborales? | El líder de proyecto se comunica conmigo a avisarme de las novedades por correo o plataformas de mensajería. | Comunicación tradicional por medios digitales. |
| 6 | ¿Qué características o funcionalidades le resultarían más | Que solo deba seleccionar los proyectos y no | Se requiere que el proceso sea más automático para evitar |

| N° | Pregunta | Respuesta | Observación |
|---|---|--------------------------------------|-----------------------------------|
| | útiles en una aplicación de gestión de las actividades laborales? | preocuparme por quien es el cliente. | esfuerzos innecesarios y errores. |
| 7 | ¿Cuál sería el tipo de aplicación informática en la que se sentiría más a gusto para registrar sus actividades? (Aplicación web, móvil, etc.) | Aplicación web. | |
| 8 | ¿Hay algún otro aspecto importante que desee agregar sobre la gestión de información laboral en la empresa "Software Evolutivo"? | No. | |
| Conclusión: Al finalizar esta entrevista, se ha concluido que el desarrollador requiere que el registro sea menos manual, con datos predefinidos que solo requieran de seleccionar opciones y registrar. | | | |

Tabla 8. Resultados de la entrevista al Desarrollador de Software N° 4

| N° | Pregunta | Respuesta | Observación |
|----|--|---|--|
| 1 | ¿Cuáles son las principales tareas y actividades que realiza como desarrollador en "Software Evolutivo"? | Planificación, desarrollo y mantenimiento de sistemas informáticos. | |
| 2 | ¿Registra sus actividades laborales diarias? | Sí. | |
| 3 | Si su respuesta fue Sí, ¿Cómo las registra actualmente? | En una hoja de cálculo. | Se puede evidenciar que las hojas de cálculo son la herramienta más utilizada para estos procesos. |

| N° | Pregunta | Respuesta | Observación |
|----|---|---|--|
| 4 | ¿Qué problemas ha enfrentado al registrar sus actividades laborales o al comunicarse con sus líderes de proyecto? | Mal cálculo de horas, textos muy largos al describir la actividad. | Se requiere que la duración de la actividad sea automática para evitar errores en el cálculo de horas. |
| 5 | ¿Cómo han solucionado los problemas encontrados al registrar sus actividades laborales? | Fórmulas de excel para el cálculo de horas y definir áreas relacionadas a cada actividad. | La solución de errores las realiza independiente después de realizar revisión a sus registros. |
| 6 | ¿Qué características o funcionalidades le resultarían más útiles en una aplicación de gestión de las actividades laborales? | Mejor cálculo de horas, proyectos y clientes predefinidos. | Se requiere la automatización del cálculo de horas, selección de proyectos y clientes más sencilla. |
| 7 | ¿Cuál sería el tipo de aplicación informática en la que se sentiría más a gusto para registrar sus actividades? (Aplicación web, móvil, etc.) | Todas. Sin embargo, prefiero hacerlo desde mi lugar y horas de trabajo. | El desarrollador se sentiría cómodo al registrar sus actividades en cualquier tipo de aplicación, pero da preferencia a una aplicación web por la facilidad de escribir en su teclado y puesto de trabajo. |
| 8 | ¿Hay algún otro aspecto importante que desee agregar sobre la gestión de información laboral en la empresa "Software Evolutivo"? | No. | |

| N° | Pregunta | Respuesta | Observación |
|--|----------|-----------|-------------|
| <p>Conclusión: Al finalizar esta cuarta entrevista a un Desarrollador de Software se puede concluir que se requiere que el proceso de registro de actividades laborales pueda ser menos propenso a errores de ingreso o cálculo de horas. El desarrollo de una aplicación web para realizar este proceso es el preferido.</p> | | | |

Tabla 9. Resultados de la entrevista al Desarrollador de Software N° 5

| N° | Pregunta | Respuesta | Observación |
|----|---|---|--|
| 1 | ¿Cuáles son las principales tareas y actividades que realiza como desarrollador en "Software Evolutivo"? | Consultoría, desarrollo de software, capacitación y colaboración en actividades de la empresa. | |
| 2 | ¿Registra sus actividades laborales diarias? | Sí. | |
| 3 | Si su respuesta fue Sí, ¿Cómo las registra actualmente? | En Google SpreadSheets. | Registra sus actividades en una hoja de cálculo en línea. |
| 4 | ¿Qué problemas ha enfrentado al registrar sus actividades laborales o al comunicarse con sus líderes de proyecto? | Ninguno en particular, solo es muy tedioso manejar de manera manual las celdas y formato del documento. | Se requiere que el registro de las actividades sea lo único de lo que el desarrollador se preocupe, más no por formatos o estructura de los archivos que maneja. |
| 5 | ¿Cómo han solucionado los problemas encontrados al registrar sus actividades laborales? | Con el líder de proyecto mediante canales de comunicación como Discord o por correo. | Actualmente, la comunicación es tradicional por medios digitales. |

| N° | Pregunta | Respuesta | Observación |
|---|---|--|---|
| 6 | ¿Qué características o funcionalidades le resultarían más útiles en una aplicación de gestión de las actividades laborales? | Que solo deba preocuparme por registrar mis actividades. | El desarrollador da a notar que se requiere una solución que le permita simplemente estar al pendiente de sus actividades y no por otros factores que no agregan valor. |
| 7 | ¿Cuál sería el tipo de aplicación informática en la que se sentiría más a gusto para registrar sus actividades? (Aplicación web, móvil, etc.) | Me parece más cómodo hacerlo desde la computadora, por lo tanto, una aplicación web. | Se da preferencia a las aplicaciones web para realizarlo por la computadora. |
| 8 | ¿Hay algún otro aspecto importante que desee agregar sobre la gestión de información laboral en la empresa "Software Evolutivo"? | Que es importante que esta información sea correcta y consistente. | Se busca la consistencia de los datos. |
| <p>Conclusión: Después de esta última entrevista, se puede concluir que los Desarrolladores de Software coinciden en que la solución al problema del registro manual de las actividades que genera incomodidad e inconsistencia en datos sería una aplicación web con interfaz intuitiva, fácil de usar, que permita que el proceso sea más sencillo, reduzca al mínimo los errores y que disponga de un canal de comunicación directo con el Líder de Proyecto para dar solución a cualquier novedad.</p> | | | |

2.2.4. Procesamiento y análisis de datos

De acuerdo con las entrevistas aplicadas a los trabajadores de la empresa Software Evolutivo, se concluyó que:

- El proceso de Gestión y Control de la Información Laboral es totalmente manual por lo que es muy propenso a los errores que se puedan cometer en el registro de sus datos, consecuentemente, el proceso es lento, inconsistente y

requiere de tiempo y esfuerzo de parte de cada rol que se ve inmerso en este proceso.

- El sistema por desarrollarse, deberá ser una aplicación web que permita a todos los miembros de la empresa en sus diferentes roles sentirse cómodos y a gusto realizando este proceso en una aplicación intuitiva, amigable, de fácil uso, de seguro acceso a los datos e información de importancia, desde su misma estación de trabajo con las herramientas que manejan a diario.
- La aplicación web debe permitir la creación, edición, eliminación y vista de la información de los Desarrolladores de Software, incluyendo:
 - Actividades.
 - Observaciones.
 - Reportes por cliente y por proyecto.
- Toda la información de las actividades laborales deberá ser accesible y disponible a todo momento para quienes así lo necesiten, para poder así permitir que el proceso de Gestión y Control de la Información Laboral sea fluido, consistente y de calidad.
- Todos los miembros que han sido entrevistados consideran necesaria la implantación de esta aplicación web, para poder mejorar este proceso haciéndolo más eficiente y automático, lo que se verá reflejado en la productividad de la empresa.
- Es necesario que la aplicación web permita el envío de notificaciones mediante correo, para notificar las novedades que se encuentren en las actividades laborales, permitiendo así brindar el canal de comunicación directo y automático que se requiere.
- La generación de los reportes deberá llevarse a cabo dentro de la misma aplicación web, permitiendo al Gerente de Operaciones tener a la mano la

información y extraerla para poder así cumplir con la gestión administrativa y financiera que le compete.

En base a la información recolectada, se identificaron y se definieron los requisitos funcionales y no funcionales alineados a los objetivos y necesidades específicas de la empresa para el desarrollo de la aplicación web de Gestión y Control de la Información Laboral.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1. Análisis y discusión de los resultados

3.1.1. Diagrama del proceso de gestión con el sistema

El proceso de gestión y control de la información laboral de la empresa Software Evolutivo con la aplicación web planteada se presenta a continuación:

1. Cada desarrollador registra diariamente en una hoja de cálculo de manera manual sus actividades diarias con descripción, horas de inicio y fin, proyecto y cliente al que le pertenece.
2. Mensualmente, el líder de proyecto revisa las actividades de todos los desarrolladores que están involucrados en cada proyecto:
 - a. Si hay registros con errores, se notifica al desarrollador.
 - b. Si no hay errores, su función termina.
3. Si hay registros con errores, el desarrollador deberá corregir los registros en cuestión.
4. Si no hay registros con errores, o los errores han sido corregidos:
 - a. El líder de proyecto notifica al gerente de operaciones que la información laboral está lista.
 - b. El gerente de operaciones genera reportes mensuales de esta información.

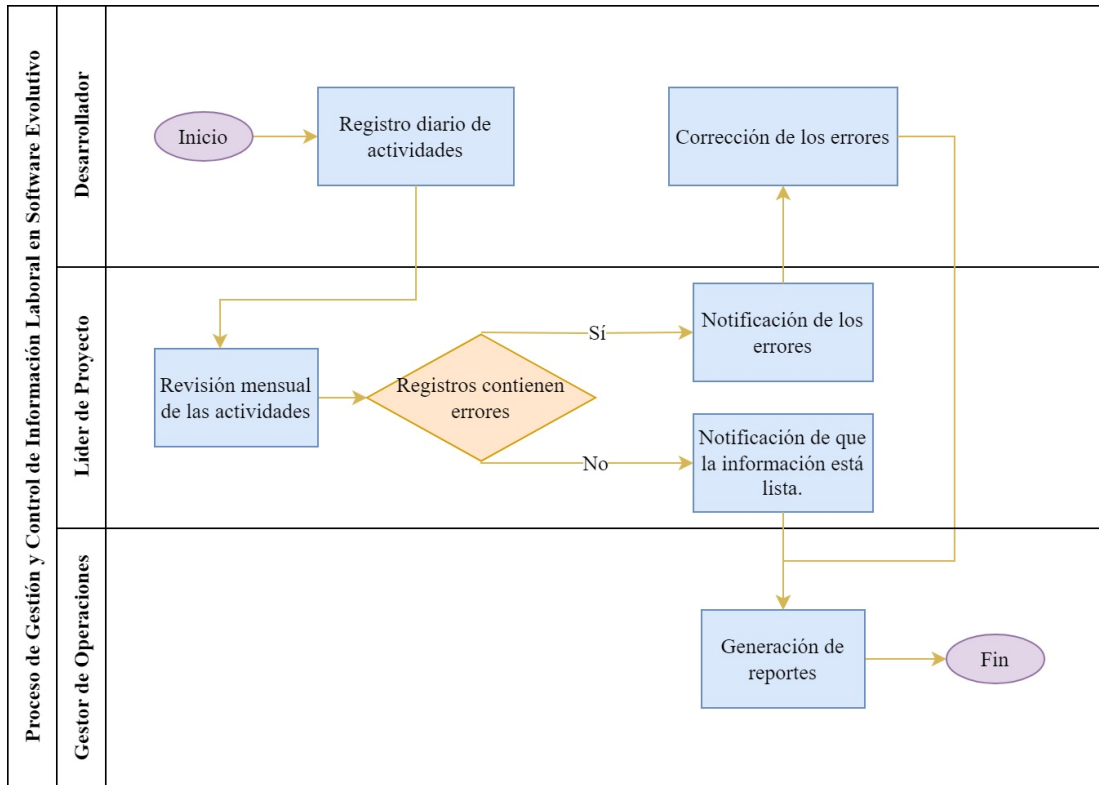


Figura 1. Diagrama del proceso de gestión con la aplicación web

3.1.2. Análisis y selección de la metodología de desarrollo

Se presentan en la siguiente tabla comparativa las principales características entre algunos de los principales marcos de trabajo para el desarrollo de software.

Tabla 10. Tabla comparativa de las metodologías de desarrollo

| Criterio | Scrum | XP | Kanban | Lean |
|----------------|--|---|--|---|
| Enfoque | Marco de trabajo ágil iterativo e incremental. | Enfoque de desarrollo ágil iterativo e incremental. | Se centra en la gestión y optimización del flujo de trabajo. | Marco de trabajo que busca mejorar la eficiencia en los procesos. |

| Criterio | Scrum | XP | Kanban | Lean |
|--|---|---|--|---|
| Ventajas | <ul style="list-style-type: none"> - Adaptación a cambios en los requerimientos. - Motiva a la colaboración. - Entregas continuas. | <ul style="list-style-type: none"> - Se enfoca en la calidad del código mediante pruebas continuas. - Motiva la comunicación y colaboración. - Eficiencia en la respuesta a cambios. | <ul style="list-style-type: none"> - Se lleva de manera visual el flujo de trabajo. - Tiempos de espera reducidos. - Se adapta a varios tipos de proyectos. | <ul style="list-style-type: none"> - Es eficiente y reduce costos sin perder la calidad. - Busca crear valor en el cliente. |
| Tamaño del Equipo de Desarrollo | 5-9 personas. | 2-12 personas. | No define. | 2-3 personas. |
| Tamaño del Proyecto | Mediano a grande. | Pequeños o medianos. | Pequeños, medianos y grandes. | Grandes, con requisitos complejos. |
| Colaboración con el Cliente | Colaboración activa con el Product Owner en reuniones de revisión. | Colaboración continua con el cliente. | Fomenta la transparencia y la comunicación con el cliente. | Participación del cliente para identificar el valor y eliminar desperdicios. |
| Plazos y Entregas | Entregas frecuentes al fin del sprint (2-4 semanas). | Entregas frecuentes con iteraciones de 1-2 semanas. | Entregas a medida que las tareas se terminan. | Sin un marco de tiempo fijo. |

| Criterio | Scrum | XP | Kanban | Lean |
|---|---|---|----------------------|---|
| Escalabilidad | Mediante Scrum a nivel de programas y carteras. | Escalable manteniendo una comunicación estrecha con el cliente. | Altamente escalable. | Escalable en términos de aplicar principios Lean en mayor cantidad en las organizaciones. |
| Conocimiento y experiencia previa. | Moderada. | Alta. | Moderada. | Baja. |

En la Tabla 10 se realizó un análisis a las características más importantes de las metodologías de desarrollo de software [34]. En base a este análisis se decidió utilizar la metodología de desarrollo ágil Extreme Programming (XP) debido a que se enfoca en aplicar prácticas de ingeniería y pruebas continuas que permitan obtener un software de calidad.

El investigador cuenta con experiencia considerable sobre Extreme Programming, así mismo esta metodología es ya conocida y practicada dentro de la empresa Software Evolutivo, por lo que la colaboración con ellos pudo ser estrecha y continua.

3.1.3. Análisis y selección del framework para el desarrollo de la aplicación web.

A continuación, se pondrá en comparación los frameworks para desarrollo frontend de aplicaciones web más conocidos.

Tabla 11. Tabla comparativa de frameworks para aplicaciones web

| Criterio | Angular | React | Flutter | Vue |
|---------------------------------|---|--|---|---|
| Lenguaje de Programación | JavaScript, TypeScript | JavaScript, TypeScript | Dart | JavaScript |
| Curva de Aprendizaje | Moderada | Moderada a Baja | Moderada a Baja | Baja |
| Documentación | Se pueden encontrar guías, tutoriales, la comunidad está activa. | Amplia y respaldada por una gran comunidad de desarrolladores. | Tiene varios recursos disponibles, pero no es tan extensa. | Documentación concisa y fácil de entender. |
| Rendimiento | Buen rendimiento gracias a su capacidad de optimización y su sistema de detección de cambios. | Rendimiento sólido, y está apto a optimizaciones manuales que el desarrollador quiera implementar. | Rendimiento sólido, especialmente en dispositivos móviles debido a la compilación nativa. | Buen rendimiento con un tamaño de paquete pequeño. |
| UI/UX | Ofrece herramientas que permiten crear aplicaciones web ricas en funcionalidades y componentes. | Altamente flexible y una de sus mejores características es que ofrece la integración de bibliotecas y componentes de terceros. | Ideal para la creación de interfaces de usuario atractivas y altamente personalizadas. | Intuitivo y versátil, con una curva de aprendizaje baja para la creación de interfaces. |

| Criterio | Angular | React | Flutter | Vue |
|---|--|---|--|--|
| Integración con Herramientas de Desarrollo | Se integra con Angular CLI y permite automatizar tareas de desarrollo. | Amplia variedad de herramientas y librerías disponibles, como Create React App. Compatible con la mayoría de IDE. | Cuenta con su propia colección de herramientas y widgets, Soporta y es compatible con la mayoría de IDE. | Ofrece una amplia gama de extensiones para diversos IDE. Además, cuenta con Vue CLI, |
| Conocimiento y experiencia previa. | Alta. | Alta. | Baja. | Baja. |

Se realizó un análisis a los diferentes frameworks [35] y se seleccionó React basados en su flexibilidad, respaldada por un extenso ecosistema de herramientas y librerías disponibles de forma gratuita. La decisión se apoyó también por la robusta arquitectura de componentes que React ofrece, la cual facilita el desarrollo de soluciones modulares y fácilmente reutilizables. La combinación de estas características proporciona una base técnica sólida que promete eficiencia, mantenibilidad y escalabilidad en el desarrollo del proyecto.

React cuenta con una comunidad activa en la cual es posible apoyarse para el correcto desarrollo de los módulos de la aplicación web, además el investigador cuenta con experiencia en el uso del framework lo cual permitió la obtención de un resultado de calidad.

Dentro de la empresa Software Evolutivo para la gestión y el desarrollo de proyectos de software se utilizan una serie de herramientas y tecnologías que se han solicitado trabajen en conjunto con React (Frontend) en el desarrollo de la aplicación web:

- JHipster con Kotlin (Framework para aplicaciones monolíticas).

- Spring Boot (Backend).
- PostgreSQL (Base de datos).
- Docker (Contenerización de aplicaciones).
- GitLab (Gestión de proyectos y CI/CD).

3.2. Desarrollo de la propuesta

Para el desarrollo de la aplicación web se utilizaron cuatro etapas de la metodología Extreme Programming [36], las cuales se detallan a continuación:

Planificación

Comenzó con la recopilación de "historias de usuarios". Se evaluó el tiempo de desarrollo y mediante una reunión de planificación se establecieron tiempos de entrega. Luego, se llevaron a cabo iteraciones para desarrollar e implementar las historias de usuarios.

Diseño

XP destaca la importancia de diseños simples y claros debido a que un diseño simple se implementa más rápido que uno complejo. Para el presente desarrollo de software, se realizaron Tarjetas CRC por cada historia de usuario para obtener el diseño más simple posible que permita cumplir con los requisitos establecidos.

Desarrollo del código

Para el desarrollo del código de la aplicación web con XP, se utilizó la programación dirigida por pruebas que permitieron validar constantemente el correcto funcionamiento de la lógica de la aplicación.

Pruebas

Con el objetivo de realizar correcciones constantemente y mantener un control automatizado sobre la lógica, se realizaron tests de integración de las principales

funcionalidades. Además, al hacer la entrega de los módulos de la aplicación web se realizaron pruebas de aceptación y de usabilidad que permitieron recibir constante retroalimentación del cliente.

3.2.1. Definición de roles

El proyecto se llevó a cabo con la participación de tres personas, cuyos detalles se encuentran en la siguiente tabla:

Tabla 12. Definición de roles

| Rol | Responsable | Descripción de funciones |
|---------------------|--------------------------|--|
| Cliente | Ing. Jorge Aguilar Bravo | Gerente General de la empresa Software Evolutivo, quien colaboró continuamente para el correcto desarrollo y entrega del producto final. |
| Encargado de Datos | Ing. Johannes Caranqui | Responsable de la seguridad de los datos en la empresa Software Evolutivo, quien se encargó de los procesos de protección de datos externos a la aplicación web. |
| Analista de Calidad | Ing. Oscar Ibarra | Encargado del aseguramiento de la calidad del software y del cumplimiento de los requerimientos establecidos. |
| Programador | Joshua Martínez | Encargado de la gestión del proyecto y de la programación de los módulos de la aplicación web. |

3.2.2. Recolección de requisitos

Los requisitos para la aplicación web de gestión y control de la información laboral se recolectaron junto al cliente mediante el desarrollo de entrevistas, utilizando historias de usuario.

Para la recolección de requisitos se interactuó con:

- **Desarrolladores de software:** quienes registran diariamente sus actividades.
- **Líderes de proyectos:** quienes supervisan que las actividades se hayan registrado correctamente.
- **Gerente de operaciones:** quien genera y envía reportes a los clientes con fines administrativos y financieros.

3.2.3. Fase I: Planificación

3.2.3.1. Historias de usuario

Se utilizó esta herramienta para poder recopilar de manera clara y concisa lo que el cliente esperaba de la aplicación web, el resultado se presenta a continuación:

| Historia de Usuario | |
|---|---|
| Número: 1 | Usuario: Administrador, Desarrollador, Líder de proyecto, Gerente de Operaciones. |
| Descripción: Como: Usuario. Quiero: Ingresar mis credenciales de usuario y contraseña. Para: Ingresar a la aplicación web. | |
| Criterios de aceptación: <ul style="list-style-type: none">• La contraseña deberá manejarse como un hash por motivos de seguridad.• Se manejarán las sesiones con JWT.• Al momento de ingresar a la aplicación se validará el rol del usuario. | |

| Historia de Usuario | |
|--|--------------------------------|
| Número: 2 | Usuario: Administrador. |
| Descripción: Como: Administrador. Quiero: Gestionar a los usuarios que tendrán acceso a la aplicación. Para: Crear, eliminar, modificar la información de usuarios. | |
| Criterios de aceptación: <ul style="list-style-type: none"> • El campo de contraseña no será visible ni editable para el administrador. | |

| Historia de Usuario | |
|--|--|
| Número: 3 | Usuario: Administrador, Desarrollador, Líder de proyecto, Gerente de Operaciones. |
| Descripción: Como: Usuario. Quiero: Modificar mis datos personales en la aplicación. Para: Tener mi información actualizada y segura. | |
| Criterios de aceptación: <ul style="list-style-type: none"> • La información personal y la de seguridad como contraseñas, deberán manejarse en diferentes pantallas. | |

| Historia de Usuario | |
|----------------------------|--|
| Número: 4 | Usuario: Administrador, Desarrollador, Líder de proyecto, Gerente de Operaciones. |

| |
|---|
| <p>Descripción:</p> <p>Como: Usuario.</p> <p>Quiero: Cambiar de contraseña.</p> <p>Para: Tener mi información confidencial actualizada y segura.</p> |
| <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • La contraseña podrá ser editada únicamente por cada usuario. • La contraseña deberá ser almacenada como hash en la base de datos. |

| Historia de Usuario | |
|--|---|
| Número: 5 | Usuario: Gerente de operaciones. |
| <p>Descripción:</p> <p>Como: Gerente de operaciones.</p> <p>Quiero: Gestionar los clientes que están en relación laboral con la empresa.</p> <p>Para: Crear, eliminar, modificar la información de los clientes.</p> | |
| <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Se deberá almacenar únicamente información relevante sobre los clientes. | |

| Historia de Usuario | |
|---|---|
| Número: 6 | Usuario: Gerente de operaciones. |
| <p>Descripción:</p> <p>Como: Líder de Proyecto.</p> <p>Quiero: Gestionar los proyectos que se desarrollan en la empresa.</p> | |

| |
|--|
| Para: Crear, eliminar, modificar la información de los proyectos de los cuales estoy a cargo. |
| Criterios de aceptación: <ul style="list-style-type: none"> • Cuando se registre un nuevo proyecto, se deberá asignar automáticamente como Líder de Proyecto al usuario que está creando el proyecto. • El Líder de Proyecto no podrá asignar un proyecto a otro usuario. |

| Historia de Usuario | |
|---|--------------------------------|
| Número: 7 | Usuario: Desarrollador. |
| Descripción: Como: Desarrollador. Quiero: Gestionar mis actividades laborales diarias. Para: Crear, eliminar, modificar la información de las actividades que a diario llevo a cabo. | |
| Criterios de aceptación: <ul style="list-style-type: none"> • A cada actividad le corresponde una hora de inicio y fin; y pertenece a un solo proyecto. | |

| Historia de Usuario | |
|---|--------------------------------|
| Número: 8 | Usuario: Desarrollador. |
| Descripción: Como: Desarrollador. Quiero: Generar reportes. Para: Poder guardar un resumen de mis actividades laborales. | |
| Criterios de aceptación: | |

- Los reportes serán por en un mes y año específicos.
- Se podrán descargar los reportes en formato PDF.

Historia de Usuario

Número: 9

Usuario: Líder de proyecto.

Descripción:

Como: Líder de proyecto.

Quiero: Visualizar las actividades laborales de los desarrolladores.

Para: Verificar que la información registrada sea correcta.

Criterios de aceptación:

- Cada líder de proyecto podrá visualizar las actividades solo de los proyectos de los que él se encuentra a cargo.

Historia de Usuario

Número: 10

Usuario: Líder de proyecto.

Descripción:

Como: Líder de proyecto.

Quiero: Registrar observaciones en las actividades registradas.

Para: Notificar al desarrollador que corresponda que su actividad es incorrecta o está incompleta.

Criterios de aceptación:

- Al registrarse una observación, se enviará una notificación al desarrollador que corresponda.

| Historia de Usuario | |
|--|--------------------------------|
| Número: 11 | Usuario: Desarrollador. |
| Descripción: Como: Desarrollador. Quiero: Gestionar las observaciones de mis actividades. Para: Corregir los errores o completar la información de mis actividades. | |
| Criterios de aceptación: <ul style="list-style-type: none"> • Cuando se corrijan los errores las observaciones su estado cambiará a resuelta y se notificará al líder de proyecto. | |

| Historia de Usuario | |
|--|------------------------------------|
| Número: 12 | Usuario: Líder de proyecto. |
| Descripción: Como: Líder de proyecto. Quiero: Notificar que las actividades están listas para ser procesadas. Para: Que el Gerente de operaciones sepa que las actividades laborales mensuales de un proyecto están corregidas y sin errores. | |
| Criterios de aceptación: <ul style="list-style-type: none"> • Se enviará una notificación al Gerente de operaciones. | |

| Historia de Usuario | |
|----------------------------|---|
| Número: 13 | Usuario: Gerente de operaciones. |
| Descripción: | |

| |
|---|
| <p>Como: Gerente de operaciones.</p> <p>Quiero: Generar reportes.</p> <p>Para: Cumplir con los procesos administrativos y financieros que se requieren mensualmente.</p> |
| <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Los reportes podrán ser por cliente en un mes y año específicos. • Se podrán descargar los reportes en formato PDF. |

3.2.4. Fase II: Diseño

3.2.4.1. Tarjetas CRC

Las tarjetas CRC (Clase – Responsabilidad - Colaboración) son parte de la metodología XP. El propósito principal de las tarjetas CRC es facilitar la comprensión de las historias de usuario. Al utilizar estas tarjetas, se pueden visualizar rápidamente cómo interactúan las clases en el sistema y comprender sus responsabilidades individuales [37].

Se creó una tarjeta CRC por cada historia de usuario que se definió, y se presentan a continuación:

Tabla 13. Tarjetas CRC - Autenticación de usuario

| Autenticación de usuario | |
|-------------------------------------|---|
| Responsabilidad | Colaboradores |
| Validar las credenciales de ingreso | Capa de presentación Controlador Métodos de lógica de negocio Capa de acceso a datos |

| |
|-----------------------|
| Observaciones: |
|-----------------------|

Tabla 14. Tarjetas CRC - Crear usuario

| Crear usuario | |
|--|---|
| Responsabilidad | Colaboradores |
| Recibir parámetros. Validar parámetros Crear usuarios. | Capa de presentación Controlador Métodos de lógica de negocio Capa de acceso a datos |
| Observaciones: | |

Tabla 15. Tarjetas CRC - Editar datos de usuario

| Editar datos de usuario | |
|---|---|
| Responsabilidad | Colaboradores |
| Recibir parámetros. Validar parámetros Editar información de usuario. | Capa de presentación Controlador Métodos de lógica de negocio Capa de acceso a datos |
| Observaciones: | |

Tabla 16. Tarjetas CRC - Cambiar contraseña

| Cambiar contraseña | |
|--------------------------------|------------------------------|
| Responsabilidad | Colaboradores |
| Recibir parámetros. | Capa de presentación |
| Validar parámetros | Controlador |
| Cambiar contraseña de usuario. | Métodos de lógica de negocio |
| | Capa de acceso a datos |
| Observaciones: | |

Tabla 17. Tarjetas CRC - Gestionar cliente

| Gestionar cliente | |
|--------------------------------|------------------------------|
| Responsabilidad | Colaboradores |
| Recibir parámetros. | Capa de presentación |
| Validar parámetros | Controlador |
| Guardar parámetros de cliente. | Métodos de lógica de negocio |
| | Capa de acceso a datos |
| Observaciones: | |

Tabla 18. Tarjetas CRC - Gestionar proyecto

| Gestionar proyecto | |
|---------------------------|----------------------|
| Responsabilidad | Colaboradores |

| | |
|---------------------------------|------------------------------|
| Recibir parámetros. | Capa de presentación |
| Validar parámetros | Controlador |
| Guardar parámetros de proyecto. | Métodos de lógica de negocio |
| | Capa de acceso a datos |
| Observaciones: | |

Tabla 19. Tarjetas CRC - Gestionar actividad

| Gestionar actividad | |
|----------------------------------|------------------------------|
| Responsabilidad | Colaboradores |
| Recibir parámetros. | Capa de presentación |
| Validar parámetros | Controlador |
| Guardar parámetros de actividad. | Métodos de lógica de negocio |
| | Capa de acceso a datos |
| Observaciones: | |

Tabla 20. Tarjetas CRC - Generar reportes de desarrollador

| Generar reportes de desarrollador | |
|--|------------------------------|
| Responsabilidad | Colaboradores |
| Recibir parámetros. | Capa de presentación |
| Validar parámetros | Controlador |
| Retornar datos de reporte. | Métodos de lógica de negocio |
| | Capa de acceso a datos |
| Observaciones: | |

Tabla 21. Tarjetas CRC - Visualizar actividades

| Visualizar actividades | |
|---|---|
| Responsabilidad | Colaboradores |
| Recibir parámetros. Validar parámetros Retornar datos de actividades. | Capa de presentación Controlador Métodos de lógica de negocio Capa de acceso a datos |
| Observaciones: | |

Tabla 22. Tarjetas CRC - Registrar observación

| Registrar observación | |
|---|---|
| Responsabilidad | Colaboradores |
| Recibir parámetros. Validar parámetros Guardar parámetros de observación. | Capa de presentación Controlador Métodos de lógica de negocio Capa de acceso a datos |
| Observaciones: | |

Tabla 23. Tarjetas CRC - Editar observación

| Editar observación | |
|---------------------------|----------------------|
| Responsabilidad | Colaboradores |
| | |

| | |
|------------------------------------|------------------------------|
| Recibir parámetros. | Capa de presentación |
| Validar parámetros | Controlador |
| Editar información de observación. | Métodos de lógica de negocio |
| | Capa de acceso a datos |
| Observaciones: | |

Tabla 24. Tarjetas CRC - Notificar información de proyecto

| Notificar información de proyecto | |
|--|------------------------------|
| Responsabilidad | Colaboradores |
| Notificar información de proyecto. | Capa de presentación |
| | Controlador |
| | Métodos de lógica de negocio |
| Observaciones: | |

Tabla 25. Tarjetas CRC - Generar reportes de cliente

| Generar reportes de cliente | |
|------------------------------------|------------------------------|
| Responsabilidad | Colaboradores |
| Recibir parámetros. | Capa de presentación |
| Validar parámetros | Controlador |
| Retornar datos de reporte. | Métodos de lógica de negocio |
| | Capa de acceso a datos |

Observaciones:

3.2.4.2. Listado de requerimientos funcionales

En base al análisis y procesamiento de las historias de usuario se presenta la siguiente tabla con los requerimientos funcionales, descripción, prioridad y riesgo:

Tabla 26. Tabla de requerimientos funcionales

| ID | Requerimiento | Descripción | Prioridad | Riesgo |
|-------|-------------------------------|--|-----------|--------|
| RF-01 | Autenticación de usuarios. | Se debe ingresar el usuario y contraseña para ingresar a la aplicación web. | Alta | Alto |
| | | Al ingresar, se generará un JWT que permitirá autenticar cada petición de los usuarios. | | |
| | | Al ingresar, se validará el rol del usuario para control de los accesos. | | |
| RF-02 | Crear usuarios. | El administrador, podrá crear usuarios para quienes requieran acceder a la aplicación, y editar información no personal del usuario. | Alta | Medio |
| | | En la base de datos se almacenarán las contraseñas cifradas en hash, mas no como texto plano. | | |
| RF-03 | Actualizar datos de usuarios. | Cada usuario, podrá acceder a cambiar sus información | Alta | Alto |

| ID | Requerimiento | Descripción | Prioridad | Riesgo |
|--------------|------------------------------|--|------------------|---------------|
| | | personal y contraseña de acceso. | | |
| RF-04 | Gestionar clientes. | Como Gerente de operaciones, puedo acceder a crear, actualizar o eliminar clientes. | Alta | Alto |
| RF-05 | Gestionar proyectos. | Como líder de proyecto, puedo acceder a crear, actualizar o eliminar proyectos, de los cuales yo estaré a cargo. | Alta | Alto |
| RF-06 | Gestionar actividades. | Como desarrollador, puedo acceder a crear, actualizar o eliminar mis actividades diarias. Como líder de proyecto, puedo visualizar las actividades de los proyectos de los que estoy a cargo. | Alta | Alto |
| RF-07 | Registrar observaciones. | Como líder de proyecto, puedo registrar observaciones de las actividades que así lo requieran. Las observaciones serán notificadas al desarrollador que corresponda. | Alta | Alto |
| RF-08 | Gestionar las observaciones. | Como desarrollador, puedo visualizar y marcar como resuelta una observación. Como Líder de Proyecto, puedo visualizar de manera | Alta | Medio |

| ID | Requerimiento | Descripción | Prioridad | Riesgo |
|--------------|----------------------|--|------------------|---------------|
| | | general las observaciones de mis proyectos. | | |
| RF-09 | Notificar proyectos. | Como líder de proyecto, puedo notificar al Gerente de operaciones que las actividades de un proyecto se encuentran revisadas y corregidas. | Alta | Medio |
| RF-10 | Generar reportes. | Como Gerente de operaciones, puedo generar reportes de las actividades de un mes, año y por cliente. | Alta | Alto |
| | | Como desarrollador, puedo generar reportes de mis actividades de un mes y un año en específico. | | |
| | | Los reportes se podrán descargar como PDF desde la aplicación web. | | |

3.2.4.3. Listado de requerimientos no funcionales

Se presenta la siguiente tabla con los requerimientos no funcionales y su descripción:

Tabla 27. Tabla de requerimientos no funcionales

| ID | Requerimiento | Descripción |
|---------------|----------------------|--|
| RNF-01 | Seguridad | La aplicación web debe garantizar la seguridad y protección de los datos mediante la autenticación de usuarios, encriptación de credenciales y control de acceso a las funciones de la aplicación. |

| ID | Requerimiento | Descripción |
|---------------|----------------------|---|
| RNF-02 | Rendimiento | Se debe tener la capacidad de manejar las transacciones de la aplicación de manera eficiente, así como la generación de reportes, asegurando buenos tiempos de respuesta. |
| RNF-03 | Usabilidad | Se requiere que la aplicación web implemente las mejores prácticas de UI/UX, permitiendo que el uso de esta sea fácil e intuitivo para los usuarios finales. |
| RNF-04 | Escalabilidad | La aplicación web debe implementar una solución de despliegue como Contenedores para permitir el escalamiento horizontal de la aplicación, sin afectar al rendimiento de la aplicación. |
| RNF-05 | Mantenimiento | Las actualizaciones deberán documentarse y llevar un correcto control de versiones del código fuente. |
| RNF-06 | Disponibilidad | En horas laborales, la aplicación web deberá estar disponible al 100% para todos los usuarios finales, el objetivo es mantener al mínimo los tiempos de inactividad. |
| RNF-07 | Interoperabilidad | Se debe mantener compatibilidad con los principales navegadores web mediante los cuales los usuarios podrán acceder a la aplicación web. |

3.2.4.4. Diseño de la base de datos

Para garantizar la consistencia y validez de la información que se maneje en la aplicación web, se diseñó una base de datos relacional que permitirá el acceso a la información de manera concisa, organizada y correctamente estructurada.

Para implementar esta base de datos se utilizó PostgreSQL como gestor de base de datos, con un total de 5 tablas que manejarán toda la gestión y control de la información laboral de la empresa.

Del proceso de diseño de la base de datos se obtuvo el siguiente Diagrama Entidad Relación:

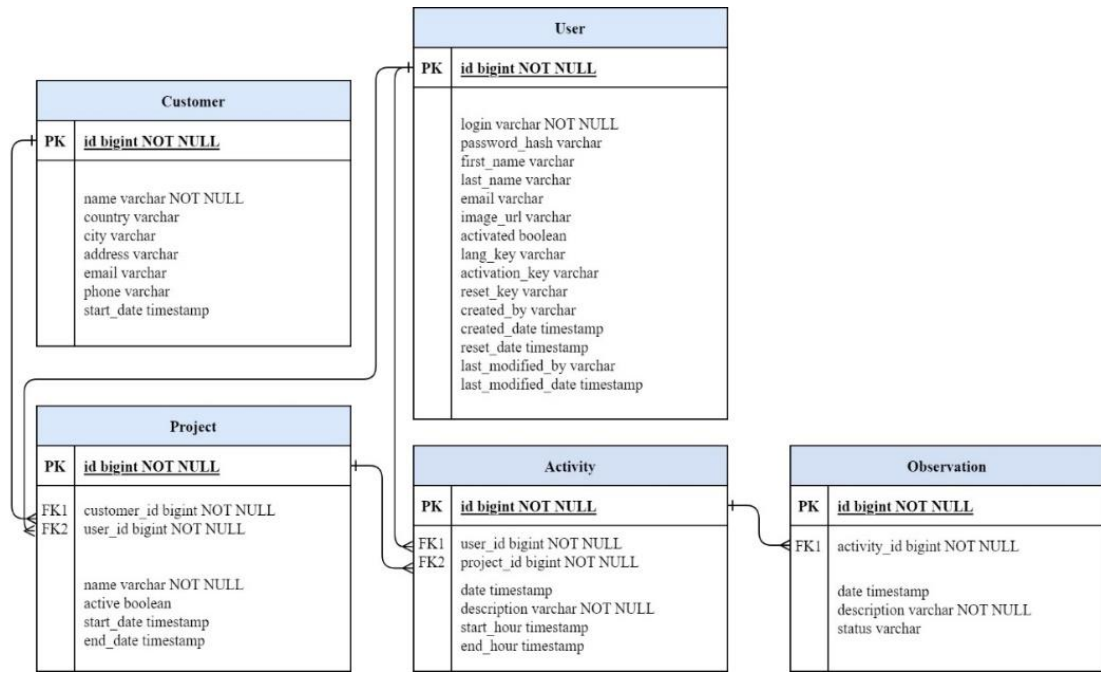


Figura 2. Diagrama Entidad Relación

3.2.4.5. Arquitectura de la aplicación web

Para el desarrollo de esta aplicación web se utilizó una arquitectura monolítica, es decir, toda la implantación de sus funcionalidades y componentes están agrupados en una única unidad de código. Debido a la necesidad de rapidez y simplicidad en el desarrollo de este proyecto de investigación, una arquitectura monolítica es lo más adecuado, además, se adapta perfectamente a la metodología de desarrollo escogida Extreme Programming.

Para esta aplicación web monolítica se diseñó una arquitectura multicapa, con la siguiente estructura:

Capa de presentación

Esta capa es la interfaz de usuario de la aplicación y se encarga de mostrar la información al usuario final. Puede consistir en páginas HTML, hojas de estilo CSS y scripts del lado del cliente.

Controlador

Este componente actúa como un intermediario entre la capa de presentación y la capa de lógica de negocio. En esta aplicación web monolítica bajo una interfaz REST, el controlador maneja las solicitudes HTTP (GET, POST, PUT, DELETE) y dirige el flujo de la aplicación en función de esas solicitudes.

Capa de negocio

Aquí reside la lógica principal de la aplicación. Se encarga de procesar los datos recibidos, realizar cálculos y aplicar reglas de negocio. La capa de lógica de negocio se comunica con la capa de acceso a datos para recuperar o almacenar información según sea necesario.

Capa de acceso a datos

Esta capa se encarga de interactuar con la base de datos. En este punto se realizan operaciones de lectura y escritura en la base de datos.

En la Figura 3 se muestra la arquitectura de la aplicación web.

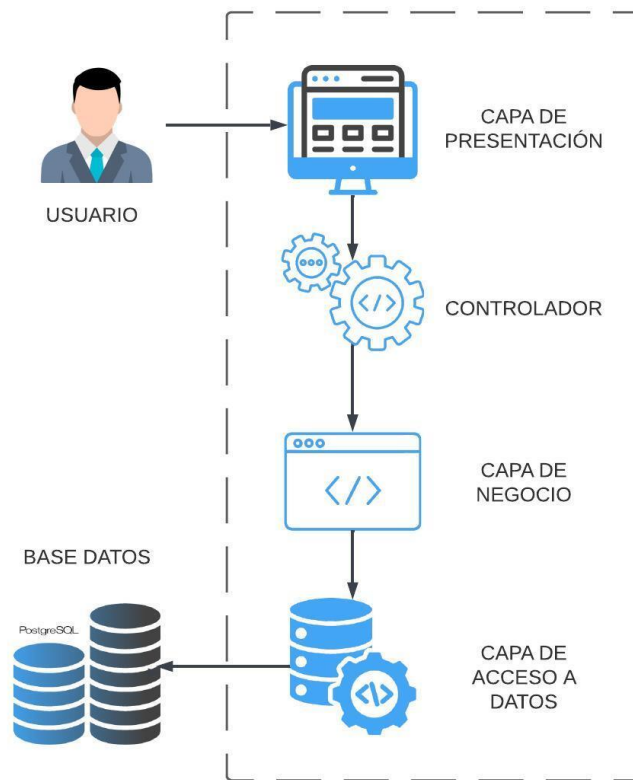


Figura 3. Arquitectura de la aplicación web

3.2.4.6. Prototipado de la aplicación web

Se elaboraron prototipos de las vistas principales de la aplicación web para posteriormente recibir la retroalimentación del cliente y realizar las mejoras necesarias para obtener un resultado de calidad.

Página de inicio

En la Figura 4 se muestra la página de inicio y bienvenida a la aplicación web donde se encuentra información relevante sobre la aplicación y se le proporciona al usuario un acceso directo que permite ir a iniciar sesión.



Bienvenido al registro de actividades de la empresa Software Evolutivo!

Da clic [aquí](#) para iniciar sesión

Figura 4. Prototipo de la página de inicio de la aplicación web

Autenticación de usuarios

En la Figura 5 se muestra el modal en el que el usuario va a poder iniciar sesión, agregando correctamente sus credenciales.

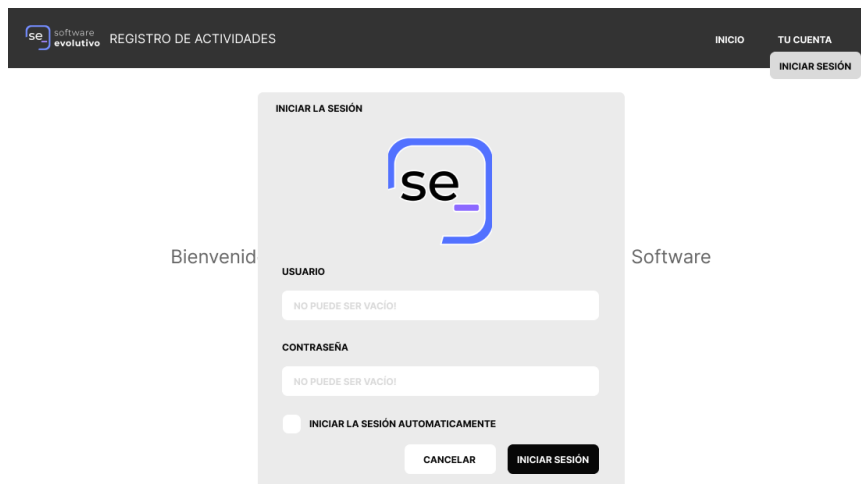


Figura 5. Prototipo de la interfaz de autenticación de usuarios

Gestión de usuarios

En la Figura 6 se muestra la interfaz de la gestión de usuarios a la cual solo los administradores podrán acceder.

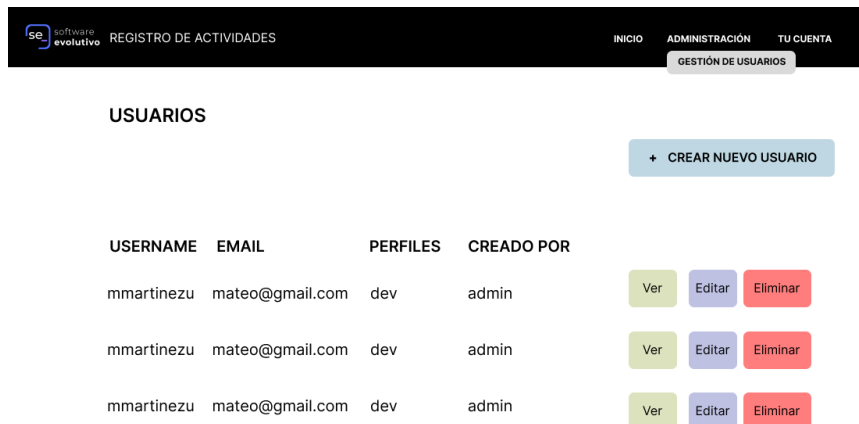


Figura 6. Prototipo de la interfaz de gestión de usuarios

Creación o edición de usuarios

En la Figura 7 se muestran la interfaz a la que tendrán acceso los administradores para la creación o edición de usuarios con los campos respectivos.

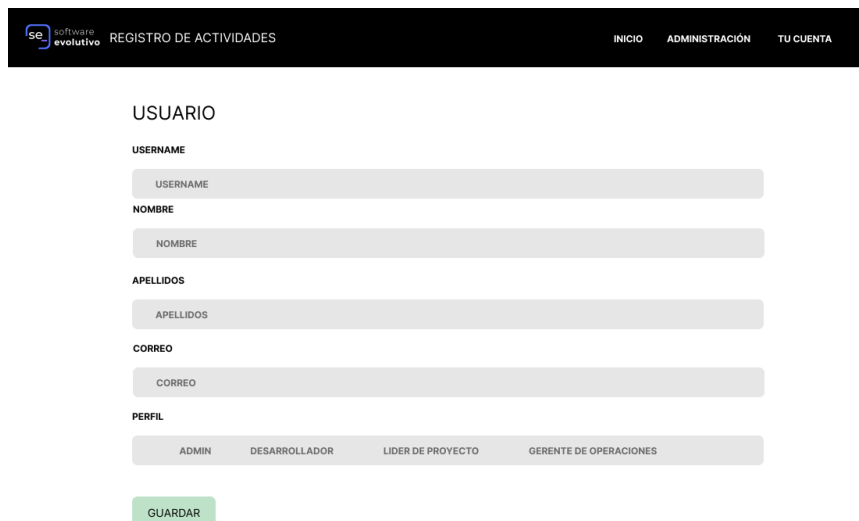


Figura 7. Prototipo de la interfaz de creación o edición de usuarios

Actualización de datos personales

En la Figura 8 se muestra la interfaz de actualización de datos personales a la cual tendrán acceso los usuarios que no sean administradores.



se software evolutivo REGISTRO DE ACTIVIDADES INICIO ACTIVIDADES TU CUENTA

AJUSTES
CONTRASEÑA
CERRAR SESIÓN

AJUSTES DE "USUARIO123"

NOMBRE

NOMBRE ACTUAL

APELLIDOS

APELLIDOS ACTUALES

CORREO

CORREO ACTUAL

GUARDAR

Figura 8. Prototipo de la interfaz de actualización de datos personales de usuario

Cambio de contraseña

En la Figura 9 se muestra la interfaz de actualización de la contraseña de acceso para que cada usuario pueda hacerlo cuando así lo requiera.



se software evolutivo REGISTRO DE ACTIVIDADES INICIO ACTIVIDADES TU CUENTA

AJUSTES
CONTRASEÑA
CERRAR SESIÓN

CONTRASEÑA DE "USUARIO123"

CONTRASEÑA ACTUAL

INGRESE LA CONTRASEÑA ACTUAL

CONTRASEÑA NUEVA

INGRESE LA CONTRASEÑA NUEVA

CONFIRME LA CONTRASEÑA NUEVA

VUELVA A INGRESAR LA CONTRASEÑA NUEVA

GUARDAR

Figura 9. Prototipo de la interfaz de actualización de contraseña de usuarios

Gestión de clientes

En la Figura 10 se muestra la interfaz de gestión de clientes a la que solo los usuarios con rol Gerente de Operaciones podrán acceder.

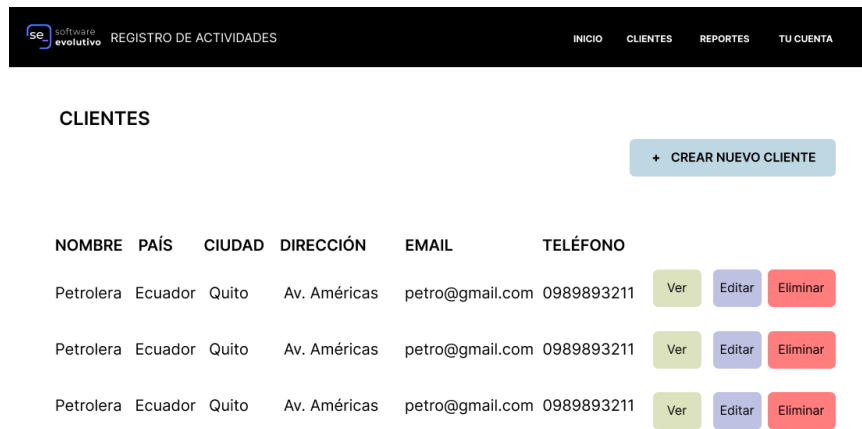


Figura 10. Prototipo de la interfaz de gestión de clientes

Creación o edición de clientes

En la Figura 11 se muestra la interfaz de creación o edición de clientes de la empresa, así mismo, solo los usuarios con el rol Gerente de Operaciones podrán acceder a esta interfaz.

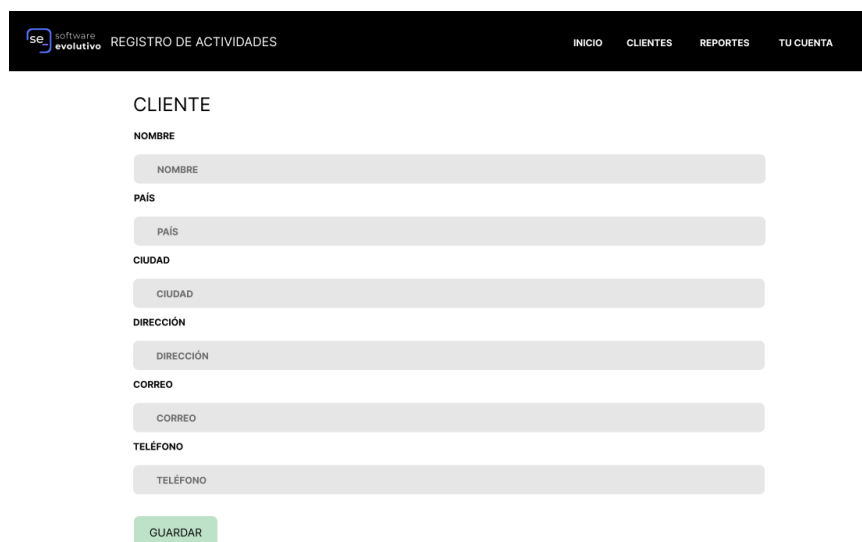


Figura 11. Prototipo de la interfaz de creación o edición de clientes

Gestión de proyectos

En la Figura 12 se muestra la interfaz de gestión de proyectos, a la que solo los usuarios con rol de Líder de Proyecto podrán acceder.



Figura 12. Prototipo de la interfaz de gestión de proyectos

Creación o edición de proyectos

En la Figura 13 se muestra la interfaz de creación o edición de proyectos, a la cual solo tendrán acceso los usuarios con rol Líder de Proyecto.

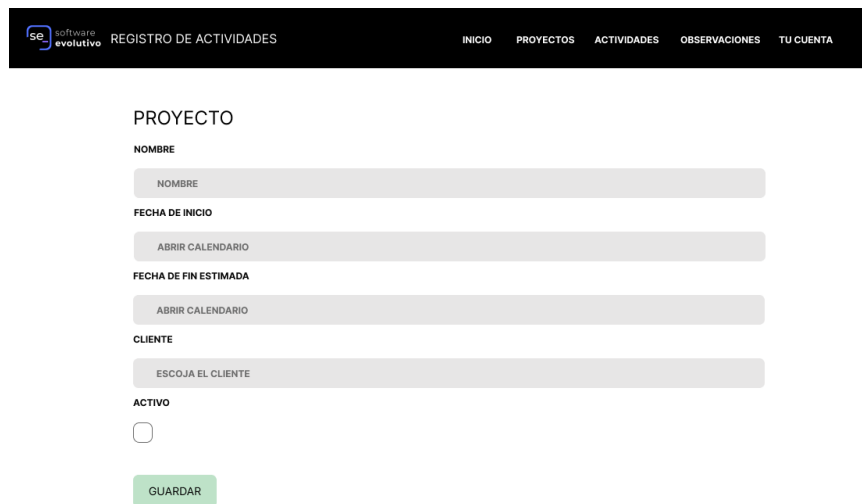
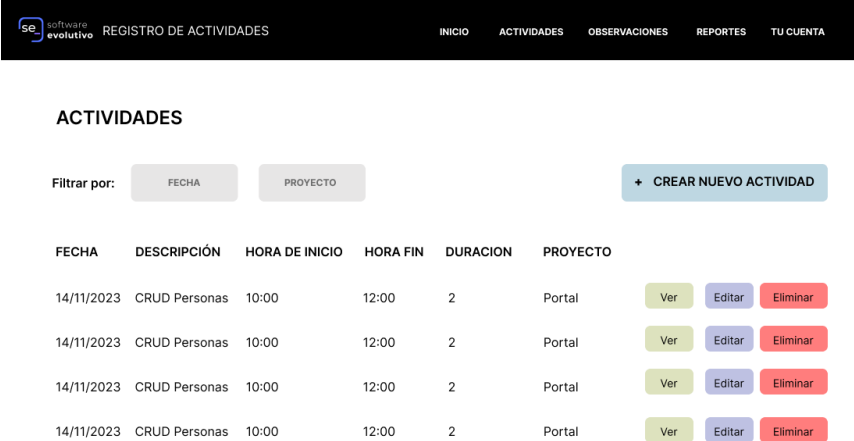


Figura 13. Prototipo de la interfaz de creación o edición de proyectos

Gestión de Actividades

En la Figura 14 se muestra la interfaz de gestión de Actividades, a la cual podrán acceder los usuarios con rol Desarrollador y Líder de Proyecto, este último con privilegios reducidos solo de visualización.



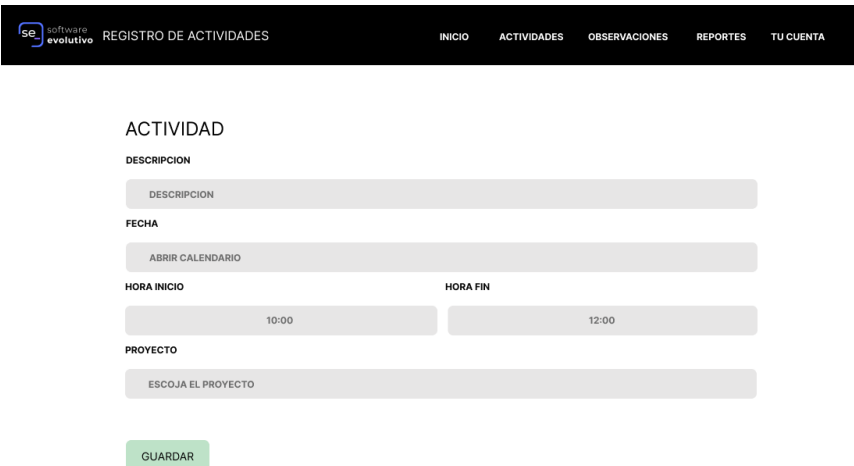
The screenshot shows a web interface for 'REGISTRO DE ACTIVIDADES'. At the top, there is a navigation bar with the logo 'se software evolutivo' and menu items: INICIO, ACTIVIDADES, OBSERVACIONES, REPORTES, and TU CUENTA. Below the navigation bar, the title 'ACTIVIDADES' is displayed. There are two filter buttons: 'FECHA' and 'PROYECTO', and a '+ CREAR NUEVO ACTIVIDAD' button. The main content is a table with the following columns: FECHA, DESCRIPCIÓN, HORA DE INICIO, HORA FIN, DURACION, and PROYECTO. Each row in the table has three action buttons: 'Ver' (green), 'Editar' (blue), and 'Eliminar' (red).

| FECHA | DESCRIPCIÓN | HORA DE INICIO | HORA FIN | DURACION | PROYECTO | | | |
|------------|---------------|----------------|----------|----------|----------|-----|--------|----------|
| 14/11/2023 | CRUD Personas | 10:00 | 12:00 | 2 | Portal | Ver | Editar | Eliminar |
| 14/11/2023 | CRUD Personas | 10:00 | 12:00 | 2 | Portal | Ver | Editar | Eliminar |
| 14/11/2023 | CRUD Personas | 10:00 | 12:00 | 2 | Portal | Ver | Editar | Eliminar |
| 14/11/2023 | CRUD Personas | 10:00 | 12:00 | 2 | Portal | Ver | Editar | Eliminar |

Figura 14. Prototipo de la interfaz de gestión de Actividades

Creación o edición de Actividades

En la Figura 15 se muestra la interfaz de creación o edición de las Actividades, a la cual podrán acceder solo los usuarios con rol Desarrollador.



The screenshot shows a web interface for 'ACTIVIDAD'. At the top, there is a navigation bar with the logo 'se software evolutivo' and menu items: INICIO, ACTIVIDADES, OBSERVACIONES, REPORTES, and TU CUENTA. Below the navigation bar, the title 'ACTIVIDAD' is displayed. The form contains the following fields: 'DESCRIPCION' (text input), 'FECHA' (calendar icon), 'HORA INICIO' (time input, 10:00), 'HORA FIN' (time input, 12:00), and 'PROYECTO' (dropdown menu, ESCOJA EL PROYECTO). There is a 'GUARDAR' button at the bottom.

Figura 15. Prototipo de la interfaz de creación o edición de Actividades

Generación de reportes para Desarrollador

En la Figura 16 se muestra la interfaz que van a tener disponible los usuarios con rol de Desarrollador para generar reportes de su trabajo, pudiendo escoger el año y el mes del cual se requiera el reporte.



El prototipo muestra una interfaz web con un encabezado negro que contiene el logo 'se software evolutivo' y el título 'REGISTRO DE ACTIVIDADES'. A la derecha del encabezado hay un menú de navegación con los ítems: 'INICIO', 'ACTIVIDADES', 'OBSERVACIONES', 'REPORTES' (destacado) y 'TU CUENTA'. El contenido principal está titulado 'REPORTES' y contiene un formulario con el título 'Parámetros para su reporte:'. Este formulario tiene dos campos de selección: 'SELECCIONE EL AÑO' y 'SELECCIONE EL MES', ambos con botones grises. A la derecha de estos campos hay un botón rojo con el texto 'COMO PDF'. Debajo del formulario hay un recuadro gris con el texto: 'AQUÍ SE MOSTRARÁ LA PREVISUALIZACIÓN DEL REPORTE GENERADO'.

Figura 16. Prototipo de la interfaz de generación de reportes para Desarrollador

Visualización de Actividades para Líder de Proyecto

En la Figura 17 se muestra el prototipo de visualización de las Actividades para los usuarios con el rol de Líder de Proyecto, en esta todos los campos son solo de lectura y se muestra un botón “Registrar Observación”.

software evolutivo REGISTRO DE ACTIVIDADES INICIO PROYECTOS ACTIVIDADES OBSERVACIONES TU CUENTA

ACTIVIDAD

DESCRIPCION

INFORMACION DE LA DESCRIPCION GUARDADA POR EL DESARROLLADOR

FECHA

14/11/2023

HORA INICIO HORA FIN

10:00 12:00

PROYECTO

PORTAL

ATRÁS REGISTRAR OBSERVACION

Figura 17. Prototipo de la interfaz de vista de Actividades para Líder de Proyecto

Registro de observación

En la Figura 18 se muestra la interfaz del modal donde el Líder de Proyecto podrá escribir una observación a la Actividad respectiva.

software evolutivo REGISTRO DE ACTIVIDADES INICIO PROYECTOS ACTIVIDADES OBSERVACIONES TU CUENTA

ACTIVIDAD

DESCRIPCION

REGISTRAR LA OBSERVACIÓN

INFORMA

FECHA

14/11/2023

HORA INICIO

PROYECTO

PORTAL

ATRÁS

NO PUEDE SER VACÍO

CANCELAR ENVIAR OBSERVACIÓN

Figura 18. Prototipo de la interfaz de registro de observaciones

Gestión de observaciones

En la Figura 19 se muestra la interfaz de gestión de las observaciones a la cual tendrán acceso los usuarios con rol Desarrollador y Líder de Proyecto, con las restricciones a la información correspondiente.



| FECHA | DESCRIPCIÓN | ESTADO | ACTIVIDAD RELACIONADA | | |
|------------|----------------------|----------|-----------------------|-----|--------|
| 14/11/2023 | Corregir el proyecto | NUEVA | CRUD Personas | Ver | Editar |
| 14/11/2023 | Corregir faltas | RESUELTA | CRUD Personas | Ver | Editar |
| 14/11/2023 | Corregir la hora | NUEVA | CRUD Personas | Ver | Editar |
| 14/11/2023 | Corregir el proyecto | NUEVA | CRUD Personas | Ver | Editar |

Figura 19. Prototipo de la interfaz de gestión de las observaciones

Edición de las observaciones

En la Figura 20 se muestra la interfaz de edición de las observaciones a la cual tendrán acceso los usuarios con rol Desarrollador y Líder de Proyecto, el primero solo podrá marcar como resuelta la observación y el segundo podrá editar la información.

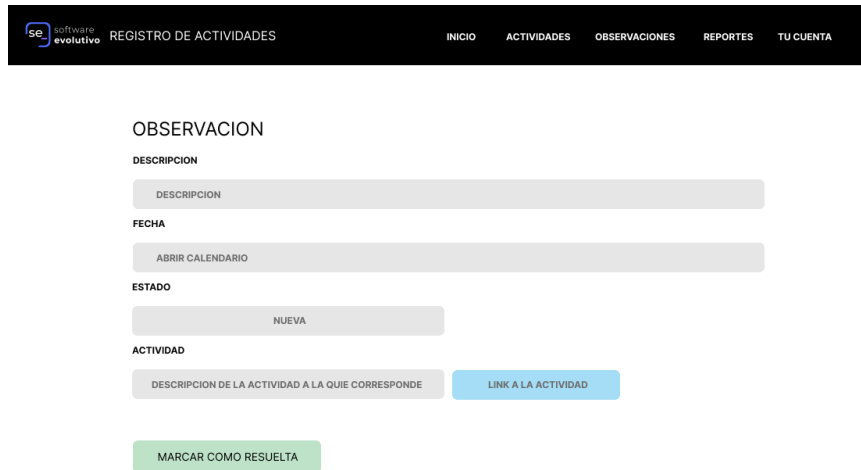


Figura 20. Prototipo de la interfaz de edición de observaciones

Generación de reportes para Gerente de Operaciones

En la Figura 21 se muestra la interfaz de generación de reportes a la que solo los usuarios con rol de Gerente de Operaciones podrán acceder, en esta se muestran los parámetros disponibles para poder generar el reporte, el cual se mostrará como previsualización antes de poder descargarlo como PDF.

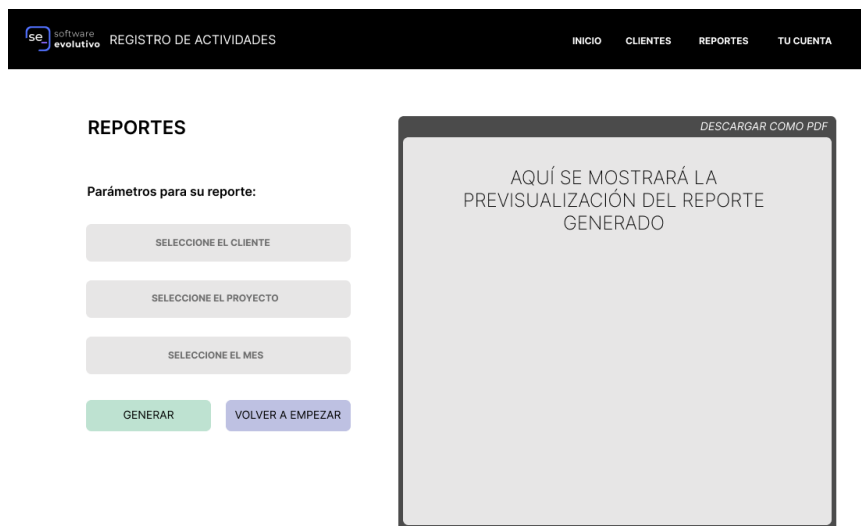


Figura 21. Prototipo de la interfaz de generación de reportes

3.2.5. Fase III: Desarrollo

3.2.5.1. Implantación de la aplicación web

Para la creación del proyecto base se utilizó el generador por la línea de comandos de KHipster.

```
[mateomartinez@fedora se-actilog]$ khipster
```

Figura 22. Comando para crear el proyecto con Khipster



Figura 23. Asistente de KHipster para la creación del proyecto base

Para el desarrollo de las funcionalidades más importantes de esta aplicación web se realizaron pruebas de integración en un ambiente de testing, las mismas que permitieron garantizar al usuario final la fiabilidad y consistencia de cada proceso que se lleve sobre la aplicación, es por eso que, el desarrollo guiado por pruebas es una de las mejores prácticas para el desarrollo de software.

Bienvenida a la aplicación web

En la Figura 24 se puede observar al ingresar a la aplicación, esta recibe al usuario con un acceso directo al inicio de sesión, un mensaje de bienvenida y el aviso de que el uso de sus datos es únicamente con fines de mejorar su experiencia de uso de la aplicación y que su información no será compartida.

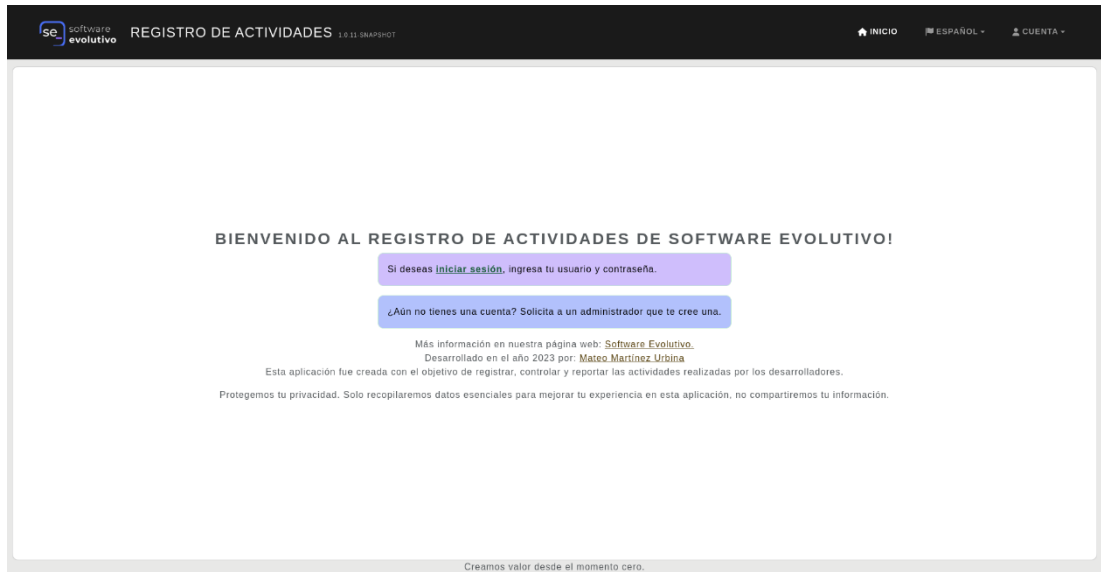


Figura 24. Pantalla de bienvenida a la aplicación web

Inicio de sesión

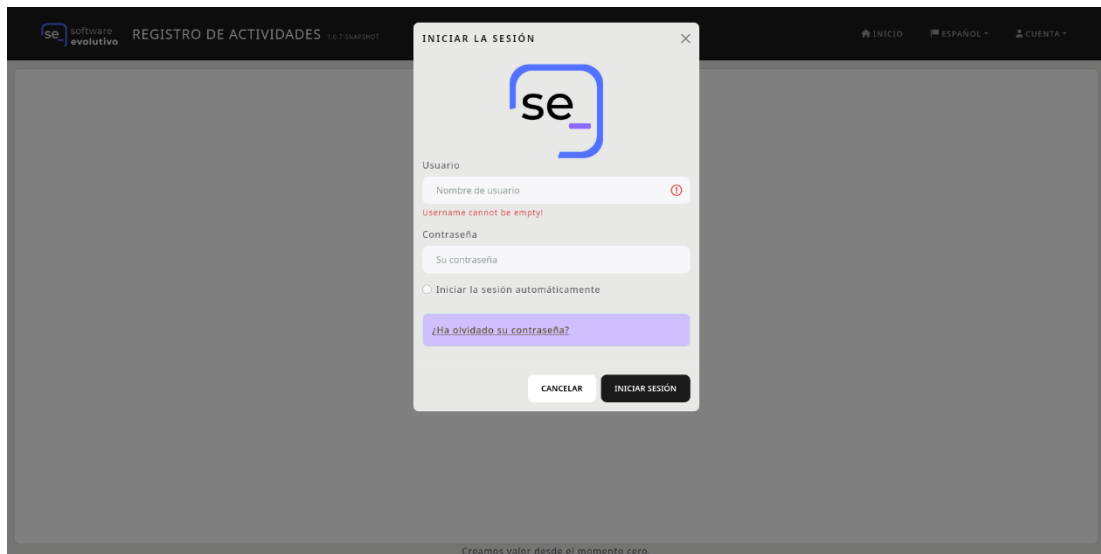


Figura 25. Interfaz de inicio de sesión

En la Figura 26 se encuentra método authorize es una petición de tipo POST que autentica a los usuarios en la aplicación web. Recibe el nombre de usuario y la contraseña, crea un token de autenticación y lo valida. Tras la autenticación exitosa, genera un token JWT (JSON Web Token) que se incluye en las respuestas futuras para representar una sesión de usuario válida. Este token JWT se retorna en el cuerpo y

encabezados de la respuesta HTTP. Si la autenticación falla, se lanza una excepción. Los tests de integración de esta función se encuentran en el *Anexo D*.

```
@PostMapping("/authenticate")
fun authorize(@Valid @RequestBody loginVM: LoginVM): ResponseEntity<JWTToken> {

    val authenticationToken = UsernamePasswordAuthenticationToken(loginVM.username, loginVM.password)

    val authentication = authenticationManagerBuilder.getObject().authenticate(authenticationToken)
    SecurityContextHolder.getContext().authentication = authentication
    val jwt = tokenProvider.createToken(authentication, rememberMe: loginVM.isRememberMe ?: false)
    val httpHeaders = HttpHeaders()
    httpHeaders.add(JWTFilter.AUTHORIZATION_HEADER, headerValue: "Bearer $jwt")
    return ResponseEntity(JWTToken(jwt), httpHeaders, HttpStatus.OK)
}
```

Figura 26. Método para el inicio de sesión

Validación de rol Desarrollador

En la Figura 27 se muestra el método que será reutilizado en las diferentes instancias de la aplicación web donde se requiera validar si es que el usuario conectado tiene el rol de Desarrollador o no. Los tests de integración de esta función se encuentran en el *Anexo E*.

```
fun userIsDeveloper(): Boolean {
    log.debug("Validating if user has role developer")

    val authentication = SecurityContextHolder.getContext().authentication
    return authentication.authorities.any { it.authority == "ROLE_DEVELOPER" }
}
```

Figura 27. Método de validación de rol Desarrollador

Validación de rol Líder de Proyecto

En la Figura 28 se muestra el método que será reutilizado en las diferentes instancias de la aplicación web donde se requiera validar si es que el usuario conectado tiene el rol de Líder de Proyecto o no. Los tests de integración de esta función se encuentran en el *Anexo F*.

```

fun userIsProjectLeader(): Boolean {
    log.debug("Validating if user has role project leader")

    val authentication = SecurityContextHolder.getContext().authentication
    return authentication.authorities.any { it.authority == "ROLE_PROJECT_LEADER" }
}

```

Figura 28. Método de validación de rol Líder de Proyecto

Validación de rol Gerente de Operaciones

En la Figura 29 se muestra el método que será reutilizado en las diferentes instancias de la aplicación web donde se requiera validar si es que el usuario conectado tiene el rol de Gerente de Operaciones o no. Los tests de integración de esta función se encuentran en el *Anexo G*.

```

fun userIsOpsManager(): Boolean {
    log.debug("Validating if user has role operations manager")

    val authentication = SecurityContextHolder.getContext().authentication
    return authentication.authorities.any { it.authority == "ROLE_OPS_MANAGER" }
}

```

Figura 29. Método de validación de rol Gerente de Operaciones

Validación de rol Administrador

En la Figura 30 se muestra el método que será reutilizado en las diferentes instancias de la aplicación web donde se requiera validar si es que el usuario conectado tiene el rol de Administrador o no. Los tests de integración de esta función se encuentran en el *Anexo H*.

```

fun userIsAdmin(): Boolean {
    log.debug("Validating if user has role admin")

    val authentication = SecurityContextHolder.getContext().authentication
    return authentication.authorities.any { it.authority == "ROLE_ADMIN" }
}

```

Figura 30. Método de validación de rol Administrador

CRUD de las entidades

Para el manejo de las entidades de la aplicación web tanto en creación, actualización, visualización y eliminación se usan las diferentes capas de la arquitectura de la aplicación web, es decir, el controlador, la capa de lógica de negocio y la capa de acceso a datos. A continuación, se tomó como ejemplo a una de las entidades para poder evidenciar cómo es el manejo del CRUD en todas las entidades:

En la Figura 31 y Figura 32 se muestra el proceso que se realiza para la creación de un nuevo registro de una entidad, se recibe mediante un método POST la información del nuevo registro, se valida que no se haya asignado un ID debido a que el Entity Manager debe asignarlo automáticamente para evitar problemas en base de datos, una vez que se ha validado, el controlador accede a la capa de lógica de negocio la misma que a su vez accede a la capa de acceso a datos para enviar y guardar el nuevo registro en la base de datos. Los tests de integración para la creación de un nuevo registro de una entidad se encuentran en el *Anexo I*.

```

@PostMapping(@"*/customers")
fun createCustomer(@Valid @RequestBody customer: Customer): ResponseEntity<Customer> {
    log.debug("REST request to save Customer : $customer")
    if (customer.id != null) {
        throw BadRequestAlertException(
            "A new customer cannot already have an ID",
            ENTITY_NAME, "idexists"
        )
    }
    val result = customerService.save(customer)
    return ResponseEntity.created(URI( str: "/api/customers/${result.id}"))
        .headers(HeaderUtil.createEntityCreationAlert(applicationName, enableTranslation: true, ENTITY_NAME, result.id.toString()))
        .body(result)
}

```

Figura 31. Método POST para la creación de un nuevo registro de una entidad.


```

fun save(customer: Customer): Customer {
    log.debug("Request to save Customer : $customer")
    return customerRepository.save(customer)
}

```

Figura 32. Capa de lógica de negocio para crear un nuevo registro de una entidad.

En la Figura 33 y Figura 34 se muestra el proceso que se realiza para la actualización de un registro de una entidad, se recibe la petición mediante un método PUT que recibe el ID y la información de la entidad que corresponde, para posteriormente validar que el ID no sea nulo, no sea inválido y que la entidad se encuentre en la base de datos. Si todo se valida correctamente se accede a la capa de lógica de negocio, la misma que a su vez llama a la capa de acceso a datos para solicitar que se guarde el registro de la entidad con los datos actualizados. Los tests de integración para la creación de un nuevo registro de una entidad se encuentran en el *Anexo J*.

```

@PutMapping("/{customers/{id}}")
fun updateCustomer(
    @PathVariable(value = "id", required = false) id: Long,
    @Valid @RequestBody customer: Customer
): ResponseEntity<Customer> {
    log.debug("REST request to update Customer : {}, {}", id, customer)
    if (customer.id == null) {
        throw BadRequestAlertException("Invalid id", ENTITY_NAME, "idnull")
    }

    if (!Objects.equals(id, customer.id)) {
        throw BadRequestAlertException("Invalid ID", ENTITY_NAME, "idinvalid")
    }

    if (!customerRepository.existsById(id)) {
        throw BadRequestAlertException("Entity not found", ENTITY_NAME, "idnotfound")
    }

    val result = customerService.update(customer)
    return ResponseEntity.ok()
        .headers(
            HeaderUtil.createEntityUpdateAlert(
                applicationName, enableTranslation: true, ENTITY_NAME,
                customer.id.toString()
            )
        )
        .body(result)
}

```

Figura 33. Método PUT para la actualización de un registro de una entidad.

```

fun update(customer: Customer): Customer {
    log.debug("Request to update Customer : {}", customer)
    return customerRepository.save(customer)
}

```

Figura 34. Capa de lógica de negocio para actualizar un registro de una entidad.

En la Figura 35, Figura 36 y Figura 37 se muestra el proceso que se sigue para la obtención de los registros paginados de una entidad, el controlador recibe la petición mediante un método GET con los criterios de búsqueda y la información para la paginación, estos datos son enviados a la capa de lógica de negocio que primero se encarga de crear una especificación con los criterios de búsqueda requeridos y solicita a otro método para que acceda a la capa de acceso a datos y se pueda retornar la información solicitada. Los tests de integración de este proceso se encuentran en el *Anexo K*.

```

@GetMapping(@"*/customers") fun getAllCustomers(
    criteria: CustomerCriteria,
    @org.springframework.web.servlet.mvc.annotation.annotation.DefaultPageable
): ResponseEntity<MutableList<Customer>> {
    log.debug("REST request to get Customers by criteria: $criteria")
    val page = customerQueryService.findByCriteria(criteria, pageable)
    val headers = PaginationUtil.generatePaginationHttpHeaders(ServletUriComponentsBuilder.fromCurrentRequest(), page)
    return ResponseEntity.ok().headers(headers).body(page.content)
}

```

Figura 35. Método GET para la obtención de los registros de la entidad.

```

@Transactional(readOnly = true)
fun findByCriteria(criteria: CustomerCriteria?, page: Pageable): Page<Customer> {
    log.debug("find by criteria : $criteria, page: $page")
    val specification = createSpecification(criteria)
    return customerRepository.findAll(specification, page)
}

```

Figura 36. Capa de lógica de negocio para agregar criterios de búsqueda de registros.

```

@Transactional(readonly = true)
fun findAll(pageable: Pageable): Page<Customer> {
    log.debug("Request to get all Customers")
    return customerRepository.findAll(pageable)
}

```

Figura 37. Capa de lógica de negocio para la obtención de registros de una entidad.

En la Figura 38 y Figura 39 se muestra el proceso que se sigue para poder obtener un registro de una entidad. Para empezar el controlador recibe una petición GET en la cual recibe únicamente el ID de la entidad y accede a la capa de lógica de negocio, solicitando finalmente a la capa de acceso a datos que se retorne el registro de la entidad con el ID correspondiente. Si no hay información que mostrar el controlador sabrá cómo responder a la petición sin ningún problema. Los tests de integración de este proceso se encuentran en el *Anexo L*.

```

@GetMapping("/{id}")
fun getCustomer(@PathVariable id: Long): ResponseEntity<Customer> {
    log.debug("REST request to get Customer : $id")
    val customer = customerService.findOne(id)
    return ResponseUtil.wrapOrNotFound(customer)
}

```

Figura 38. Método GET para la obtención de un registro de una entidad.

```

@Transactional(readonly = true)
fun findOne(id: Long): Optional<Customer> {
    log.debug("Request to get Customer : $id")
    return customerRepository.findById(id)
}

```

Figura 39. Capa de lógica de negocio para obtener un registro de una entidad.

En la Figura 40 y Figura 41 se muestra el proceso para la eliminación de un registro de una entidad, el controlador recibe una petición DELETE con el ID del registro en cuestión, para acceder posteriormente a la capa de lógica de negocio, la misma que se comunicará con la capa de acceso a datos y si el registro existe se lo eliminará. Los tests de integración de este proceso se encuentran en el *Anexo M*.

```
@DeleteMapping("/{customers/{id}")
fun deleteCustomer(@PathVariable id: Long): ResponseEntity<Void> {
    log.debug("REST request to delete Customer : $id")

    customerService.delete(id)
    return ResponseEntity.noContent()
        .headers(HeaderUtil.createEntityDeletionAlert(applicationName, enableTranslation: true, ENTITY_NAME, id.toString()))
        .build()
}
```

Figura 40. Método DELETE para la eliminación de un registro de una entidad.

```
fun delete(id: Long) {
    log.debug("Request to delete Customer : $id")

    customerRepository.deleteById(id)
}
```

Figura 41. Capa de lógica de negocio para la eliminación de un registro de una entidad.

Gestión de los usuarios para Administradores

En la Figura 42 se muestra la interfaz que los usuarios con rol Administrador podrán visualizar sobre todos los usuarios de la aplicación web. La información presentada es únicamente información relevante para brindar una buena experiencia de uso en la aplicación. Se tienen disponibles opciones básicas para la gestión de los usuarios tales como creación, edición y eliminación de usuarios.

| ID | LOGIN | EMAIL | ESTADO | IDIOMA | PERFILES | FECHA DE CREACIÓN | MODIFICADO POR | FECHA DE MODIFICACIÓN | ACCIONES |
|----|-------------------|-----------------------------|----------|--------|---------------------|-------------------|----------------|-----------------------|-----------------------|
| 1 | admin | admin@localhost | ACTIVADO | en | ROLE ADMIN | | system | | VER, EDITAR, ELIMINAR |
| 2 | dev | dev@localhost | ACTIVADO | en | ROLE DEVELOPER | | system | | VER, EDITAR, ELIMINAR |
| 3 | projectlead | projectlead@localhost | ACTIVADO | en | ROLE PROJECT LEADER | | system | | VER, EDITAR, ELIMINAR |
| 4 | operationsmanager | operationsmanager@localhost | ACTIVADO | en | ROLE OPS MANAGER | | system | | VER, EDITAR, ELIMINAR |

Figura 42. Interfaz de visualización general de los usuarios

En la Figura 43 se muestra una opción con la que cuentan los usuarios con rol Administrador, esta les permite desactivar o retirar el acceso a la aplicación a un usuario, esta medida de seguridad causa que por más que el usuario ingrese correctamente sus credenciales, este no podrá acceder a la aplicación.

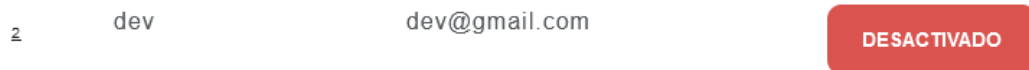


Figura 43. Opción de usuario desactivado

En la Figura 44 se presenta una interfaz con un formulario de la información más importante del usuario para su creación o eliminación. Cabe recalcar que entre esta información no se encuentra la contraseña, debido a que esta solo podrá ser manejada por cada usuario de manera privada, con el fin de proteger su información.

CREAR O EDITAR UN USUARIO

ID: 2

Login: dev

Nombre: Miguel

Apellidos: Martinez

Correo electrónico: dev@kicallhost

Idioma: English

Roles:

- ROLE_ADMIN
- ROLE_DEVELOPER
- ROLE_PROJECT_LEADER
- ROLE_OPS_MANAGER

[VOLVER](#) [GUARDAR](#)

Creamos valor desde el momento cero.

Figura 44. Interfaz de creación o edición de usuarios

Por motivos de seguridad, cuando un usuario solicite la eliminación de su información al Encargado de Datos de la empresa, este realizará un proceso externo y comunicará al Administrador que proceda con la eliminación de los datos, es en este momento en el que se muestra la interfaz de la Figura 45.

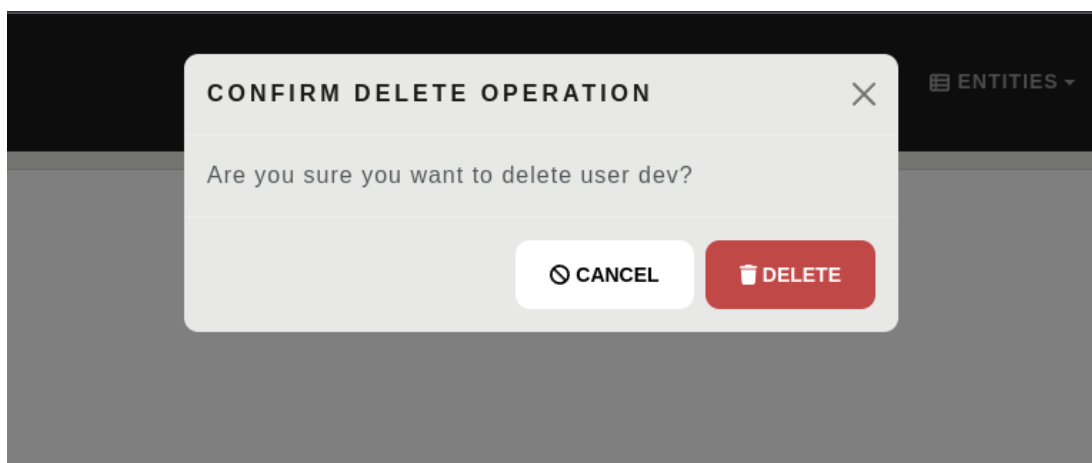


Figura 45. Interfaz de confirmación de eliminación de usuarios

Actualización de datos personales para usuarios

En la Figura 46 se muestra la interfaz con la que los usuarios sin importar el rol se encontrarán para poder modificar datos personales cuando así lo requieran.

AJUSTES DEL USUARIO [DEV]

Nombre
Maleo

Apellidos
Martinez

Correo electrónico
dev@localhost

Idioma
Español ✓

GUARDAR

Creamos valor desde el momento cero.

Figura 46. Interfaz de actualización de datos personales para usuarios

Actualización de contraseña para usuarios

En la Figura 47 se presenta la interfaz de actualización de contraseña a la que libremente podrán acceder los usuarios de la aplicación.

CONTRASEÑA DE [DEV]

Contraseña actual
**** ✓

Nueva contraseña
***** ✓

Seguridad de la contraseña:
■

Confirmación de la nueva contraseña
***** ✓

GUARDAR

Figura 47. Interfaz de actualización de contraseña para usuarios

En la Figura 48 se puede observar el método que se usa para cuando una contraseña es cambiada. Primero, se obtiene el nombre de usuario actual y se busca el usuario correspondiente en el repositorio, una vez que se verifica si la contraseña actual coincide con la almacenada encriptada en la base de datos, se procede a encriptar la nueva contraseña utilizando el codificador de contraseñas passwordEncoder. La

contraseña encriptada se asigna al usuario, lo que actualiza su registro en la base de datos. La función se encarga de limpiar las cachés asociadas al usuario y registra el cambio en el registro de depuración. Cabe recalcar que el resultado de la función no se expone explícitamente, y la anotación `@Transactional` asegura que todas las operaciones se realicen de manera atómica en términos de base de datos.

```
↳ Mateo Martinez
@Transactional
fun changePassword(currentClearTextPassword: String, newPassword: String) {
    getCurrentUserLogin()
        .flatMap(userRepository::findOneByLogin)
        .ifPresent { user ->
            val currentEncryptedPassword = user.password
            if (!passwordEncoder.matches(currentClearTextPassword, currentEncryptedPassword)) {
                throw InvalidPasswordException()
            }
            val encryptedPassword = passwordEncoder.encode(newPassword)
            user.password = encryptedPassword
            clearUserCaches(user)
            log.debug("Changed password for User: $user")
            user
        }
}
```

Figura 48. Método de la actualización de contraseña

Gestión de Clientes para Gerentes de Operaciones

En la Figura 49 se muestra la interfaz con la que los usuarios con rol Gerente de Operaciones cuentan para poder visualizar la información más relevante de los clientes de la empresa, contando con las opciones de creación, edición, visualización de detalle y eliminación de clientes.

| CLIENTES | | | | | | | | REFRESCAR LISTA | CREAR NUEVO CLIENTE |
|----------|----------------------|---------|--------|-----------------------|------------------------|------------|-----------------|---------------------|---------------------|
| ID | NOMBRE | PAIS | CIUDAD | DIRECCION | EMAIL | TELÉFONO | FECHA DE INICIO | | |
| 1001 | Petrolera | Ecuador | Quito | Av. Shyris | lapetrolera@gmail.com | 0982387112 | 25/01/2020 | VER EDITAR ELIMINAR | |
| 1003 | Farmacias Nueva Vida | Ecuador | Cuenca | Coral y Tres Esquinas | lasfarmacias@gmail.com | 0528391391 | 25/06/2022 | VER EDITAR ELIMINAR | |

Mostrando 1 - 2 de 2 elementos.

« 1 »

Creamos valor desde el momento cero.

Figura 49. Interfaz de visualización general de Clientes para Gerente de Operaciones

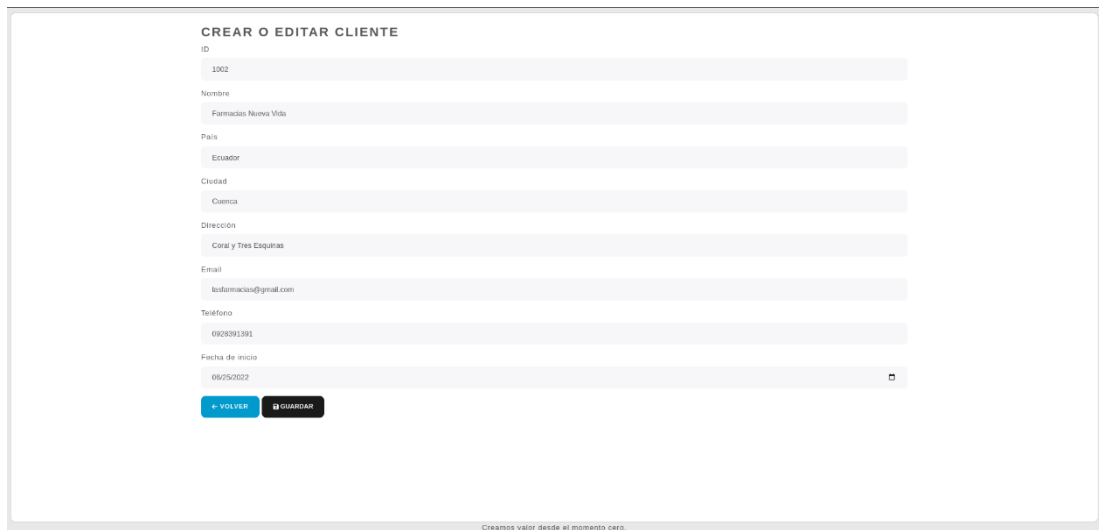
En la Figura 50 se encuentra el detalle de un Cliente para que se lo pueda visualizar de una mejor manera, pensando en la experiencia de uso del usuario final que en este caso es el Gerente de Operaciones.

| CLIENTE |
|---|
| ID 1002 |
| Nombre Farmacias Nueva Vida |
| País Ecuador |
| Ciudad Cuenca |
| Dirección Coral y Tres Esquinas |
| Email lasfarmacias@gmail.com |
| Teléfono 0528391391 |
| Fecha de inicio 25/06/2022 |
| + VOLVER |

Creamos valor desde el momento cero.

Figura 50. Interfaz del detalle de Clientes para Gerente de Operaciones

En la Figura 51 se muestra la interfaz que tienen disponibles los usuarios con el rol respectivo para poder crear o editar la información de un Cliente de la empresa, es importante recalcar que todos los datos solicitados son exclusivamente de uso interno para la empresa y mejora del uso de esta aplicación web.



Formulario de creación o edición de cliente con los siguientes campos:

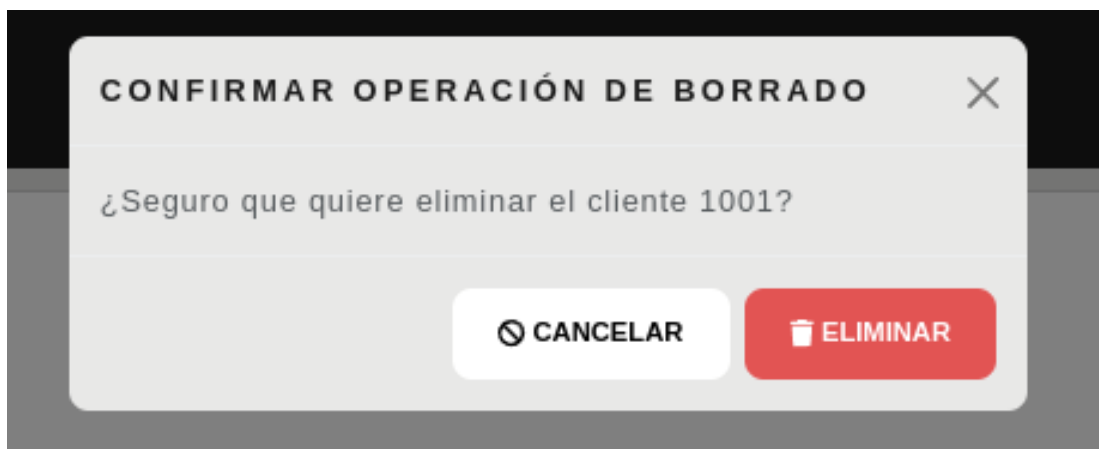
- ID: 1002
- Nombre: Farmacia Nueva Vida
- País: Ecuador
- Ciudad: Quito
- Dirección: Cora y Tres Esquinas
- Email: testfarmacia@gmail.com
- Teléfono: 092031391
- Fecha de inicio: 09/25/2022

Botones: **VOLVER** (azul) y **GUARDAR** (negro).

Creamos valor desde el momento cero.

Figura 51. Interfaz de creación o edición de Clientes para Gerente de Operaciones

En la Figura 52 se muestra un modal con el cual se podrá confirmar la eliminación de la base de datos del Cliente.



Modal de confirmación de borrado con el siguiente contenido:

CONFIRMAR OPERACIÓN DE BORRADO [X]

¿Seguro que quiere eliminar el cliente 1001?

Botones: **CANCELAR** (gris) y **ELIMINAR** (rojo).

Figura 52. Interfaz de confirmación de borrado de un Cliente

Gestión de Proyectos para Líderes de Proyecto

En la Figura 53 se muestra la interfaz principal de Proyectos que tendrán disponible los usuarios con rol Líder de Proyecto, en cada registro se muestran las opciones de creación, visualización de detalle, edición y eliminación del Proyecto.

| ID | NOMBRE | ACTIVO | FECHA DE INICIO | FECHA FIN | CLIENTE | |
|------|---------------|--------|-----------------|------------|----------------------|---|
| 1051 | Portal | true | 01/10/2023 | 31/01/2024 | Petrolera | VER EDITAR ELIMINAR |
| 1052 | Servicios Web | true | 01/12/2023 | 31/01/2024 | Petrolera | VER EDITAR ELIMINAR |
| 1052 | RRHH Admin | true | 01/12/2023 | 28/02/2024 | Farmacias Nueva Vida | VER EDITAR ELIMINAR |

Mostrando 1 - 3 de 3 elementos.

Creemos valor desde el momento cero.

Figura 53. Interfaz de visualización general de Proyectos para Líderes de Proyecto

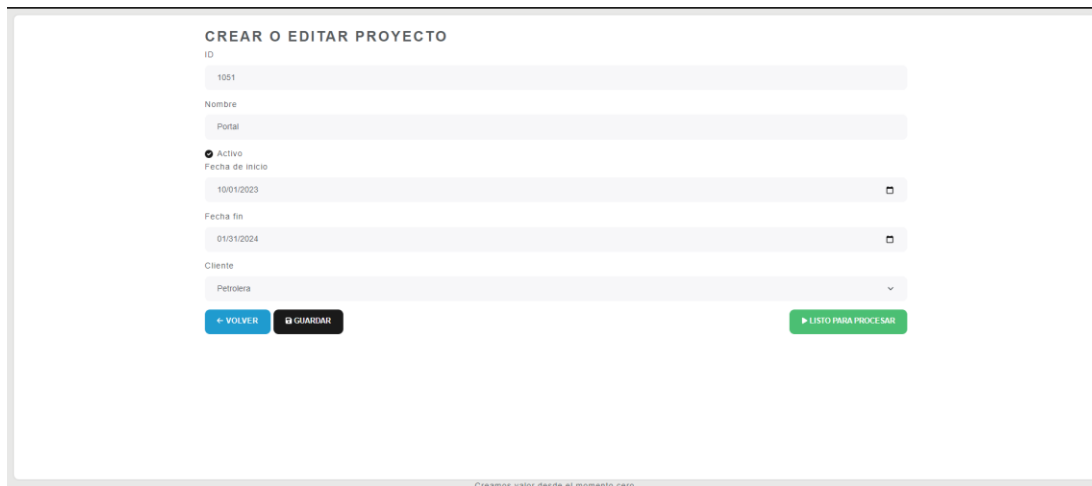
En la Figura 54 se presenta la interfaz que verán los usuarios con rol de Líder de Proyecto al entrar en el detalle de cada Proyecto. Se cuenta con dos botones de acción los cuales permiten regresar o ir a la edición del Proyecto.

| PROYECTO | |
|---|----------------------|
| ID | 1052 |
| Nombre | RRHH Admin |
| Activo | true |
| Fecha de inicio | 01/12/2023 |
| Fecha fin | 28/02/2024 |
| Cliente | Farmacias Nueva Vida |
| Lider de proyecto | projectlead |
| ← VOLVER EDITAR | |

Creemos valor desde el momento cero.

Figura 54. Interfaz del detalle del Proyecto para Líderes de Proyecto

En la Figura 55 se muestra la interfaz mediante la cual el Líder de Proyecto podrá crear o editar la información de un Proyecto, cabe recalcar que cuando se crea uno nuevo, el usuario que lo crea es asignado automáticamente como Líder de Proyecto.



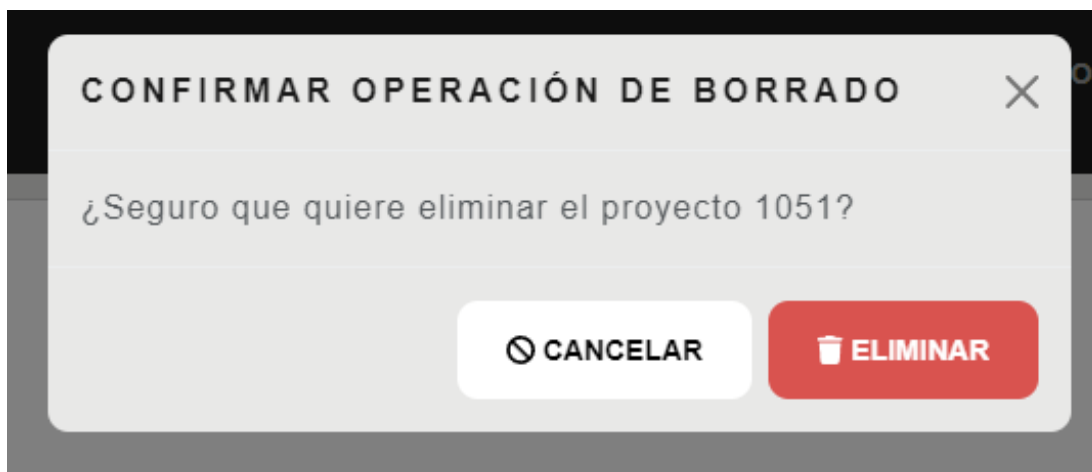
The screenshot shows a web form titled "CREAR O EDITAR PROYECTO". The form contains the following fields and controls:

- ID: Input field with the value "1051".
- Nombre: Input field with the value "Portal".
- Activo: Radio button selected, labeled "Activo".
- Fecha de inicio: Date picker showing "10/01/2023".
- Fecha fin: Date picker showing "01/31/2024".
- Cliente: Dropdown menu showing "Petrolera".
- Buttons: "VOLVER" (blue), "GUARDAR" (black), and "LISTO PARA PROCESAR" (green).

At the bottom of the form, there is a small footer text: "Creamos valor desde el momento cero."

Figura 55. Interfaz de creación o edición de Proyectos para Líderes de Proyecto

En la Figura 56 se muestra un modal con el cual se podrá confirmar la eliminación de un proyecto.



The screenshot shows a modal dialog box titled "CONFIRMAR OPERACIÓN DE BORRADO". The modal contains the following elements:

- Title: "CONFIRMAR OPERACIÓN DE BORRADO" with a close button (X) in the top right corner.
- Text: "¿Seguro que quiere eliminar el proyecto 1051?"
- Buttons: "CANCELAR" (white) and "ELIMINAR" (red).

Figura 56. Interfaz de confirmación de eliminación de Proyecto

Gestión de Actividades para Desarrolladores

En la Figura 57 se muestra la interfaz que tienen los usuarios con rol Desarrollador, la cual muestra únicamente las Actividades que a cada Desarrollador pertenece, con la

información más relevante únicamente. Se encuentran en la parte superior los campos para filtrar la información. Para cada actividad existen 3 botones de acción los cuales permiten entrar al detalle de la Actividad, editar la información de la Actividad o eliminar por completo la Actividad. Así mismo, en la columna de Proyecto, se muestra el nombre del Proyecto al que pertenece la Actividad con un hipervínculo que nos lleva al detalle del Proyecto.

ACTIVIDADES

REFRESCAR LISTA + CREAR NUEVA ACTIVIDAD

Todo Selecciona un año Selecciona un proyecto Q BUSCAR

| FECHA # | DÍA | DESCRIPCIÓN | HORA DE INICIO | HORA FIN | HORAS | PROYECTO |
|------------|-----------|--|----------------|----------|-------|---------------|
| 14/12/2023 | Jueves | Reunion con el cliente | 14:00 | 16:00 | 2 | Portal |
| 14/12/2023 | Jueves | Analisis a la implementacion y mejoras | 11:00 | 13:00 | 2 | Servicios_Web |
| 14/12/2023 | Jueves | Planificacion de trabajo | 09:00 | 11:00 | 314 | Servicios_Web |
| 13/12/2023 | Miercoles | Mejoras en el login de la aplicacion | 14:00 | 18:00 | 4 | RRHH_Admin |
| 13/12/2023 | Miercoles | Desarrollo del modulo de administracion | 10:00 | 12:00 | 2 | Portal |
| 13/12/2023 | Miercoles | Revision de pendientes con el equipo de desarrollo | 09:00 | 10:00 | 1 | Portal |

Creamos valor desde el momento cero.

Figura 57. Interfaz de visualización general de Actividades para Desarrolladores

En la Figura 58 se presenta la interfaz que verán los usuarios con rol Desarrollador al entrar en el detalle de cada Actividad, con la finalidad de que, si la descripción de la Actividad es grande, aquí lo puedan visualizar de mejor manera. Se cuenta con dos botones de acción los cuales permiten regresar o ir a la edición de la Actividad.

ACTIVIDAD

ID
1151

Fecha
08/12/2023

Descripción
Implementación de web services

Hora de inicio
17:00

Hora fin
18:00

Horas
1

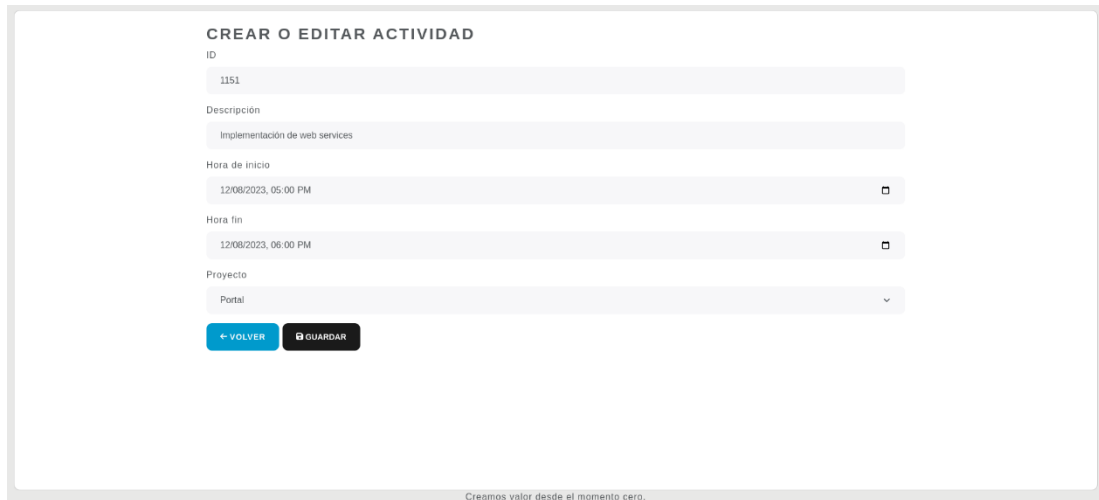
Proyecto
Portal

+ VOLVER EDITAR

Creamos valor desde el momento cero.

Figura 58. Interfaz del detalle de Actividades para Desarrolladores

En la Figura 59 se muestra la interfaz que tendrán disponibles los usuarios para poder crear o editar sus Actividades, aquí se presentan únicamente los datos relevantes para el caso.



The screenshot shows a form titled "CREAR O EDITAR ACTIVIDAD". It contains the following fields: "ID" with the value "1151"; "Descripción" with the value "Implementación de web services"; "Hora de inicio" with the value "12/08/2023, 05:00 PM"; "Hora fin" with the value "12/08/2023, 06:00 PM"; and "Proyecto" with the value "Portal". At the bottom, there are two buttons: "← VOLVER" and "GUARDAR". A small footer text at the bottom of the form reads "Creamos valor desde el momento cero."

Figura 59. Interfaz de creación o edición de Actividades para Desarrolladores

En la Figura 60 se presenta un modal con el cual se podrá confirmar la eliminación de una Actividad.

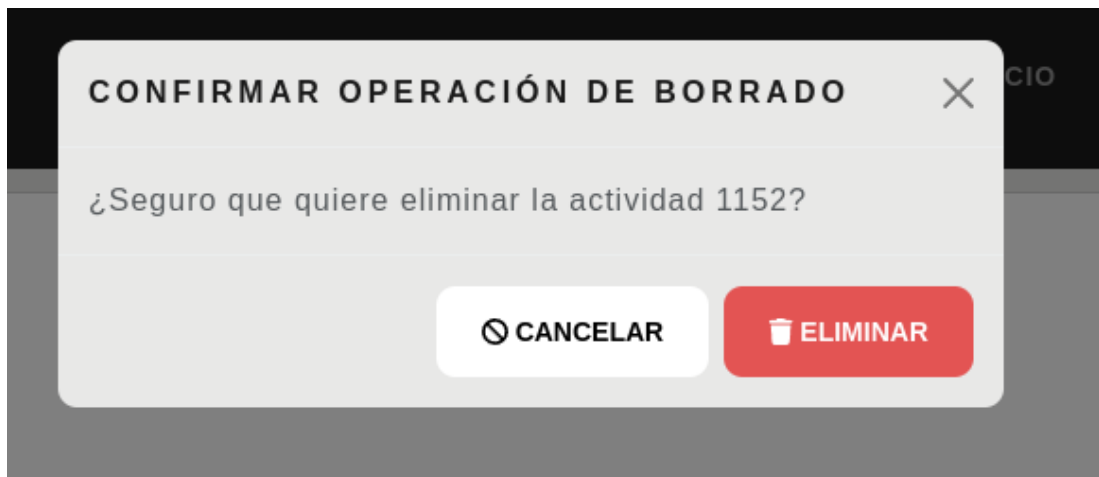


Figura 60. Interfaz de confirmación de borrado de una Actividad

En la Figura 61 se muestra el detalle del proyecto al que cada Actividad pertenece, permitiendo acceder a esta interfaz desde la pantalla de visualización general. Cabe recalcar que, en esta pantalla no existe ningún botón de acción que permita editar el

proyecto, debido a que el usuario con rol de Desarrollador no está autorizado para manejar la información de los proyectos, solo visualizarla.

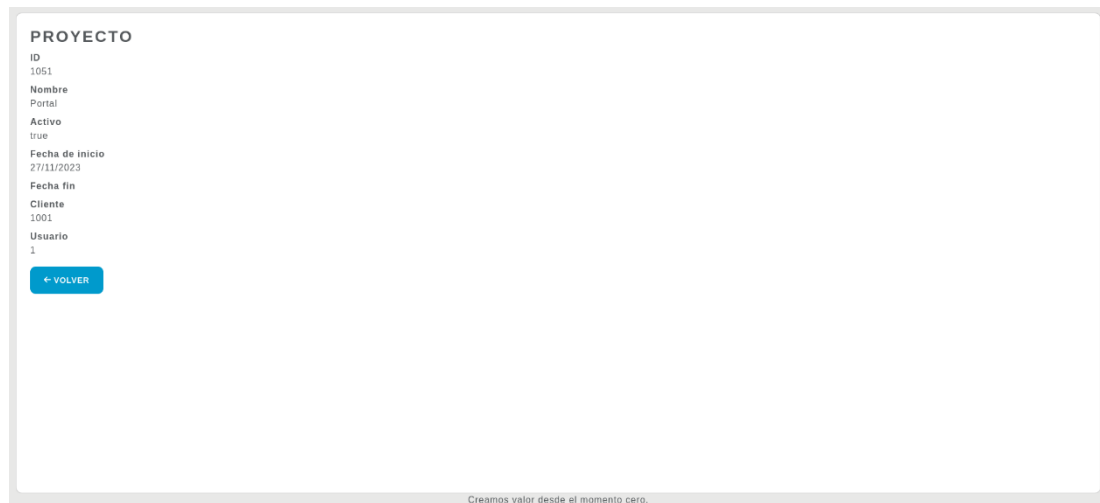


Figura 61. Interfaz del detalle del proyecto al que pertenece la Actividad

Seguimiento de Actividades para Líderes de Proyecto

En la Figura 62 se muestra la interfaz de Actividades que está disponible para los usuarios con rol Líder de Proyecto. En esta, se cargarán únicamente las actividades de los proyectos de los cuales este usuario es Líder de Proyecto. Se cuenta con opciones de filtrado para una mejor experiencia de uso, así mismo, al lado derecho de cada Actividad se encuentra un botón para ver el detalle de la actividad y a su vez poder registrar una o más Observaciones a la Actividad.

| FECHA | DÍA | DESCRIPCIÓN | HORA DE INICIO | HORA FIN | HORAS | PROYECTO | DESARROLLADOR |
|------------|--------|--|----------------|----------|-------|--------------|---------------|
| 14/12/2023 | Jueves | Análisis a la implementación y mejoras | 11:00 | 13:00 | 2 | Servicio Web | dev |
| 14/12/2023 | Jueves | Planificación de trabajo | 09:00 | 11:00 | 314 | Servicio Web | dev |

Figura 62. Interfaz de visualización general de Actividades para Líderes de Proyecto

En la Figura 63 y Figura 64 se presentan los métodos utilizados para obtener las Actividades de los Proyectos de los cuales el usuario conectado es Líder de Proyecto. En el controlador se implementó un método GET que llama a la capa de lógica de negocio, esta obtiene el usuario conectado y lo agrega como un filtro para la búsqueda de Proyectos y posteriormente filtrar las Actividades de los Proyectos de interés.

```
@GetMapping("/activities-tracking")
fun getAllProjectLeaderActivities(
    criteria: ActivityCriteria
): ResponseEntity<MutableList<Activity>> {
    log.debug("REST request to get Activities for project leader")

    val entityList = activityQueryService.findByUserLoginProjects()
    return ResponseEntity.ok().body(entityList)
}
```

Figura 63. Método GET para obtener las actividades de interés de un Líder de Proyecto

```
Mateo Martinez +1
@Transactional(readOnly = true)
fun findByUserLoginProjects(): MutableList<Activity> {
    log.debug("Find activities of user logged in")
    val user = userService.getUserWithAuthorities().get()

    val userIdFilter = LongFilter()
    userIdFilter.setEquals(user.id)
    val projectCriteria = ProjectCriteria()
    projectCriteria.userId = userIdFilter

    val projects = projectQueryService.findByCriteria(projectCriteria)

    return activityRepository.findAllByProjectIn(projects)
}
```

Figura 64. Capa de lógica de negocio para obtener las actividades de interés de un Líder de Proyecto

En la Figura 65 se presenta una interfaz disponible para los Líderes de Proyecto, en el lado izquierdo se encuentra el detalle de la Actividad y al lado derecho se encuentra a disposición la lista de las Observaciones registradas para la Actividad y la opción de registrarlas.



Figura 65. Interfaz del detalle de la Actividad para Líderes de Proyecto

Gestión de las Observaciones

El proceso que seguirá la Observación de una Actividad desde que un Líder de Proyecto la genera, hasta que el Desarrollador dueño de la Actividad la resuelve se muestra a continuación:

En la Figura 66 se muestra el modal que se abre al entrar en la opción de “Crear Nueva Observación” de la Figura 65 en la que los Líderes de Proyecto pueden registrar una nueva Observación a una Actividad o editar una existente. Cuando se cree una, el Desarrollador dueño de la Actividad en cuestión será notificado vía correo electrónico, para que pueda resolver esta Observación.



Figura 66. Interfaz de creación o edición de Observaciones para Líderes de Proyecto

En la Figura 67 se puede visualizar el correo electrónico que la aplicación envía al Desarrollador dueño de la Actividad que tiene una nueva Observación. En este correo se envía la descripción de la Observación, el ID de la Actividad implicada y la url que llevará al Desarrollador a corregir la información de la Actividad reportada.

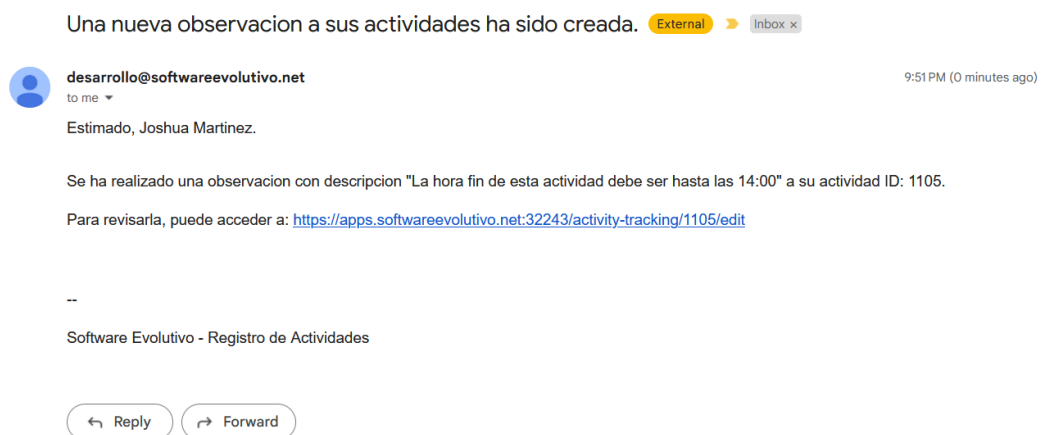


Figura 67. Notificación mediante correo electrónico de nueva Observación

En la Figura 68 se muestra el método que recibe la información de usuario, descripción de la Observación y el ID de la Actividad, para agregarlo como variables en el contexto del correo junto a la preferencia de lenguaje del usuario destinatario. Esta información

es enviada a una función propia de Spring Boot que permite el envío de correos en base a un template. El template de este correo se encuentra en el *Anexo N*.

```
@Async
fun sendNewObservationEmail(
    user: User,
    completeName: String,
    observationDescription: String,
    activityId: Long
) {
    log.debug("Sending new observation email to '${user.email}'")

    val locale = Locale.forLanguageTag(user.langKey)

    val context = Context(locale).apply { this: Context
        setVariable( name: "completeName", completeName)
        setVariable( name: "observationDescription", observationDescription)
        setVariable( name: "activityId", activityId.toString())
        setVariable(BASE_URL, jHipsterProperties.mail.baseUrl)
    }

    sendEmailFromTemplateCore(
        user,
        templateName: "mail/newObservationEmail",
        context,
        titleKey: "email.newobservation.title"
    )
}
```

Figura 68. Método para el envío de correo electrónico al crear una Observación

En la Figura 69 se muestra la interfaz a donde nos redirige la url del correo electrónico de la Figura 67, en esta interfaz el Desarrollador puede corregir la información de la Actividad y marcar como resuelta una Observación. Pudiendo visualizar todas las Observaciones de esa Actividad sean nuevas o resueltas.



Figura 69. Interfaz de visualización de Observaciones y edición de Actividad para Desarrollador

En la Figura 70 se muestra el modal de confirmación que tiene el Desarrollador para hacer saber que ha resuelto la Observación.

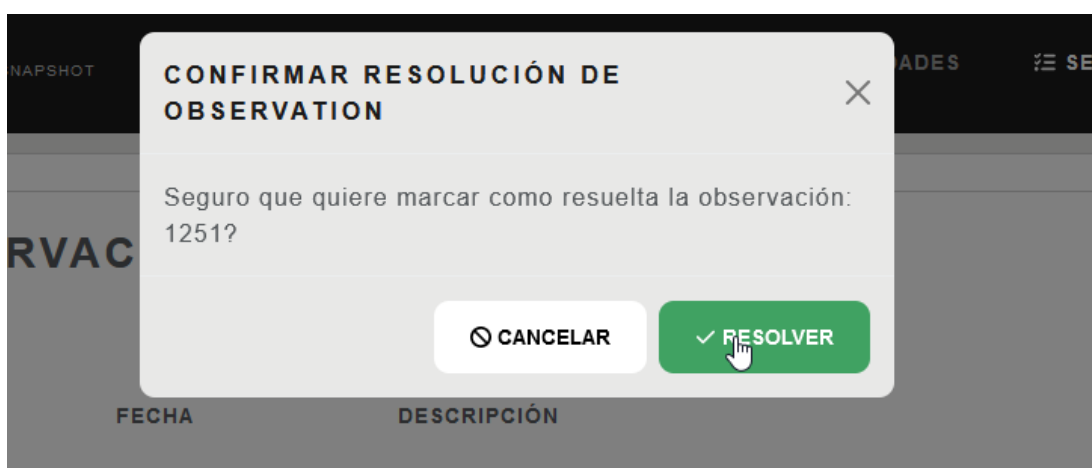


Figura 70. Interfaz de confirmación de resolución de Observación para Desarrollador

En la Figura 71 se presenta para los usuarios con rol Desarrollador la interfaz de visualización general de las Actividades que él tenga con Observaciones, para que en cualquier momento pueda acceder a esta información y validar que las Observaciones que se le fueron notificadas ya se han resuelto o aún las tiene que resolver. El botón de Editar le permitirá acceder a la interfaz de la Figura 69.

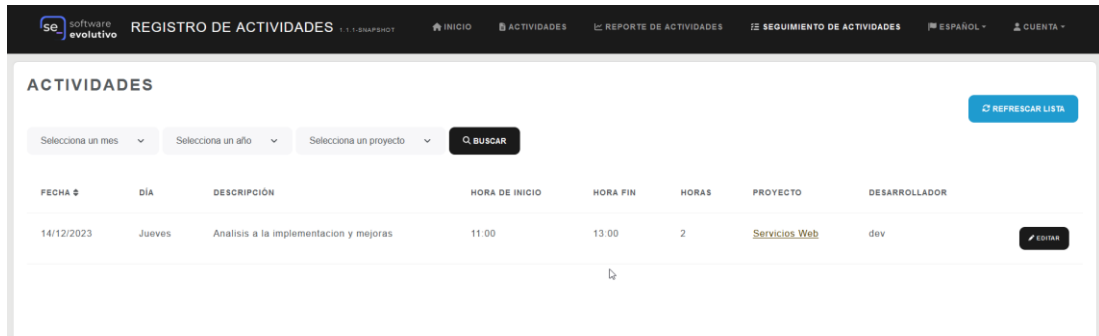


Figura 71. Interfaz de seguimiento de Actividades para Desarrolladores

En la Figura 72 y Figura 73 se presentan los métodos utilizados para obtener las Actividades que tienen Observaciones del usuario conectado. En el controlador se implementó un método GET que llama a la capa de lógica de negocio, esta obtiene el usuario conectado y lo agrega como un filtro para la búsqueda de Actividades y solicita al Entity Manager que las Actividades que regresan sean también las que tengan Observaciones.

```

@GetMapping("/{activities-tracking/activities-with-observation}")
fun getAllDevActivitiesWithObservation(
    criteria: ActivityCriteria
): ResponseEntity<MutableList<Activity>> {
    log.debug("REST request to get dev Activities with observation")

    val entityList = activityQueryService.findByUserLoginAndObservationExists()
    return ResponseEntity.ok().body(entityList)
}

```

Figura 72. Método GET para obtener las actividades con Observaciones de un Desarrollador

```

@Transactional(readonly = true)
fun findByUserLoginAndObservationExists(): MutableList<Activity> {
    log.debug("Find activities that has observations of user logged in")
    val user = userService.getUserWithAuthorities().get()

    val observations =
        observationQueryService.findByCriteria(ObservationCriteria()) as MutableCollection<MutableSet<Observation>>

    return activityRepository.findDistinctByObservationsInAndUserIs(observations, user)
}

```

Figura 73. Capa de lógica de negocio para obtener las actividades con Observación de un Desarrollador

En la Figura 74 se muestra la interfaz en la que los usuarios con rol Líder de Proyecto tendrán una visualización general de las Observaciones de los Proyectos de los que son Líder de Proyecto y en la columna de Actividad tendrán el url que los llevará a la interfaz de la Figura 65 para validar que si la información de la Actividad se corrigió o no.

| ID | FECHA | DESCRIPCIÓN | ESTADO | ACTIVIDAD |
|------|----------------|--|----------|-----------|
| 1105 | 27/12/23 00:00 | La hora fin de esta actividad debe ser hasta las 14:00 | RESUELTA | 1105 |

Mostrando 1 - 1 de 1 elementos

Figura 74. Interfaz de visualización general de Observaciones para Líder de Proyecto

Notificación de Proyecto listo para procesar

En la Figura 55 en la interfaz de creación o edición de Proyectos contamos con un botón que permite a los Líderes de Proyecto comunicar al Gerente de Operaciones que el Proyecto en cuestión está listo para procesar, a continuación, se va a presentar que es lo que permite realizar este botón:

En la Figura 75 se muestra el modal en el que se solicita la confirmación del Líder de Proyecto para notificar al Gerente de Operaciones sobre este Proyecto.

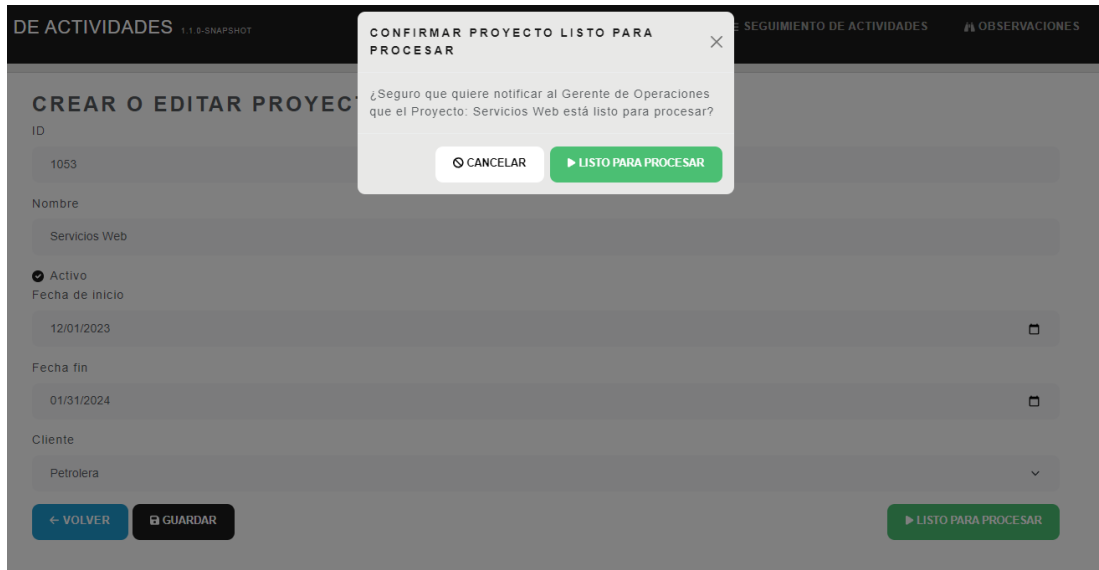


Figura 75. Interfaz de confirmación de Proyecto listo para procesar

En la Figura 76 se puede visualizar el correo electrónico que la aplicación envía para notificar que un proyecto está listo para ser procesado, lo que significa que el Líder de Proyecto ha revisado y realizado observaciones a los Desarrolladores que han sido corregidas. Con el objetivo de dejar constancia, en el correo se incluye el nombre del Líder de Proyecto.

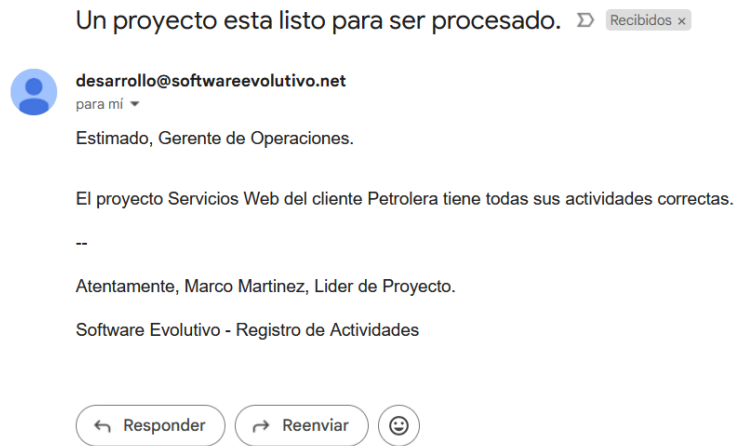


Figura 76. Notificación mediante correo electrónico de Proyecto listo para procesar

En la Figura 77 se muestra el método que recibe la información del Gerente de Operaciones, nombre del Proyecto, nombre del Cliente, y nombre del Líder de Proyecto para agregarlo como variables en el contexto del correo junto a la preferencia

de lenguaje del usuario destinatario. Esta información es enviada a una función propia de Spring Boot que permite el envío de correos en base a un template. El template de este correo se encuentra en el *Anexo O*.

```
@Async
fun sendProjectReadyEmail(
    opsManagers: List<User>,
    projectName: String,
    customerName: String,
    projectLeader: String
) {
    Log.debug("Sending project ready email to operations managers")

    for (user in opsManagers) {
        val locale = Locale.forLanguageTag(user.langKey)

        val context = Context(locale).apply { this: Context
            setVariable( name: "projectName", projectName)
            setVariable( name: "customerName", customerName)
            setVariable( name: "projectLeader", projectLeader)
            setVariable(BASE_URL, jHipsterProperties.mail.baseUrl)
        }

        sendEmailFromTemplateCore(
            user,
            templateName: "mail/projectReadyEmail",
            context,
            titleKey: "email.projectready.title"
        )
    }
}
```

Figura 77. Método para el envío de correo electrónico de Proyecto listo para procesar

Generación de reportes para Gerente de Operaciones

En la Figura 78 se muestra la interfaz con la que cuentan los usuarios con rol Gerente de Operaciones para generar reportes filtrados por mes, año y Cliente. Al enviar la solicitud en la misma interfaz se mostrará una previsualización en formato HTML del reporte, y se habilitará la opción de descargarlo como PDF.

En este reporte se puede visualizar el mes, año y nombre del Cliente del cual se está generando el reporte. Seguido del detalle, en este se muestran los Proyectos y los Desarrolladores que han trabajado en estos Proyectos con la suma de las horas de sus Actividades, al final se muestra el total de horas que la empresa trabajó para este Cliente en este periodo. El reporte en formato PDF se encuentra en el *Anexo P*.

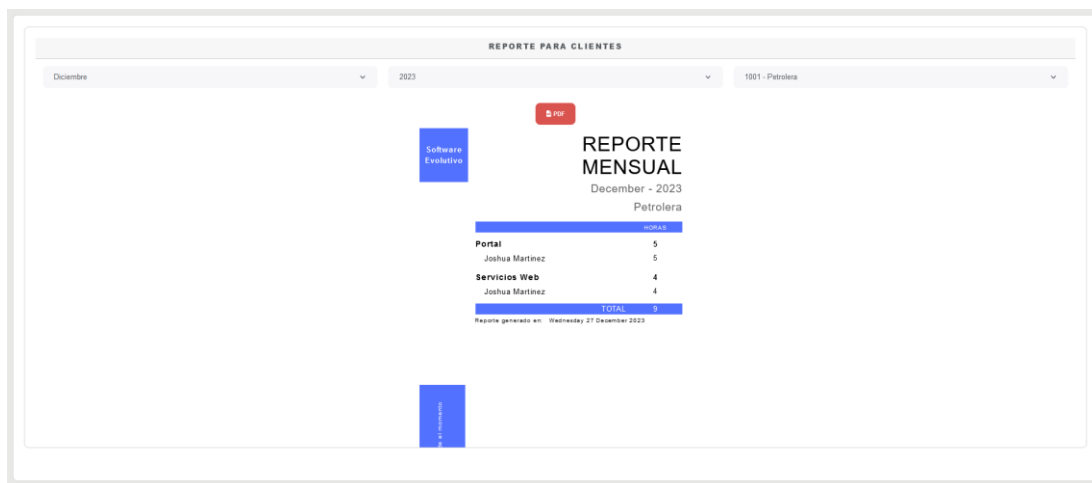


Figura 78. Interfaz para la generación de reportes para Gerente de Operaciones

En la Figura 79 se muestra el mensaje que obtiene el Gerente de Operaciones cuando no existen datos el mes, año y Cliente escogido.



Figura 79. Interfaz de reportes al no encontrar datos para el Gerente de Operaciones

En la Figura 80 se tiene parte de la implementación necesaria para la generación de reportes para Gerente de Operaciones, en este caso se establece la conexión con el Data Source local, el mismo al que nuestra aplicación está conectada. Posteriormente envía los parámetros necesarios para ejecutar la consulta necesaria para generar el reporte, es decir, el ID del Cliente, el mes y el año. Cuando el reporte se genera, lo compila y lo guarda en un archivo de sistema temporal de la extensión que se necesite.

```

@Autowired
protected var localDataSource: DataSource? = null

Mateo Martinez
fun loadReport(conn: Connection, customerId: Long, month: Int, year: Int, ignorePagination: Boolean = false): JasperPrint {
    log.debug("Load report $CUSTOMER_REPORT_URL for client: $customerId")

    val report: JasperReport
    val fileUrl = CustomerMonthlyReportCoreService::class.java.classLoader.getResource(CUSTOMER_REPORT_URL)
    val file = FileInputStream(fileUrl!!.path)

    val params = HashMap<String, Any>()
    params["customer_id"] = customerId
    params["month"] = month
    params["year"] = year
    params[JRParameter.IS_IGNORE_PAGINATION] = ignorePagination

    report = JasperCompileManager.compileReport(file)
    return JasperFillManager.fillReport(report, params, conn)
}

```

Figura 80. Método para la generación de reporte con JasperReport para Gerente de Operaciones

En la Figura 81 tenemos la consulta ejecutada en PostgreSQL que, dado el ID del Cliente, el mes y el año de las de los cuales quiere obtener el reporte ejecuta la sentencia y mediante LEFT JOIN va encontrando y relacionando los datos. Necesariamente realiza sumas internas para poder obtener las diferentes sumas del reporte, tales como la suma por Desarrollador, por Proyecto y la suma total para el Cliente en el periodo escogido.

```

SELECT
    _month,
    _year,
    customer_name,
    project_name,
    dev_name,
    dev_hours,
    SUM(dev_hours) OVER (PARTITION BY _month, _year, customer_name, project_name) AS project_hours,
    SUM(dev_hours) OVER (PARTITION BY _month, _year, customer_name) AS total_hours
FROM (
    SELECT
        TO_CHAR(DATE_TRUNC('MONTH', a.start_hour), 'Month') AS _month,
        EXTRACT(YEAR FROM a.start_hour) AS _year,
        CONCAT(us.first_name, ' ', us.last_name) AS dev_name,
        c.name AS customer_name,
        p.name AS project_name,
        CAST(SUM(EXTRACT(HOUR FROM a.end_hour - a.start_hour)) AS INTEGER) AS dev_hours
    FROM
        jhi_user us
        JOIN activity a ON us.id = a.user_id
        LEFT JOIN project p ON a.project_id = p.id
        LEFT JOIN customer c ON p.customer_id = c.id
    WHERE
        EXTRACT(MONTH FROM a.start_hour) = $P{month}
        AND EXTRACT(YEAR FROM a.start_hour) = $P{year}
        AND c.id = $P{customer_id}
    GROUP BY
        _month, _year, customer_name, project_name, dev_name
) AS subquery

```

Figura 81. Consulta en PostgreSQL para el reporte para Gerente de Operaciones

Generación de reportes para Desarrolladores

En la Figura 82 se muestra la interfaz con la que cuentan los usuarios con rol Desarrollador para generar reportes de sus actividades filtrados por mes y año. Al enviar la solicitud en la misma interfaz se mostrará una previsualización en formato HTML del reporte, y se habilitará la opción de descargarlo como PDF.

En este reporte se puede visualizar el mes, año y nombre completo del Desarrollador del cual se está generando el reporte. Seguido del detalle, en este se muestran los Clientes y los Proyectos con la suma de sus horas de Actividades, al final se muestra el total de horas trabajadas por el Desarrollador en este periodo. El reporte en formato PDF se encuentra en el *Anexo Q*.



| REPORTE DE ACTIVIDADES | |
|--|-------|
| Diciembre | 2023 |
| Software Evolutivo | |
| REPORTE MENSUAL | |
| December - 2023 | |
| Joshua Martinez | |
| | HORAS |
| Farmacias Nueva Vida | 4 |
| RRHH Admin | 4 |
| Petroliera | 9 |
| Portal | 5 |
| Servicios Web | 4 |
| TOTAL | 13 |
| Reporte generado en: Wednesday, 27 December 2023 | |

Figura 82. Interfaz para la generación de reportes para Desarrolladores

En la Figura 83 se muestra el mensaje que obtiene el Desarrollador cuando no existen datos de sus actividades para ese mes y año.



| REPORTE DE ACTIVIDADES | |
|--------------------------------------|------|
| Noviembre | 2023 |
| No existen datos para este mes y año | |

Figura 83. Interfaz de reportes al no encontrar datos para el Desarrollador

En la Figura 84 se puede ver parte de la implementación necesaria para la generación de reportes para Desarrollador, en este caso se establece la conexión con el Data Source local, el mismo al que nuestra aplicación está conectada. Posteriormente envía los parámetros necesarios para ejecutar la consulta necesaria para generar el reporte, es decir, el username del Desarrollador, el mes y el año. Cuando el reporte se genera, lo compila y lo guarda en un archivo de sistema temporal de la extensión que se necesite.

```
@Autowired
protected var localDataSource: DataSource? = null

// Mateo Martinez
fun loadReport(conn: Connection, username: String, month: Int, year: Int, ignorePagination: Boolean = false): JasperPrint {
    log.debug("Load report $REPORT_URL for user: $username")

    val report: JasperReport
    val fileUrl = DevActivityReportCoreService::class.java.classLoader.getResource(REPORT_URL)
    val file = FileInputStream(fileUrl!!.path)

    val params = HashMap<String, Any>()
    params["user_login"] = username
    params["month"] = month
    params["year"] = year
    params[JRParameter.IS_IGNORE_PAGINATION] = ignorePagination

    report = JasperCompileManager.compileReport(file)
    return JasperFillManager.fillReport(report, params, conn)
}
```

Figura 84. Método para la generación de reporte con JasperReport para Desarrollador

En la Figura 85 tenemos la consulta ejecutada en PostgreSQL que, dado el nombre de usuario del Desarrollador, el mes y el año de las Actividades de las cuales quiere obtener el reporte ejecuta la sentencia y mediante LEFT JOIN va encontrando y relacionando los datos. Necesariamente realiza sumas internas para poder obtener las diferentes sumas del reporte, tales como la suma por Proyecto, por Cliente y la suma total de las Actividades del periodo escogido.

```

SELECT
  _month,
  _year,
  dev_name,
  customer_name,
  project_name,
  project_hours,
  SUM(project_hours) OVER (PARTITION BY _month, _year, dev_name, customer_name) AS customer_hours,
  SUM(project_hours) OVER (PARTITION BY _month, _year, dev_name) AS total_hours
FROM (
  SELECT
    TO_CHAR(DATE_TRUNC('MONTH', a.start_hour), 'Month') AS _month,
    EXTRACT(YEAR FROM a.start_hour) AS _year,
    CONCAT(us.first_name, ' ', us.last_name) AS dev_name,
    c.name AS customer_name,
    p.name AS project_name,
    CAST(SUM(EXTRACT(HOUR FROM a.end_hour - a.start_hour)) AS INTEGER) AS project_hours
  FROM
    jhi_user us
    JOIN activity a ON us.id = a.user_id
    LEFT JOIN project p ON a.project_id = p.id
    LEFT JOIN customer c ON p.customer_id = c.id
  WHERE
    us.login = ${user_login}
    AND EXTRACT(MONTH FROM a.start_hour) = ${month}
    AND EXTRACT(YEAR FROM a.start_hour) = ${year}
  GROUP BY
    _month, _year, dev_name, customer_name, project_name
) AS subquery

```

Figura 85. Consulta en PostgreSQL para el reporte para Desarrollador

3.2.6. Fase IV: Pruebas

3.2.6.1. Pruebas de aceptación

Se realizaron pruebas de aceptación a la aplicación web para poder validar que las funcionalidades cumplan con los requerimientos funcionales establecidos, a continuación, se presenta cada una de ellas:

Tabla 28. Tabla de pruebas de aceptación de la aplicación web

| Pruebas de aceptación | | | | |
|-----------------------|----------------------------|--|----------|-------------|
| ID Req. | Requerimiento | Criterio de aceptación | Estado | Responsable |
| RF-01 | Autenticación de usuarios. | Para ingresar a la aplicación se inserta nombre de usuario y contraseña. | Aceptado | Usuario |

| Pruebas de aceptación | | | | |
|------------------------------|-------------------------------|---|---------------|----------------------------------|
| ID Req. | Requerimiento | Criterio de aceptación | Estado | Responsable |
| | | Al ingresar, se generará un JWT. | Aceptado | Programador |
| | | Al ingresar correctamente el sistema informará quien es el usuario conectado. | Aceptado | Usuario |
| | | Al ingresar, se mostrará las opciones personalizadas de acuerdo con el rol del usuario. | Aceptado | Usuario |
| RF-02 | Crear usuarios. | El administrador, podrá crear usuarios para la aplicación. | Aceptado | Usuario |
| | | La información ingresada por el administrador es información no privada. | Aceptado | Usuario |
| | | En la base de datos se almacenarán las contraseñas cifradas en hash. | Aceptado | Programador - Encargado de Datos |
| RF-03 | Actualizar datos de usuarios. | El usuario puede cambiar su información personal cuando así lo requiera. | Aceptado | Usuario |
| | | El usuario puede cambiar su contraseña de acceso cuando así lo requiera. | Aceptado | Usuario |
| | | La información del usuario y su cuenta | Aceptado | Encargado de Datos |

| Pruebas de aceptación | | | | |
|------------------------------|----------------------|---|---------------|--------------------|
| ID Req. | Requerimiento | Criterio de aceptación | Estado | Responsable |
| | | pueden ser eliminadas por completo de la aplicación. | | |
| RF-04 | Gestionar clientes. | El Gerente de operaciones puede visualizar todos los Clientes de la empresa. | Aceptado | Usuario |
| | | El Gerente de operaciones puede visualizar el detalle de cada Cliente. | Aceptado | Usuario |
| | | El Gerente de operaciones puede crear un nuevo Cliente con todos sus datos. | Aceptado | Usuario |
| | | El Gerente de operaciones puede editar la información de un Cliente. | Aceptado | Usuario |
| | | El Gerente de operaciones puede eliminar por completo la información de un Cliente. | Aceptado | Usuario |
| RF-05 | Gestionar proyectos. | El Líder de Proyecto puede visualizar únicamente los Proyectos de los cuales él es Líder de Proyecto. | Aceptado | Usuario |
| | | El Líder de Proyecto puede visualizar el detalle de cada Proyecto. | Aceptado | Usuario |

| Pruebas de aceptación | | | | |
|------------------------------|------------------------|--|---------------|--------------------|
| ID Req. | Requerimiento | Criterio de aceptación | Estado | Responsable |
| | | El Líder de Proyecto puede visualizar el detalle del Cliente al que pertenece un Proyecto. | Aceptado | Usuario |
| | | El Líder de Proyecto puede crear un nuevo Proyecto en el que él será el Líder de Proyecto. | Aceptado | Usuario |
| | | El Líder de Proyecto puede editar la información de un Proyecto del cual él es Líder de Proyecto. | Aceptado | Usuario |
| | | El Líder de Proyecto puede eliminar por completo la información de un Proyecto del cual él es Líder de Proyecto. | Aceptado | Usuario |
| RF-06 | Gestionar actividades. | El Desarrollador, puede visualizar sus Actividades. | Aceptado | Usuario |
| | | El Desarrollador puede filtrar sus Actividades por mes. | Aceptado | Usuario |
| | | El Desarrollador puede filtrar sus Actividades por año. | Aceptado | Usuario |

| Pruebas de aceptación | | | | |
|------------------------------|----------------------|--|---------------|--------------------|
| ID Req. | Requerimiento | Criterio de aceptación | Estado | Responsable |
| | | El Desarrollador puede filtrar sus Actividades por Proyecto. | Aceptado | Usuario |
| | | El Desarrollador puede visualizar el detalle de una Actividad. | Aceptado | Usuario |
| | | El Desarrollador puede visualizar el detalle del Proyecto al que pertenece una Actividad. | Aceptado | Usuario |
| | | El Desarrollador puede crear una Actividad únicamente para sí mismo. | Aceptado | Usuario |
| | | El Desarrollador puede editar la información de sus Actividades. | Aceptado | Usuario |
| | | El Desarrollador puede eliminar una de sus Actividades. | Aceptado | Usuario |
| | | El Líder de Proyecto puede visualizar únicamente las Actividades de los Proyectos de los que está a cargo. | Aceptado | Usuario |
| | | El Líder de Proyecto puede filtrar las Actividades por mes. | Aceptado | Usuario |

| Pruebas de aceptación | | | | |
|------------------------------|------------------------------|---|---------------|--------------------|
| ID Req. | Requerimiento | Criterio de aceptación | Estado | Responsable |
| | | El Líder de Proyecto puede filtrar las Actividades por año. | Aceptado | Usuario |
| | | El Líder de Proyecto puede filtrar las Actividades por Proyecto. | Aceptado | Usuario |
| RF-07 | Registrar observaciones. | El Líder de Proyecto puede visualizar el detalle de una Actividad junto a sus Observaciones. | Aceptado | Usuario |
| | | El Líder de Proyecto puede crear observaciones para las actividades que así lo requieran. | Aceptado | Usuario |
| | | Cuando se crea una Observación el Desarrollador recibe una notificación por correo electrónico. | Aceptado | Usuario |
| | | El Líder de Proyecto puede editar la información de una Observación. | Aceptado | Usuario |
| | | El Líder de Proyecto puede eliminar una Observación. | Aceptado | Usuario |
| RF-08 | Gestionar las observaciones. | El Desarrollador puede visualizar en una página de Seguimiento de | Aceptado | Usuario |

| Pruebas de aceptación | | | | |
|------------------------------|----------------------|--|---------------|--------------------|
| ID Req. | Requerimiento | Criterio de aceptación | Estado | Responsable |
| | | Actividades que tienen Observación. | | |
| | | El Desarrollador puede acceder el detalle de una Actividad junto a sus Observaciones. | Aceptado | Usuario |
| | | El Desarrollador puede editar la información de la Actividad desde la vista del detalle. | Aceptado | Usuario |
| | | El Desarrollador puede marcar como resuelta una Observación. | Aceptado | Usuario |
| | | El Líder de Proyecto, puede visualizar en una página dedicada las Observaciones de sus proyectos. | Aceptado | Usuario |
| | | El Líder de Proyecto puede acceder al detalle de Seguimiento Actividades desde la página principal de Observaciones. | Aceptado | Usuario |
| RF-09 | Notificar proyectos. | El Líder de proyecto puede notificar al Gerente de Operaciones que las actividades de un | Aceptado | Usuario |

| Pruebas de aceptación | | | | |
|------------------------------|----------------------|---|---------------|--------------------|
| ID Req. | Requerimiento | Criterio de aceptación | Estado | Responsable |
| | | proyecto se encuentran revisadas y corregidas. | | |
| | | El Gerente de Operaciones recibe un correo en el que se especifica el nombre del Proyecto, el Cliente y el nombre del Líder de Proyecto que notifica. | Aceptado | Usuario |
| RF-10 | Generar reportes. | El Gerente de operaciones puede generar reportes de las actividades de un mes, año y por Cliente. | Aceptado | Usuario |
| | | El Desarrollador puede generar reportes de sus actividades de un mes y un año en específico. | Aceptado | Usuario |
| | | Si la consulta y generación de reporte no devuelve datos se muestra un mensaje en pantalla. | Aceptado | Usuario |
| | | Los reportes pueden ser previsualizados en la pantalla de la aplicación web como HTML. | Aceptado | Usuario |
| | | Los reportes se pueden descargar como PDF. | Aceptado | Usuario |

3.2.6.2. Pruebas de usabilidad

Con el fin de evaluar la experiencia y la facilidad de uso de la aplicación web, se realizaron pruebas de usabilidad que evalúan 5 importantes métricas.

Tabla 29. Tabla de pruebas de usabilidad de la aplicación web

| Pruebas de usabilidad | | | | |
|------------------------------|----------------------|--|---|---------------|
| No. | Métrica | Criterio de aceptación | Observación | Estado |
| 1 | Consistencia Gráfica | Los componentes de la aplicación web son consistentes en sus propiedades de diseño. | Todos los colores y formas utilizadas son acordes y se usan de manera coherente. | Aceptado |
| 2 | Navegación | Navegar por los diferentes menús y pantallas es fácil e intuitivo. | Elementos de uso común y fáciles de reconocer. | Aceptado |
| 3 | Manejo de errores | En caso de cometer errores en el uso de la aplicación, está dará retroalimentación al usuario. | Se muestran mensajes en casos de éxito y de error. | Aceptado |
| 4 | Fuente y tipografía | Las fuentes usadas en la aplicación tienen el tamaño y la tipografía adecuados. | El texto usa una tipografía común y sencilla, en colores que contrastan con el resto de la pantalla. | Aceptado |
| 5 | Guía del usuario | La aplicación contiene mensajes de ayuda y guía para el uso de esta. | Los botones y pantallas tienen íconos y textos que guían en el uso de la aplicación al usuario final. | Aceptado |

3.3. Resultados

Una vez culminada la implantación de la aplicación web y obteniendo un resultado positivo en las pruebas de aceptación y usabilidad realizadas en base a los requerimientos establecidos, se procede a evaluar estadísticamente los resultados de esta investigación mediante el Modelo de Aceptación Tecnológica (TAM).

3.3.1. Modelo TAM

Marco teórico ampliamente utilizado en el campo de la investigación tecnológica, propuesto en el año 1989 por Fred Davis. Este modelo se utiliza para determinar cómo usuarios conciben y adoptan una tecnología nueva [38]. Su análisis se basa en dos variables principales:

- Utilidad percibida (PU).
- Facilidad de uso percibida (PEOU).

3.3.1.1. Cuestionario para aplicar TAM

Para aplicar el cuestionario se ha utilizado una escala de Likert, en la cual 1 es el valor más bajo y 5 el más alto.

Tabla 30. Cuestionario de evaluación TAM

| Cuestionario de Evaluación | | | | |
|---|---------------------------|---------------------------------------|-------------------|------------------------------|
| Indica tu nivel de acuerdo con las siguientes afirmaciones donde: | | | | |
| 1 | 2 | 3 | 4 | 5 |
| Totalmente en desacuerdo | En desacuerdo | Ni de acuerdo ni en desacuerdo | De acuerdo | Totalmente de acuerdo |
| PU | UTILIDAD PERCIBIDA | | | |

| Cuestionario de Evaluación | | | | |
|---|--|---------------------------------------|-------------------|------------------------------|
| Indica tu nivel de acuerdo con las siguientes afirmaciones donde: | | | | |
| 1 | 2 | 3 | 4 | 5 |
| Totalmente en desacuerdo | En desacuerdo | Ni de acuerdo ni en desacuerdo | De acuerdo | Totalmente de acuerdo |
| PU1 | La aplicación web optimiza la organización de la información laboral del equipo de desarrollo. | | | |
| PU2 | El uso de la aplicación web mejora la eficiencia operativa en la gestión y control de la información laboral. | | | |
| PU3 | La aplicación web permite obtener reportes importantes dentro del área de operaciones de manera eficaz y eficiente. | | | |
| PU4 | El uso de la aplicación web permite tomar acciones de control, supervisión y corrección de errores de manera continua. | | | |
| PEOU | FACILIDAD DE USO PERCIBIDA | | | |
| PEOU1 | La aplicación web es fácil, sencilla y comprensible para usar. | | | |
| PEOU2 | El proceso de registro de información laboral mediante la aplicación web fue intuitivo y sencillo. | | | |
| PEOU3 | La gestión y control de actividades mediante la aplicación web es fácil y comprensible. | | | |
| PEOU4 | El proceso de generación de reportes mediante la aplicación web no requiere de esfuerzo para su uso y comprensión. | | | |

3.3.1.2. Resultados de aplicar TAM

El cuestionario se aplicó a las 8 personas del equipo de desarrollo de la empresa Software Evolutivo, entre Gerente de Operaciones, Líderes de Proyecto y Desarrolladores, se obtuvieron los siguientes resultados.

Utilidad percibida (PU)

Tabla 31. Tabla de frecuencias de TAM - Utilidad percibida

| Resultados de la utilidad percibida | | | | | |
|-------------------------------------|--------------------------|---------------|--------------------------------|---------------|-----------------------|
| PU | Totalmente en desacuerdo | En desacuerdo | Ni de acuerdo ni en desacuerdo | De acuerdo | Totalmente de acuerdo |
| PU1 | 0% | 0% | 0% | 37,5% | 62,5% |
| PU2 | 0% | 0% | 12,5% | 50% | 37,5% |
| PU3 | 0% | 0% | 0% | 25% | 75% |
| PU4 | 0% | 0% | 25% | 12,5% | 62,5% |
| MEDIA | 0% | 0% | 9,375% | 31,25% | 59,375% |

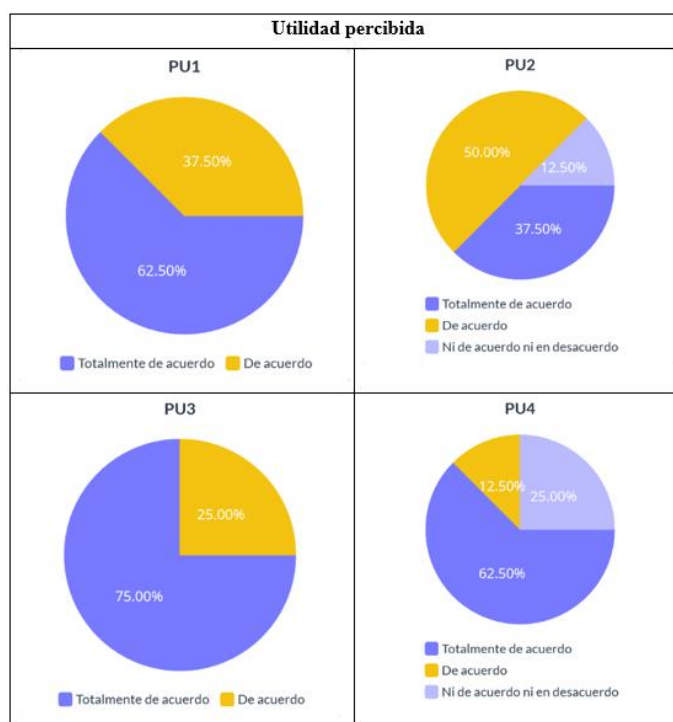


Figura 86. Resultados de TAM - Utilidad percibida

Los usuarios consideran una mejora en los procesos de Gestión y Control de la Información laboral, por lo tanto, se puede deducir que se encuentran satisfechos con la implantación de la aplicación web.

Facilidad de uso (PEOU)

Tabla 32. Tabla de frecuencias de TAM - Facilidad de uso

| Resultados de la facilidad de uso | | | | | |
|--|---------------------------------|----------------------|---------------------------------------|-------------------|------------------------------|
| PEOU | Totalmente en desacuerdo | En desacuerdo | Ni de acuerdo ni en desacuerdo | De acuerdo | Totalmente de acuerdo |
| PEOU1 | 0% | 0% | 0% | 50% | 50% |
| PEOU2 | 0% | 0% | 12,5% | 12,5% | 75% |
| PEOU3 | 0% | 0% | 12,5% | 25% | 62,5% |
| PEOU4 | 0% | 0% | 0% | 37,5% | 62,5% |
| MEDIA | 0% | 0% | 12,5% | 31,25% | 62,5% |

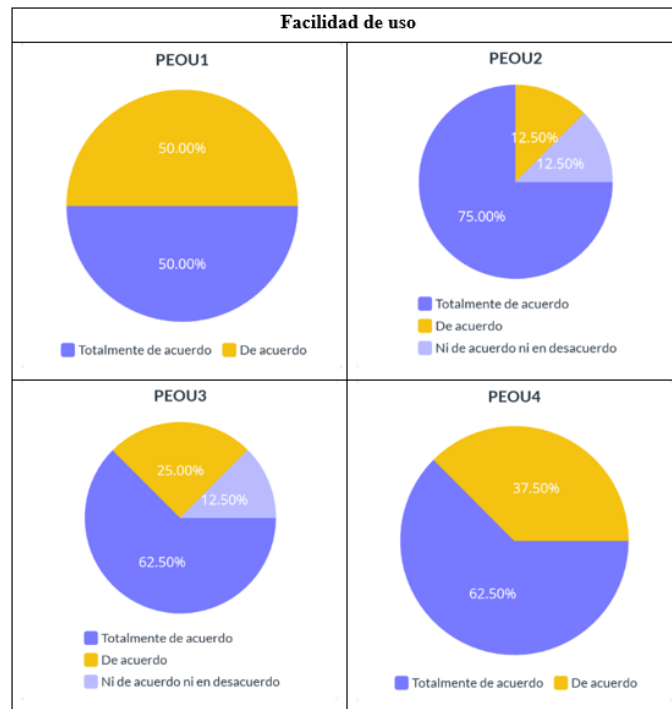


Figura 87. Resultados de TAM - Facilidad de uso

Se puede observar en los resultados de la facilidad de uso, que los usuarios consideran que esta aplicación web es fácil e intuitiva de usar, fortaleciendo así, los resultados de las pruebas de usabilidad.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

En esta investigación el objetivo planteado fue la implantación de una aplicación web para la Gestión y Control de la Información Laboral del equipo de desarrollo de la empresa Software Evolutivo, de la cual se concluye que:

- Se implantó una aplicación web para la Gestión y Control de la Información Laboral del equipo de desarrollo de la empresa Software Evolutivo que beneficia directamente al área de operaciones de la empresa, debido a que proporciona una solución y mejora a los procesos internos de la empresa. La capacidad de generar informes mensuales, gestionar Clientes y Proyectos, la comunicación entre los diferentes roles contribuye significativamente a la eficiencia operativa. Una vez realizadas las pruebas de aceptación y de usabilidad se demostró que el uso de las tecnologías y estándares adecuados permiten obtener un producto de software de calidad, es decir, fácil e intuitivo de usar, de utilidad y que entrega valor a los usuarios finales.
- Se analizaron los procesos que intervienen en la Gestión y Control de la Información Laboral y junto a la recolección de datos fueron dos pasos indispensables para el éxito del desarrollo y validación de esta aplicación web. Aplicar entrevistas correctamente formuladas a las personas implicadas en el proceso de Control y Gestión de la Información Laboral, permitió la comprensión adecuada de las necesidades de la empresa y de su entorno operativo, a partir de ello se pudo recolectar adecuadamente los requerimientos funcionales y no funcionales para el desarrollo de la aplicación web. La documentación detallada de los requerimientos, respaldada por reuniones y revisiones continuas con el cliente permitió abordar correctamente las funcionalidades esenciales del sistema, sentando así las bases para un desarrollo eficiente y preciso.

- Se determinaron las metodologías y herramientas necesarias para el desarrollo de la aplicación web, se llevó a cabo un análisis exhaustivo con el objetivo de establecer una base sólida para el proyecto. Este proceso de toma de decisiones es un paso crítico para garantizar un desarrollo eficiente, modular y orientado a la reutilización. El análisis realizado para determinar las metodologías y herramientas adecuadas permitió establecer una base sólida para el proyecto desde el inicio. Este proceso de elección no solo consideró la adecuación técnica de las herramientas, sino también su capacidad para adaptarse a las necesidades específicas del proyecto y del cliente Software Evolutivo. En particular, para el desarrollo ágil de software, se eligió la metodología Extreme Programming, esta permitió al programador mantener una colaboración continua con el cliente de la empresa Software Evolutivo, facilitando el progreso de la investigación y entrega de valor mediante la misma. También se observó la eficiencia en el levantamiento de requerimientos, la adaptabilidad a cambios, ideas y mejoras que surgieron durante todo el proceso.
- Se desarrollaron los módulos informáticos para la Gestión y Control de la Información Laboral de Software Evolutivo, el uso del framework JHipster fue crucial para este proceso. Su flexibilidad en la integración de diferentes tecnologías y lenguajes, junto con su capacidad para generar una aplicación web contenerizada, proporcionó un entorno de desarrollo eficiente y modular, lo que facilitó la implementación ágil de los requerimientos. Para la base de datos PostgreSQL, se implementó un esquema con todas las tablas necesarias que permitió mantener la integridad, confiabilidad de la información, la seguridad y el tratamiento de datos, además, se contó con el conocimiento y experticia del Encargado de Datos de la empresa Software Evolutivo.
- El valor principal de la aplicación web es que es una herramienta con tecnologías actuales, siguiendo estándares de UI/UX, alineada a las buenas prácticas de desarrollo, con una arquitectura adecuada a sus necesidades, contenerizada pensando en la escalabilidad de esta. Esto no solo permitió obtener hoy un software de calidad, sino que dejó sentadas las bases para futuras mejoras y optimizaciones que se le puedan implantar a la aplicación.

4.2. Recomendaciones

- Se recomienda para futuras investigaciones, buscar siempre la participación de todos los roles que estén involucrados, esto permite obtener una retroalimentación continua que se traduce en un mejor producto al final.
- Antes del inicio del desarrollo de un software es importante tener lo más claras posibles las expectativas del cliente, los criterios de aceptación y su definición del producto.
- El uso de una metodología de desarrollo ágil de software permite implementar un enfoque iterativo y de revisiones frecuentes con el cual podremos asegurar que el desarrollo está alineado con los requerimientos funcionales y no funcionales definidos.
- Se recomienda usar frameworks que no solo sean actuales, sino que sean versiones estables y testeadas, que tengan soporte activo, y como adicional, que cuenten con una comunidad de desarrolladores activa, esto puede ser de mucha utilidad para el desarrollo de software.
- Se recomienda a la empresa Software Evolutivo que se realicen periódicamente retroalimentaciones de los usuarios finales de la aplicación para poder encontrar mejoras y optimizaciones que se puedan realizar tanto a la aplicación como al proceso de Gestión y Control de la Información Laboral.
- Este proyecto de investigación puede ser compartido y aplicado en otras empresas de desarrollo de software donde también se requiera mejorar los procesos internos de operaciones.

REFERENCIAS BIBLIOGRÁFICAS

- [1] B. Ortiz Alvarez, “Herramienta para la gestión de actividades en los proyectos de software”, 2022, Consultado: el 18 de noviembre de 2023. [En línea]. Disponible en: <https://bibliotecadigital.udea.edu.co/handle/10495/31918>
- [2] J. J. Zapata Arbeláez, G. P. Gasca-Hurtado, B. Manrique-Losada, y L. Machuca-Villegas, “Caracterización de métodos de evaluación de desempeño para equipos de desarrollo de software”, *Ingeniare Rev. Chil. Ing.*, vol. 29, núm. 1, pp. 129–140, mar. 2021, doi: 10.4067/S0718-33052021000100129.
- [3] E. Moya Serrat y L. V. Callejas Pizanan, “Análisis diseño e implementación de un sistema de gestión de Talento Humano en la comunidad Salesiana San Juan Bosco.”, bachelorThesis, 2015. Consultado: el 18 de noviembre de 2023. [En línea]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/10342>
- [4] A. B. Tacuri Luna y E. M. Pérez Quishpe, “Desarrollo de aplicación web y móvil para control de operaciones eléctricas orientado a trabajadores de empresa Rimaelec.”, bachelorThesis, Quito, 2022, 2022. Consultado: el 18 de noviembre de 2023. [En línea]. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/22199>
- [5] L. S. Taco Vilatuña, “Desarrollo de un sistema móvil de gestión del talento humano para empresas de seguridad”, bachelorThesis, QUITO/EPN/2013, 2013. Consultado: el 18 de noviembre de 2023. [En línea]. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/6679>
- [6] C. V. Padilla Vega y C. A. Quispe Carrión, “Sistema web para la gestión y retención del talento de una empresa outsourcing de TI basado en el aprendizaje automático”, *Univ. Peru. Cienc. Apl. UPC*, jun. 2022, Consultado: el 5 de febrero de 2024. [En línea]. Disponible en: <https://repositorioacademico.upc.edu.pe/handle/10757/660875>
- [7] F. G. Cerna Fukuzaki, “Sistema WebVR para el proceso de evaluación psicológica en una consultora de recursos humanos en Lima 2019”, *Univ.*

Tecnológica Perú, 2019, Consultado: el 5 de febrero de 2024. [En línea]. Disponible en: <http://repositorio.utp.edu.pe/handle/20.500.12867/1857>

[8] C. Gomez Ruiz y E. Pulido Neira, “Diseño e implementación de un sistema web para categorización docente de la institución universitaria antonio José camacho (catedoc)”, feb. 2021, Consultado: el 5 de febrero de 2024. [En línea]. Disponible en: <https://repositorio.uniajc.edu.co/handle/uniajc/242>

[9] “Estudio comparativo para la toma de requerimientos de software para el desarrollo de una aplicación web en investigación psicológica del bullying educativo: Comparative study for software requirements gathering for the development of a web application in psychological research on educational bullying | LATAM Revista Latinoamericana de Ciencias Sociales y Humanidades”. Consultado: el 5 de febrero de 2024. [En línea]. Disponible en: <http://latam.redilat.org/index.php/lt/article/view/906>

[10] J. K. Pinto, *Gerencia de proyectos cómo lograr la ventaja competitiva*. Pearson, 2015. Consultado: el 18 de noviembre de 2023. [En línea]. Disponible en: <http://repositorio.uasb.edu.bo/handle/54000/1207>

[11] A. M. Jara Martínez, N. S. Asmat Vega, N. E. Alberca Pintado, y J. J. Medina Guzmán, “Gestión del talento humano como factor de mejoramiento de la gestión pública y desempeño laboral”, *Rev. Venez. Gerenc.*, vol. 23, núm. 83, pp. 740–760, ago. 2019, doi: 10.37960/revista.v23i83.24500.

[12] R. N. Martínez, *El Proceso de Desarrollo de Software*. IT Campus Academy, 2015.

[13] J. L. Jurado-Muñoz y C. J. Pardo-Calvache, “La Gestión de Proyectos Software, una Prospectiva en la Aplicación de Estrategias en la Ingeniería Colaborativa”, *Lámpsakos Rev. Descontinuada*, núm. 9, Art. núm. 9, ene. 2013, doi: 10.21501/21454086.854.

[14] J. A. Amaya, *Sistemas de información gerenciales: Hardware, software, redes, Internet, diseño*. ECOE EDICIONES, 2010.

- [15] R. M. ALICIA y R. M. JESUS MARIA, *Aplicaciones Web 2.^a edición*. Ediciones Paraninfo, S.A., 2014.
- [16] S. Luján-Mora, *Programación de aplicaciones web: historia, principios básicos y clientes web*. Editorial Club Universitario, 2002. Consultado: el 18 de noviembre de 2023. [En línea]. Disponible en: <http://rua.ua.es/dspace/handle/10045/16995>
- [17] “Framework: qué es, para qué sirve y algunos ejemplos”, UNIR FP. Consultado: el 18 de noviembre de 2023. [En línea]. Disponible en: <https://unirfp.unir.net/revista/ingenieria-y-tecnologia/framework/>
- [18] N. Simmonds Samper, “Análisis de frameworks para desarrollo de aplicaciones móviles y web”, 2019, Consultado: el 18 de noviembre de 2023. [En línea]. Disponible en: <https://repositorio.uniandes.edu.co/entities/publication/c1e22684-b39c-43ce-b93a-397a607598b9>
- [19] A. Banks y E. Porcello, *Learning React: Functional Web Development with React and Redux*. O’Reilly Media, Inc., 2017.
- [20] A. Fedosejev, *React.js Essentials*. Packt Publishing Ltd, 2015.
- [21] G. Geetha, M. Mittal, K. M. Prasad, y J. G. Ponsam, “Interpretation and Analysis of Angular Framework”, en *2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, dic. 2022, pp. 1–6. doi: 10.1109/ICPECTS56089.2022.10047474.
- [22] O. Filipova, *Learning Vue.js 2*. Packt Publishing Ltd, 2016.
- [23] C. Burch, “Django, a web framework using Python: tutorial presentation”, *J. Comput. Sci. Coll.*, vol. 25, núm. 5, pp. 154–155, may 2010.
- [24] R. Johnson *et al.*, “Spring Framework Reference Documentation”.

- [25] Y. Jiang, “Research on Application Value of Computer Software Development in Java Programming Language”, *J. Phys. Conf. Ser.*, vol. 1648, núm. 3, p. 032152, oct. 2020, doi: 10.1088/1742-6596/1648/3/032152.
- [26] E. P. Martínez, *Desarrollo de aplicaciones mediante el Framework de spring*. Ra-Ma Editorial, 2016.
- [27] A. Villa, A. Puerta, y R. Núñez, *Curso de Consultoría TIC. Gestión, Software ERP y CRM*. IT Campus Academy, 2015.
- [28] M. V. N. Cabello, *Introducción a las Bases de Datos relacionales*. Vision Libros, 2010.
- [29] I. H. G. del Busto y I. O. Y. Enríquez, “BASES DE DATOS NoSQL”, *Telemática*, vol. 11, núm. 3, Art. núm. 3, 2012.
- [30] P. Beynon-Davies, *Sistemas de bases de datos*. Reverte, 2018.
- [31] D. K. Sasidharan y S. K. N, *Full Stack Development with JHipster: Build full stack applications and microservices with Spring Boot and modern JavaScript frameworks, 2nd Edition*. Packt Publishing Ltd, 2020.
- [32] A. N. Cadavid, “Revisión de metodologías ágiles para el desarrollo de software”, *Prospectiva*, vol. 11, núm. 2, p. 30, sep. 2013, doi: 10.15665/rp.v11i2.36.
- [33] J. J. Chavez Elias, “Factores y modelos sobre metodologías híbridas para el desarrollo de software: Una revisión sistemática de la literatura”, dic. 2020, Consultado: el 3 de enero de 2024. [En línea]. Disponible en: <https://repositorio.upeu.edu.pe/handle/20.500.12840/4141>
- [34] A. Stellman y J. Greene, *Learning Agile: Understanding Scrum, XP, Lean, and Kanban*. O’Reilly Media, Inc., 2014.
- [35] M. W. A. Kesiman, I. M. D. Maysanjaya, y I. M. G. Sunarya, *IConVET 2021: Proceedings of the 4th International Conference on Vocational Education*

and Technology, IConVET 2021, 27 November 2021, Singaraja, Bali, Indonesia.
European Alliance for Innovation, 2022.

[36] J. Joskowicz, “Reglas y Prácticas en eXtreme Programming”.

[37] A. Rosado-Gómez, A. Quintero-Duarte, y C. D. Meneses-Guevara, “Desarrollo ágil de software aplicando programación extrema”, *Rev. Ingenio*, vol. 5, núm. 1, Art. núm. 1, dic. 2012, doi: 10.22463/2011642X.2003.

[38] J. Cabero-Almenara, V. Marín-Díaz, y B. E. Sampedro-Requena, “Aceptación del Modelo Tecnológico en la enseñanza superior”, *Rev. Investig. Educ.*, vol. 36, núm. 2, Art. núm. 2, jun. 2018, doi: 10.6018/rie.36.2.292951.

ANEXOS

Anexo A. Guía de entrevista para Gerente de Operaciones

| Estimado Gerente de Operaciones | | | |
|---|--|------------------|--------------------|
| <i>La información recolectada en esta entrevista será utilizada únicamente para fines académicos.</i> | | | |
| Entrevistador: Mateo Martínez | | | |
| Nº | Pregunta | Respuesta | Observación |
| 1 | ¿Cuáles son sus principales responsabilidades y objetivos como gerente de operaciones en la empresa? | | |
| 2 | ¿Qué tipo de información y datos necesita para tomar decisiones estratégicas en la empresa? | | |
| 3 | ¿Qué problemas ha encontrado en los datos para tomar decisiones? | | |
| 4 | ¿Cómo evalúa actualmente el rendimiento de los proyectos y el equipo de desarrollo? | | |
| 5 | ¿En base a la información que dispone, considera que la evaluación de los proyectos ha sido la correcta, si su respuesta es negativa, por qué? | | |
| 6 | ¿Qué tipo de informes o reportes le serían más útiles en la gestión de operaciones? | | |
| 7 | ¿Cuál sería el tipo de aplicación informática en la que se sentiría más a gusto para revisar informes o analizar | | |

Estimado Gerente de Operaciones

La información recolectada en esta entrevista será utilizada únicamente para fines académicos.

Entrevistador: Mateo Martínez

| N° | Pregunta | Respuesta | Observación |
|----|--|-----------|-------------|
| | informes del registro de actividades? (Aplicación web, móvil, etc.) | | |
| 8 | ¿Hay algún otro aspecto importante que desee agregar sobre la gestión de información laboral en la empresa "Software Evolutivo"? | | |

Anexo B. Guía de entrevista para Líderes de Proyecto

| Estimado Líder de Proyecto | | | |
|---|--|-----------|-------------|
| <i>La información recolectada en esta entrevista será utilizada únicamente para fines académicos.</i> | | | |
| Entrevistador: Mateo Martínez | | | |
| N° | Pregunta | Respuesta | Observación |
| 1 | ¿Cuáles son sus responsabilidades como líder de proyecto en "Software Evolutivo"? | | |
| 2 | ¿Cómo supervisa y gestiona actualmente las actividades laborales de los desarrolladores en sus proyectos? | | |
| 3 | ¿Qué problemas ha encontrado al supervisar y gestionar las actividades laborales de los desarrolladores? | | |
| 4 | ¿Cómo se comunica con los desarrolladores sobre novedades o cambios en las actividades laborales? | | |
| 5 | ¿Qué problemas ha encontrado el momento de comunicar a los desarrolladores sobre novedades o cambios en las actividades laborales? | | |
| 6 | ¿Qué información o funcionalidades específicas le ayudarían a ser más efectivo en su rol? | | |
| 7 | ¿Cuál sería el tipo de aplicación informática en la que se sentiría más a gusto para supervisar y gestionar | | |

Estimado Líder de Proyecto

La información recolectada en esta entrevista será utilizada únicamente para fines académicos.

Entrevistador: Mateo Martínez

| N° | Pregunta | Respuesta | Observación |
|----|--|-----------|-------------|
| | actividades? (Aplicación web, móvil, etc.) | | |
| 8 | ¿Hay algún otro aspecto importante que desee agregar sobre la gestión de información laboral en la empresa "Software Evolutivo"? | | |

Anexo C. Guía de entrevista para Desarrolladores de Software

| Estimado Desarrollador | | | |
|---|--|------------------|--------------------|
| <i>La información recolectada en esta entrevista será utilizada únicamente para fines académicos.</i> | | | |
| Entrevistador: Mateo Martínez | | | |
| N° | Pregunta | Respuesta | Observación |
| 1 | ¿Cuáles son las principales tareas y actividades que realiza como desarrollador en "Software Evolutivo"? | | |
| 2 | ¿Registra sus actividades laborales diarias? | | |
| 3 | Si su respuesta fue Sí, ¿Cómo las registra actualmente? | | |
| 4 | ¿Qué problemas ha enfrentado al registrar sus actividades laborales o al comunicarse con sus líderes de proyecto? | | |
| 5 | ¿Cómo han solucionado los problemas encontrados al registrar sus actividades laborales? | | |
| 6 | ¿Qué características o funcionalidades le resultarían más útiles en una aplicación de gestión de las actividades laborales? | | |
| 7 | ¿Cuál sería el tipo de aplicación informática en la que se sentiría más a gusto para registrar sus actividades? (Aplicación web, móvil, etc.) | | |
| 8 | ¿Hay algún otro aspecto importante que desee agregar sobre la gestión de | | |

Estimado Desarrollador

La información recolectada en esta entrevista será utilizada únicamente para fines académicos.

Entrevistador: Mateo Martínez

| N° | Pregunta | Respuesta | Observación |
|----|--|-----------|-------------|
| | información laboral en la empresa "Software Evolutivo"? | | |

Anexo D. Test de integración del método para Inicio de Sesión

En la Figura D1 se muestra un primer test en el cual se simula un usuario que no existe en la base de datos, y lo que se espera al enviar al método de inicio de sesión es que responda que no está autorizado.

```
@Test
@Throws(Exception::class)
fun testAuthorizeFails() {
    val login = LoginVM(username = "wrong-user", password = "wrong password")
    mockMvc.perform(
        post( urlTemplate: "/api/authenticate")
            .contentType(MediaType.APPLICATION_JSON)
            .content(convertObjectToJsonBytes(login))
        )
        .andExpect(status().isUnauthorized)
        .andExpect(jsonPath( expression: "\$.id_token").doesNotExist())
        .andExpect(header().doesNotExist( name: "Authorization"))
}
```

Figura D1. Test de integración de autorización fallida

En la Figura D2 mediante un test, se simula un usuario activo y con una contraseña correcta, por lo tanto, se espera que la respuesta sea positiva y se autorice al usuario a entrar a la aplicación.

```

@Test
@Transactional
@Throws(Exception::class)
fun testAuthorize() {
    val user = User(
        login = "user-jwt-controller",
        email = "user-jwt-controller@example.com",
        activated = true,
        password = passwordEncoder.encode( rawPassword: "test")
    )

    userRepository.saveAndFlush(user)

    val login = LoginVM(username = "user-jwt-controller", password = "test")
    mockMvc.perform(
        post( uriTemplate: "/api/authenticate")
            .contentType(MediaType.APPLICATION_JSON)
            .content(convertObjectToJsonBytes(login))
        )
        .andExpect(status().isOk)
        .andExpect(jsonPath( expression: "\$.id_token").isString)
        .andExpect(jsonPath( expression: "\$.id_token").isNotEmpty)
        .andExpect(header().string( name: "Authorization", not( nullValue() )))
        .andExpect(header().string( name: "Authorization", not( `is` emptyString() )))
    }
}

```

Figura D2. Test de integración de autorización correcta de usuario

En la Figura D3 se muestra un test en el cual se repite la autorización correcta de un usuario, con la diferencia de que aquí se envía en verdadero el parámetro que permite recordar el usuario para la sesión.

```

@Test
@Transactional
@Throws(Exception::class)
fun testAuthorizeWithRememberMe() {
    val user = User(
        login = "user-jwt-controller-remember-me",
        email = "user-jwt-controller-remember-me@example.com",
        activated = true,
        password = passwordEncoder.encode(rawPassword: "test")
    )

    userRepository.saveAndFlush(user)

    val login = LoginVM(
        username = "user-jwt-controller-remember-me",
        password = "test",
        isRememberMe = true
    )

    mockMvc.perform(
        post(uriTemplate: "/api/authenticate")
            .contentType(MediaType.APPLICATION_JSON)
            .content(convertObjectToJsonBytes(login))
    )

        .andExpect(status().isOk)
        .andExpect(jsonPath("expression: \"\$.id_token\").isString)
        .andExpect(jsonPath("expression: \"\$.id_token\").isNotEmpty)
        .andExpect(header().string(name: "Authorization", not(nullValue()))))
        .andExpect(header().string(name: "Authorization", not(`is`emptyString()))))
}

```

Figura D3. Test de integración de autorización con opción de recordar usuario

Anexo E. Test de integración del método de validación de rol Desarrollador

En la Figura E1 se muestran dos tests en los cuales se simula la conexión de un usuario con rol de Desarrollador y otro con rol Anónimo, por lo tanto, la funcionalidad de validación de rol en el primer test se espera que el método devuelva verdadero y en el segundo falso.

```

  👤 Mateo Martinez
  @Test
  @WithMockUser(authorities = [DEVELOPER])
  fun shouldReturnTrueIfUserIsDeveloper() {
    val isDeveloper = userService.userIsDeveloper()
    assertTrue(isDeveloper)
  }

  👤 Mateo Martinez
  @Test
  @WithMockUser(authorities = [ANONYMOUS])
  fun shouldReturnFalseIfUserIsNotDeveloper() {
    val isDeveloper = userService.userIsDeveloper()
    assertFalse(isDeveloper)
  }

```

Figura E1. Tests de integración para validación de rol Desarrollador

Anexo F. Test de integración del método de validación de rol Líder de Proyecto

En la Figura F1 se muestran dos tests en los cuales se simula la conexión de un usuario con rol de Líder de Proyecto y otro con rol Anónimo, por lo tanto, la funcionalidad de validación de rol en el primer test se espera que el método devuelva verdadero y en el segundo falso.

```
↳ Mateo Martinez
@Test
@WithMockUser(authorities = [PROJECT_LEADER])
fun shouldReturnTrueIfUserIsProjectLeader() {
    val isProjectLeader = userCoreService.userIsProjectLeader()
    assertTrue(isProjectLeader)
}

↳ Mateo Martinez
@Test
@WithMockUser(authorities = [ANONYMOUS])
fun shouldReturnTrueIfUserIsNotProjectLeader() {
    val isProjectLeader = userCoreService.userIsProjectLeader()
    assertFalse(isProjectLeader)
}
```

Figura F1. Tests de integración para validación de rol Líder de Proyecto

Anexo G. Test de integración del método de validación de rol Gerente de Operaciones

En la Figura G1 se muestran dos tests que simulan la conexión de un usuario con rol de Gerente de Operaciones y otro con rol Anónimo, por lo tanto, la funcionalidad de validación de rol en el primer test se espera que el método devuelva verdadero y en el segundo falso.

```
↳ Mateo Martinez
@Test
@WithMockUser(authorities = [OPS_MANAGER])
fun shouldReturnTrueIfUserIsOpsManager() {
    val isOpsManager = userCoreService.userIsOpsManager()
    assertTrue(isOpsManager)
}

↳ Mateo Martinez
@Test
@WithMockUser(authorities = [ANONYMOUS])
fun shouldReturnTrueIfUserIsNotOpsManager() {
    val isOpsManager = userCoreService.userIsOpsManager()
    assertFalse(isOpsManager)
}
```

Figura G1. Tests de integración para validación de rol rol Gerente de Operaciones

Anexo H. Test de integración del método de validación de rol Administrador

En la Figura H1 se muestran dos tests que simulan la conexión de un usuario con rol de Administrador y otro con rol Anónimo, por lo tanto, la funcionalidad de validación de rol en el primer test se espera que el método devuelva verdadero y en el segundo falso.

```
↳ Mateo Martinez
@Test
@WithMockUser(authorities = [ADMIN])
fun shouldReturnTrueIfUserIsAdmin() {
    val isAdmin = userCoreService.userIsAdmin()
    assertTrue(isAdmin)
}

↳ Mateo Martinez
@Test
@WithMockUser(authorities = [ANONYMOUS])
fun shouldReturnFalseIfUserIsNotAdmin() {
    val isAdmin = userCoreService.userIsAdmin()
    assertFalse(isAdmin)
}
```

Figura H1. Tests de integración para validación de rol Administrador

Anexo I. Tests de integración del proceso de creación de un registro de una entidad

En la Figura I1 se muestra un test que llama al controlador enviándole la información de la entidad para posteriormente comprobar que el tamaño de la base de datos haya aumentado en uno, y que el registro de la entidad tenga todos los datos asignados correctamente.

```
@Test
@Transactional
@Throws(Exception::class)
fun createCustomer() {
    val databaseSizeBeforeCreate = customerRepository.findAll().size
    // Create the Customer
    restCustomerMockMvc.perform(
        post(ENTITY_API_URL)
            .contentType(MediaType.APPLICATION_JSON)
            .content(convertObjectToJsonBytes(customer))
    ).andExpect(status().isCreated)

    // Validate the Customer in the database
    val customerList = customerRepository.findAll()
    assertThat(customerList).hasSize(databaseSizeBeforeCreate + 1)
    val testCustomer = customerList[customerList.size - 1]

    assertThat(testCustomer.name).isEqualTo(DEFAULT_NAME)
    assertThat(testCustomer.country).isEqualTo(DEFAULT_COUNTRY)
    assertThat(testCustomer.city).isEqualTo(DEFAULT_CITY)
    assertThat(testCustomer.address).isEqualTo(DEFAULT_ADDRESS)
    assertThat(testCustomer.email).isEqualTo(DEFAULT_EMAIL)
    assertThat(testCustomer.phone).isEqualTo(DEFAULT_PHONE)
    assertThat(testCustomer.startDate).isEqualTo(DEFAULT_START_DATE)
}
```

Figura I1. Test de integración de creación correcta de un registro

En la Figura I2 se muestra el test en el cual se intenta crear un registro asignándole un ID manualmente, lo que se espera es que el controlador devuelva un Bad Request y que el tamaño de la base de datos sea el mismo que antes de intentar la creación del registro.

```
@Test
@Transactional
@Throws(Exception::class)
fun createCustomerWithExistingId() {
    // Create the Customer with an existing ID
    customer.id = 1L

    val databaseSizeBeforeCreate = customerRepository.findAll().size
    // An entity with an existing ID cannot be created, so this API call must fail
    restCustomerMockMvc.perform(
        post(ENTITY_API_URL)
            .contentType(MediaType.APPLICATION_JSON)
            .content(convertObjectToJsonBytes(customer))
    ).andExpect(status().isBadRequest)

    // Validate the Customer in the database
    val customerList = customerRepository.findAll()
    assertThat(customerList).hasSize(databaseSizeBeforeCreate)
}
```

Figura I2. Test de integración de creación de registro con ID asignado

Anexo J. Tests de integración del proceso de actualización de un registro de una entidad

En la Figura J1 se muestra un test en el cual se actualizan los campos de un registro de la entidad en cuestión, validando que el tamaño de la base de datos sea el mismo que antes de llamar al proceso del controlador y que los campos que se encuentren en este registro actualizado sean los esperados.

```
fun putExistingCustomer() {
    // Initialize the database
    customerRepository.saveAndFlush(customer)

    val databaseSizeBeforeUpdate = customerRepository.findAll().size

    // Update the customer
    val updatedCustomer = customerRepository.findById(customer.id).get()
    // Disconnect from session so that the updates on updatedCustomer are not directly saved in db
    em.detach(updatedCustomer)
    updatedCustomer.name = UPDATED_NAME
    updatedCustomer.country = UPDATED_COUNTRY
    updatedCustomer.city = UPDATED_CITY
    updatedCustomer.address = UPDATED_ADDRESS
    updatedCustomer.email = UPDATED_EMAIL
    updatedCustomer.phone = UPDATED_PHONE
    updatedCustomer.startDate = UPDATED_START_DATE

    restCustomerMockMvc.perform(
        put(ENTITY_API_URL_ID, updatedCustomer.id)
            .contentType(MediaType.APPLICATION_JSON)
            .content(convertObjectToJsonBytes(updatedCustomer))
    ).andExpect(status().isOk)

    // Validate the Customer in the database
    val customerList = customerRepository.findAll()
    assertThat(customerList).hasSize(databaseSizeBeforeUpdate)
    val testCustomer = customerList[customerList.size - 1]
    assertThat(testCustomer.name).isEqualTo(UPDATED_NAME)
    assertThat(testCustomer.country).isEqualTo(UPDATED_COUNTRY)
    assertThat(testCustomer.city).isEqualTo(UPDATED_CITY)
    assertThat(testCustomer.address).isEqualTo(UPDATED_ADDRESS)
    assertThat(testCustomer.email).isEqualTo(UPDATED_EMAIL)
    assertThat(testCustomer.phone).isEqualTo(UPDATED_PHONE)
    assertThat(testCustomer.startDate).isEqualTo(UPDATED_START_DATE)
}
```

Figura J1. Test de integración de actualización correcta de un registro

En la Figura J2 se muestra un test en el cual se intenta actualizar un registro de la base de datos del cual no se puede encontrar su id, por lo tanto, lo que se espera en un Bad Request.

```
@Test
@Transactional
fun putNonExistingCustomer() {
    val databaseSizeBeforeUpdate = customerRepository.findAll().size
    customer.id = count.incrementAndGet()

    // If the entity doesn't have an ID, it will throw BadRequestAlertException
    restCustomerMockMvc.perform(
        put(ENTITY_API_URL_ID, customer.id)
            .contentType(MediaType.APPLICATION_JSON)
            .content(convertObjectToJsonBytes(customer))
    )
        .andExpect(status().isBadRequest)

    // Validate the Customer in the database
    val customerList = customerRepository.findAll()
    assertThat(customerList).hasSize(databaseSizeBeforeUpdate)
}
```

Figura J2. Test de actualización de un registro con ID no encontrado

Anexo K. Test de integración del proceso de obtención de los registros de una entidad

En la Figura K1 se muestra un test, en el cual se guarda un registro de la entidad y se hace la petición GET, esperando como respuesta que se retorne la entidad con toda su información completa.

```
@Test
@Transactional
@Throws(Exception::class)
fun getAllCustomers() {
    // Initialize the database
    customerRepository.saveAndFlush(customer)

    // Get all the customerList
    restCustomerMockMvc.perform(get( uriTemplate: ENTITY_API_URL + "?sort=id_desc"))
        .andExpect(status().isOk)
        .andExpect(content().contentType(MediaType.APPLICATION_JSON_VALUE))
        .andExpect(jsonPath("expression: "$[*].id").value(hasItem(customer.id?.toInt())))
        .andExpect(jsonPath("expression: "$[*].name").value(hasItem(DEFAULT_NAME)))
        .andExpect(jsonPath("expression: "$[*].country").value(hasItem(DEFAULT_COUNTRY)))
        .andExpect(jsonPath("expression: "$[*].city").value(hasItem(DEFAULT_CITY)))
        .andExpect(jsonPath("expression: "$[*].address").value(hasItem(DEFAULT_ADDRESS)))
        .andExpect(jsonPath("expression: "$[*].email").value(hasItem(DEFAULT_EMAIL)))
        .andExpect(jsonPath("expression: "$[*].phone").value(hasItem(DEFAULT_PHONE)))
        .andExpect(jsonPath("expression: "$[*].startDate").value(hasItem(DEFAULT_START_DATE.toString())))
}
```

Figura K1. Test de integración para la correcta obtención de registros

Anexo L. Tests de integración del proceso de obtención de un registro de una entidad

En la Figura L1 se muestra un test en el cual se inserta un registro de una entidad en la base de datos para posteriormente solicitarlo al método del controlador.

```
@Test
@Transactional
@Throws(Exception::class)
fun getCustomer() {
    // Initialize the database
    customerRepository.saveAndFlush(customer)

    val id = customer.id
    assertNotNull(id)

    // Get the customer
    restCustomerMockMvc.perform(get(ENTITY_API_URL_ID, customer.id))
        .andExpect(status().isOk)
        .andExpect(content().contentType(MediaType.APPLICATION_JSON_VALUE))
        .andExpect(jsonPath("id").value(customer.id?.toInt()))
        .andExpect(jsonPath("name").value(DEFAULT_NAME))
        .andExpect(jsonPath("country").value(DEFAULT_COUNTRY))
        .andExpect(jsonPath("city").value(DEFAULT_CITY))
        .andExpect(jsonPath("address").value(DEFAULT_ADDRESS))
        .andExpect(jsonPath("email").value(DEFAULT_EMAIL))
        .andExpect(jsonPath("phone").value(DEFAULT_PHONE))
        .andExpect(jsonPath("startDate").value(DEFAULT_START_DATE.toString()))
}
```

Figura L1. Test de obtención de un registro

En la Figura L2 se muestra un test en el cual no existen registros en la base de datos, por lo tanto, lo que se espera como respuesta es un Not Found.

```
@Test
@Transactional
@Throws(Exception::class)
fun getNonExistingCustomer() {
    // Get the customer
    restCustomerMockMvc.perform(get(ENTITY_API_URL_ID, Long.MAX_VALUE))
        .andExpect(status().isNotFound)
}
```

Figura L2. Test de obtención de un registro no existente

Anexo M. Test de integración del proceso de eliminación de un registro

En la Figura M1 se muestra un test en el cual se inserta un registro en la base de datos para poder mediante el método del controlador eliminarlo enviando su ID. Se valida que el tamaño de la base de datos sea uno menor al tamaño antes de enviar la petición.

```
@Test
@Transactional
@Throws(Exception::class)
fun deleteCustomer() {
    // Initialize the database
    customerRepository.saveAndFlush(customer)
    val databaseSizeBeforeDelete = customerRepository.findAll().size
    // Delete the customer
    restCustomerMockMvc.perform(
        delete(ENTITY_API_URL_ID, customer.id)
            .accept(MediaType.APPLICATION_JSON)
    ).andExpect(status().isNoContent)

    // Validate the database contains one less item
    val customerList = customerRepository.findAll()
    assertThat(customerList).hasSize(databaseSizeBeforeDelete - 1)
}
```

Figura M1. Test de integración de la correcta eliminación de un registro

Anexo N. Template del envío de correo de Nueva Observación

En la Figura N1 se muestra la estructura en HTML del correo de Nueva Observación, en este se declaran los estilos y elementos con variables que deberán ser declaradas mediante un archivo de propiedades.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org" th:lang="#{locale.language}" lang="en">
  <head>
    <title th:text="#{email.newobservation.title}"></title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  </head>
  <body style="margin: 0; padding: 0">
    <p
      style="font-size: 15px; margin: 0 0 30px 0; font-family: Arial, sans-serif"
      th:text="#{email.newobservation.dear({completeName})}"
    ></p>
    <p
      style="font-size: 15px; font-family: Arial, sans-serif"
      th:text="#{email.newobservation.asunt({observationDescription}, {activityId})}"
    ></p>
    <p
      style="font-size: 15px; margin: 0 0 55px 0; font-family: Arial, sans-serif"
      th:text="#{email.newobservation.url({baseUrl}, {activityId})}"
    ></p>
    <p style="font-size: 15px; font-family: Arial, sans-serif">--</p>
    <p style="font-size: 15px; font-family: Arial, sans-serif" th:text="#{email.newobservation.firm}"></p>
  </body>
</html>
```

Anexo N1. Template del correo de Nueva Observación

En la Figura N2 se presentan las variables que reemplazarán en el template junto a los atributos que se envíen desde el método de envío de correos.

```
# New observation
email.newobservation.title=Una nueva observacion a sus actividades ha sido creada.
email.newobservation.dear=Estimado, {0}.
email.newobservation.asunt=Se ha realizado una observacion con descripcion "{0}" a su actividad ID: {1}.
email.newobservation.url=Para revisarla, puede acceder a: {0}/activity-tracking/{1}/edit
email.newobservation.firm= Software Evolutivo - Registro de Actividades
```

Anexo N2. Variables de los textos usados en el template de Nueva Observación

Anexo O. Template del envío de correo de Proyecto listo para procesar

En la Figura O1 se muestra la estructura en HTML del correo de Proyecto listo para procesar, en este se declaran los estilos y elementos con variables que deberán ser declaradas mediante un archivo de propiedades.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org" th:lang="${#locale.language}" lang="en">
  <head>
    <title th:text="#{email.projectready.title(${projectName})}"></title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  </head>
  <body style="margin: 0; padding: 0">
    <p style="font-size: 15px; margin: 0 0 30px 0; font-family: Arial, sans-serif" th:text="#{email.projectready.dear}"></p>
    <p style="font-size: 15px; font-family: Arial, sans-serif" th:text="#{email.projectready.asunt(${projectName}, ${customerName})}"></p>
    <p style="font-size: 15px; font-family: Arial, sans-serif">--</p>
    <p style="font-size: 15px; font-family: Arial, sans-serif" th:text="#{email.projectready.projectleader(${projectLeader})}"></p>
    <p style="font-size: 15px; font-family: Arial, sans-serif" th:text="#{email.projectready.firm}"></p>
  </body>
</html>
```

Anexo O1. Template del correo de Proyecto listo para procesar

En la Figura O2 se presentan las variables que reemplazarán en el template junto a los atributos que se envíen desde el método de envío de correos.

```
# Project ready
email.projectready.title=Un proyecto esta listo para ser procesado.
email.projectready.dear=Estimado, Gerente de Operaciones.
email.projectready.asunt=El proyecto {0} del cliente {1} tiene todas sus actividades correctas.
email.projectready.projectleader=Atentamente, {0}, Lider de Proyecto.
email.projectready.firm= Software Evolutivo - Registro de Actividades
```

Anexo O2. Variables de los textos usados en el template de Proyecto listo para procesar

Anexo P. Generación de reportes para Gerente de Operaciones en formato PDF

En la Figura P1 se muestra el reporte en formato PDF, con ciertas mejoras estéticas respecto a la versión HTML que se previsualiza en la aplicación web.

| Software Evolutivo | |
|------------------------|----------|
| REPORTE MENSUAL | |
| December - 2023 | |
| Petrolera | |
| | HORAS |
| Portal | 5 |
| Joshua Martinez | 5 |
| Servicios Web | 4 |
| Joshua Martinez | 4 |
| TOTAL | 9 |

Creamos valor desde el momento cero.

Reporte generado en: Wednesday 27 December 2023

Figura P1. Reporte para Gerente de Operaciones en formato PDF

Anexo Q. Generación de reportes para Desarrolladores en formato PDF

En la Figura Q1 se muestra el reporte en formato PDF, con ciertas mejoras estéticas respecto a la versión HTML que se previsualiza en la aplicación web.

Software Evolutivo

REPORTE MENSUAL

December - 2023
Joshua Martinez

| | HORAS |
|-----------------------------|-----------|
| Farmacias Nueva Vida | 4 |
| RRHH Admin | 4 |
| Petrolera | 9 |
| Portal | 5 |
| Servicios Web | 4 |
| TOTAL | 13 |

Reporte generado en: Wednesday 27 December 2023

Creemos valor desde el momento cero.

Figura Q1. Reporte para Desarrolladores en formato PDF



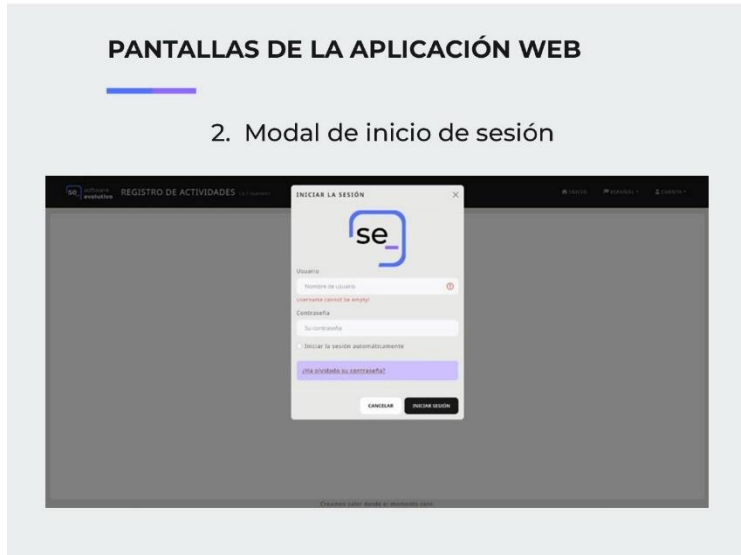
Figura R1. Carátula del manual de usuario



Figura R2. Roles de usuario de la aplicación web



Figura R3. Pantalla de bienvenida a la aplicación web



El usuario debe ingresar su usuario y contraseña:

- Si las credenciales no coinciden o no existen se mostrará un mensaje de error.
- Si las credenciales son correctas continuará a la pantalla de bienvenida con el usuario conectado.

Figura R4. Modal de inicio de sesión para usuarios



- Se tiene una vista general de todos los usuarios de la aplicación web.
- Se tienen disponibles opciones básicas para la gestión de los usuarios tales como creación, edición y eliminación de usuarios.

Figura R5. Pantalla de gestión de usuarios para Administradores



- Se tiene la opción de desactivar o retirar el acceso a la aplicación a un usuario, esta medida de seguridad causa que el usuario en cuestión no pueda acceder a la aplicación.

Figura R6. Opción de activación de usuarios para Administradores



- Al pulsar en crear o editar un usuario se tiene este formulario donde se deberán ingresar los datos personales del usuario, se puede Volver a la pantalla anterior o Guardar la información.

Figura R7. Creación o edición de usuarios para Administradores



- Al pulsar en eliminar cualquier registro de usuario o de cualquier otra entidad de la aplicación se tiene este modal donde se debe confirmar la intención de eliminar el registro.

Figura R8. Modal para confirmación de eliminación de un registro



- Todo usuario tendrá acceso a esta pantalla por medio del menú de CUENTA. Aquí se puede editar la información necesaria y Guardar.

Figura R9. Pantalla para actualización de datos personales para usuarios



- Todo usuario tendrá acceso a esta pantalla por medio del menú de CUENTA. Aquí se puede editar la contraseña de acceso a la aplicación y Guardar.

Figura R10. Pantalla para actualización de contraseña para usuarios



- En esta pantalla se puede visualizar la información más relevante de los clientes de la empresa, contando con las opciones de creación, edición, visualización de detalle y eliminación.

Figura R11. Pantalla de gestión de Clientes para Gerente de Operaciones



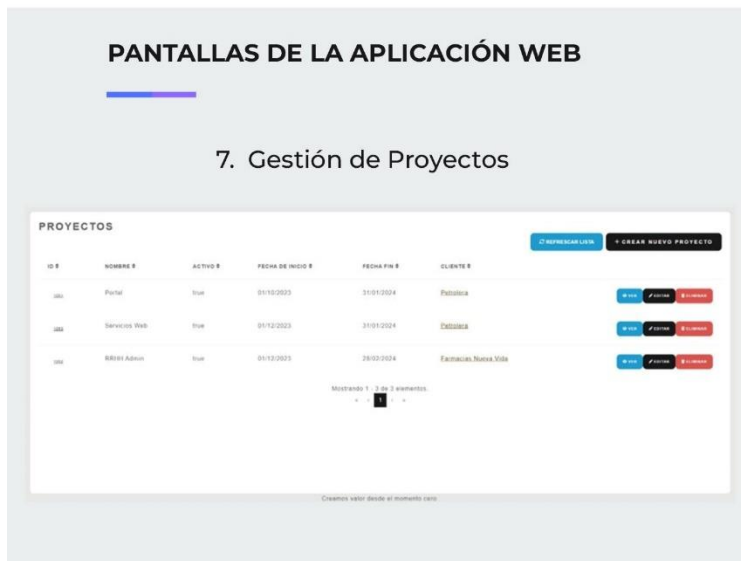
- Al pulsar en Crear o Editar nos encontramos con esta pantalla, se cuenta con un formulario para completar la información más relevante de un Cliente de la empresa, se puede Volver o Guardar la información.

Figura R12. Pantalla para creación o edición de Clientes



- Al pulsar en Ver nos encontramos con esta pantalla, esta pantalla cuenta con la información a detalle de un Cliente de la empresa, se puede Volver cuando se desee.

Figura R13. Pantalla de visualización de detalle de Clientes



- En esta pantalla se puede visualizar la información más relevante de los Proyectos de la empresa, contando con las opciones de creación, edición, visualización de detalle y eliminación.

Figura R14. Pantalla de gestión de Proyectos para Líder de Proyectos



- Al pulsar en Crear o Editar nos encontramos con esta pantalla, se cuenta con un formulario para completar la información más relevante de un Proyecto de la empresa, se puede Volver o Guardar la información y Marcar el Proyecto como Listo para procesar.

Figura R15. Pantalla para creación o edición de Proyectos



- Al pulsar en Ver nos encontramos con esta pantalla, esta pantalla cuenta con la información a detalle de un Proyecto de la empresa, se puede Volver o Editar la información cuando se desee.

Figura R16. Pantalla de visualización de detalle de Proyectos



- En esta pantalla se muestra únicamente las Actividades de cada Desarrollador.
- Se encuentran en la parte superior los campos para filtrar la información.
- Para cada actividad existen 3 botones de acción los cuales permiten entrar al detalle de la Actividad, editar la información de la Actividad o eliminar por completo la Actividad.
- En la columna de Proyecto, se muestra el nombre del Proyecto al que pertenece la Actividad con un hipervínculo que nos lleva al detalle del Proyecto.

Figura R17. Pantalla de gestión de Actividades para Desarrollador



- Al pulsar en Crear o Editar nos encontramos con esta pantalla, se cuenta con un formulario para completar la información más relevante de una Actividad junto al Proyecto al que pertenece, se puede Volver o Guardar la información.

Figura R18. Pantalla para creación o edición de Actividades



- Al pulsar en Ver nos encontramos con esta pantalla, esta pantalla cuenta con la información a detalle de un Proyecto de la empresa, se puede Volver o Editar la información cuando se desee.

Figura R19. Pantalla de visualización de detalle de Actividades



- En esta pantalla, se cargarán únicamente las actividades de los proyectos de los cuales este usuario es Líder de Proyecto.
- Se cuenta con opciones de filtrado para una mejor experiencia de uso.
- Al lado derecho de cada Actividad se encuentra un botón para ver el detalle de la actividad y a su vez poder registrar una o más Observaciones a la Actividad.

Figura R20. Pantalla seguimiento de Actividades para Líder de Proyecto



- En el lado izquierdo se encuentra el detalle de la Actividad y al lado derecho se encuentra a disposición la lista de las Observaciones registradas para la Actividad y la opción de gestionar las Observaciones de la misma.

Figura R21. Pantalla de detalle del seguimiento de Actividades para Líder de Proyecto



- Se muestra el modal que se abre al entrar en la opción de "Crear Nueva Observación" en la que se puede registrar una nueva Observación a una Actividad o editar una existente.
- Cuando se cree una, el Desarrollador dueño de la Actividad en cuestión será notificado vía correo electrónico, para que pueda resolver esta Observación.

Figura R22. Pantalla de creación o edición de una Observación para una Actividad



- Observaciones de los Proyectos de los que cada Líder de Proyecto está a cargo.
- En la columna de Actividad tendrán el url que los llevará al Detalle de Seguimiento de Actividades para validar que si la información de la Actividad se corrigió o no.

Figura R23. Pantalla de visualización general de las Observaciones para Líder de Proyecto



- En esta pantalla, se encuentra la visualización general de las Actividades que un Desarrollador tenga con Observaciones.
- Se puede acceder a esta información y validar que las Observaciones que se le fueron notificadas ya se han resuelto o aún las tiene que resolver.
- El botón de Editar le permitirá acceder a la Pantalla 9.3

Figura R24. Pantalla de seguimiento de Actividades para Desarrollador



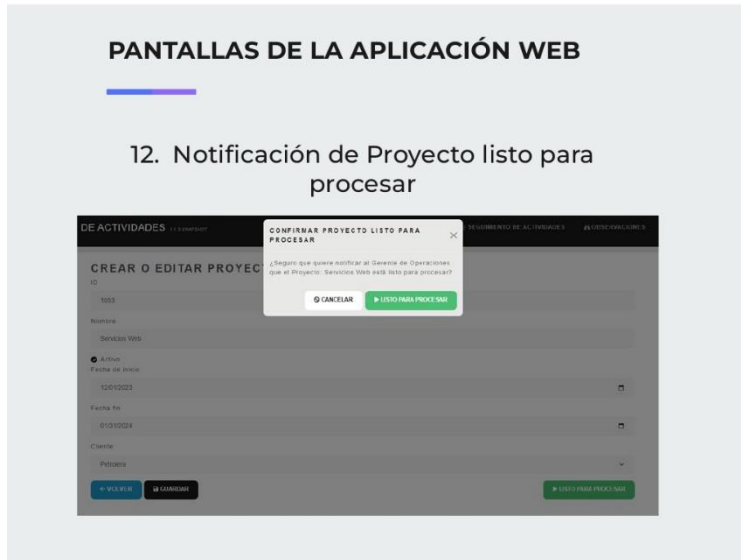
- En esta pantalla el Desarrollador puede corregir la información de la Actividad y marcar como resuelta una Observación.
- Se visualizan todas las Observaciones de esa Actividad sean nuevas o resueltas.

Figura R25. Pantalla de detalle del seguimiento de Actividades para Desarrollador



- Modal que se tiene para confirmar que se ha resuelto la Observación.

Figura R26. Modal de confirmación de resolución de una Observación



- En la Pantalla 7.1 se tiene que pulsar el botón de Listo para Procesar, se muestra el modal en el que se solicita la confirmación del Líder de Proyecto para notificar al Gerente de Operaciones sobre este Proyecto.
- Se notificará por correo electrónico al Gerente de Operaciones de que el Proyecto en cuestión está listo para procesar.

Figura R27. Modal de confirmación de Proyecto listo para procesar



- Mediante esta pantalla se pueden generar reportes filtrados por mes, año y Cliente.
- Al enviar la solicitud en la misma interfaz se mostrará una previsualización en formato HTML del reporte, y se habilitará la opción de descargarlo como PDF.

Figura R28. Pantalla de generación de reportes de Clientes para Gerente de Operaciones



- Visualización del reporte al descargarlo como PDF.

Figura R29. Reporte de Clientes en formato PDF



- Mediante esta pantalla se pueden generar reportes de sus actividades filtrados por mes y año.
- Al enviar la solicitud en la misma interfaz se mostrará una previsualización en formato HTML del reporte, y se habilitará la opción de descargarlo como PDF.

Figura R30. Pantalla de generación de reportes de Desarrollador

PANTALLAS DE LA APLICACIÓN WEB

14.1 Reporte de Desarrolladores en formato PDF



| REPORTE MENSUAL | |
|-----------------------------|-----------|
| December - 2023 | |
| Joshua Martinez | |
| ----- | |
| Farmacias Nueva Vida | 4 |
| Hospitales | 4 |
| Patriceria | 9 |
| Postal | 5 |
| Servicios Web | 4 |
| TOTAL | 11 |

Reporte generado en: Wednesday, 27 December 2023

- Visualización del reporte al descargarlo como PDF.

Figura R31. Reporte de Desarrollador en formato PDF