

**UNIVERSIDAD POLITÉCNICA SALESIANA**  
**SEDE QUITO**

**CARRERA:**  
**INGENIERÍA DE SISTEMAS**

Trabajo de titulación previo a la obtención del título de:  
Ingeniero de Sistemas

**TEMA:**  
**REGISTRO DE CAMBIOS DE INFORMACIÓN Y FLUJO DE ASIGNACIÓN EN  
EL CONTROL Y VALIDACIÓN DE LA INFORMACIÓN CARGADA EN  
SOFTWARE ESPECIALIZADOS PARA EL CONTROL DE OPERACIONES DE  
EXPLORACIÓN.**

**AUTORES:**  
**JOSÉ RAMÓN BASTIDAS SÁNCHEZ**  
**JIMMY ALEXANDER PUENTE QUINGA**

**TUTOR:**  
**DANIEL GIOVANNY DÍAZ ORTIZ**

Quito, febrero 2020

## CESIÓN DE DERECHOS DE AUTOR

Nosotros, **José Ramón Bastidas Sánchez**, con documento de identificación N° 1725201907, y **Jimmy Alexander Puente Quinga**, con documento de identificación N° 1719201160, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: “**REGISTRO DE CAMBIOS DE INFORMACIÓN Y FLUJO DE ASIGNACIÓN EN EL CONTROL Y VALIDACIÓN DE LA INFORMACIÓN CARGADA EN SOFTWARE ESPECIALIZADOS PARA EL CONTROL DE OPERACIONES DE EXPLORACIÓN**”, mismo que ha sido desarrollado para optar por el título de INGENIEROS DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada.

En concordancia, suscribimos este documento en el momento que hacemos la entrega del trabajo final en digital a la Biblioteca de la Universidad Politécnica Salesiana.

**Quito, febrero 2020**



JIMMY ALEXANDER  
PUENTE QUINGA  
CI: 1719201160



JOSÉ RAMÓN  
BASTIDAS SÁNCHEZ  
CI: 1725201907

## DECLARATORIA DE COAUTORÍA DEL TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el proyecto técnico, con el tema: **“REGISTRO DE CAMBIOS DE INFORMACIÓN Y FLUJO DE ASIGNACIÓN EN EL CONTROL Y VALIDACIÓN DE LA INFORMACIÓN CARGADA EN SOFTWARE ESPECIALIZADOS PARA EL CONTROL DE OPERACIONES DE EXPLORACIÓN”**, realizado por: **José Ramón Bastidas Sánchez y Jimmy Alexander Puente Quinga**, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

**Quito, febrero 2020**



.....  
**DANIEL GIOVANNY DÍAZ ORTIZ**

**CI: 1716975501**

## **DEDICATORIA**

El presente trabajo está dedicado a las personas que a lo largo de la carrera universitaria han aportado con conocimiento, experiencias, a quienes han ayudado a recorrer el camino del conocimiento y a quienes marcaron con sus habilidades el camino del aprendizaje.

Está dedicado a la persona que supo aportar paciencia, ayuda y tiempo, a quien a pesar de las circunstancias supo brindar palabras de apoyo en los momentos más difíciles, a quien compartió su tiempo en un instante y forma inesperado. A ti, Ivonne, muchas gracias.

El presente trabajo representa la culminación de muchas partes únicas que conforman una totalidad de conocimiento, un conocimiento que representa experiencias de vida, de las cuales queda una huella imborrable en el tiempo.

Jimmy Alexander Puente Quinga

## **AGRADECIMIENTO**

Agradezco en primer lugar A DIOS, por bendecirme siempre en mi camino, por guiarme, por ser mi apoyo y fortaleza en aquellos momentos de dificultad y de debilidad, y porque inspiro a mi espíritu para la culminación de mis estudios. A mis padres Ramón Bastidas y Marcia Sánchez, quienes me dieron la vida, educación, apoyo y consejos, pues son ellos quienes me motivaron constantemente para alcanzar todos mis sueños.

Adicionalmente, quiero dar un reconocimiento muy especial a todos mis docentes, por haber compartido sus conocimientos a lo largo de la preparación de mi carrera profesional, de manera especial, al Ing. Daniel Díaz Msc, tutor de nuestro proyecto de TITULACIÓN quien siempre nos ha guiado con paciencia, y rectitud todo este tiempo.

Finalmente quiero expresar mi más grande y sincero agradecimiento a todas las autoridades y personal que integran la AGENCIA DE CONTROL DE HIDROCARBUROS por su valioso aporte en nuestro proyecto de grado.

José Ramón Bastidas Sánchez

Agradezco a mi familia que ha sido un pilar fundamental a lo largo de mis estudios, apoyándome siempre y siendo la base sobre la cual he fundamentado mis valores e identidad.

De la misma manera agradezco a las personas que han sido mis docentes, gracias a quienes he forjado mi conocimiento y experiencia, de manera especial agradezco al ingeniero Daniel Díaz, quien ha sido el tutor del presente proyecto de titulación siendo una guía ante todas las inquietudes que le hemos presentado.

Jimmy Alexander Puente Quinga

# ÍNDICE

## Contenido

INTRODUCCIÓN.....	1
Antecedentes.....	1
Justificación.....	1
Objetivos.....	4
Objetivo General.....	4
Objetivos Específicos.....	4
CAPÍTULO 1.....	6
1.1. Marco Referencial o Institucional.....	6
1.2. Flujo de procesos institucionales.....	7
1.3. Marco teórico.....	9
1.3.1. Angular.....	9
1.3.2. Spring.....	10
1.4. Metodología de desarrollo.....	10
1.4.1. Análisis de la Metodología.....	10
1.4.2. Metodología XP.....	11
1.4.3. Fases de la metodología.....	12
1.4.3.1. Primera fase: Planificación del proyecto.....	12
1.4.3.2. Segunda fase: Diseño.....	13
1.4.3.3. Tercera fase: Codificación.....	13
1.4.3.4. Cuarta fase: Pruebas.....	13
CAPÍTULO 2.....	15
2.1. Requerimientos de aplicación.....	15
2.1.1. Requisitos funcionales.....	15
2.1.1.1. Seguimiento de novedades detectadas.....	15
2.1.1.2. Crear portafolio.....	16
2.1.1.2.1. Creación de portafolio de trabajos en pozos.....	16
2.1.1.2.2. Buscar portafolio de trabajos en pozos.....	18
2.1.1.2.3. Editar o anular portafolio de trabajos en pozos.....	18
2.1.1.3. Registro reporte diario trabajos en pozo.....	20
2.1.1.4. Registro documentos habilitantes operadora.....	21

2.1.1.5.	Informe de resultados de trabajo .....	23
2.1.1.6.	Verificación de fechas de entrega de información .....	24
2.1.1.7.	Verificación de producción .....	25
2.1.1.8.	Verificación de inyección / reinyección .....	26
2.1.2.	Requisitos no funcionales.....	27
2.1.2.1.	Seguridad de aplicativo .....	27
2.1.2.2.	Interfaz de aplicativo .....	28
2.1.2.3.	Rendimiento de la aplicación .....	28
2.1.2.4.	Requisitos operacionales del aplicativo.....	29
2.2.	Artefactos – Diagramas .....	30
2.2.1.	Seguimiento de novedades.....	30
2.2.1.1.	Detalle de caso de uso .....	30
2.2.1.2.	Secuencia de actividades entre usuario y aplicativo.....	30
2.2.1.3.	Diagrama de caso de uso UML .....	31
2.2.1.4.	Diagrama de secuencia UML .....	31
2.2.2.	Creación de portafolio de trabajos en pozos .....	32
2.2.2.1.	Detalle de caso de uso .....	32
2.2.2.2.	Secuencia de actividades entre usuario y aplicativo.....	33
2.2.2.3.	Diagrama de caso de uso UML .....	33
2.2.2.4.	Diagrama de secuencia UML .....	34
2.2.3.	Buscar portafolio de trabajos en pozos .....	35
2.2.3.1.	Detalle de caso de uso .....	35
2.2.3.2.	Secuencia de actividades entre usuario y aplicativo.....	35
2.2.3.3.	Diagrama de caso de uso UML .....	36
2.2.3.4.	Diagrama de secuencia UML .....	36
2.2.4.	Editar o anular portafolio de trabajos en pozos.....	37
2.2.4.1.	Detalle de caso de uso .....	37
2.2.4.2.	Secuencia de actividades entre usuario y aplicativo.....	38
2.2.4.3.	Diagrama de caso de uso UML .....	38
2.2.4.4.	Diagrama de secuencia UML .....	39
2.2.5.	Informe de resultados de trabajo .....	40
2.2.5.1.	Detalle de caso de uso .....	40

2.2.5.2.	Secuencia de actividades entre usuario y aplicativo.....	41
2.2.5.3.	Diagrama de caso de uso UML .....	41
2.2.5.4.	Diagrama de secuencia UML .....	42
CAPÍTULO 3 .....		43
3.1.	Descripción de Clases (con código relevante).....	43
3.1.1.	Explicación sobre clases y código de aplicativo web .....	43
3.1.1.1.	Encapsulación de métodos Service .....	43
3.1.1.2.	Ejemplo de una definición de base de datos.....	43
3.1.1.3.	Listar producción.....	44
3.1.1.4.	Guardar datos de producción.....	45
3.1.1.5.	Actualización de datos de Verificación de Producción .....	46
3.1.1.6.	Replicar datos de producción .....	46
3.1.1.7.	Encapsulación métodos DAO.....	48
3.1.1.8.	Generación de datos.....	48
3.1.1.9.	Método Data Access Object para búsqueda de portafolio.....	49
3.1.1.10.	Parámetros de conexión con base de datos desde Spring.....	50
3.1.1.11.	Módulo Controlador .....	50
3.2.	Implementación .....	51
3.2.1.	Características de Servidor físico.....	51
3.2.2.	Software utilizado en la aplicación .....	52
3.3.	Base de datos .....	52
3.3.1.	Patrones para especificaciones de base de datos .....	52
3.3.1.1.	Definiciones para tablas de base de datos .....	53
3.3.1.2.	Campos de base de datos .....	54
3.3.1.3.	Claves Primarias y Llaves Foráneas.....	54
3.3.2.	Diagrama Conceptual de base de datos.....	55
3.3.3.	Diagrama físico de base de datos .....	55
3.4.	Diagramas de aplicación.....	55
3.4.1.	Diagrama de despliegue .....	56
3.4.2.	Diagrama de navegación .....	56
3.5.	Plan de Pruebas .....	57
3.5.1.	Pruebas Unitarias.....	57

3.5.2. Pruebas de Carga y Stress .....	63
3.5.2.1. Escenario de 25 peticiones .....	63
3.5.2.2. Escenario de 50 peticiones .....	66
3.5.2.3. Escenario de 75 peticiones .....	68
3.5.2.4. Análisis de Resultados.....	70
3.5.2.5. Pruebas de stress.....	71
3.5.2.5.1. Escenario 300 peticiones .....	71
3.5.2.5.2. Escenario 600 peticiones .....	74
3.5.2.5.3. Análisis de resultados stress .....	76
CONCLUSIONES.....	77
RECOMENDACIONES .....	79
GLOSARIO DE TERMINOS .....	81
LISTA DE REFERENCIAS.....	82
Bibliografía.....	82
ANEXOS.....	83

## ÍNDICE DE TABLAS

Tabla 1. Propiedades metodologías de desarrollo .....	11
Tabla 2. Requisito funcional Seguimiento de novedades.....	15
Tabla 3. Requisito funcional Creación de portafolio .....	16
Tabla 4. Requisito funcional Buscar portafolio.....	18
Tabla 5. Requisito funcional Editar o Anular portafolio.....	19
Tabla 6. Requisito funcional Registro de reporte diario .....	20
Tabla 7. Requisito funcional Registro documentos habilitantes .....	21
Tabla 8. Requisito funcional Informe resultados de trabajo.....	23
Tabla 9. Requisito funcional Verificación de fechas .....	24
Tabla 10. Requisito funcional Verificación de producción.....	25
Tabla 11. Requisito funcional Verificación de inyección / reinyección .....	26
Tabla 12. Requisito no funcional Seguridad de aplicativo.....	27
Tabla 13. Requisito no funcional Interfaz de aplicativo .....	28
Tabla 14. Requisito no funcional Rendimiento de aplicativo .....	28
Tabla 15. Requisito no funcional Propiedades operacionales de aplicativo .....	29
Tabla 16. Caso de Uso Seguimiento de Novedades.....	30
Tabla 17. Secuencia de Actividades Seguimiento de Novedades .....	30
Tabla 18. Caso de Uso Creación de portafolio de trabajos en pozos .....	32
Tabla 19. Secuencia de Actividades Creación de portafolio de trabajos en pozos .....	33
Tabla 20. Caso de Uso Buscar portafolio de trabajos en pozos .....	35
Tabla 21. Secuencia de Actividades Buscar portafolio de trabajos en pozos.....	35
Tabla 22. Caso de Uso Editar o anular portafolio de trabajos en pozos.....	37
Tabla 23. Secuencia de Actividades Editar o anular portafolio de trabajos en pozos.....	38
Tabla 24. Caso de Uso Informe de resultados de trabajo.....	40
Tabla 25. Secuencia de Actividades Informe de resultados de trabajo .....	41
Tabla 26. Características de servidor .....	51
Tabla 27. Software utilizado .....	52
Tabla 28. Objeto y atributos Informe Operadora .....	58
Tabla 29. Resultados obtenidos de pruebas unitarias de Informe Operadora .....	58
Tabla 30. Objeto y atributos Conclusiones y Recomendaciones .....	59
Tabla 31. Resultados obtenidos de pruebas unitarias de Conclusiones y Recomendaciones.....	59
Tabla 32. Objeto y atributos Seguimiento de novedad .....	59
Tabla 33. Resultados obtenidos de pruebas unitarias de Seguimiento de novedad.....	60
Tabla 34. Objeto y atributos Verificación de producción .....	60
Tabla 35. Resultados obtenidos de pruebas unitarias de Verificación de producción .....	61
Tabla 36. Objeto y atributos Verificación de Reinyector.....	61
Tabla 37. Resultados obtenidos de pruebas unitarias de Verificación de Reinyector.....	62
Tabla 38. Objeto y atributos Verificar Fechas .....	62
Tabla 39. Resultados obtenidos de pruebas unitarias de Verificar Fechas.....	62
Tabla 40. Prueba de carga para Buscar portafolio 25 peticiones .....	64
Tabla 41. Prueba de carga para Seguimiento Novedad 25 peticiones.....	64
Tabla 42. Prueba de carga para Verificación Producción 25 peticiones .....	64
Tabla 43. Prueba de carga para Verificación Inyector 25 peticiones .....	65

Tabla 44. Prueba de carga para Verificación Fechas 25 peticiones .....	65
Tabla 45. Prueba de carga para Buscar portafolio 50 peticiones .....	66
Tabla 46. Prueba de carga para Seguimiento Novedad 50 peticiones.....	66
Tabla 47. Prueba de carga para Verificación Producción 50 peticiones .....	67
Tabla 48. Prueba de carga para Verificación Inyector 50 peticiones .....	67
Tabla 49. Prueba de carga para Verificación Fechas 50 peticiones .....	67
Tabla 50. Prueba de carga para Buscar portafolio 75 peticiones .....	68
Tabla 51. Prueba de carga para Seguimiento Novedad 75 peticiones.....	68
Tabla 52. Prueba de carga para Verificación Producción 75 peticiones .....	69
Tabla 53. Prueba de carga para Verificación Inyector 75 peticiones .....	69
Tabla 54. Prueba de carga para Verificación Fechas 75 peticiones .....	70
Tabla 55. Comparativa de peticiones de carga.....	70
Tabla 56. Prueba de stress Buscar Portafolio 300 peticiones.....	71
Tabla 57. Prueba de stress Seguimiento Novedad 300 peticiones .....	72
Tabla 58. Prueba de stress Verificación Producción 300 peticiones.....	72
Tabla 59. Prueba de stress Verificar Inyector 300 peticiones .....	73
Tabla 60. Prueba de stress Verificar Fechas 300 peticiones .....	73
Tabla 61. Prueba de stress Buscar Portafolio 600 peticiones.....	74
Tabla 62. Prueba de stress Seguimiento Novedad 600 peticiones .....	74
Tabla 63. Prueba de stress Verificación Producción 600 peticiones.....	74
Tabla 64. Prueba de stress Verificar Inyector 600 peticiones .....	75
Tabla 65. Prueba de stress Verificar Fechas 600 peticiones .....	75
Tabla 66. Comparativa de peticiones de stress.....	76

## ÍNDICE DE FIGURAS

Figura 1. Mapa de procesos Institucional ARCH.....	7
Figura 2. Esquema de procesos ARCH.....	8
Figura 3. Caso de uso Seguimiento de novedades .....	31
Figura 4. Diagrama de secuencia de Seguimiento de novedades.....	32
Figura 5. Caso de uso Creación de portafolio de trabajos en pozos.....	34
Figura 6. Diagrama de secuencia de Creación de portafolio de trabajos en pozos .....	34
Figura 7. Caso de uso Creación de portafolio de trabajos en pozos.....	36
Figura 8. Diagrama de secuencia de Buscar portafolio de trabajos en pozos .....	37
Figura 9. Caso de uso Creación de portafolio de trabajos en pozos.....	39
Figura 10. Diagrama de secuencia de Editar o anular portafolio de trabajos en pozos.....	40
Figura 11. Caso de uso Creación de portafolio de trabajos en pozos.....	42
Figura 12. Diagrama de secuencia de Informe de resultados de trabajo .....	42
Figura 13. Método Service de Informe Operadora.....	43
Figura 14. Definición de base de datos .....	44
Figura 15. Funcionalidad para listar producción.....	45
Figura 16. Funcionalidad para guardar datos de producción .....	46
Figura 17. Método de actualización de datos de producción .....	46
Figura 18. Funcionalidad para replicación de datos de producción .....	48
Figura 19. Funcionalidad para encapsulación de métodos DAO .....	48
Figura 20. Funcionalidad para generación de datos .....	49
Figura 21. Funcionalidad para búsqueda de portafolio .....	49
Figura 22. Funcionalidad para conexión a repositorio de datos.....	50
Figura 23. Funcionalidad para replicación de datos de producción .....	51
Figura 24. Diagrama de despliegue.....	56
Figura 25. Diagrama navegacional.....	57

## ÍNDICE DE ANEXOS

Diagrama Conceptual.....	83
Diagrama Físico.....	83

Para revisar los anexos de este trabajo, por favor diríjase al CD.

## **Resumen**

En el presente trabajo se expone una solución a la problemática presentada dentro de la Agencia de Regulación y Control Hidrocarburífero (ARCH), debido a la falta de soluciones tecnológicas que permitan guardar y modificar los datos sobre la producción petrolera. Actualmente la ARCH, gestiona sus datos de producción de manera manual, tanto para ingreso de datos como para modificación de los mismos, y esta tarea al ser ejecutada de manera enteramente manual dificulta y ralentiza el flujo de las operaciones que la ARCH realiza normalmente.

Dados estos antecedentes se ha planteado crear una solución integral para resolver las demandas de la ARCH, con tecnologías que facilitarán la tarea de administrar la información producida por parte de todas las operadoras que reporten directamente a la ARCH, a través de un aplicativo para gestión de información.

## **Abstract**

This paper presents a solution to the problems presented within the Hydrocarbons Regulation and Control Agency, due to the lack of technological solutions to save and modify the data on oil production. Currently, the Hydrocarbons Regulation and Control Agency manages its production data manually, both for data entry and for their modification, and this task, when carried out entirely manually, hinders and slows down the flow of operations that the Agency for Hydrocarbon Regulation and Control normally performs.

Given these antecedents, it has been proposed to create an integral solution to solve the demands of the Hydrocarbons Regulation and Control Agency, with technologies that facilitate the task of managing the information produced by all the operators that report directly to the Regulation and Control Agency Hydrocarbons, through an application for information management.

# INTRODUCCIÓN

## **Antecedentes**

Al hablar de sistemas de gestión de información podemos considerar distintas clasificaciones sobre las cuales se agrupan los sistemas de gestión de información y en base a los parámetros de información que manejan podemos encontrar: sistemas de procesamiento de transacciones, sistemas de control de procesos de negocio, sistemas de colaboración empresarial, sistemas de información de gestión, sistemas de apoyo a la toma de decisiones y sistemas de información ejecutiva. (KYOCERA Document Solutions, 2017).

Entre las principales características que podemos destacar de los sistemas de gestión de información se encuentran los sistemas de procesamiento de transacciones, los cuales funcionan a nivel operacional dentro de una organización llevando y registrando transacciones necesarias para el funcionamiento de la empresa. (KYOCERA Document Solutions, 2017).

## **Justificación**

Actualmente la ARCH no cuenta con un sistema que permita registrar la producción por períodos diarios y mensuales de petróleo, así como ciertas observaciones pertinentes a la misma, pues todo este control se lo lleva a cabo de manera manual.

En un entorno en el cual es de vital importancia contar con datos precisos en tiempo real, que son necesarios para formar criterios de decisión basados en información y dada la cantidad de variables que pueden resultar de todos los procesos que maneja la ARCH se ha considerado la necesidad de contar con herramientas que faciliten la administración de la

información, enfocándose en tareas sensibles tales como la asignación del flujo de control de los datos y la modificación de la información sobre los datos de producción. Sin embargo, el hecho de no contar con una plataforma centralizada que sea capaz de recolectar información, además de tener múltiples sistemas que están constantemente presentando información de manera independiente, hace que la información no llegue adecuadamente y de forma oportuna, lo cual dificulta el manejo y la administración de los sistemas de producción de la agencia. La empresa actualmente cuenta con varios programas que reportan los detalles de los pozos petroleros (Open Wells, Petrel, etc.), pero estos programas funcionan de manera independiente y la información se encuentra dispersa, ya que estas aplicaciones son entregadas por los operadores de los pozos de las distintas petroleras. Adicionalmente la información de otras operadoras se las recibe mediante correos electrónicos, los cuales muchas veces no son recibidos, o no son leídos, aparte de que se los envía en diferentes formatos (XML, XSL, TXT). Cuando un operador envía cambios de ciertos datos que se enviaron, no existe ningún sistema que permita gestionar dichos cambios, lo cual genera un problema de integridad y consistencia de la información al momento de una auditoría. Toda la información que se maneja hasta este momento es gestionada de manera manual y sin ningún tipo de control o seguridad, lo que compromete gravemente dicha información, incluso se puede incurrir en una omisión de la información ya que los cambios que se realizan están sujetos a la gestión de las personas encargadas. Dada la sensibilidad de la información que se maneja es importante contar con un sistema que sea capaz de registrar y controlar los cambios que son realizados sobre la información de producción de pozos petroleros y demás datos relevantes a la ARCH.

En base a lo mencionado, se ha propuesto diseñar una aplicación que permita registrar los eventos y las modificaciones de los informes presentados. Al contar con un sistema capaz de registrar los cambios es posible brindar seguridad, confidencialidad, e integridad sobre la información, asegurando la calidad óptima de la información, además de que los datos de producción representan una parte central sobre la cual se desenvuelve las operaciones de la ARCH, es por este motivo que es necesaria una manera de controlar y registrar estos cambios sobre la información de manera que se pueda actuar en base a los datos registrados realizando las gestiones pertinentes y se pueda dar el uso necesario para el cual se guardaron los registros de cambios de información, actuando a manera de bitácora con la cual se podrían realizar auditorías cuando las autoridades lo consideren pertinente. Entre los beneficios que aportará el desarrollo de la aplicación se encuentra el manejo de la información de una manera centralizada con la finalidad de controlar y gestionar cambios, al tener un aplicativo de este tipo se puede brindar mayor seguridad al controlar y vigilar las acciones que se realizan sobre los datos de producción petrolera, se maneja un estándar de calidad en la presentación de datos al contar con actualizaciones verificadas dentro del sistema y que están autorizadas por parte de las personas que se encuentra encargadas de realizar las gestiones de cambio o actualización sobre los datos. Registrar el ingreso de nueva información, de eventos, es una tarea de vital importancia para la gestión de la ARCH, dado que la información que se presenta como resultado de su operación se basa en los datos recolectados por las operadoras y es necesario tener una correcta administración de dichos datos a fin de presentar resultados congruentes y útiles para la organización. La aplicación ayudará a que se manejen de forma eficiente los recursos de la empresa, teniendo un control del ingreso de los datos por parte de las operadoras, a través de un aplicativo web que registre las novedades que sucedan con respecto al manejo de dicha información. Un manejo adecuado y oportuno de la información

permite que se tomen acciones pertinentes sobre sucesos y al mismo tiempo permite ayudar a gestionar cambios de dirección sobre las eventualidades (que tipo de eventualidades) que se presenten, tomando como objetivo principal la optimización de los procesos (que tipo de procesos) que se llevan actualmente.

## **Objetivos**

### **Objetivo General**

Analizar, diseñar y elaborar un aplicativo que permita registrar los cambios y actualizar la información procedente de la producción diaria y mensual de petróleo, así como los trabajos de reacondicionamiento, y trabajos sin torre por la operadora en las que se encuentren laborando.

### **Objetivos Específicos**

Analizar y determinar los requerimientos de la ARCH para el registro de cambios en información de producción hidrocarburífero, mediante el uso diagramas y herramientas para modelado de procesos.

Diseñar una herramienta de gestión que permita evidenciar las inobservancias o rectificaciones de los datos recibidos en la administración central de la ARCH por parte de las empresas petroleras, las cuales envían datos e información de producción.

Respaldar cambios que se realizaron en información generada por parte de pozos petroleros afiliados a la ARCH mediante el registro y actualización de las operaciones producidas dentro del proceso de validación de la información y que previamente fue entregada a entidades gubernamentales

Desarrollar una aplicación para la ARCH que permita centralizar toda la información correspondiente a la producción periódica de los múltiples pozos petroleros que se encuentran afiliados a la ARCH y que reportan su producción de manera periódica.

Realizar una valoración del funcionamiento de la aplicación desarrollada mediante el uso de herramientas que permitirán medir el desempeño y la funcionalidad de los módulos desarrollados dentro del proyecto técnico.

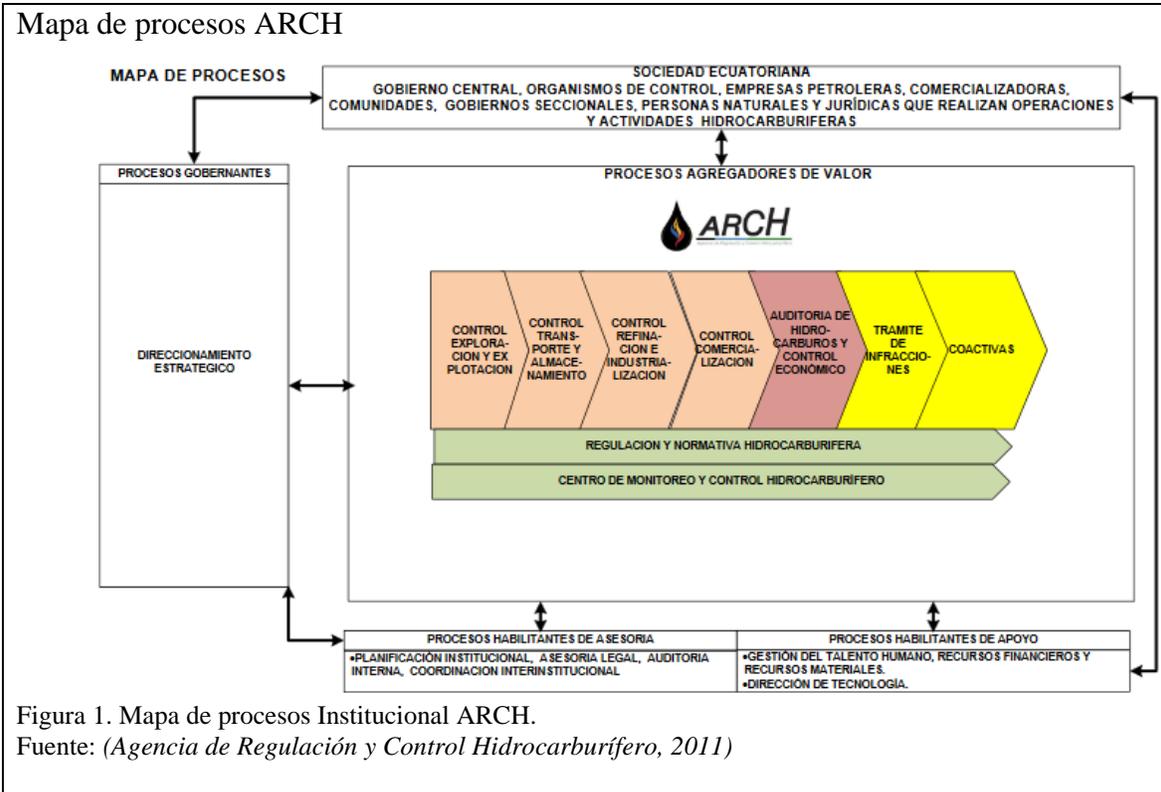
## CAPÍTULO 1

### 1.1. Marco Referencial o Institucional

La Agencia de regulación y control de hidrocarburos, fue creada en la Ley Reformativa a la Ley de Hidrocarburos y a la Ley de Régimen Tributario Interno, publicada en el suplemento del Registro Oficial No. 244 del 27 de julio del 2010 que en su Art. 11 dice: Créase la Agencia de Regulación y Control Hidrocarburífero, ARCH, como un organismo técnico administrativo, encargado de regular, controlar y fiscalizar las actividades técnicas y operacionales en las diferentes fases de la industria hidrocarburífera, que realicen las empresas públicas o privadas, nacionales, extranjeras, empresas mixtas, consorcios, asociaciones u otras formas contractuales y demás personas naturales o jurídicas nacionales o extranjeras que ejecuten actividades hidrocarburíferas en el Ecuador. (Agencia de Regulación y Control Hidrocarburífero, 2011)

La Planificación Estratégica de la Agencia de Regulación y Control Hidrocarburífero (ARCH) está encaminada a la alineación del Nuevo Plan Nacional del Buen Vivir 2014-2017, siguiendo los preceptos y directrices establecidas por SENPLADES, en torno al buen vivir de la población ecuatoriana. (Agencia de Regulación y Control Hidrocarburífero, 2011)

La estructura organizacional de la institución, en la cual se visibilizan los procesos que rigen el cumplimiento de funciones dentro de la institución se presenta a continuación:



## 1.2. Flujo de procesos institucionales

En el siguiente esquema se muestran los procesos institucionales que se registran dentro de la ARCH, se muestra además la manera en cómo interactúan los procesos dentro de las distintas dependencias y unidades de control de la ARCH.

Dentro de las actividades presentadas en la Agencia de Regulación y Control Hidrocarburífero se presenta el procedimiento que describe las actividades internas que los funcionarios de la Agencia de Regulación y Control Hidrocarburífero - ARCH, deben realizar para el control técnico y seguimiento de las operaciones hidrocarburíferas con respecto a

trabajos de pozos en operaciones de producción de crudo y gas natural que se realizan en Ecuador.

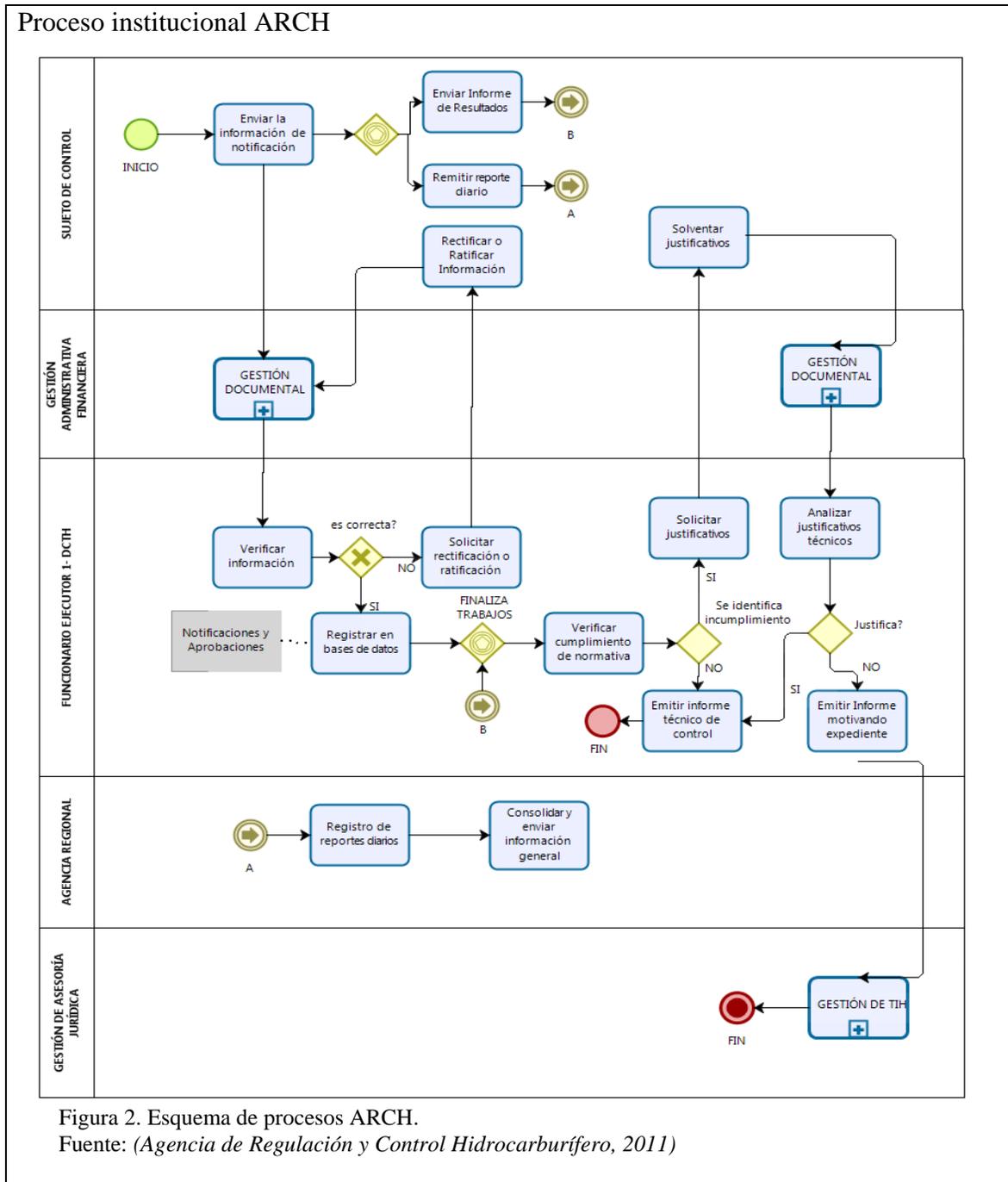


Figura 2. Esquema de procesos ARCH.

Fuente: (Agencia de Regulación y Control Hidrocarburiífero, 2011)

Las actividades descritas en este procedimiento son de uso y cumplimiento de todos los funcionarios encargados de la Gestión de producción, además de involucrados en las actividades de control y fiscalización de las operaciones de producción hidrocarburífera.

La descripción de las actividades contempla desde la entrega de la información técnica para notificación o solicitud de aprobación de trabajos o programas alternos en pozos por parte de los sujetos de control y de funcionarios de la ARCH Regionales, hasta la emisión del Informe técnico para fundamentar la apertura de expedientes administrativos, en el caso de que sea necesario. La gestión de las operaciones de control es aplicada para pozos productores de crudo, de inyección y de reinyección incluyendo completaciones y pruebas iniciales, reacondicionamientos y trabajos sin torre. (Agencia de Regulación y Control Hidrocarburífero, 2011)

### **1.3. Marco teórico**

#### **1.3.1. Angular**

Angular es un framework para desarrollo de aplicaciones web, de código abierto, el cual es mantenido por Google, utiliza componentes, directivas y servicios. Angular y su arquitectura están basados en la orientación a objetos, presenta un esquema de trabajo en clases y permite la utilización del patrón Modelo-Vista-Controlador (MVC). (Freeman, 2018, pág. 20)

Una de las principales características que presenta Angular es la presentación de aplicaciones web en una sola página, es decir que usa la característica SPA (Single-Page Application) mediante la cual se carga todos los datos de la aplicación, de forma asíncrona, en una única página web. La característica SPA permite una mejora en el rendimiento de la aplicación, brinda contenidos más fluidos al usuario, y tiempos de respuesta mucho menores a

tecnologías tradicionales, el funcionamiento de SPA se encuentra optimizado en dispositivos móviles ya que maneja de una manera óptima los ciclos de CPU y memoria en el dispositivo. (Boada & Gómez, 2018, págs. 30-32)

### **1.3.2. Spring**

Spring es un framework de desarrollo para Java el cual fue lanzado a mediados del año 2003 bajo licencia Apache 2.0, a partir de esta versión se han publicado varias versiones de Spring en las cuales se han implementado mejoras sustanciales de manera secuencial y modular a las funcionalidades propias de la herramienta y con las versiones más recientes se ha ido adaptando a nuevas tendencias de desarrollo. (Hinkula, 2018, pág. 45)

Spring proporciona un entorno de desarrollo escalable primando ante toda la seguridad y la separación de las capas de negocio. (Cosmina, Harrop, Schaefer, Chris, & Ho, 2017, pág. 27)

## **1.4. Metodología de desarrollo**

### **1.4.1. Análisis de la Metodología**

En base al análisis y comparación que se ha realizado sobre las distintas metodologías presentadas, se ha optado por trabajar con la metodología XP, ya que dada la situación en la cual se va a desarrollar el proyecto de registro de cambios, la metodología XP presenta características que se adaptan de mejor manera para el desenvolvimiento del proyecto, entre las cuales están: (Reina & L., 2012)

- Desarrollo iterativo e incremental
- Pruebas unitarias continuas
- Refactorización del código

- Propiedad del código compartida (Reina & L., 2012)

Tabla 1. Propiedades metodologías de desarrollo

Área/Metodología	XP	SCRUM	OPENUP	RUP	UP
Presenta tolerancia en el proceso de desarrollo	X				
Orientado al manejo de riesgos, centrado en la arquitectura.	X	X	X	X	X
Acepta variabilidad en requerimientos	X	X		X	
Utiliza técnicas y herramientas para trabajar en equipo.	X			X	
		X			
Revisión constante de ajustes de calidad	X	X	X	X	X
Establece y define procesos al interior de una organización.	X			X	
Realiza pruebas unitarias de manera constante	X				
Se permite el trabajo o desarrollo en parejas	X				
Mantiene comunicación constante con el product owner.	X				
Mantiene un enfoque en la producción y presentación del producto	X				X
Maneja desarrollo de software iterativo	X	X	X	X	X
Presenta una orientación hacia el cumplimiento de requisitos.	X	X	X	X	
Centrado en la productividad e innovación.	X	X		X	
Durante el desarrollo del producto permite un alto flujo de participación en el equipo.	X				
Maneja un esquema de trabajo variable	X				

Nota: Se presentan las características de las metodologías ágiles

Elaborado por: José Bastidas, Jimmy Puente

#### 1.4.2. Metodología XP

Dentro del presente proyecto se ha optado por la utilización de esta metodología debido a que sus características se adaptan a la amplitud del proyecto de registro de cambios. Entre las características de esta metodología se tiene el desarrollo incremental, lo que significa que a medida que se desarrolla el código de la aplicación se van realizando pequeñas mejoras, esta característica en conjunto con la comunicación constante con el usuario final ha permitido generar avances significativos para la presentación del producto final. (Reina & L., 2012)

Otra característica que se puede evidenciar dentro del desarrollo del proyecto es la facilidad que brinda para el trabajo en parejas, facilitando de esta manera la productividad ya que, el código, al estar a cargo de dos personas es constantemente revisado y evaluado, dando lugar a una mayor productividad.

### **1.4.3. Fases de la metodología**

Dentro de las fases que componen un proyecto desarrollado bajo la metodología de programación extrema se tiene 4 etapas en las cuales se enmarcan todas las tareas relacionadas al avance del proyecto, las mencionadas etapas se describen a continuación:

#### **1.4.3.1. Primera fase: Planificación del proyecto**

El primer paso al desarrollar un proyecto bajo la metodología de programación extrema es definir las historias de usuario a fin de servir como base para el posterior desarrollo, en las historias de usuario se puede definir tiempos para desarrollo de la aplicación y a la vez una historia de usuario permite diagramar un bosquejo de la funcionalidad que se va a presentar. Posteriormente la historia de usuario permite comprobar si la funcionalidad presentada cumple con los requisitos especificados previamente. (Reina & L., 2012)

Para el desarrollo del presente proyecto, se está llevando a cabo por un equipo de dos desarrolladores, basándonos en los preceptos de la metodología, el trabajo en parejas posibilita la creación de un software enfocado en la calidad y la productividad al desarrollar componentes totalmente funcionales que son presentados al usuario final. Finalmente, en esta etapa se contempla las reuniones periódicas con el usuario final, por lo cual se ajusta y se alinea constantemente el proyecto con las necesidades del cliente. (Reina & L., 2012)

#### **1.4.3.2. Segunda fase: Diseño**

Una de las propuestas de la metodología de programación extrema es la flexibilidad y sencillez de diseños. Se procura mantener un diseño mínimamente complicado para conseguir un producto final que sea de fácil implementación y que resulte comprensible permitiendo optimizar tiempo y esfuerzo al momento de desarrollar el proyecto.

Una parte importante de la metodología de programación extrema es la refactorización de código, ya que permite modificar la estructura de la codificación del proyecto favoreciendo la calidad del proyecto. Esta parte del diseño está centrada en optimizar la funcionalidad del producto y busca evitar código inestable o fallas en el diseño que pueden comprometer el correcto desarrollo de la aplicación. (Reina & L., 2012)

#### **1.4.3.3. Tercera fase: Codificación**

En la fase de codificación se deben atender se tienen como pilar fundamental las fases previas, ya que todo el desarrollo del proyecto está basado en la diagramación propuesta por el usuario.

La codificación del proyecto debe apegarse a estándares que mantengan coherencia para facilitar la flexibilidad y comprensión del proyecto.

#### **1.4.3.4. Cuarta fase: Pruebas**

Es fundamental, dentro de la metodología XP, la verificación de cumplimiento de los componentes q se vayan desarrollando. Para esta fase se debe crear en primer lugar las pruebas que se van a ejecutar.

Se puede implementar una prueba de funcionamiento luego de una refactorización de código para evidenciar que el funcionamiento no ha sufrido ninguna modificación en su salida

## CAPÍTULO 2

### 2.1. Requerimientos de aplicación

En el siguiente apartado se detalla la especificación de requisitos funcionales y no funcionales que se ven involucrados dentro del desarrollo de la aplicación.

#### 2.1.1. Requisitos funcionales

En la sección de requerimientos funcionales se describen los servicios que el software prestará a los usuarios finales de la ARCH, se define además el comportamiento interno del software; cálculos, detalles técnicos, administración de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica.

A continuación, se detallan los requerimientos funcionales:

##### 2.1.1.1. Seguimiento de novedades detectadas

Para el seguimiento de novedades se presentan los requisitos funcionales los cuales muestran las necesidades del módulo de control de novedades.

Tabla 2. Requisito funcional Seguimiento de novedades

SITP – Requisito funcional	
<b>Código</b>	CRF – 01
<b>Nombre de requerimiento</b>	Seguimiento de novedades
<b>Descripción</b>	Permite registrar el seguimiento de novedades detectadas
<b>Relevancia</b>	Alta
<b>Proceso</b>	<ol style="list-style-type: none"><li>1. Se ingresa al sistema con las credenciales de usuario y clave</li><li>2. El usuario debe ingresar a la opción seguimiento de novedades</li><li>3. Un número secuencial de novedad debe ser cargado</li><li>4. Se debe registrar el número de documento, entrada o salida de documento.</li><li>5. Se debe seleccionar asunto, tipo de novedad.</li><li>6. El usuario debe seleccionar el estado</li><li>7. Se debe guardar toda la información generada</li><li>8. Se visualiza los datos registrados.</li></ol>
<b>Tipo</b>	Nueva opción

<b>Condición previa</b>	Ninguna
<b>Condición posterior</b>	Ninguna
<b>Salida / Resultado</b>	<ul style="list-style-type: none"> <li>• Se debe guardar la información generada por el usuario, adicional se guarda el usuario y fecha de registro.</li> <li>• El estado de la novedad cambia de registrado a activo según el parámetro de días en el que se lleve a cabo la actividad.</li> </ul>

Nota: Se presentan los requisitos funcionales de Seguimiento de novedades  
Elaborado por: José Bastidas, Jimmy Puente

### 2.1.1.2. Crear portafolio

Para la creación de portafolio se presentan los requisitos funcionales los cuales abarcan las condiciones para la apertura de un portafolio de trabajo.

#### 2.1.1.2.1. Creación de portafolio de trabajos en pozos

Al crear un portafolio de trabajo es necesario tomar en cuenta los perfiles que se encuentran marcados para esta tarea.

Tabla 3. Requisito funcional Creación de portafolio

<b>SITP – Requisito funcional</b>	
<b>Código</b>	CRF - 02
<b>Nombre de requerimiento</b>	Creación de portafolio de trabajos en pozos
<b>Descripción</b>	Permite la creación del portafolio de trabajos en pozo
<b>Relevancia</b>	Alta
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. Se debe ingresar al sistema con las credenciales de usuario y contraseña.</li> <li>2. El usuario debe ingresar a la opción de crear portafolio</li> <li>3. Se debe seleccionar el campo y el pozo al cual se realizará el trabajo. Se deben visualizar los trabajos registrados de anteriores portafolios correspondientes al pozo en caso de existir.</li> <li>4. Se debe seleccionar el consorcio, si es el caso dentro del portafolio que se está creando.</li> <li>5. Se debe seleccionar trabajo.</li> <li>6. Adicional al paso anterior se debe seleccionar el número de trabajo.</li> </ol>

	<p>7. Se debe seleccionar la fecha de notificación o solicitud de aprobación de TST</p> <p>8. Se procede a guardar toda la información generada por el usuario.</p> <p>9. Los datos cargados deben ser visualizados o se debe mostrar un mensaje de confirmación de datos cargados dentro del sistema.</p>
<b>Tipo</b>	Nueva opción
<b>Condición previa</b>	Ninguna
<b>Condición posterior</b>	<p>Para la selección del consorcio en caso de que no se tenga un consorcio se debe seleccionar no aplica o N/A.</p> <p>En la selección de número de trabajo (paso número 6 del procedimiento), en el caso de CYPI no se debe activar el campo número y fecha de trabajo sin torre.</p> <p>En la selección de número de trabajo (paso número 6 del procedimiento), para el caso de TST, y al no tener el número de trabajo se debe permitir registrar no aplica en número y además se debe habilitar el campo fecha de trabajo sin torre. (Fecha notificación o solicitud de aprobación).</p> <p>En relación a la selección de campo y pozo en la cual se realizará el trabajo se debe cargar la operadora, el bloque y el campo del pozo seleccionado.</p> <p>Dentro de las opciones de selección de trabajo se puede tener como opciones: CYPI: Completación y pruebas iniciales, Reacondicionamiento y Trabajo sin torre.</p> <p>Con respecto a la selección de número de trabajo se debe realizar una validación para que no se repita el pozo, ni el trabajo, ni el número.</p> <p>Cuando se seleccione la fecha de notificación o solicitud de aprobación de TST la fecha no debe ser menor a 2010 y tampoco puede ser mayor que la fecha actual.</p>
<b>Salida / Resultado</b>	<p>Al guardar la información generada por el usuario se deben realizar validaciones para que no se repita el pozo, ni el trabajo, ni el número o la fecha de notificación de TST.</p> <p>Adicionalmente se debe generar un número de portafolio, se debe presentar la fecha de registro actual junto al funcionario que realizó la creación, y se debe notificar a los funcionarios responsables del seguimiento al pozo.</p>

Nota: Se presentan los requisitos funcionales de Creación de portafolio  
Elaborado por: José Bastidas, Jimmy Puente

### 2.1.1.2.2. Buscar portafolio de trabajos en pozos

Dentro de las tareas que se pueden realizar con un portafolio se encuentra la búsqueda para ítems u objetivos específicos.

Tabla 4. Requisito funcional Buscar portafolio

SITP – Requisito funcional	
<b>Código</b>	CRF - 03
<b>Nombre de requerimiento</b>	Buscar portafolio de trabajos en pozos
<b>Descripción</b>	Permite buscar un portafolio de trabajos en pozos
<b>Relevancia</b>	Alta
<b>Proceso</b>	<ol style="list-style-type: none"><li>1. El usuario deberá ingresar al sistema con las credenciales de usuario y clave</li><li>2. Se selecciona la opción de buscar portafolio</li><li>3. Se debe permitir buscar por número de portafolio, entre fechas desde hasta o por el funcionario que creo, o pozo.</li><li>4. Se procede a buscar dentro del sistema.</li><li>5. Los resultados de búsqueda son visualizados</li><li>6. Selecciona en acción las opciones del menú según el perfil de usuario.</li><li>7. Marca Editar (únicamente en el caso de ser administrador)</li></ol>
<b>Condición previa</b>	Ninguna
<b>Condición posterior</b>	<p>Al realizar la búsqueda dentro del sistema se deben cargar los resultados de la búsqueda por: Operadora, bloque, campo, pozo, trabajo, número, fecha.</p> <p>Dependiendo del usuario se deben habilitar diferentes opciones, los usuarios pueden ser funcionario matriz o funcionario regional.</p>
<b>Entrada</b>	
<b>Salida / Resultado</b>	<p>Se presentan los resultados de búsqueda dependiendo de usuario que haya solicitado la información.</p> <p>Si el usuario es administrador se debe cargar la pantalla de edición de portafolio de trabajo en pozo.</p>

Nota: Se presentan los requisitos funcionales de Buscar portafolio

Elaborado por: José Bastidas, Jimmy Puente

### 2.1.1.2.3. Editar o anular portafolio de trabajos en pozos

Una tarea adicional a las que se pueden realizar con un portafolio se encuentra la edición o anulación para ítems u objetivos específicos.

Tabla 5. Requisito funcional Editar o Anular portafolio

SITP – Requisito funcional	
<b>Código</b>	CRF - 04
<b>Nombre de requerimiento</b>	Editar o anular portafolio de trabajos en pozos
<b>Descripción</b>	Permite editar o anular un portafolio de trabajos en pozos
<b>Relevancia</b>	Alta
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. Ingresa al sistema con las credenciales de usuario y clave</li> <li>2. Ingresar a la opción de buscar portafolio o ingresar desde el menú</li> <li>3. Selecciona el editar.</li> <li>4. Permite editar los campos pozo, consorcio, trabajo, número y fecha de trabajo sin torre.</li> <li>5. Guardar</li> <li>6. Anular</li> <li>7. Visualizar historial de pozo.</li> <li>8. Visualizar datos editados</li> </ol>
<b>Condición previa</b>	Ninguna
<b>Condición posterior</b>	<p>Se debe realizar el respaldo o guardado de la información dentro del aplicativo previamente al cambio de todos los atributos.</p> <p>El motivo de cambio debe ser registrado en una sección de la pantalla que será activada al seleccionar la opción de cambio.</p> <p>También se debe activar el campo denominado observación, en el cual se podrán registrar datos adicionales.</p> <p>Al anular un portafolio se debe tomar en consideración si existe información o anexos referentes al portafolio, este hecho debe ser notificado al usuario antes de realizar la anulación.</p>
<b>Salida / Resultado</b>	<p>Se espera una tener activas y funcionales las opciones de edición y anulación para portafolio de trabajos en pozo.</p> <p>Para realizar una anulación de manera satisfactoria se mostrará una pantalla de confirmación.</p> <p>Se tendrá un historial de las anulaciones realizadas.</p>

Nota: Se presentan los requisitos funcionales de Editar o Anular portafolio

Elaborado por: José Bastidas, Jimmy Puente

### 2.1.1.3. Registro reporte diario trabajos en pozo

Un módulo fundamental que se encuentra dentro del sistema para la ARCH es el registro de reportes diarios dentro de los trabajos efectuados en un pozo.

Tabla 6. Requisito funcional Registro de reporte diario

SITP – Requisito funcional	
<b>Código</b>	CRF – 05
<b>Nombre de requerimiento</b>	Registro reporte diario trabajos en pozo
<b>Descripción</b>	Permite registrar el reporte diario, las suspensiones de trabajos y los reinicios de trabajo en un pozo. Y el registro de la fecha de finalización de todo el trabajo.
<b>Relevancia</b>	Alta
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con las credenciales de usuario y clave</li> <li>2. Ingresar a la opción de Reporte Diario/ menú de búsqueda de portafolio.</li> <li>3. Se selecciona registrar trabajo</li> <li>4. Se carga automáticamente la información de: Número de portafolio, fecha registro de portafolio, el funcionario que registro el portafolio, el pozo, trabajo, número de trabajo o fecha de trabajo sin torre.</li> <li>5. Permite el registro de la fecha de inicio de trabajo, y una vez culminado el trabajo se debe registrar la fecha de fin de operaciones.</li> <li>6. Guardar (permite guardar la fecha de inicio y fecha de fin de trabajos en pozo).</li> <li>7. Añadir registro diario             <ol style="list-style-type: none"> <li>a. Permite añadir una acción de registro diario</li> <li>b. Permite registrar la fecha de acción del trabajo</li> <li>c. Permite registrar la actividad</li> </ol> </li> <li>8. Añadir suspensión             <ol style="list-style-type: none"> <li>a. Permite añadir una acción de suspensión</li> <li>b. Permite registrar la fecha de acción (Fecha de inicio de suspensión).</li> <li>c. Permite registrar la actividad (Justificación de suspensión).</li> </ol> </li> <li>9. Añadir reinicio             <ol style="list-style-type: none"> <li>a. Permite registrar una acción de reinicio</li> <li>b. Permite registrar fecha de acción (fecha de reinicio)</li> <li>c. Permite registrar la acción</li> </ol> </li> </ol>

	<p>10. Guardar</p> <p>11. En la tabla se visualiza la información registrada y permite editar o anular los reportes.</p>
<b>Condición previa</b>	Ninguna
<b>Condición posterior</b>	<p>Con relación al guardado de información se debe ejecutar un proceso diario para realizar una verificación de los registros diarios en la cual se compruebe si dichos registros tienen el parámetro de tiempo deben cambiar de estado a cerrado.</p> <p>Al visualizar la información registrada, en la cual también se permite editar o anular, se puede editar fuera del tiempo de corrección mediante una solicitud al administrador.</p> <p>Cuando ya se vaya a guardar la información, la fecha no puede ser mayor a la actual, además la fecha de inicio y fin no puede ser menor al año 2010.</p> <p>Al añadir un registro diario, al añadir una suspensión y al añadir un reinicio la fecha de acción no puede ser menor a la fecha de inicio de trabajos en pozo, la fecha de acción no puede ser mayor a la fecha actual. En caso de suspensión y reinicio, después de realizar las comprobaciones, se deben activar las opciones correspondientes a cada acción, ya sea suspensión o reinicio.</p> <p>Con respecto a la opción de guardado se debe guardar el usuario que creo el registro diario, la suspensión o el reinicio. Adicionalmente el estado del registro debe cambiar a registrado.</p> <p>Los únicos reportes que se pueden anular son los que constan como estado registrado.</p>
<b>Salida / Resultado</b>	Se espera un módulo con funcionalidad que al añadir un registro diario permita guardar la información, añadir suspensión o reinicio a un registro.

Nota: Se presentan los requisitos funcionales de Registro de reporte diario

Elaborado por: José Bastidas, Jimmy Puente

#### **2.1.1.4. Registro documentos habilitantes operadora**

Dentro de los detalles necesarios para el registro correcto de un trabajo se encuentra los documentos habilitantes los cuales facultan y respaldan un trabajo.

Tabla 7. Requisito funcional Registro documentos habilitantes

<b>SITP – Requisito funcional</b>	
<b>Código</b>	CRF - 06
<b>Nombre de requerimiento</b>	Registro documentos habilitantes operadora
<b>Descripción</b>	Permite registrar la documentación habilitante de los trabajos en pozo, información de pagos por tasas de servicios de control y adjuntar los archivos correspondientes.
<b>Relevancia</b>	Alta
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. Ingresa al sistema con las credenciales de usuario y clave</li> <li>2. Ingresar a la opción de buscar portafolio, seleccionar adjuntar documentos</li> <li>3. Se carga automáticamente la información de: Número de portafolio, fecha registro de portafolio, el funcionario que registro el portafolio, el pozo, trabajo, fecha de inicio, número de trabajo o fecha de trabajo sin torre (MEMBRETE DE IDENTIFICACIÓN DEL PORTAFOLIO)</li> <li>4. Selecciona agregar documentos operadora.</li> <li>5. Permite el registro de: Nro. SGC, Nro. Oficio, Fecha de Oficio, Fecha ARCH, objetivo y el RIG.</li> <li>6. Permite adjuntar archivos (Oficio, anexos del Oficio y Documento del pago)</li> <li>7. Seleccionar: categorización, asunto del documento</li> <li>8. Guardar</li> <li>9. Permite el registro de: fecha de pago, # ítem, Nro. de comprobante, Nro. de transferencia, Nro. de factura</li> <li>10. Guardar</li> <li>11. Se visualiza en la tabla la información guardada o mensaje de confirmación de guardado.</li> <li>12. Permite visualizar los documentos adjuntos.</li> <li>13. Editar - Anular</li> </ol>
<b>Condición previa</b>	Ninguna
<b>Condición posterior</b>	<p>Con referencia a los archivos adjuntos únicamente se habilita para el registro de datos de pago dependiendo del tipo de asunto que pueden ser: notificación o solicitud de inicio de trabajo.</p> <p>Al seleccionar la opción de categorización se debe deshabilitar para el tipo de trabajo CYPI.</p> <p>Cuando se guarde la información se debe ejecutar un proceso que verifique si los documentos tienen el parámetro de tiempo, en caso de contener el campo el estado debe cambiar a cerrado.</p> <p>Los documentos de operadora que pueden ser agregados están en el rango de 1 a 10 por portafolio.</p>

	Al guardar la información también se debe guardar el usuario que realizó el registro y la fecha de registro.  Los archivos que se pueden adjuntar pueden ser hasta 6.
<b>Entrada</b>	
<b>Salida / Resultado</b>	Se espera la funcionalidad para registro de documentos habilitantes.

Nota: Se presentan los requisitos funcionales de Registro documentos habilitantes

Elaborado por: José Bastidas, Jimmy Puento

### 2.1.1.5. Informe de resultados de trabajo

El informe de resultado de trabajo presenta de manera detallada los trabajos realizados en un pozo.

Tabla 8. Requisito funcional Informe resultados de trabajo

<b>SITP – Requisito funcional</b>	
<b>Código</b>	CRF - 07
<b>Nombre de requerimiento</b>	Informe de resultados de trabajo
<b>Descripción</b>	Permite registrar el informe de resultados de trabajo remitido por la operadora y adjuntar archivos
<b>Relevancia</b>	Alta
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. Ingresa al sistema con las credenciales de usuario y clave</li> <li>2. Buscar el portafolio, según los filtros de búsqueda</li> <li>3. Marcar la opción Informe de resultados en acciones</li> <li>4. Se visualiza el membrete de identificación del portafolio</li> <li>5. Se carga automáticamente número de informe</li> <li>6. Registrar Nro. SGC, Nro. de oficio, fecha de oficio, fecha ARCH, RIG, costo real.</li> <li>7. Adjuntar documentos</li> <li>8. Marcar ver documentos adjuntos</li> <li>9. Registra estado</li> <li>10. Guardar</li> <li>11. Se visualiza los datos guardados o se indica un mensaje de guardado.</li> </ol>
<b>Condición previa</b>	Ninguna
<b>Condición posterior</b>	<p>Con referencia a la opción de Informe de resultados en acciones solamente se puede cargar un informe de resultado, la numeración debe estar ligada al número de portafolio.</p> <p>Al registrar los números de SGC y de oficio la fecha debe ser menor a la fecha actual.</p>

	Al adjuntar documentos se pueden agregar hasta 3 archivos entre oficios y anexos.
	Al registrar el estado se cambia el estado según el parámetro de días y además se permite la edición o cierre del registro
<b>Salida / Resultado</b>	Se tiene un módulo con el cual se pueda presentar un informe de los resultados de trabajo.

Nota: Se presentan los requisitos funcionales de Informe resultados de trabajo

Elaborado por: José Bastidas, Jimmy Puento

### 2.1.1.6. Verificación de fechas de entrega de información

Cuando se realiza una operación dentro de un portafolio es necesario realizar una comprobación de las fechas en las cuales se entregó la información.

Tabla 9. Requisito funcional Verificación de fechas

<b>SITP – Requisito funcional</b>	
<b>Código</b>	CRF - 08
<b>Nombre de requerimiento</b>	Verificación de fechas de entrega de información
<b>Descripción</b>	Permite verificar las fechas de entrega de informes de resultados.
<b>Relevancia</b>	Alta
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con las credenciales de usuario y clave</li> <li>2. Buscar el portafolio</li> <li>3. Ingresar a la opción de acción: verificación de fechas.</li> <li>4. Mostrar el membrete de identificación del portafolio</li> <li>5. Se carga automáticamente fecha de inicio y fin de trabajo, fecha de entrega de informe de resultados y fecha de notificación o solicitud de aprobación.</li> <li>6. Valoración o verificación (Cumple /No cumple)</li> <li>7. Seleccionar Justificativo (SI/NO)</li> <li>8. Describir la observación</li> <li>9. Seleccionar estado</li> <li>10. Guardar</li> <li>11. Se visualiza los datos registrados.</li> </ol>
<b>Condición previa</b>	Ninguna
<b>Condición posterior</b>	Cuando se cargan las fechas de notificación o solicitud de aprobación pueden existir opciones de cumplimiento cuando la fecha de inicio de operación es mayor o igual a la fecha de notificación o solicitud, y no cumplimiento cuando la fecha de inicio de operación es menor a la fecha de notificación o solicitud.

	Al seleccionar un estado se debe guardar adicionalmente la información de usuario y fecha de registro.
<b>Salida / Resultado</b>	Como resultado se espera un módulo que proporcione opciones para validar fechas de entrega de información.

Nota: Se presentan los requisitos funcionales de Verificación de fechas

Elaborado por: José Bastidas, Jimmy Puento

### 2.1.1.7.Verificación de producción

La verificación de producción permite a un funcionario realizar una comprobación sobre los datos que han sido obtenidos de la producción efectuada en un determinado pozo.

Tabla 10. Requisito funcional Verificación de producción

<b>SITP – Requisito funcional</b>	
<b>Código</b>	CRF - 09
<b>Nombre de requerimiento</b>	Verificación de producción
<b>Descripción</b>	Permite verificar la producción antes y después del trabajo
<b>Relevancia</b>	Alta
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. Ingresa al sistema con las credenciales de usuario y clave</li> <li>2. Buscar el portafolio</li> <li>3. Ingresar a la opción de acción: verificación de producción.</li> <li>4. Se carga automáticamente el membrete de identificación del portafolio y el tipo de pozo</li> <li>5. Seleccionar el Nro. de registros de producción del SICOHI que se desea verificar</li> <li>6. Marcar cargar antes de inicio de trabajo y cargar después de fin de trabajo</li> <li>7. Se visualiza los registros antes y después del trabajo: permitir seleccionar mínimo 3 valores de producción y realizar promedio de datos seleccionados para realizar la verificación</li> <li>8. El porcentaje de control se carga automáticamente de parámetros al atributo de la tabla.</li> <li>9. Se registra automáticamente Valoración</li> <li>10. Seleccionar justificativo</li> <li>11. Describir la observación</li> <li>12. Guardar</li> <li>13. Se visualiza los datos registrados.</li> </ol>
<b>Condición previa</b>	Ninguna
<b>Condición posterior</b>	Al verificar la producción se debe mostrar la opción de si el tipo de pozo es productor.

	<p>Con respecto a la visualización de registros antes y después del trabajo se debe mostrar el porcentaje de aumento o disminución de volumen de producción después del trabajo.</p> <p>El porcentaje de control que se muestra dependerá de las siguientes opciones:  <b>Exitoso.</b> - Cuando el resultado de las pruebas de producción se incrementó en un 10%  <b>Medianamente exitoso.</b> - Cuando el resultado de las pruebas de producción se incrementó hasta en un 9%  <b>No exitoso.</b> - Cuando el resultado de las pruebas de producción es menor a la producción antes del trabajo.</p> <p>Al guardar la información generada debe guardar adicionalmente la información de usuario y fecha de registro.</p>
<b>Salida / Resultado</b>	Se espera un módulo que permita realizar validaciones sobre los resultados de producción y evaluar su nivel de éxito.

Nota: Se presentan los requisitos funcionales de Verificación de producción  
Elaborado por: José Bastidas, Jimmy Puente

#### 2.1.1.8. Verificación de inyección / reinyección

La verificación de inyección permite a un funcionario realizar una comprobación sobre los datos que han sido obtenidos de los trabajos de inyección o reinyección efectuados en un pozo.

Tabla 11. Requisito funcional Verificación de inyección / reinyección

SITP – Requisito funcional	
<b>Código</b>	CRF - 10
<b>Nombre de requerimiento</b>	Verificación de inyección / reinyección
<b>Descripción</b>	Permite verificar la inyección / reinyección antes y después del trabajo
<b>Relevancia</b>	Alta
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. Ingresa al sistema con las credenciales de usuario y clave</li> <li>2. Buscar portafolio</li> <li>3. Ingresar a la opción acción: verificación de inyección o reinyección.</li> <li>4. Se carga automáticamente: Membrete de identificación del portafolio, tipo de pozo, fecha de inicio, fecha de fin del trabajo</li> <li>5. Seleccionar el Nro. de registros de inyección /reinyección del SICOHI que se desea verificar</li> </ol>

	6. Marcar cargar antes de inicio de trabajo y cargar después de fin de trabajo 7. Se visualiza los registros antes y después del trabajo 8. Se carga el porcentaje de control 9. Se registra automáticamente Valoración 10. Seleccionar justificativo 11. Describir la observación 12. Guardar 13. Se visualiza los datos registrados.
<b>Condición previa</b>	Ninguna
<b>Condición posterior</b>	Para el registro de la valoración de inyector o re - inyector se debe tomar en cuenta los siguientes parámetros de medición: <b>Exitoso.</b> - Cuando el volumen de inyección evidencie un incremento de un 10% de inyección <b>Medianamente exitoso.</b> - Cuando el volumen de inyección sea equivalente a un incremento de 9% antes del trabajo <b>No exitoso.</b> - Cuando el volumen de inyección sea menor al volumen de inyección antes del trabajo.  Al guardar la información generada debe guardar adicionalmente la información de usuario y fecha de registro.
<b>Salida / Resultado</b>	Se tendrá un módulo de aplicativo en el que se pueda verificar valores pertenecientes a procesos de inyección o re – inyección.

Nota: Se presentan los requisitos funcionales de Verificación de inyección / reinyección

Elaborado por: José Bastidas, Jimmy Puente

## 2.1.2. Requisitos no funcionales

### 2.1.2.1. Seguridad de aplicativo

El requisito de seguridad en aplicativo especifica los detalles acerca de los controles y parámetros que se deben establecer para un control sobre el acceso a la aplicación.

Tabla 12. Requisito no funcional Seguridad de aplicativo

SITP – Requisito No funcional	
<b>Código</b>	CRNF - 01
<b>Relevancia</b>	Alta
<b>Nombre de requerimiento</b>	Seguridad de aplicativo
<b>Descripción</b>	La seguridad que maneje la aplicación debe manejarse bajo el esquema de acceso en base a usuarios y credenciales, ya que de acuerdo al perfil que acceda al aplicativo se tendrán distintas funcionalidades.

	Las contraseñas que manejará la aplicación serán generadas y guardadas usando un esquema de encriptación.
--	---

Nota: Se presentan los requisitos no funcionales de Seguridad de aplicativo  
Elaborado por: José Bastidas, Jimmy Puente

### 2.1.2.2. Interfaz de aplicativo

El requisito no funcional de interfaz de aplicativo señala las características visuales que debe presentar la aplicación.

Tabla 13. Requisito no funcional Interfaz de aplicativo

SITP – Requisito No funcional	
<b>Código</b>	CRNF - 02
<b>Relevancia</b>	Alta
<b>Nombre de requerimiento</b>	Interfaz de aplicativo
<b>Descripción</b>	<p>La apariencia que será presentada al usuario deberá asegurar un manejo intuitivo y funcional que facilite las distintas tareas que incorpora el aplicativo.</p> <p>La distribución de las funciones que incorpora la aplicación deberán ser organizadas en categorías presentadas en un menú vertical juntando funciones pertenecientes a un mismo trabajo en el mismo grupo.</p> <p>El aplicativo generado será ejecutado y visualizado en navegadores web, la presentación y utilización del aplicativo será realizada enteramente a través de un navegador web.</p> <p>Las notificaciones y alertas generadas en la aplicación serán mostradas de una manera clara y sencilla al usuario final para que el manejo de alertas sea simple y conciso.</p>

Nota: Se presentan los requisitos no funcionales de Interfaz de aplicativo  
Elaborado por: José Bastidas, Jimmy Puente

### 2.1.2.3. Rendimiento de la aplicación

Las características de rendimiento indican el comportamiento que se espera que tenga la aplicación dentro del ambiente de trabajo de los funcionarios de la ARCH.

Tabla 14. Requisito no funcional Rendimiento de aplicativo

SITP – Requisito No funcional
-------------------------------

<b>Código</b>	CRNF – 03
<b>Relevancia</b>	Alta
<b>Nombre de requerimiento</b>	Rendimiento de la aplicación
<b>Descripción</b>	<p>El sistema deberá aceptar múltiples conexiones concurrentes y simultaneas debido al alcance que se dará a la aplicación.</p> <p>Dado el alcance que tendrá la aplicación es importante que dentro de la arquitectura se tome en cuenta acciones ante fallos que se puedan dar dentro de la aplicación.</p> <p>La utilización de la base datos deberá ser capaz de soportar la cantidad de peticiones y transacciones que se vayan a realizar manteniendo una alta disponibilidad.</p>

Nota: Se presentan los requisitos no funcionales de Rendimiento de aplicativo  
Elaborado por: José Bastidas, Jimmy Puente

#### 2.1.2.4. Requisitos operacionales del aplicativo

Las características operacionales del aplicativo indican el esquema sobre el cual funcionará la aplicación.

Tabla 15. Requisito no funcional Propiedades operacionales de aplicativo

SITP – Requisito No funcional	
<b>Código</b>	CRNF – 04
<b>Relevancia</b>	Alta
<b>Nombre de requerimiento</b>	Requisitos operacionales del aplicativo
<b>Descripción</b>	<p>La aplicación que será generada tendrá un uso destinado a navegadores web, para una mayor facilidad de uso.</p> <p>La arquitectura que manejará la aplicación será realizada bajo el esquema MVC.</p> <p>La base de datos que será utilizada por la aplicación será de tipo Oracle SQL, debido a que la organización ya utiliza esa base de datos y se desea integrar los sistemas bajo arquitecturas similares.</p>

Nota: Se presentan los requisitos no funcionales de Propiedades operacionales de aplicativo  
Elaborado por: José Bastidas, Jimmy Puente

## 2.2. Artefactos – Diagramas

En el siguiente apartado se presentan diagramas de casos de uso y de secuencia de los requisitos funcionales descritos para la aplicación.

### 2.2.1. Seguimiento de novedades

#### 2.2.1.1. Detalle de caso de uso

El detalle de caso uso de Seguimiento de Novedades describe las acciones que el actor efectuará en el sistema.

Tabla 16. Caso de Uso Seguimiento de Novedades

SITP – Caso de Uso	
<b>Nombre</b>	Seguimiento de novedades
<b>Actores</b>	Usuario matriz
<b>Características y funciones de caso de uso</b>	Ingresar al sistema Escoger opción de seguimiento de novedades Registrar documentos Seleccionar estado Guardar
<b>Fuentes</b>	Representantes Agencia de Regulación y Control Hidrocarburífero

Nota: Se presentan las características de caso de uso de Seguimiento de Novedades

Elaborado por: José Bastidas, Jimmy Puente

#### 2.2.1.2. Secuencia de actividades entre usuario y aplicativo

La secuencia de actividades de Seguimiento de Novedades detalla las acciones efectuadas tanto por el usuario, así como por el sistema.

Tabla 17. Secuencia de Actividades Seguimiento de Novedades

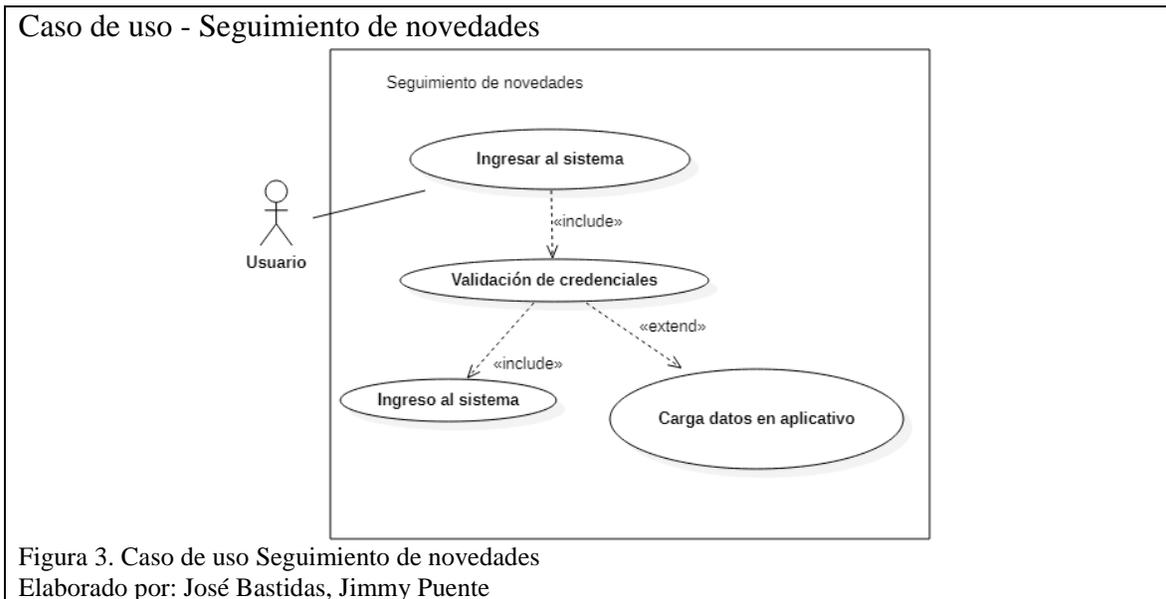
Acciones de usuarios	Acciones del sistema
El usuario ingresa al sistema con sus credenciales	
	Dentro del sistema se realizan validaciones para comprobar la autenticidad de las credenciales
El usuario debe seleccionar el estado del seguimiento de la novedad	

	El sistema carga los datos dentro de la pantalla del aplicativo mostrado al usuario.
--	--

Nota: Se presenta la secuencia de actividades de Seguimiento de Novedades  
Elaborado por: José Bastidas, Jimmy Puente

### 2.2.1.3. Diagrama de caso de uso UML

Se presenta el caso de uso para el seguimiento de novedades en el cuál se describe las acciones que el usuario realizará sobre el sistema.



### 2.2.1.4. Diagrama de secuencia UML

Se presenta el diagrama de secuencia para el seguimiento de novedades en el cuál se describe la interacción entre el usuario y el sistema.

## Diagrama de secuencia - Seguimiento de novedades

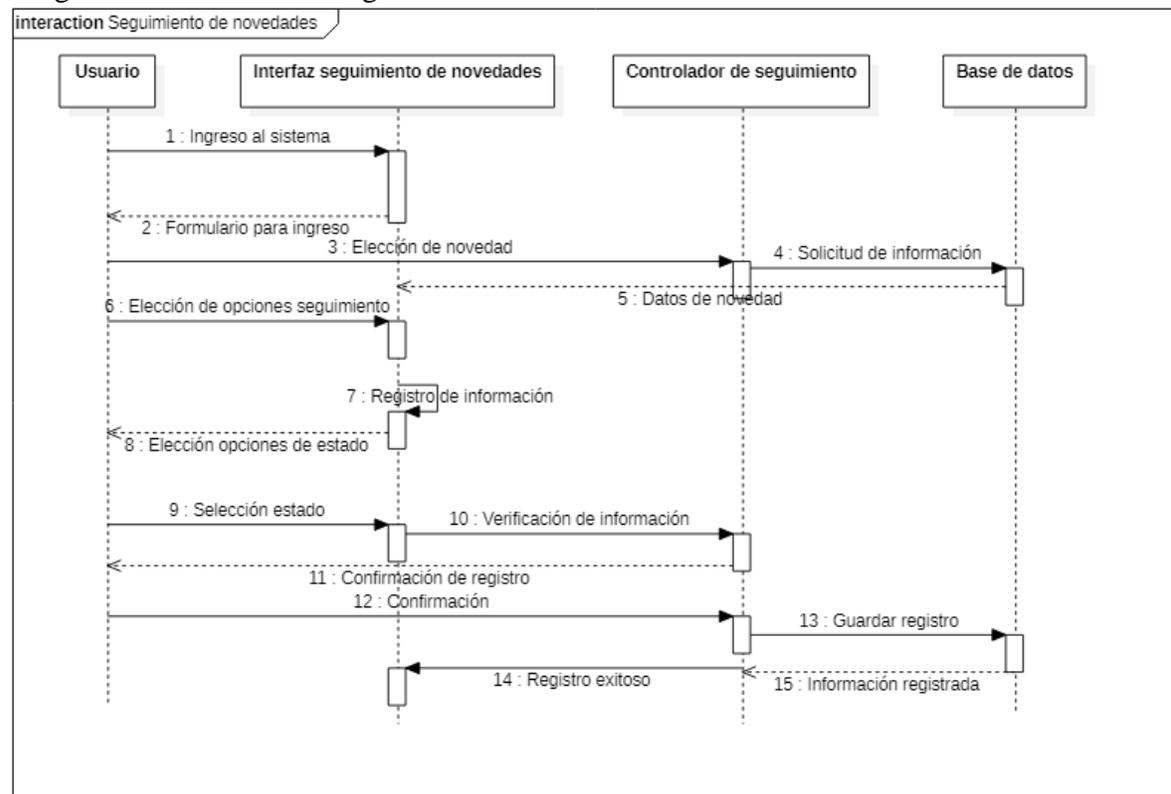


Figura 4. Diagrama de secuencia de Seguimiento de novedades  
Elaborado por: José Bastidas, Jimmy Puente

## 2.2.2. Creación de portafolio de trabajos en pozos

### 2.2.2.1. Detalle de caso de uso

El detalle de caso de uso de Creación de portafolio de trabajos en pozos describe las acciones que el actor efectuará en el sistema para iniciar un nuevo portafolio.

Tabla 18. Caso de Uso Creación de portafolio de trabajos en pozos

SITP – Caso de Uso	
<b>Nombre</b>	Creación de portafolio de trabajos en pozos
<b>Actores</b>	Funcionarios matriz o regional
<b>Características y funciones de caso de uso</b>	Creación de portafolio Selección de campo y pozo Selección de consorcio o trabajo Seleccionar opciones de fecha o aprobación Guardar datos Visualizar información guardada

<b>Fuentes</b>	Representantes Agencia de Regulación y Control Hidrocarburífero
----------------	---

Nota: Se presentan las características de caso de uso de Creación de portafolio de trabajos en pozos  
Elaborado por: José Bastidas, Jimmy Punte

### 2.2.2.2. Secuencia de actividades entre usuario y aplicativo

La secuencia de actividades de Creación de portafolio de trabajos en pozos detalla las acciones efectuadas tanto por el usuario, así como por el sistema para iniciar un nuevo portafolio.

Tabla 19. Secuencia de Actividades Creación de portafolio de trabajos en pozos

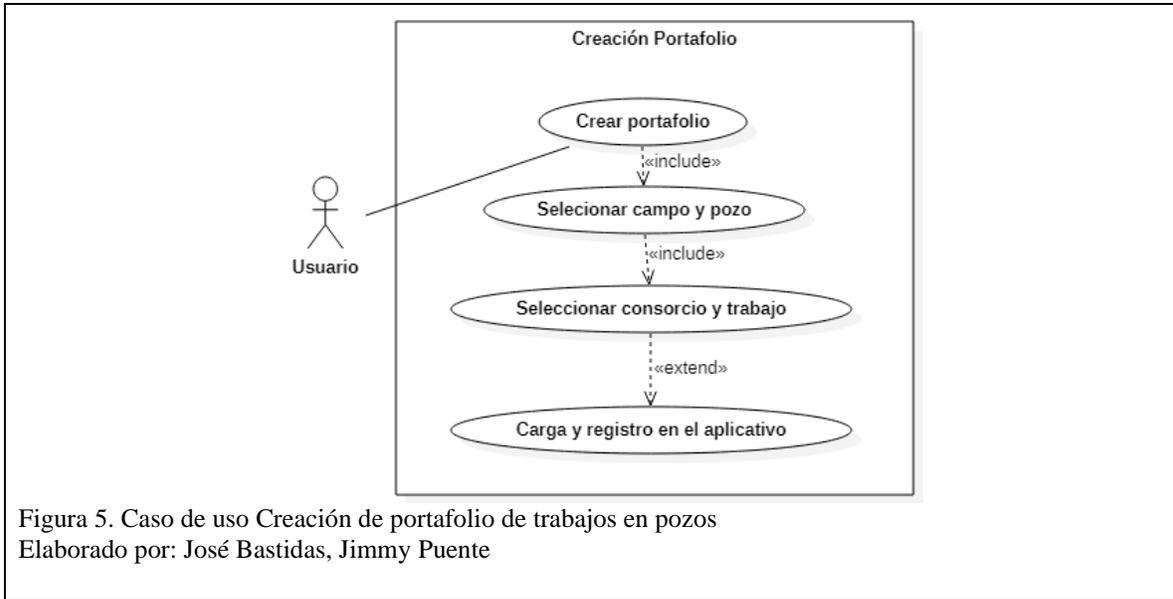
Acciones de usuarios	Acciones del sistema
El usuario selecciona la opción de crear portafolio	El sistema muestra dentro del formulario la información de campos o pozos guardados.
El usuario selecciona campo y pozo donde se realizaran los trabajos	
Se selecciona el consorcio y/o el trabajo	
	El sistema guarda en la base de datos la información ingresada por el usuario
	Se visualizan los datos del portafolio y se muestra un mensaje de confirmación de los datos cargados

Nota: Se presenta la secuencia de actividades de Creación de portafolio de trabajos en pozos  
Elaborado por: José Bastidas, Jimmy Punte

### 2.2.2.3. Diagrama de caso de uso UML

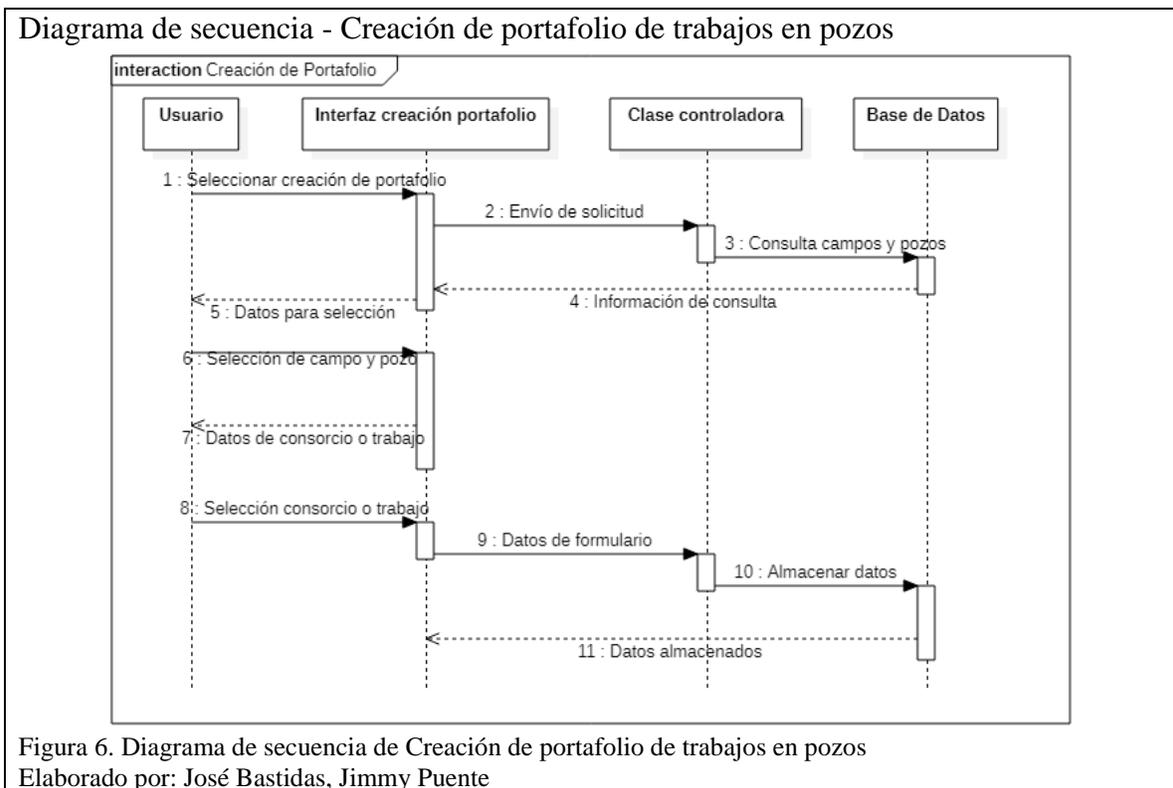
Se presenta el caso de uso para el Creación de portafolio de trabajos en pozos en el cuál se describe las acciones que el usuario realizará sobre el sistema.

Caso de uso - Creación de portafolio de trabajos en pozos
---



#### 2.2.2.4. Diagrama de secuencia UML

Se presenta el diagrama de secuencia para Creación de portafolio de trabajos en pozos en el cuál se describe la interacción entre el usuario y el sistema.



### 2.2.3. Buscar portafolio de trabajos en pozos

#### 2.2.3.1. Detalle de caso de uso

El detalle de caso uso de Buscar portafolio de trabajos en pozos describe las acciones que el actor efectuará en el sistema para realizar un búsqueda.

Tabla 20. Caso de Uso Buscar portafolio de trabajos en pozos

SITP – Caso de Uso	
<b>Nombre</b>	Buscar portafolio de trabajos en pozos
<b>Actores</b>	Administrador/Funcionario de regional y de matriz
<b>Características y funciones de caso de uso</b>	El usuario ingresa al sistema con sus credenciales Ingresar a opción de búsqueda de portafolio Buscar por número de portafolio, fechas Búsqueda en base a parámetros Visualización de datos
<b>Fuentes</b>	Representantes Agencia de Regulación y Control Hidrocarburífero

Nota: Se presentan las características de caso de uso de Buscar portafolio de trabajos en pozos  
Elaborado por: José Bastidas, Jimmy Puentes

#### 2.2.3.2. Secuencia de actividades entre usuario y aplicativo

La secuencia de actividades de Buscar portafolio de trabajos en pozos detalla las acciones efectuadas tanto por el usuario, así como por el sistema.

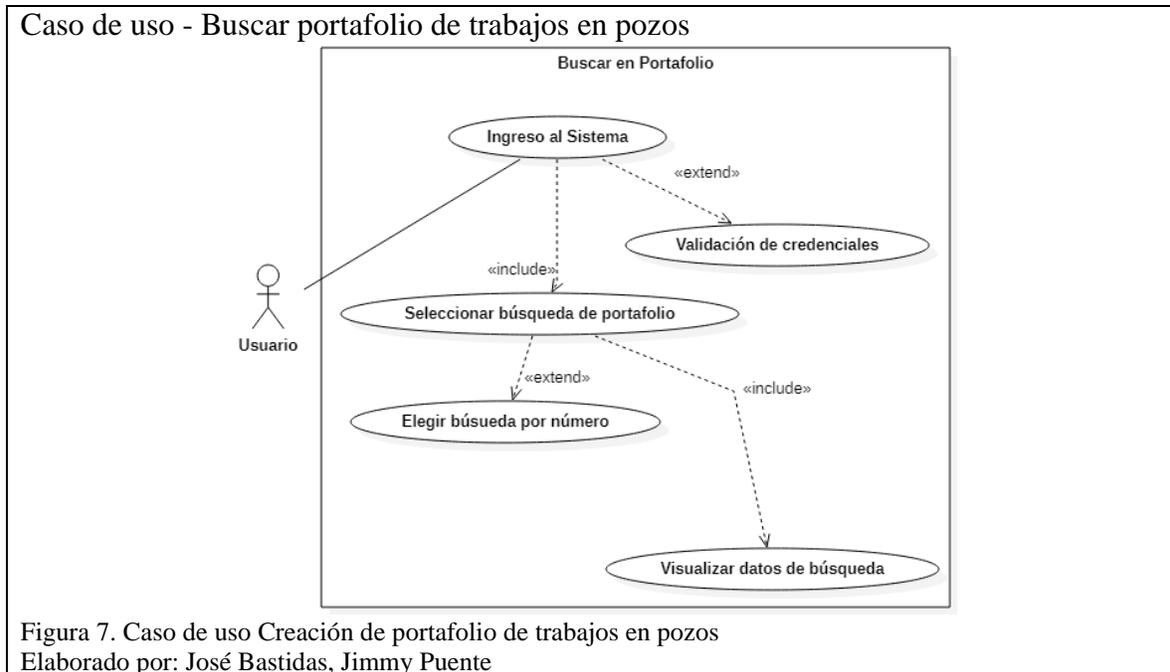
Tabla 21. Secuencia de Actividades Buscar portafolio de trabajos en pozos

Acciones de usuarios	Acciones del sistema
Ingreso al sistema con credenciales	
	El sistema valida las credenciales ingresadas
El usuario utiliza la opción de búsqueda de portafolio	
	El sistema permite la búsqueda mediante número de portafolio, fechas o pozo
El usuario procede a buscar en base a los parámetros ingresados	
	Se muestra la información en base a los parámetros de búsqueda

Nota: Se presenta la secuencia de actividades de Buscar portafolio de trabajos en pozos  
Elaborado por: José Bastidas, Jimmy Puente

### 2.2.3.3. Diagrama de caso de uso UML

Se presenta el caso de uso para Buscar portafolio de trabajos en pozos en el cuál se describe las acciones que el usuario realizará sobre el sistema.



### 2.2.3.4. Diagrama de secuencia UML

Se presenta el diagrama de secuencia para Buscar portafolio de trabajos en pozos en el cuál se describe la interacción entre el usuario y el sistema.

Diagrama de secuencia - Buscar portafolio de trabajos en pozos

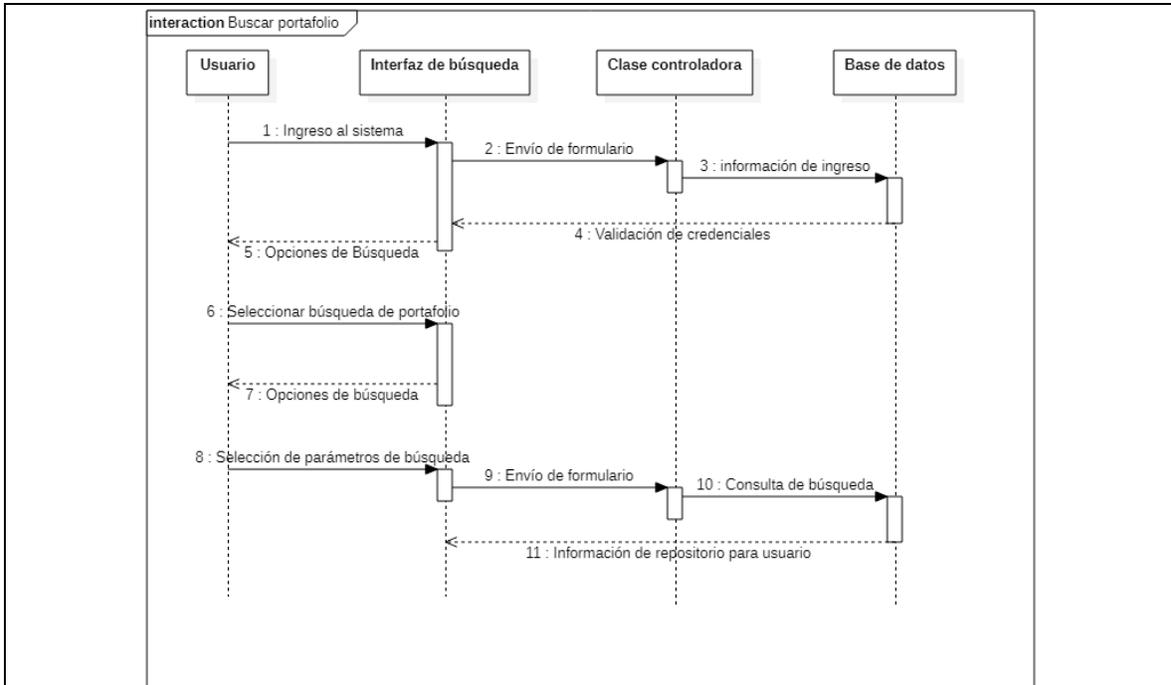


Figura 8. Diagrama de secuencia de Buscar portafolio de trabajos en pozos  
Elaborado por: José Bastidas, Jimmy Puente

## 2.2.4. Editar o anular portafolio de trabajos en pozos

### 2.2.4.1. Detalle de caso de uso

El detalle de caso uso de Editar o anular portafolio de trabajos en pozos describe las acciones que el actor efectuará en el sistema.

Tabla 22. Caso de Uso Editar o anular portafolio de trabajos en pozos

SITP – Caso de Uso	
<b>Nombre</b>	Editar o anular portafolio de trabajos en pozos
<b>Actores</b>	Administrador
<b>Características y funciones de caso de uso</b>	Ingreso al sistema en base a credenciales Abrir la opción de buscar portafolio. Seleccionar editar Se puede editar opciones de: campos, pozo, consorcio, trabajo, número y fecha de trabajo sin torre La información editada se guarda dentro del sistema. En caso de que se requiera la información puede ser anulada Se visualizan los datos editados
<b>Fuentes</b>	Representantes Agencia de Regulación y Control Hidrocarburiífero

Nota: Se presentan las características de caso de uso de Editar o anular portafolio de trabajos en pozos  
Elaborado por: José Bastidas, Jimmy Puente

### 2.2.4.2. Secuencia de actividades entre usuario y aplicativo

La secuencia de actividades de Editar o anular portafolio de trabajos en pozos detalla las acciones efectuadas tanto por el usuario, así como por el sistema.

Tabla 23. Secuencia de Actividades Editar o anular portafolio de trabajos en pozos

Acciones de usuarios	Acciones del sistema
Se ingresa al sistema mediante las credenciales de usuario	El sistema procede a validar las credenciales ingresadas.
En el menú de búsqueda se selecciona la opción de editar	
	Se permite la opción de edición en: campos, pozo, consorcio, trabajo, número y fecha de trabajo sin torre.
	La información editada se guarda dentro del sistema
Es posible anular la información editada	
	El sistema muestra el resultado de la interacción del usuario al editar o anular información.

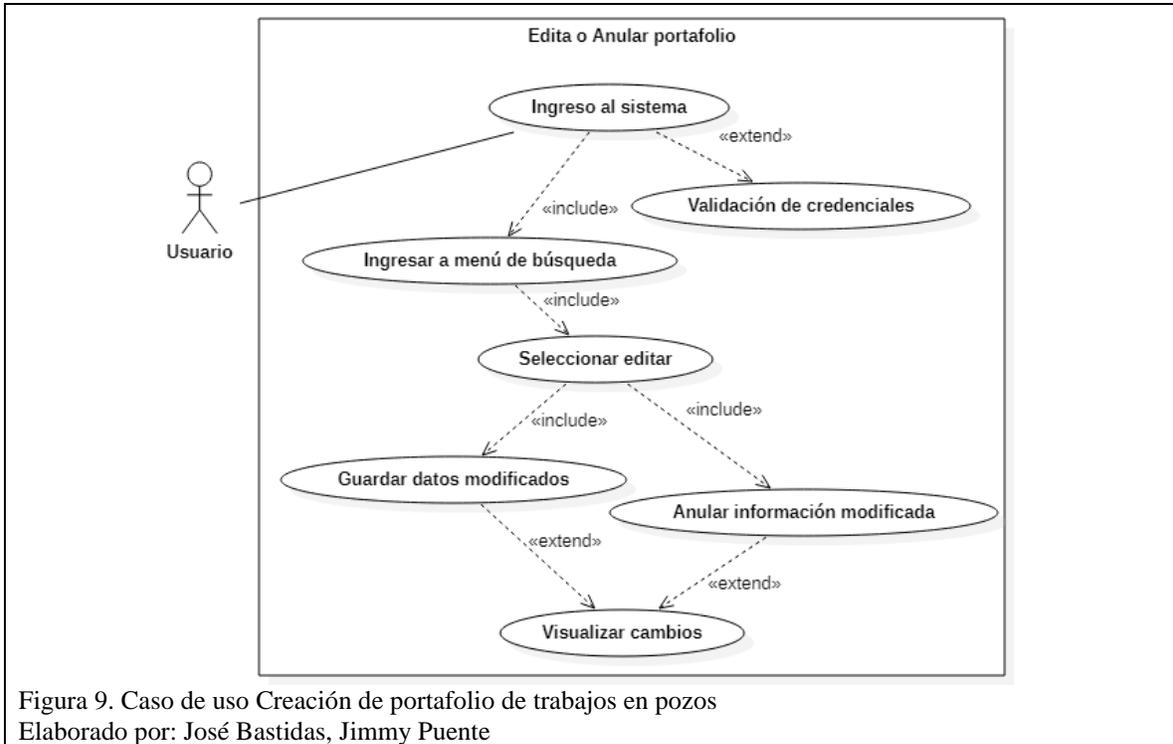
Nota: Se presenta la secuencia de actividades de Editar o anular portafolio de trabajos en pozos

Elaborado por: José Bastidas, Jimmy Puente

### 2.2.4.3. Diagrama de caso de uso UML

Se presenta el caso de uso para Editar o anular portafolio de trabajos en pozos en el cuál se describe las acciones que el usuario realizará sobre el sistema.

Caso de uso - Editar o anular portafolio de trabajos en pozos
---



#### 2.2.4.4. Diagrama de secuencia UML

Se presenta el diagrama de secuencia para Editar o anular portafolio de trabajos en pozos en el cuál se describe la interacción entre el usuario y el sistema.

Diagrama de secuencia - Editar o anular portafolio de trabajos en pozos

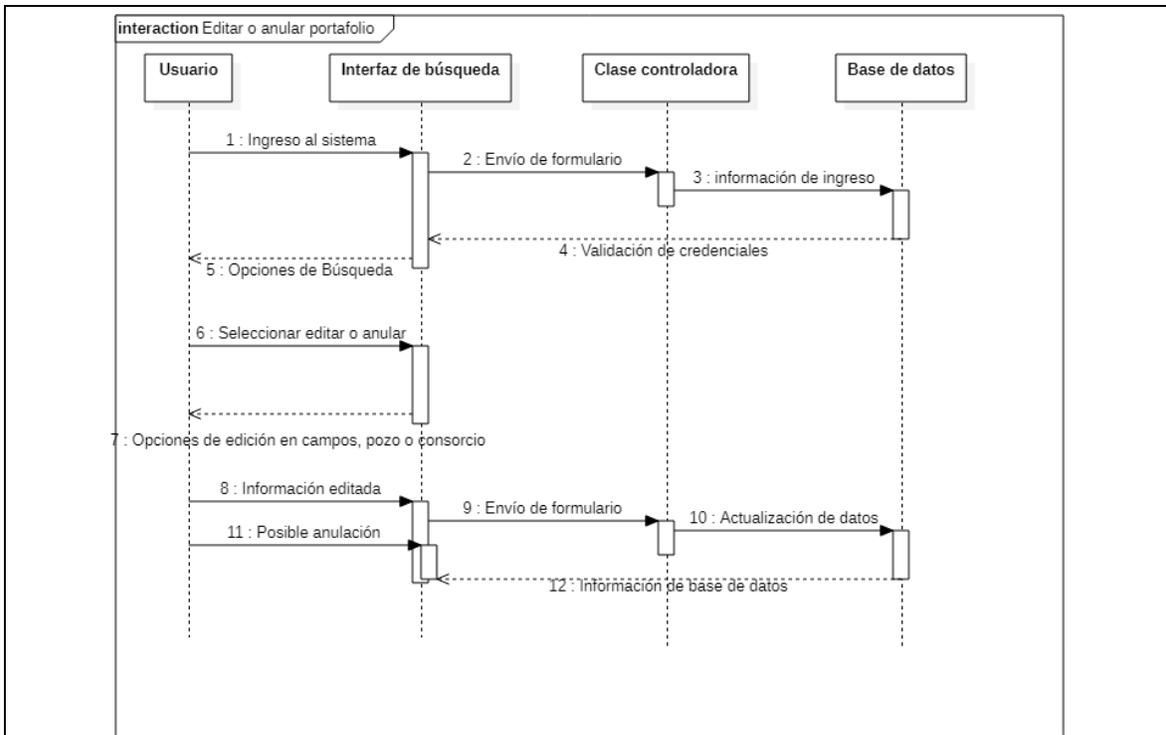


Figura 10. Diagrama de secuencia de Editar o anular portafolio de trabajos en pozos  
Elaborado por: José Bastidas, Jimmy Punte

## 2.2.5. Informe de resultados de trabajo

### 2.2.5.1. Detalle de caso de uso

El detalle de caso de uso de Informe de resultados de trabajo describe las acciones que el actor efectuará en el sistema.

Tabla 24. Caso de Uso Informe de resultados de trabajo

SITP – Caso de Uso	
<b>Nombre</b>	Informe de resultados de trabajo
<b>Actores</b>	Usuario matriz
<b>Características y funciones de caso de uso</b>	Búsqueda de portafolio mediante filtros de búsqueda Elegir opción de informe de resultados Visualización mediante membrete de portafolio Registro de número de y fechas de oficio Adjuntar documentos Registrar el estado Guardar información
<b>Fuentes</b>	Representantes Agencia de Regulación y Control Hidrocarburífero

Nota: Se presentan las características de caso de uso de Informe de resultados de trabajo  
Elaborado por: José Bastidas, Jimmy Punte

### 2.2.5.2. Secuencia de actividades entre usuario y aplicativo

La secuencia de actividades de Informe de resultados de trabajo se detalla las acciones efectuadas tanto por el usuario, así como por el sistema.

Tabla 25. Secuencia de Actividades Informe de resultados de trabajo

Acciones de usuarios	Acciones del sistema
Busca portafolio mediante filtros de búsqueda	
	Muestra el membrete de identificación de portafolio
	Carga de datos relacionados al informe
Registrar número y fecha de oficio	
Ajuntar documentos relacionados al trabajo	
Registro de estado de trabajo	
	La información es almacenada dentro de la base de datos

Nota: Se presenta la secuencia de actividades de Informe de resultados de trabajo  
 Elaborado por: José Bastidas, Jimmy Puente

### 2.2.5.3. Diagrama de caso de uso UML

Se presenta el caso de uso para Informe de resultados de trabajo en el cuál se describe las acciones que el usuario realizará sobre el sistema.

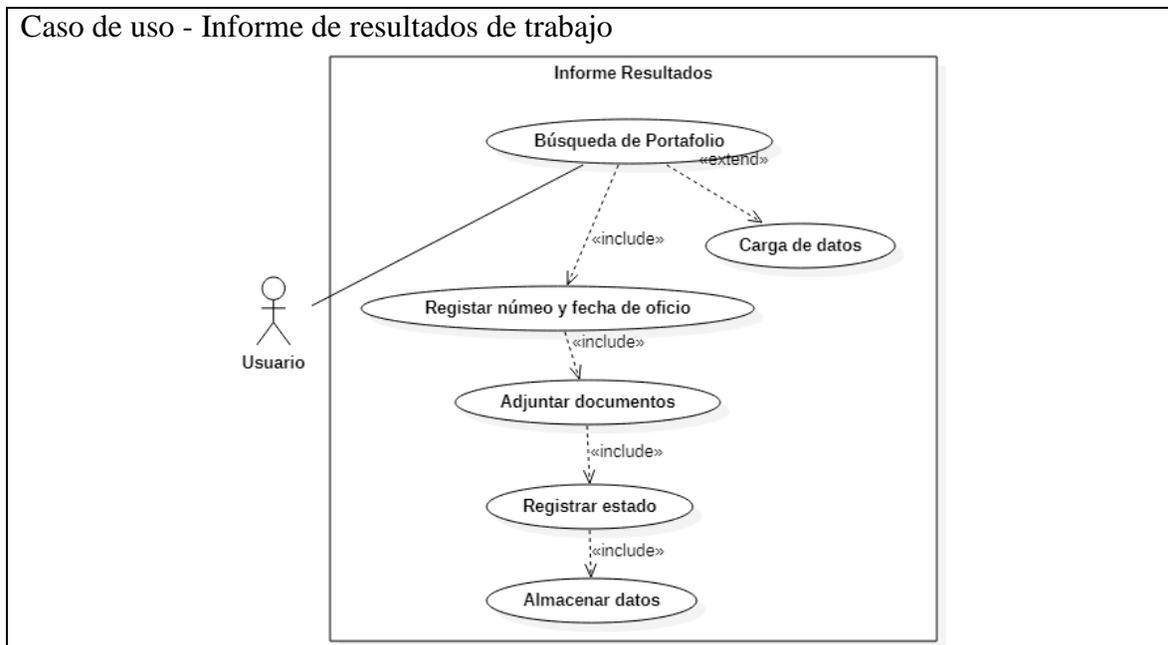
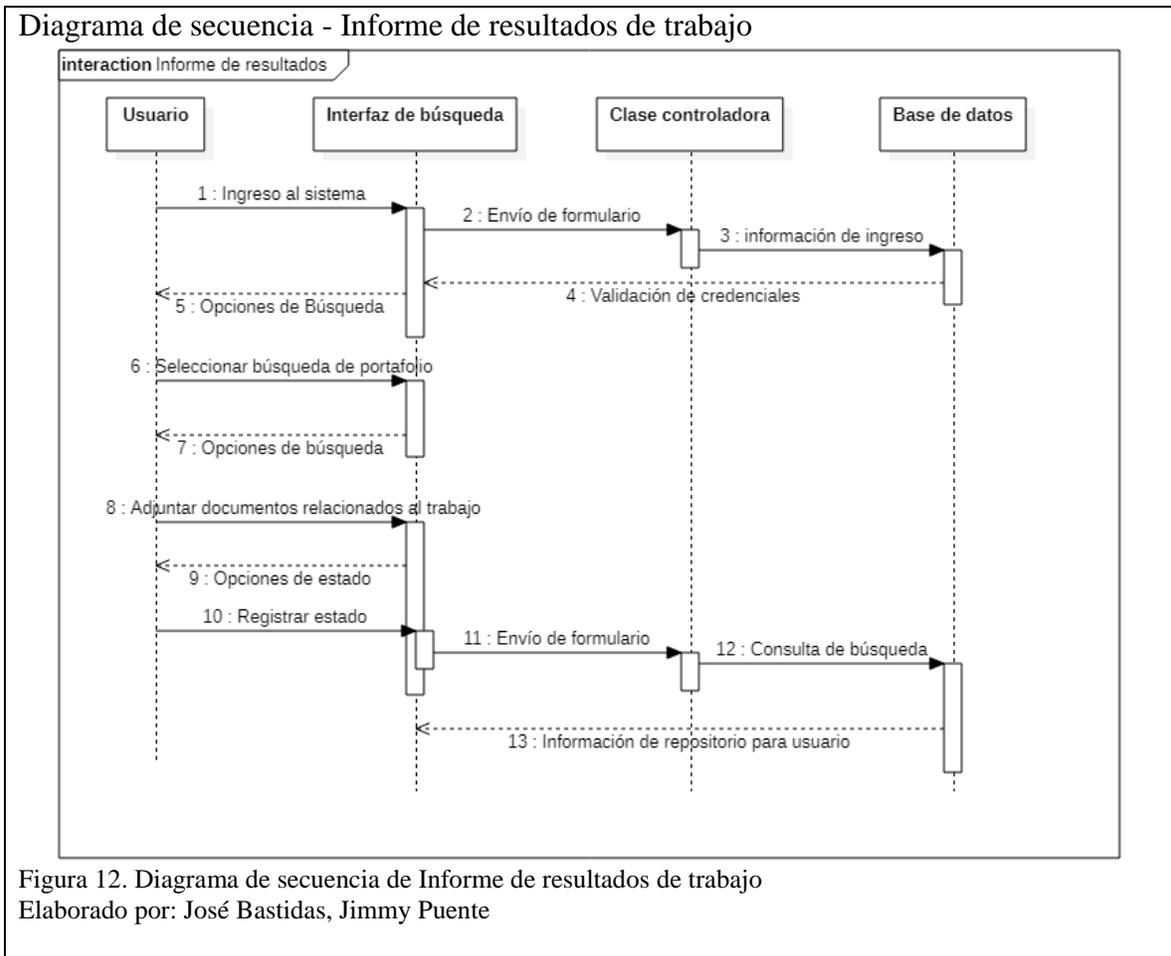


Figura 11. Caso de uso Creación de portafolio de trabajos en pozos  
Elaborado por: José Bastidas, Jimmy Puente

#### 2.2.5.4. Diagrama de secuencia UML

Se presenta el diagrama de secuencia para Informe de resultados de trabajo en el cuál se describe la interacción entre el usuario y el sistema.



## CAPÍTULO 3

### 3.1. Descripción de Clases (con código relevante)

#### 3.1.1. Explicación sobre clases y código de aplicativo web

A continuación, se detallan las secciones de código más importantes para los principales módulos dentro la aplicación.

##### 3.1.1.1. Encapsulación de métodos Service

Esa porción de código esconde el código fuente de la clase *InformeOperadoraService* a través de una interfaz, lo que hace que solamente sea visible los métodos, mas no su estructura interna.

Método de Informe Operadora Service

```
public interface IInformeOperadoraService {
    Collection<InformeOperadora> informeOperadoraList();

    void CrearInformeTrabajos(InformeOperadora informeOperadora) throws Exception;

    Collection<InformeOperadora> findById(Long id);

    InformeOperadora save(String path, InformeOperadoraVO informeOperadoraVO);

    void transActualizarInformeOperadora(InformeOperadora informeOperadora) throws
Exception;
}
```

Figura 13. Método Service de Informe Operadora  
Elaborado por: José Bastidas, Jimmy Puente

##### 3.1.1.2. Ejemplo de una definición de base de datos

Dentro de la arquitectura de Spring es necesario crear una clase específica que es usada para modelar una entidad de la base de datos dentro de Spring.

La estructura con la que se crea esta clase puede ser modificada mediante anotaciones o decoradores que permiten definir secciones específicas de la entidad, los atributos de la

entidad son representados como variables dentro de la clase de Spring y son identificados mediante el decorador: @Column, en dicho decorador también se debe definir el nombre o identificador de dicho campo dentro de la entidad.

El decorador que es usado para identificar a la clase como entidad es el decorador: @Entity, mientras que para la definición de identificador de la entidad se usa la anotación: @Table, en dicha anotación se debe establecer el nombre de dicha entidad.

Es necesario importar dentro de la clase el conjunto de librerías necesarias para establecer los decoradores de cada elemento. A continuación, se muestra una sección de código propia de la definición de una entidad dentro de una clase de Spring.

#### Ejemplo de definición de base de datos

```
import java.util.Date;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="TP_VERIFICACION_FECHAS")
public class VerificacionFechas {

    @Id
    @Column(name="CODIGO_VERIFICACION_FECHAS")
    private String codigoVerificacionFechas;

    @Column(name="CODIGO_PORTAFOLIO")
    private Long codigoPortafolio;
```

Figura 14. Definición de base de datos

Elaborado por: José Bastidas, Jimmy Puente

### 3.1.1.3. Listar producción

Esta parte del código permite listar un conjunto de datos en base a criterios específicos, en este caso condicionando que la fecha de producción sea menor a un parámetro, el código de verificación producción que sea igual a un parámetro, que los resultados se ordenen de

manera descendente y que el número máximo de registros que devuelva sea igual a un parámetro dado.

#### Método para Listar Producción

```
@Override
public Collection<Produccion> ListarProduccionInternaAntes(Date fechaAntes, int
resgistros, Long codVerfProduccion) {

    Criteria insertarDatos =
sessionFactory.openSession().createCriteria(Produccion.class, "produccion");
insertarDatos.add(Restrictions.lt("produccion.fecha_registro", fechaAntes));
insertarDatos.add(Restrictions.eq("produccion.CodVerfProd",
codVerfProduccion));
insertarDatos.addOrder(Order.desc("fecha_registro"));
insertarDatos.setMaxResults(resgistros);
return insertarDatos.list();
}
```

Figura 15. Funcionalidad para listar producción  
Elaborado por: José Bastidas, Jimmy Puente

#### 3.1.1.4. Guardar datos de producción

Esta parte del código permite almacenar en la base de datos un objeto que previamente se lo ha creado dentro del paquete Model. El almacenamiento se lo hace a través del objeto sessionFactory de Hibernate, que es un Framework de Java, el cual permite el mapeo de objetos. El método session.save es aquel que permite almacenar los datos, en este caso de la clase producción.

#### Método para guardar datos de producción

```
@Override
public void GuardarDatosAntes(Produccion produccion) throws Exception {
    try {

        Transaction transaction = null;
        Session session = sessionFactory.openSession();
        transaction = session.beginTransaction();
        transaction.begin();
        session.save(produccion);
        transaction.commit();
    } catch (Exception e) {
        throw new Exception("Error al crear el Informe", e);
    }
}
```

Figura 16. Funcionalidad para guardar datos de producción  
Elaborado por: José Bastidas, Jimmy Puente

### 3.1.1.5. Actualización de datos de Verificación de Producción

Esta parte del código permite actualizar los datos de la tabla VerificacionProduccion en la base de datos un objeto que previamente se lo ha creado dentro del paquete Model. El almacenamiento se lo hace a través del objeto sessionFactory de Hibernate, que es un Framework de Java, el cual permite el mapeo de objetos. El método session.update es aquel que permite actualizar los datos, en este caso de la clase VerificacionProduccion.

Este método utiliza también try y catch, lo que hace que el programa bote excepciones en el caso de que suceda algún error.

Método de Actualización de datos de producción

```
@Override
public void Actualizar(VerificacionProduccion verificacionProduccion) throws
Exception {
    try {
        Transaction transaction = null;
        Session session = sessionFactory.openSession();
        transaction = session.beginTransaction();
        transaction.begin();
        session.update(verificacionProduccion);
        transaction.commit();
    } catch (Exception e) {
        throw new Exception("Error al crear el Informe", e);
    }
}
```

Figura 17. Método de actualización de datos de producción  
Elaborado por: José Bastidas, Jimmy Puente

### 3.1.1.6. Replicar datos de producción

Esta porción de código es la encargada de replicar los datos de producción de un esquema a otro. Lo primero que hace este método es llamar al método que realiza la búsqueda de datos

en base a ciertos criterios. En el caso que la consulta devuelva valores, se procede a añadir los datos que se han devuelto a la clase producción y finalmente los almacena en la tabla producción del esquema PERF de la base de datos. Este proceso se lo debe llevar a cabo porque muchas veces la información almacenada en otros esquemas de la base de datos puede ser alterada, lo cual podría generar problemas una vez que se haya generado informes de Producción.

#### Método para replicar datos de Producción

```
@Override
public Collection<Produccion> ListarProduccionInterna(Date fechaAntes, int registros,
String pozo, Long verfProd)

    throws Exception {

    Collection<Produccion> ProduccionList =
    verificarProduccionDAO.ListarProduccionInternaAntes(fechaAntes,
        registros, verfProd);

    Collection<Produccion> listarPorducccionList = new ArrayList<>();

    if (ProduccionList.isEmpty()) {
        Collection<ProduccionDiaria> ProduccionDiariaList = verificarProduccionDAO
            .ListarProduccionAntes3(fechaAntes, registros, pozo);
        Long tipo = new Long(1);

        Produccion produccion = new Produccion();

        for (ProduccionDiaria produccionDiaria : ProduccionDiariaList) {
            produccion.setApi(produccionDiaria.getApi());
            produccion.setBfpd(produccionDiaria.getBfpd());
            produccion.setBppd(produccionDiaria.getBppd());
            produccion.setBsw(produccionDiaria.getBsw());
            produccion.setCodVerfProd(verfProd);

            produccion.setFecha_registro(produccionDiaria.getF_produccion());

            produccion.setIdProduccion(produccionDiaria.getCodProduccion());
            produccion.setTipo(tipo);

            produccion.setSalinidad_ppm(produccionDiaria.getSalinidad_ppm());
            produccion.setEquipo(produccionDiaria.getEquipo());
            produccion.setYacimiento(produccionDiaria.getYacimiento());

            try {
                verificarProduccionDAO.GuardarDatosAntes(produccion);
            }

            catch (Exception e) {
                throw new Exception(e);
            }
        }
        ProduccionDiariaList =
    verificarProduccionDAO.ListarProduccionAntes3(fechaAntes, registros, pozo);
```

```
}
```

Figura 18. Funcionalidad para replicación de datos de producción  
Elaborado por: José Bastidas, Jimmy Puente

### 3.1.1.7. Encapsulación métodos DAO

Esta porción del código permite encapsular todos los métodos que se encuentren dentro de la clase `InformeOperadoraDAO`, con el objetivo de ocultar el estado de los métodos que se encuentren dentro de los mismos.

#### Método de Encapsulación de métodos DAO

```
public interface IInformeOperadoraDAO {  
    Collection<InformeOperadora> informeOperadoraList();  
    Collection<InformeOperadora> findById(Long id);  
    void CrearInformeTrabajos(InformeOperadora informeOperadora) throws Exception;  
    void transActualizarInformeOperadora(InformeOperadora informeoperadora) throws  
Exception;  
}
```

Figura 19. Funcionalidad para encapsulación de métodos DAO  
Elaborado por: José Bastidas, Jimmy Puente

### 3.1.1.8. Generación de datos

Esta porción de código devuelve todos los valores existentes de la tabla `InformeOperadora`, utilizando el método `EntityManager`. El método `InformeOperadoraList`, devuelve un `arraylist` que son los resultados de la búsqueda en la base de datos.

#### Método para Generación de datos

```
@Component  
public class InformeOperadoraDAO implements IInformeOperadoraDAO{  
    @Autowired  
    private IInformeOperadoraDAO informeOperadoraDAO;  
    @Autowired
```

```

private EntityManagerFactory entityManagerFactory;

@Override
public Collection<InformeOperadora> informeOperadoraList() {
    EntityManager em = entityManagerFactory.createEntityManager();
    Query query = em.createQuery("SELECT tp FROM InformeOperadora tp");
    Collection<InformeOperadora> resultList = query.getResultList();
    em.close();
    return resultList;
}

```

Figura 20. Funcionalidad para generación de datos  
Elaborado por: José Bastidas, Jimmy Puente

### 3.1.1.9. Método Data Access Object para búsqueda de portafolio

Para el método de búsqueda de portafolio se establece un método que interactúa de forma activa con el repositorio de datos mediante sentencias SQL para manejo de información, el cual retorna una colección de datos en base a la petición realizada por la aplicación. En el método mostrado en esta sección se muestra una funcionalidad de código que devuelve todos los datos correspondientes a pozos registrados dentro de los datos almacenados en la base de datos de la aplicación.

Los datos obtenidos a partir de este método son retornados mediante un objeto de tipo resultList.

#### Método DAO para búsqueda de portafolio

```

@Override
public Collection<EEPozo> findPozoList() {
    EntityManager em = entityManagerFactory.createEntityManager();
    Query query = em.createQuery("SELECT pozo FROM EEPozo pozo");
    Collection<EEPozo> resultList = query.getResultList();
    em.close();
    return resultList;
}

```

Figura 21. Funcionalidad para búsqueda de portafolio  
Elaborado por: José Bastidas, Jimmy Puente

### 3.1.1.10. Parámetros de conexión con base de datos desde Spring

Para un correcto funcionamiento y comunicación de la aplicación Spring con el repositorio de datos es necesario establecer parámetros de rutas y credenciales con la finalidad de acceder a los datos almacenados. La configuración necesaria es establecida dentro las propiedades de la aplicación Spring y además dentro de las mencionadas propiedades de la aplicación se pueden establecer parámetros como: driver de comunicación que será usado, el tiempo de espera antes que se agote la conexión, entre otros parámetros.

#### Método de parámetros de conexión

```
# Parámetros de conexión con repositorio de datos
spring.datasource.url=jdbc:oracle:thin:@database-12.rds.amazonaws.com:1521/ORCL
spring.datasource.username=admin
spring.datasource.password=Passw

# Driver de comunicación de la aplicación
spring.datasource.driver.class=oracle.jdbc.driver.OracleDriver

# Parámetros de tiempo de espera de conexión
spring.datasource.hikari.connectionTimeout=20000
spring.datasource.hikari.maximumPoolSize=5
spring.datasource.hikari.poolName=HikariPoolZZZ
logging.level.org.hibernate.SQL=debug
```

Figura 22. Funcionalidad para conexión a repositorio de datos  
Elaborado por: José Bastidas, Jimmy Puente

### 3.1.1.11. Módulo Controlador

Dentro de la clase controladora se establecen los métodos a través de los cuales la aplicación enviará los datos a través de sus servicios, además se puede establecer los valores correspondientes a la URL correspondiente a la función, la forma de envío de datos, GET o POST, por ejemplo, y las cabeceras de la petición correspondiente. Dentro de un método del controlador se establecen los objetos que acceden a métodos que devuelven valores almacenados en el repositorio de datos. Un controlador puede ser visto como una URL a través de la cual se conecta y envía datos o peticiones a la aplicación.

### Método de Módulo Controlador

```
@Autowired
private IBuscarPortafolioServicio buscarPortafolioServicio;

@RequestMapping(value = "/findPortafolioList", method = RequestMethod.POST,
headers = "Accept=application/json;charset=UTF-8")
@ResponseBody
ResponseEntity<Collection<PortafolioVO>> findPortafolioList(@RequestBody
BusquedaParametros bParam) {
    HttpHeaders responseHeaders = new HttpHeaders();

    Collection<PortafolioVO> portafolioList =
buscarPortafolioServicio.findPortafolioList(bParam);

    return new ResponseEntity<>(portafolioList, responseHeaders,
HttpStatus.OK);
}
```

Figura 23. Funcionalidad para replicación de datos de producción  
Elaborado por: José Bastidas, Jimmy Puente

## 3.2. Implementación

Al finalizar la construcción de la aplicación se procedió al despliegue del aplicativo en un servidor de la Agencia de Regulación y Control Hidrocarburífero. Dentro de los detalles referentes a la implementación se presentan las siguientes características.

### 3.2.1. Características de Servidor físico

A continuación, se detallan las características del servidor en el cuál la aplicación se encuentra desplegada.

Tabla 26. Características de servidor

Tipo de Software	Descripción
Sistema Operativo	Linux
Memoria RAM	16 GB
Espacio de almacenamiento	100 GB
Dirección IP de máquina	7.3.9

Nota: Se presentan las características de servidor físico  
Elaborado por: José Bastidas, Jimmy Puente

### 3.2.2. Software utilizado en la aplicación

A continuación, se detallan las características del software utilizado en el desarrollo de la aplicación para la ARCH.

Tabla 27. Software utilizado

Tipo de Software	Descripción
Oracle 11g Enterprise Edition	11.2.0.4.0
Spring Tools	4
JBoss	7.2
Angular CLI	7.3.9
OJDBC	6.1.0.1

Nota: Se presentan las características de software utilizado

Elaborado por: José Bastidas, Jimmy Puente

### 3.3. Base de datos

La base de datos ha sido desarrollada en Oracle, utilizando la versión “Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production”. Para el modelado de la base de datos, se utilizó Power Designer, donde se crearon los modelos conceptuales y lógicos. El controlador utilizado para la conexión con el componente backend es Ojdbc8.

#### 3.3.1. Patrones para especificaciones de base de datos

Los estándares y especificaciones para las denominaciones usadas en la base de datos son detallados a continuación.

El estándar de la base de datos se realizó en base a las especificaciones de la ARCH en cuanto a su estructura de modelamiento que ya tienen pactado internamente en el departamento de Innovación y Tecnología de dicha agencia. Todas las tablas se encuentran creadas en el esquema PERF.

### **3.3.1.1. Definiciones para tablas de base de datos**

Para la creación de tablas se ha mantenido un formato unificado bajo el cual se puede identificar y catalogar de manera óptima las entidades de la base de datos:

1. Cada tabla que pertenece al sistema debe llevar el prefijo TP\_
2. Es necesario el uso de nombres que identifiquen las entidades.
3. Las tablas deben estar compuestas por un máximo de 30 caracteres
4. Existen tablas que tienen relaciones con tablas de otros esquemas de la base de datos, en donde se utiliza datos que son externos a la aplicación que se desarrolla.
5. Los nombres de las tablas deben ser escritos en mayúsculas.
6. A continuación, se muestran varios ejemplos.

#### **Ejemplos:**

1. TP\_CATEGORIA
2. TP\_ESTADO
3. TP\_INFORME\_OPERADORA
4. TP\_SEGUIMIENTO\_NOVEDAD
5. TP\_PRODUCION\_REINYECTOR
6. TP\_VERIFICACION\_INYECTOR
7. TP\_VERIFICACION\_TRABAJO

### **3.3.1.2. Campos de base de datos**

Los campos que se encuentran representados dentro de las tablas de la base de datos mantienen un formato con el cual se los puede identificar de manera más clara.

1. Los nombres de los campos deben estar escritos en mayúsculas.
2. La cantidad máxima de caracteres que se debe usar como nombre de cada campo debe ser de 30.
3. El número máximo de caracteres que tendrá el nombre de cada campo es de 35.
4. No se permite el uso de caracteres especiales o tildes.
5. A continuación, se muestran varios ejemplos:

#### **Ejemplos:**

1. CODIGO\_PRODUCION\_REINYECTOR
2. RECOMENDACIONES
3. ID\_USUARIO

### **3.3.1.3. Claves Primarias y Llaves Foráneas**

1. Las claves primarias son autoincrementales.
2. Los identificadores deben contener el prefijo CODIGO\_
3. Las claves foráneas deben incluir el mismo nombre de la clave primaria.
4. Las claves primarias y claves foráneas deben ser tipo Number, con una longitud máxima de 255 caracteres.

A continuación, se citan algunos ejemplos:

### **Ejemplos:**

1. TP\_PRODUCUCCION\_REINYECTOR
2. CODIGO\_PORTAFOLIO
3. CODIGO\_VERIFICACION\_TRABAJO

#### **3.3.2. Diagrama Conceptual de base de datos**

En el siguiente apartado se presenta el diagrama conceptual de base de datos en el cual se describe la representación “entidad – relación”, mostrando las relaciones y las entidades que componen la base de datos.

Ver documento anexo.

#### **3.3.3. Diagrama físico de base de datos**

Un diagrama físico de base datos permite explicar y detallar las relaciones y las tablas generadas en base al modelo conceptual de la base de datos. En este diagrama se detallan los atributos de las entidades y sus respectivas relaciones con las demás tablas.

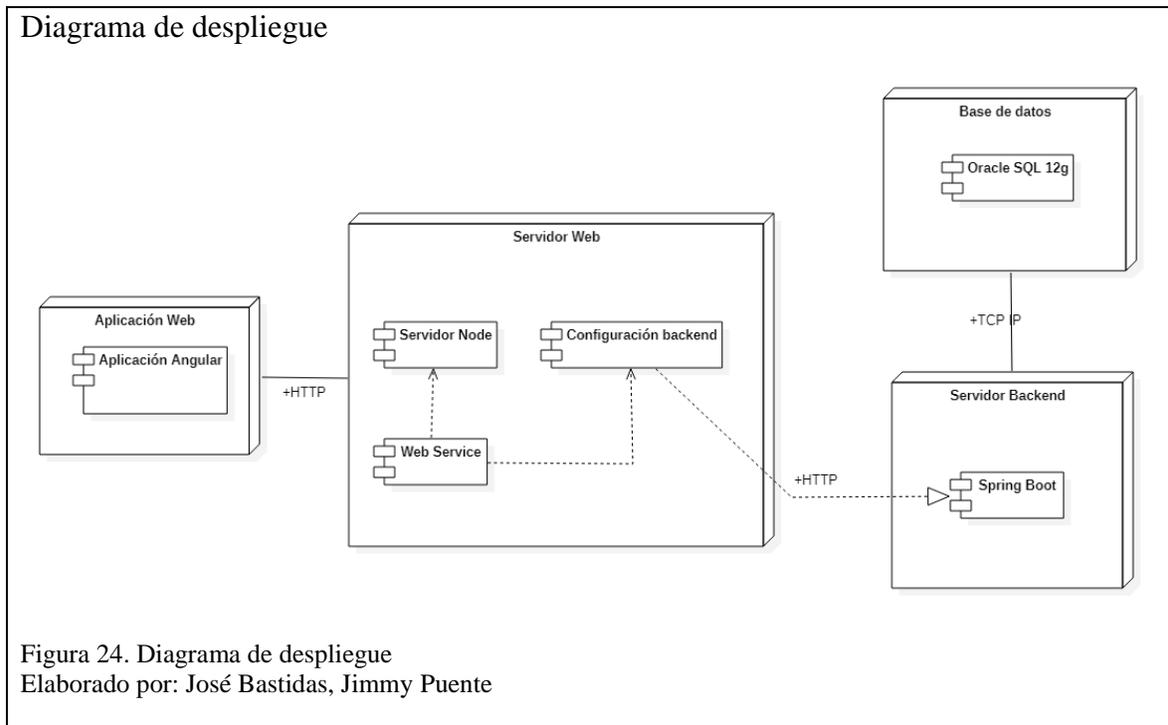
Ver documento anexo.

#### **3.4. Diagramas de aplicación**

Como parte del desarrollo realizado para el aplicativo web se presentan esquemas y diagramas en los cuales se muestran las relaciones presentes entre los distintos componentes que contiene la aplicación.

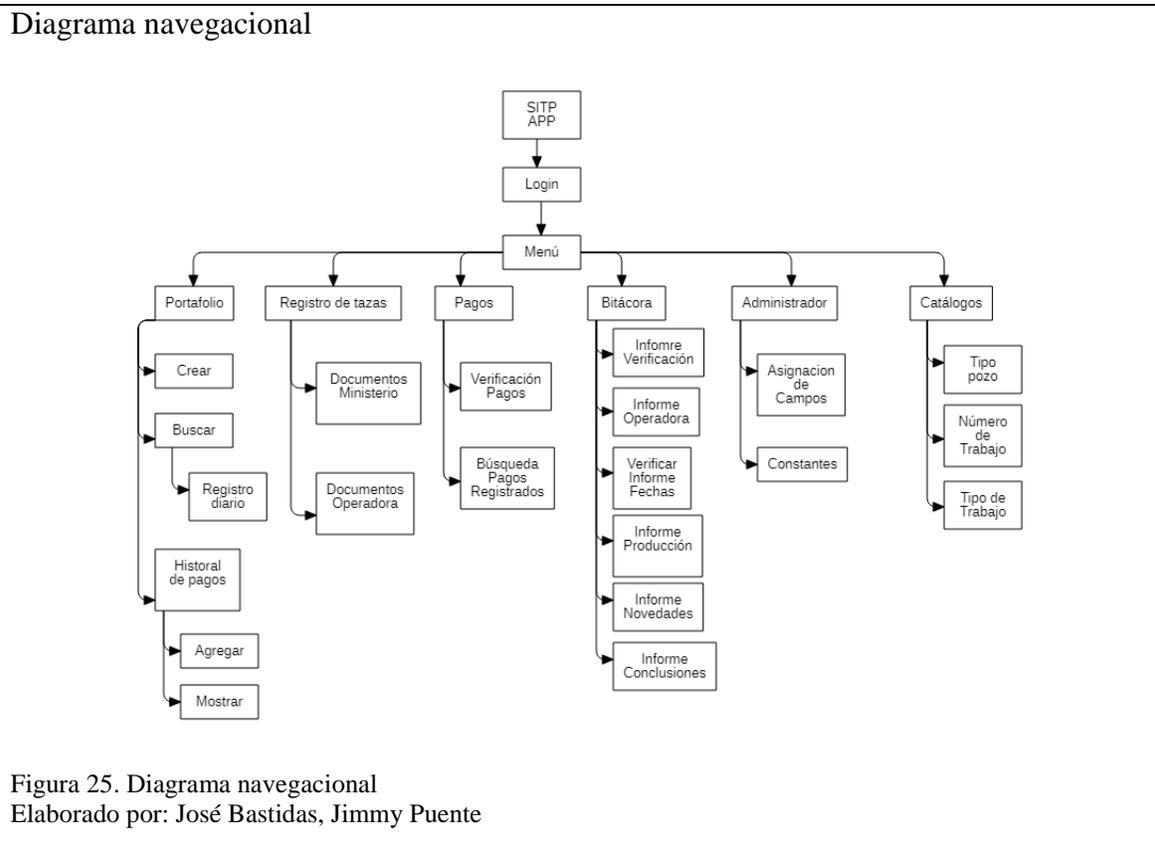
### 3.4.1. Diagrama de despliegue

El diagrama de despliegue permite exponer los componentes de la aplicación que conforman la arquitectura necesaria para un óptimo funcionamiento, y de la misma manera, se presenta la interacción del usuario con dichos componentes.



### 3.4.2. Diagrama de navegación

Con la finalidad de dar a conocer la estructura del aplicativo web se presenta el diagrama de navegación, en el cual se muestra la organización del sitio web.



### 3.5. Plan de Pruebas

#### 3.5.1. Pruebas Unitarias

En el apartado de pruebas unitarias se evaluaron módulos dentro de la aplicación para comprobar su correcto funcionamiento, como parte de los módulos en los cuales se realizaron pruebas están los módulos de: Seguimiento de novedades, verificación de producción, verificación de fechas, entre otros. Para la ejecución de las pruebas unitarias se utilizó la herramienta “JUnit”, la cual puede ser integrada a Spring mediante la adición de un archivo

de extensión “jar” dentro de sus librerías. En el siguiente apartado se detallan los resultados al momento de ejecución de las pruebas.

Para el módulo de Informe de operadora, en el cual se han tomado tres funcionalidades, se tienen los siguientes resultados:

Tabla 28. Objeto y atributos Informe Operadora

Objeto	Atributos
InformeOperadora	Cod_informe NumeroSGC NumeroOficio AnexoDoc FechaOficio FechaArch Resultado CostoReal Estado FechaActualizacion
Portafolio	CodPortafolio

Nota: Objetos y atributos de Informe Operadora  
Elaborado por: José Bastidas, Jimmy Puente

Tabla 29. Resultados obtenidos de pruebas unitarias de Informe Operadora

Funcionalidad	Tiempo (ms)	Tamaño (Kb)	Estado
Listar Datos	14,44	141	OK
Crear Informe	13,8	572	OK
Actualizar Informe	12,6	578	OK

Nota: Resultados de pruebas unitarias de Informe Operadora  
Elaborado por: José Bastidas, Jimmy Puente

Para el módulo de Conclusiones y Recomendaciones, se tienen tres funcionalidades, y se han obtenido los siguientes resultados:

Tabla 30. Objeto y atributos Conclusiones y Recomendaciones

Objeto	Atributos
ConclusionRecomendacion	Cod_ConclusionRecomendacion Conclusion Recomendacion Estado FechaActualizacion IdUsuario
Portafolio	CodPortafolio

Nota: Objetos y atributos de Conclusiones y Recomendaciones  
Elaborado por: José Bastidas, Jimmy Punte

Tabla 31. Resultados obtenidos de pruebas unitarias de Conclusiones y Recomendaciones

Funcionalidad	Tiempo (ms)	Tamaño (Kb)	Estado
Obtener Datos	39,2	156	OK
Crear Conclusión	67	425	OK
Actualizar	18,9	599	OK

Nota: Resultados de pruebas unitarias de Conclusiones y Recomendaciones  
Elaborado por: José Bastidas, Jimmy Punte

Para el módulo de Seguimiento de novedad, en el cual se han tomado tres funcionalidades, se tienen los siguientes resultados

Tabla 32. Objeto y atributos Seguimiento de novedad

Objeto	Atributos
SeguimientoNovedad	CodSeguimientoNovedad CodTipoNovedad NumeroDocumento Documento EntradaSalida Asunto TipoNovedad Observacion Estado FechaActualizacion

	IdUsuario
Novedad	CodNovedad

Nota: Objetos y atributos de Seguimiento de novedad

Elaborado por: José Bastidas, Jimmy Puente

Tabla 33. Resultados obtenidos de pruebas unitarias de Seguimiento de novedad

Funcionalidad	Tiempo (ms)	Tamaño (Kb)	Estado
Creación de novedad	9,6	630	OK
Listar Datos	34,1	141	OK
Actualizar de datos	13,9	428	OK

Nota: Resultados de pruebas unitarias de Seguimiento de novedad

Elaborado por: José Bastidas, Jimmy Puente

Para el módulo de Verificación de producción, se ha hecho un análisis de seis funcionalidades y se tienen los siguientes resultados

Tabla 34. Objeto y atributos Verificación de producción

Objeto	Atributos
VerificarProduccion	CodVerificacionProduccion NumeroRegistros PorcentajeControl Valoracion Justificado Estado FechaActualizacion ValorAntes ValorDespues
Producción	CodProduccion Tipo Yacimiento FechaRegistro BPPD BFPD BSW API Salinidad_PPM Equipo

	RangoMinMax IdProduccion
Portafolio	Cod. Portafolio

Nota: Objetos y atributos de Verificación de producción  
Elaborado por: José Bastidas, Jimmy Puentes

Tabla 35. Resultados obtenidos de pruebas unitarias de Verificación de producción

Funcionalidad	Tiempo	Tamaño	Estado
Crear Verificación Producción	11	640	OK
Actualizar Verificación Producción	10	633	OK
Obtener Datos	12,6	578	OK
Listar Datos	13,33	156	OK
Buscar antes de obtener datos	12	738	OK
Buscar después de obtener datos	12,6	740	OK

Nota: Resultados de pruebas unitarias de Verificación de producción  
Elaborado por: José Bastidas, Jimmy Puentes

Para el módulo de Verificación de Reinyector, en el cual se han tomado cuatro funcionalidades y se han obtenido los siguientes resultados

Tabla 36. Objeto y atributos Verificación de Reinyector

Objeto	Atributos
VerificarReinyeccion	CodVerificacionReinyeccion NumeroRegistros PorcentajeControl Valoracion Justificado Estado FechaActualizacion ValorAntes ValorDespues
ProduccionReinyector	CodProduccionReinyector CodVerificacionReinyector Fecha BAIPD IdProduccionReinyector
Portafolio	Cod. Portafolio

Nota: Objetos y atributos de Verificación de Reinyector

Elaborado por: José Bastidas, Jimmy Puente

Tabla 37. Resultados obtenidos de pruebas unitarias de Verificación de Reinector

Funcionalidad	Tiempo ms	Tamaño Kb	Estado
Crear Reinector	25	636	OK
Actualizar reinyector	27,4	141	OK
Buscar Antes Trabajo	10,5	735	OK
Buscar Después Trabajo	8,5	747	OK

Nota: Resultados de pruebas unitarias de Verificación de Reinector

Elaborado por: José Bastidas, Jimmy Puente

Para el módulo de Verificar Fechas, en el cual se han tomado cuatro funcionalidades, se tienen los siguientes resultados

Tabla 38. Objeto y atributos Verificar Fechas

Objeto	Atributos
VerificarFechas	CodVerificacionFechas Valoración Justificado Observación FechaActualizacion Estado IdUsuario
Portafolio	CodPortafolio

Nota: Objetos y atributos de Verificar Fechas

Elaborado por: José Bastidas, Jimmy Puente

Tabla 39. Resultados obtenidos de pruebas unitarias de Verificar Fechas

Funcionalidad	Tiempo ms	Tamaño Kb	Estado
Listar Datos	14,444	141	OK
Crear Informe Verificación Fechas	13,8	572	OK

Actualizar Informe Verificación Fechas	12,6	578	OK
Obtener Fecha Máxima	12,6	578	OK

Nota: Resultados de pruebas unitarias de Verificar Fechas  
 Elaborado por: José Bastidas, Jimmy Puente

### 3.5.2. Pruebas de Carga y Stress

Las pruebas de carga y stress dentro de la aplicación fueron realizadas mediante el uso de la herramienta de tipo open source “JMeter”. El funcionamiento de esta herramienta está basado en el envío de peticiones a distintas URL de la aplicación sobre la cual se realizarán las pruebas de stress. Al configurar “JMeter” se permite definir parámetros como: número de peticiones, tiempo en el que se ejecutan las peticiones, tipos de petición, que puede ser GET o POST, protocolo de URL, entre otras opciones.

Otra funcionalidad que ofrece la herramienta es la posibilidad de visualizar comparativas de resultados de distintas pruebas, además de graficas que indican el comportamiento de la aplicación en el período que se realizó las pruebas de carga, mostrando porcentajes de error y cantidad de paquetes enviados y recibidos.

Se evaluarán tres escenarios de manera progresiva dentro de las pruebas de carga, en el primer escenario en un período de tiempo de 1 segundo se enviarán 25 peticiones, en el segundo escenario en un periodo de tiempo de 1 segundo se enviarán 50 peticiones y en el tercer escenario en un periodo de 1 segundo se enviarán 75 peticiones. Cabe mencionar que todas las peticiones que se envían son concurrentes.

#### 3.5.2.1. Escenario de 25 peticiones

Tabla 40. Prueba de carga para Buscar portafolio 25 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.	Rendimiento
Buscar Portafolio	HTTP GET	25	31	498	203	22,32	26,43	15,6/seg

Nota: Resultados de escenario de 25 peticiones para Buscar portafolio

Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP GET se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,031 segundos y un tiempo máximo de 0,498, el tiempo promedio que se obtuvo fue de 0,203 segundos. Durante esta prueba se enviaron 22,32 KB y se recibieron 26,43 KB para las peticiones realizadas.

Tabla 41. Prueba de carga para Seguimiento Novedad 25 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.	Rendimiento
Seguimiento Novedad	HTTP POST	25	24	375	155	18,73	22,75	21,3/seg

Nota: Resultados de escenario de 25 peticiones para Seguimiento Novedad

Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de seguimiento de novedad se realizaron peticiones de tipo HTTP POST se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,024 segundos y un tiempo máximo de 0,375, el tiempo promedio que se obtuvo fue de 0,155 segundos. Durante esta prueba se enviaron 18,73 KB y se recibieron 22,75 KB para las peticiones realizadas.

Tabla 42. Prueba de carga para Verificación Producción 25 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.	Rendimiento
Verificación Producción	HTTP POST	25	89	823	516	40,78	55,43	13,44/seg

Nota: Resultados de escenario de 25 peticiones para Verificación Producción

Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP POST se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,089 segundos y un tiempo máximo de 0,823, el tiempo promedio que se obtuvo fue de 0,516 segundos. Durante esta prueba se enviaron 40,78 KB y se recibieron 55,43 KB para las peticiones realizadas.

Tabla 43. Prueba de carga para Verificación Inyector 25 peticiones

Módulo	Tipo	Peticiones	T mín	T máx.	Promedio (ms)	Bytes env.	Bytes recib.	Rendimiento
Verificar Inyector	HTTP GET	25	110	1054	737	99,27	108,96	14,313/seg

Nota: Resultados de escenario de 25 peticiones para Verificación Inyector  
Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP GET se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,110 segundos y un tiempo máximo de 1,054, el tiempo promedio que se obtuvo fue de 0,737 segundos. Durante esta prueba se enviaron 99,27 KB y se recibieron 108,96 KB para las peticiones realizadas.

Tabla 44. Prueba de carga para Verificación Fechas 25 peticiones

Módulo	Tipo	Peticiones	T mín	T máx.	Promedio (ms)	Bytes env.	Bytes recib.	Rendimiento
Verificar Fechas	HTTP POST	25	38	256	194	35,28	42,11	17,81/seg

Nota: Resultados de escenario de 25 peticiones para Verificación Fechas  
Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP POST se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,038 segundos y un tiempo máximo de 0,256, el tiempo promedio que se obtuvo fue de

0,194 segundos. Durante esta prueba se enviaron 35,28 KB y se recibieron 42,11 KB para las peticiones realizadas.

### 3.5.2.2. Escenario de 50 peticiones

Tabla 45. Prueba de carga para Buscar portafolio 50 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.	Rendimiento
Buscar Portafolio	HTTP GET	50	26	512	289	23,61	31,44	27,29/seg

Nota: Resultados de escenario de 50 peticiones para Buscar portafolio

Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP GET se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,026 segundos y un tiempo máximo de 0,512, el tiempo promedio que se obtuvo fue de 0,289 segundos. Durante esta prueba se enviaron 23,61 KB y se recibieron 31,44 KB para las peticiones realizadas.

Tabla 46. Prueba de carga para Seguimiento Novedad 50 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.	Rendimiento
Seguimiento Novedad	HTTP POST	50	42	694	376	22,59	30,51	20,39/seg

Nota: Resultados de escenario de 50 peticiones para Seguimiento Novedad

Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP POST se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,042 segundos y un tiempo máximo de 0,694, el tiempo promedio que se obtuvo fue de 0,376 segundos. Durante esta prueba se enviaron 22,59 KB y se recibieron 30,51 KB para las peticiones realizadas.

Tabla 47. Prueba de carga para Verificación Producción 50 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.	Rendimiento
Verificación Producción	HTTP POST	50	105	924	678	50,34	63,13	23,68,/seg

Nota: Resultados de escenario de 50 peticiones para Verificación Producción

Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP POST se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,105 segundos y un tiempo máximo de 0,924, el tiempo promedio que se obtuvo fue de 0,678 segundos. Durante esta prueba se enviaron 50,34 KB y se recibieron 63,13 KB para las peticiones realizadas.

Tabla 48. Prueba de carga para Verificación Inyector 50 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.	Rendimiento
Verificar Inyector	HTTP GET	50	180	1259	847	108,72	120,48	21,06/seg

Nota: Resultados de escenario de 50 peticiones para Verificación Inyector

Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP GET se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,180 segundos y un tiempo máximo de 1,259, el tiempo promedio que se obtuvo fue de 0,847 segundos. Durante esta prueba se enviaron 108,72 KB y se recibieron 120,48 KB para las peticiones realizadas.

Tabla 49. Prueba de carga para Verificación Fechas 50 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.	Rendimiento
Verificar Fechas	HTTP POST	50	47	311	224	54,13	68,26	25,89/seg

Nota: Resultados de escenario de 50 peticiones para Verificación Fechas

Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP POST se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,047 segundos y un tiempo máximo de 0,311, el tiempo promedio que se obtuvo fue de 0,224 segundos. Durante esta prueba se enviaron 54,13 KB y se recibieron 68,26 KB para las peticiones realizadas.

### 3.5.2.3. Escenario de 75 peticiones

Tabla 50. Prueba de carga para Buscar portafolio 75 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.	Rendimiento
Buscar Portafolio	HTTP GET	75	44	687	359	26,71	30,53	40,68/seg

Nota: Resultados de escenario de 75 peticiones para Buscar portafolio  
Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP GET se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,044 segundos y un tiempo máximo de 0,687, el tiempo promedio que se obtuvo fue de 0,359 segundos. Durante esta prueba se enviaron 26,71 KB y se recibieron 30,53 KB para las peticiones realizadas.

Tabla 51. Prueba de carga para Seguimiento Novedad 75 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.	Rendimiento
Seguimiento Novedad	HTTP POST	75	39	760	535	27,63	35,48	39,91/seg

Nota: Resultados de escenario de 75 peticiones para Seguimiento Novedad  
Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP POST se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,039 segundos y un tiempo máximo de 0,760, el tiempo promedio que se obtuvo fue de 0,535 segundos. Durante esta prueba se enviaron 27,63 KB y se recibieron 35,48 KB para las peticiones realizadas.

Tabla 52. Prueba de carga para Verificación Producción 75 peticiones

Módulo	Tipo	Peticiones	T mín	T máx.	Promedio (ms)	Bytes env.	Bytes recib.	Rendimiento
Verificación Producción	HTTP POST	75	129	1042	769	60,20	68,93	45,107./seg

Nota: Resultados de escenario de 75 peticiones para Verificación Producción

Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP POST se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,129 segundos y un tiempo máximo de 1,042, el tiempo promedio que se obtuvo fue de 0,769 segundos. Durante esta prueba se enviaron 60,20 KB y se recibieron 68,93 KB para las peticiones realizadas.

Tabla 53. Prueba de carga para Verificación Inyector 75 peticiones

Módulo	Tipo	Peticiones	T mín	T máx.	Promedio (ms)	Bytes env.	Bytes recib.	Rendimiento
Verificar Inyector	HTTP GET	75	201	1536	964	127,66	149,30	46,71/seg

Nota: Resultados de escenario de 75 peticiones para Verificación Inyector

Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP GET se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,201 segundos y un tiempo máximo de 1,536, el tiempo promedio que se obtuvo fue de 0,964 segundos. Durante esta prueba se enviaron 127,66 KB y se recibieron 149,30 KB para las peticiones realizadas.

Tabla 54. Prueba de carga para Verificación Fechas 75 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.	Rendimiento
Verificar Fechas	HTTP POST	75	59	406	303	70,21	83,17	48,06/seg

Nota: Resultados de escenario de 75 peticiones para Verificación Fechas

Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP POST se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,059 segundos y un tiempo máximo de 0,406, el tiempo promedio que se obtuvo fue de 0,303 segundos. Durante esta prueba se enviaron 70,21 KB y se recibieron 83,17 KB para las peticiones realizadas.

#### 3.5.2.4. Análisis de Resultados

Luego de la realización de las pruebas de carga se puede evidenciar un manejo adecuado de las peticiones por parte del aplicativo y del servidor en el que se encuentra alojado, dentro de los resultados obtenidos se obtuvieron tiempos que no son superiores a un segundo tomando en cuenta que se probaron tres escenarios de 25, 50 y 75 peticiones.

Tabla 55. Comparativa de peticiones de carga

Módulo	25 peticiones promedio (ms)	50 peticiones promedio (ms)	75 peticiones promedio (ms)
Buscar Portafolio	203	289	359
Seguimiento Novedad	155	376	535
Verificación Producción	516	678	769
Verificar Inyector	737	847	964
Verificar Fechas	194	224	303

Nota: Comparativa de promedio de tiempo de peticiones  
Elaborado por: José Bastidas, Jimmy Puento

Al concluir las pruebas de carga en tres escenarios distintos podemos observar que los tiempos de respuesta obtenidos en todas las funcionalidades sometidas a pruebas no superan el segundo en tiempo de respuesta. En un análisis general podemos observar que el mayor tiempo obtenido es 0,964 segundos y el menor tiempo obtenido es 0,155 segundos.

### 3.5.2.5. Pruebas de stress

Para la realización de las pruebas de stress se evaluaron dos escenarios, el primero con 30 peticiones concurrentes en un segundo y en el segundo escenario se evaluaron 600 peticiones concurrentes en un segundo. Las pruebas de stress son realizadas para evaluar el comportamiento de la aplicación ante condiciones que podrían ser consideradas atípicas dentro del desempeño normal de la aplicación.

#### 3.5.2.5.1. Escenario 300 peticiones

Tabla 56. Prueba de stress Buscar Portafolio 300 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.
--------	------	------------	-------	--------	---------------	------------	--------------

Buscar Portafolio	HTTP GET	300	248	8650	4812	28,63	34,14
-------------------	----------	-----	-----	------	------	-------	-------

Nota: Resultados pruebas de stress de 300 peticiones para Buscar Portafolio  
Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP GET se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,248 segundos y un tiempo máximo de 8,650, el tiempo promedio que se obtuvo fue de 4,812 segundos. Durante esta prueba se enviaron 28,63 KB y se recibieron 34,14 KB para las peticiones realizadas.

Tabla 57. Prueba de stress Seguimiento Novedad 300 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.
Seguimiento Novedad	HTTP POST	300	374	9836	5527	21,50	37,66

Nota: Resultados pruebas de stress de 300 peticiones para Seguimiento Novedad  
Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP POST se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,374 segundos y un tiempo máximo de 9,836, el tiempo promedio que se obtuvo fue de 5,527 segundos. Durante esta prueba se enviaron 21,50 KB y se recibieron 37,66 KB para las peticiones realizadas.

Tabla 58. Prueba de stress Verificación Producción 300 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.
Verificación Producción	HTTP POST	300	587	9976	6125	58,17	70,82

Nota: Resultados pruebas de stress de 300 peticiones para Verificación Producción  
Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP POST se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,587 segundos y un tiempo máximo de 9,976, el tiempo promedio que se obtuvo fue de 6,125 segundos. Durante esta prueba se enviaron 58,17 KB y se recibieron 70,82 KB para las peticiones realizadas.

Tabla 59. Prueba de stress Verificar Inyector 300 peticiones

Módulo	Tipo	Peticiones	T mín	T máx.	Promedio (ms)	Bytes env.	Bytes recib.
Verificar Inyector	HTTP GET	300	673	8185	4340	115,49	155,64

Nota: Resultados pruebas de stress de 300 peticiones para Verificar Inyector  
Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP GET se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,673 segundos y un tiempo máximo de 8,185, el tiempo promedio que se obtuvo fue de 4,340 segundos. Durante esta prueba se enviaron 115,49 KB y se recibieron 155,64 KB para las peticiones realizadas.

Tabla 60. Prueba de stress Verificar Fechas 300 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.
Verificar Fechas	HTTP POST	300	402	9638	3741	65,18	85,47

Nota: Resultados pruebas de stress de 300 peticiones para Verificar Fechas  
Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP POST se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,402 segundos y un tiempo máximo de 9,638, el tiempo promedio que se obtuvo fue de 3,741 segundos. Durante esta prueba se enviaron 65,18 KB y se recibieron 85,47 KB para las peticiones realizadas.

### 3.5.2.5.2. Escenario 600 peticiones

Tabla 61. Prueba de stress Buscar Portafolio 600 peticiones

4. Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.
Buscar Portafolio	HTTP GET	600	349	8423	5562	33,41	38,96

Nota: Resultados pruebas de stress de 600 peticiones para Buscar Portafolio

Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP GET se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,349 segundos y un tiempo máximo de 8,423, el tiempo promedio que se obtuvo fue de 5,562 segundos. Durante esta prueba se enviaron 33,41 KB y se recibieron 38,96 KB para las peticiones realizadas.

Tabla 62. Prueba de stress Seguimiento Novedad 600 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.
Seguimiento Novedad	HTTP POST	600	508	7473	5619	18,90	33,43

Nota: Resultados pruebas de stress de 600 peticiones para Seguimiento Novedad

Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP POST se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,508 segundos y un tiempo máximo de 7,473, el tiempo promedio que se obtuvo fue de 5,619 segundos. Durante esta prueba se enviaron 18,90 KB y se recibieron 33,43 KB para las peticiones realizadas.

Tabla 63. Prueba de stress Verificación Producción 600 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.
Verificación Producción	HTTP POST	600	748	8936	6067	64,18	72,41

Nota: Resultados pruebas de stress de 600 peticiones para Verificación Producción

Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP POST se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,748 segundos y un tiempo máximo de 8,936, el tiempo promedio que se obtuvo fue de 6,067 segundos. Durante esta prueba se enviaron 64,18 KB y se recibieron 72,41 KB para las peticiones realizadas.

Tabla 64. Prueba de stress Verificar Inyector 600 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.
Verificar Inyector	HTTP GET	600	819	7369	6210	105,65	137,13

Nota: Resultados pruebas de stress de 600 peticiones para Verificar Inyector  
Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP GET se obtuvo como resultado un tiempo mínimo de respuesta equivalente a 0,819 segundos y un tiempo máximo de 7,369, el tiempo promedio que se obtuvo fue de 6,210 segundos. Durante esta prueba se enviaron 105,65 KB y se recibieron 137,13 KB para las peticiones realizadas.

Tabla 65. Prueba de stress Verificar Fechas 600 peticiones

Módulo	Tipo	Peticiones	T min	T máx.	Promedio (ms)	Bytes env.	Bytes recib.
Verificar Fechas	HTTP POST	600	801	9743	6567	69,70	80,22

Nota: Resultados pruebas de stress de 600 peticiones para Verificar Fechas  
Elaborado por: José Bastidas, Jimmy Puente

Durante la prueba realizada al módulo de búsqueda de portafolio se realizaron peticiones de tipo HTTP POST se obtuvo como resultado un tiempo mínimo de respuesta equivalente a

0,801 segundos y un tiempo máximo de 9,743, el tiempo promedio que se obtuvo fue de 6,567 segundos. Durante esta prueba se enviaron 69,70 KB y se recibieron 80,22 KB para las peticiones realizadas.

### 3.5.2.5.3. Análisis de resultados stress

Al finalizar las pruebas de stress se puede evidenciar que la aplicación es capaz de manejar correctamente las peticiones en casos que pueden ser considerados de tráfico intenso o atípico dentro de la aplicación. Durante las pruebas realizadas se obtuvieron tiempos máximos de repuesta que no superan los 10 segundos en los casos más extremos, donde la aplicación fue exigida de manera que se imiten escenarios que puedan darse durante en escenarios reales.

Tabla 66. Comparativa de peticiones de stress

Módulo	300 peticiones promedio (ms)	600 peticiones (ms)	promedio
Buscar Portafolio	4812	5562	
Seguimiento Novedad	5527	5619	
Verificación Producción	6125	6067	
Verificar Inyector	4340	6210	
Verificar Fechas	3741	6567	

Nota: Resultados comparativo de peticiones de stress  
Elaborado por: José Bastidas, Jimmy Puente

Al realizar las pruebas de stress se evidenció que, en escenarios extremos, en los cuales se probó 300 y 600 peticiones concurrentes los tiempos promedio de respuesta oscilan entre 3 y 6,5 segundos, siendo el menor tiempo igual 3,741 segundos y el mayor tiempo obtenido igual a 6,567 segundos.

## CONCLUSIONES

- El entorno en el cual se ha implementado el entorno del aplicativo ha mostrado una autonomía acorde a las necesidades de la Agencia de Regulación y Control Hidrocarburífero, ya que al realizar pruebas de carga y utilización se han obtenido respuestas acordes a funcionamiento dentro de escenarios reales.
- Se ha desarrollado un sistema integral para la Agencia de Regulación y Control Hidrocarburífero el cual permite llevar a cabo un registro de control de cambios sobre la producción de operadores petroleros, evidenciando una administración óptima de la producción petrolera y una mejoría sobre la forma tradicional de registro, la cual era enteramente manual.
- Durante las fases iniciales de despliegue de la aplicación la plataforma se ha ido acoplando paulatinamente a las actividades realizadas por los funcionarios de la ARCH reemplazando al proceso de registro de producción, que anteriormente se lo hacía de manera manual por el registro mediante el aplicativo evidenciando de esta una funcionalidad cien por ciento correcta.
- Al utilizar el servidor de aplicaciones *Jboss*, se puede evidenciar que la velocidad de respuesta a las peticiones mejora en un 50%, esto se debe a su compatibilidad con aplicaciones en formato *War*, y que al estar basado en Java consume más memoria de máquina.
- El sistema implementado de registro de cambios en la producción diaria de Petróleo ha generado un cambio significativo en la manera que se lleva a cabo este proceso. Esto se debe, a que anteriormente los usuarios se manejaban mediante una plantilla

de Excel teniendo todos los datos de producción dispersos, lo que demoraba en completar el proceso un promedio de 35 minutos, y actualmente con el aplicativo web tarda 8 minutos en generar el mismo proceso de registro de producción diaria y novedades.

- Mediante el correcto análisis de todos los requerimientos, se logró centralizar la información que se utiliza para el registro diario de novedades y asignación del flujo de información de producción diaria de petróleo, y adicionalmente, se puede tener pistas de auditorías de los cambios que se realicen, gracias al acceso mediante roles.
- Los tiempos en las pruebas de carga entre 25, 50 y 75 peticiones, tuvieron respuestas en un tiempo menos a 1 segundo, y al exponer a la aplicación a un escenario extremo (300 a 600 peticiones en un segundo), los tiempos de respuesta no superaron los 10 segundos; lo que se pudo concluir que el sistema es capaz de gestionar adecuadamente altas cargas de tráfico sin que este colapse.

## RECOMENDACIONES

- Para dejar pistas de auditoría, se recomienda respaldar los datos de las tablas desde las tablas de Producción Diaria, que corresponden a otro esquema de la base de datos, hacia el esquema PERF que contiene todas las tablas del módulo que se ha desarrollado. Esto se lo debe hacer porque muchas veces los operadores cambian arbitrariamente los datos de la tabla Producción Diaria, dejando así incongruencias en los resultados de los informes de Producción, Inyección y Reinyección.
- Se recomienda que para futuras modificaciones en el código fuente o para añadir nuevas funcionalidades en el sistema, se los realice cumpliendo las normativas estipuladas en las actas de requerimientos, utilizando el formato adecuado para el uso de variables, creación de campos y métodos.
- Para una futura versión del sistema implementado, se recomienda añadir módulos de reportes para la gestión de incidencias en el sistema.
- Como un futuro de proyecto de Grado, se podría tomar en cuenta mejorar el módulo de seguridades del sistema de la ARCH, analizando las vulnerabilidades que podría tener, e implementar nuevas tecnologías para su desarrollo.
- Se recomienda que todo cambio que se realice en el sistema sea previamente modelado para determinar su factibilidad y evitar redundancias a nivel de módulos, ya que considerando la magnitud del aplicativo podría darse el caso de

funcionalidades con que cumplan el mismo objetivo pero que estén en distintos módulos del sistema.

- Es recomendable realizar capacitaciones a los funcionarios que se vean involucrados en el registro de cambios mediante el software, ya que facilitara la utilización de la plataforma de registro optimizando tiempos y carga de trabajo.
- Es recomendable también dar capacitaciones constantes al personal de infraestructura y tecnología con respecto a las tecnologías usadas para este proyecto, ya que actualmente los técnicos encargados de esta área no conocen las herramientas de Spring que se encarga de crear servicios y de Angular que se utiliza para el consumo de los mismos.
- Se recomienda que para futuros cambios que se realicen, se lo haga utilizando la misma versión de las herramientas, tanto para el *Backend*, (Spring 4) así como para el *Frontend* (Angular 7), ya que, si se lo hace con una versión superior o inferior, pueden suceder inconvenientes con las llamadas a ciertos métodos nativos del lenguaje de programación utilizado.

## **GLOSARIO DE TERMINOS**

MRNNR. - Ministerio de Recursos Naturales No Renovables.

TST. – Trabajo sin torre.

PERF. – Perforaciones en Pozos Petroleros.

ARCH. – Agencia de Regulación y Control Hidrocarburífero.

CYPI. – Competición y pruebas iniciales.

BPPD. - Barriles por Producción Diaria.

BSW. - Contenido de agua pura que trae el crudo de Petróleo.

WAR. - Sus siglas en español significan Aplicación de Archivo Web.

HTTP. - Protocolo de Transferencia de Hipertexto.

STS. - Spring Tool Suits.

KB. - Kilobytes

MS. – Milisegundos.

## LISTA DE REFERENCIAS

### Bibliografía

- Agencia de Regulación y Control Hidrocarburífero. (14 de Mayo de 2011). *Agencia de Regulación y Control Hidrocarburífero - acuerdo 264*. Obtenido de <https://www.controlhidrocarburos.gob.ec/wp-content/uploads/downloads/2016/04/acuerdo-264.pdf>
- Boada, M., & Gómez, J. A. (2018). *El gran libro de Angular*. Barcelona: MARCOMBO S.A.
- Cosmina, I., Harrop, R., Schaefer, Chris, & Ho, C. (2017). *Pro Spring 5*. New York: Apress.
- Deinum, M. (2018). *Spring Boot 2 Recipes: A Problem-Solution Approach*. New York: Apress.
- Freeman, A. (2018). *Pro Angular 6*. Londres: Apress.
- Gutierrez, F. (2019). *Pro Spring Boot 2: An Authoritative Guide to Building Microservices, Web and*. Albuquerque: Apress.
- Haljian, M. (2019). *Progressive Web Apps with Angular*. Oslo: Apress.
- Hinkula, J. (2018). *Hands-On Full Stack Development with Spring Boot 2.0 and React*. Birmingham: Packt Publishing.
- KYOCERA Document Solutions. (19 de Julio de 2017). *Los 6 principales tipos de sistemas de información*. Obtenido de <https://smarterworkspaces.kyocera.es/blog/los-6-principales-tipos-sistemas-informacion/>
- Murray, N., Coury, F., Lerner, A., & Taborda, C. (2018). *Ng-Book: The Complete Guide to Angular*. San Francisco: CreateSpace Independent Publishing Platform.
- Nair, V. (2019). *Practical Domain-Driven Design in Enterprise Java*. New York: Apress.
- Nayrolles, M. (2018). *Angular Design Patterns*. Birmingham: Packt Publishing Ltd.
- Reina, G., & L., F. (8 de Julio de 2012). *METODOLOGIA XP*. Obtenido de <http://ingsoftware072301.obolog.es/metodologia-xp-2012877>
- Reprintsev, A. (2018). *Oracle SQL Revealed*. New York: Apress.
- Seshadri, S. (2018). *Angular: Up and Running*. Sebastopol CA: O'Reilly Media.
- Stellman, A., & Greene, J. (2015). *Learning Agile*. Sebastopol CA: O'Reilly Media.
- Wilken, J. (2018). *Angular in Action*. New York: Manning Publications.
- Wolf, D., & Henley, A. J. (2017). *Java EE Web Application Primer*. New York: Apress.
- Zhang, P. (2018). *Practical Guide to Oracle SQL, T-SQL and MySQL*. Boca Raton: CRC Press.

## **ANEXOS**

Para revisar los anexos de este trabajo, por favor diríjase al CD.