**Universidad Nacional Mayor de San Marcos**
**Universidad del Perú. Decana de América**
Facultad de Ingeniería de Sistemas e Informática
Escuela Profesional de Ingeniería de Sistemas

# Aplicación web para la gestión de banco de datos para las empresas del sector hidrocarburo

**TRABAJO DE SUFICIENCIA PROFESIONAL**

Para optar el Título Profesional de Ingeniero de Sistemas

**AUTOR**

Yul Anthony NUÑEZ MOREANO

**ASESOR**

César Augusto ANGULO CALDERÓN

Lima, Perú

2021




Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

https://creativecommons.org/licenses/by-nc-sa/4.0/

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

**Referencia bibliográfica**

Nuñez, Y. (2021). *Aplicación web para la gestión de banco de datos para las empresas del sector hidrocarburo*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Sistemas]. Repositorio institucional Cybertesis UNMSM.

## Metadatos complementarios autor/ asesor

| | |
|---|---|
| **Datos de autor** | |
| Nombres y apellidos | Yul Anthony Nuñez Moreano |
| Tipo de documento de identidad | DNI |
| Número de documento de identidad | 43389406 |
| URL de ORCID | https://orcid.org/0000-0003-4278-3480 |
| **Datos de asesor** | |
| Nombres y apellidos | César Augusto Angulo Calderón |
| Tipo de documento de identidad | DNI |
| Número de documento de identidad | 32907109 |
| URL de ORCID | https://orcid.org/0000-0002-5123-5008 |
| **Datos del jurado** | |
| **Presidente del jurado** | |
| Nombres y apellidos | Jorge Santiago Pantoja Collantes |
| Tipo de documento | DNI |
| Número de documento de identidad | 06254022 |
| **Miembro del jurado 1** | |
| Nombres y apellidos | Mario Agustín Huapaya Chumpitaz |
| Tipo de documento | DNI |
| Número de documento de identidad | 15386891 |
| **Datos de investigación** | |
| Línea de investigación | No aplica |

| | |
|---|---|
| Grupo de investigación | No aplica |
| Agencia de financiamiento | Financiamiento Propio |
| Ubicación geográfica de la investigación | País: Perú<br>Departamento: Lima<br>Provincia: Lima<br>Distrito: Los Olivos<br>Latitud: -11.994724736601333<br>Longitud: -77.07558354233873 |
| Año o rango de años en que se realizó la investigación | Año 2021 |
| URL de disciplinas OCDE | Ingeniería de sistemas y comunicaciones<br>https://purl.org/pe-repo/ocde/ford#2.02.04 |

***UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS***
***FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA***
***Escuela Profesional de Ingeniería de Sistemas***

***Acta Virtual de Sustentación***
***del Trabajo de Suficiencia Profesional***

*Siendo las 18:05 horas del día 05 de agosto del año 2021, se reunieron virtualmente los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por el Lic. Pantoja Collantes Jorge Santiago (Presidente), Mg. Huapaya Chumpitaz Mario Agustín (Miembro) y el Lic. Angulo Calderón Cesar Augusto (Miembro Asesor), usando la plataforma Meet (https://meet.google.com/pmb-bpem-jjo), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado:* ***"APLICACIÓN WEB PARA LA GESTIÓN DE BANCO DE DATOS PARA LAS EMPRESAS DEL SECTOR HIDROCARBURO ",*** *por el Bachiller* ***Nuñez Moreano Yul Anthony****; para obtener el Título Profesional de Ingeniero de Sistemas.*

*Acto seguido de la exposición del Trabajo de Suficiencia Profesional, el Presidente invitó al Bachiller a dar las respuestas a las preguntas establecidas por los miembros del Jurado.*

*El Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.*

*Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el Bachiller obtuvo la nota de* ***17 DIECISIETE.***

*A continuación el Presidente de Jurados el Lic. Pantoja Collantes Jorge Santiago, declara al Bachiller* ***Ingeniero de Sistemas.***

*Siendo las 19:10 horas, se levantó la sesión.*

UNMSM Firmado digitalmente por PANTOJA COLLANTES Jorge Santiago FAU 20140292202 hard Motivo: Soy el autor del documento Fecha: 05.08.2021 19:31:17 -05:00

_____
***Presidente***
*Lic. Pantoja Collantes Jorge Santiago*

_____
***Miembro***
*Mg. Huapaya Chumpitaz Mario Agustín*

_____
***Miembro Asesor***
*Lic. Angulo Calderón Cesar Augusto*

## Dedicatoria

Dedico este trabajo a mis Padres, a mi Madrina por todo el apoyo, la confianza, la paciencia y el amor brindado que me ha permito llegar a esta etapa profesional.

A mi hija por darme la inspiración y motivación de seguir adelante y de no rendirme.

A mi familia por su apoyo y consejos a lo largo de mi vida.

## Agradecimientos

**Dios:** Por su bendición y por haber puesto en mí camino a las personas correctas.

**A mis Padres:** María Moreano Jimenez, por tu gran amor, por tus atenciones, por preocuparte en mis estudios, por tu confianza en el momento crítico en donde los recursos económicos no eran favorables sin tu apoyo no me hubiese matriculado en ese ciclo semestral en la academia y tampoco hubiese postulado a la universidad. Wilber Nuñez Hurtado, por tus grandes consejos, por todos los principios que me inculcaste, por tu gran esfuerzo para sacarnos adelante y por todas esas clases a las 5: 00 A.M. en la pizarra de nuestro humilde hogar. A ambos por ser padres ejemplares y por el gran apoyo que me dieron a lo largo de mi vida.

**A mi Madrina:** Rosa Moreano Jimenez, por ser una segunda madre en mi vida, por todo su amor, sus atenciones, sus consejos, por todos sus cuidados, por brindarme las puertas de su hogar en los momentos difíciles, por ser mi gran soporte cuando nació mi hija y por todo su apoyo en todas las etapas de mi vida, que contribuyeron en todos mis éxitos.

**A mi Hija:** Lia Samira Nuñez Moreano, por ser fuente de inspiración y motivación a salir adelante en los momentos difíciles y a no rendirme.

**A mi Familia:** Carmen N., Justina M., Gregorio J., Wilber M, Violeta M., Felix M., Nilda M., Olga M. y Alfredo M. por su apoyo y consejos.

**A mis Colegas:** Jimmy, Raul, Diego, Rony, Lucia, Xiomar, Lina, Simone, Liliana, Miguel por compartir su experiencia, conocimiento y contribuir en mi formación profesional.

**Asesor:** César A., por todo su apoyo, conocimiento y experiencia en el presente informe.

**A la UNMSM:** Por su prestigio y mística, a quien siempre llevare con orgullo, mi Decana de América.

# UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
## FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA
## ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS

**Aplicación web para la gestión de banco de datos para las empresas del sector hidrocarburo.**


Autor: Yul Anthony Nuñez Moreano
Asesor: César Augusto Angulo Calderón


Título: Trabajo de Suficiencia Profesional para optar el Título Profesional de Ingeniero de Sistemas
Fecha: Junio del 2021


---


**Resumen**

El presente trabajo de experiencia profesional describe el desarrollo de la Aplicación Web de Gestión de Órdenes de Trabajo para el Banco de Datos de una Empresa del Sector Hidrocarburos. Esta propuesta reemplaza a diversas aplicaciones heredadas, en donde se identificaron errores de software, tecnología obsoleta, deficiencia en escalabilidad, baja modularidad y reducida integración. Lo que generaba retraso y retrabajo en los entregables de cada mes, exponiendo a la Empresa a penalidades y desprestigio institucional.

La construcción de la aplicación se realizó en base a una Metodología Híbrida, incorporando los mejores conceptos de Kanban y RUP, para el desarrollo se utilizó como herramientas tecnológicas el Framework de Angular, Biblioteca de Kendo Telerik, Framework de Spring Boot, Base de Datos PostgreSQL y Tibco Spotfire para la implementación de diversos Dashboards estratégicos.

El proyecto fue exitoso resolvió los problemas identificados y se adaptó a las nuevas necesidades de los usuarios, en diciembre del 2018 se puso en producción en la Empresa de Hidrocarburos. También se realizó una segunda implementación en la institución del Servicio Geológico ubicado en Colombia, esto fue realizado en Julio del 2019, la aplicación sigue operando hasta la fecha en ambas instituciones.



**Palabras claves:** Orden de Trabajo, Banco de Datos, Metadata, Dashboards, Framework Angular, Hidrocarburo.

# NATIONAL UNIVERSITY OF SAN MARCOS
## FACULTY OF SYSTEMS AND COMPUTER ENGINEERING
## PROFESSIONAL SCHOOL OF SYSTEMS ENGINEERING

**Web application for data bank management for companies in the hydrocarbon sector.**


Author: Yul Anthony Nuñez Moreano
Advisor: César Augusto Angulo Calderón


Title: Work, to opt for the Professional Title of Systems Engineer
Date: June 2021


---


**Abstract**

The present work of professional experience describes the development of the Web Application for the Management of Work Orders for the Data Bank of a Company in the Hydrocarbon Sector. This proposal replaces several legacy applications, where software errors, obsolete technology, scalability deficiencies, low modularity and reduced integration were identified. What generated delay and rework in the deliverables of each month, exposing the Company to penalties and institutional discredit.

The construction of the application was carried out based on a Hybrid Methodology, incorporating the best concepts of Kanban and RUP, for the development the Angular Framework, Kendo Telerik Library, Spring Boot Framework, PostgreSQL Database were used as technological tools. And Tibco Spotfire for the implementation of various strategic Dashboards.

The project was successful, solved the problems identified and adapted to the new needs of users, in December 2018 it was put into production in the Hydrocarbons Company. A second implementation was also carried out in the institution of the Geological Service located in Colombia, this was carried out in July 2019, the application continues to operate to date in both institutions.



**Keywords:** Work Order, Data Bank, Metadata, Dashboards, Angular Framework, Hydrocarbon.

## Tabla de contenidos


*Ficha Catalográfica* ---------- *II*

*Dedicatoria* ---------- *III*

*Agradecimientos* ---------- *IV*

*Resumen* ---------- *V*

*Abstract* ---------- *VI*

*Introducción* ---------- *XV*

*Capítulo I - Trayectoria Profesional* ---------- *17*

*Capitulo II - Contexto en el que se desarrolló la experiencia* ---------- *21*

**2.1. Empresa – Actividad que realiza** ---------- **21**

**2.2. Visión** ---------- **22**

**2.3. Misión** ---------- **22**

**2.4. Organización de la empresa** ---------- **23**

**2.5. Área, cargo y funciones desempeñadas** ---------- **25**

**2.6. Experiencia profesional realizada en la organización** ---------- **25**

*Capitulo III - Actividades desarrolladas* ---------- *26*

**3.1. Situación problemática** ---------- **26**

3.1.1. Definición del problema ---------- 26

**3.2. Solución** ---------- **28**

3.2.1. Objetivos ---------- 28

3.2.2. Alcance ---------- 28

3.2.3. Etapas y metodología ---------- 30

3.2.4. Fundamentos utilizados ---------- 95

3.2.5. Implementación de la aplicación ---------- 106

**3.3. Evaluación** ---------- **144**

3.3.1. Evaluación económica ---------- 144

3.3.2. Beneficio para la organización ---------- 145

*Capitulo IV - Reflexión crítica de la experiencia* ---------- *146*

*Capítulo V - Conclusiones y recomendaciones* ---------- *151*

**5.1. Conclusiones** ---------- **151**

**5.2. Recomendaciones** ---------- **153**

**5.3. Fuentes de información** ---------- **154**

**5.3.1. Documentos administrativos** ---------- **154**

*5.3.2. Bibliografía académica ----------154*

**5.4. Glosario---------- 156**

***Anexos ----------159***

**Anexo A. Tabla de los principales errores tecnológicos de los softwares heredados.---------- 159**

**Anexo B. Código fuente del backend ---------- 161**

## Índice de diagramas

Diagrama 1: Organigrama General ..... 23
Diagrama 2: Organigrama Perú ..... 24
Diagrama 3: Organigrama del Área de Sistemas ..... 24
Diagrama 4: Mapa Proceso de un Banco de Datos ..... 39
Diagrama 5. Flujo de Trabajo del Grupo Recepción – Información Nueva ..... 40
Diagrama 6: Flujo de Trabajo del Grupo Recepción – Información Heredada ..... 40
Diagrama 7: Flujo de Trabajo General – Caso Información Nueva ..... 41
Diagrama 8: Flujo de Trabajo General – Caso Información Heredada ..... 41
Diagrama 9: Flujo de Entrega de Información ..... 42
Diagrama 10: Diagrama de Actores del Banco de Datos ..... 48
Diagrama 11: Diagrama de Paquetes de la Aplicación. ..... 49
Diagrama 12: Diagrama de Caso de Uso Listar Ordenes de Trabajo ..... 50
Diagrama 13: Diagrama de Caso de Uso Consultar Orden de Trabajo ..... 51
Diagrama 14: Diagrama de Caso de Uso Registrar Orden de Trabajo ..... 51
Diagrama 15: Diagrama de Caso de Uso Actualizar Orden de Trabajo ..... 52
Diagrama 16: Diagrama de Caso de Uso Registrar Seguimiento Documental ..... 52
Diagrama 17: Diagrama de Caso de Uso Registrar Dumps ..... 53
Diagrama 18: Diagrama de Caso de Uso Listar Tareas ..... 53
Diagrama 19: Diagrama de Caso de Uso Consultar Tarea ..... 54
Diagrama 20: Diagrama de Caso de Uso Crear Tareas ..... 54
Diagrama 21: Diagrama de Caso de Uso Actualizar Tarea ..... 55
Diagrama 22: Diagrama de Caso de Uso Eliminar Tarea ..... 55
Diagrama 23: Diagrama de Caso de Uso Registrar Tracking Tarea ..... 56
Diagrama 24: Diagrama de Caso de Uso Cargar Metadata ..... 56
Diagrama 25: Diagrama de Caso de Uso Registrar Metadata para Catalogo ..... 57
Diagrama 26: Diagrama de Caso de Uso Registrar Metadata para Archivo Técnico ..... 57
Diagrama 27: Diagrama de Caso de Uso Registrar Metadata para GIS ..... 58
Diagrama 28 Diagrama de Caso de Uso Registrar Metadata para Pozos ..... 58
Diagrama 29: Diagrama de Caso de Uso Registrar Metadata para Sísmica ..... 59
Diagrama 30: Diagrama de Caso de Uso Actualizar Metadata para Catalogo ..... 59
Diagrama 31: Diagrama de Caso de Uso Actualizar Metadata para Archivo ..... 60
Diagrama 32: Diagrama de Caso de Uso Actualizar Metadata para GIS ..... 60
Diagrama 33: Diagrama de Caso de Uso Actualizar Metadata para Pozos ..... 61
Diagrama 34: Diagrama de Caso de Uso Actualizar Metadata para Sísmica ..... 61
Diagrama 35: Diagrama de Caso de Uso Eliminar Metadata para Catalogo ..... 62
Diagrama 36: Diagrama de Caso de Uso Eliminar Metadata para Archivo ..... 62
Diagrama 37: Diagrama de Caso de Uso Eliminar Metadata para GIS ..... 63
Diagrama 38: Diagrama de Caso de Uso Eliminar Metadata para Pozos ..... 63
Diagrama 39: Diagrama de Caso de Uso Eliminar Metadata para Sísmica ..... 64
Diagrama 40: Diagrama de Caso de Uso Generar Certificado para Recepción ..... 64
Diagrama 41: Diagrama de Caso de Uso Generar Certificado para Pozos ..... 65
Diagrama 42: Diagrama de Caso de Uso Generar Certificado para GIS ..... 65

Diagrama 43: Diagrama de Caso de Uso Generar Certificado para Archivo Técnico ... 66
Diagrama 44: Diagrama de Caso de Uso Generar Certificado para Sísmica ... 66
Diagrama 45: Arquitectura General de la Aplicación ... 90
Diagrama 46: Arquitectura Frontend de la Aplicación ... 91
Diagrama 47: Arquitectura Backend de la Aplicación ... 91
Diagrama 47: Arquitectura Dashboartds ... 92
Diagrama 48: Modelo Entidad Relación de la Aplicación ... 93

## Índice de figuras


Figura 1 Identificando causas del problema 26
Figura 2 Diagrama Causa – Efecto del Problema 27
Figura 3: Ubicación del área en donde se utilizara la aplicación web. 29
Figura 4 Metodologia en las diferentes etapas del proyecto 30
Figura 5: Flujo de Trabajo del Tablero Kanban 31
Figura 6: Lista de Tarjetas Kanban en el Backlog 32
Figura 7: Lista de Tarjetas Kanban en el To Do 33
Figura 8: Lista de Tarjetas Kanban en el Fase Doing 34
Figura 9: Lista de Tarjetas Kanban en el Fase Stop 35
Figura 10: Lista de Tarjetas Kanban en el Fase Done 36
Figura 11: Estadística Metodología de desarrollo de software practicadas el 2021 99
Figura 12: Estadística de Framework más utilizados en el 2020 101
Figura 13: Estadística de Base de Datos más utilizados en el 2020 102
Figura 14: PostgreSQL selección versión a instalar 106
Figura 15: PostgreSQL selección de paquetes a instalar 107
Figura 16: PostgeSQL ubicación de archivo pg_hba.conf 107
Figura 17: PostgreSQL configuración de acceso dentro del dominio 107
Figura 18: PostgreSQL modificar reglas de firewall 108
Figura 19: PostgreSQl instalar herramienta de administración 108
Figura 20: Angular instalar NodeJS 109
Figura 21: Angular instalación mediante angular/cli 109
Figura 22: Angular corroborar instalación y versión 109
Figura 23: Apache Tomcat instalación 110
Figura 24: Apache Tomcat modificación de server.xml 110
Figura 25: Apache Tomcat modificación de puerto 111
Figura 26: Apache Tomcat herramienta de administración 111
Figura 27: Keycloak descarga 112
Figura 28: Keycloak instalación standalone.bat 112
Figura 29: Keycloak  agregar realm 112
Figura 30: Keycloak agregar cliente 113
Figura 31: Keycloak zona de configuración 113
Figura 32 Dashboard Transformando datos de la vista View_Orders_PP 115
Figura 33: Dashboard agregando gráficos en base a la vista View_Orders_PP 115
Figura 34: Dashboard transformando datos de la Vista Orders_Join_Task 116
Figura 35: Dashboard agregando gráficos en base a la Vista Orders_Join_Task 117
Figura 36: Dashboard transformando datos de la Vista View_Memos 118
Figura 37: Dashboard agregando gráficos en base a la Vista View_Memos 119
Figura 38: Frontend interface login 119
Figura 39: Frontend interface Listado de Ordenes de Trabajo parte 1 de 2 120
Figura 40: Frontend interface Listado de Ordenes de Trabajo parte 2 de 2 120
Figura 41: Frontend interface Consulta de Orden de Trabajo 121

Figura 42: Frontend interface Registrar Orden de Trabajo 121
Figura 43: Frontend interface Registrar Tarea 122
Figura 44: Frontend interface Listado de Tareas 122
Figura 45: Frontend interface Consulta de Tarea 123
Figura 46: Frontend interface Consulta Tracking de Tarea 123
Figura 47: Frontend interface Cargar Metadata 124
Figura 48: Frontend interface Registrar Metadata de Catalogo parte 1 de 2 124
Figura 49: Frontend interface Registrar Metadata de Catalogo parte 2 de 2 125
Figura 50: Frontend interface Registrar Metadata de Archivo parte 1 de 6 125
Figura 51: Frontend interface Registrar Metadata de Archivo parte 2 de 6 126
Figura 52: Frontend interface Registrar Metadata de Archivo parte 3 de 6 126
Figura 53: Frontend interface Registrar Metadata de Archivo parte 4 de 6 127
Figura 54: Frontend interface Registrar Metadata de Archivo parte 5 de 6 127
Figura 55: Frontend interface Registrar Metadata de Archivo parte 6 de 6 128
Figura 56: Frontend interface Registrar Metadata de GIS parte 1 de 5 128
Figura 57: Frontend interface Registrar Metadata de GIS parte 2 de 5 129
Figura 58: Frontend interface Registrar Metadata de GIS parte 3 de 5 129
Figura 59: Frontend interface Registrar Metadata de GIS parte 4 de 5 129
Figura 60: Frontend interface Registrar Metadata de GIS parte 5 de 5 130
Figura 61: Frontend interface Registrar Metadata de Pozo parte 1 de 11 130
Figura 62: Frontend interface Registrar Metadata de Pozo parte 2 de 11 130
Figura 63: Frontend interface Registrar Metadata de Pozo parte 3 de 11 131
Figura 64: Frontend interface Registrar Metadata de Pozo parte 4 de 11 131
Figura 65: Frontend interface Registrar Metadata de Pozo parte 5 de 11 131
Figura 66: Frontend interface Registrar Metadata de Pozo parte 6 de 11 131
Figura 67: Frontend interface Registrar Metadata de Pozo parte 7 de 11 132
Figura 68: Frontend interface Registrar Metadata de Pozo parte 8 de 11 132
Figura 69: Frontend interface Registrar Metadata de Pozo parte 9 de 11 132
Figura 70: Frontend interface Registrar Metadata de Pozo parte 10 de 11 133
Figura 71: Frontend interface Registrar Metadata de Pozo parte 11 de 11 133
Figura 72: Frontend interface Alerta códigos duplicados 134
Figura 73: Frontend interface Certificado del Grupo de Recepción 134
Figura 74: Frontend interface Certificado del Grupo de Archivo Técnico 135
Figura 75: Frontend interface Certificado del Grupo de GIS 135
Figura 76: Frontend interface Alerta de errores detectados en Metadata de Pozos 136
Figura 77: Frontend interface Certificado del Grupo de Pozos 136
Figura 78: Frontend interface Reporte Ordenes de Trabajo en Proceso 137
Figura 79: Frontend interface Reporte de Tareas Pendientes 137
Figura 80: Frontend interface Dashboard Gerencial 138
Figura 81: Frontend interface Dashboard Ordenes de Trabajo 138
Figura 82: Frontend interface Dashboard Tareas 139
Figura 83: Frontend interface Dashboard Cuadro Control 139
Figura 84: Tarjeta Kanban actividada test de servicios. 140

Figura 85: Postman para validar servicios REST. 141
Figura 86: NodeJS Iniciamos todos los servicios y configuraciones. 142
Figura 87: Terminal Visual Code generando carpeta DIST. 142
Figura 88: Estructura de desarrollo ubicando la carpeta DIST. 142
Figura 89: Ejecutando Maven Install. 143
Figura 90: Apache tomcat Replegar .WAR. 143
Figura 91: Apache Tomcat Desplegar .WAR. 143
Figura 92: Publicación del Proyecto en Open Earth 147
Figura 93: Lista de Integrantes del Proyecto 147

## Índice de tablas

Tabla 1: Etapas del desarrollo del Proyecto ---------- 37
Tabla 2: Lista de Requerimientos Funcionales---------- 43
Tabla 10: Requerimiento  No Funcionales ---------- 44
Tabla 11: Actores del Sistema Nivel Gerencia ---------- 45
Tabla 12: Actores del Sistema Nivel Lideres---------- 46
Tabla 13: Actores del Sistema Nivel Consultores ---------- 47
Tabla 14: Caso de Uso Registrar Órdenes de Trabajo ---------- 68
Tabla 15: Caso de Uso Actualizar Órdenes de Trabajo ---------- 70
Tabla 16: Caso de Uso Registrar Tareas---------- 71
Tabla 17: Caso de Uso Actualizar Tareas---------- 73
Tabla 18: Caso de Uso Eliminar Tareas ---------- 74
Tabla 19: Caso de Uso Cargar Metadata---------- 76
Tabla 20: Caso de Uso Registrar Metadata Catalogo---------- 78
Tabla 21: Caso de Uso Registrar Metadata Archivo---------- 79
Tabla 22: Caso de Uso Registrar Metadata GIS---------- 81
Tabla 23: Caso de Uso Generar Certificados de Recepción ---------- 82
Tabla 24: Caso de Uso Generar Certificados de Archivo ---------- 84
Tabla 25: Caso de Uso Generar Certificados de GIS ---------- 85
Tabla 26: Caso de Uso Reporte de Ordenes de Trabajo en Proceso---------- 87
Tabla 27: Caso de Uso Reporte Tareas Pendientes ---------- 88
Tabla 28: Caso de Uso Visualizar Dashboard Gerencial---------- 89
Tabla 56 Dashboard generando la vista View_Orders_PP ----------114
Tabla 57: Dashboard generando la vista Orders_Join_Task----------116
Tabla 58: Dashboard generando la vista View_Memos ----------117
Tabla 59: Estimación de Costos----------144

## Introducción

El presente trabajo de experiencia profesional explica la metodología y las etapas del desarrollo de la Aplicación Web Gestión de Ordenes de Trabajo, que tiene como principal objetivo reemplazar los softwares heredados relacionados con este proceso de negocio.

Por motivos contractuales y de confidencialidad, el nombre de la empresa en donde se realizó la implementación se mantendrá en reserva, refiriéndonos a ella de forma genérica como Empresa del Sector de Hidrocarburo.

La Empresa de Hidrocarburo cuenta con un Banco de Datos, quien está encargado de gestionar la información de petróleo y gas, brindando a los inversionistas información técnica que facilite la toma de decisiones para la exploración y explotación de recursos en el país. El Banco de Datos por reglamento de la empresa debe ser administrado por terceros entrando en licitación pública cada 3 años. Las diversas gestiones realizan cambios de infraestructura, reemplazo de software licenciados, cambios en los flujos de trabajo y mantenimiento de software heredados. Estos últimos no son actualizados tecnológicamente, sus modificaciones no son documentadas y han estado operando con errores y realizando correcciones manuales.

El proyecto se presenta como prioridad muy alta, porque afecta los plazos de entrega excediendo los tiempos límites para cerrar una orden lo que genera penalidades y desprestigio institucional.

Ante ello surge la necesidad de cambiar estos softwares heredados, con herramientas modernas como el Framework de Angular, Kendo Telerik, Spring Boot, PostgreSQL y elaboración de Dashboards mediante Tibco Spotfire como soporte en la toma de decisiones.

El desarrollo del software se realizó dentro del marco de una metodología híbrida, basándonos en los estudios de (S. D. Orantes Jiménez, 2013 citado en Molina Ríos, Zea Ordóñez, Contento Segarra, & García Zerda, 2018, p. 5) quien valida el uso de metodologías existentes, combinando así sus mejores prácticas.

El contenido de este trabajo de experiencia profesional se explica en cinco capítulos, los cuales se detallan de la siguiente forma.

El Capítulo I, se presenta mi experiencia profesional y académica siendo base fundamental para formar parte de la empresa Halliburton, en donde asumí el rol de Analista Programador en diversos proyectos.

El Capítulo II, se detalla el contexto del desarrollo del trabajo profesional, se lista las funciones desempeñadas en la organización, se describe a la empresa Halliburton, su organigrama, visión y misión.

El Capítulo III, se plantea el problema, objetivos generales, objetivos específicos, el alcance, la metodología, las etapas de desarrollo, los fundamentos teóricos, la implementación y los beneficios del software.

El Capítulo IV, compartiré las diversas reflexiones, lecciones aprendidas y críticas constructivas.

El Capítulo V, conclusiones y recomendaciones.

## Capítulo I - Trayectoria Profesional

La trayectoria profesional se detalla a continuación con la información de mi resumen profesional:

| **Experiencia Profesional** | |
|---|---|
| Associate Consultant.<br>**Halliburton del Peru S.R.L**<br>• Desarrollo de Software de Aplicación de Administración de Ordenes de Trabajo.<br>• Desarrollo de módulo de Certificados de Calidad.<br>• Desarrollo del Módulo de Gestión de Usuarios para el Portal del Banco de Datos.<br>• Desarrollo de módulo de carga de datos desde un Excel.<br>• Implementación de Dashboard de Recepción y Entrega de Información.<br>• Implementación de Dashboard de Tareas.<br>• Implementación de Dashboard de Ordenes de Trabajo.<br>• Implementación de Dashboard de Cuadro Control.<br>• Implementación de Dashboard Memorándums.<br>• Implementación de Dashboard Gerencial.<br>• Implementación de Dashboard de Visitas y Descargas de Usuario del Portal Web.<br>• Desarrollo de módulo de Reporte Económico para el SGC (Colombia).<br>• Administrador de Base de Datos: Oracle, SLQ Server, PostgreSQL y MySQL. | 02.07.2018 – Hasta la Actualidad |

| | |
|---|---|
| • Administrador de Aplicaciones: Petrobank, Dropsite, WebFramework, Power Explorer.<br>• Administrador de usuarios: FortiClient VPN y Active Directory.<br>• Implementación de Software de Aplicación de Carga de Datos y visualización en un Dashboard de los Pozos Petroleros en Producción y Reservorio. | |
| Programador de Sistemas Informáticos.<br>**Grupo San Marcos S.A.C.**<br>• Desarrollo de Pagina Web Institucional.<br>• Desarrollo de módulo de encuesta de satisfacción de usuario.<br>• Community Manager. | 15.12.2015 – 31.12.2017 |

| Formación Profesional | |
|---|---|
| Estudiante de Maestría en Gestión de Tecnologías de Información y Comunicación. (2do Ciclo).<br>**Universidad Nacional Mayor de San Marcos.** | 2020 - Actualidad |
| Grado Académico de Bachiller.<br>Ingeniería de Sistemas e Informática.<br>**Universidad Nacional Mayor de San Marcos**. | 2004 – 2016 |

| Cursos | |
|---|---|
| Remote Work and Virtual Collaboration.<br>**CertiProf.** | Diciembre 2020 |
| Visualización y Análisis de datos en TIBCO Cloud Spotfire.<br>**Udemy.** | Agosto 2020 |
| Scrum Foundation Professional Certificate<br>**CertiProf.** | Agosto 2020 |
| Curso de Scrum<br>**Platzi.** | Junio 2020 |
| Regresión Lineal con Python.<br>**Platzi**. | Abril 2020 |
| Machine Learning para Análisis Predictivo.<br>**Platzi**. | Enero 2020 |
| Web Apps y Logic Apps en Azure<br>**Platzi**. | Enero 2020 |

| Seminarios | |
|---|---|
| Project Management Awareness – An Introduction to Project Management **Halliburton.** | Halliburton |
| Emergency Preparedness. **Halliburton.** | Halliburton |
| Code of Business Conduct. **Halliburton.** | Halliburton |
| Critical Focus Areas. **Halliburton.** | Halliburton |
| Email Etiquette. **Halliburton**. | Halliburton |
| Environmental Awareness. **Halliburton**. | Halliburton |
| Hazard Comunication. **Halliburton.** | Halliburton |
| Halliburton information Security Awareness Training **Halliburton.** | Halliburton |
| Fatigue Management **Halliburton.** | Halliburton |
| HSE Aspects and Impacts **Halliburton.** | Halliburton |
| Project Management Awareness **Halliburton.** | Halliburton |

## Capitulo II - Contexto en el que se desarrolló la experiencia

### 2.1. Empresa – Actividad que realiza

Empresa Halliburton del Perú S.R.L. está dedicada principalmente a las actividades de servicios relacionadas con la extracción de petróleo, gas y consultoría.

Datos de Empresa

- Razón Social: Halliburton del Perú S.R.L.
- Domicilio Legal: Calle German Schreiber 276 San Isidro, Lima, Perú.
- Teléfono: 57883046056
- Ruc: 20100118506
- Web: www.halliburton.com

Principales Cliente en Perú

- Empresa del Sector Hidrocarburos.

Principales Cliente en Colombia

- Ecopetrol.
- Servicio Geológico Colombiano (SGC).

Principales Proveedores en Perú

- Open Network System S.A.C.

Mercado:

Halliburton es uno de los principales proveedores de productos y servicios del mundo para la industria energética. Halliburton es una empresa de servicios y de una sólida posición competitiva que no solo proviene de su presencia geográfica y su amplia gama de productos y servicios, sino también, lo que es más importante, de la calidad del trabajo y la dedicación de sus empleados. Con aproximadamente 55,000 empleados, que representan a 140 nacionalidades en más de 80 países, Halliburton toca gran parte del petróleo y el gas que alimentan nuestra sociedad. Halliburton comprende 14 líneas de servicio de productos (PSL). Los PSL operan en dos divisiones: Perforación y Evaluación, y Terminación y Producción.

La PSL de consultoría y gestión de proyectos trabaja en ambas divisiones y es la punta de lanza de nuestra estrategia de servicios integrados.

## 2.2. Visión

"Ser reconocidos como líder a nivel mundial como proveedor de productos y servicios para las necesidades en constante evolución de la industria energética."

## 2.3. Misión

"Garantizar una explotación segura y económica de los hidrocarburos, para lo cual el desarrollo de tecnologías de alto desempeño, permiten realizar esta la tarea más racional y productiva."

## 2.4. Organización de la empresa

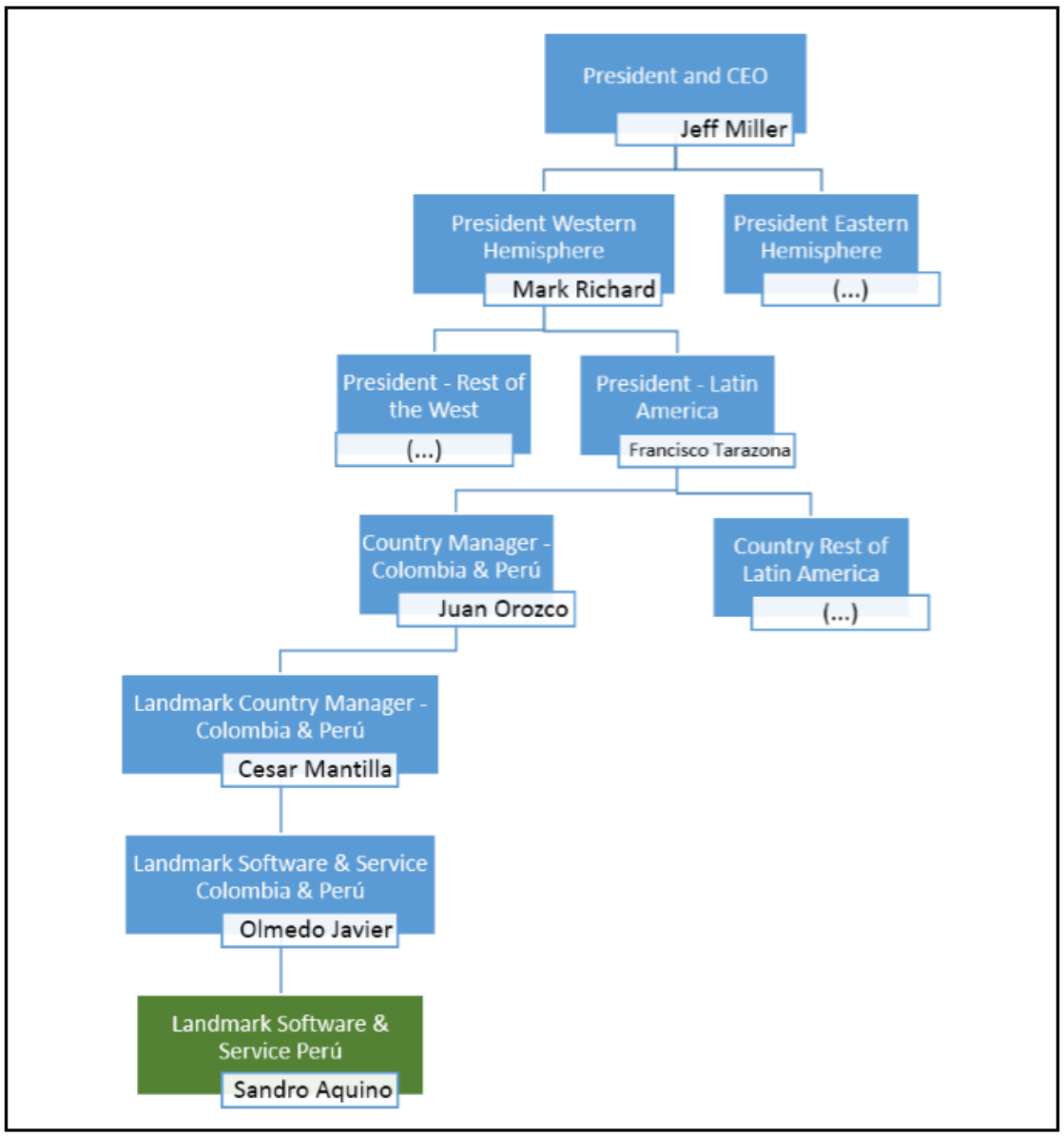



Diagrama 1: Organigrama General

Fuente: Elaboración Propia (Año 2021)

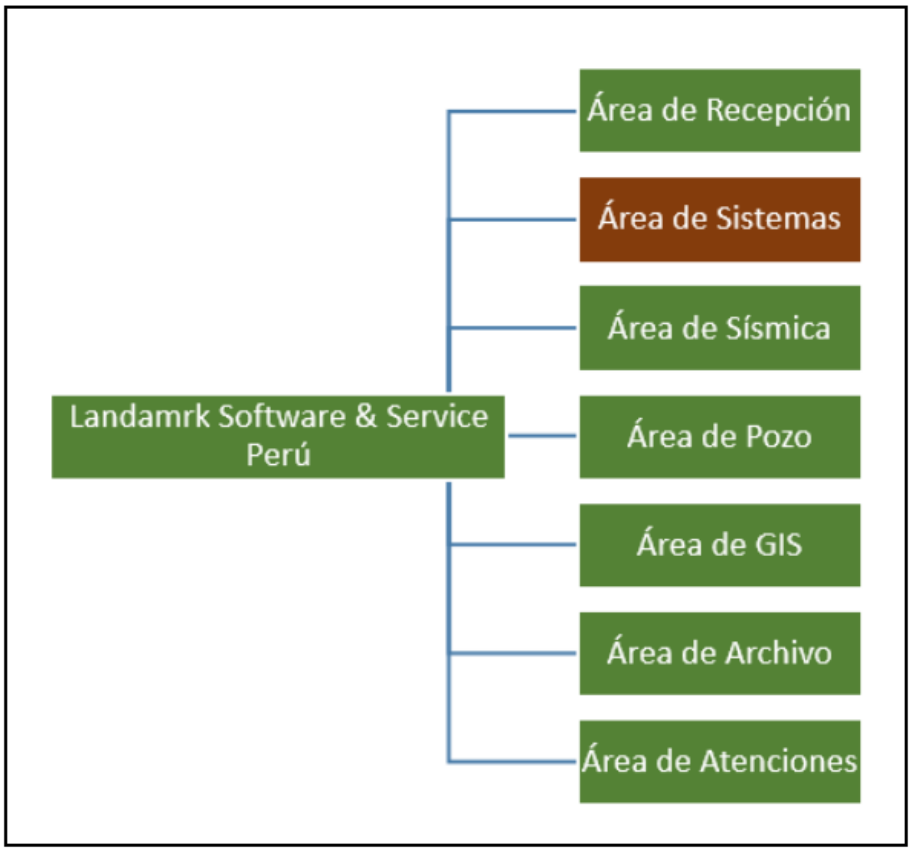



Diagrama 2: Organigrama Perú

Fuente elaboración propia (Año 2021)

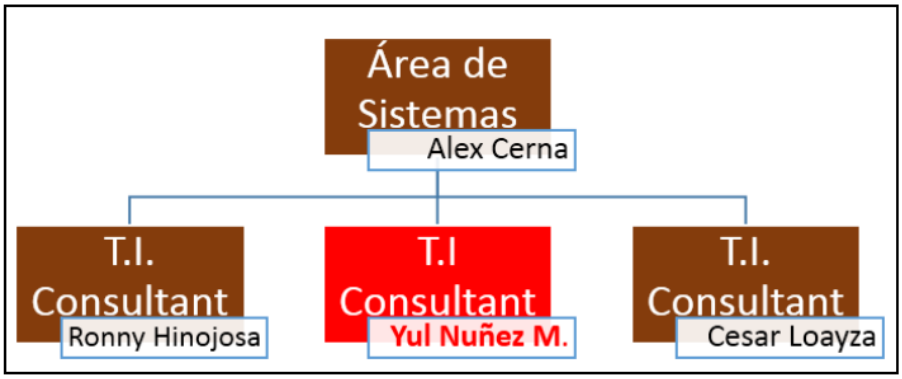



Diagrama 3: Organigrama del Área de Sistemas

Fuente elaboración propia (Año 2021)

## 2.5. Área, cargo y funciones desempeñadas

## 2.6. Experiencia profesional realizada en la organización

**Funciones realizadas:**

- ✓ Administrador de Base de Datos: Oracle, PostgreSQL, Mysql y SQL Server.
- ✓ Desarrollador de Software: Java, TypeScript, JavaScript, NodeJS, Express.js, PrimeFaces, Spring Boot, Angular, Maven, Tomcat, GlassFish, PrimeNG, Boostrap y ASP.net.
- ✓ Elaboración de Dashboards estratégicos, tácticos y operativos en Tibco Spotfire.
- ✓ Elaboración de Scripts.
- ✓ Elaboración mensual del Informe de Monitoreo de la Infraestructura, en donde se determina el porcentaje de disponibilidad mensual de los diferentes servicios: aplicaciones, base de datos, internet, almacenamiento.
- ✓ Administrador de Aplicaciones: Petrobank, Dropsite, Webframework, Power Explore y DSIS.
- ✓ HelpDesk.

En el proyecto de desarrollo de la aplicación web para la administración de órdenes de trabajo para el banco de datos de Perupetro, me desempeñe como Analista Programador y mis funciones fueron:

- ✓ Análisis de los requerimientos.
- ✓ Desarrollo del Frontend.
- ✓ Desarrollo del Backend.
- ✓ Implementación de Dashboards.

# Capitulo III - Actividades desarrolladas

## 3.1. Situación problemática

### 3.1.1. Definición del problema

La administración del Banco de Datos entra en licitación cada 3 años, desde su funcionamiento ha sido administrado por diversas empresas, donde en cada gestión se realiza cambios tecnológicos entre ellos el reemplazo de software por motivo de licencia, cambios en los flujos de trabajo y mantenimiento de software heredados sin licencia.

El problema de los softwares heredados es que no eran actualizados tecnológicamente, las modificaciones no eran documentadas, no se encontraban integradas y se operaba con errores de software que eran corregidos manualmente. (Ver Anexo A) Por ejemplo:

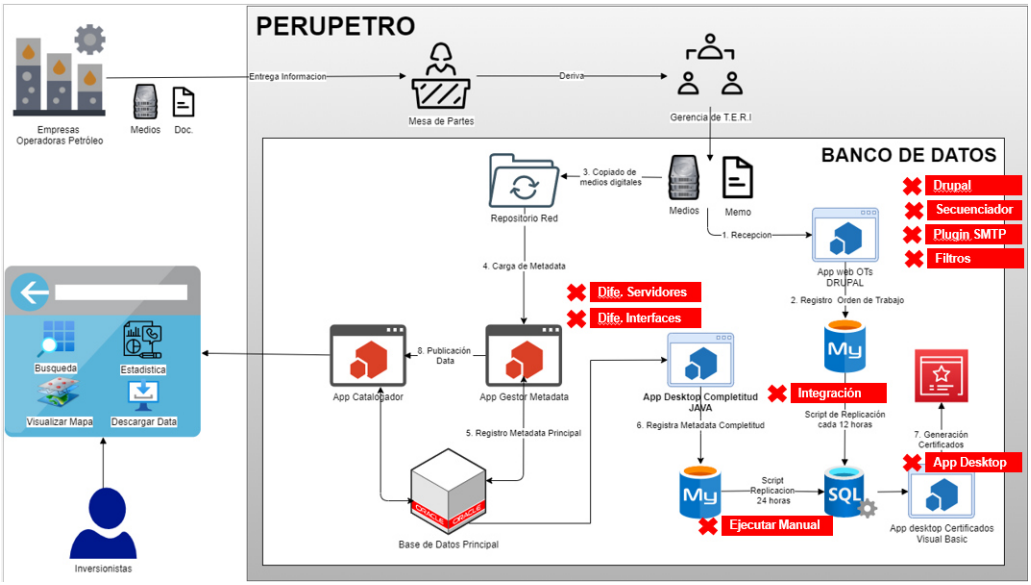



Figura 1 Identificando causas del problema

Fuente Elaboración Propia (Año 2021)

Lo que generaba los siguientes inconvenientes:

- Difícil mantenimiento de las aplicaciones, lo que podría generar costos adicionales en soporte.
- Bajos niveles de modularidad, lo que generaba demora en personalizaciones, dificultad en adaptar la aplicación a nuevos flujos de trabajo y complejidad en el desarrollo de nuevos módulos requerido por el cliente, lo que afectaba el prestigio de la empresa Halliburton.
- El conjunto de aplicaciones no se encontraban integradas, lo que generaba retrabajo en registrar los mismos datos en las diferentes aplicaciones, errores de digitación en campos sensibles, duplicidad de datos, procesos largos en los flujos de trabajo e inseguridad de los datos.
- Errores de software por ejemplo: Contadores automáticos no se generaban de manera continua, las notificaciones de correo electrónico en algunas ocasiones no se enviaban, vistas congeladas en los certificados, plugins desfasados, entre otros. Lo que Generaba retrasos en los entregables que podrían ocasionar penalidades económicas a la empresa.

Las consecuencias de estos inconvenientes podrían generar desprestigio de la institución y riesgo de penalidades.

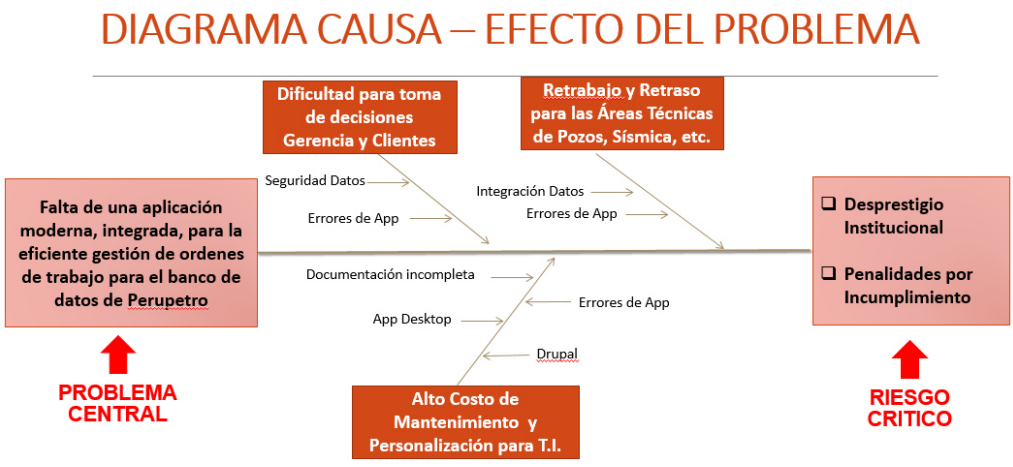



Figura 2 Diagrama Causa – Efecto del Problema

Fuente Elaboración Propia (Año 2021)

## 3.2. Solución

Implementar una aplicación web para Gestionar las Órdenes de Trabajo en el Banco de Datos de la Empresa del Sector Hidrocarburo.

### 3.2.1. Objetivos

#### 3.2.1.1. Objetivo general

Implementar un software de aplicación web utilizando herramientas tecnológicas modernas como el Framework de Angular, Biblioteca de componentes de Kendo-Telerik, Bootstrap, Spring Boot y la inclusión de Dashboards estratégicos.

Que Cumpla con las siguientes características: Escalable, Modular, Usable, Flexible, Integrado y Seguro.

#### 3.2.1.2. Objetivos específicos

- Analizar los requerimientos de la aplicación.
- Diseñar modelo de datos y arquitectura de la aplicación.
- Implementar la aplicación web.
- Implementar los Dashboards.

### 3.2.2. Alcance

#### 3.2.2.1. Alcance funcional

- La aplicación web para Gestionar las Órdenes de Trabajo contara con los principales módulos:
  - ✓ Gestión de Solicitud de Ordenes de Trabajo.
  - ✓ Gestión de Tareas.
  - ✓ Gestión de Metadata.
  - ✓ Generación de Certificados.
  - ✓ Reportes y Dashboards.
- Esta fuera del alcance para fines de esta documentación:
  - ✓ Pruebas y despliegue de la Aplicación.
  - ✓ Módulo Gestión Usuario.
  - ✓ Módulo de Carta de Transferencia.
  - ✓ Módulo de Importación desde Excel.

### 3.2.2.2. Alcance organizacional

- El requerimiento es solicitado por el Área de Recepción y Entrega de la Empresa del Sector Hidrocarburo.

### 3.2.2.3. Alcance geográfico

- La aplicación es desarrollada por Halliburton S.R.L. para la empresa del Sector Hidrocarburo quien se ubica en:




Figura 3: Ubicación del área en donde se utilizará la aplicación web.

Fuente: Google Maps (2021)

### 3.2.2.4. Limitaciones

Por un tema de privacidad y protección de datos se está nombrando EMPRESA DEL SECTOR DE HIDROCARBURO a la entidad en donde se realizó el presente trabajo profesional.

### 3.2.3. Etapas y metodología

#### 3.2.3.1. Metodología

El desarrolló de la aplicación fue realizada con una metodología hibrida tomado como referencia a RUP y Kanban. Reemplazando las historias de usuario por casos de uso, para evitar las constantes interpretaciones en los requerimientos y optimizar los tiempos de interacción con el cliente.

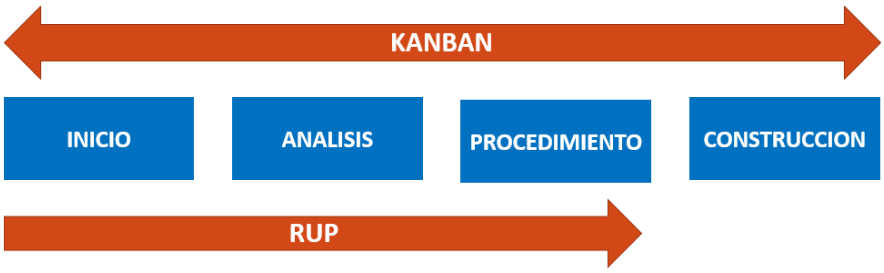



Figura 4 Metodología en las diferentes etapas del proyecto
Fuente Elaboración Propia (Año 2021)

La Metodología fue soportada en la Plataforma de Microsoft Teams quien facilito la integración de usuarios, comunicación y colaboración del equipo, considerando que los integrantes se encontraban en diferentes ubicaciones geográficas.

Dentro de la plataforma se utilizó la herramienta de Planer para simular el Tablero Kanban, el objetivo era dar visibilidad de las actividades del equipo a Gerencia Perú, Gerencia Colombia y Clientes.

### 3.2.3.2. Aplicación KANBAN

#### 3.2.3.2.1. Flujo de trabajo

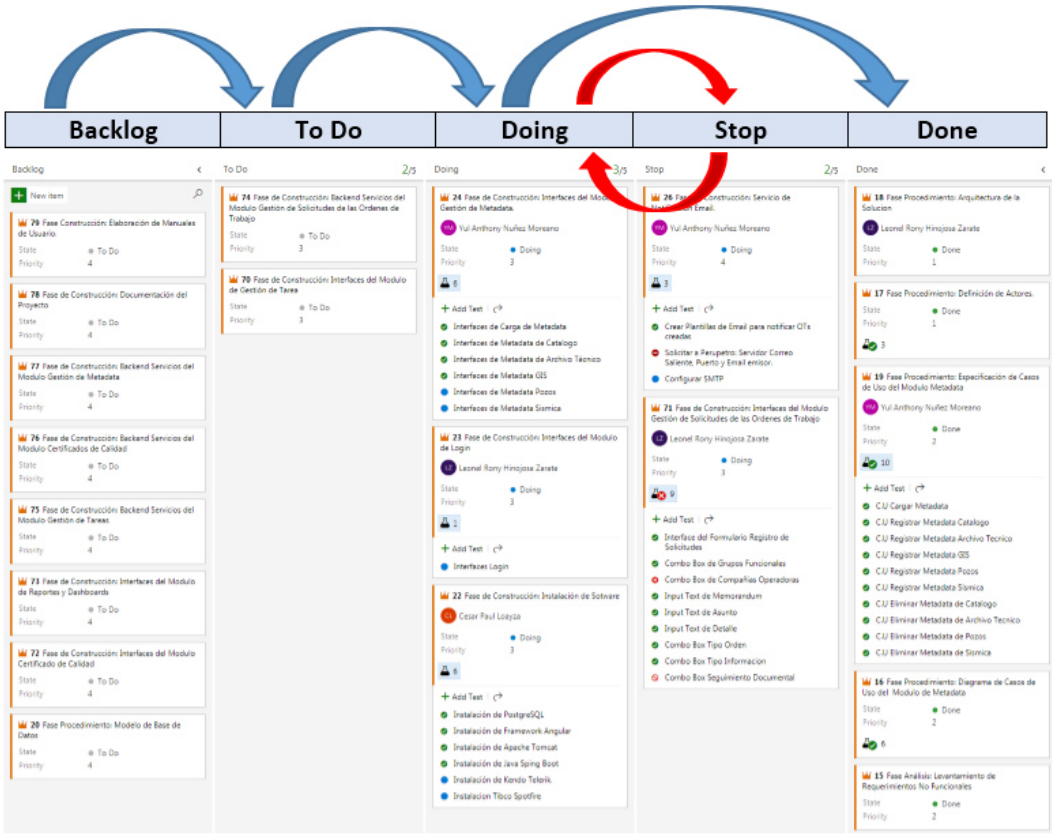



Figura 5: Flujo de Trabajo del Tablero Kanban
Fuente Elaboración Propia (Año 2021)

#### 3.2.3.2.2. Fases del ciclo

##### 3.2.3.2.2.1. Backlog

Se detallan todos los requerimientos y actividades que realizaran el flujo Kanban en esta aún no se asigna el responsable y las prioridades son relativas.




Figura 6: Lista de Tarjetas Kanban en el Backlog

Fuente Elaboración Propia (Año 2021)

### 3.2.3.2.2.2. Tareas Diarias (To Do)

Se lista tareas por tarjeta, se asigna prioridad, la asignación de usuario es en base a la carga de su trabajo.

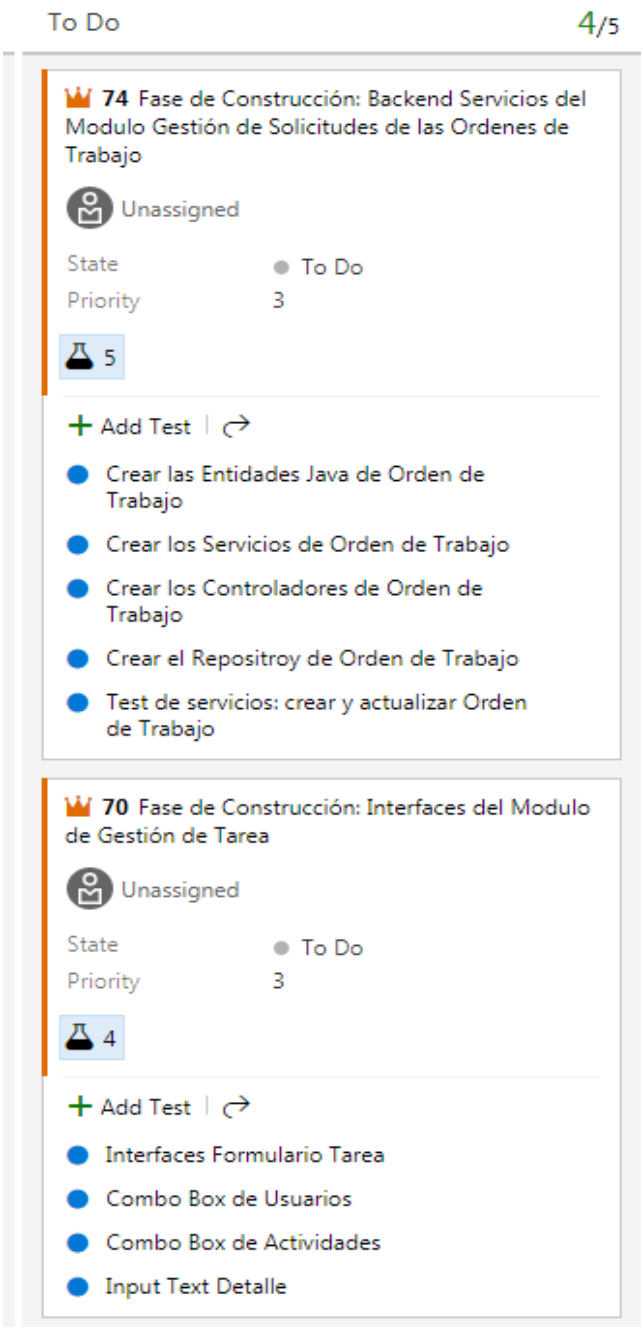



Figura 7: Lista de Tarjetas Kanban en el To Do

Fuente Elaboración Propia (Año 2021)

### 3.2.3.2.2.3. Progreso (Doing)

Las tareas están asignadas a un responsable, se puede visualizar cuales están terminadas ckeck color verde y cuales siguen en proceso punto azul

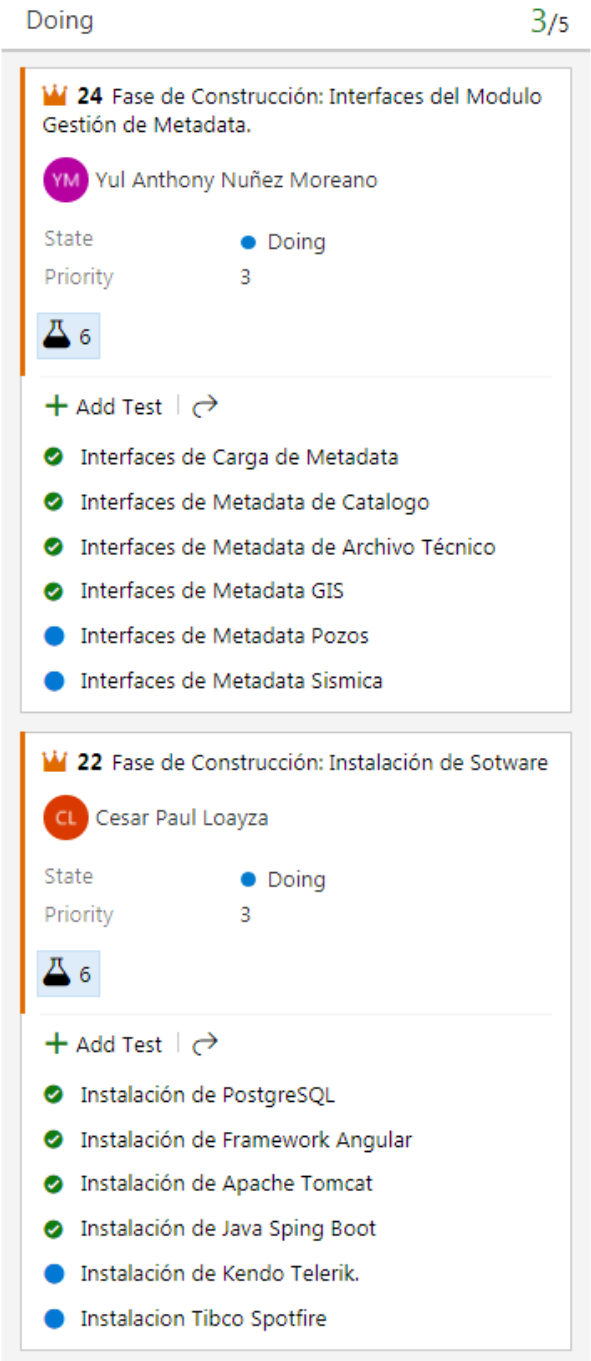



Figura 8: Lista de Tarjetas Kanban en el Fase Doing

Fuente Elaboración Propia (Año 2021)

### 3.2.3.2.2.4. Detenido (Stop)

Encontramos las tarjetas bloqueadas ya sea por falta de recursos, burocracia del cliente, fallos en los test de pruebas, fallo en el requerimiento, etc.

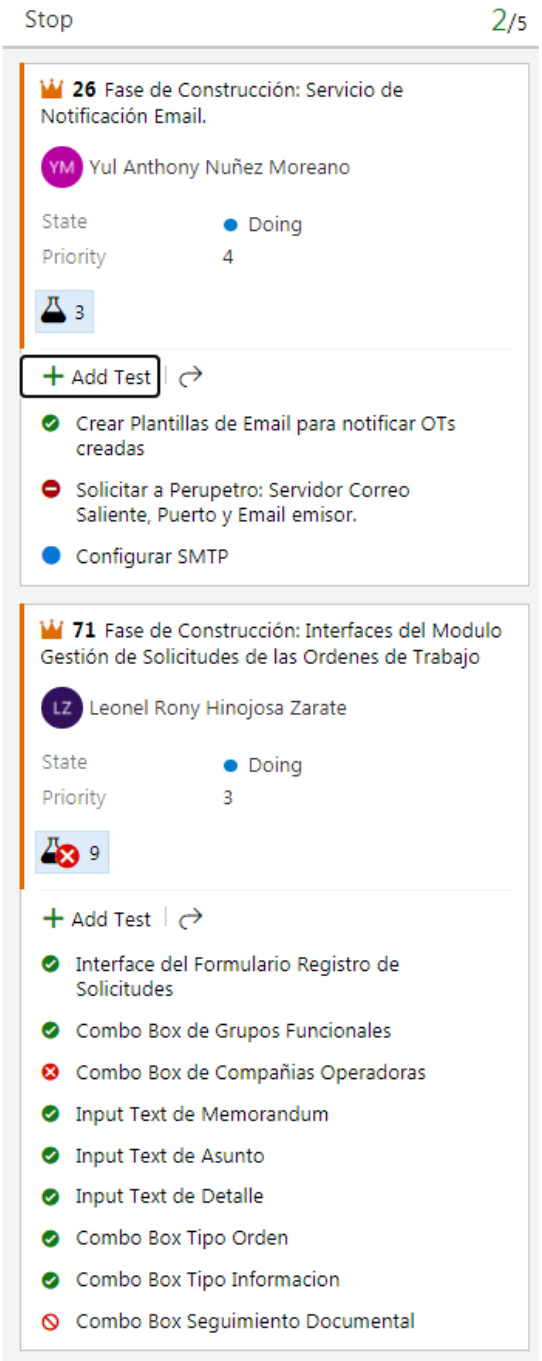



Figura 9: Lista de Tarjetas Kanban en el Fase Stop

Fuente Elaboración Propia (Año 2021)

### 3.2.3.2.2.5. Resuelto (Done)

En esta etapa las actividades cumplieron con los requerimientos y pruebas, se requiere visto bueno del cliente.

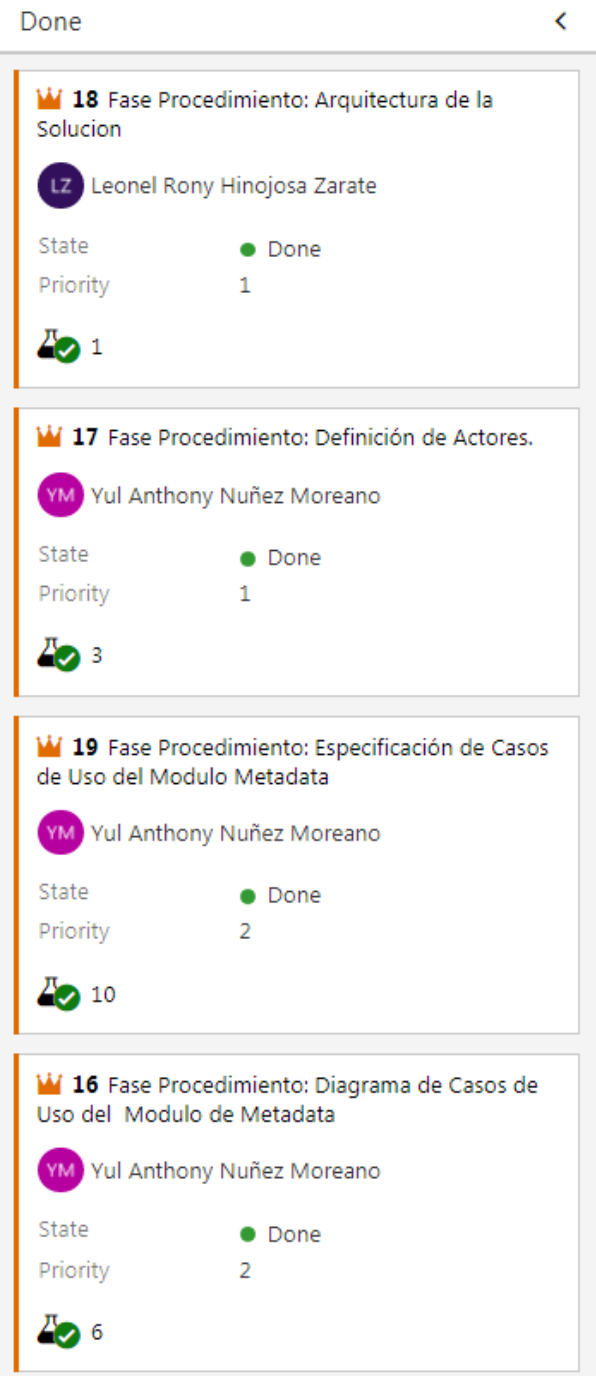



Figura 10: Lista de Tarjetas Kanban en el Fase Done

Fuente Elaboración Propia (Año 2021)

### 3.2.3.3. Etapas

| Etapas | Objetivo | Actividades |
|---|---|---|
| Inicio | Análisis y Recolección de datos | • Alcance<br>• Recolección de documentos históricos.<br>• Análisis de la situación actual<br>• Diagrama de la situación actual<br>• Documentación de la situación actual |
| Análisis | Interpretación de datos | • Requerimientos Funcionales<br>• Requerimientos No Funcionales |
| Procedimiento | Análisis y diseño de requerimientos | • Diagrama de Flujo<br>• Definición de Actores.<br>• Diagrama de Casos de Uso.<br>• Especificación de Casos de Uso.<br>• Arquitectura de General<br>• Arquitectura Frontend y Backend.<br>• Modelo de Base de Datos. |
| Construcción | Desarrollo de los requerimientos | • Configuración de Servidor.<br>• Instalación y Configuración de Base de Datos, Java Spring Boot, Tomcat, Kendo Telerik, Framework Angular y Tibco Spotfire.<br>• Implementación de los servicios del Backend e interfaces del FrontEnd.<br>• Implementación de Dashboards en el FrontEnd. |

Tabla 1: Etapas del desarrollo del Proyecto

Fuente Elaboración Propia (Año 2021)

#### 3.2.3.3.1. Etapa inicio

En esta fase nos centramos principalmente en definir el alcance, recolectar información documental e información nueva.

##### 3.2.3.3.1.1. Recolección de documentación

Se definieron como fuente de información:

- Lo estipulado en los T.D.R. (Términos de Referencia).
- Manuales de los sistemas heredados.
- Actas de acuerdo entre la Gerencia y Cliente, sobre la precisión de los requerimientos de la aplicación web.

##### 3.2.3.3.1.2. Análisis de la situación actual

EL objetivo es la descripción, entendimiento, causas de la situación actual de trabajo. Para ello se utilizó diversas técnicas:

- Observación pasiva, tomando notas del flujo de trabajo sin interferir en el desempeño laboral.
- Observación activa, realizando preguntas sobre las actividades relacionadas en el flujo de trabajo.
- Entrevistas, con los usuarios, líderes y cliente, para complementar la información obtenida.
- Reuniones grupales con todos los miembros del equipo de desarrollo, consultores, líderes, cliente, usuarios y gerencia para un entendimiento global, cruzar información y validar los requerimientos.

#### 3.2.3.3.2. Etapa análisis

Procedemos a interpretar las actas, las encuestas, las notas, términos de referencia y manuales, estos requerimientos son presentados al cliente para su aprobación.

##### 3.2.3.3.2.1. Flujos del proceso de negocio

###### 3.2.3.3.2.1.1. Mapa de interacción de procesos

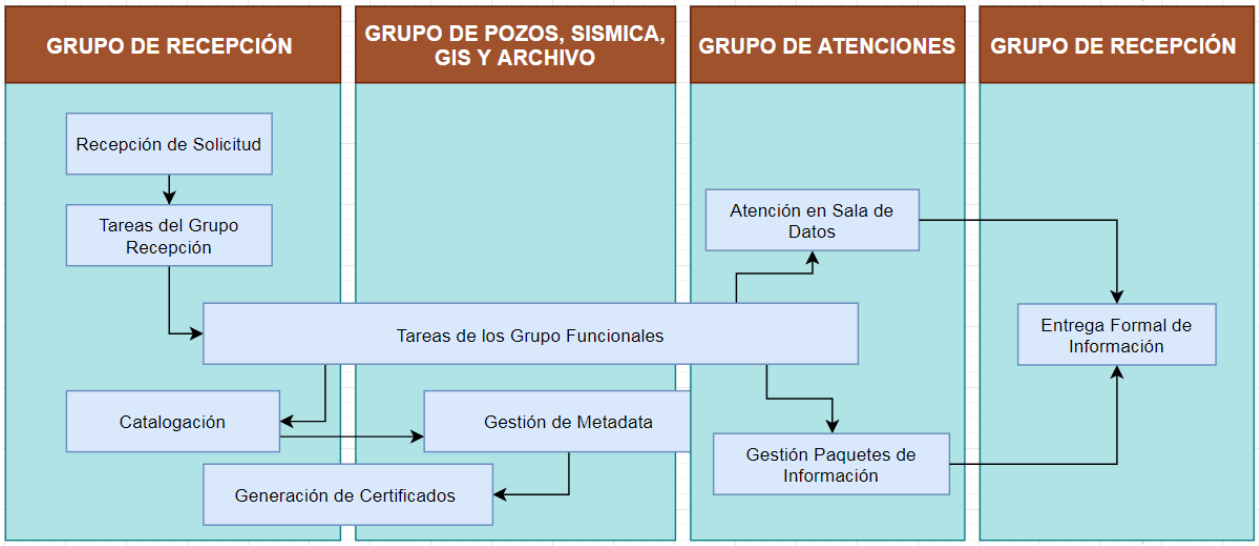



Diagrama 4: Mapa Proceso de un Banco de Datos

Fuente: Elaboración Propia (2021)

### 3.2.3.3.2.1.2. Proceso de recepción de información nueva

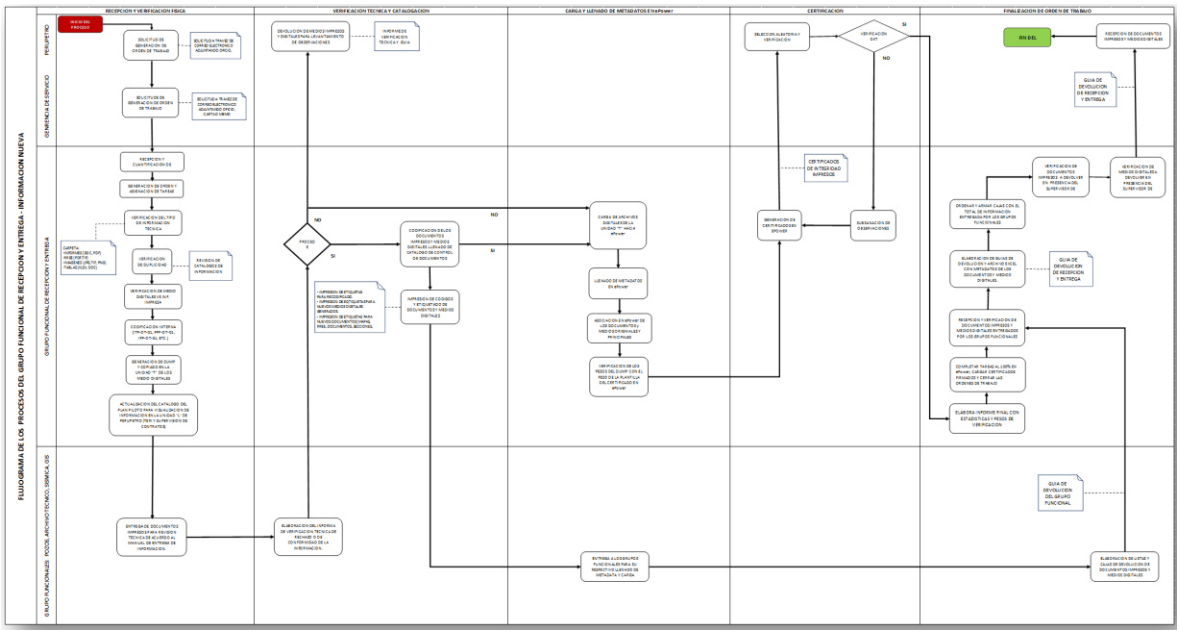



Diagrama 5. Flujo de Trabajo del Grupo Recepción – Información Nueva

Fuente: Halliburton (2019)

### 3.2.3.3.2.1.3. Proceso de recepción de información heredada

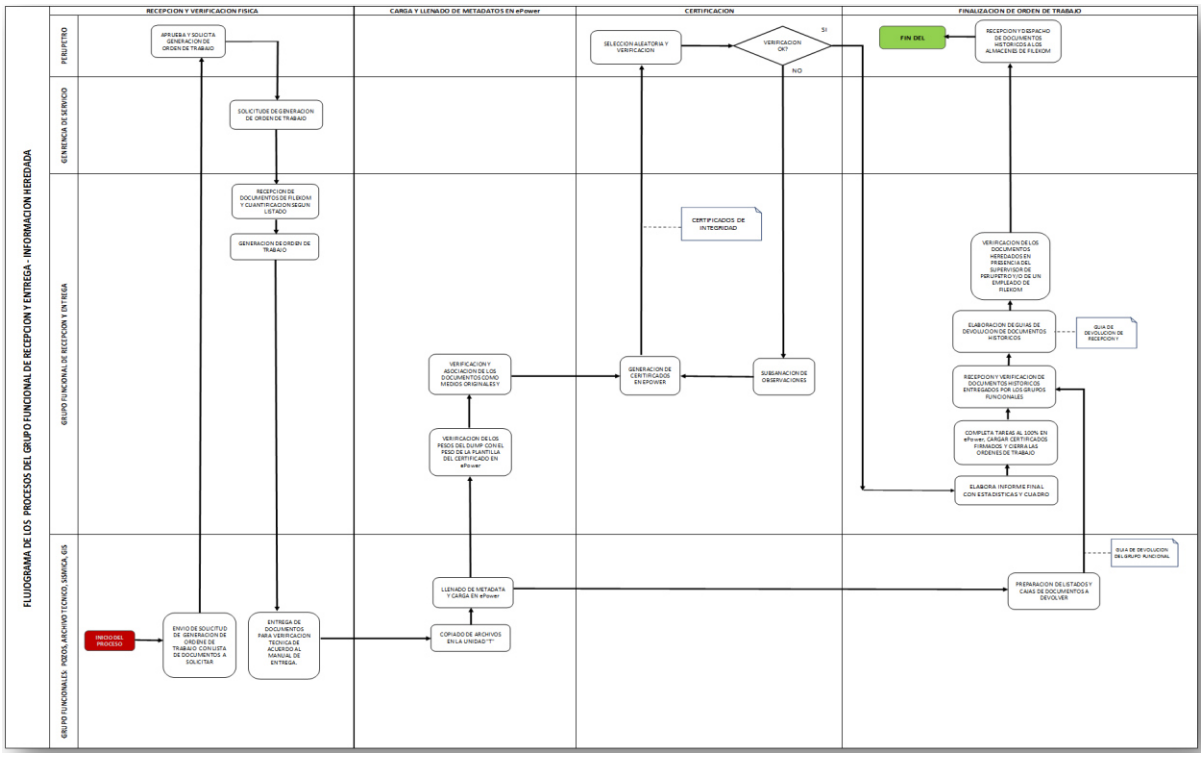



Diagrama 6: Flujo de Trabajo del Grupo Recepción – Información Heredada

Fuente: Halliburton (2019)

### 3.2.3.3.2.1.4. Proceso de verificación técnica, catalogación, certificación y carga de información nueva

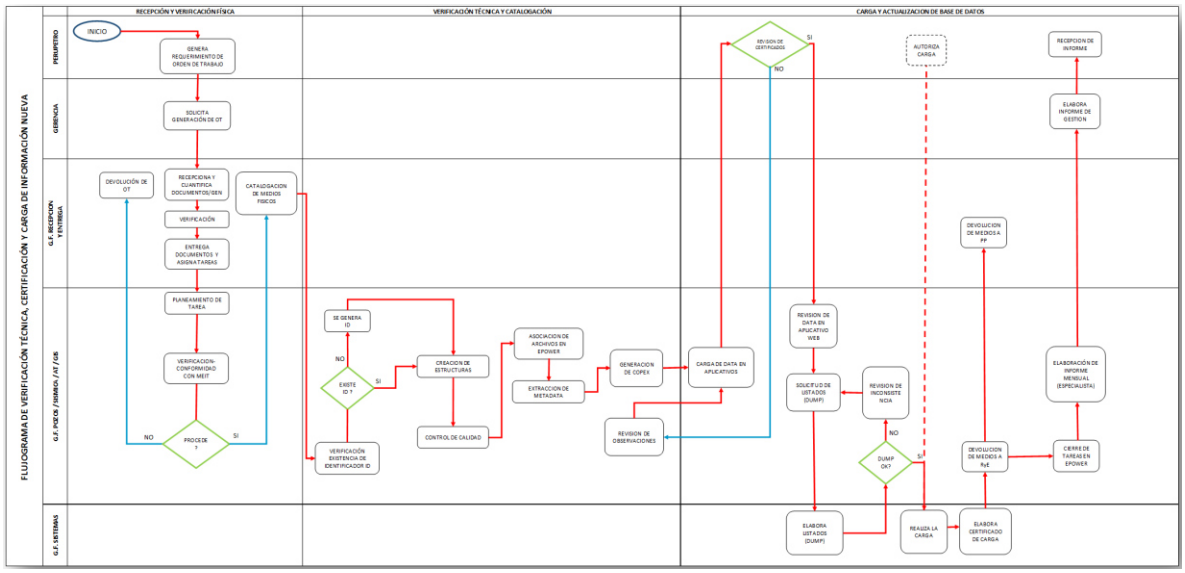



Diagrama 7: Flujo de Trabajo General – Caso Información Nueva
Fuente: Halliburton (2019)

### 3.2.3.3.2.1.5. Proceso de verificación técnica, catalogación, certificación y carga de información heredada

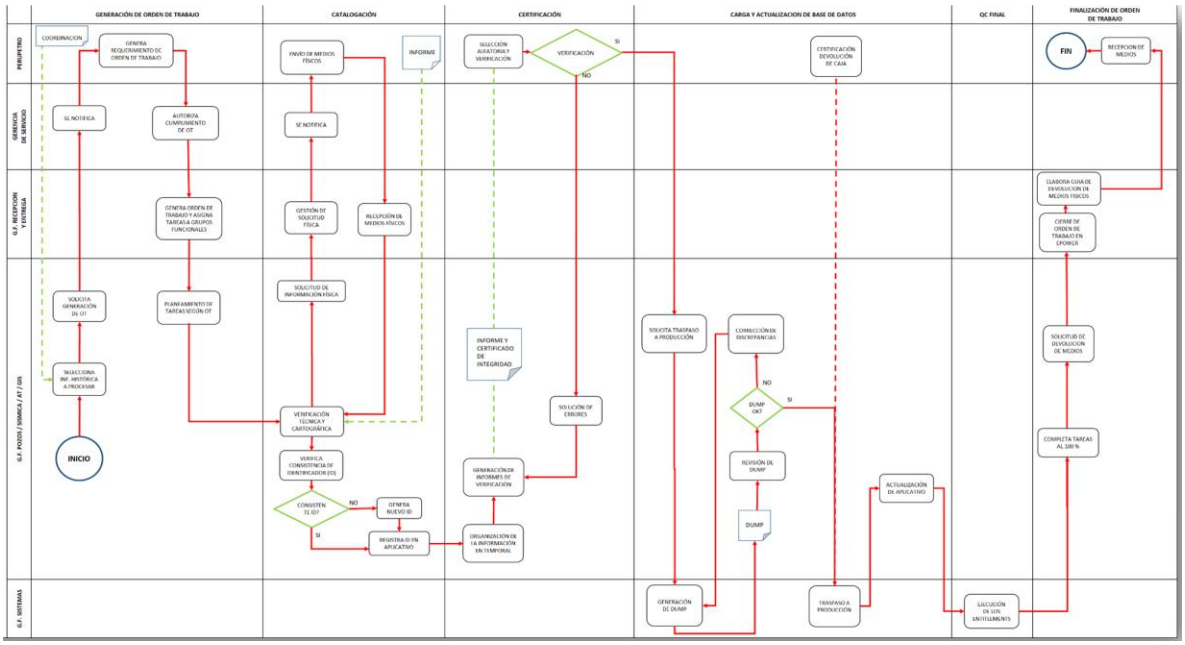



Diagrama 8: Flujo de Trabajo General – Caso Información Heredada
Fuente: Halliburton (2019)

### 3.2.3.3.2.1.6. Entrega de información técnica

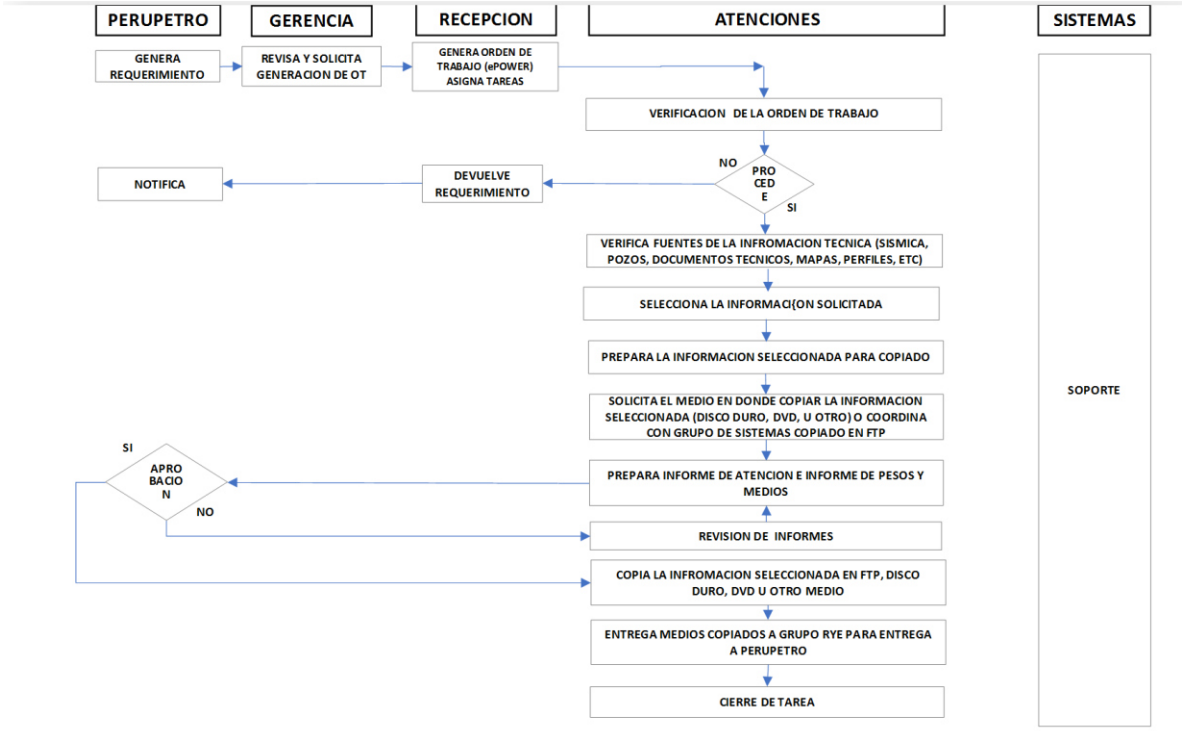



Diagrama 9: Flujo de Entrega de Información

Fuente: Halliburton (2019)

#### 3.2.3.3.2.2. Requerimientos funcionales

| Nro. | Requerimiento Funcional |
|---|---|
| RF001 | Listar Ordenes de Trabajo |
| RF002 | Consultar Orden de Trabajo |
| RF003 | Registrar Solicitud de Orden T. |
| RF004 | Registrar Dumps |
| RF005 | Actualizar Solicitud de Orden T. |
| RF006 | Registrar Seguimiento Doc. |
| RF007 | Registrar Tarea |
| RF008 | Actualizar Tarea |
| RF009 | Registrar Tracking de Tarea |
| RF010 | Consultar actividades |
| RF011 | Registrar Transferencia de Doc. |
| RF012 | Registrar Metadata Catalogo |
| RF013 | Registrar Metadata Gis |
| RF014 | Registrar Metadata Pozos |
| RF015 | Registrar Metadata Sísmica |
| RF016 | Registrar Metadata Archivo |
| RF017 | Generar Certificado Recepción |
| RF018 | Generar Certificado Gis |
| RF019 | Generar Certificado Sísmica |
| RF020 | Generar Certificado Archivo |
| RF021 | Actualizar Metadata Catalogo |
| RF022 | Actualizar Metadata Gis |
| RF023 | Actualizar Metadata Pozos |
| RF024 | Actualizar Metadata Sísmica |
| RF025 | Actualizar Metadata Archivo |
| RF026 | Eliminar Metadata Catalogo |
| RF027 | Eliminar Metadata GIS |
| RF028 | Eliminar Metadata Sísmica |
| RF029 | Eliminar Metadata Archivo |
| RF030 | Actualizar Metadata Catalogo |
| RF031 | Eliminar Metadata Pozo |
| RF032 | Listar Códigos Medios Digitales |
| RF033 | Listar Códigos Barcodes |
| RF034 | Listar Lotes |
| RF035 | Listar Pozos |
| RF036 | Reporte de Tareas |
| RF037 | Reporte de Certificados |
| RF038 | Reporte de Ordenes de Trabajo |
| RF039 | Búsqueda Medio Digital |
| RF040 | Búsqueda Código Barcode |

Tabla 2: Lista de Requerimientos Funcionales

Fuente Elaboración Propia (Año 2021)

#### 3.2.3.3.2.3. Requerimientos no funcionales

| Requerimiento | Descripción |
|---|---|
| Acceso | ✓ El acceso de la aplicación web no es público, está restringido dentro de la red del Banco de Datos<br>✓ Acceso por Roles a las interfaces<br>✓ Acceso por credenciales de autenticación. |
| Auditoria | ✓ Registrar fecha de creación, actualización y/o eliminación de algún dato del sistema<br>✓ Registrar datos del usuario que registro, actualizo y/o elimino de algún dato del sistema |
| Alertas | ✓ Notificar por email la creación de una orden o de una tarea<br>✓ Notificar por email las ordenes que tienen menos de 5 días por vencer el plazo de entrega<br>✓ Alertas de data duplicada en memorándum, código barcode, pozo, medio digital, etc. |
| Integración | ✓ Permite procesar transacciones en tiempo real<br>✓ Permitir consultar y registrar datos de otras aplicaciones del banco de datos |
| Facilidad de uso | ✓ Estandarizar las interfaces, tener un sentido de continuidad<br>✓ Facilitar Manuales de usuario<br>✓ Filtros simples y combinados para búsqueda de información<br>✓ Exportar los datos en formato Excel |
| Rendimiento y Escalabilidad | ✓ La data cargada será por demanda<br>✓ Tiempos de respuesta adecuada para las funciones habituales<br>✓ Tiempo de 5 segundos como máximo para consultar, buscar o filtrar datos. |
| Disponibilidad | ✓ Disponible 99% por Mes |
| Despliegue | ✓ Las fallas de software o hardware debe resultar posible devolver el sistema a un estado conocido. |
| Infraestructura | ✓ Servidor virtual de Windows 2012 R2 con una capacidad de 200 GB de almacenamiento, memoria RAM de 56 GB, procesador Intel Xeon Gold 6136 @ 3.00 GHz (8 procesos). |
| Tecnología | ✓ Framework Angular, Spring Boot, Kendo Telerik y PostgreSQL |

Tabla 3: Requerimiento No Funcionales

Fuente Elaboración Propia (Año 2021)

#### 3.2.3.3.3. Etapa procedimiento

##### 3.2.3.3.3.1. Lista de actores

###### 3.2.3.3.3.1.1. Definición de Actores

| Actores del Sistema | |
|---|---|
| Actor 1 | |
| Identificación | AC-01 |
| Nombre | Administrador |
| Actividades | Funciones de configuración del sistema.<br>Gestiona Usuarios.<br>Gestiona Gerencias.<br>Gestiona las Actividades de las Tareas.<br>Gestiona Grupos Funcionales. |
| Actor 2 | |
| Identificación | AC-02 |
| Nombre | Gerente Servicio |
| Actividades | Consulta Ordenes de Trabajo<br>Consulta Tareas de cada Orden de Trabajo<br>Consulta las Alertas de vencimiento de Tareas y de OTs.<br>Consulta Reportes y Dashboards. |
| Actor 3 | |
| Identificación | AC-03 |
| Nombre | Supervisor del Banco de Datos |
| Actividades | Consulta Ordenes de Trabajo<br>Consulta Tareas de cada Orden de Trabajo<br>Consulta las Alertas de vencimiento de Tareas y de OTs.<br>Consulta Seguimiento Documental.<br>Consulta Reportes y Dashboards.<br>Consulta Registro de Dumps.<br>Consulta Certificados de Calidad.<br>Aprueba Certificados de Calidad.<br>Verificación de Metadata. |

Tabla 4: Actores del Sistema Nivel Gerencia

Fuente Elaboración Propia (Año 2021)

| Actores del Sistema | |
|---|---|
| Actor 4 | |
| Identificación | AC-04 |
| Nombre | Lideres |
| Actividades | Funciones de Registrar Tareas, Actualizar Tareas, Cargar Metadata, Registrar, Actualizar y Eliminar Metadata, Generar Certificados y Consultar Reportes. |
| Actor 6 | |
| Identificación | AC-06 |
| Nombre | Líder de Recepción y Entrega |
| Actividades | Funciones de Registrar Tareas, Actualizar Tareas, Cargar Metadata, Registrar, Actualizar y Eliminar Metadata de Catalogo, Generar Certificados de Recepción y Consultar Reportes. |
| Actor 7 | |
| Identificación | AC-07 |
| Nombre | Líder de Archivo Técnico |
| Actividades | Funciones de Registrar Tareas, Actualizar Tareas, Cargar Metadata, Registrar, Actualizar y Eliminar Metadata de Archivo, Generar Certificados de Archivo y Consultar Reportes. |
| Actor 8 | |
| Identificación | AC-08 |
| Nombre | Líder de Pozos |
| Actividades | Funciones de Registrar Tareas, Actualizar Tareas, Cargar Metadata, Registrar, Actualizar y Eliminar Metadata de Pozos, Generar Certificados de Pozos y Consultar Reportes. |
| Actor 9 | |
| Identificación | AC-09 |
| Nombre | Líder de Sísmica |
| Actividades | Funciones de Registrar Tareas, Actualizar Tareas, Cargar Metadata, Registrar, Actualizar y Eliminar Metadata de Sísmica, Generar Certificados de Sísmica y Consultar Reportes. |
| Actor 10 | |
| Identificación | AC-10 |
| Nombre | Líder de Gis |
| Actividades | Funciones de Registrar Tareas, Actualizar Tareas, Cargar Metadata, Registrar, Actualizar y Eliminar Metadata de GIS, Generar Certificados de GIS y Consultar Reportes. |
| Actor 11 | |
| Identificación | AC-11 |
| Nombre | Líder de Atenciones |
| Actividades | Funciones de Registrar Tareas, Actualizar Tareas. |

Tabla 5: Actores del Sistema Nivel Lideres

Fuente Elaboración Propia (Año 2021)

| Actores del Sistema | |
|---|---|
| Actor 12 | |
| Identificación | AC-12 |
| Nombre | Consultores |
| Actividades | Funciones de carga de Metadata, completitud de Metadata, verificación de Metadata, generación de certificados y consultar reportes. |
| Actor 13 | |
| Identificación | AC-13 |
| Nombre | Consultor de Recepción y Entrega |
| Actividades | Funciones de Actualizar tareas,<br>Cargar Metadata, Registrar, Actualizar y Eliminar Metadata de Recepción, Generar Certificados de Recepción y Consultar Reportes. |
| Actor 14 | |
| Identificación | AC-014 |
| Nombre | Consultor de Archivo Técnico |
| Actividades | Funciones de Actualizar tareas,<br>Cargar Metadata, Registrar, Actualizar y Eliminar Metadata de Archivo, Generar Certificados de Archivo y Consultar Reportes. |
| Actor 15 | |
| Identificación | AC-15 |
| Nombre | Consultor de Pozos |
| Actividades | Funciones de Actualizar tareas,<br>Cargar Metadata, Registrar, Actualizar y Eliminar Metadata de Pozos, Generar Certificados de Pozos y Consultar Reportes. |
| Actor 16 | |
| Identificación | AC-16 |
| Nombre | Consultor de Sísmica |
| Actividades | Funciones de Actualizar tareas,<br>Cargar Metadata, Registrar, Actualizar y Eliminar Metadata de Sísmica, Generar Certificados de Sísmica y Consultar Reportes. |
| Actor 17 | |
| Identificación | AC-17 |
| Nombre | Consultor de Gis |
| Actividades | Funciones de Actualizar tareas,<br>Cargar Metadata, Registrar, Actualizar y Eliminar Metadata de GIS, Generar Certificados de GIS y Consultar Reportes. |
| Actor 18 | |
| Identificación | AC-18 |
| Nombre | Consultor de Atenciones |
| Actividades | Funciones de Actualizar tareas y Consultar Reportes. |

Tabla 6: Actores del Sistema Nivel Consultores

Fuente Elaboración Propia (Año 2021)

### 3.2.3.3.3.1.2. Diagrama de Actores

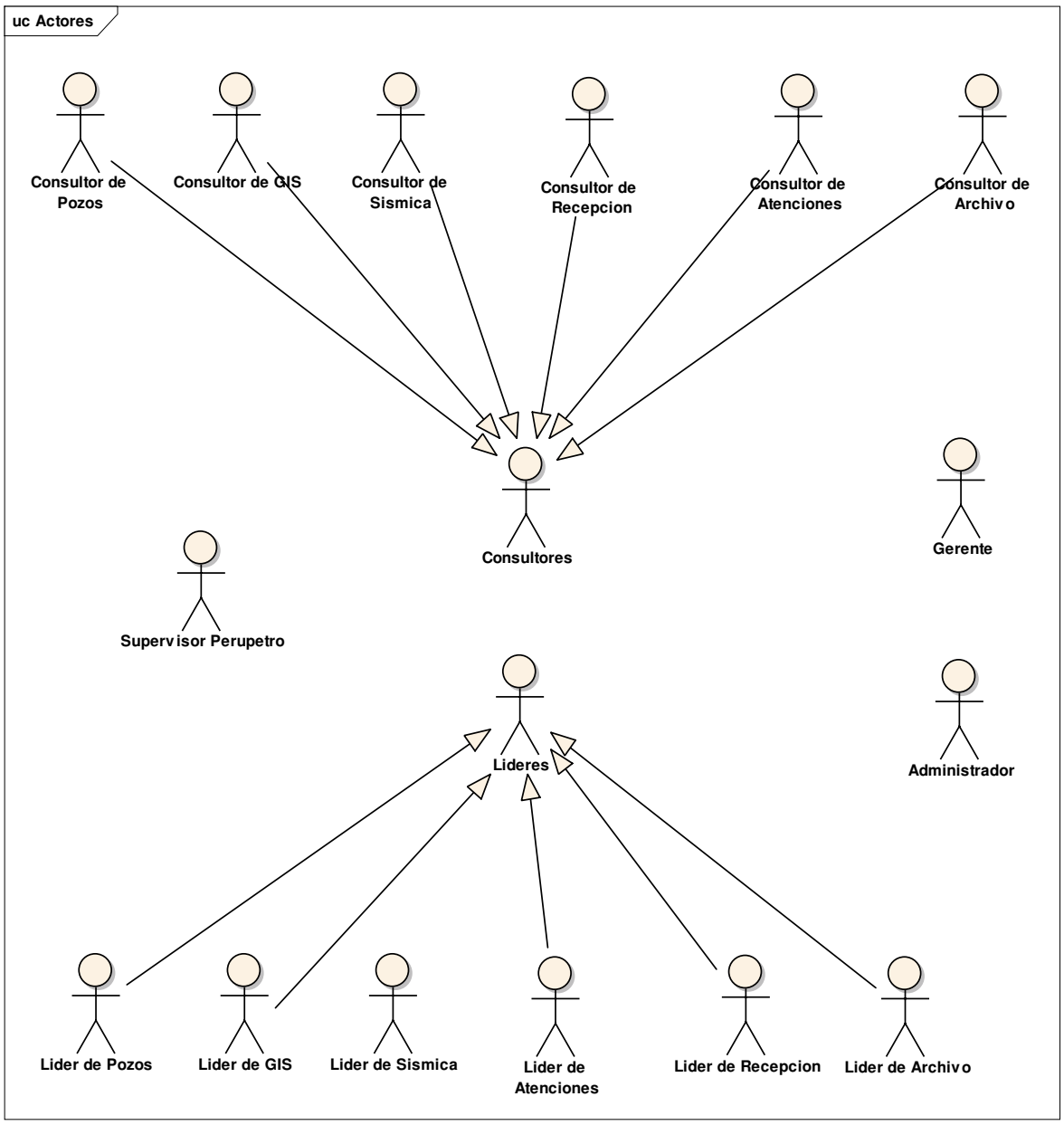



Diagrama 10: Diagrama de Actores del Banco de Datos

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.2. Diagrama de paquetes

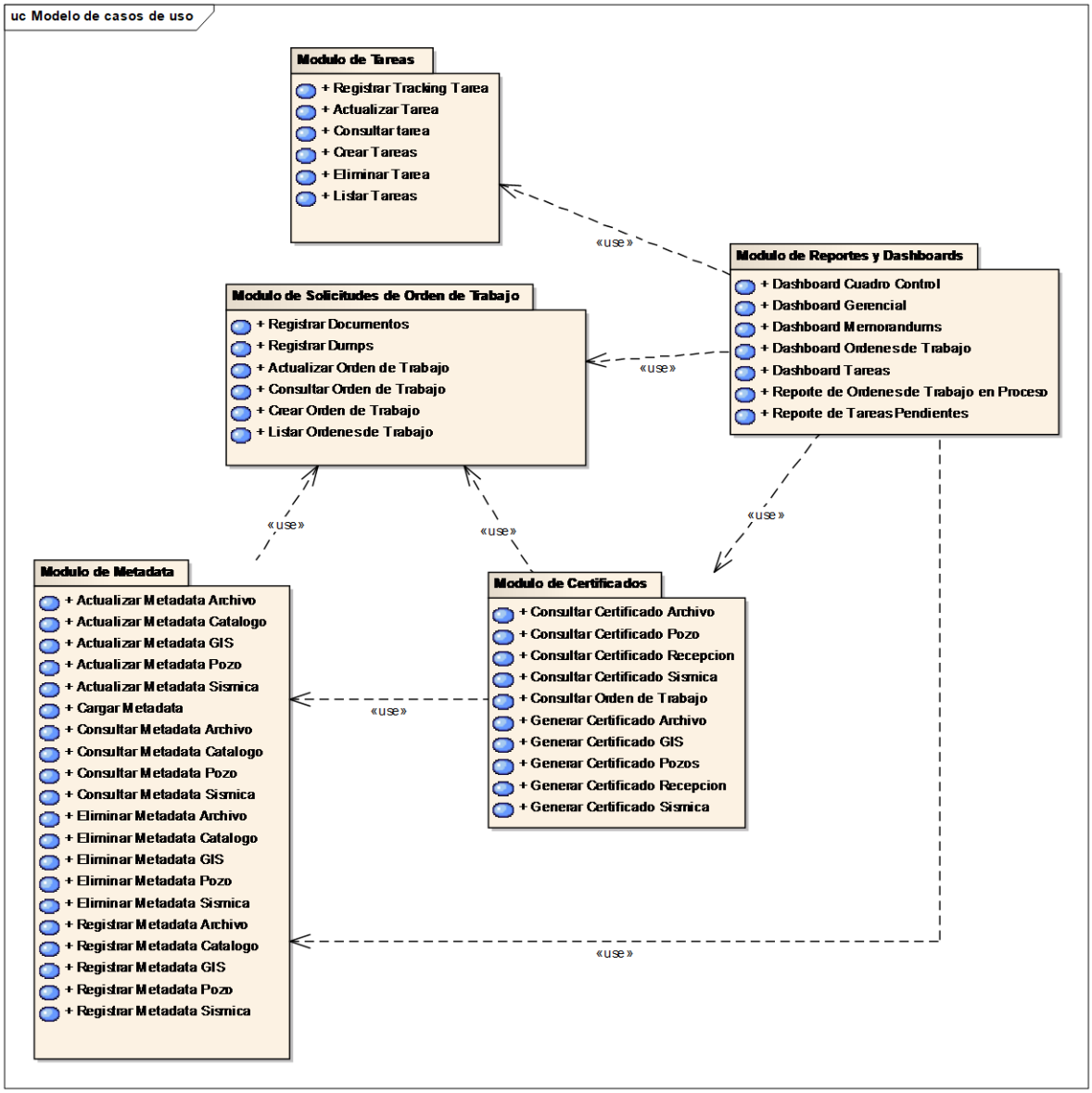



Diagrama 11: Diagrama de Paquetes de la Aplicación.

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3. Diagramas de casos de uso (D.C.U)

En esta sección se presentará los principales diagramas de casos de uso identificados.

#### 3.2.3.3.3.3.1. D.C.U – Listar de órdenes de trabajo.

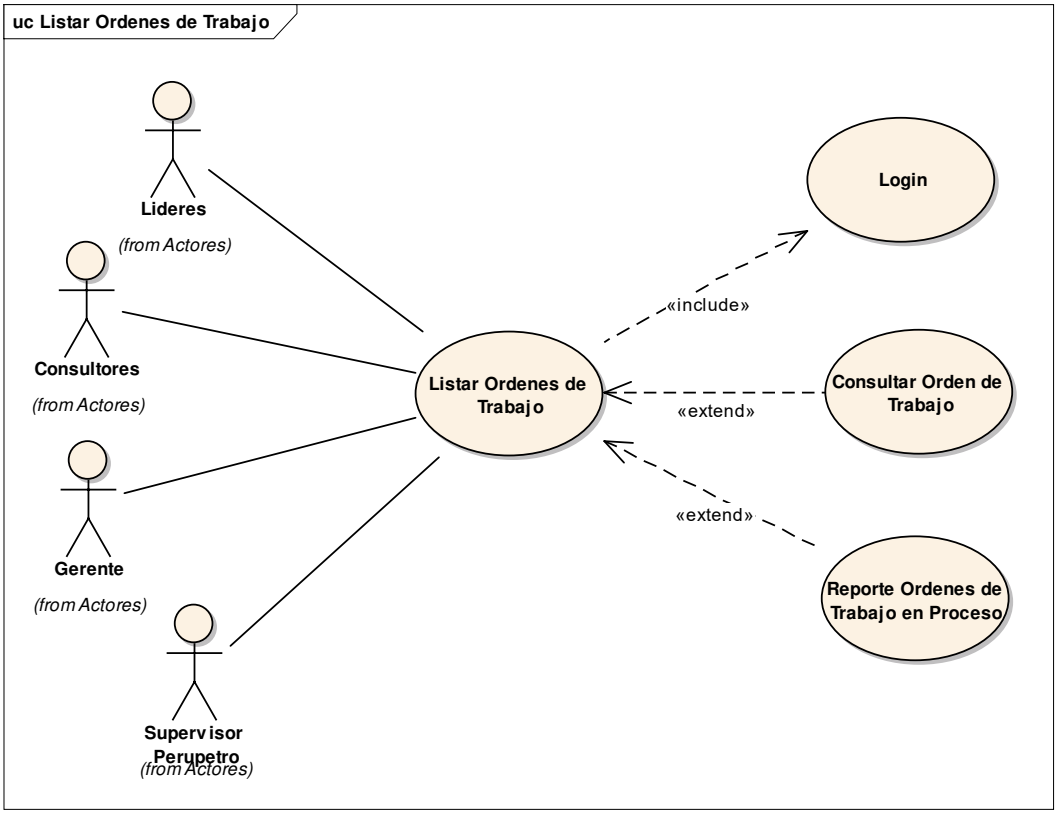



Diagrama 12: Diagrama de Caso de Uso Listar Ordenes de Trabajo

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.2. D.C.U – Consultar orden de trabajo.

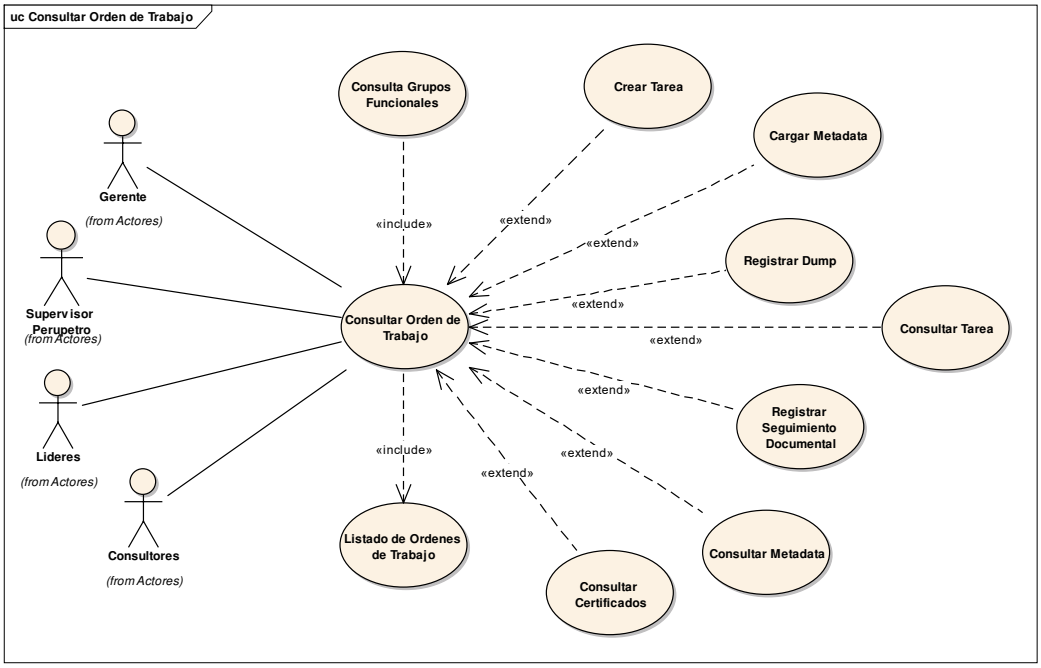



Diagrama 13: Diagrama de Caso de Uso Consultar Orden de Trabajo

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.3. D.C.U – Registrar orden de trabajo

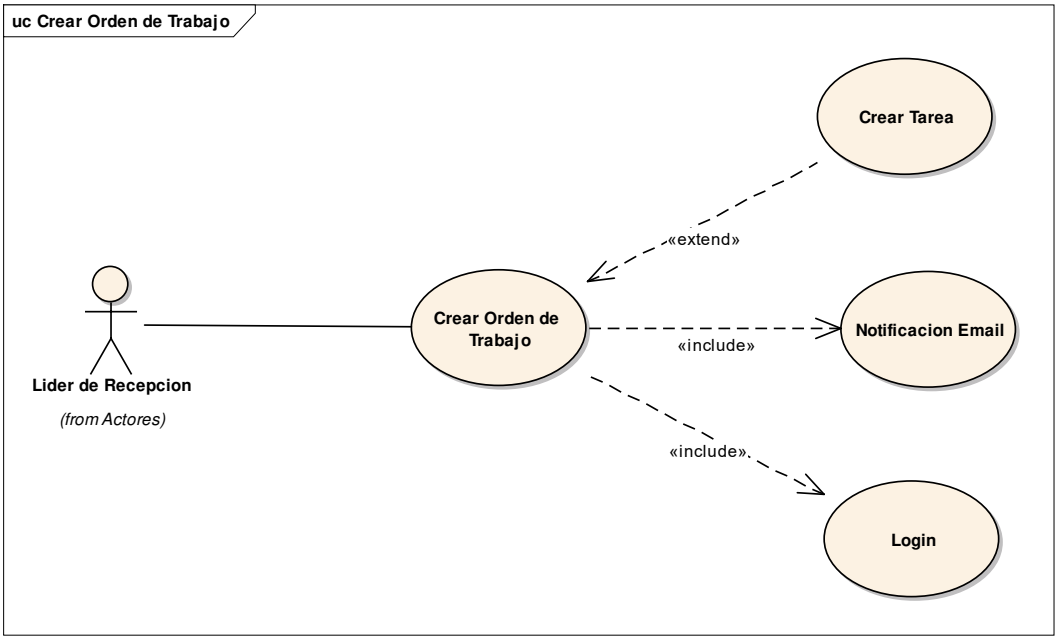



Diagrama 14: Diagrama de Caso de Uso Registrar Orden de Trabajo

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.4. D.C.U – Actualizar orden de trabajo.

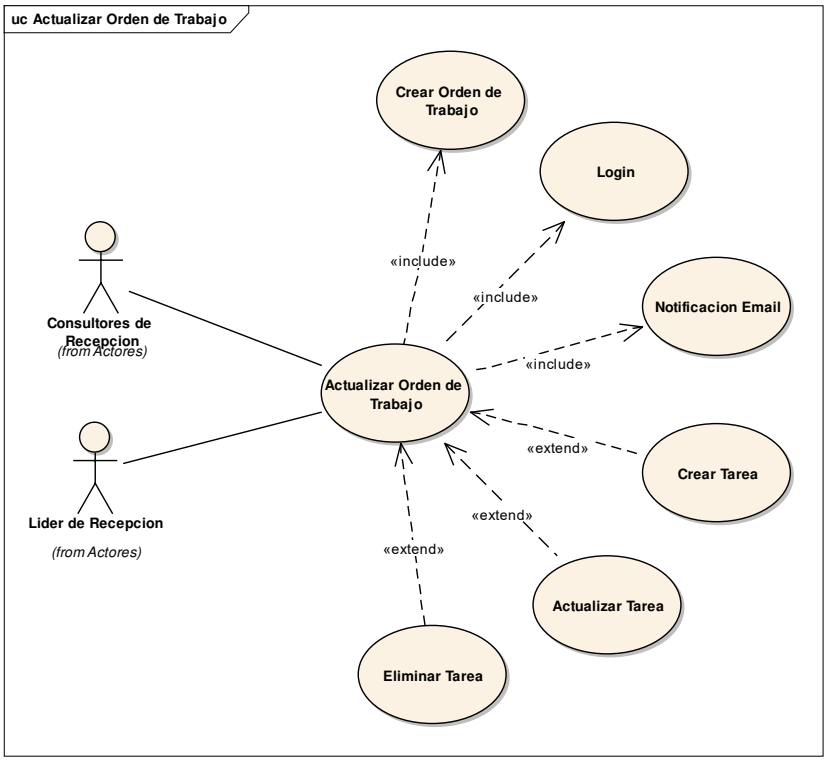



Diagrama 15: Diagrama de Caso de Uso Actualizar Orden de Trabajo

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.5. D.C.U – Registrar seguimiento documental

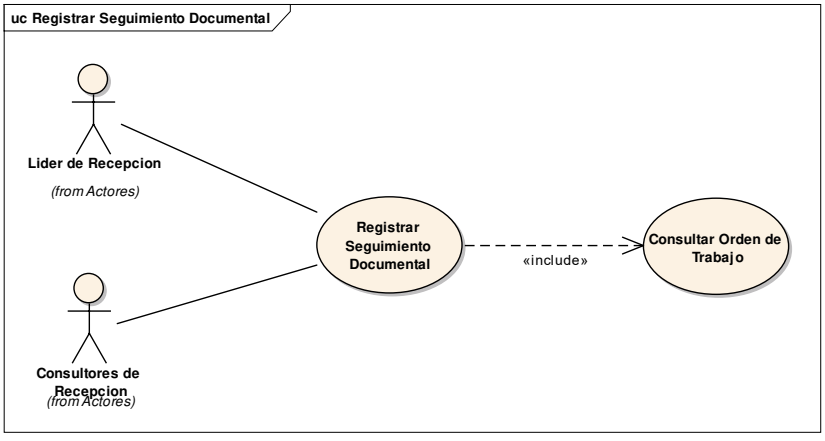



Diagrama 16: Diagrama de Caso de Uso Registrar Seguimiento Documental

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.6. D.C.U – Registrar dump

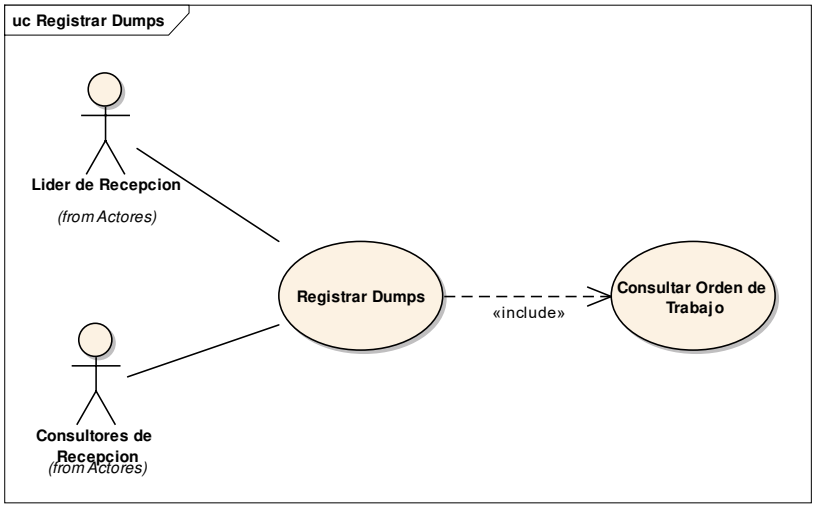



Diagrama 17: Diagrama de Caso de Uso Registrar Dumps

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.7. D.C.U – Listar tareas.

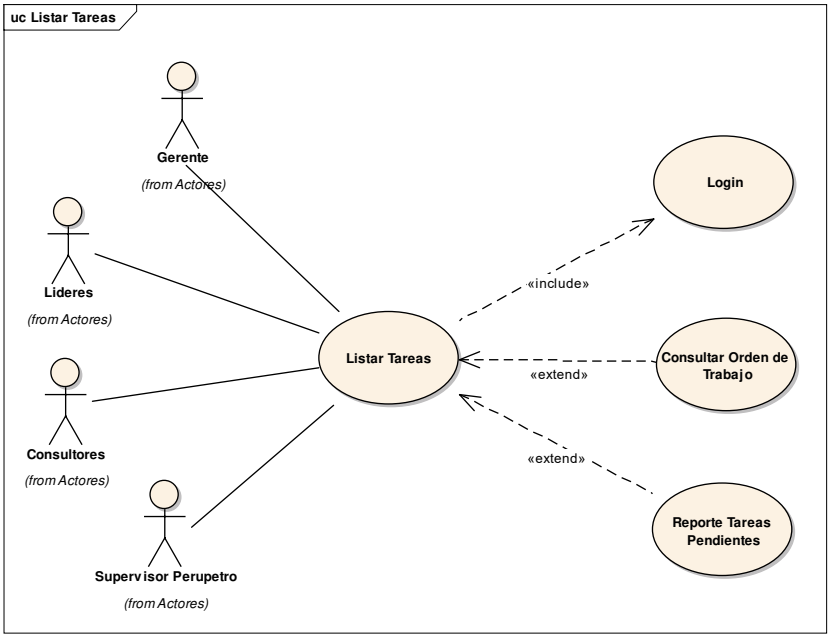



Diagrama 18: Diagrama de Caso de Uso Listar Tareas

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.8. D.C.U – Consultar tareas.

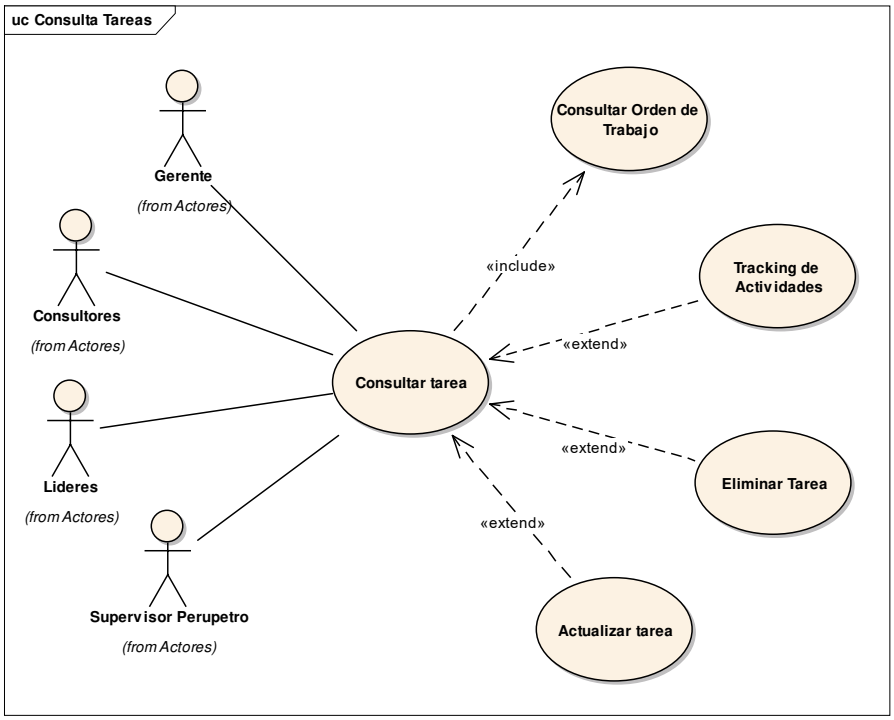



Diagrama 19: Diagrama de Caso de Uso Consultar Tarea

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.9. D.C.U – Registrar tareas.

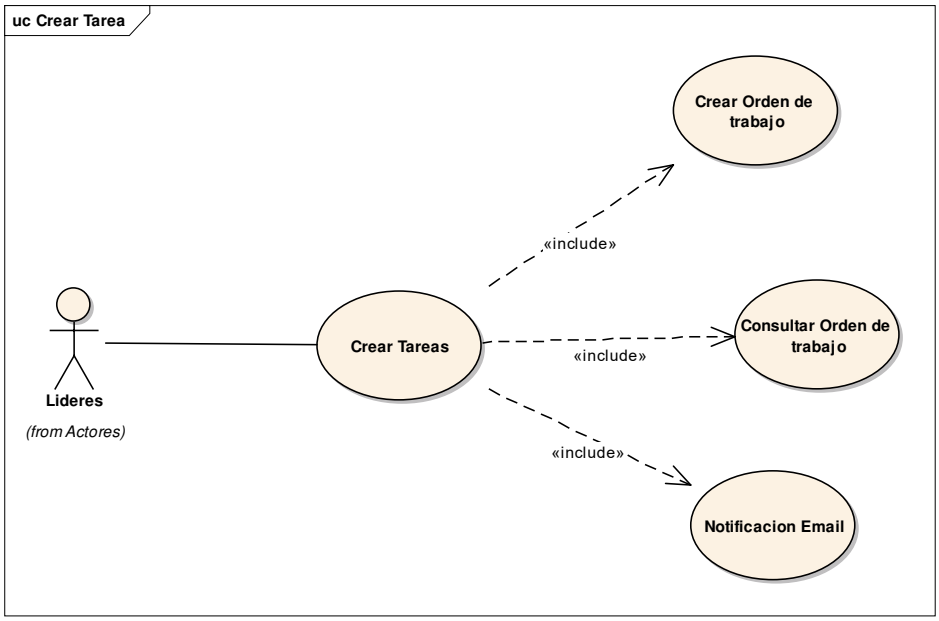



Diagrama 20: Diagrama de Caso de Uso Crear Tareas

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.10. D.C.U – Actualizar tarea.

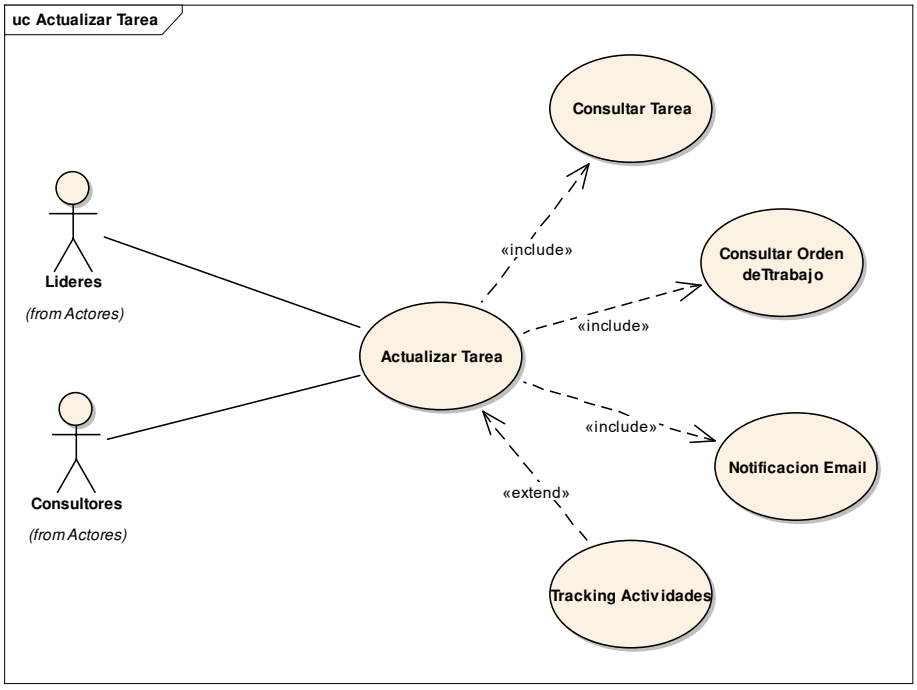



Diagrama 21: Diagrama de Caso de Uso Actualizar Tarea

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.11. D.C.U – Eliminar tarea.

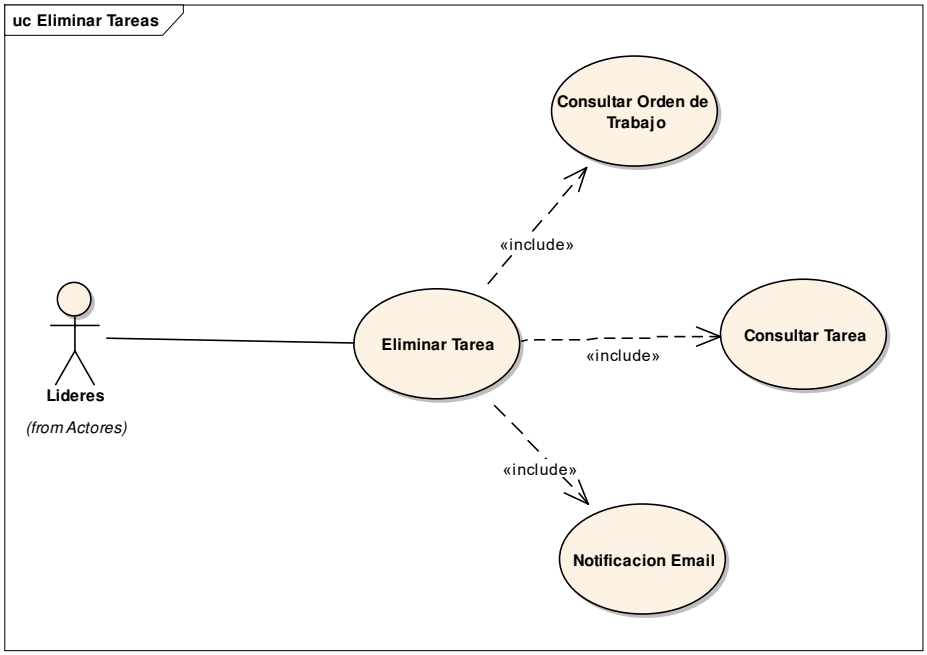



Diagrama 22: Diagrama de Caso de Uso Eliminar Tarea

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.12. D.C.U – Registrar tracking tarea

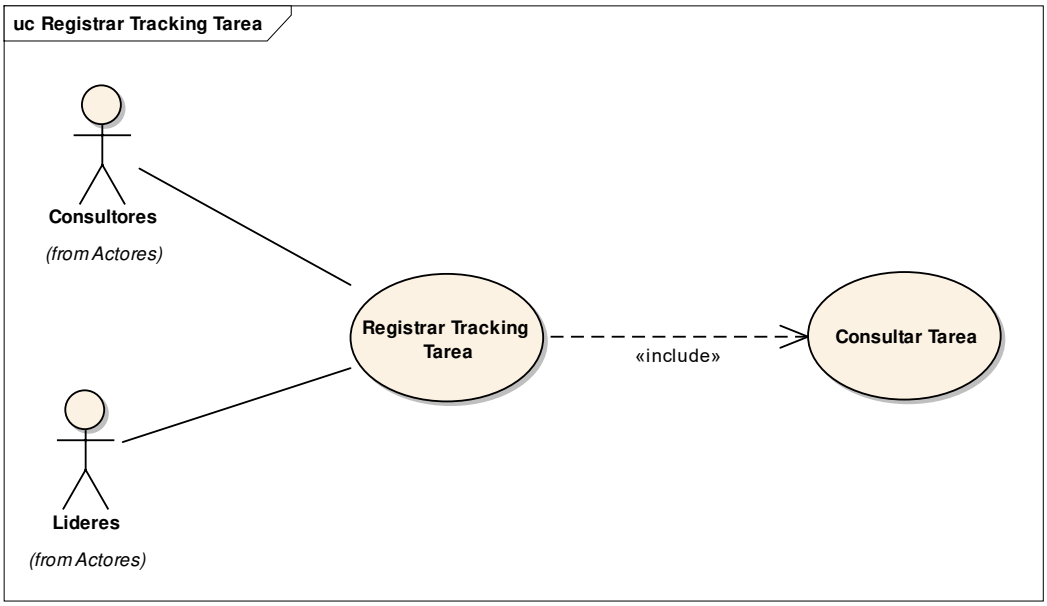



Diagrama 23: Diagrama de Caso de Uso Registrar Tracking Tarea

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.13. D.C.U – Cargar metadata.

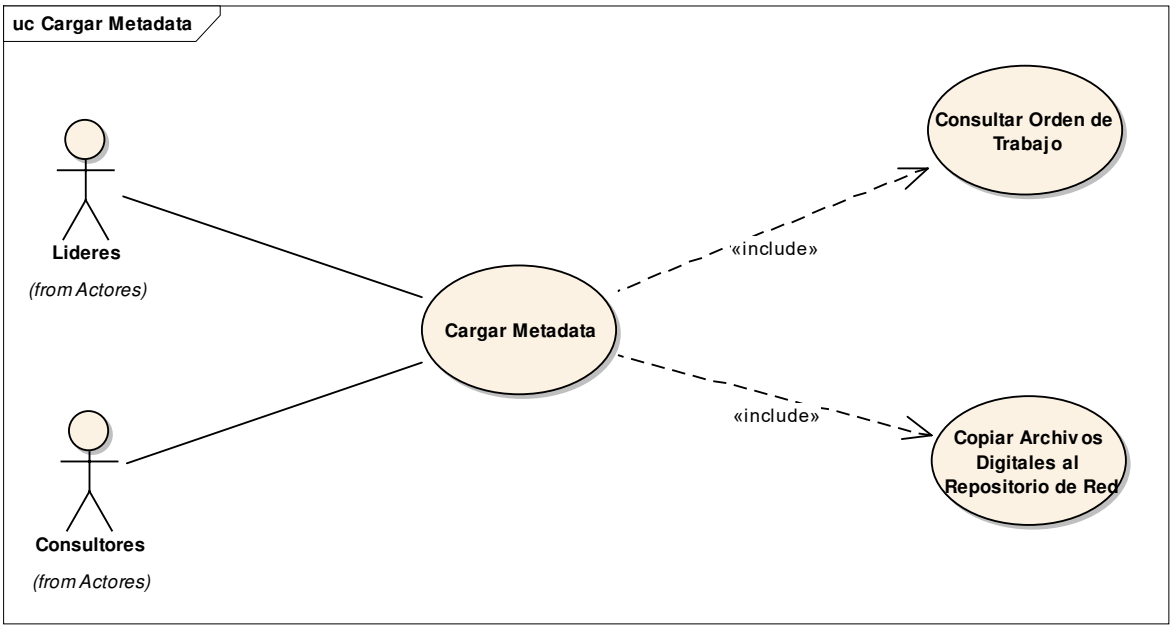



Diagrama 24: Diagrama de Caso de Uso Cargar Metadata

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.14. D.C.U – Registrar metadata en catálogo.

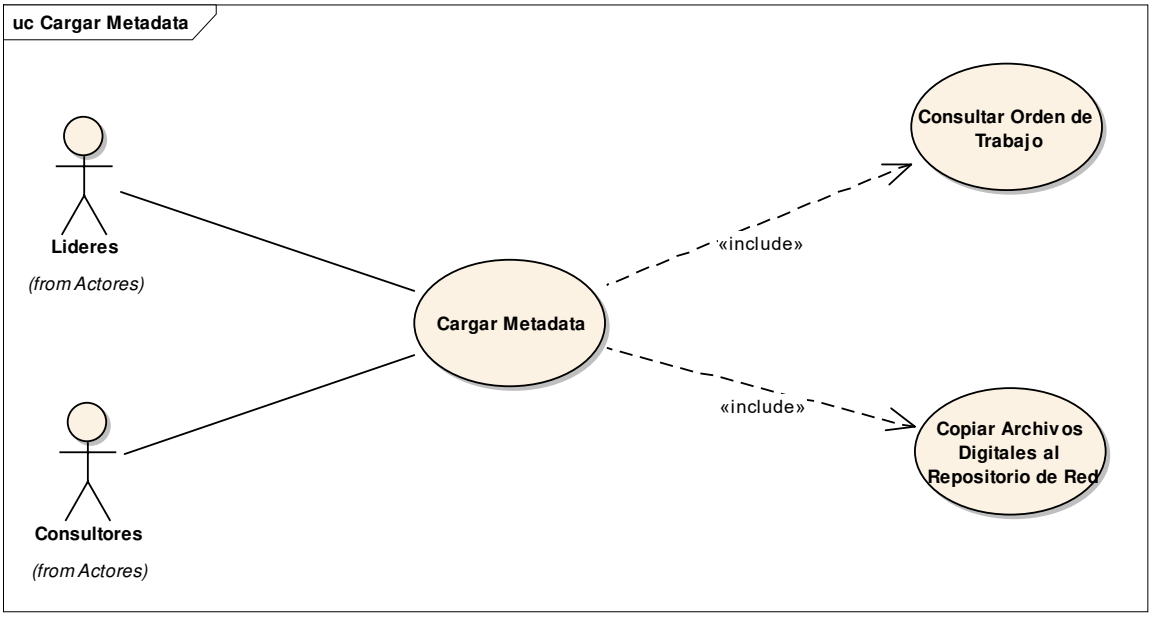



Diagrama 25: Diagrama de Caso de Uso Registrar Metadata para Catalogo

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.15. D.C.U – Registrar metadata en archivo.

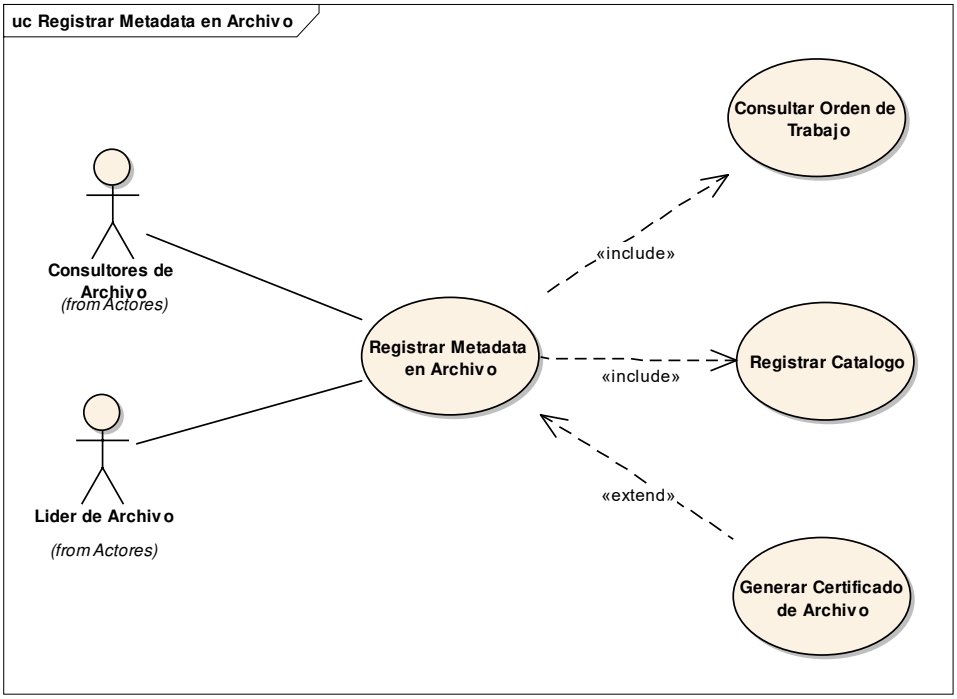



Diagrama 26: Diagrama de Caso de Uso Registrar Metadata para Archivo Técnico

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.16. D.C.U – Registrar metadata en GIS.

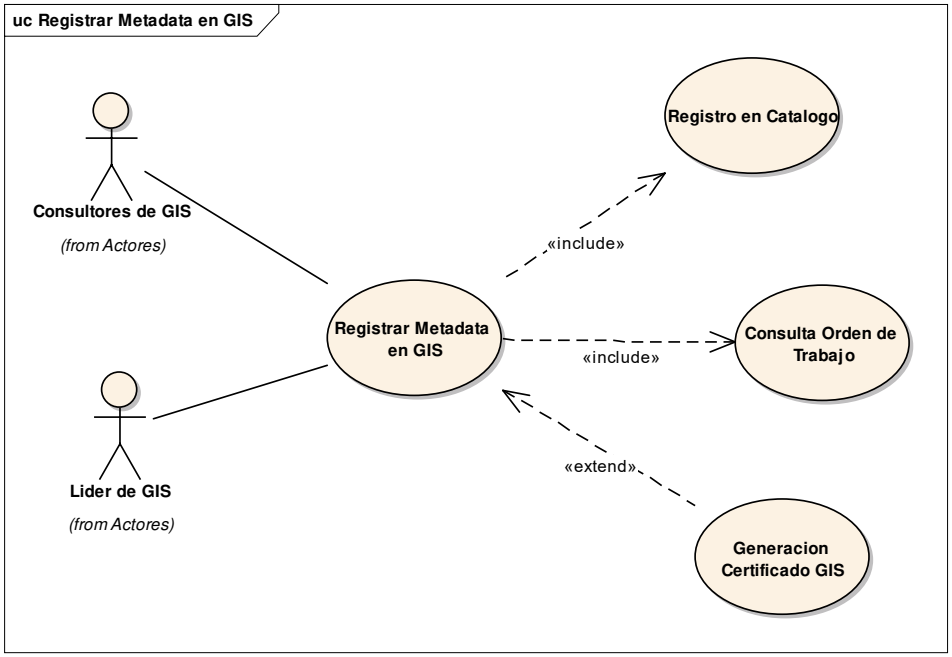



Diagrama 27: Diagrama de Caso de Uso Registrar Metadata para GIS

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.17. D.C.U – Registrar metadata en pozos.

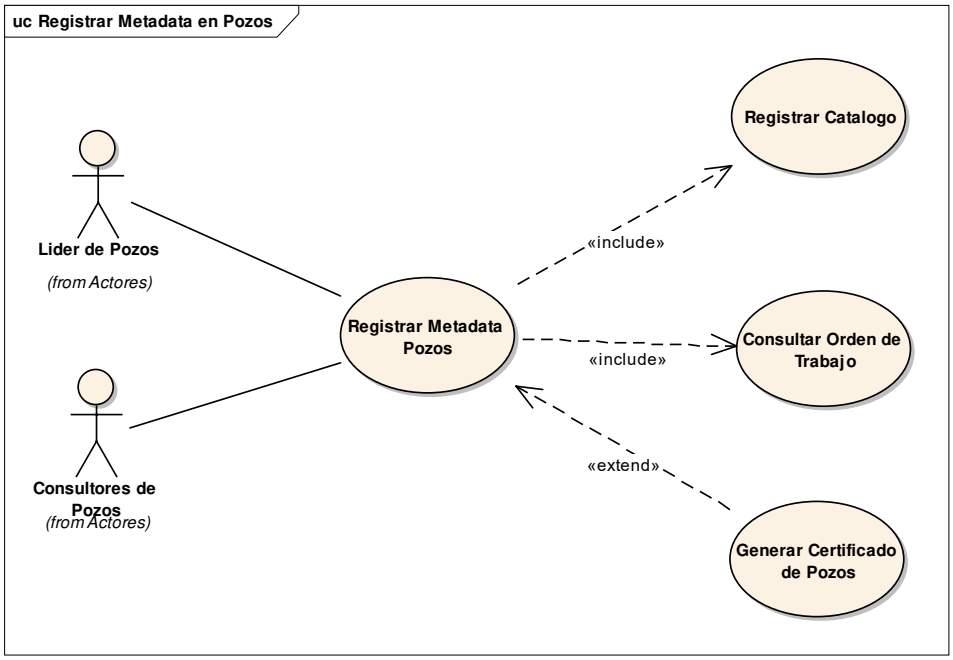



Diagrama 28 Diagrama de Caso de Uso Registrar Metadata para Pozos

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.18. D.C.U – Registrar metadata en sísmica.

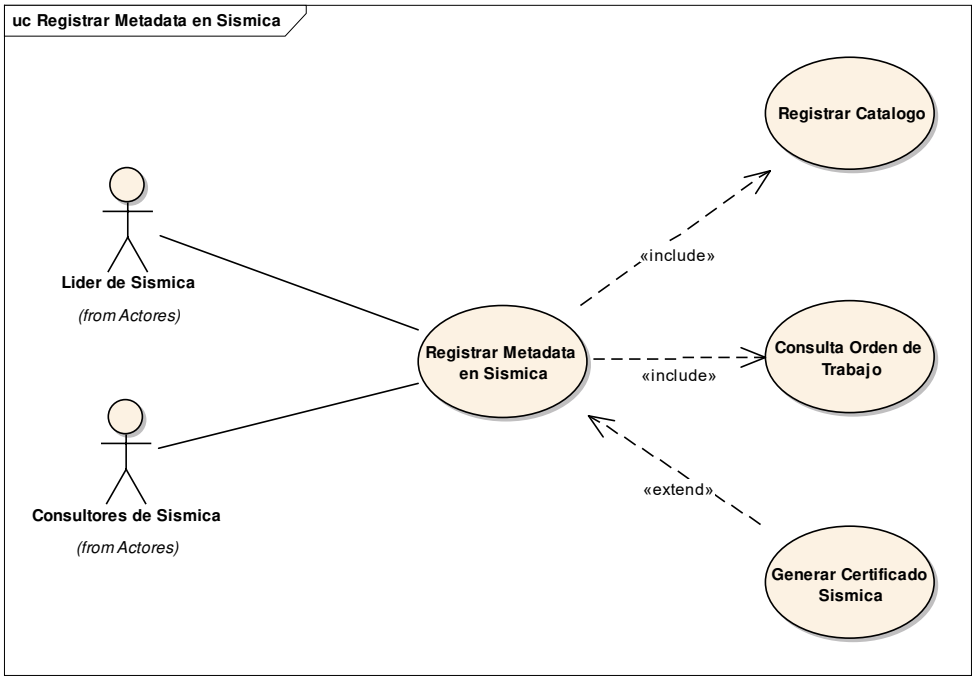



Diagrama 29: Diagrama de Caso de Uso Registrar Metadata para Sísmica

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.19. D.C.U – Actualizar metadata de catálogo.

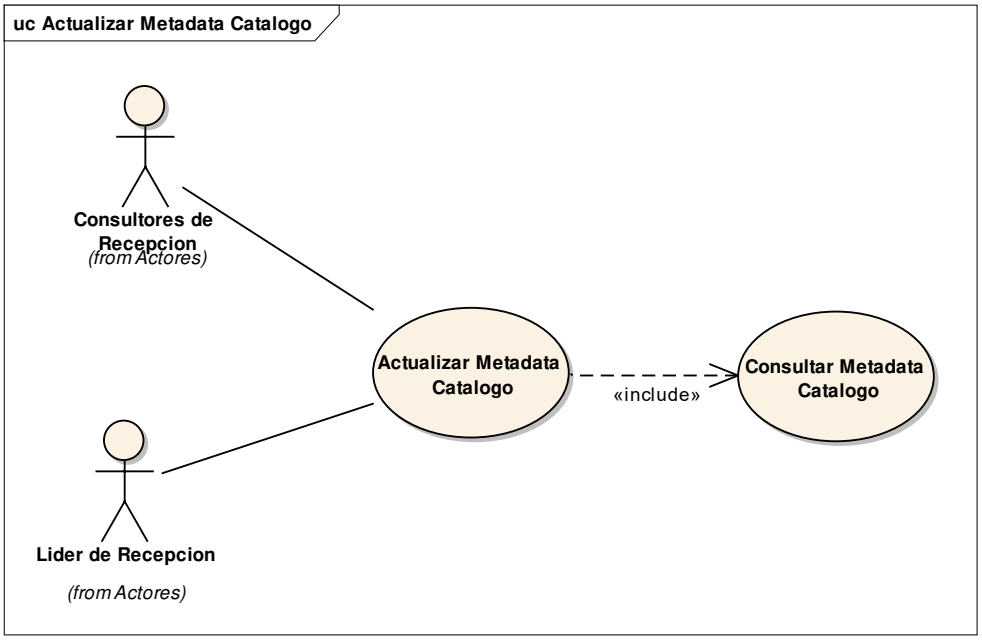



Diagrama 30: Diagrama de Caso de Uso Actualizar Metadata para Catalogo

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.20. D.C.U – Actualizar metadata de archivo.

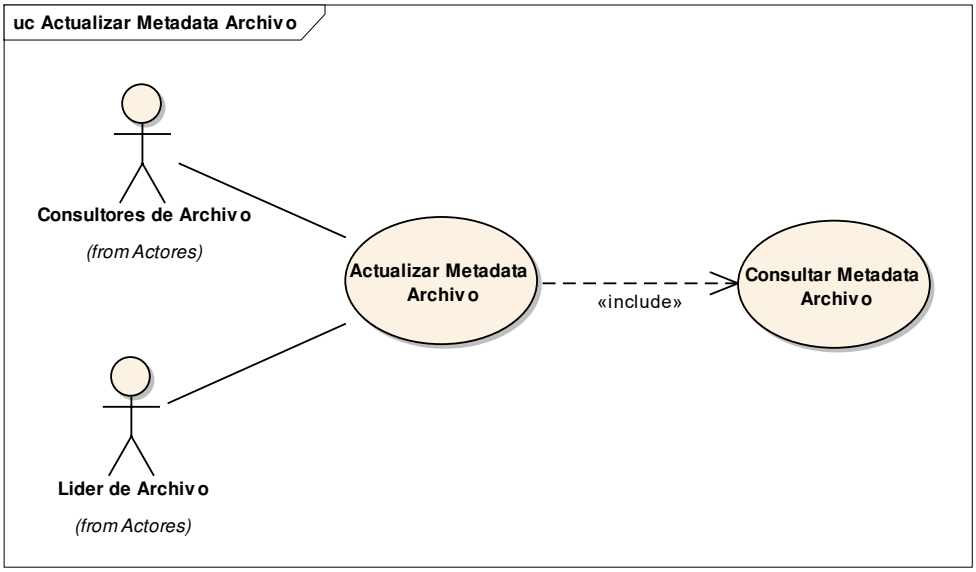



Diagrama 31: Diagrama de Caso de Uso Actualizar Metadata para Archivo

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.21. D.C.U – Actualizar metadata de GIS.

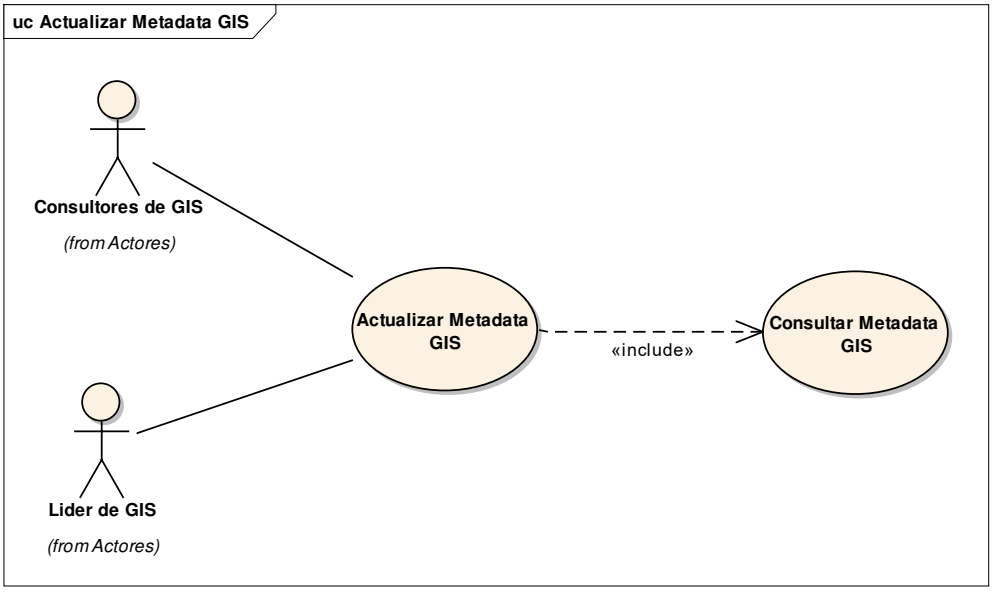



Diagrama 32: Diagrama de Caso de Uso Actualizar Metadata para GIS

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.22. D.C.U – Actualizar metadata de pozos.

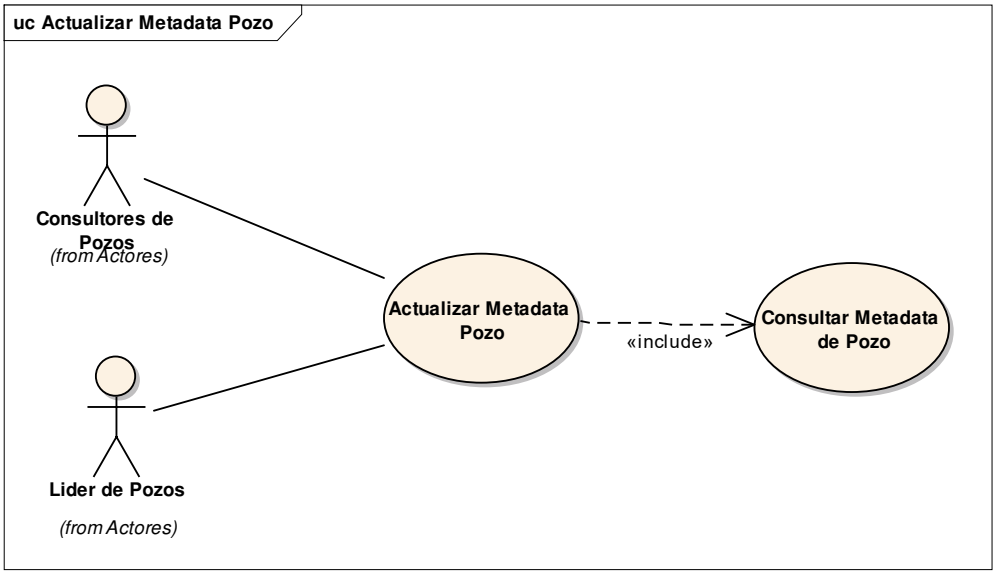



Diagrama 33: Diagrama de Caso de Uso Actualizar Metadata para Pozos

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.23. D.C.U – Actualizar metadata de sísmica.

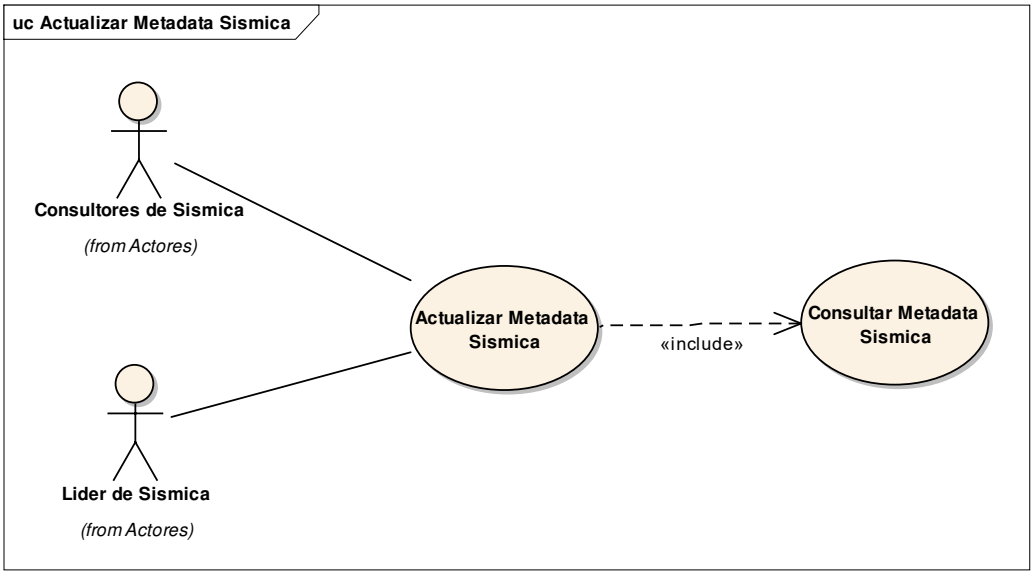



Diagrama 34: Diagrama de Caso de Uso Actualizar Metadata para Sísmica

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.24. D.C.U – Eliminar metadata de catálogo.

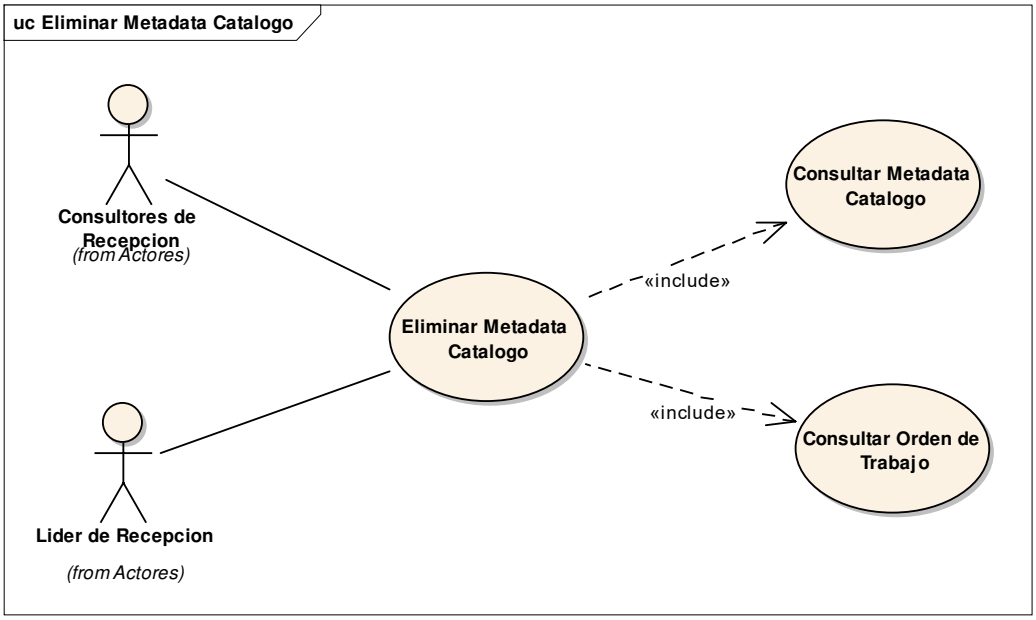



Diagrama 35: Diagrama de Caso de Uso Eliminar Metadata para Catalogo

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.25. D.C.U – Eliminar metadata de archivo.

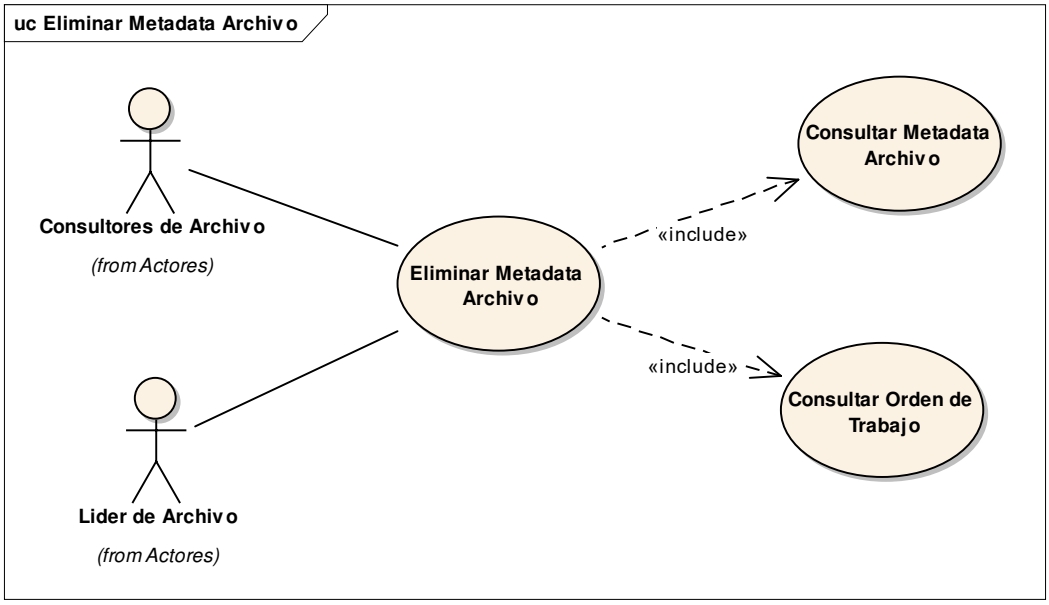



Diagrama 36: Diagrama de Caso de Uso Eliminar Metadata para Archivo

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.26. D.C.U – Eliminar metadata de GIS.

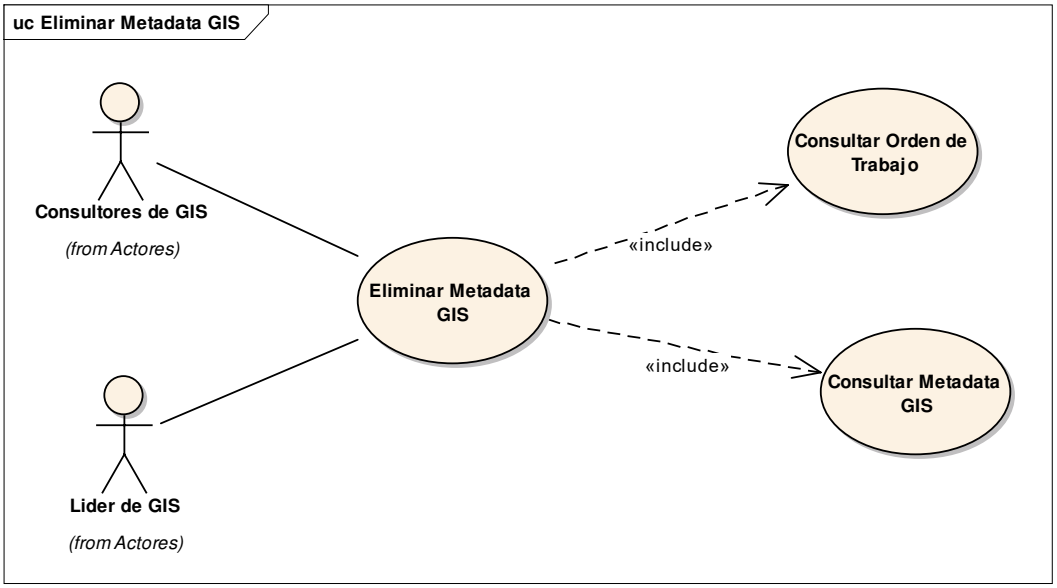



Diagrama 37: Diagrama de Caso de Uso Eliminar Metadata para GIS

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.27. D.C.U – Eliminar metadata de pozos.

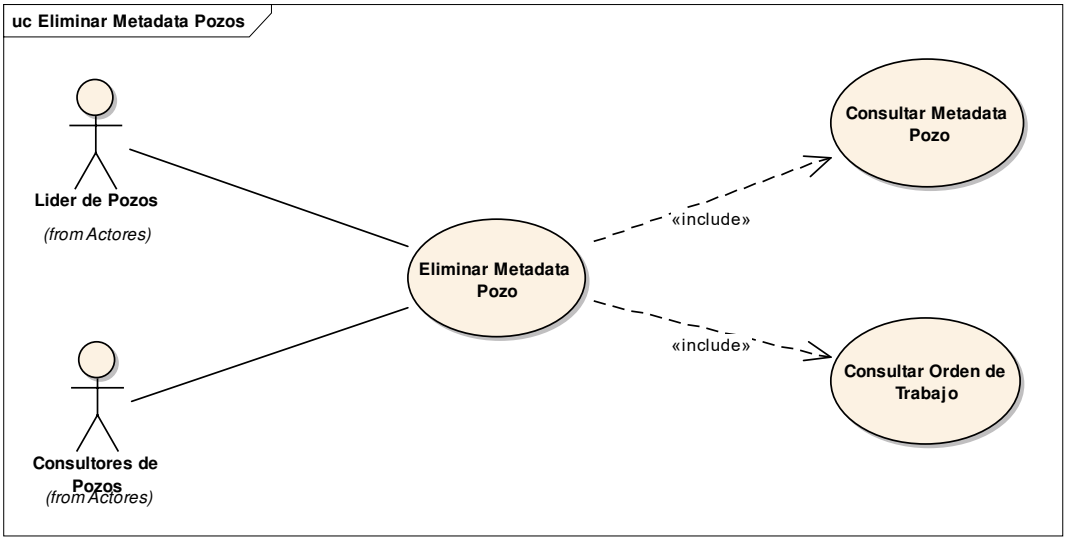



Diagrama 38: Diagrama de Caso de Uso Eliminar Metadata para Pozos

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.28. D.C.U – Eliminar metadata de sísmica.

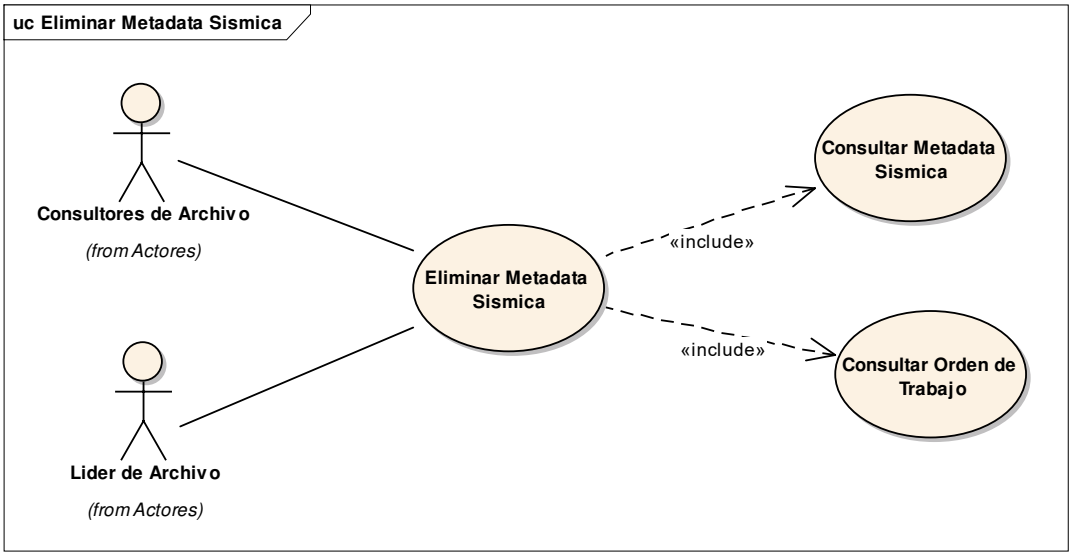



Diagrama 39: Diagrama de Caso de Uso Eliminar Metadata para Sísmica

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.29. D.C.U – Generar certificado recepción.

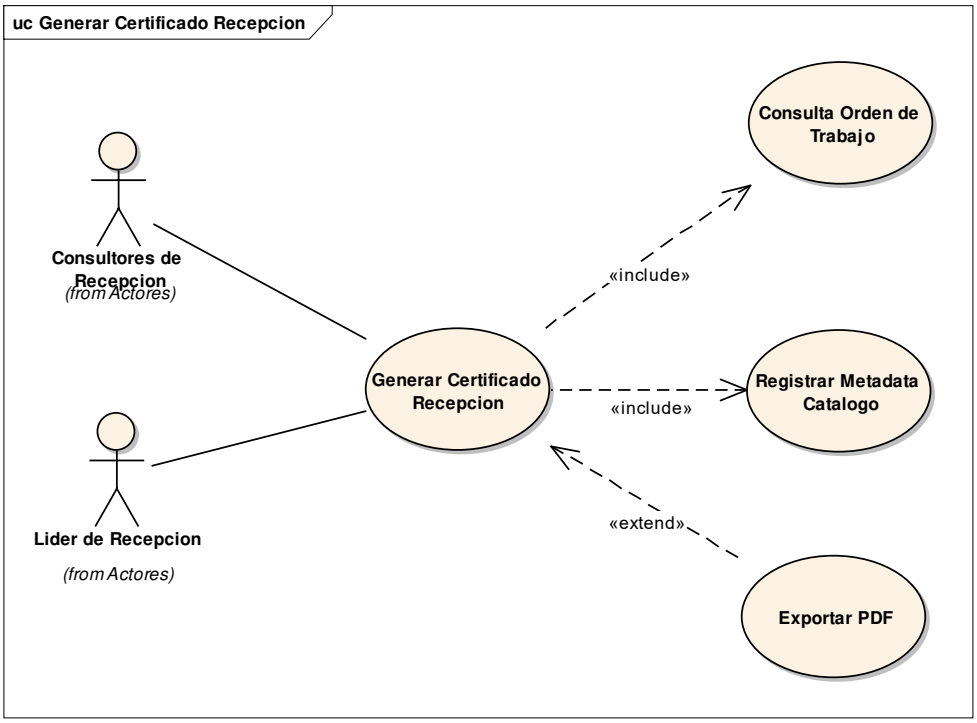



Diagrama 40: Diagrama de Caso de Uso Generar Certificado para Recepción

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.30. D.C.U – Generar certificado pozos.

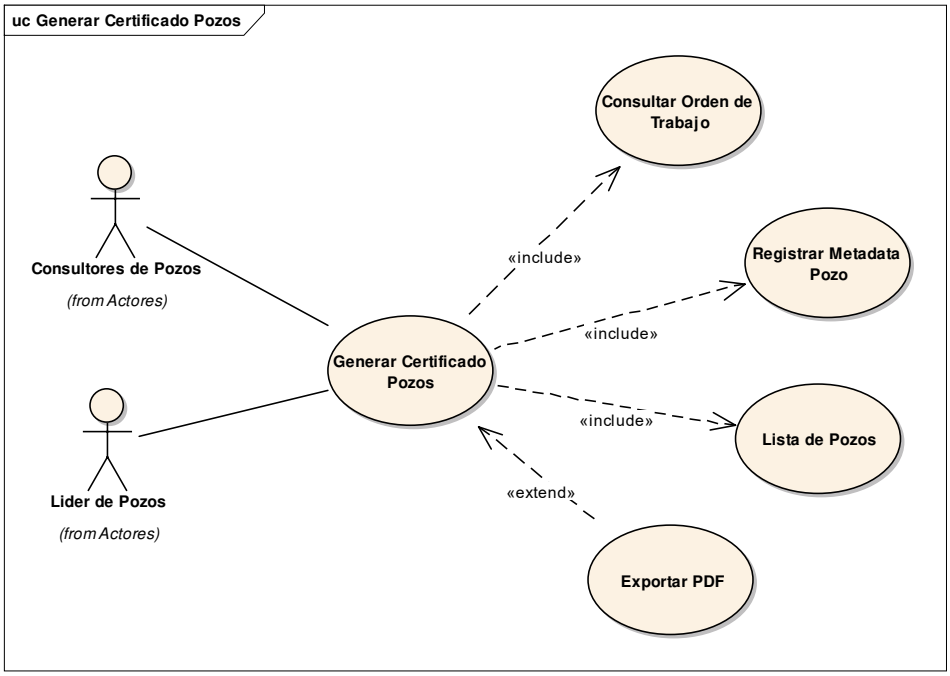



Diagrama 41: Diagrama de Caso de Uso Generar Certificado para Pozos

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.31. D.C.U – Generar certificado gis.

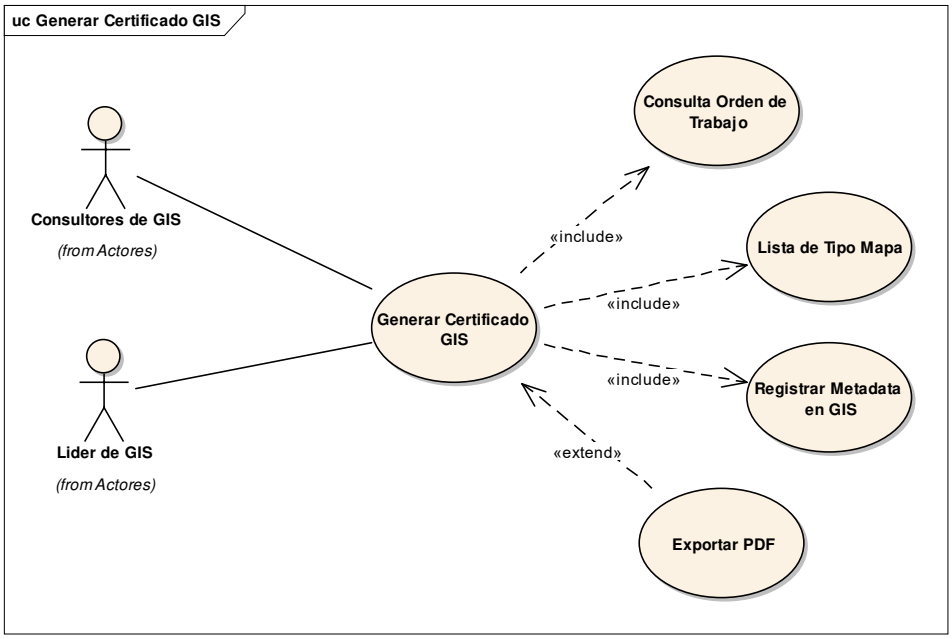



Diagrama 42: Diagrama de Caso de Uso Generar Certificado para GIS

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.32. D.C.U – Generar certificado archivo.

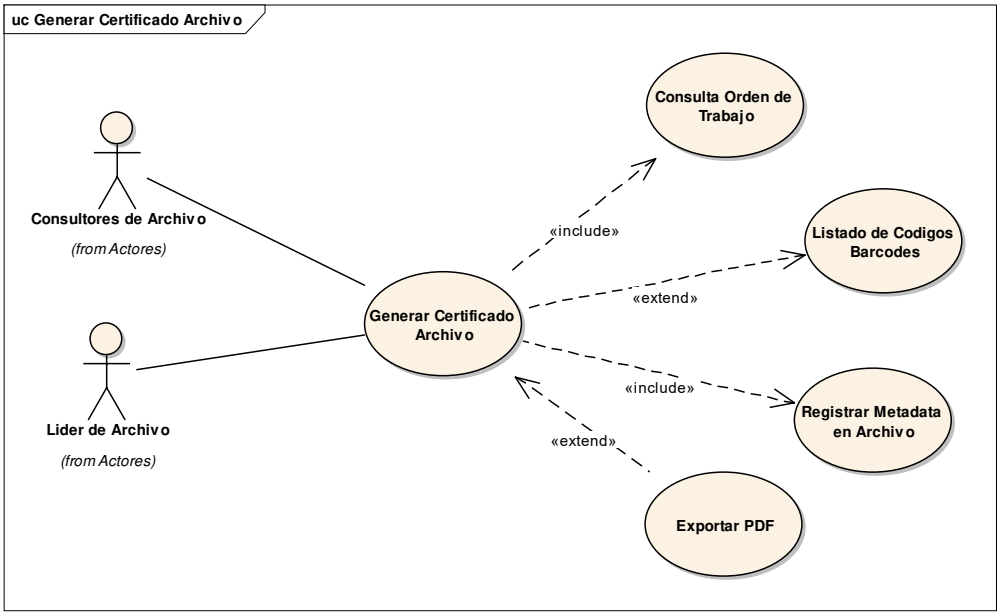



Diagrama 43: Diagrama de Caso de Uso Generar Certificado para Archivo Técnico

Fuente Elaboración Propia (2021)

### 3.2.3.3.3.3.33. D.C.U – Generar certificado sísmica.

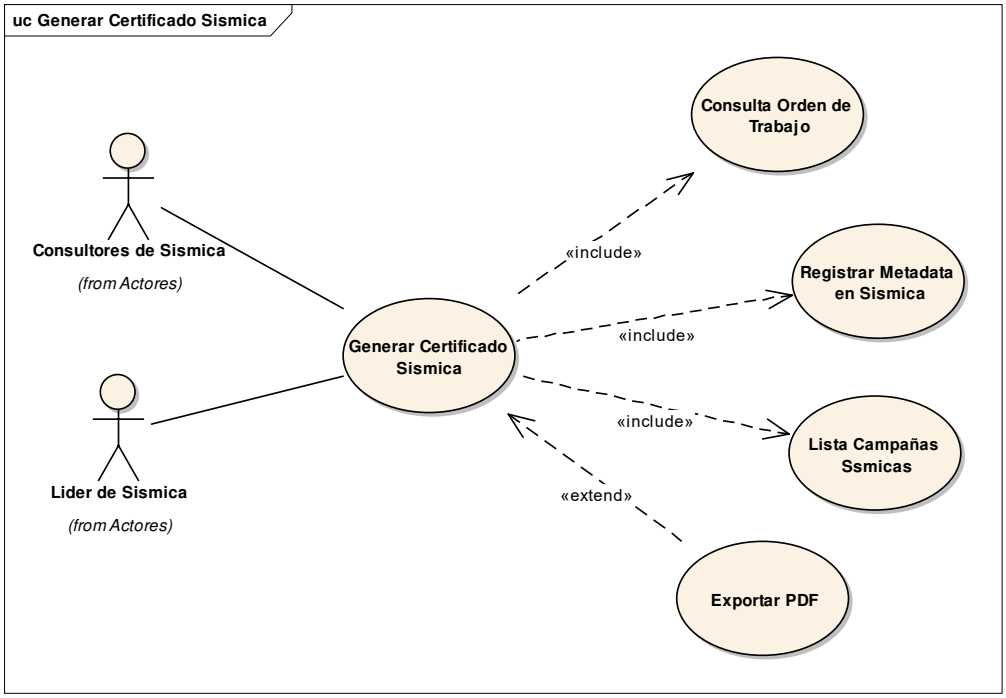



Diagrama 44: Diagrama de Caso de Uso Generar Certificado para Sísmica

Fuente Elaboración Propia (2021)

#### 3.2.3.3.3.4. Especificación de casos de uso

##### 3.2.3.3.3.4.1. C.U Registrar ordenes de trabajo

| | |
|---|---|
| **Id caso de uso** | CU001 |
| **Modulo** | Gestión de Ordenes de Trabajo |
| **Nombre** | Registrar Orden de Trabajo |
| **Descripción** | Permite crear una Orden de Trabajo asociado a un memorándum. |
| **Excepciones** | No se puede crear una Orden de Trabajo sin la aprobación de Gerencia de Servicios.<br>No se puede crear una Orden de Trabajo sin una solicitud de mediante un memorándum. |
| **Casos de uso asociados** | Notificación por Email, Crear Tarea. |
| **Flujo normal** | |
| **Actores** | Grupo Recepción |
| **Precondiciones** | El actor debe estar validado en el sistema y debe tener un perfil con privilegios para ingresar a la opción Registrar Orden de trabajo. |
| **Activación** | El actor selecciona la correspondiente opción de menú |
| **Descripción** | 1. El sistema muestra la pantalla de Registrar Orden de Trabajo, mostrando un formulario con los siguientes campos: Combo box de Origen, Combo box de Solicitado por, una caja de texto de Memo, una caja de texto de Asunto, una caja de texto de Detalle, una caja de texto de Observación, Combo box de Estado, Combo box de Tipo de OT, Combo box de Tipo de Información, Combo box de Gerencia Solicitante, Combo box de Grupos de Atención.<br>2. Se muestra la fecha de inicio y por default la fecha del día. Esta fecha no podrá ser cambiada.<br>3. EL mes generado se crea automáticamente y no necesariamente debería coincidir con el mes del año, para el banco de datos el ciclo de mes inicia el día 18. Por ejemplo: Mes 02 está en el rango del 18/08/2018 – 19/09/2018. |

| | |
|---|---|
| | 4. EL mes cerrado se genera automáticamente después de cambiar el campo estado ha cerrado y de guardar el formulario.<br>5. El actor después de ingresar los datos en el formulario, presiona el botón Registrar Orden de Trabajo.<br>6. El sistema crea la OT con los siguientes datos:<br>- Los datos del formulario del punto 1<br>- El número de la OT: "EXPL-GFBD-####-YYYY", donde #### es un correlativo único en el año YYYY. Ejemplo: EXPL-GFBD-00119-2021.<br>- Estado de la OT: "PROCESO"<br>- La fecha y hora de creación es dada por el Sistema. No visible. |
| **Postcondiciones** | 7. Notificación de nueva Orden de Trabajo<br>El sistema envía notificaciones vía email a los siguientes destinos: Supervisores de PP, Gerente de Servicio, Líderes y Consultores asociados al Combo box Grupos de Atención. |

Tabla 7: Caso de Uso Registrar Órdenes de Trabajo
Fuente Elaboración Propia (Año 2021)

#### 3.2.3.3.3.4.2. C.U Actualizar ordenes de trabajo

| | |
|---|---|
| **ID Caso de Uso** | CU002 |
| **Modulo** | Gestión de Ordenes de Trabajo |
| **Nombre** | Actualizar Orden de Trabajo |
| **Descripción** | Actualiza una Orden de Trabajo. |
| **Excepciones** | No se puede actualizar una Orden de Trabajo sin la aprobación de Gerencia de Servicios. |
| **C.U Asociados** | Notificación por Email, Crear Tarea. |
| **Flujo Normal** | |
| **Actores** | Grupo Recepción, Lideres |
| **Precondiciones** | El actor debe estar validado en el sistema y debe tener un perfil con privilegios para ingresar a la opción de actualizar orden de trabajo. |
| **Activación** | El actor lista las ordenes de trabajo y busca la Orden de Trabajo le da clic a la opción editar. |
| **Descripción** | 1. El sistema muestra la pantalla de Orden de Trabajo, mostrando un formulario con los siguientes campos habilitados: una caja de texto de Detalle, una caja de texto de Observación, Combo box de Estado, Combo box de Grupos de Atención.<br>2. El sistema muestra la pantalla de Orden de Trabajo, mostrando un formulario con los siguientes campos bloqueados: Fecha de inicio, Mes generado, Origen, Solicitado por, Memo, Asunto, Tipo de OT, Tipo de Información, Gerencia Solicitante.<br>3. El actor después de editar los datos en el formulario, presiona el botón Actualizar Orden de Trabajo. |

| | |
|---|---|
| | 4. El sistema actualiza la Orden de Trabajo con los siguientes datos:<br>- La fecha y hora de actualización es dada por el Sistema. No visible. |
| **Postcondiciones** | 5. Notificación de actualización de Orden de Trabajo<br>El sistema envía notificaciones vía email a los siguientes destinos: Supervisores de PP, Gerente de Servicio, Lideres y Consultores asociados al input select Grupos de Atención. |

Tabla 8: Caso de Uso Actualizar Órdenes de Trabajo
Fuente Elaboración Propia (Año 2021)

#### 3.2.3.3.3.4.3. C.U Registrar tareas

| | |
|---|---|
| **Id caso de uso** | CU003 |
| **Modulo** | Gestión de Ordenes de Trabajo |
| **Nombre** | Registrar Tareas |
| **Descripción** | Registrar Tareas para cada Orden de Trabajo. |
| **Excepciones** | No se pueden Registrar Tareas que no existen o estén previamente autorizadas por el Grupo de Recepción. |
| **Casos de uso asociados** | Notificación por Email, Consulta de Tarea, Consulta de Ordenes de Trabajo. |
| **Flujo normal** | |
| **Actores** | Grupo Recepción, Lideres |
| **Precondiciones** | El actor debe estar validado en el sistema y debe tener un perfil con privilegios para ingresar a la opción Registrar Tareas. |
| **Activación** | El actor selecciona la Orden de Trabajo y hace clic a la opción de menú |
| **Descripción** | 1. El sistema muestra el listado de Órdenes de Trabajo.<br>2. El actor podrá utilizar los filtros de cabecera, en caso que la Orden de Trabajo no se encuentre en la lista mostrada, al ubicar deberá darle clic en abrir Orden de Trabajo. |

| | |
|---|---|
| | 3. El sistema mostrara los datos de la Orden de Trabajo y en la parte superior en una pestaña la opción Tareas<br>4. El actor deberá darle clic a la Pestaña Tareas.<br>5. En sistema mostrara en el caso de que se hallan creado tareas, un listado de tareas y el botón de Agregar Tarea.<br>6. El actor deberá darle clic en Agregar Tarea.<br>7. EL sistema abrirá una ventana emergente con el formulario de tareas con los siguientes campos: Combo box de Usuario, Combo box de Nombres, Combo box de Grupo Funcional, Combo box de Tipo de Tarea, una caja de texto de Detalle/Comentarios.<br>8. El actor completara los datos del formulario y le dará clic al botón Guardar.<br>9. Se registra automáticamente la fecha de creación, usuario responsable, porcentaje de avance y Orden de Trabajo asociada.<br>10. El sistema crea la Tarea con los siguientes datos:<br>- Los datos del formulario mencionado en el punto 7<br>- El número de la Tarea: "####", donde #### es un correlativo. Ejemplo: 00015.<br>- Porcentaje de avance de la Tarea: "0%<br>- Usuario Responsable.<br>- Orden de Trabajo asociada.<br>- La fecha y hora de creación es dada por el Sistema. No visible. |
| **Postcondiciones** | 11. Notificación de nueva Tarea<br>El sistema envía notificaciones vía email a los siguientes destinos: Gerente de Servicio, Líderes y Consultores asociados al Combo box de Usuario. |

Tabla 9: Caso de Uso Registrar Tareas
Fuente Elaboración Propia (Año 2021)

### 3.2.3.3.3.4.4. C.U Actualizar tareas

| | |
|---|---|
| **Id caso de uso** | CU004 |
| **Modulo** | Gestión de Ordenes de Trabajo |
| **Nombre** | Actualizar Tareas |
| **Descripción** | Actualizar tareas para cada Orden de Trabajo. |
| **Excepciones** | No se pueden actualizar tareas que no existen o estén previamente autorizadas por los Lideres de cada Grupo Funcional. |
| **Casos de uso asociados** | Notificación por Email, Consulta de Tarea, Consulta de Ordenes de Trabajo. |
| **Flujo normal** | |
| **Actores** | Lideres |
| **Precondiciones** | El actor debe estar validado en el sistema y debe tener un perfil con privilegios para ingresar a la opción Registrar Tareas. |
| **Activación** | El actor selecciona la Orden de Trabajo y hace clic a la opción de menú |
| **Descripción** | 1. El sistema muestra el listado de Órdenes de Trabajo.<br>2. El actor podrá utilizar los filtros de cabecera, en caso que la Orden de Trabajo no se encuentre en la lista mostrada, al ubicar deberá darle clic en abrir Orden de Trabajo.<br>3. El sistema mostrara los datos de la Orden de Trabajo y en la parte superior en una pestaña la opción Tareas<br>4. El actor deberá darle clic a la Pestaña Tareas.<br>5. En sistema mostrara en el caso de que se hallan creado tareas, un listado de tareas y el botón de Editar Tarea.<br>6. El actor deberá darle clic en Editar Tarea.<br>7. EL sistema abrirá una ventana emergente con el formulario de tareas con los siguientes campos habilitados: Combo box de Tipo de Tarea, una caja de texto de Detalle/Comentarios y Combo box de Porcentaje de Avance. |

| | |
|---|---|
| | 8. EL sistema abrirá una ventana emergente con el formulario de tareas con los siguientes campos bloqueados: Combo box de Usuario, Combo box de Nombres, Combo box de Grupo Funcional<br>9. El actor completara los datos del formulario y le dará clic al botón Actualizar.<br>10. Se registra automáticamente la fecha de actualización, usuario responsable y Orden de Trabajo asociada.<br>11. El sistema actualiza la Tarea con los siguientes datos:<br>- Los datos del formulario mencionado en el punto 7<br>- El número de la Tarea: "####", donde ####<br>- Ejemplo: 00015.<br>- Porcentaje de avance de la Tarea: "0%<br>- Usuario Responsable.<br>- Orden de Trabajo asociada.<br>- La fecha y hora de actualización es dada por el Sistema. No visible. |
| **Postcondiciones** | 12. Notificación de nueva Tarea<br>El sistema envía notificaciones vía email a los siguientes destinos: Gerente de Servicio, Líderes y Consultores asociados al Combo box de Usuario. |

Tabla 10: Caso de Uso Actualizar Tareas
Fuente Elaboración Propia (Año 2021)

### 3.2.3.3.3.4.5. C.U Eliminar tareas

| | |
|---|---|
| **Id caso de uso** | CU005 |
| **Modulo** | Gestión de Tareas |
| **Nombre** | Eliminar Tareas |
| **Descripción** | Permite actualizar tareas para cada Orden de Trabajo. |
| **Excepciones** | No se pueden eliminar tareas que no existen o estén previamente autorizadas por los Lideres de cada Grupo Funcional. |

| | |
|---|---|
| **Casos de uso asociados** | Notificación por Email, Consulta de Tarea, Consulta de Ordenes de Trabajo. |
| **Flujo normal** | |
| **Actores** | Lideres |
| **Precondiciones** | El actor debe estar validado en el sistema y debe tener un perfil con privilegios para ingresar a la opción Registrar Tareas. |
| **Activación** | El actor selecciona la Orden de Trabajo y hace clic a la opción de menú |
| **Descripción** | 1. El sistema muestra el listado de Órdenes de Trabajo.<br>2. El actor podrá utilizar los filtros de cabecera, en caso que la Orden de Trabajo no se encuentre en la lista mostrada, al ubicar deberá darle clic en abrir Orden de Trabajo.<br>3. El sistema mostrara los datos de la Orden de Trabajo y en la parte superior en una pestaña la opción Tareas<br>4. El actor deberá darle clic a la Pestaña Tareas.<br>5. En sistema mostrara en el caso de que se hallan creado tareas, un listado de tareas y el botón de Eliminar Tarea.<br>6. El actor deberá darle clic en Eliminar Tarea.<br>7. EL sistema abrirá una ventana emergente solicitando confirmación.<br>8. El actor le da clic en aceptar.<br>9. Se registra automáticamente la fecha de eliminación, usuario responsable y Orden de Trabajo asociada. |
| **Postcondiciones** | 10. Notificación de nueva Tarea<br>El sistema envía notificaciones vía email a los siguientes destinos: Gerente de Servicio, Líderes y Consultores asociados al Combo box de Usuario. |

Tabla 11: Caso de Uso Eliminar Tareas
Fuente Elaboración Propia (Año 2021)

### 3.2.3.3.3.4.6. C.U Cargar metadata

| | |
|---|---|
| **Id caso de uso** | CU006 |
| **Modulo** | Gestión de Metadata |
| **Nombre** | Cargar Metadata |
| **Descripción** | Cargar Metadata básica para cada Orden de Trabajo. |
| **Excepciones** | Debe existen archivos digitales en el repositorio asociado a la Orden de Trabajo<br>Debe estar designado por la Orden de Trabajo y autorización del Líder. |
| **Casos de uso asociados** | Consulta de Tarea, Consulta de Ordenes de Trabajo, Consulta Metadata, Registro de Metadata, Consulta de Certificados, Importar Metadata. |
| **Flujo normal** | |
| **Actores** | Grupo Recepción, Líderes y Consultores. |
| **Precondiciones** | El actor debe estar validado en el sistema y debe tener un perfil con privilegios para ingresar a esta opción. |
| **Activación** | El actor selecciona la Orden de Trabajo y hace clic a la opción de menú |
| **Descripción** | 1. El sistema muestra el listado de Órdenes de Trabajo.<br>2. El actor podrá utilizar los filtros de cabecera, en caso de que la Orden de Trabajo no se encuentre en la lista mostrada, al ubicar deberá darle clic en abrir Orden de Trabajo.<br>3. El sistema mostrara los datos de la Orden de Trabajo y en la parte superior encontrara la pestaña Documentos Asociados.<br>4. El actor deberá darle clic a la Pestaña Documentos Asociados.<br>5. En sistema mostrará una pestaña de nombre catalogo donde se visualizará un menú de opciones y una tabla con la Metadata cargada anteriormente o caso contraria una tabla vacía.<br>6. El actor dará clic en la opción de menú: Abrir Explorador. |

| | |
|---|---|
| | 7. EL sistema abrirá una ventana emergente dividida en dos paneles, el panel izquierdo mostrará la estructura de carpetas del repositorio asociado a la OT, en el panel de la izquierda superior el botón Asociar Archivos y en el panel izquierdo inferior un visualizador de archivos seleccionados.<br>8. El actor navegara por la estructura de carpeta y seleccionara los archivos de su interés.<br>9. EL sistema mostrara un listado de solo archivos dentro de una tabla con la siguiente cabecera:<br>- Nombre de Archivo.<br>- Nro. de Orden de Trabajo.<br>- Ruta.<br>- Extensión.<br>- Peso en Bytes.<br>10. El actor dentro de la tabla podrá seleccionar o retirar los archivos.<br>11. El actor le dará clic al botón Asociar Archivos<br>12. El sistema registra en la base de datos lo seleccionado, se registra automáticamente la fecha de creación y usuario responsable. |
| **Postcondiciones** | 13. La Metadata cargada se visualizará automáticamente en las pestañas: Catalogo, Pozos, GIS Map, Técnico, Sísmica. |

Tabla 12: Caso de Uso Cargar Metadata
Fuente Elaboración Propia (Año 2021)

### 3.2.3.3.3.4.7. C.U Registrar de metadata catalogo

| | |
|---|---|
| **Id caso de uso** | CU007 |
| **Modulo** | Gestión de Metadata |
| **Nombre** | Registrar Metadata de Catalogo |
| **Descripción** | Permite registrar y actualizar Metadata de Catalogo |
| **Excepciones** | Debe estar cargado la Metadata básica en Catalogo.<br>Debe estar designado en la O.T y autorización del Líder. |
| **Casos de uso asociados** | Consulta de Tarea, Consulta de O.T, Consulta Metadata, Consulta de Certificados, Importar Metadata. |
| **Flujo normal** | |
| **Actores** | Grupo Recepción. |
| **Precondiciones** | El actor debe estar validado en el sistema y debe tener un perfil con privilegios para ingresar a esta opción. |
| **Activación** | El actor selecciona la Orden de Trabajo y hace clic a la opción de menú |
| **Descripción** | 1. El sistema muestra el listado de Órdenes de Trabajo.<br>2. El actor podrá utilizar los filtros de cabecera, para ubicar la Orden de Trabajo y darle clic en abrir.<br>3. El sistema mostrara los datos de la O.T y en la parte superior encontrara la pestaña Documentos Asociados.<br>4. El actor le da clic a la Pestaña Documentos Asociados.<br>5. En sistema mostrara una pestaña de nombre catalogo donde se visualizará un menú de opciones y una tabla con Metadata registrada. La Metadata cargada según el proceso explicado en el Caso de Uso CU006 siempre estará visible pero no editable siendo sus columnas: **Nombre de Archivo, Nro. de Orden de Trabajo, Ruta, Extensión y Peso en Bytes**.<br>La Metadata para registrar o editar pertenece a las siguientes columnas: **Tipo de Dato, Código Medio Digital, Código Medio Editado, Tipo Barcode, Código Barcode, Doc.** |

| | |
|---|---|
| | **Original, Titulo, Doc. Principal, Pozo, Uwi, Región, Cuenca, Lote, Yacimiento**.<br>6. El actor dará clic en las celdas de la tabla donde podrá registrar o actualizar la Metadata de forma individual o grupal.<br>7. EL sistema después de darle clic a una celda diferente a la seleccionada guardara automáticamente en la base de datos lo ingresado en las celdas seleccionadas.<br>8. El sistema registra en la base de datos automáticamente la fecha de creación y usuario responsable. |
| **Postcondiciones** | 9. La Metadata cargada se visualizará automáticamente en las pestañas: Pozos, GIS Map, Técnico, Sísmica. |

Tabla 13: Caso de Uso Registrar Metadata Catalogo
Fuente Elaboración Propia (Año 2021)

### 3.2.3.3.3.4.8. C.U Registrar metadata archivo

| | |
|---|---|
| **Id caso de uso** | CU008 |
| **Modulo** | Gestión de Metadata |
| **Nombre** | Registrar de Metadata Archivo |
| **Descripción** | Permite registrar y actualizar Metadata de Archivo Técnico |
| **Excepciones** | Debe estar completo la Metadata en Catalogo.<br>Debe estar designado en la O.T y autorización del Líder. |
| **Casos de uso asociados** | Consulta de Tarea, Consulta de O.T, Consulta Metadata, Consulta de Certificados, Importar Metadata. |
| **Flujo normal** | |
| **Actores** | Grupo de Archivo Técnico. |
| **Precondiciones** | El actor debe estar validado en el sistema y debe tener un perfil con privilegios para ingresar a esta opción. |
| **Activación** | El actor selecciona la Orden de Trabajo y hace clic a la opción de menú |
| **Descripción** | 1. El sistema muestra el listado de Órdenes de Trabajo. |

| | |
|---|---|
| | 2. El actor podrá utilizar los filtros de cabecera, para ubicar la Orden de Trabajo y darle clic en abrir.<br>3. El sistema mostrara los datos de la O.T y en la parte superior encontrara la pestaña Documentos Asociados.<br>4. El actor le da clic a la Pestaña Documentos Asociados y en la sub pestaña Archivo Técnico.<br>5. En sistema mostrara la tabla Archivo Técnico. En donde La Metadata de Catalogo estará visible pero no editable.<br>La Metadata para registrar o editar pertenece a las siguientes columnas: **Materia, Titulo, Informe Técnico, Paginas, Fecha Documento, Autor Institucional, Operadora, Resumen, Contenido, Estándares, Verificado, Certificado, Comentarios**.<br>6. El actor dará clic en las celdas de la tabla donde podrá registrar o actualizar la Metadata de forma individual o grupal.<br>7. EL sistema después de darle clic a una celda diferente a la seleccionada guardara automáticamente en la base de datos lo ingresado en las celdas seleccionadas.<br>8. El sistema registra en la base de datos automáticamente la fecha de creación y usuario responsable. |
| **Postcondiciones** | 9. La Metadata cargada se visualizará en Certificado de Archivo Técnico. |

Tabla 14: Caso de Uso Registrar Metadata Archivo
Fuente Elaboración Propia (Año 2021)

### 3.2.3.3.3.4.9. C.U Registrar metadata GIS

| | |
|---|---|
| **Id caso de uso** | CU009 |
| **Modulo** | Gestión de Metadata |
| **Nombre** | Registrar de Metadata GIS |
| **Descripción** | Permite registrar y actualizar Metadata de GIS |
| **Excepciones** | Debe estar completo la Metadata en Catalogo.<br>Debe estar designado en la O.T y autorización del Líder. |
| **Casos de uso asociados** | Consulta de Tarea, Consulta de O.T, Consulta Metadata, Consulta de Certificados, Importar Metadata. |
| **Flujo normal** | |
| **Actores** | Grupo de GIS. |
| **Precondiciones** | El actor debe estar validado en el sistema y debe tener un perfil con privilegios para ingresar a esta opción. |
| **Activación** | El actor selecciona la Orden de Trabajo y hace clic a la opción de menú |
| **Descripción** | 10. El sistema muestra el listado de Órdenes de Trabajo.<br>11. El actor podrá utilizar los filtros de cabecera, para ubicar la Orden de Trabajo y darle clic en abrir.<br>12. El sistema mostrara los datos de la O.T y en la parte superior encontrara la pestaña Documentos Asociados.<br>13. El actor le da clic a la Pestaña Documentos Asociados y en la sub pestaña GIS.<br>14. En sistema mostrara la tabla GIS. En donde La Metadata de Catalogo estará visible pero no editable.<br>La Metadata para registrar o editar pertenece a las siguientes columnas: **Medio Estado de Mapas, Escaneo Mapas, Medio Referencia, Tipo de Mapa, Titulo, Campaña, Compañía, Fecha Mapa, Zona, Longitud Oeste-ESI, Latitud Sur-ESI, Longitud Oeste-EID, Latitud Sur-EID, Verificado, Certificado y Comentarios**. |

| | |
|---|---|
| | 15. El actor dará clic en las celdas de la tabla donde podrá registrar o actualizar la Metadata de forma individual o grupal.<br>16. EL sistema después de darle clic a una celda diferente a la seleccionada guardara automáticamente en la base de datos lo ingresado en las celdas seleccionadas.<br>17. El sistema registra en la base de datos automáticamente la fecha de creación y usuario responsable. |
| **Postcondiciones** | 18. La Metadata cargada se visualizará en Certificado de Archivo Técnico. |

Tabla 15: Caso de Uso Registrar Metadata GIS
Fuente Elaboración Propia (Año 2021)

#### 3.2.3.3.3.4.10. C.U Generación de certificados recepción

| | |
|---|---|
| **Id caso de uso** | CU010 |
| **Modulo** | Gestión de Certificados |
| **Nombre** | Generación de Certificados de Recepción y Entrega |
| **Descripción** | El sistema generar un certificado de calidad y permite exportarlo en PDF para la posterior firma del supervisor. |
| **Excepciones** | Debe estar completo la Metadata en Catalogo. |
| **Casos de uso asociados** | Consulta de Tarea, Consulta de O.T, Consulta Metadata, Consulta de Certificados. |
| **Flujo normal** | |
| **Actores** | Grupo de Recepción y Entrega. |
| **Precondiciones** | El actor debe estar validado en el sistema y debe tener un perfil con privilegios para ingresar a esta opción. |
| **Activación** | El actor selecciona la Orden de Trabajo y hace clic a la opción de menú |
| **Descripción** | 1. El sistema muestra el listado de Órdenes de Trabajo.<br>2. El actor podrá utilizar los filtros de cabecera, para ubicar la Orden de Trabajo y darle clic en abrir. |

| | |
|---|---|
| | 3. El sistema mostrara los datos de la O.T y en la parte superior encontrara la pestaña Certificados.<br>4. El actor le da clic a la Pestaña Certificados y en la sub pestaña Recepción.<br>5. El sistema mostrara una plantilla de certificado con las siguientes características: **Logo, Orden de Trabajo, Fecha de O.T., Fecha de Recepción, Descripción, Comentarios, Total de Documentos por Tipo de Barcode, Listado de Códigos, Titulo, Total de archivos por extensión (PDF, PPT, TAR, TIF, LAS, XLSX, JPG), Peso Total en Bytes y Firma de Responsables**.<br>6. El actor después de validar que este conforme, le dará clic en Generar Secuenciador.<br>7. El sistema genera el número de certificado y lo guarda en la Base de Datos.<br>8. EL actor le da clic en descargar certificado.<br>9. El sistema descarga el certificado en formato PDF.<br>10. El sistema registra en la base de datos la fecha de creación, usuario responsable, peso certificado, peso verificado. |
| **Postcondiciones** | 11. Se actualiza el Reporte de Certificados. |

Tabla 16: Caso de Uso Generar Certificados de Recepción
Fuente Elaboración Propia (Año 2021)

### 3.2.3.3.3.4.11. C.U Generación de certificados de archivo técnico

| | |
|---|---|
| **Id caso de uso** | CU011 |
| **Modulo** | Gestión de Certificados |
| **Nombre** | Generación de Certificados de Archivo Técnico |
| **Descripción** | El sistema generar entre uno a varios certificados de calidad que deben ser exportados en PDF para la posterior firma del supervisor. |
| **Excepciones** | Debe estar completo la Metadata en Catalogo y Archivo. |
| **Casos de uso asociados** | Consulta de Tarea, Consulta de O.T, Consulta Metadata, Consulta de Certificados. |
| **Flujo normal** | |
| **Actores** | Grupo de Archivo Técnico. |
| **Precondiciones** | El actor debe estar validado en el sistema y debe tener un perfil con privilegios para ingresar a esta opción. |
| **Activación** | El actor selecciona la O.T y hace clic a la opción de menú |
| **Descripción** | 1. El sistema muestra el listado de Órdenes de Trabajo.<br>2. El actor podrá utilizar los filtros de cabecera, para ubicar la Orden de Trabajo y darle clic en abrir.<br>3. El sistema mostrara los datos de la O.T y en la parte superior encontrara la pestaña Certificados.<br>4. El actor le da clic a la Pestaña Certificados y en la sub pestaña Archivo Técnico.<br>5. El sistema mostrara un Combo box de Códigos Barcodes<br>6. El actor deberá seleccionar el código de su interés.<br>7. El sistema mostrara una plantilla de certificado con las siguientes características: **Logo, Orden de Trabajo, Tipo de O.T., Fecha, Operadora, Calidad de Documento, Código Barcode, Titulo, Uwi/Usli, Paginas, Archivo Digital, Medio Original, Archivos Originales, Pdf coincide con impreso, Peso** |

| | |
|---|---|
| | **Verificación, Peso Certificación, Comentarios, Pesos Totales y Firma de Responsables**.<br>8. El actor después de validar que este conforme, le dará clic en Generar Secuenciador.<br>9. El sistema genera el número de certificado y lo guarda en la Base de Datos.<br>10. EL actor le da clic en descargar certificado.<br>11. El sistema descarga el certificado en formato PDF.<br>12. El actor repetirá el paso 5 -11 por cada Código Barcode.<br>13. El sistema registra en la base de datos la fecha de creación, usuario responsable, peso certificado, peso verificado. |
| **Postcondiciones** | 14. Se actualiza el Reporte de Certificados. |

Tabla 17: Caso de Uso Generar Certificados de Archivo
Fuente Elaboración Propia (Año 2021)

### 3.2.3.3.3.4.12. C.U Generación de certificados de GIS

| | |
|---|---|
| **Id caso de uso** | CU012 |
| **Modulo** | Gestión de Certificados |
| **Nombre** | Generación de Certificados de Gis |
| **Descripción** | El sistema generar entre uno a varios certificados de calidad que deben ser exportados en PDF para la posterior firma del supervisor. |
| **Excepciones** | Debe estar completo la Metadata en Catalogo y GIS. |
| **Casos de uso asociados** | Consulta de Tarea, Consulta de O.T, Consulta Metadata, Consulta de Certificados. |
| **Flujo normal** | |
| **Actores** | Grupo de GIS. |
| **Precondiciones** | El actor debe estar validado en el sistema y debe tener un perfil con privilegios para ingresar a esta opción. |
| **Activación** | El actor selecciona la O.T y hace clic a la opción de menú |

| | |
|---|---|
| **Descripción** | 1. El sistema muestra el listado de Órdenes de Trabajo.<br>2. El actor podrá utilizar los filtros de cabecera, para ubicar la Orden de Trabajo y darle clic en abrir.<br>3. El sistema mostrara los datos de la O.T y en la parte superior encontrara la pestaña Certificados.<br>4. El actor le da clic a la Pestaña Certificados y en la sub pestaña GIS.<br>5. El sistema mostrara un Combo box de Tipo de Mapas<br>6. El actor deberá seleccionar el mapa de su interés.<br>7. El sistema mostrara una plantilla de certificado con las siguientes características: **Logo, Orden de Trabajo, Fecha de Recepción, Medio Original, Medio de Referencia, Archivos Shapefile, Estado de Mapa, Tipo de Mapa, Código Barcode, Titulo, Operadora, Región, Cuenca, Lote, Datum, Proyección, Zona, Coordenadas, Peso Verificación, Peso Certificación, Anotaciones, Pesos Totales y Firma de Responsable**.<br>8. El actor después de validar que este conforme, le dará clic en Generar Secuenciador.<br>9. El sistema genera el número de certificado y lo guarda en la Base de Datos.<br>10. EL actor le da clic en descargar certificado.<br>11. El sistema descarga el certificado en formato PDF.<br>12. El actor repetirá el paso 5 -11 por cada Tipo de Mapa.<br>13. El sistema registra en la base de datos la fecha de creación, usuario responsable, peso certificado, peso verificado. |
| **Postcondiciones** | 14. Se actualiza el Reporte de Certificados. |

Tabla 18: Caso de Uso Generar Certificados de GIS
Fuente Elaboración Propia (Año 2021)

### 3.2.3.3.3.4.13. C.U Reporte de órdenes de trabajo en proceso

| | |
|---|---|
| **Id caso de uso** | CU013 |
| **Modulo** | Reportes y Dashboard |
| **Nombre** | Reporte de Ordenes de Trabajo en Proceso. |
| **Descripción** | El sistema permite visualizar solo las Órdenes de Trabajo en Proceso y el estado de avance por cada Grupo Funcional. |
| **Excepciones** | |
| **Casos de uso asociados** | Consulta de Tarea, Consulta de O.T. |
| **Flujo normal** | |
| **Actores** | Gerencia Servicios, Supervisores, Líderes y Consultores. |
| **Precondiciones** | El actor debe estar validado en el sistema y debe tener un perfil con privilegios para ingresar a esta opción. |
| **Activación** | El actor inicia sesión con éxito. |
| **Descripción** | 1. El sistema muestra la página de inicio y un menú vertical en el lado derecho.<br>2. El actor ubica en el menú la opción de Listar Ordenes de Trabajo y le da clic.<br>**3.** El sistema antes de mostrar el Listado de Ordenes, carga en una ventana emergente el Reporte de Ordenes de Trabajo en proceso en donde se muestra una tabla con las siguientes columnas: **Mes, Orden de Trabajo, Estado O.T, Recepción, Pozos, Gis, Archivo, Atenciones y Sísmica.**<br>4. EL sistema mostrara los estados de las columnas Recepción, Gis, Archivo, Atenciones y Sísmica en colores de Semáforo donde: Verde = Cerrada, Rojo = Denegada, Amarillo = Proceso y Negro = Anulada. Todo esto para darle una mejor visibilidad.<br>5. El actor podrá exportar el reporte en los siguientes formatos: Excel, PDF y JPG. |

| | |
|---|---|
| | 6. El sistema muestra en la parte inferior el botón cerrar reporte.<br>7. EL actor le da clic al botón cerrar.<br>8. El sistema cierra la ventana emergente de Reportes de Ordenes de Trabajo en Proceso. |
| **Postcondiciones** | |

Tabla 19: Caso de Uso Reporte de Ordenes de Trabajo en Proceso
Fuente Elaboración Propia (Año 2021)

### 3.2.3.3.3.4.14. C.U Reporte de tareas pendientes

| | |
|---|---|
| **Id caso de uso** | CU014 |
| **Modulo** | Reportes y Dashboard |
| **Nombre** | Reporte de Tareas pendientes. |
| **Descripción** | El sistema permite visualizar solo las Tareas Pendientes y el estado de avance por cada Grupo Funcional. |
| **Excepciones** | |
| **Casos de uso asociados** | Consulta de Tarea, Consulta de O.T. |
| **Flujo normal** | |
| **Actores** | Gerencia Servicios, Supervisores, Líderes y Consultores. |
| **Precondiciones** | El actor debe estar validado en el sistema y debe tener un perfil con privilegios para ingresar a esta opción. |
| **Activación** | El actor inicia sesión con éxito. |
| **Descripción** | 1. El sistema muestra la página de inicio y un menú vertical en el lado derecho.<br>2. El actor ubica en el menú la opción de Listar Tareas y le da clic.<br>**3.** El sistema antes de mostrar el Listado de Tareas, carga en una ventana emergente el Reporte de Tareas Pendientes en donde se muestra una tabla con las siguientes columnas: **Mes, Orden de Trabajo, Nombre y Apellido, Estado O.T y Cantidad de Tareas.** |

| | |
|---|---|
| | 4. EL sistema también mostrara un listado de usuarios top de tareas pendientes ordenados ascendentemente.<br>5. El sistema permite la opción de usar filtros de búsqueda por Nombre, Apellido y Grupo Funcional<br>6. El actor podrá exportar el reporte en los siguientes formatos: Excel, PDF y JPG.<br>7. El sistema muestra en la parte inferior el botón cerrar reporte.<br>8. EL actor le da clic al botón cerrar.<br>9. El sistema cierra la ventana emergente de Reportes de Tareas Pendientes. |
| **Postcondiciones** | |

Tabla 20: Caso de Uso Reporte Tareas Pendientes
Fuente Elaboración Propia (Año 2021)

### 3.2.3.3.3.4.15. C.U Dashboard gerencial

| | |
|---|---|
| **Id caso de uso** | CU015 |
| **Modulo** | Reportes y Dashboard |
| **Nombre** | Dashboard Gerencial. |
| **Descripción** | El objetivo es visualizar indicadores, tendencia mensual, semáforo de estados de los memorándums. |
| **Excepciones** | |
| **Casos de uso asociados** | Consulta de O.T. |
| **Flujo normal** | |
| **Actores** | Gerencia Servicios, Supervisores, Líderes y Consultores. |
| **Precondiciones** | El actor debe estar validado en el sistema y debe tener un perfil con privilegios para ingresar a esta opción. |
| **Activación** | El actor inicia sesión con éxito. |

| | |
|---|---|
| **Descripción** | 1. El sistema carga el Dashboard Gerencial, en donde se muestra en la parte superior los siguientes indicadores: Nro. Total Memorándums, Nro. Total de Memorándums Cerrados, Nro. Total de Memorándums en Proceso, Nro. Total de Memorándums Denegados.<br><br>En la parte inferior se muestra una tabla con las siguientes columnas: Memo, Ordenes de Trabajo, Estado, Fecha Creación, Días Restantes para Vencimiento, Asunto, Archivo y Link de Documentos.<br>La columna Estado se presentará los estados en colores de Semáforo para una mejor visibilidad en donde: Verde = Cerrada, Rojo = Denegada, Amarillo = Proceso y Negro = Anulada.<br>La Columna Días Restantes Mostrara el texto en color rojo para los memos que ya vencieron los plazos de trabajo.<br><br>En la parte derecha se mostrarán los filtros por: Memorándum, Estado, Asunto y Detalle<br>2. El actor le da clic a una de las celdas de la columna Link de Documentos.<br>3. El sistema abrirá en una nueva pestaña el documento escaneado.<br>4. El actor podrá exportar el reporte en los siguientes formatos: Excel, PDF y JPG. |
| **Postcondiciones** | |

Tabla 21: Caso de Uso Visualizar Dashboard Gerencial
Fuente Elaboración Propia (Año 2021)

### 3.2.3.3.3.5. Arquitectura propuesta

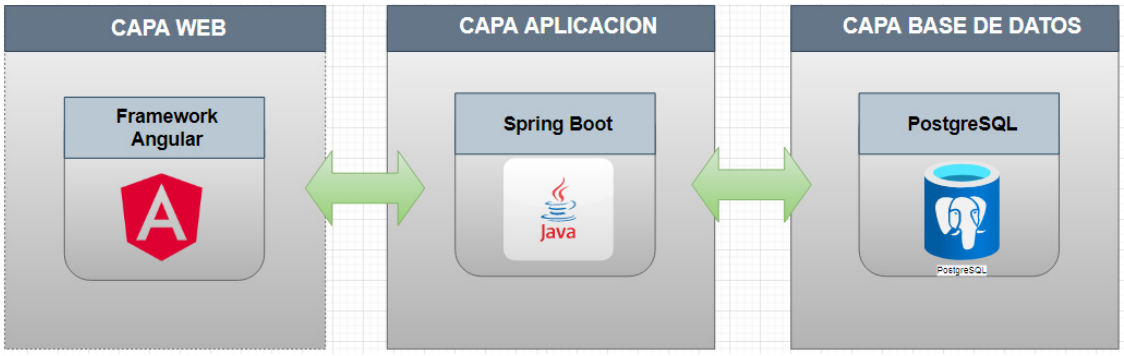



Diagrama 45: Arquitectura General de la Aplicación

Fuente: Elaboración Propia (2021)

**Capa web**

Es la capa que interactúa con el usuario aquí encontraremos las siguientes características:

- ✓ Se utilizó el Patrón de diseño MVVM (Modelo Vista Vista Modelo).
- ✓ Se utilizó el Framework de Angular, mediante el cual se construyó los templates (Interfaces).
- ✓ Se Instaló las Librerías de Boostrap y Kendo Telerik
- ✓ Seguridad por Keycloak
- ✓ Se utilizó Dashboards de Spotfire.

**Capa aplicación**

Aquí encontraremos las siguientes características:

- ✓ Se utilizó el Framework de Spring Boot
- ✓ Patrón JPA
- ✓ Compilador Maven
- ✓ Servidor Tomcat
- ✓ Arquitectura REST

**Capa base de datos**

- ✓ PostgreSQL

#### 3.2.3.3.3.6. Arquitectura frontend

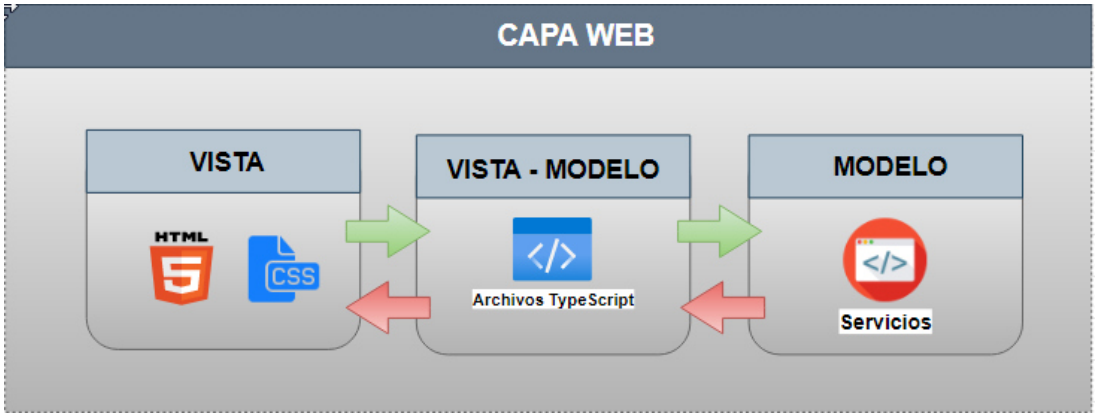



Diagrama 46: Arquitectura Frontend de la Aplicación

Fuente: Elaboración Propia (2021)

#### 3.2.3.3.3.7. Arquitectura Backend

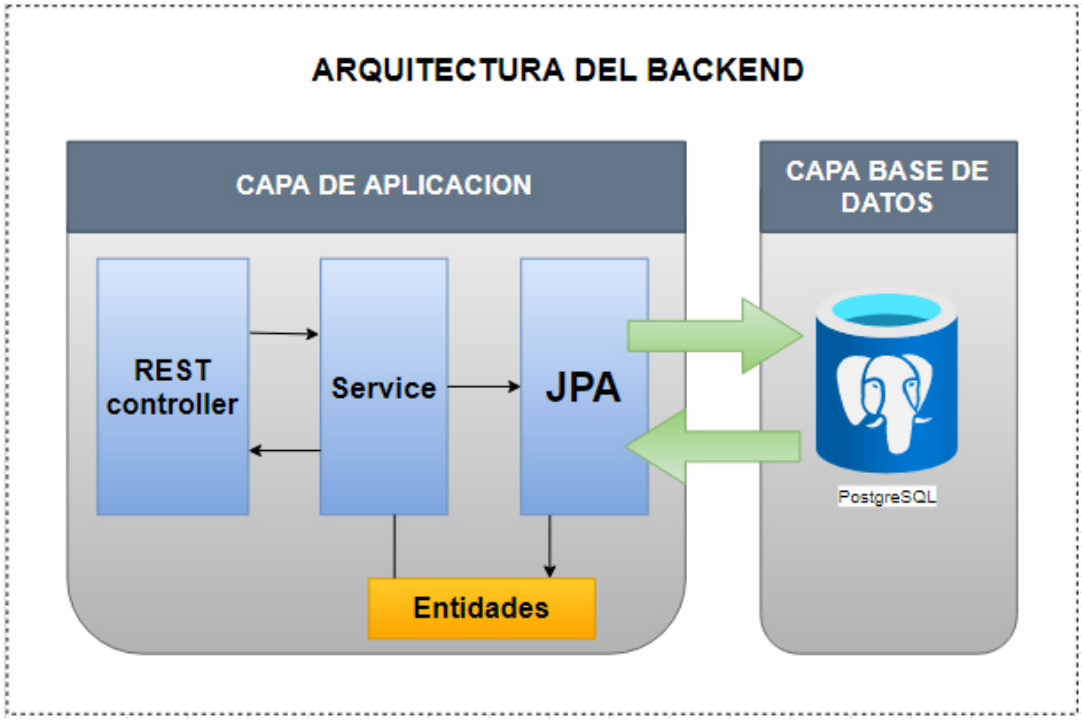
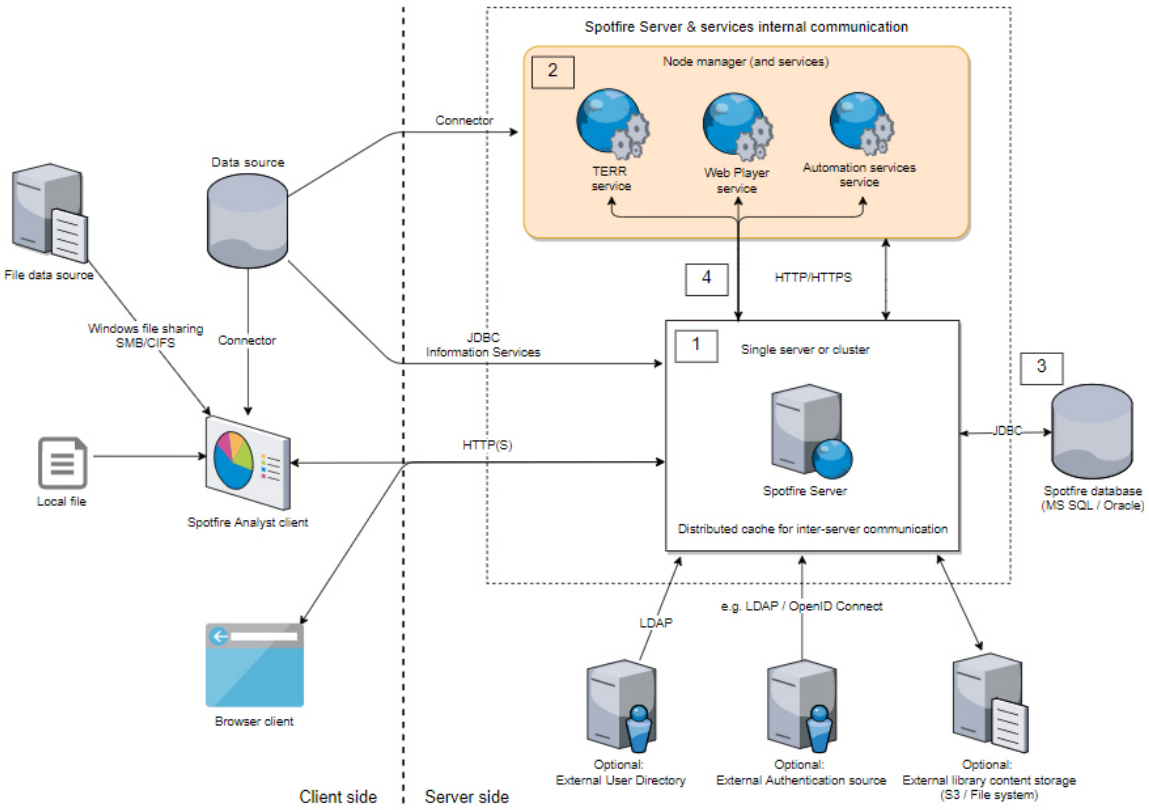



Diagrama 47: Arquitectura Backend de la Aplicación

Fuente: Elaboración Propia (2021)

#### 3.2.3.3.3.8. Arquitectura backend




Copyright © TIBCO Software Inc. All rights reserved.

Diagrama 48: Arquitectura Dashboartds

Fuente: Elaboración Tibco Spotfire (2021)

### 3.2.3.3.3.9. Modelo de datos

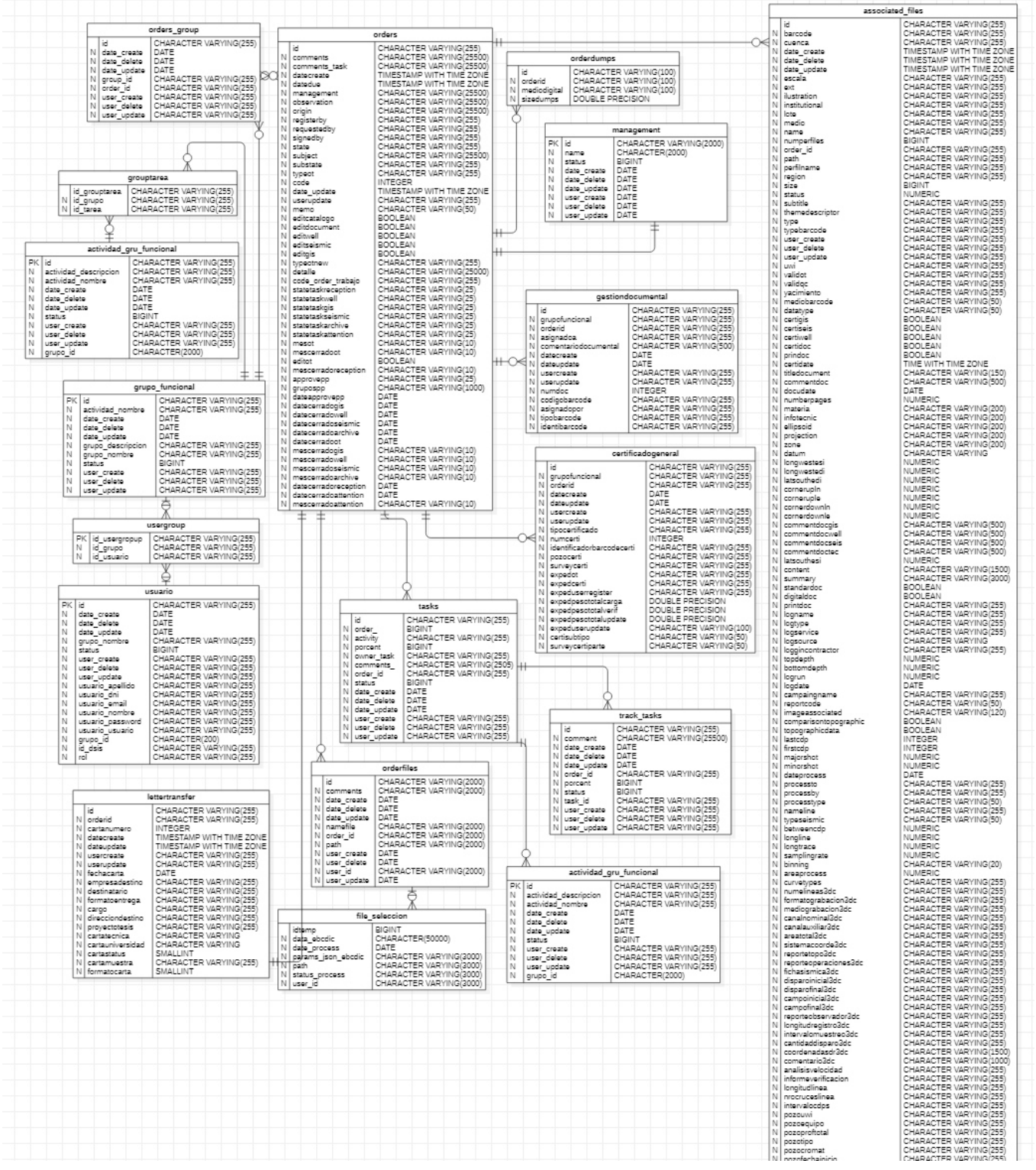



Diagrama 49: Modelo Entidad Relación de la Aplicación

Fuente: Elaboración Propia (2021)

#### 3.2.3.3.4. Etapa de construcción

Esta etapa se explica en la Sección 3.2.5. Y está desarrollado de la siguiente forma:

- ✓ Configuración de Servidor (Máquina Virtual).
- ✓ Configuración de Base de Datos (PostgreSQL).
- ✓ Configuración de Framework Angular.
- ✓ Configuración de Apache Tomcat.
- ✓ Configuración de Keycloak.
- ✓ Resultados del Desarrollo de Backend.
- ✓ Resultado del Desarrollo de Dashboards.
- ✓ Resultado del Desarrollo de Frontend.
- ✓ Pruebas.
- ✓ Despliegue.

### 3.2.4. Fundamentos utilizados

#### 3.2.4.1. Sistemas web

Considerados como aplicación cliente/servidor, donde el cliente (navegador, explorador o visualizador), el servidor (el servidor Web) y el protocolo de comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones. (L. Mora, 2002 citado en Molina Ríos, Zea Ordóñez, Contento Segarra, & García Zerda, 2018, p. 4).

Las aplicaciones Web están en constante crecimiento y su uso ha acaparado diversos ámbitos, esto es debido a las múltiples ventajas respecto a los programas de escritorio, por ejemplo: sistema operativo multiplataforma, acceso y ejecución por cualquier dispositivo informático que tenga conexión a internet, no requiere de la instalación de programas solo un navegador, las copias de seguridad son almacenadas en los servidores, la información que se genera puede ser compartida de forma simultánea por varias personas, el espacio ocupado por los datos está a cargo del servidor y es de fácil uso. (A. Ramos, 2014 citado en Molina Ríos, Zea Ordóñez, Contento Segarra, & García Zerda, 2018, p. 4).

#### 3.2.4.2. Postman

Postman es una herramienta para realización de pruebas de servicios web. Ofrece una interfaz de usuario para realizar peticiones hacia una API, evita la molestia de escribir código exhaustivo al diseccionar la funcionalidad de una API. Se utiliza Postman en esta investigación para realizar peticiones HTTP a la API REST creada y de esta manera validar el aseguramiento del servicio web y el acceso no autorizado al mismo. (Muyón & Montaluisa , 2020, pág. 203)

#### 3.2.4.3. Keycloak

Keycloak es un proyecto de código abierto desarrollado y mantenido por la Comunidad RedHat. Es una solución de gestión de acceso e identidad de código abierto dirigida a aplicaciones y servicios modernos. Facilita la protección de aplicaciones y servicios con poco o ningún código. (El Hajj Hussein, 2019 citado en Muyón & Montaluisa, 2020, p. 202).

Según (Muyón & Montaluisa, 2020, p. 202) ofrece un amplio conjunto de características, como inicio de sesión único, autenticación, autorización, inicio de sesión social, autenticación multifactor y administración centralizada de usuarios. Se listan las siguientes funcionalidades:

- ✓ Single Sign On
- ✓ Puente Kerberos
- ✓ Cortaje de identidad e inicio de sesión social
- ✓ Federación de usuarios
- ✓ Adaptadores cliente
- ✓ Consola de administración

#### 3.2.4.4. Metodologías hibridas

Existen un sin número de metodologías, ya sean agiles o tradicionales, ante ello surgen las metodologías híbridas, que viene hacer la combinación de las dos anteriores, tomando de cada uno las mejores prácticas, el objetivo es crear un método firme y flexible que se adapte a todo tipo de proyectos para el desarrollo de software. (Molina Ríos, Zea Ordóñez, Contento Segarra, & García Zerda, 2018, pág. 5).

#### 3.2.4.5. Metodología KANBAN

En el periodo de 1940 la Empresa Toyota crea el termino Just in Time (JIT) por la necesidad de sostener las mejoras continuas, impulsando el máximo rendimiento en sus procesos de fabricación. Así es como se ha definido Kanban, diseñado para el control de inventario y reducción de desperdicios. Se utilizaban tarjetas conectadas a diversos contenedores, comunicando que materiales deben reponer y que materiales se deben recargar. Hoy existe variaciones de las tarjetas Kanban incluyendo tableros y sistemas electrónicos. (ORDYSINSKI, 2013 citado en Yépez Llerena & Armijos Guillen, 2020, p. 20).

Kanban consiste en un sistema de señales visuales de control de producción que mantiene activo el proceso de reabastecimiento. Para mandar la señal de reabastecimiento existen una amplia variedad de métodos, desde tarjetas o tableros, señales visuales o electrónica. (Castellano Lendínez, 2019, pág. 37).

El sistema Kanban está basado en una serie de principios según (Castellano Lendínez, 2019, pág. 36), los cuales son:

- ✓ Representación: Permite tener una visualización global del desarrollo de las tareas en la cadena de producción, esto ayuda a la organización a balancear tareas.
- ✓ Calidad: Asegurar que los procesos se cumplen desde el principio.
- ✓ Disminución de los desperdicios: Hacer lo justo y necesario.
- ✓ Priorización – flexibilidad: Permite gestionar las prioridades de cada tarea para facilitar el trabajo de todo el equipo.
- ✓ En proceso: Kanban fomenta la continua corrección de las actividades que se están realizando.

### Kanban aplicado al Desarrollo de Software

El pionero en implementar Kanban para el desarrollo de software fue David Anderson en el año 2004 en la guía de Don Reinertsen utilizo Kanban en un proyecto de TI de Microsoft. El objetivo primordial es definir tareas por hacer y modificar sus prioridades en función de los sucesos, otra ventaja es que la cadena de trabajo está visible para todos, y en el caso de existir cuellos de botella es mucho más fácil identificar las causas, estos principios fueron apreciados por Toyota y hoy en día es muy útil en el desarrollo de software. (BBVAOPEN4U, 2018 citado en Yépez Llerena & Armijos Guillen, 2020, p. 24).

### Kanban combinado con otras metodologías

Kanban puede ser combinado con otras metodologías como herramienta de apoyo, así como diversas metodologías y filosofías de desarrollo son, casi siempre, compatibles entre sí, esto permite crear nuevos métodos que cumpla con las necesidades del equipo y la empresa. Por ejemplo, una metodología híbrida publicada es Scrumban donde se combina prácticas de Scrum y Kanban. En esta metodología se propone a Kanban como una base sobre la cual uno puede, o no, colocar prácticas de Scrum según la necesidad. (Sanhueza, 2017 citado en Yépez Llerena & Armijos Guillen, 2020, p. 26).

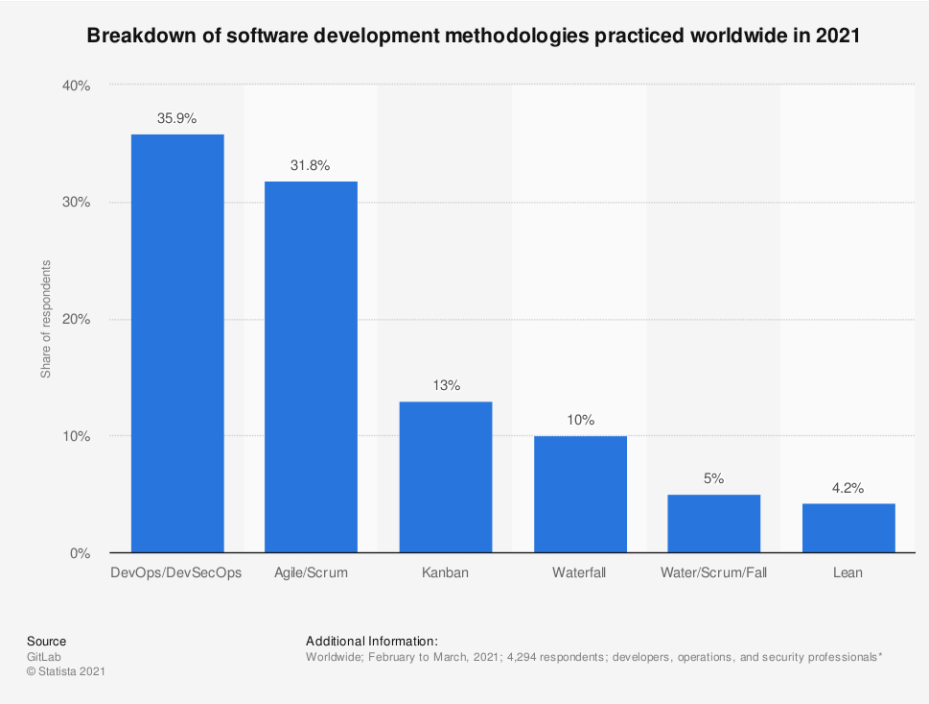



Figura 11: Estadística Metodología de desarrollo de software practicadas el 2021
Fuente (GitLab, 2021)

#### 3.2.4.6. Typescript

Es un lenguaje de programación de código abierto creado por Microsoft, trabaja como un compilador encargado de traducir instrucciones de un lenguaje a otro, ofrece soporte para las diferentes funciones presentadas por JavaScript para el desarrollo de componentes sólidos y entrega de desarrollo de aplicaciones de óptimas condiciones y de alta confianza. (Valverde Ramos & Mora de Fuentes, 2014 citado en Tito Chulde, 2020, p. 16).

TypeScript una vez compilado de forma limpia y simple JavaScript puede ser ejecutado desde cualquier navegador web. (Microsoft, 2020 citado en Tito Chulde, 2020, p. 16).

#### 3.2.4.7. Servicios web REST

Son un conjunto de restricciones arquitectónicas sincronizadas, con el objetivo de reducir latencia y comunicación de red, mientras robustece la independencia y escalabilidad en implementaciones de componentes. (Ekblom, 2011 citado en Rea-Peñafiel, Mancero-Menoscal, Rosero-Rea, & Imbaquingo-Esparza, 2020, p. 42).

Se diferencian 3 tipos de elementos en su arquitectura como son los elementos de datos, de conexión (conectores) y de procesamiento (componentes), cada uno de ellos son parte esencial de su funcionamiento. (Jakl, 2006 citado en Rea-Peñafiel, Mancero-Menoscal, Rosero-Rea, & Imbaquingo-Esparza, 2020, p. 43).

Según (Cardador, 2014 citado en Tito Chulde, 2020, p. 17), la arquitectura REST presenta cuatro tipos de operaciones para compartir y manipular recursos como:

- ✓ Read (Leer): El cliente envía una petición al servidor para leer una determinada información de un recurso.
- ✓ Create (Crear): El cliente envía una petición al servidor para la creación de un nuevo recurso en el sistema.
- ✓ Delete (Eliminar): El cliente envía una petición al servidor, esta acción permite eliminar un recurso del sistema o del servidor.
- ✓ Update (Actualizar): El cliente puede realizar una petición al servidor para modificar un recurso en el sistema.

Primordialmente los estudios efectuados en la práctica de ingeniería de software manifiestan que los servicios REST son altamente mantenibles del lado del servidor y los REST HTTP son más fáciles de componer. (Athanasopoulos & Kontogiannis, 2015 citado en Rea-Peñafiel, Mancero-Menoscal, Rosero-Rea, & Imbaquingo-Esparza, 2020, p. 43).

### 3.2.4.8. Framework Angular

Se define que Angular es un potente framework de JavaScript creado para el desarrollo de aplicaciones web dinámicas. Nos permite extender la sintaxis de HTML por medio de atributos propios del framework, para expresar componentes de nuestra aplicación de manera dinámica. (Arizmendi Paimi, 2018 citado en Arrieta Ticliahuanca & Castillo Maticorena, 2019, p. 27) .

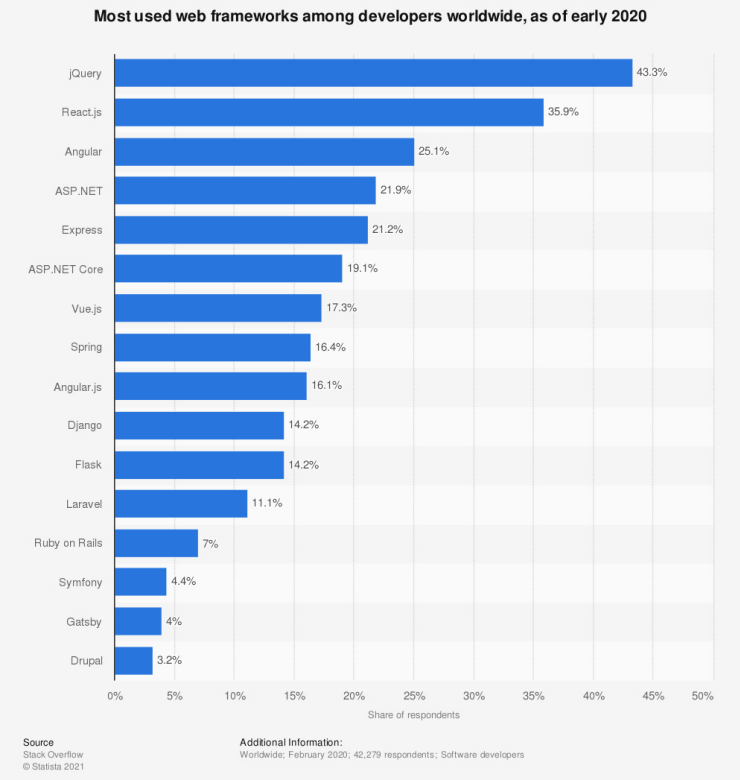



Figura 12: Estadística de Framework más utilizados en el 2020
(Stack Overflow, 2020)

### 3.2.4.9. PostgreSQL

Es un sistema de gestión de base de datos de propósito general y relacionado a objetos. Se dice que es el sistema de base de datos más avanzado de código abierto. (PostgreSQL Tutorial, 2020 citado en Vargas Rioja, 2020, p. 14).

Se usará este DBMS pues además de ser gratuito, según la experiencia de los tesistas, es bastante robusto y tiene mayor nivel de performance comparado con otras bases de datos gratuitas; lo que será muy útil para los centros de salud con menos recursos. (Vargas Rioja, 2020, pág. 14)

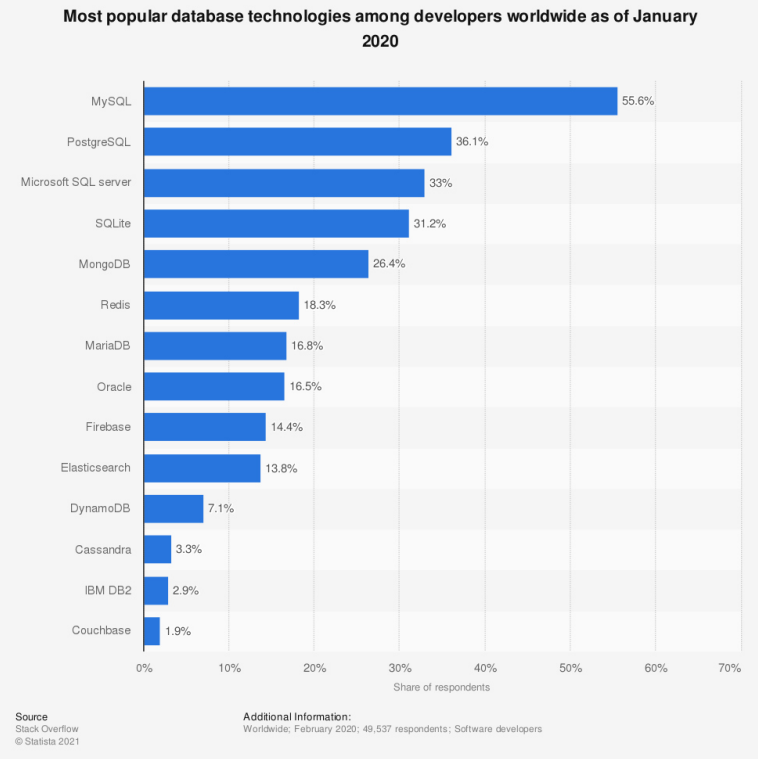



Figura 13: Estadística de Base de Datos más utilizados en el 2020
Fuente (DB-Engines, 2021)

#### 3.2.4.10. Kendo Telerik

Kendo es una estructura de software para diseño de páginas Web basado en JQuery HTML5, usado para construir aplicaciones Web modernas, cuenta con gran cantidad de widgets (aplicaciones pequeñas) de interfaz de usuario para la visualización de datos. (telerik.com, 2001 citado en Morales Timbela & Moya Portilla, 2015, p. 17).

Kendo integra su funcionalidad con java Script y Ajax (técnica para cargar dados o partes de HTML sin refrescar la ventana del navegador, permitiendo comunicación con el servidor utilizando el objeto XMLHttpRequest) para ejecutar las pequeñas aplicaciones propuestas y mostrar datos al usuario. Su utilidad y versatilidad, hace que sea compatible con la mayoría de navegadores web. Siendo un paquete de librerías desarrolladas íntegramente en java Script y basadas en jQuery (simplifica la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones, etc.). La variedad de componentes para interfaces de usuario de Kendo UI incluye menús dinámicos, gráficas, paneles, rejillas de datos, árboles, ventanas, sistemas de upload de archivos, etc. (Morales Timbela & Moya Portilla, 2015, págs. 17-18).

#### 3.2.4.11. Spring Boot

Spring es un marco de trabajo para aplicación web, que se ha popularizado entre los programadores, con el tiempo los proyectos basado en él fueron generando inconvenientes por un gran número de archivos XML que tenían que estar correctamente configurados para que los componentes individuales y la aplicación funcionarían correctamente. (Webb et al., 2018 citado en Tito Chulde, 2020, p. 12)

Esto ha causado muchas frustraciones entre los programadores al invertir tiempo que puede ser valiosa en la lógica de negocios. Por lo tanto, se creó Spring Boot un módulo desarrollado a partir de Spring, que nace con la finalidad de simplificar aún más la programación de aplicaciones basadas en el framework, ayudando al programador solo centrarse en el desarrollo de la solución del problema, olvidándose por completo la compleja configuración inicial que actualmente tiene Spring. (Gajewski & Zabierowski, 2019 citado en Tito Chulde, 2020, p. 12).

Spring Boot simplifica enormemente tareas convencionales como son el despliegue del servidor, el cual ya se encuentra embebido dentro del proyecto que por defecto es un Tomcat permitiendo al proyecto se ejecute en un Jar. Las configurará se realizan automáticamente como el gestor de base de datos, el caso que trabaje con varias bases de datos diferente lo único que debe hacer el programador es proporcionar una dirección IP; también permite usar las herramientas de gestor de proyectos como Maven y Gradle mediante la administración de dependencias. (Suryotrisongko, Jayanto, & Tjahyanto, 2017 citado en Tito Chulde, 2020, p. 12).

#### 3.2.4.12. Tibco Spotfire

TIBCO Spotfire® es la solución de análisis más completa del mercado, y hace que sea rápido y fácil para todos visualizar los nuevos descubrimientos en los datos. Utilizando la búsqueda y las recomendaciones impulsadas por un motor de inteligencia artificial incorporado, cualquiera puede explorar los datos para aportar información. Ya sea por la creación de métricas simples en dashboards, aplicaciones predictivas o aplicaciones dinámicas de análisis en tiempo real, Spotfire® ofrece numerosas capacidades a escala, entre las que se incluyen la analítica visual, el manejo de datos, la analítica predictiva, geoanalítica y analítica en tiempo real. (Tibco Spotfire, 2021).

#### 3.2.4.13. Apache TOMCAT

También llamado Jakarta Tomcat o Apache Tomcat, funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Implementa las especificaciones de los servlets y de Java Server Pages (JSP) de Sun Microsystems. Dado que fue escrito en lenguaje Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java. (AST & JHA, 2008 citado en Gutiérrez Zúñiga & Chanca de la Cruz, 2011, p. 56).

### 3.2.5. Implementación de la aplicación

#### 3.2.5.1. Configuración de servidor

La aplicación web se instalará en un servidor virtual, donde desde el VMware podremos escalar con facilidad el hardware necesario para un óptimo rendimiento, siendo las características principales del servidor:

- ✓ Windows 2012 –R2
- ✓ Memoria Ram 56 GB
- ✓ Procesador Intel Xeon @ 3.00Ghz (8 Procesos)
- ✓ 1.1 TB de Disco Duro

Así mismo debemos configurar:

- ✓ Protocolo de Internet red debe estar en la versión IPv4
- ✓ Se debe agregar el puerto de enlace, los DNS y la IP en el segmento establecido.
- ✓ El servidor virtual debe encontrar en el dominio del Active Directory.

#### 3.2.5.2. Configuración de base de datos

Primero procedemos a instalar y configurar PostgreSQL versión 9.012:

- ✓ Descarga

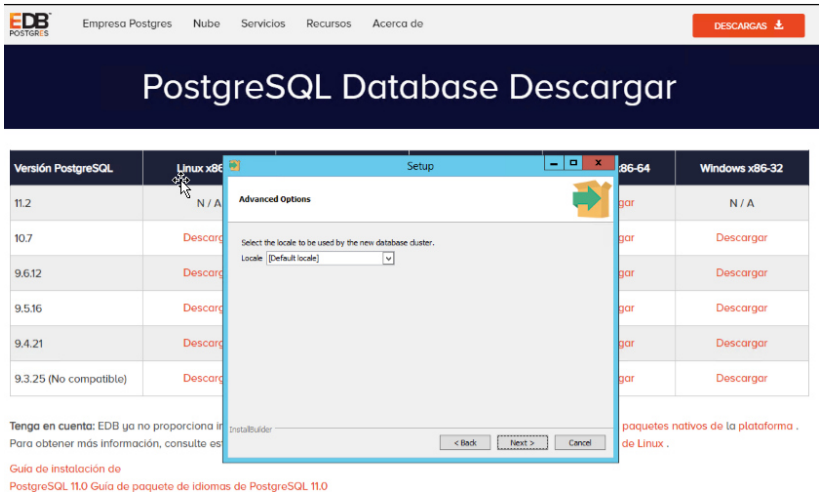



Figura 14: PostgreSQL selección versión a instalar
Fuente Elaboración Propia (Año 2021)

✓ Selección de Paquetes a instalar

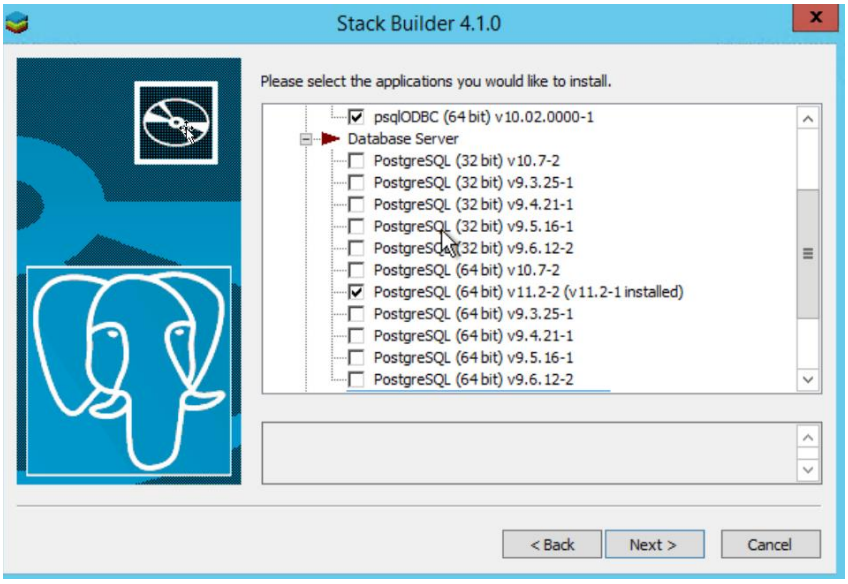



Figura 15: PostgreSQL selección de paquetes a instalar
Fuente Elaboración Propia (Año 2021)

✓ Ubicamos el Archivo pg_hba.conf

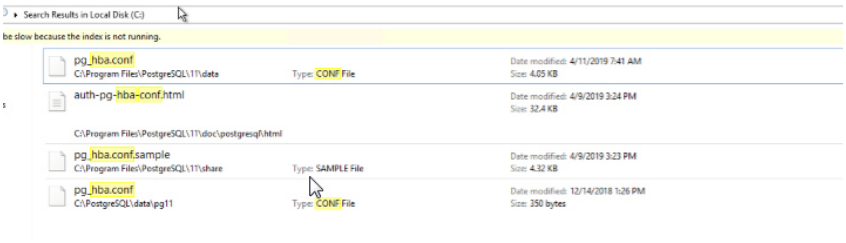



Figura 16: PostgeSQL ubicación de archivo pg_hba.conf
Fuente Elaboración Propia (Año 2021)

✓ Cambiamos la configuración pasando de local a global (Dentro de la Red de Dominio)

```
75
76
77  # IPv4 local & remote connections:
78  host    all             all             127.0.0.1/32            trust
79  host    all             all             0.0.0.0/0               md5
80  # IPv6 local connections:
81  host    all             all             ::1/128                 trust
82
83
84
85
```

Figura 17: PostgreSQL configuración de acceso dentro del dominio
Fuente Elaboración Propia (Año 2021)

✓ Agregamos una nueva regla en el Firewall, para evitar bloqueos.

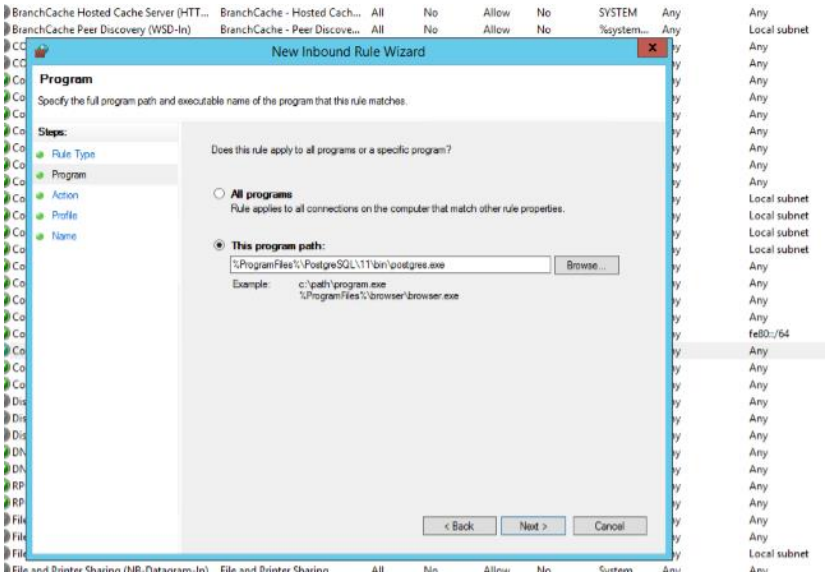



Figura 18: PostgreSQL modificar reglas de firewall
Fuente Elaboración Propia (Año 2021)

✓ Instalamos el PgAdmin versión 4.29 para la Administración de la Base de Datos

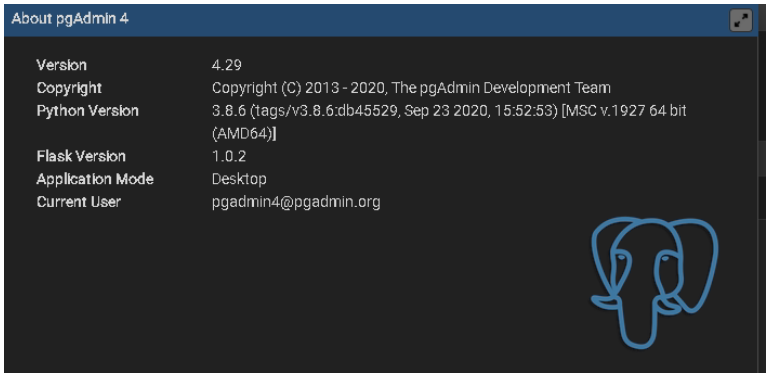



Figura 19: PostgreSQl instalar herramienta de administración
Fuente Elaboración Propia (Año 2021)

### 3.2.5.3. Configuración de Framework de Angular

- ✓ Procedemos a descargar e instalar NodeJS versión 14.17 para Windows

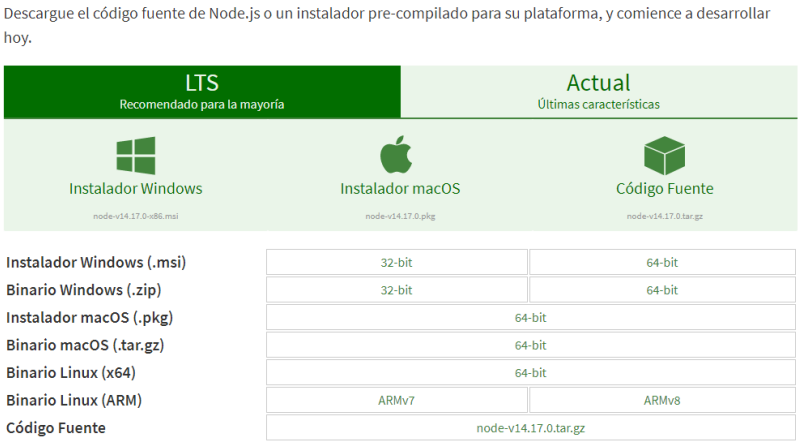



Figura 20: Angular instalar NodeJS
Fuente Elaboración Propia (Año 2021)

- ✓ Por comandos desde cmd procedemos a instalar angular cli, la versión debe ser la 6.1.7901

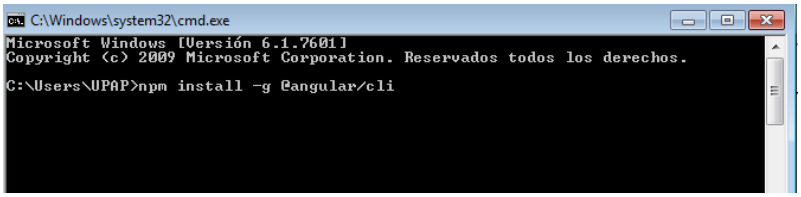



Figura 21: Angular instalación mediante angular/cli
Fuente Elaboración Propia (Año 2021)

- ✓ Corroboramos la instalación




Figura 22: Angular corroborar instalación y versión
Fuente Elaboración Propia (Año 2021)

### 3.2.5.4. Configuración de Apache Tomcat

- ✓ Instalamos la versión 8.5.35 en el servicio desde consola, en la instalación debemos tener marcado la casilla host manager

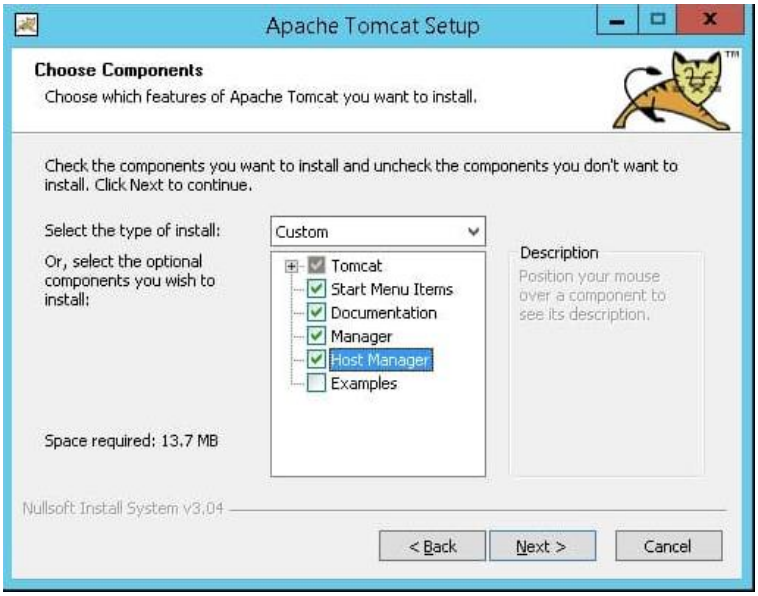



Figura 23: Apache Tomcat instalación
Fuente Elaboración Propia (Año 2021)

- ✓ Cambiamos el puerto por defecto editamos el server.xml

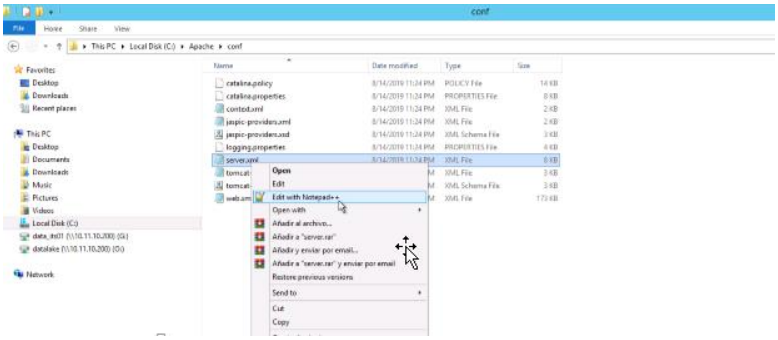


Figura 24: Apache Tomcat modificación de server.xml
Fuente Elaboración Propia (Año 2021)

```
-->

<!-- A "Connector" represents an endpoint by which requests are received
     and responses are returned. Documentation at :
     Java HTTP Connector: /docs/config/http.html
     Java AJP  Connector: /docs/config/ajp.html
     APR (HTTP/AJP) Connector: /docs/apr.html
     Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
-->
<Connector port="8090" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
<!--
```

Figura 25: Apache Tomcat modificación de puerto
Fuente Elaboración Propia (Año 2021)

- ✓ Agregamos las variables de entorno JAVA_HOME: C:\Program File\Java\jdk<version> y JRE_HOME: C:\ Program File\Java\jdk<version>
- ✓ Abrimos en un navegador http://localhost:8090/ se abrirá el Panel de Administración de Tomcat.

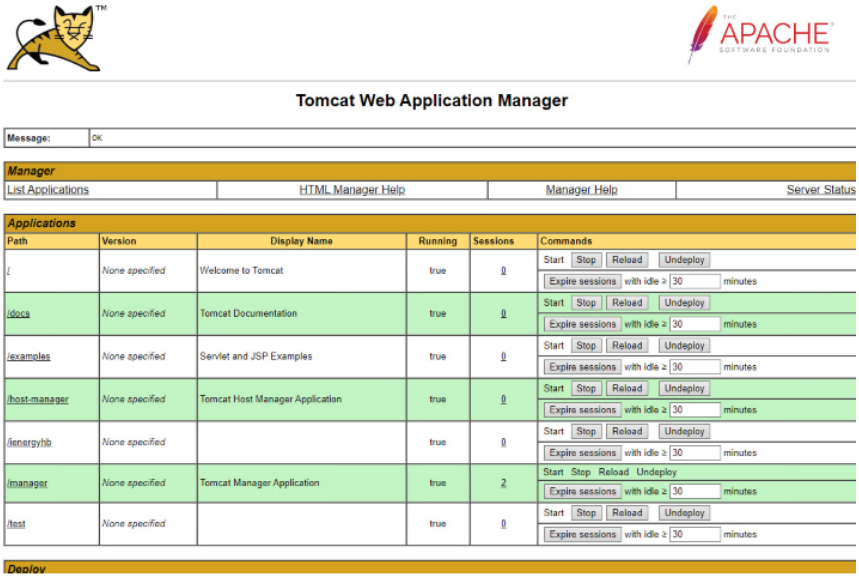



Figura 26: Apache Tomcat herramienta de administración
Fuente Elaboración Propia (Año 2021)

- ✓ Para levantar una nueva versión de nuestro backend procedemos a darle clic en undeploy, cargar el nuevo .WAR y luego clic en deploy.

### 3.2.5.5. Configuración de keycloak

Todo lo relacionado con Gestión de Usuarios como por ejemplo sus roles, autenticación, token, entre otros será administrado por este software para esto debemos seguir los siguientes pasos:

- ✓ Procedemos a descargar la versión 3.0.0 en modo standlone

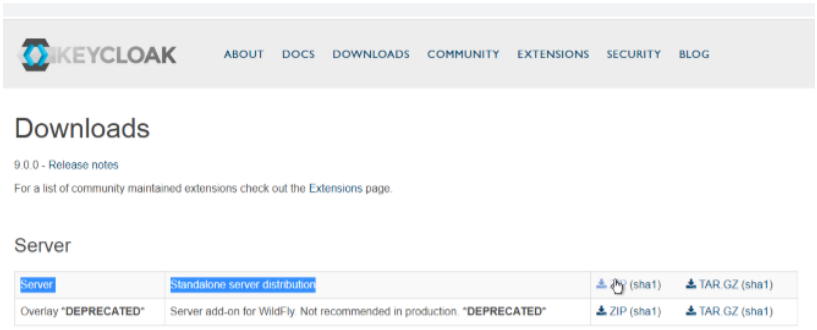



Figura 27: Keycloak descarga
Fuente Elaboración Propia (Año 2021)

- ✓ No dirigimos a la ubicación de la carpeta descargada y corremos la aplicación.

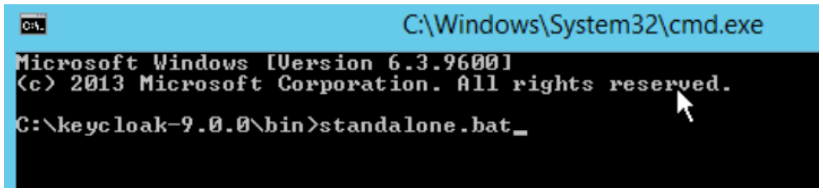



Figura 28: Keycloak instalación standalone.bat
Fuente Elaboración Propia (Año 2021)

- ✓ Crear Realm que viene a ser un objeto de gestión de usuarios

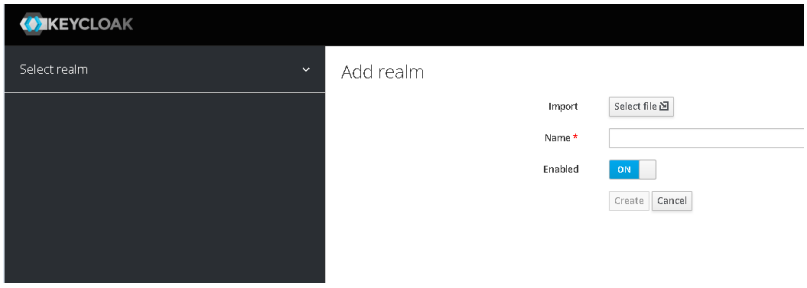



Figura 29: Keycloak agregar realm
Fuente Elaboración Propia (Año 2021)

✓ Crear un Cliente por ejemplo Autenticador FISI

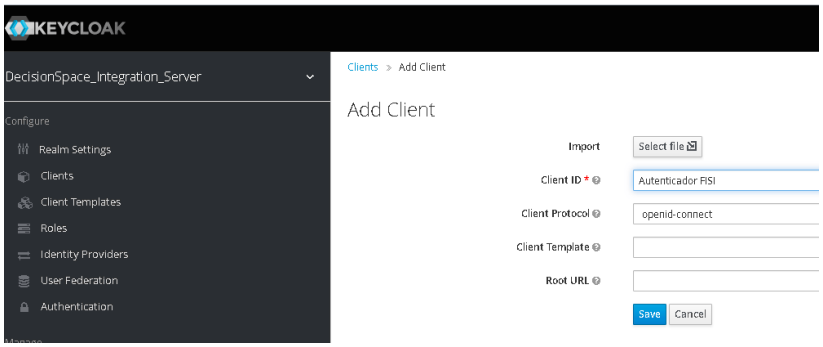



Figura 30: Keycloak agregar cliente
Fuente Elaboración Propia (Año 2021)

✓ Después de crear el cliente Autenticado FISI, procedemos con la configuración por ejemplo en el label de Valid Redirect URIS, ubicamos la URL a donde queremos que se redireccione después de hacer login.

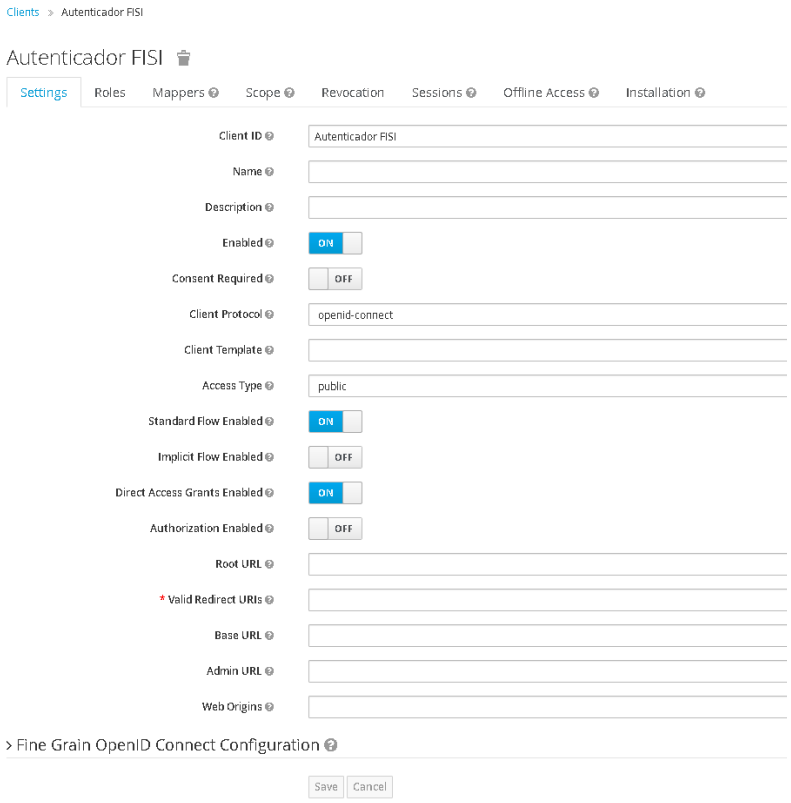



Figura 31: Keycloak zona de configuración
Fuente Elaboración Propia (Año 2021)

### 3.2.5.6. Resultado del desarrollo de dashboards

#### 3.2.5.6.1. Dashboard de órdenes en proceso

##### 3.2.5.6.1.1. Generando la Vista VIEW_ORDERS_PP

| Orden de Trabajo | |
|---|---|
| **Vista** | view_orders_pp |

```
CREATE OR REPLACE VIEW public.view_orders_pp
    AS
     SELECT orders.id AS id_ot,  orders.comments,  orders.comments_task,  orders.datecreate,  orders.datedue,
    orders.management,  orders.observation,  orders.origin,  orders.registerby,  orders.requestedby,  orders.signedby,
    orders.state,  orders.subject,  orders.substate,  orders.typeot,  orders.code,  orders.date_update,  orders.userupdate,
    orders.memo,  orders.editcatalogo,  orders.editdocument,  orders.editwell,  orders.editseismic,  orders.editgis,
    orders.typeotnew,  orders.detalle,  orders.code_order_trabajo,  orders.statetaskreception,  orders.statetaskwell,
    orders.statetaskgis,  orders.statetaskseismic,  orders.statetaskarchive,  orders.statetaskattention,  orders.mesot,
    orders.mescerradoot,  orders.editot,  orders.mescerradoreception,  orders.approvepp,  orders.grupospp,
    orders.dateapprovepp,  orders.datecerradogis,  orders.datecerradowell,  orders.datecerradoseismic,
    orders.datecerradoarchive,  orders.datecerradoot,  orders.mescerradogis,  orders.mescerradowell,  orders.mescerradoseismic,
    orders.mescerradoarchive,  orders.datecerradoreception,  orders.datecerradoattention,  orders.mescerradoattention,
    associated_files.id AS id_associate,  associated_files.barcode,  associated_files.cuenca,  associated_files.ext,
    associated_files.lote,  associated_files.order_id,  associated_files.region,  associated_files.size,  associated_files.status,
    associated_files.typebarcode,  associated_files.uwi,  associated_files.yacimiento,  associated_files.mediobarcode,
    associated_files.datatype,  associated_files.titledocument,  associated_files.campaingname,  associated_files.nameline,
    associated_files.typeseismic,  associated_files.pozouwi,  associated_files.originaldoc,  associated_files.receptypefile,
    associated_files.pozotipo,  associated_files.order_completo

FROM (associated_files LEFT JOIN orders ON (((associated_files.order_id)::text = (orders.id)::text)))  WHERE (associated_files.status =
(0)::numeric)

UNION

SELECT orders.id AS id_ot,  orders.comments,  orders.comments_task,  orders.datecreate,  orders.datedue,
    orders.management,  orders.observation,  orders.origin,  orders.registerby,  orders.requestedby,  orders.signedby,
    orders.state,  orders.subject,  orders.substate,  orders.typeot,  orders.code,  orders.date_update,  orders.userupdate,
    orders.memo,  orders.editcatalogo,  orders.editdocument,  orders.editwell,  orders.editseismic,  orders.editgis,
    orders.typeotnew,  orders.detalle,  orders.code_order_trabajo,  orders.statetaskreception,  orders.statetaskwell,
    orders.statetaskgis,  orders.statetaskseismic,  orders.statetaskarchive,  orders.statetaskattention,  orders.mesot,
    orders.mescerradoot,  orders.editot,  orders.mescerradoreception,  orders.approvepp,  orders.grupospp,
    orders.dateapprovepp,  orders.datecerradogis,  orders.datecerradowell,  orders.datecerradoseismic,
    orders.datecerradoarchive,  orders.datecerradoot,  orders.mescerradogis,  orders.mescerradowell,  orders.mescerradoseismic,
    orders.mescerradoarchive,  orders.datecerradoreception,  orders.datecerradoattention,  orders.mescerradoattention,
    associated_files.id AS id_associate,  associated_files.barcode,  associated_files.cuenca,  associated_files.ext,
    associated_files.lote,  associated_files.order_id,  associated_files.region,  associated_files.size,  associated_files.status,
    associated_files.typebarcode,  associated_files.uwi,  associated_files.yacimiento,  associated_files.mediobarcode,
    associated_files.datatype,  associated_files.titledocument,  associated_files.campaingname,  associated_files.nameline,
    associated_files.typeseismic,  associated_files.pozouwi,  associated_files.originaldoc,  associated_files.receptypefile,
    associated_files.pozotipo,  associated_files.order_completo

    FROM (orders
    LEFT JOIN associated_files ON (((associated_files.order_id)::text = (orders.id)::text)))
    WHERE (associated_files.order_id IS NULL);
```

Tabla 22 Dashboard generando la vista View_Orders_PP
Fuente Elaboración Propia (Año 2021)

### 3.2.5.6.1.2. Creando el modelo orders en Tibco Spotfire

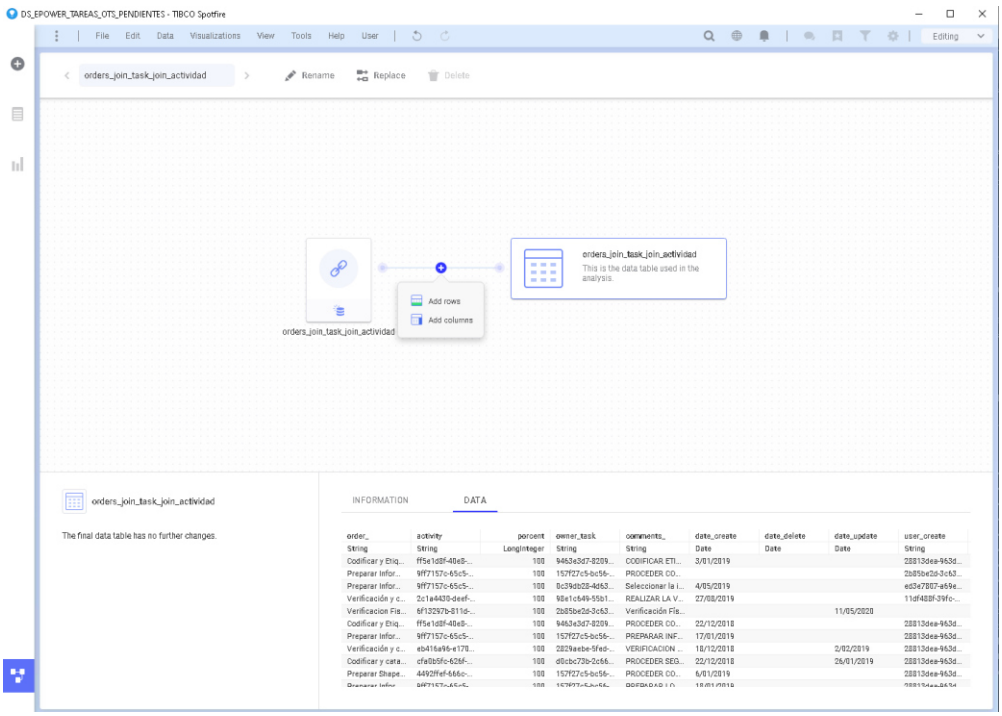



Figura 32 Dashboard Transformando datos de la vista View_Orders_PP
Fuente Elaboración Propia (Año 2021)

### 3.2.5.6.1.3. Creando los gráficos en Tibco Spotfire

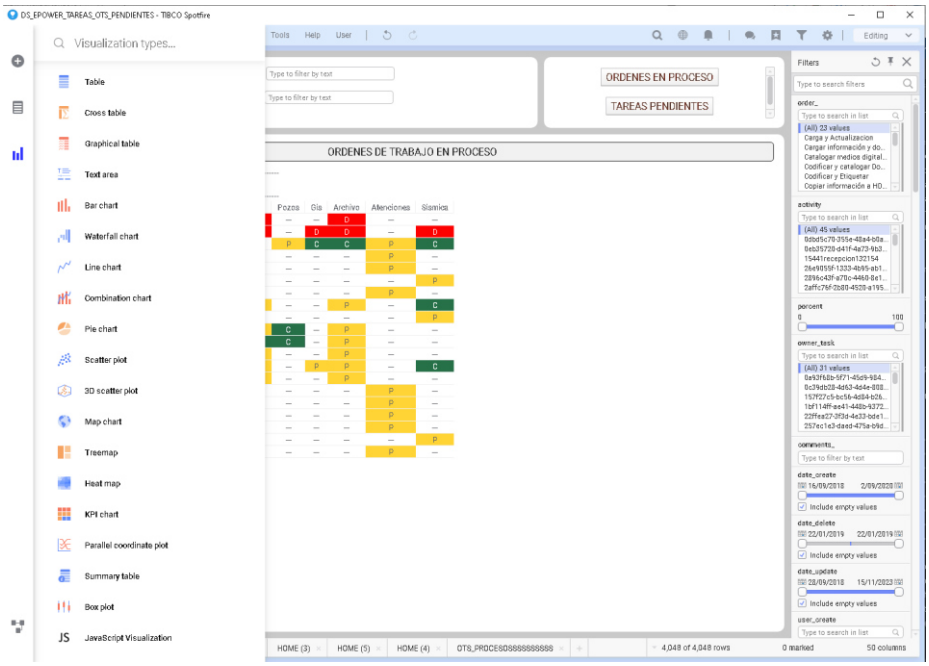



Figura 33: Dashboard agregando gráficos en base a la vista View_Orders_PP
Fuente Elaboración Propia (Año 2021)

### 3.2.5.6.2. Dashboard de tareas en proceso

#### 3.2.5.6.2.1. Generando la vista orders_join_task

| Tareas | |
|---|---|
| **Vista** | orders_join_task |

```
CREATE OR REPLACE VIEW public.orders_join_task
    AS
     SELECT tasks.id,  tasks.order_,  tasks.activity,  tasks.porcent,  tasks.owner_task,  tasks.comments_,  tasks.order_id,
    tasks.status,  tasks.date_create,  tasks.date_delete,  tasks.date_update,  tasks.user_create,  tasks.user_delete,
    tasks.user_update,  orders.code_order_trabajo,  orders.code,  orders.memo,  orders.comments,  orders.comments_task,
    orders.datecreate,  orders.datedue,  orders.management,  orders.observation,  orders.origin,  orders.registerby,
    orders.requestedby,  orders.signedby,  orders.state,  orders.subject,  orders.substate,  orders.typeot,  orders.userupdate,
    orders.editcatalogo,  orders.editdocument,  orders.editwell,  orders.editseismic,  orders.editgis,  orders.typeotnew,
    orders.detalle

   FROM (tasks     JOIN orders ON (((orders.id)::text = (tasks.order_id)::text)))
  WHERE (tasks.status = 0);
```

Tabla 23: Dashboard generando la vista Orders_Join_Task
Fuente Elaboración Propia (Año 2021)

#### 3.2.5.6.2.2. Creando el modelo tareas en Tibco Spotfire

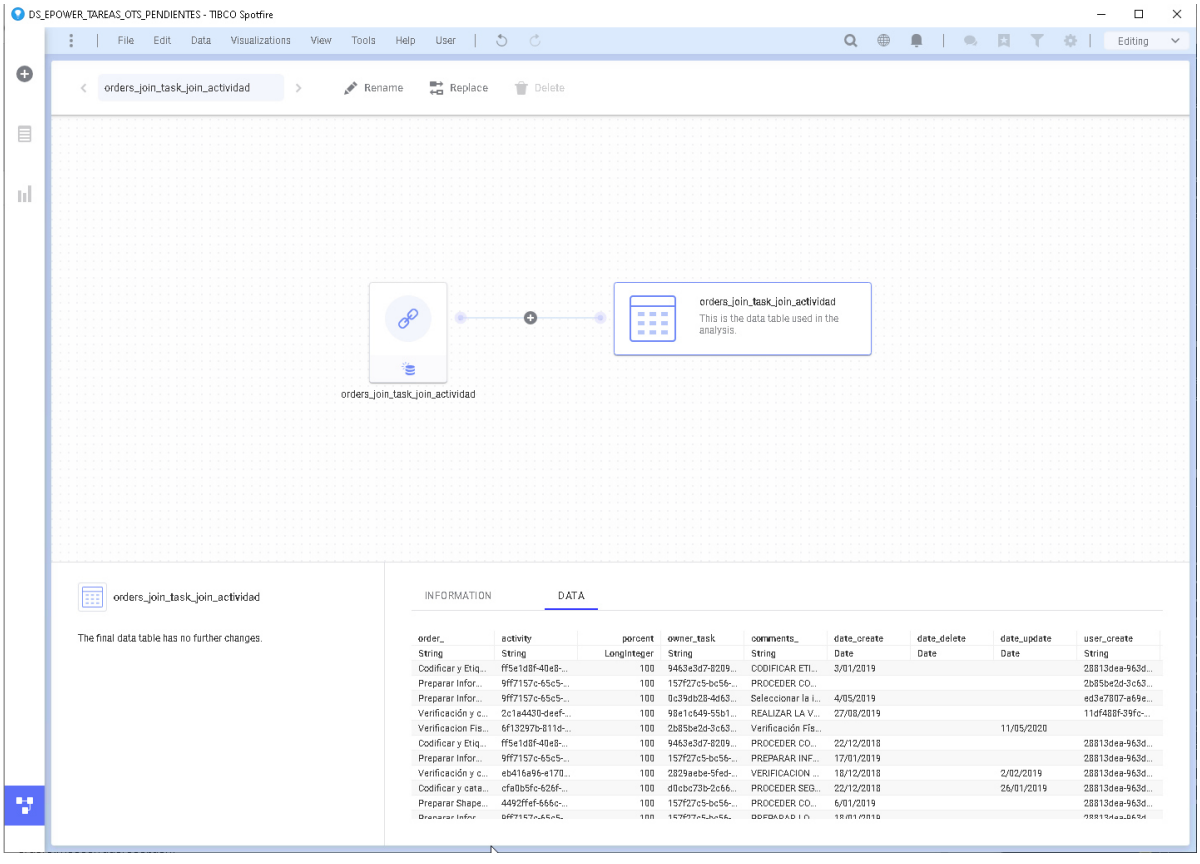



Figura 34: Dashboard transformando datos de la Vista Orders_Join_Task
Fuente Elaboración Propia (Año 2021)

### 3.2.5.6.2.3. Creando los gráficos en Tibco Spotfire

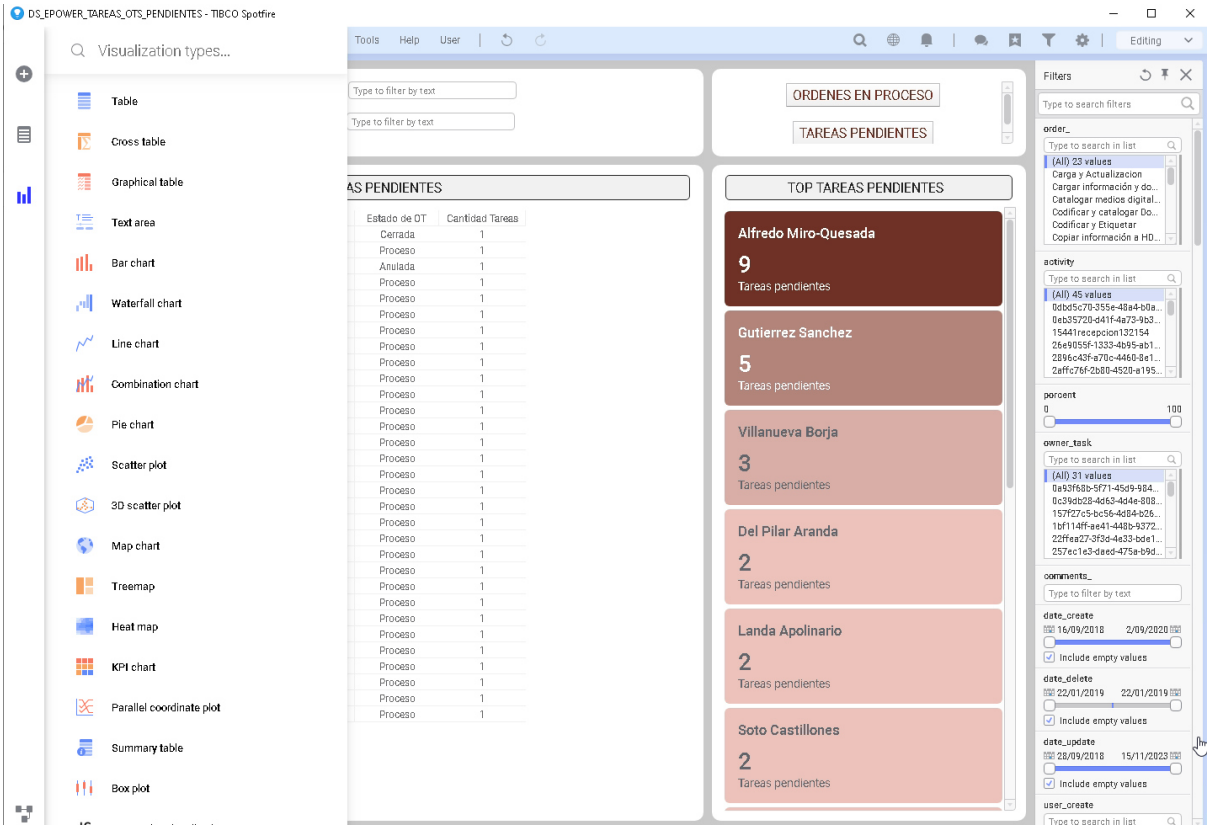



Figura 35: Dashboard agregando gráficos en base a la Vista Orders_Join_Task
Fuente Elaboración Propia (Año 2021)

### 3.2.5.6.3. Dashboard gerencial

#### 3.2.5.6.3.1. Generando la vista view_memos

| Gerencial | |
|---|---|
| **Vista** | Vista_memos |

```
CREATE OR REPLACE VIEW public.vista_memos
  AS
  SELECT orders.id,  orders.comments,  orders.comments_task,  orders.datecreate,  orders.datedue,  orders.management,
  orders.observation,  orders.origin,  orders.registerby,  orders.requestedby,  orders.signedby,  orders.state,  orders.subject,
  orders.substate,  orders.typeot,  orders.code,  orders.date_update,  orders.userupdate,  orders.memo,  orders.editcatalogo,
  orders.editdocument,  orders.editwell,  orders.editseismic,  orders.editgis,  orders.typeotnew,  orders.detalle,
  orders.code_order_trabajo,  orders.statetaskreception,  orders.statetaskwell,  orders.statetaskgis,  orders.statetaskseismic,
  orders.statetaskarchive,  orders.statetaskattention,  orders.mesot,  orders.mescerradoot,  orders.editot,
  orders.mescerradoreception,  orders.approvepp,  orders.grupospp,  orders.dateapprovepp,  orders.datecerradogis,
  orders.datecerradowell,  orders.datecerradoseismic,  orders.datecerradoarchive,  orders.datecerradoot,  orders.mescerradogis,
  orders.mescerradowell,  orders.mescerradoseismic,  orders.mescerradoarchive,  orders.datecerradoreception,
  orders.datecerradoattention,  orders.mescerradoattention

  FROM orders
 WHERE (upper((orders.memo)::text) !~~ '%EMAIL%'::text);
```

Tabla 24: Dashboard generando la vista View_Memos
Fuente Elaboración Propia (Año 2021)

### 3.2.5.6.3.2. Creando el modelo view_memo en tibco spotfire

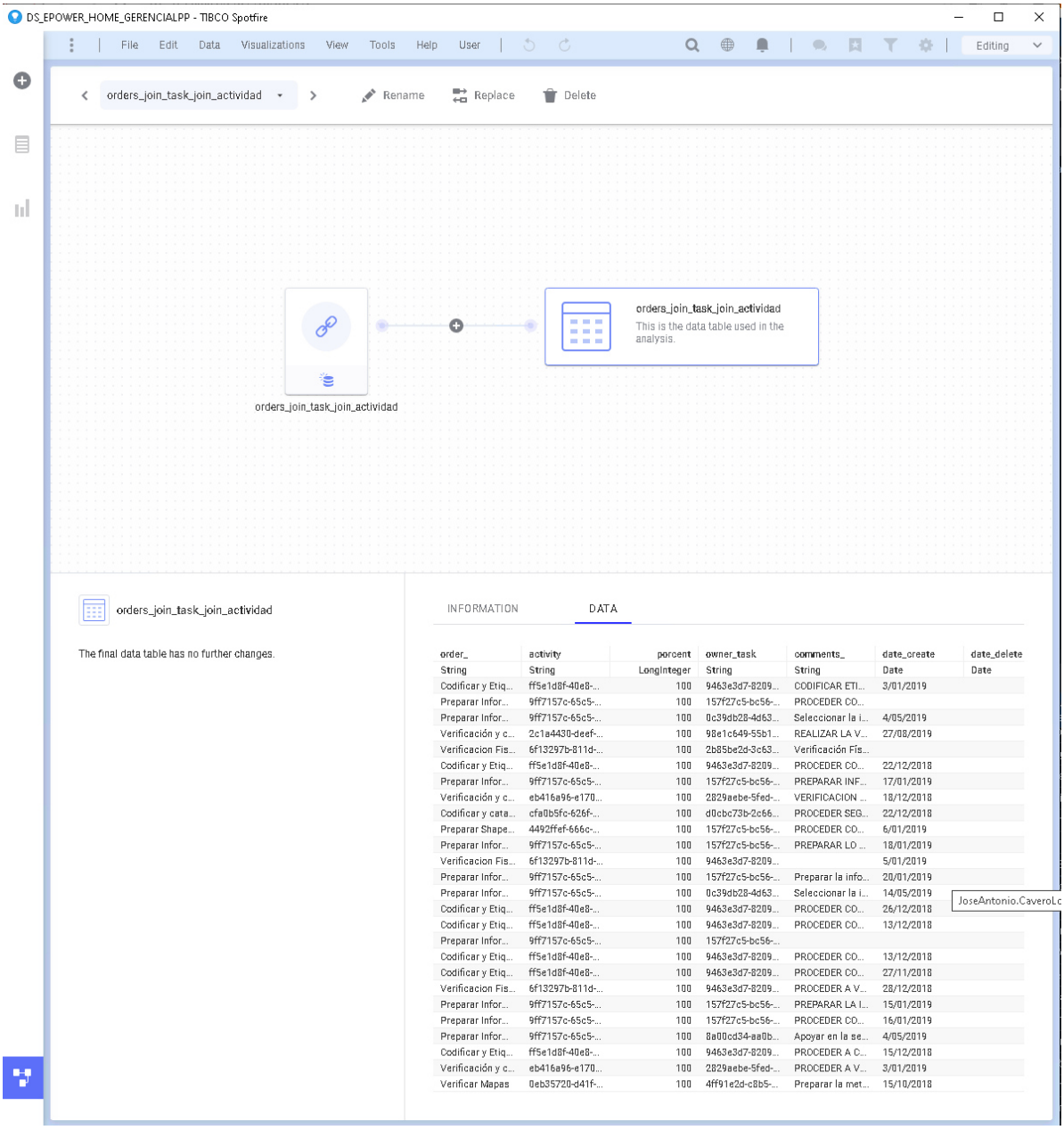



Figura 36: Dashboard transformando datos de la Vista View_Memos
Fuente Elaboración Propia (Año 2021)

#### 3.2.5.6.3.3. Creando los gráficos en Tibco Spotfire

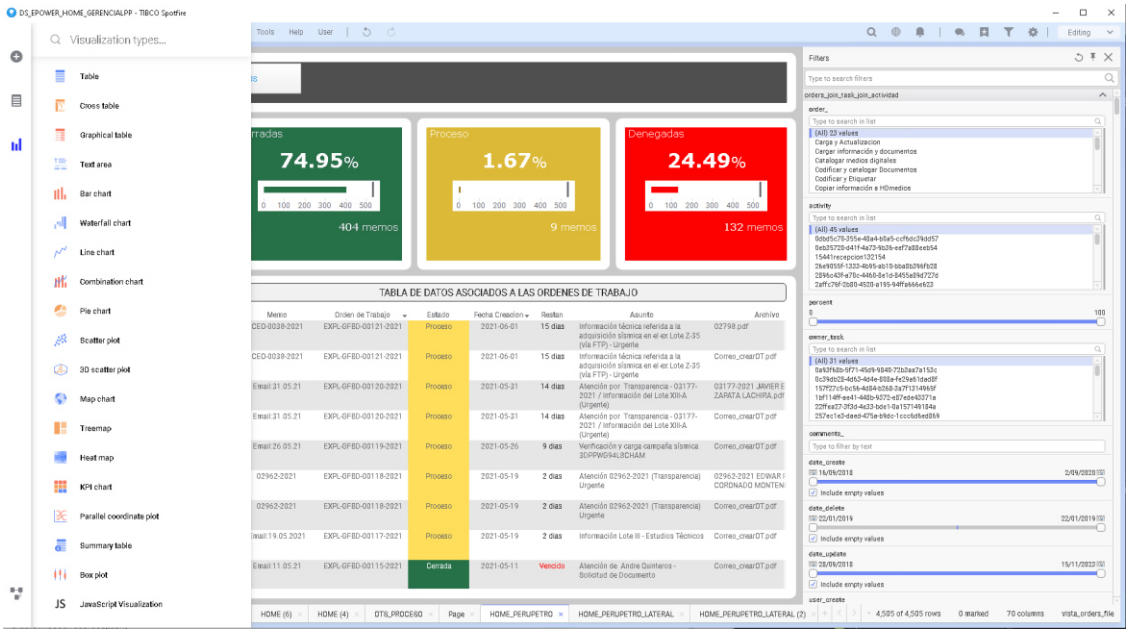



Figura 37: Dashboard agregando gráficos en base a la Vista View_Memos
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7. Resultados del desarrollo de frontend

Por motivos de seguridad la información mostrada es de las principales interfaces, los datos en cada imagen son datos de prueba.

#### 3.2.5.7.1. Módulo login

##### 3.2.5.7.1.1. Interface de login




Figura 38: Frontend interfaz login
Fuente Elaboración Halliburton (Año 2018)

### 3.2.5.7.2. Módulo gestión de solicitudes de una O.T.

#### 3.2.5.7.2.1. Listado de órdenes de trabajo

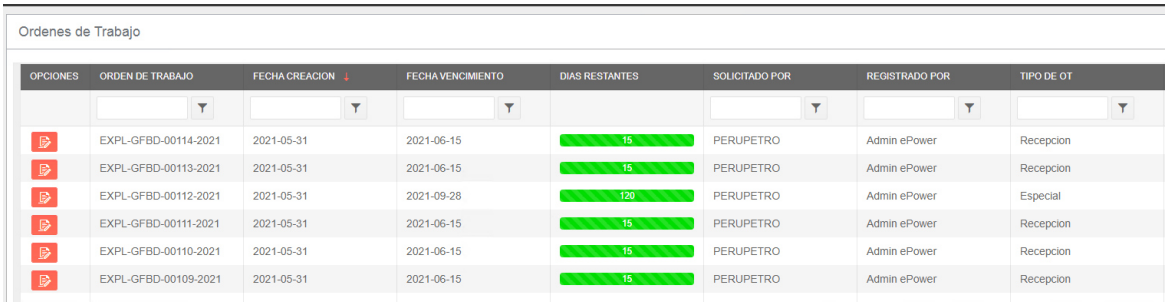



Figura 39: Frontend interface Listado de Ordenes de Trabajo parte 1 de 2
Fuente Elaboración Propia (Año 2021)

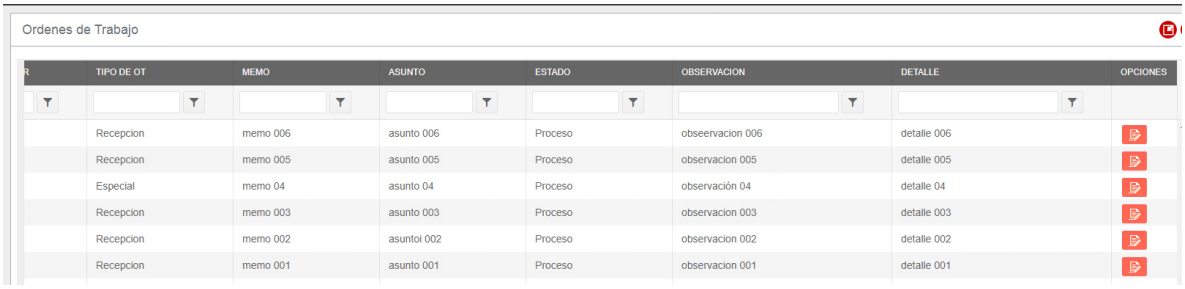



Figura 40: Frontend interface Listado de Ordenes de Trabajo parte 2 de 2
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7.2.2. Consulta de orden de trabajo

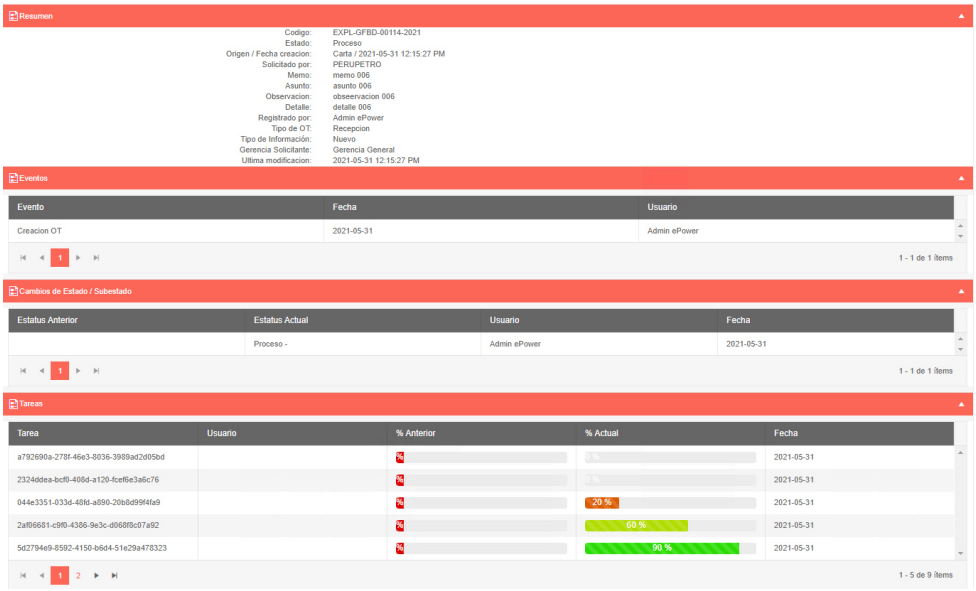



Figura 41: Frontend interface Consulta de Orden de Trabajo
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7.2.3. Registrar orden de trabajo

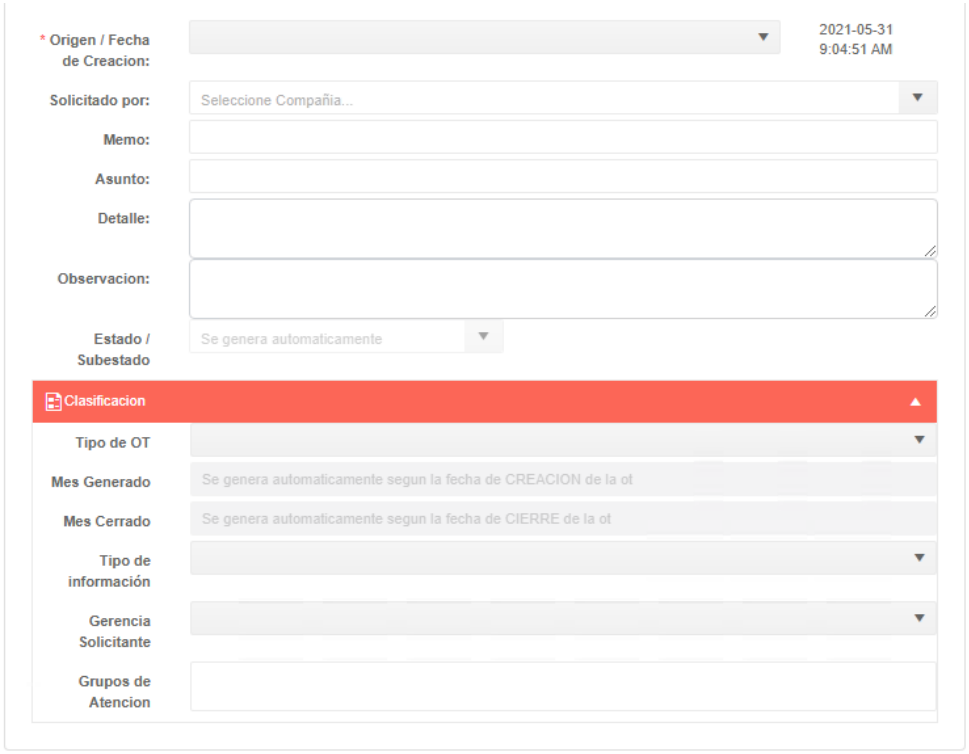



Figura 42: Frontend interfaz Registrar Orden de Trabajo
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7.3. Módulo de gestión de tareas

#### 3.2.5.7.3.1. Registrar tarea

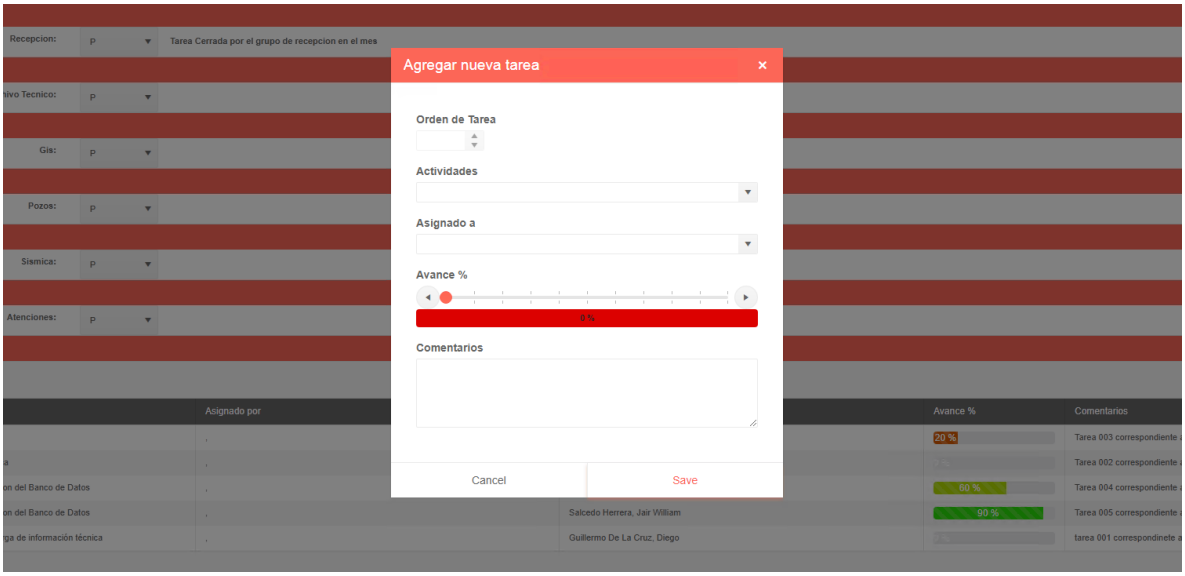



Figura 43: Frontend interface Registrar Tarea
Fuente Elaboración Propia (Año 2021)

#### 3.2.5.7.3.2. Listado de tareas

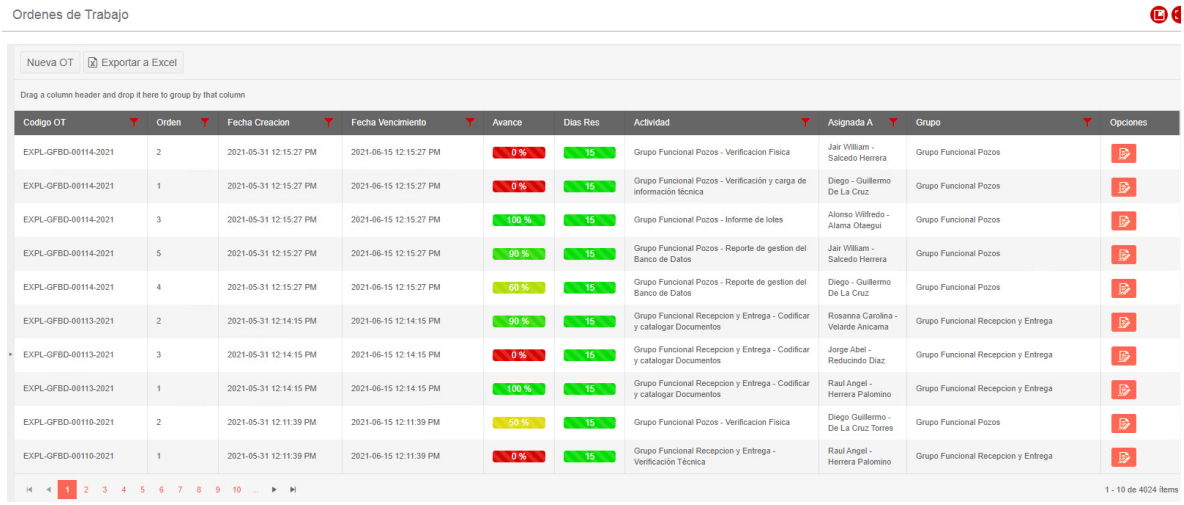



Figura 44: Frontend interface Listado de Tareas
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7.3.3. Consulta tarea

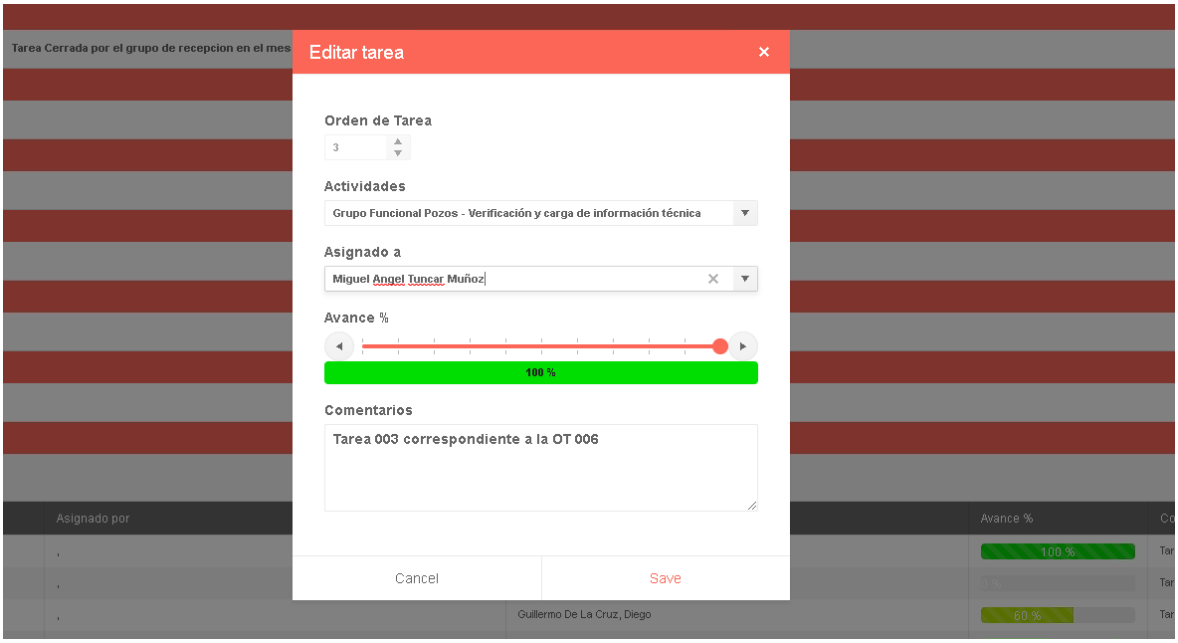



Figura 45: Frontend interface Consulta de Tarea
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7.3.4. Consulta tracking tarea

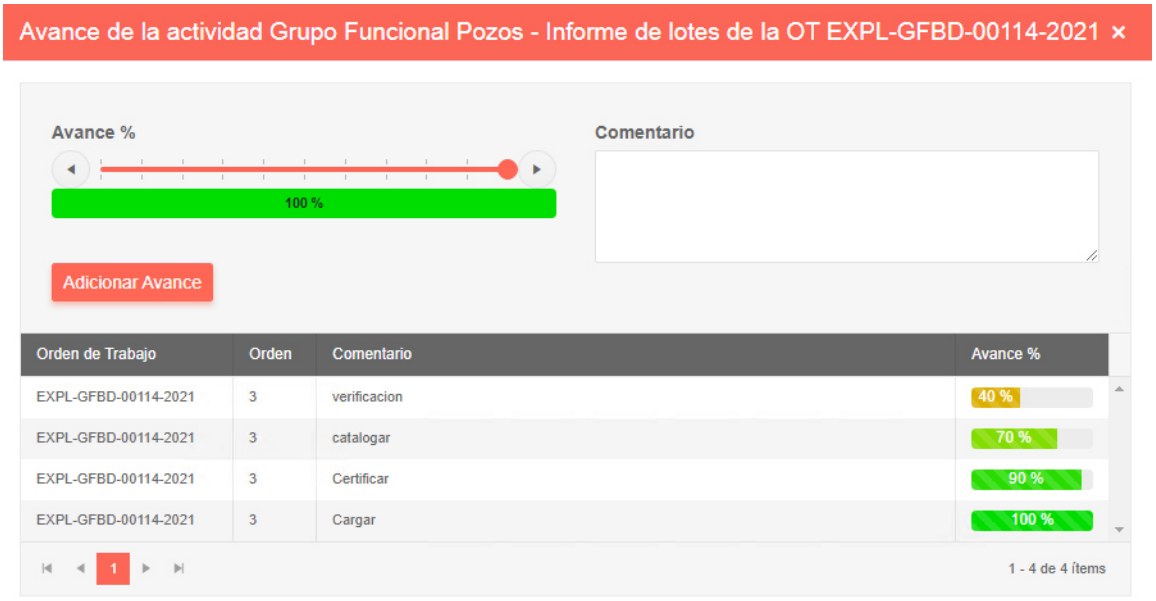



Figura 46: Frontend interface Consulta Tracking de Tarea
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7.4. Módulo de gestión de metadata

#### 3.2.5.7.4.1. Cargar metadata

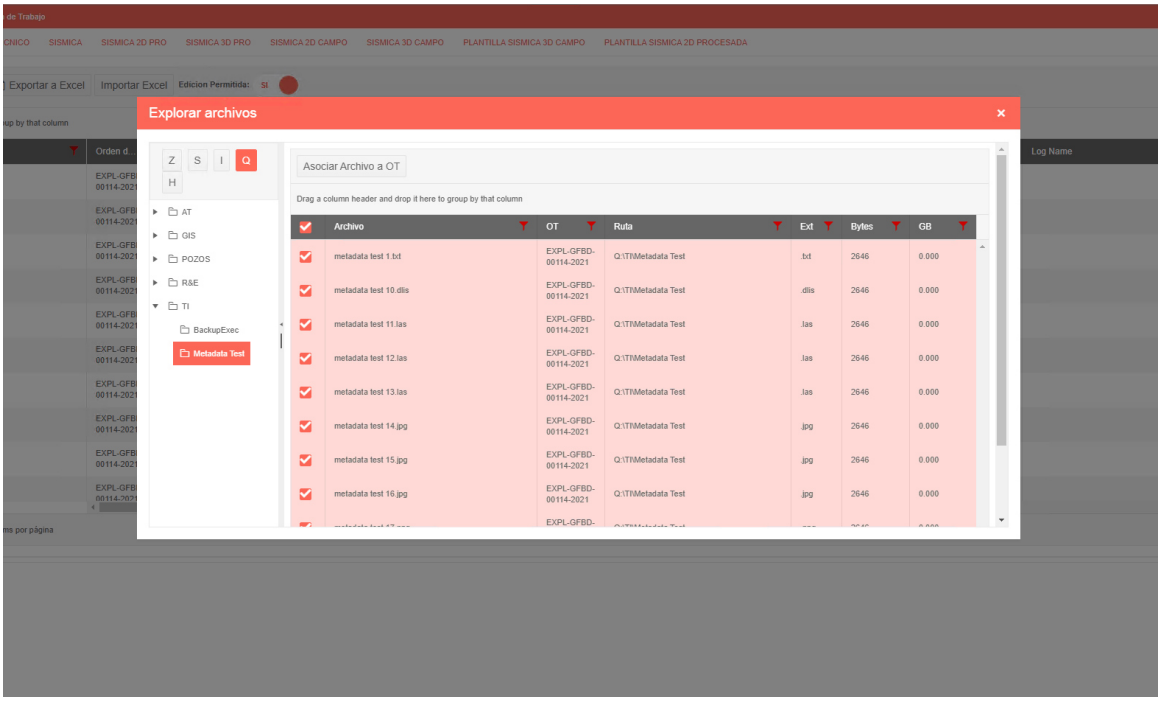



Figura 47: Frontend interface Cargar Metadata
Fuente Elaboración Propia (Año 2021)

#### 3.2.5.7.4.2. Registrar de metadata catalogo

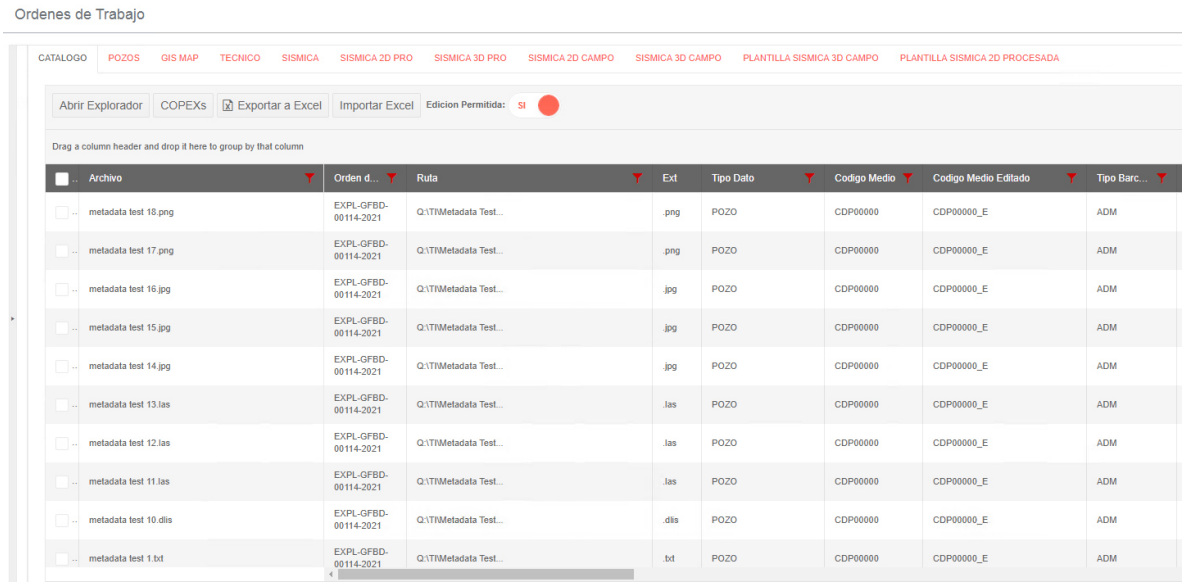



Figura 48: Frontend interface Registrar Metadata de Catalogo parte 1 de 2
Fuente Elaboración Propia (Año 2021)

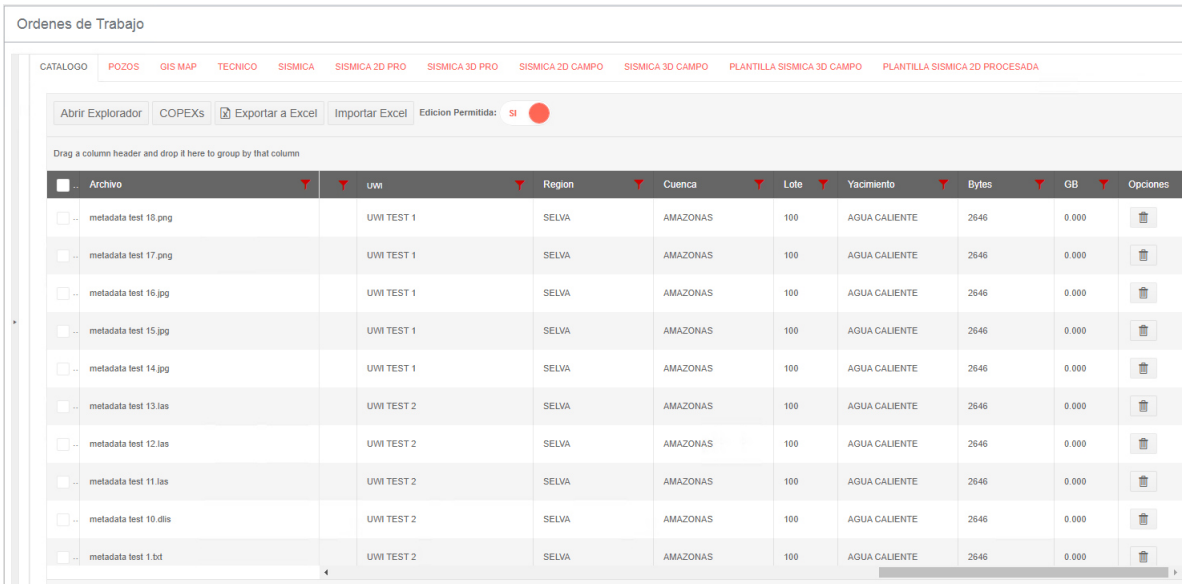



Figura 49: Frontend interface Registrar Metadata de Catalogo parte 2 de 2
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7.4.3. Registrar de metadata archivo técnico

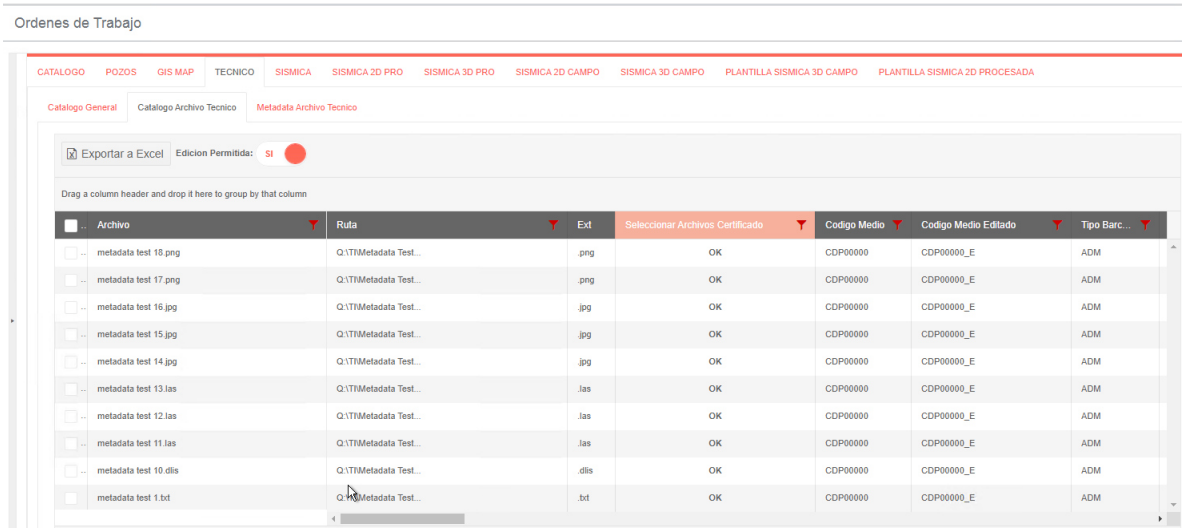



Figura 50: Frontend interface Registrar Metadata de Archivo parte 1 de 6
Fuente Elaboración Propia (Año 2021)

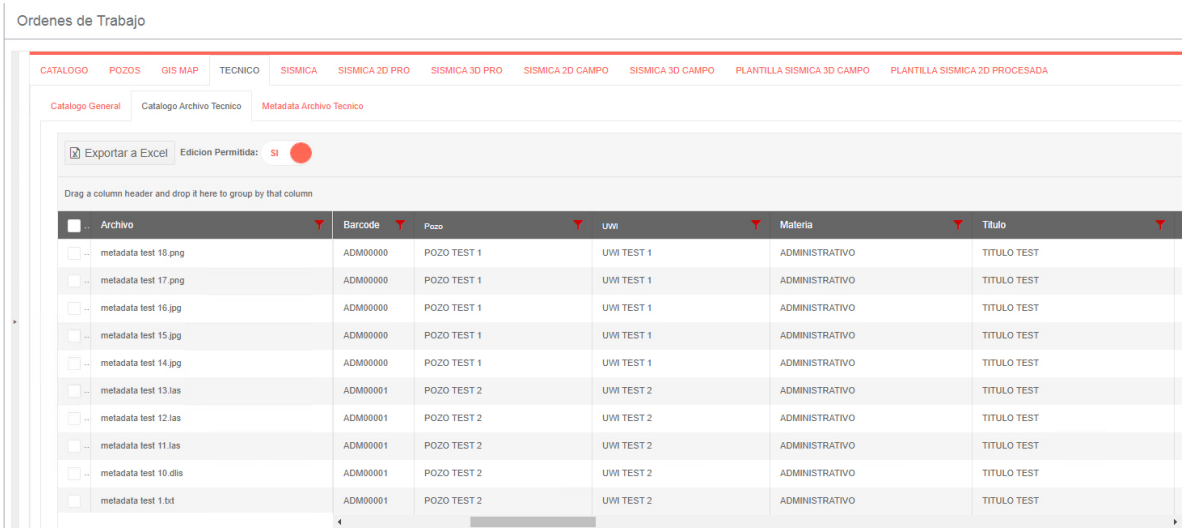



Figura 51: Frontend interface Registrar Metadata de Archivo parte 2 de 6
Fuente Elaboración Propia (Año 2021)

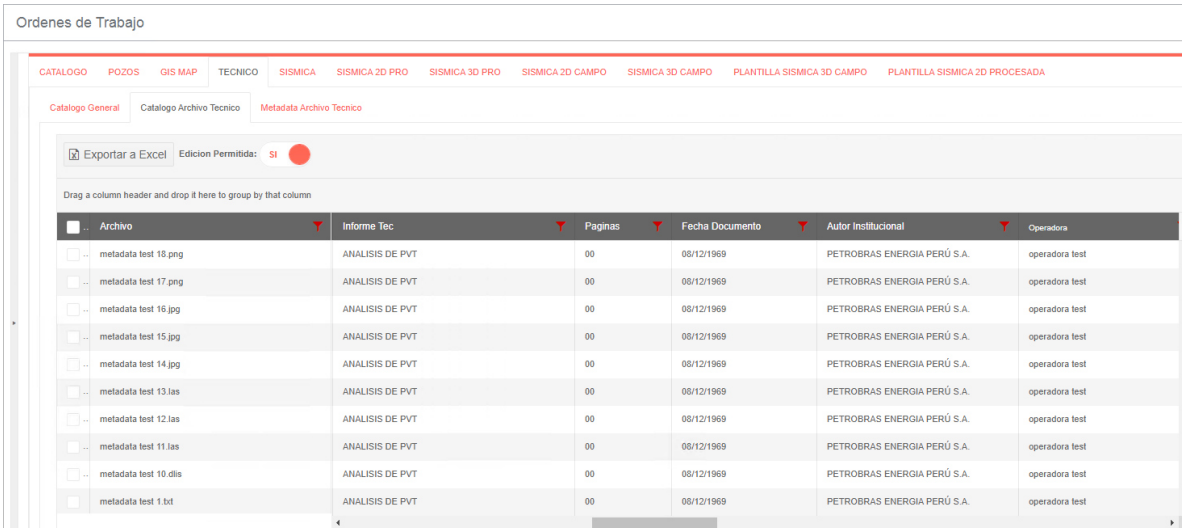



Figura 52: Frontend interface Registrar Metadata de Archivo parte 3 de 6
Fuente Elaboración Propia (Año 2021)

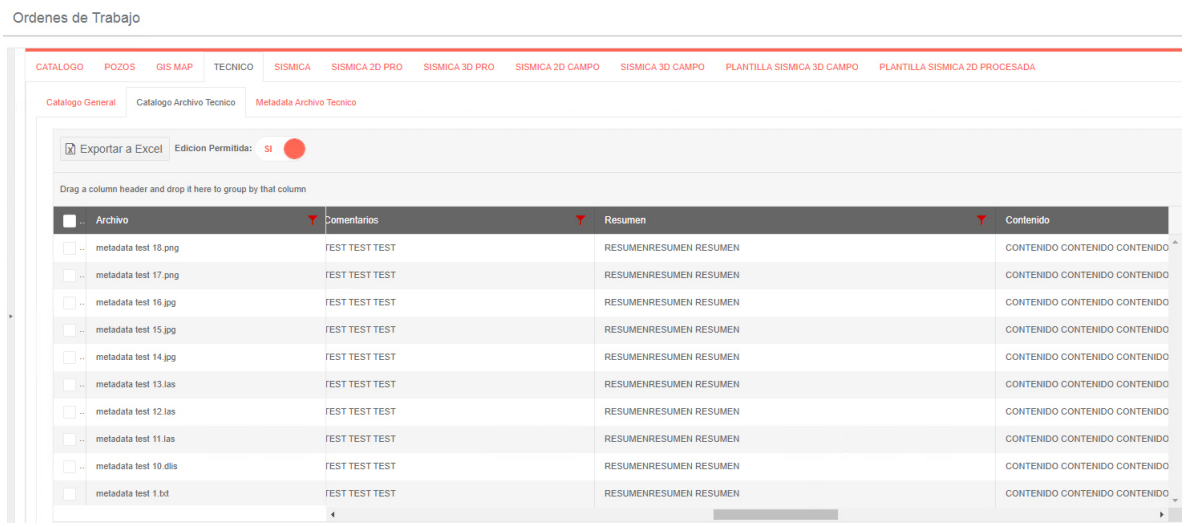



Figura 53: Frontend interface Registrar Metadata de Archivo parte 4 de 6
Fuente Elaboración Propia (Año 2021)

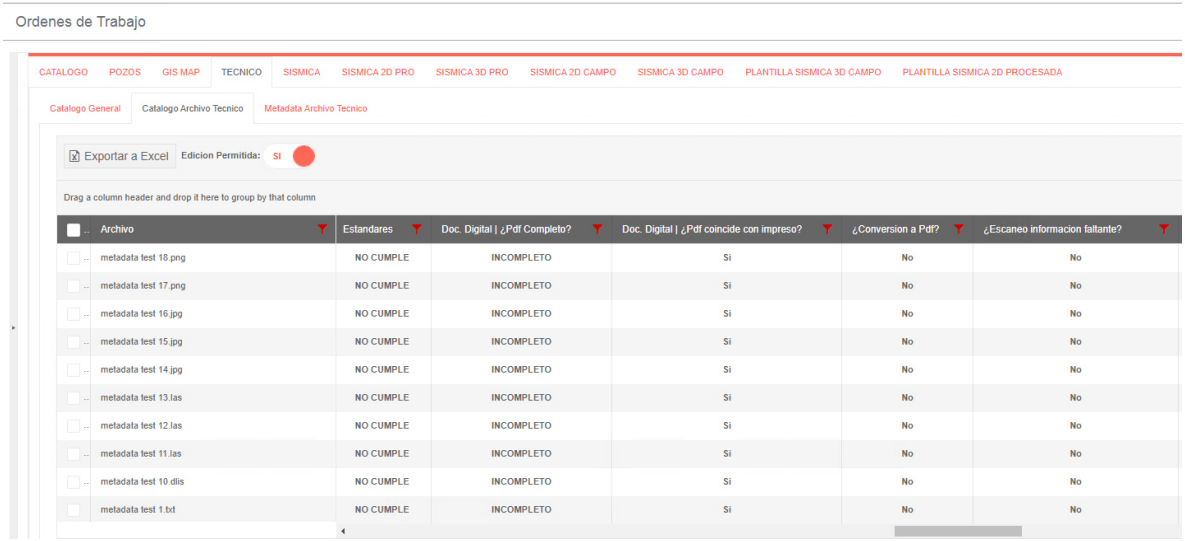



Figura 54: Frontend interface Registrar Metadata de Archivo parte 5 de 6
Fuente Elaboración Propia (Año 2021)

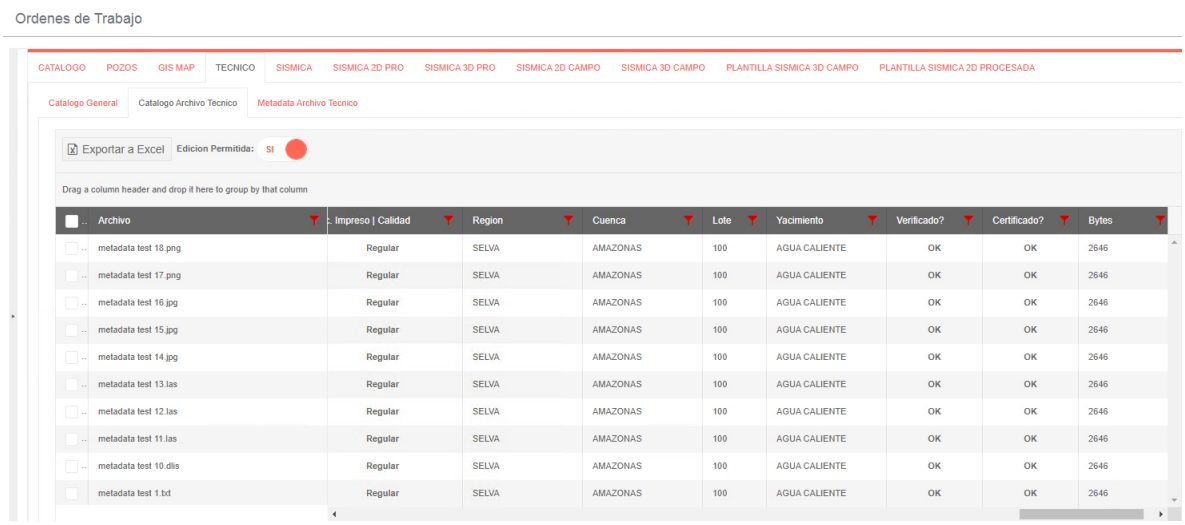



Figura 55: Frontend interface Registrar Metadata de Archivo parte 6 de 6
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7.4.4. Registrar de metadata GIS

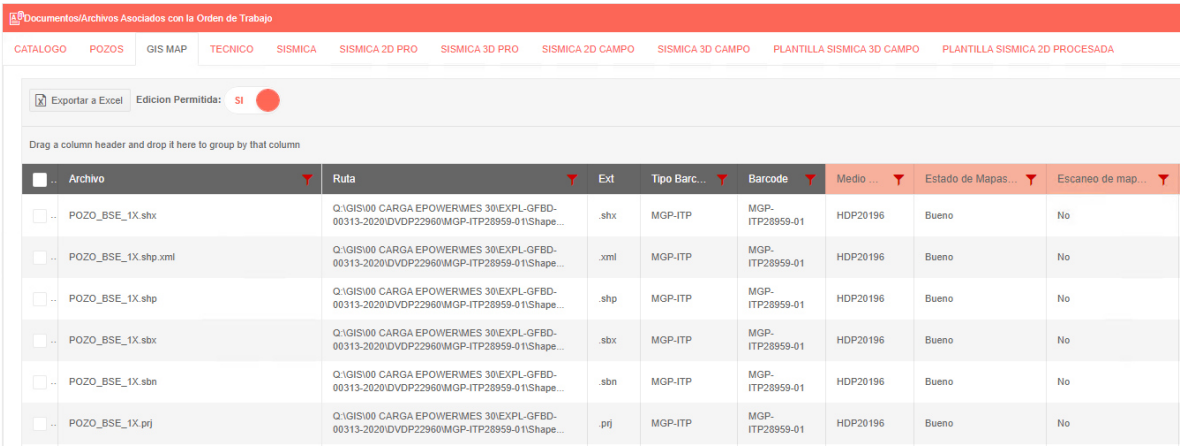



Figura 56: Frontend interface Registrar Metadata de GIS parte 1 de 5
Fuente Elaboración Propia (Año 2021)

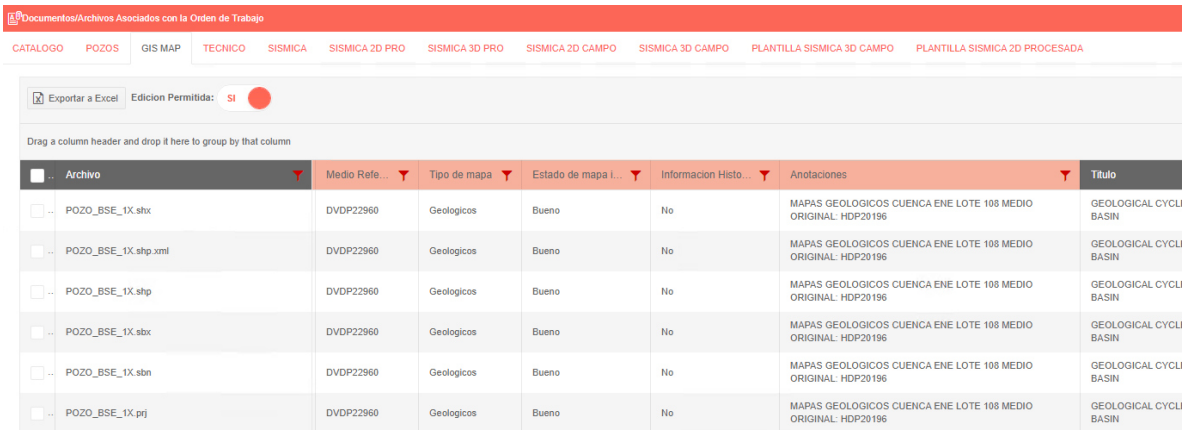



Figura 57: Frontend interface Registrar Metadata de GIS parte 2 de 5
Fuente Elaboración Propia (Año 2021)

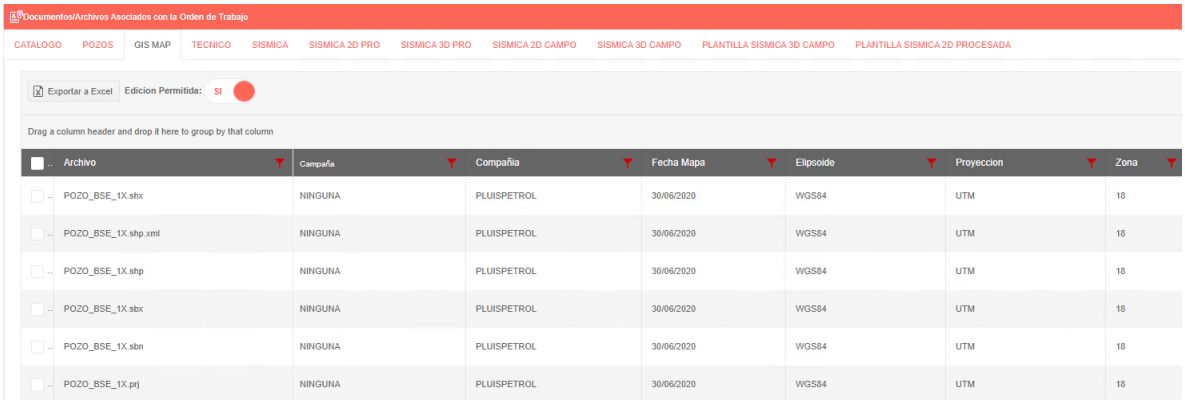



Figura 58: Frontend interface Registrar Metadata de GIS parte 3 de 5
Fuente Elaboración Propia (Año 2021)

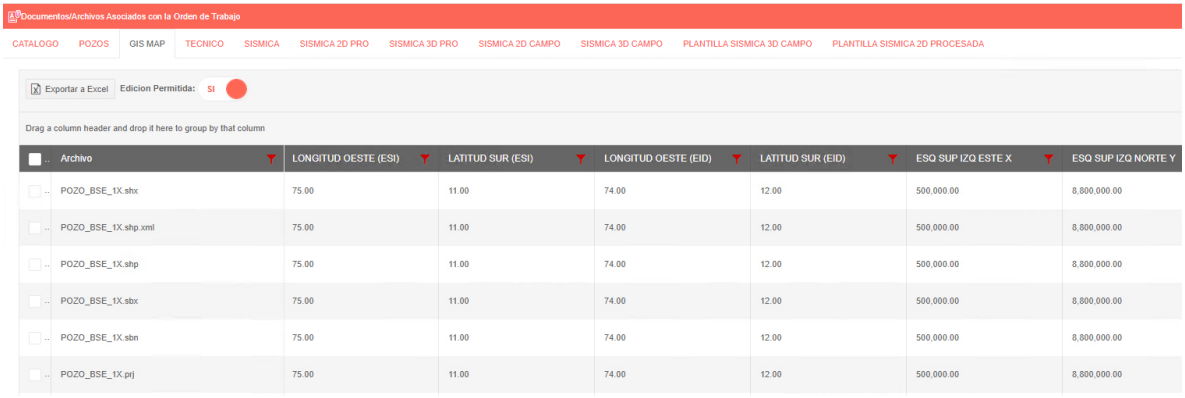



Figura 59: Frontend interface Registrar Metadata de GIS parte 4 de 5
Fuente Elaboración Propia (Año 2021)

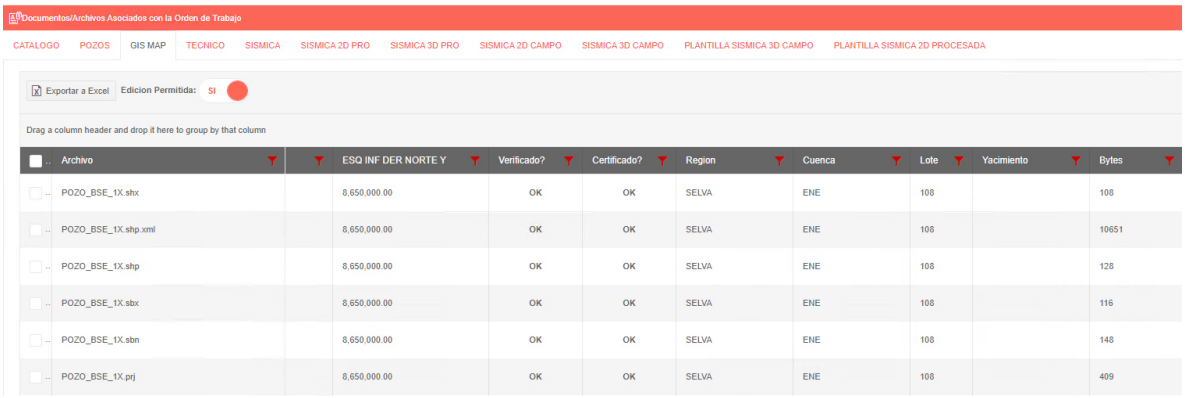



Figura 60: Frontend interface Registrar Metadata de GIS parte 5 de 5
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7.4.5. Registrar de metadata pozos




Figura 61: Frontend interface Registrar Metadata de Pozo parte 1 de 11
Fuente Elaboración Propia (Año 2021)




Figura 62: Frontend interface Registrar Metadata de Pozo parte 2 de 11
Fuente Elaboración Propia (Año 2021)

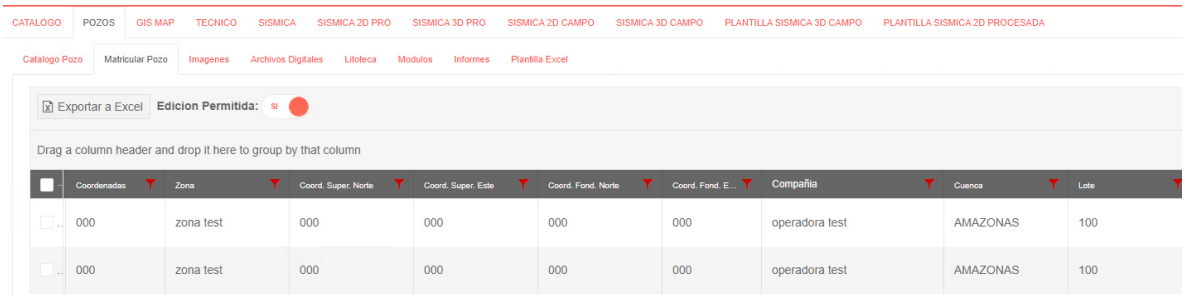



Figura 63: Frontend interface Registrar Metadata de Pozo parte 3 de 11
Fuente Elaboración Propia (Año 2021)




Figura 64: Frontend interface Registrar Metadata de Pozo parte 4 de 11
Fuente Elaboración Propia (Año 2021)

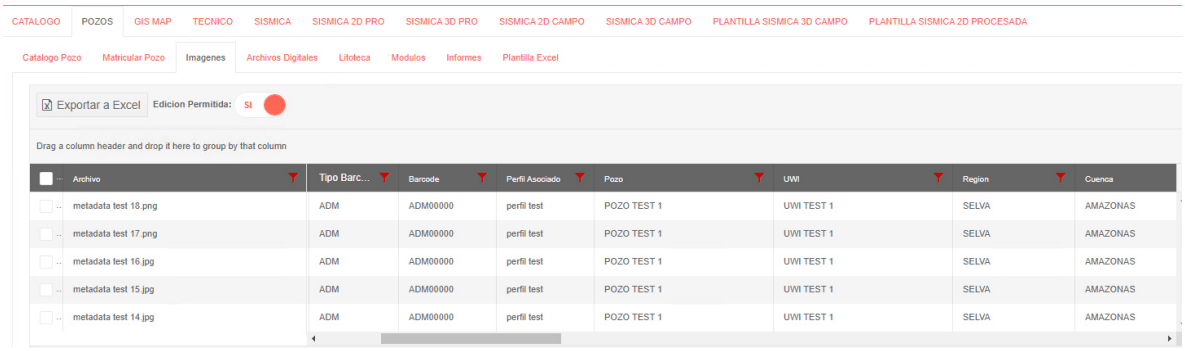



Figura 65: Frontend interface Registrar Metadata de Pozo parte 5 de 11
Fuente Elaboración Propia (Año 2021)




Figura 66: Frontend interface Registrar Metadata de Pozo parte 6 de 11
Fuente Elaboración Propia (Año 2021)




Figura 67: Frontend interface Registrar Metadata de Pozo parte 7 de 11
Fuente Elaboración Propia (Año 2021)

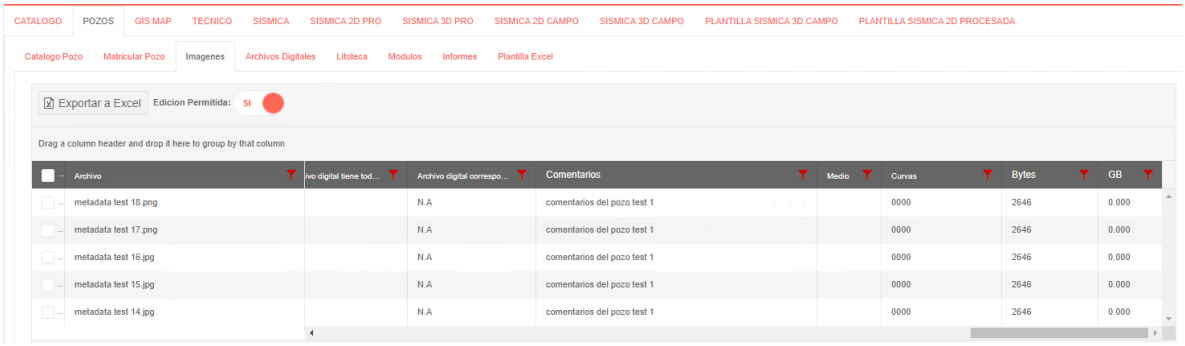



Figura 68: Frontend interface Registrar Metadata de Pozo parte 8 de 11
Fuente Elaboración Propia (Año 2021)

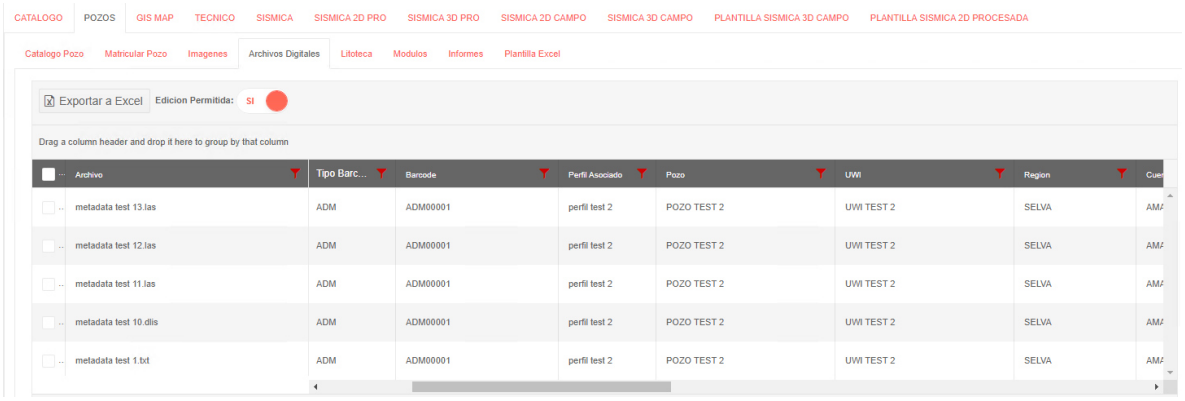



Figura 69: Frontend interface Registrar Metadata de Pozo parte 9 de 11
Fuente Elaboración Propia (Año 2021)

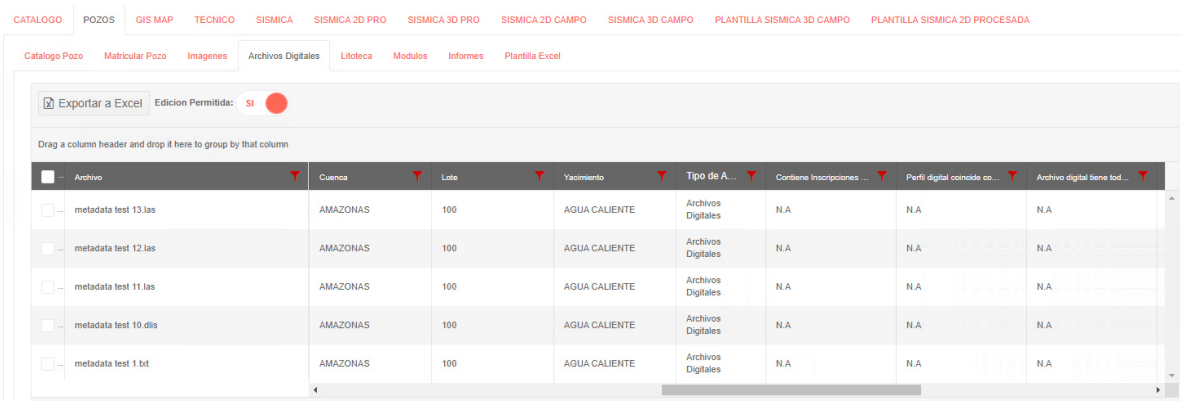



Figura 70: Frontend interface Registrar Metadata de Pozo parte 10 de 11
Fuente Elaboración Propia (Año 2021)

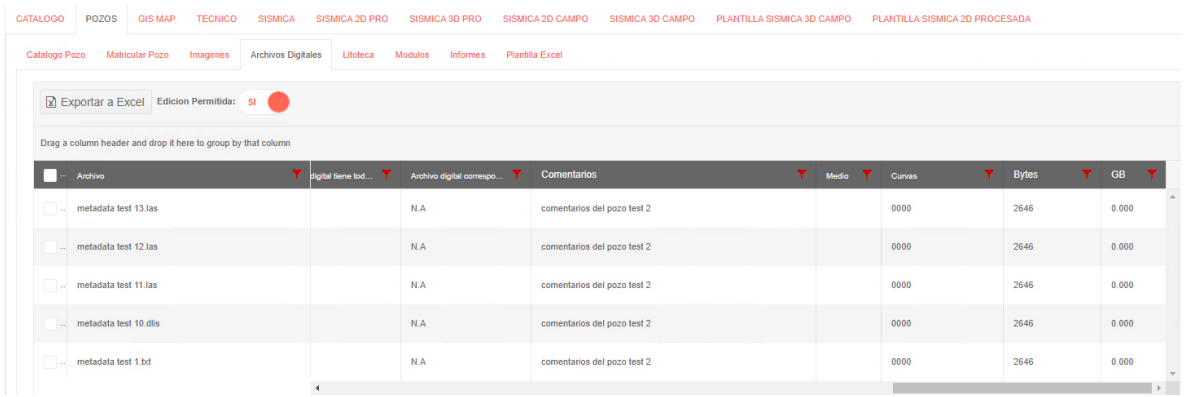



Figura 71: Frontend interface Registrar Metadata de Pozo parte 11 de 11
Fuente Elaboración Propia (Año 2021)

#### 3.2.5.7.5. Módulo de generación de certificados

##### 3.2.5.7.5.1. Alerta de códigos barcodes duplicados

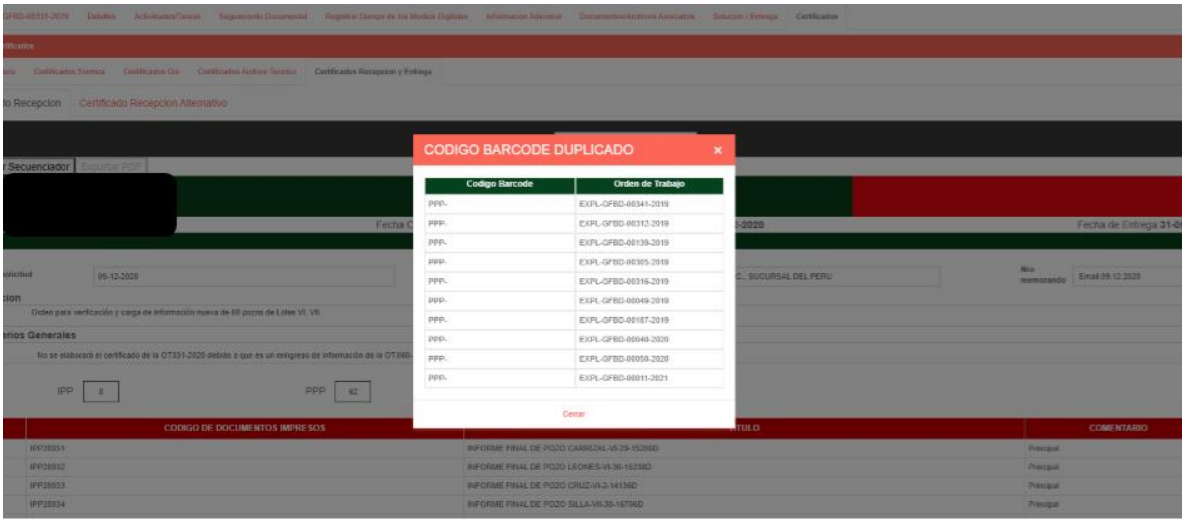



Figura 72: Frontend interface Alerta códigos duplicados
Fuente Elaboración Propia (Año 2021)

##### 3.2.5.7.5.2. Generación de certificados de recepción y entrega

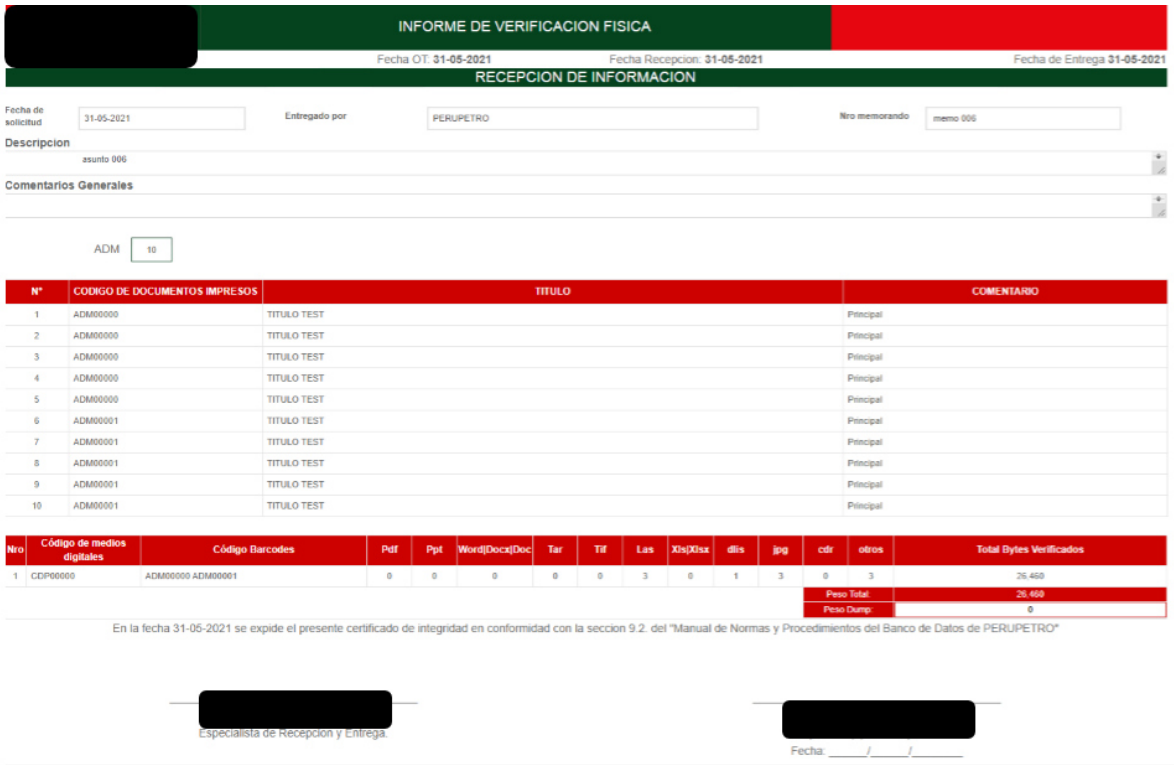



Figura 73: Frontend interface Certificado del Grupo de Recepción
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7.5.3. Generación de certificados de archivo técnico

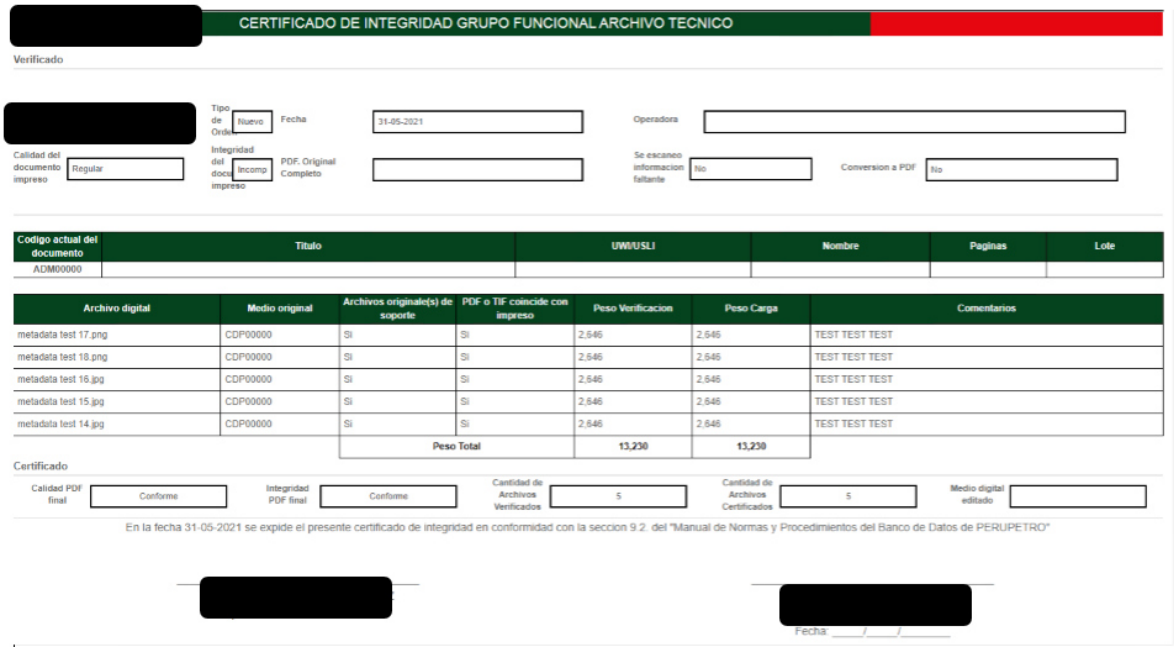



Figura 74: Frontend interface Certificado del Grupo de Archivo Técnico
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7.5.4. Generación de certificados de GIS




Figura 75: Frontend interface Certificado del Grupo de GIS
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7.5.5. Generación de certificados de pozos

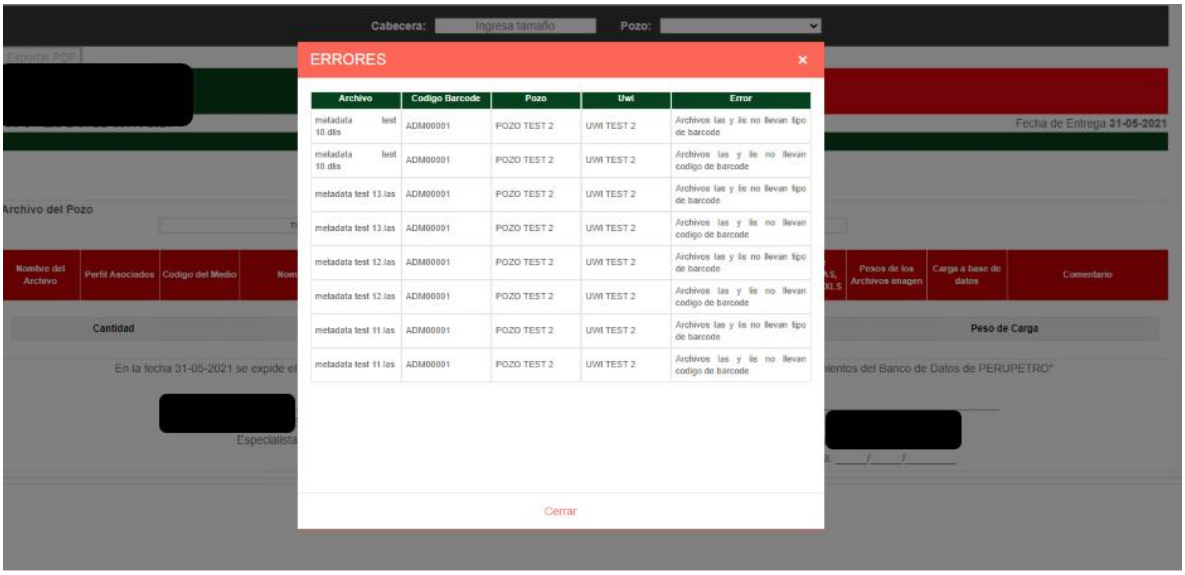



Figura 76: Frontend interface Alerta de errores detectados en Metadata de Pozos
Fuente Elaboración Propia (Año 2021)

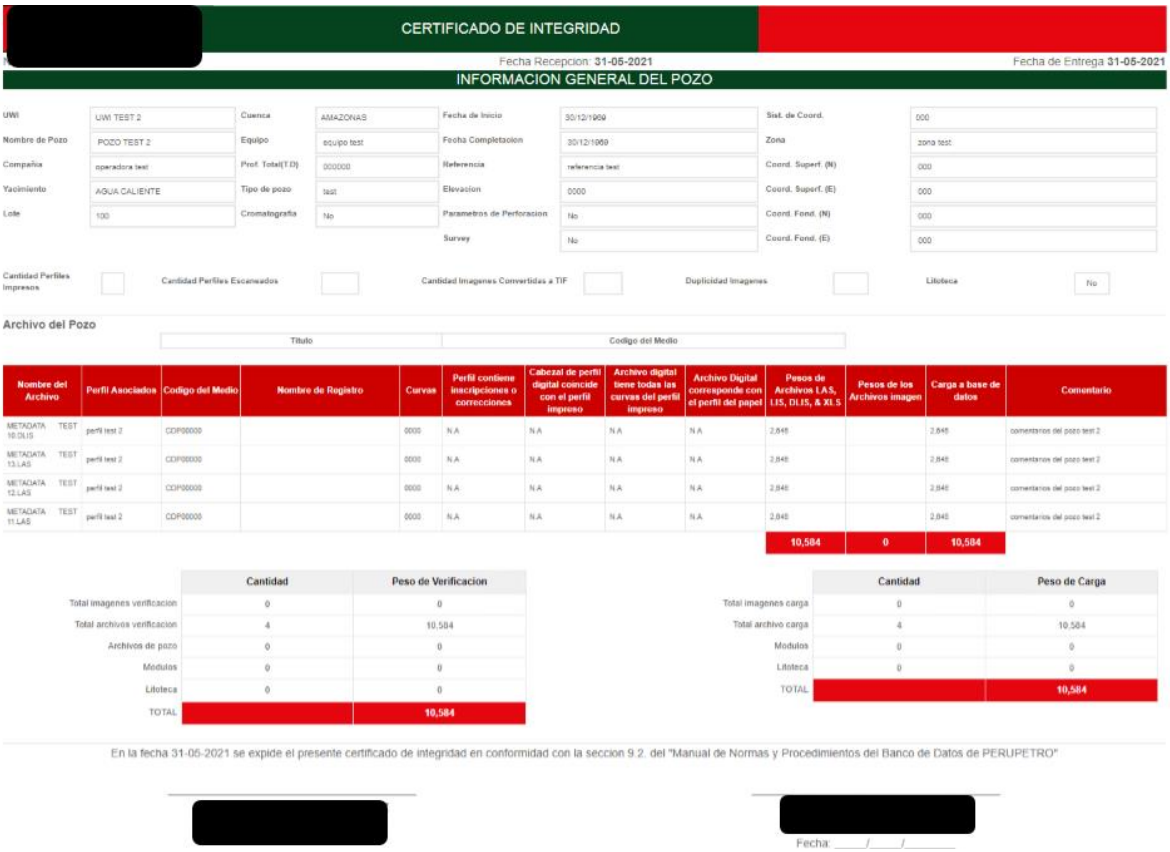



Figura 77: Frontend interface Certificado del Grupo de Pozos
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7.6. Módulo de reportes y dashboards

#### 3.2.5.7.6.1. Reporte de órdenes de trabajo en proceso




Figura 78: Frontend interface Reporte Ordenes de Trabajo en Proceso
Fuente Elaboración Propia (Año 2021)

#### 3.2.5.7.6.2. Reporte de tareas pendientes

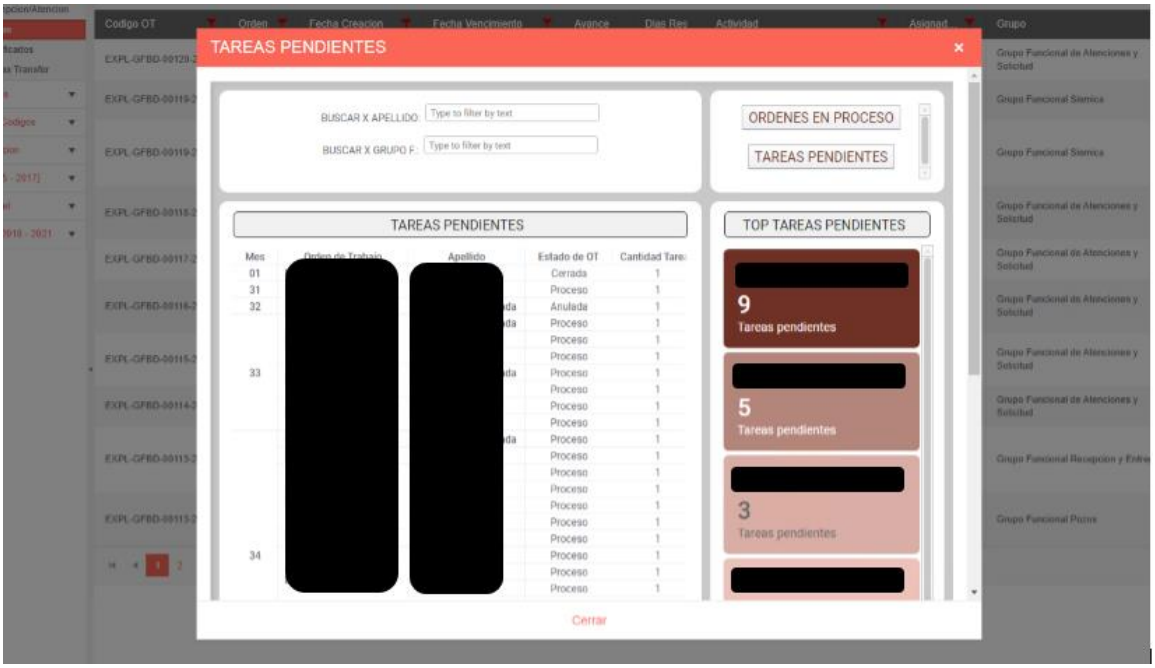



Figura 79: Frontend interface Reporte de Tareas Pendientes
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7.6.3. Dashboard gerencial

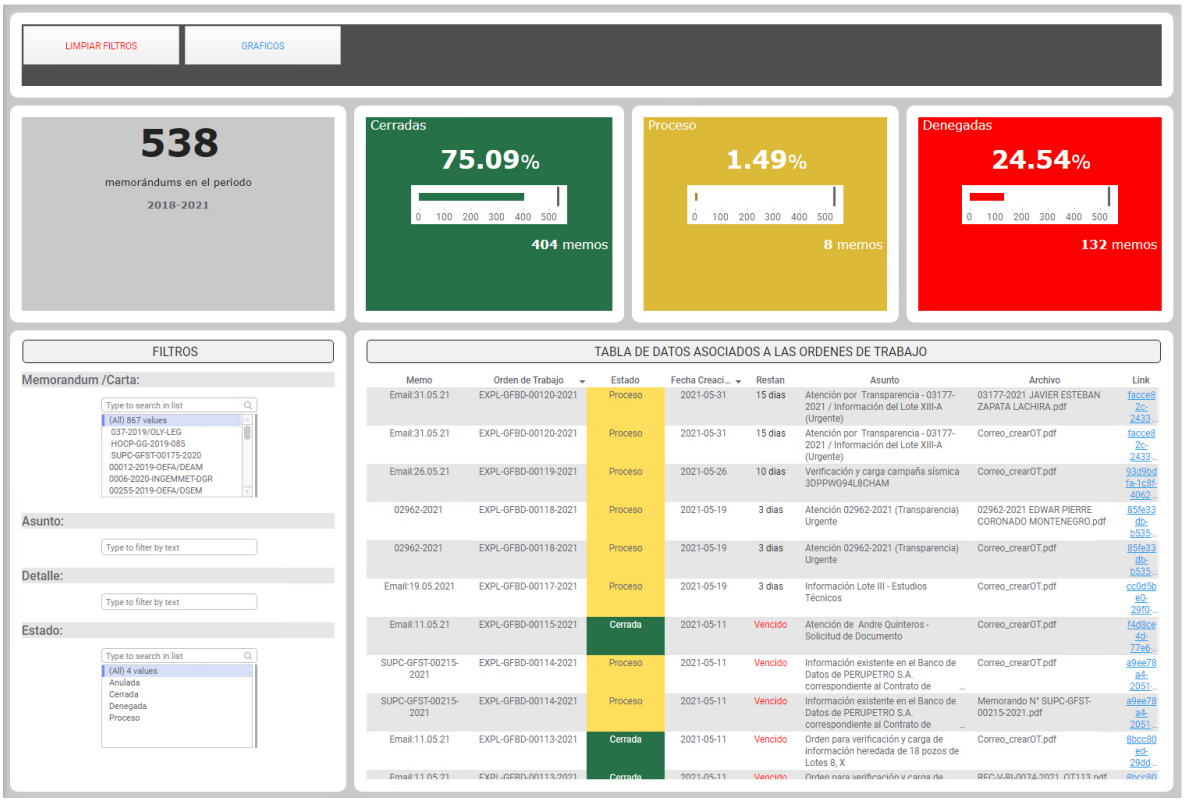



Figura 80: Frontend interface Dashboard Gerencial
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7.6.4. Dashboard ordenes de trabajo

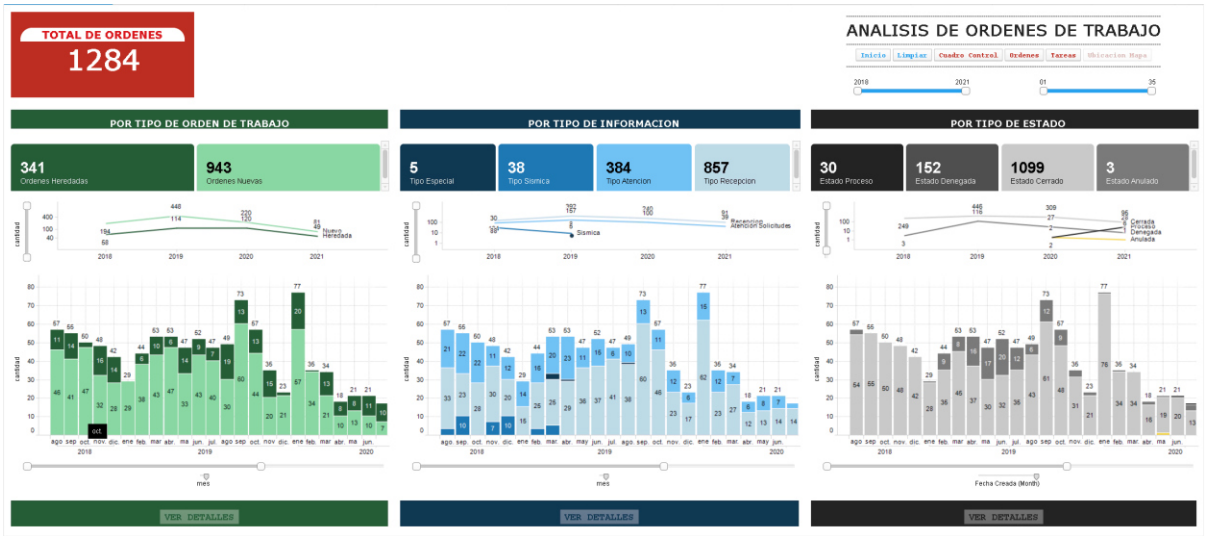



Figura 81: Frontend interface Dashboard Ordenes de Trabajo
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7.6.5. Dashboard tareas




Figura 82: Frontend interface Dashboard Tareas
Fuente Elaboración Propia (Año 2021)

### 3.2.5.7.6.6. Dashboard cuadro control

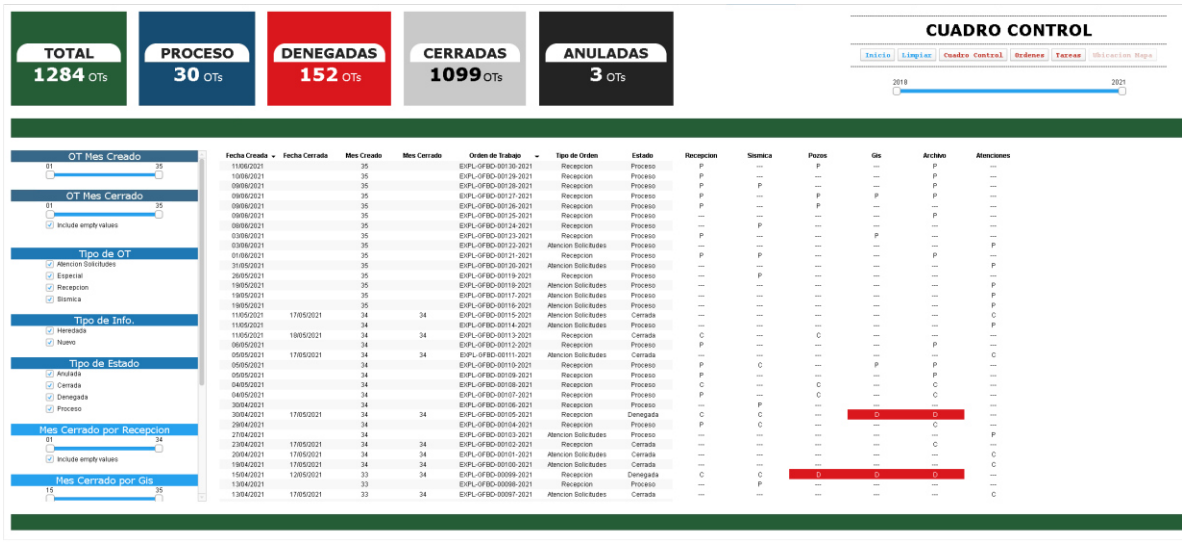



Figura 83: Frontend interface Dashboard Cuadro Control
Fuente Elaboración Propia (Año 2021)

### 3.2.5.8. Pruebas

Dentro del flujo Kanban, las pruebas se realizan dentro de la fase DOING, cada tarjeta Kanban debe contener la actividad **TEST**

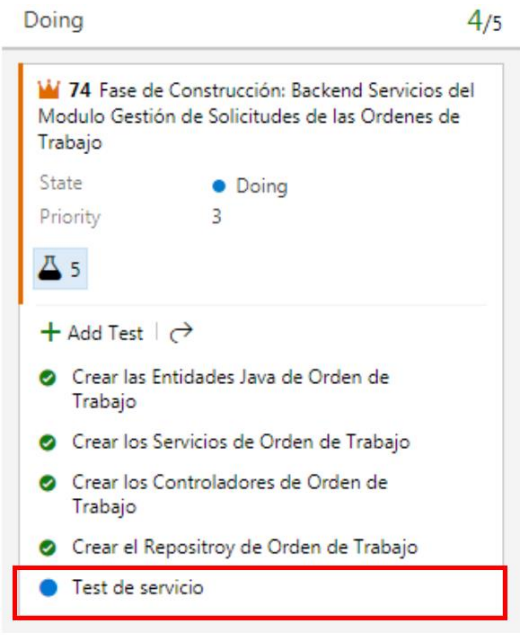



Figura 84: Tarjeta Kanban actividad test de servicios.
Fuente Elaboración Propia (Año 2021).

#### 3.2.5.8.1. Pruebas de humo

Estas pruebas son ejecutados por el equipo de desarrollo, se centran básicamente en:

- ✓ Configuración de la Máquina Virtual que albergara la aplicación web.
- ✓ Configuraciones de Base de Datos.
- ✓ Configuraciones Backend.
- ✓ Configuración de Servidor Web.
- ✓ Configuración del Framework Angular.
- ✓ Configuración de Tibco Spotfire.
- ✓ Pruebas de conectividad, vía telnet y/o ping a los diferentes servidores y workstations que se encuentren en el dominio del Active Directory.
- ✓ Funcionalidades Principales.

- ✓ Integridad de Datos.
- ✓ Protocolos de seguridad.

El objetivo es determinar si la aplicación está en condiciones para realizar el ciclo de pruebas.

#### 3.2.5.8.2. Pruebas de diseño

Las pruebas se realizaron a las interfaces de la aplicación, donde fueron evaluados por los usuarios in situ, el objetivo es tener la aceptación y conformidad de colores, legibilidad, continuidad, dimensiones y claridad.

#### 3.2.5.8.3. Pruebas de unitarias

Utilizando la herramienta de Postman validamos a nivel del backend, que los servicios REST están funcionando.

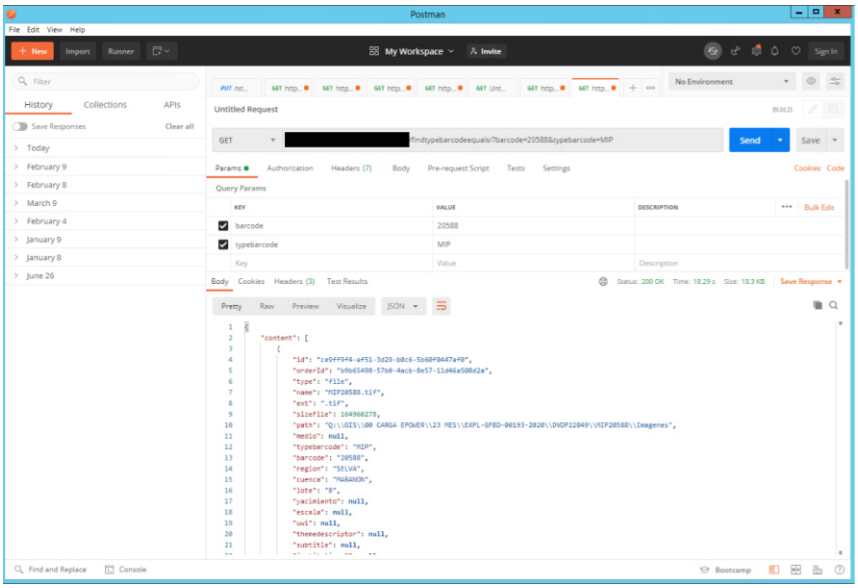



Figura 85: Postman para validar servicios REST.
Fuente Elaboración Propia (Año 2021).

### 3.2.5.9. Despliegue

#### 3.2.5.9.1. Despliegue frontend

Ejecutamos los siguientes comandos:

- Forever start forever.json

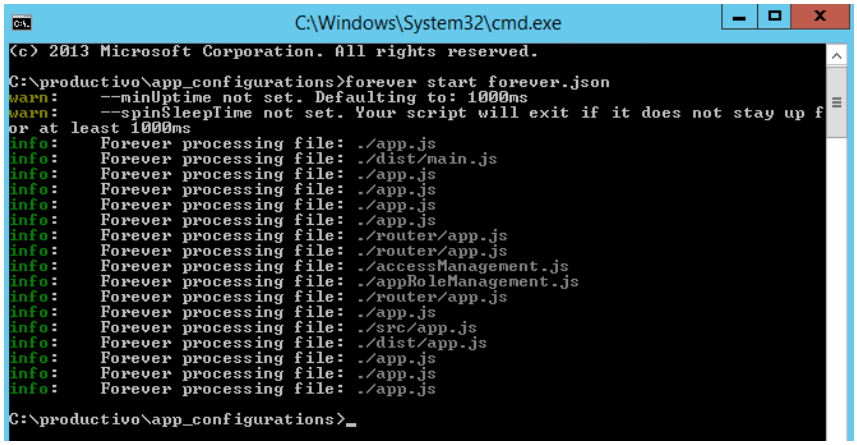



Figura 86: NodeJS Iniciamos todos los servicios y configuraciones.
Fuente Elaboración Propia (Año 2021).

- Ejecutamos dentro del proyecto npm run build

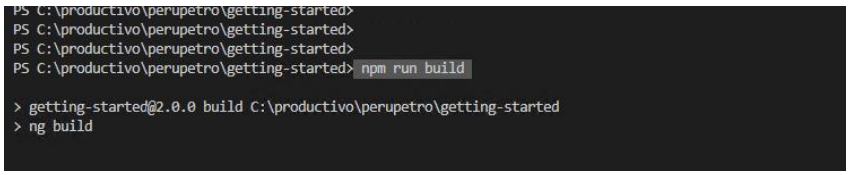



Figura 87: Terminal Visual Code generando carpeta DIST.
Fuente Elaboración Propia (Año 2021).

- Generando se la carpeta dist

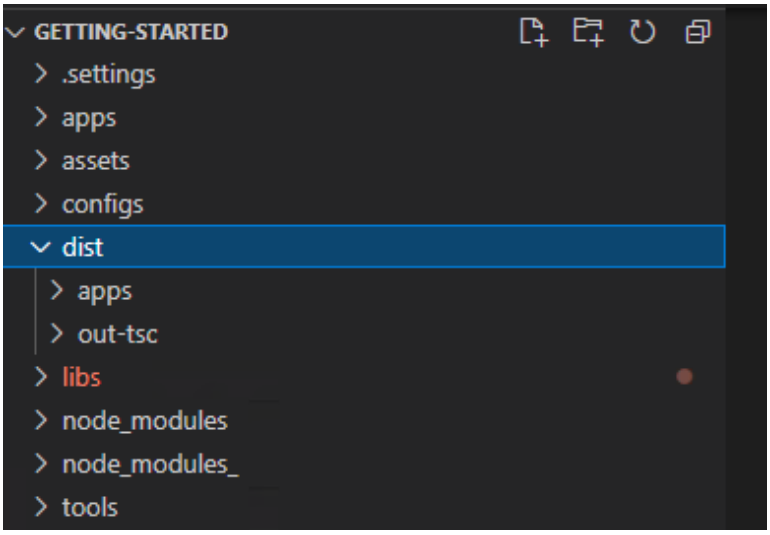



Figura 88: Estructura de desarrollo ubicando la carpeta DIST.
Fuente Elaboración Propia (Año 2021).

### 3.2.5.9.2. Despliegue backend

Realizamos los siguientes pasos:

- Generamos el archivo .war ejecutando Maven Install

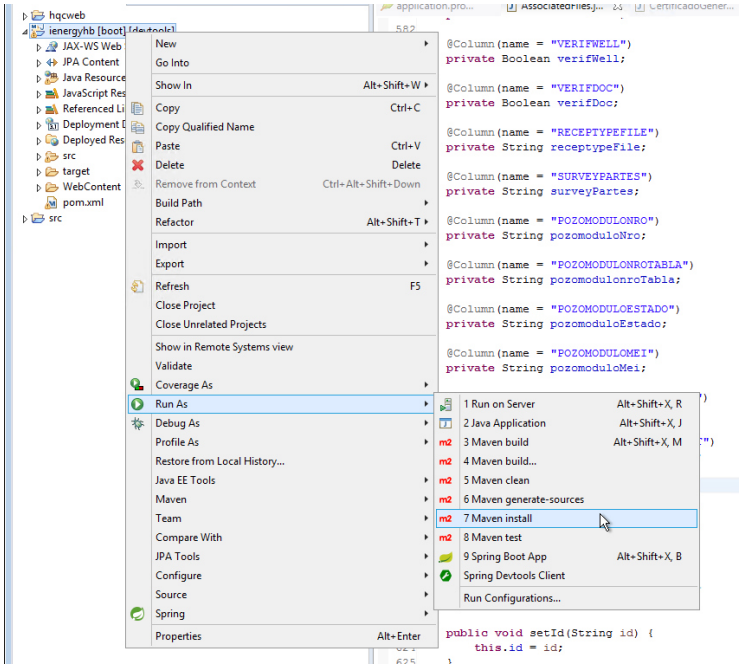



Figura 89: Ejecutando Maven Install.
Fuente Elaboración Propia (Año 2021).

- En el Gestor de aplicaciones web Tomcat procedemos a replegar el antig y desplegar el nuevo .war generado.

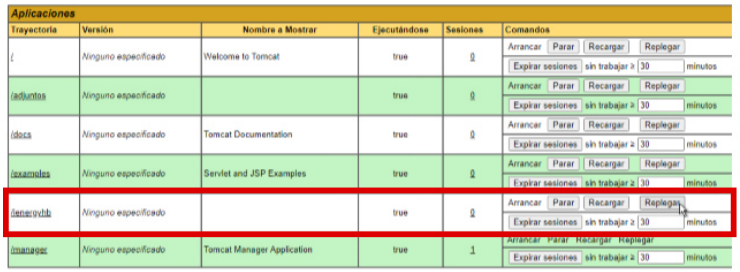



Figura 90: Apache tomcat Replegar .WAR.
Fuente Elaboración Propia (Año 2021).




Figura 91: Apache Tomcat Desplegar .WAR.
Fuente Elaboración Propia (Año 2021).

## 3.3. Evaluación

### 3.3.1. Evaluación económica

El desarrollo de la aplicación web se realizó dentro de los lineamientos del Proyecto de Administración del Banco de Datos, en donde a la empresa Halliburton del Perú S.R.L se le dio la buena pro asignándole un presupuesto aproximado de $ 6, 500, 000 (Seis millones quinientos mil con 00/100 Dólares Americanos). La administración del presupuesto cubre todos los gastos necesarios para el correcto funcionamiento del Banco de Datos, en donde se realizaron diversas compras de: Servidores, Workstation, UPS, Plotters, Impresoras, Licencias de Software, Licencia de Base de Datos, Licencia de Sistemas Operativos, Planilla de Trabajadores, etc.

Así mismo la empresa está expuesto a penalidades ante la indisponibilidad de los servicios del Banco de Datos, por ejemplo: $ 500 (Quinientos con 00/100 Dólares Americanos) por cada día de atraso en los informes, óptimo funcionamiento de software entre otros.

Bajo este contexto, se procedió actualizar las diferentes aplicaciones heredadas de otras gestiones del Banco de Datos, siendo la principal el Software de Gestión de Ordenes de Trabajo, con el objetivo de asegurar una óptima Administración del Banco de Datos.

El costo Aproximado del Proyecto:

| | 25/07/2018 | 25/08/2018 | 25/09/2018 | 25/10/2018 | 25/11/2018 | 25/12/2018 | Total |
|---|---|---|---|---|---|---|---|
| Analista | S/. 7 000 | S/. 7 000 | S/. 7 000 | S/. 7 000 | S/. 7 000 | S/. 7 000 | S/. 42 000 |
| Programador Frontend | | S/. 4 500 | S/. 4 500 | S/. 4 500 | S/. 4 500 | S/. 4 500 | S/. 22 500 |
| Programador Backend | | S/. 4 500 | S/. 4 500 | S/. 4 500 | S/. 4 500 | S/. 4 500 | S/. 22 500 |
| Gastos Operativos | S/. 4 000 | S/. 3 000 | | | S/. 4 000 | S/.4 000 | S/. 15 000 |
| Laptops | S/. 12 000 | | | | | | S/. 12 000 |
| Total | S/. 23 000 | S/. 19 000 | S/. 16 000 | S/. 16 000 | S/. 20 000 | S/. 20 000 | S/. 114 000 |

Tabla 25: Estimación de Costos
Fuente Elaboración Propia (Año 2021)

Detalles:

- El pago del recurso humano corresponde al pago de planilla de cada trabajador, se considera costo por que fueron designados a tiempo completo en el proyecto.
- Los gastos operativos incluyen pasaje aéreo ida y vuelta (Colombia - Perú), Hospedaje, Viáticos, Movilidad, entre otros.
- Se asignaron 4 Laptops para el desarrollo del proyecto, se trabajó mediante conexión VPN.

Beneficio Económico:

- El resultado fue exitoso, la aplicación no fue expuesto a ninguna penalidad.
- Se optimizaron tiempos de trabajo, así como se automatizo trabajos manuales, esto permitió que la empresa no realice pagos extras para el cumplimiento de objetivos.
- El software se implementó en conjunto con otras aplicaciones en el Servicio Geológico de Colombia en el 2019 lo que generó ingresos a la empresa, fue fundamental la referencia del funcionamiento y operación del software en el Banco de Datos.

### 3.3.2. Beneficio para la organización

- Incluir el software de Gestión de Órdenes de Trabajo del Banco de Datos dentro del catálogo de ventas de la empresa, el contar con un software probado y en funcionamiento otorga: Ventaja competitiva, punto de referencia y confianza. Para ser presentado y vendido a otros Bancos de Datos a nivel Mundial.
- Implementación del software de Gestión de Ordenes de Trabajo en Colombia para el Servicio Geológico de Colombia, bajo el nombre de Gestión de Transmital.

## Capitulo IV - Reflexión crítica de la experiencia

- El autor de este trabajo profesional participo en el proyecto asumiendo el rol de "Analista Programador", desde la etapa de inicio hasta la puesta en producción. El equipo de desarrollo estuvo conformado por 4 personas, de los cuales 2 eran colegas de Colombia y 1 de Brasil, uno de los colegas de Colombia asumió el rol de Líder del Proyecto y Analista Funcional.

  **Fue gratificante la experiencia de trabajar con un equipo internacional conocer parte de su cultura, tendencias tecnológicas, comparar procesos de negocio en organizaciones análogas al Banco de Datos.**

  **En anteriores experiencias solo asumía el rol de programador, realizar las funciones de Analista con el soporte del Líder del proyecto, extendió mi experiencia en dominar la empatía con el cliente, analizando sus necesidades y a documentar sus requerimientos. Si bien es cierto se designó enfocarnos al 100% en el Proyecto, en la práctica se terminaba colaborando en otras actividades críticas dentro del Banco de Datos, completando las tareas del Proyecto en horas extras que fueron reconocidos por la empresa.**

- Las coordinaciones de trabajo se realizaban mediante Microsoft Teams usando las herramientas de video llamada y planner. El desarrollo se inició en la plataforma de Open Earth Community en una máquina virtual compartida, esto era temporal, hasta configurar el servidor de ambiente de pruebas en el Banco de Datos, en donde posteriormente se migro los desarrollos elaborados.

  **Pertenecer a la comunidad de Open Earth fue conectarse a diversos proyectos privados y públicos realizados en el mundo por Halliburton, proyectos que podrían servir en futuros desarrollos adaptados al Banco de Datos.**

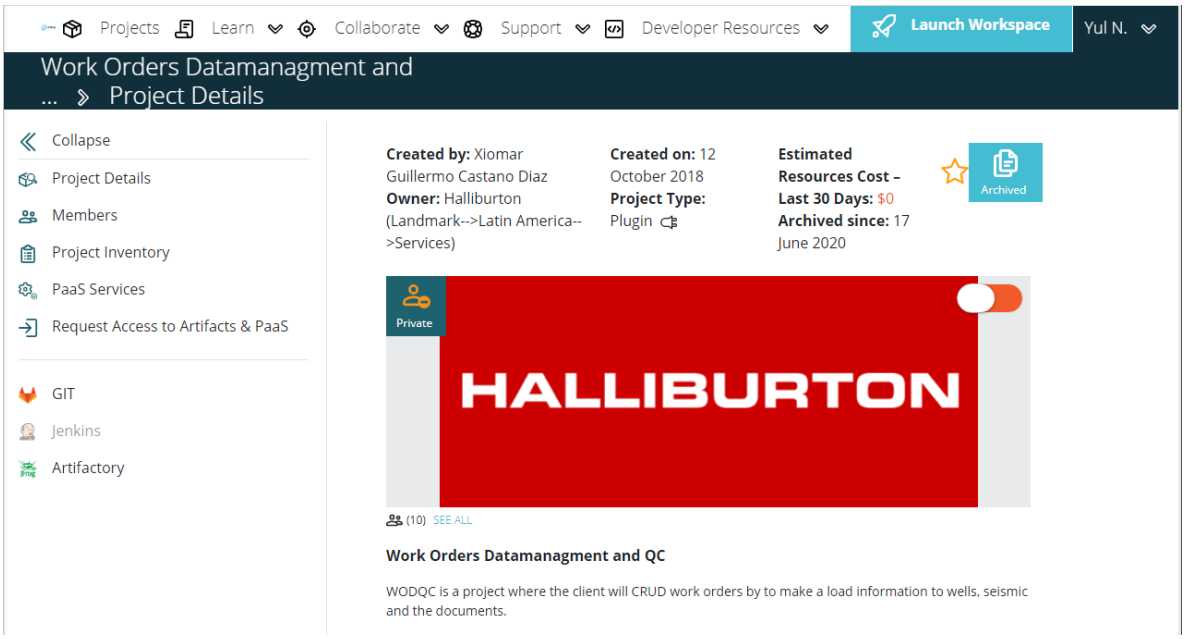



Figura 92: Publicación del Proyecto en Open Earth
Fuente Open Earth Community (Año 2018)

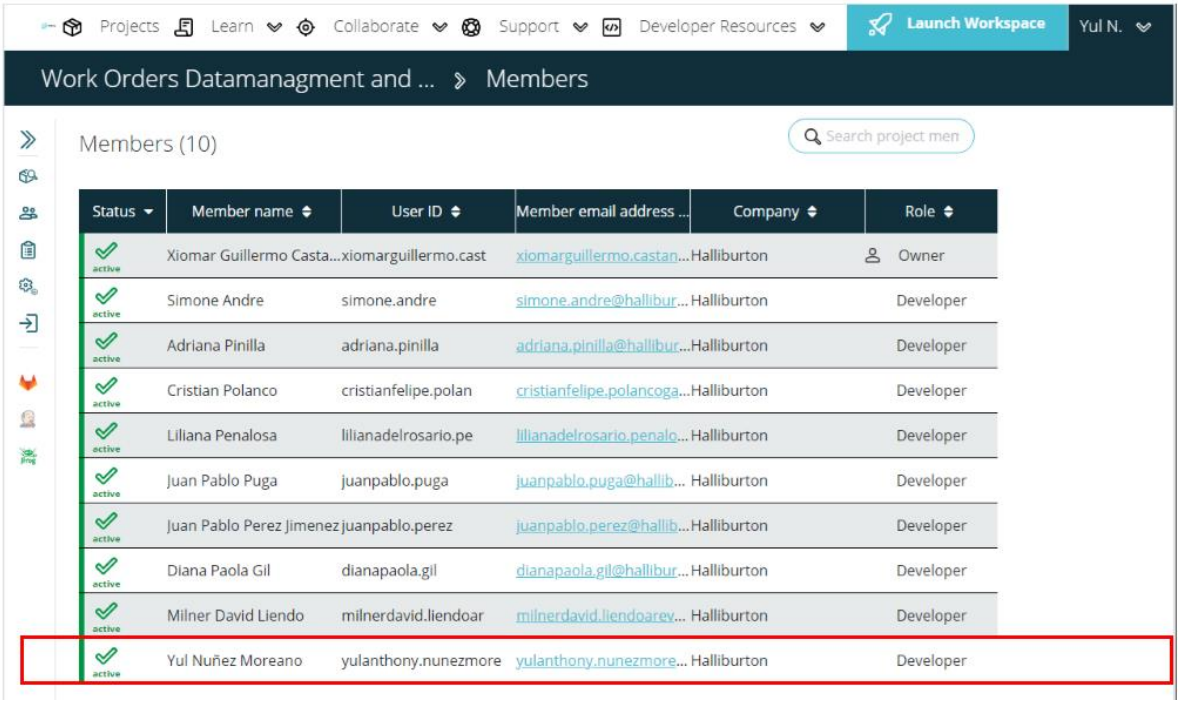



Figura 93: Lista de Integrantes del Proyecto
Fuente Open Earth Community (Año 2018)

**Usar Microsoft Teams como herramienta colaborativa fue de gran soporte para un equipo separado geográficamente, el proceso de adaptación fue progresivo logramos implementar un tablero kanban para el seguimiento de tareas.**

- Las principales actividades que he realizado durante la fase del proyecto fueron:
  - Canal de comunicación entre el equipo de desarrollo con el cliente y usuarios del Banco de Datos.
  - Análisis de documentos históricos y de los términos de referencia del presente contrato.
  - Levantamiento de información mediante entrevistas, documentación histórica y de los términos de referencia.
  - Desarrollo del FrontEnd, siendo los principales: Listar Ordenes de Trabajo, Listar Certificados, Consulta de Código Barcode, Registro de Dumps Consulta de Medio Barcode, Registro de Metadata de Archivo Técnico, Registro de Metadata de Pozos, Seguimiento Documental, Mantenimiento de Usuario, Mantenimiento de Grupos Funcionales, Mantenimiento Tipo de Gerencia y Mantenimiento de Tareas
  - Desarrollo del Backend, siendo los principales:
    - ✓ Servicios de consultar, búsqueda, registrar, actualizar y eliminar Certificados de Calidad.
    - ✓ Servicios de consulta y brusquedades Códigos Barcodes.
    - ✓ Servicios de consulta y búsqueda de Medios Digitales.
    - ✓ Servicios de consultar, registrar, actualizar y eliminar Tipo de Gerencia.
    - ✓ Servicios de consultar, registrar, actualizar y eliminar Mantenimiento de Tipo de Gerencia.
    - ✓ Servicios de consultar, registrar, actualizar y eliminar Mantenimiento de Tareas.
    - ✓ Integración mediante los servicios de consulta a compañías operadoras, pozos, región, cuenca, lote y yacimiento registrados por otro software del Banco de Datos.

- Desarrollo de los Módulos:
    - ✓ Importación de datos desde Plantillas de Excel, para los Grupos Funcionales de Pozos y Sísmica, el desarrollado de este módulo se realizó en Java.
    - ✓ Certificados de Calidad, para los Grupos Funcionales de Recepción, GIS, Archivo Técnico, Sísmica y Pozos. EL desarrollo se realizó usando el Framework de Angular.
    - ✓ Reporte y Dashboards, elaborado en Tibco Spotfire.
- Migración de datos de MySQL a PostgreSQL, de acuerdo a los lineamientos de la empresa.

**La directiva de la empresa fue que el desarrollo se tenía que realizar en el Framework de Angular usando como biblioteca de componentes a Kendo Telerik, como estándar de tecnología de desarrollo. El equipo pasó por asesorías, pero considero que se debió realizar una inducción completa y asegurar que los miembros del proyecto están capacitados.**

- Las principales dificultades que se presentaron durante el proyecto está relacionado:
    - Al alcance, este fue variando en las reuniones que se realizaban con el cliente quedando como acuerdos. El alcance inicial de la aplicación era la Gestión de Solicitudes, pero progresivamente se fue incorporando la Carga de Metadata, Gestión de Metadata, Cartas de Transferencias y Certificados de Calidad. Estos cambios afectaban el cronograma y se estaban ejecutando sin terminar lo requerido inicialmente.
    - La comunicación, si bien es cierto que esto se realizaba mediante Microsoft Teams, al inicio no se tenía la fluidez por que el equipo se encontraba en proceso de adaptación, en ese tiempo se generó trabajo duplicado, interfaces fuera de contexto entre otros.
    - No se elaboró un Plan de Migración de Datos, se subestimo al considerarlo un proceso fácil, no se tomó en cuenta realizar: Copias de seguridad, asegurar que los datos cumplen con los requisitos del nuevo sistema, los

tiempos de inactividad que podrían ocasionar penalidades, corrupción de datos en los formatos, duplicidad de datos, disponibilidad de datos en el nuevo sistema y perdida de datos.

- Las pruebas fueron llevadas al final del proyecto y realizadas por el equipo del proyecto y los usuarios principales de cada grupo funcional, estas se realizaron mediante un check list en donde no hubo observaciones, puesto en producción la aplicación después de 2 meses aproximadamente presento incidencias que fueron resueltas posteriormente, esto sucedió por que no se realizaron pruebas integrales y de flujo completo abarcando los diferentes escenarios.

# Capítulo V - Conclusiones y recomendaciones

## 5.1. Conclusiones

- La aplicación web Gestión de Ordenes de Trabajo ha permitido resolver los problemas reportados a inicios de la Administración del Banco de Datos:
  - ✓ Resolvió la estandarización de tecnología de desarrollo a nivel del FrontEnd utilizando el Framework de Angular y la Biblioteca de Componentes de Kendo – Telerik, dándole un sentido de continuidad a las diversas interfaces utilizadas por los usuarios.
  - ✓ Resolvió los problemas de escalabilidad y modularidad de la aplicación, gracias a la estandarización de tecnologías, esto permitió resolver de forma rápida las personalizaciones, cambios de funcionalidad, cambios de diseño y nuevos módulos requeridos por el cliente.
  - ✓ Resolvió los problemas de integración de datos gestionados por otras aplicaciones, registrando datos de completitud mediante combo box y autocompletado reduciendo el riesgo por error de digitación de parte del usuario, aumentando la seguridad e integridad de datos y mejorando la percepción de los usuarios.
  - ✓ Se implementaron alertas por email cuando una Orden de Trabajo se aproxima a su fecha de vencimiento, ayudaron a los Líderes a priorizar sus actividades y a Gerencia a monitorear y dar seguimiento del cierre de cada Orden de Trabajo para así evitar atrasos por ende penalidades económicas.
  - ✓ Los reportes usan colores del tipo semáforo para determinar los estados de proceso, denegado y cerrado, mejoran la visibilidad y dan una visión rápida de la administración, lo que genero felicitaciones verbales de parte del cliente.

- Se realizó el análisis de los requerimientos funcionales y no funcionales en base a las entrevistas realizadas, a los términos de referencia del contrato y a la documentación histórica. En base a esto se generó el Backlog del Proyecto.
- Se diseñó el modelo de datos lógico, la arquitectura de la aplicación y las interfaces web de acuerdo a los requerimientos del cliente y que cumplan con sus necesidades funcionales.
- Se implementaron con éxito los módulos del Sistema:
    - ✓ Gestión de Solicitudes de Ordenes de Trabajo.
    - ✓ Gestión de Tareas de Ordenes de Trabajo.
    - ✓ Gestión de Metadata.
    - ✓ Certificados de Calidad.
    - ✓ Reportes.
    - ✓ Dashboards
- La aplicación continúa en funcionamiento en el Banco de Datos de la Empresa de Hidrocarburos, desde el 2018 hasta la actualidad, se realizó una segunda implementación en el Servicio Geológico de Colombia (Colombia), y esta también se encuentra en funcionamiento desde el 2019 hasta la actualidad. Para que la empresa Halliburton gane esta licitación fue determinante la referencia de contar con un software en funcionamiento en Perú. Ahora que el software cuenta con una segunda referencia abre las posibilidades de futuras implementaciones en otros Bancos de Datos a nivel mundial, lo que generara ingresos a la empresa Halliburton.

## 5.2. Recomendaciones

- Incluir e Implementar Metodología Agile Scrum para la gestión y desarrollo de proyectos de software. Esto ayudara al seguimiento de las actividades, eliminar interrupciones, visualizar el flujo de trabajo y validar con el cliente mediante entregables parciales.

- Se debe mantener un ambiente de prueba que sea réplica del ambiente productivo, manteniendo los mismos componentes, interfaces, controladores, versión de base de datos, etc. Una herramienta tecnológica recomendable seria el uso de DOCKER, teniendo como principales ventajas la portabilidad de la aplicación, generación de copias e imágenes, autonomía, optimiza tiempos de mantenimiento y el despliegue a producción es rápido habilitando/deshabilitando cualquier imagen en el contenedor.

- Se debe migrar la aplicación web, actualmente se encuentra en un servidor Windows 2012 – R2 se recomienda usar un servidor Linux, siendo los motivos: la compatibilidad con las otras aplicaciones del Banco de Datos, mayor estabilidad, mayor rendimiento y mayor seguridad.

- Se recomienda que ante una directiva de la empresa en el uso de una nueva tecnología se brinde una capacitación completa, para reducir tiempos en la curva de aprendizaje y en la producción de software.

- Se recomienda realizar la documentación del software en todas las fases del proyecto de desarrollo y no realizarlo al final o meses después, ya que puede hacerse complejo y se puede omitir involuntariamente procedimientos importantes

## 5.3. Fuentes de información

### 5.3.1. Documentos administrativos

- Términos de Referencia de Banco de Datos en el Mundo.
- Manual Normas y Procedimientos del Servicio Geológico y del Banco de Datos.

### 5.3.2. Bibliografía académica


Arrieta Ticliahuanca, R., & Castillo Maticorena, A. C. (2019). *Sistema web utilizando Framework Angular para el proceso de mantenimiento de equipos de TI en la Empresa MRJ Servicios TEC.* Lima: Universidad Cesar Vallejo. Obtenido de https://hdl.handle.net/20.500.12692/53437

Castellano Lendínez, L. (2019). KANBAN. METODOLOGÍA PARA AUMENTAR LA EFICIENCIA DE LOS PROCESOS. *3C Tecnologia, 8*, 30-40. doi:10.17993/3ctecno/2019.v8n1e29/30-41

DB-Engines. (14 de Junio de 2021). Ranking de los sistemas de gestión de bases de datos más populares en todo el mundo, a junio de 2021. Statista. Recuperado el 14 de Junio de 2021, de https://www.statista.com/estadísticas/809750/worldwide-popularity-ranking-database-management-systems/

GitLab. (03 de Mayo de 2021). *Desglose de las metodologías de desarrollo de software practicadas en todo el mundo en 2021*. Statista. Recuperado el 14 de Junio de 2021, de https://www.statista.com/statistics/1233917/software-development-methodologies-practiced/

Gutiérrez Zúñiga, J., & Chanca de la Cruz, M. M. (2011). *Análisis, diseño e implementación de un sistema de información para el control de becarios y exbecarios de una asociación educativa.* Lima: Pontificia Universidad Católica del Perú. Obtenido de http://hdl.handle.net/20.500.12404/329

Molina Ríos, J., Zea Ordóñez, M., Contento Segarra, M., & García Zerda, F. (2018). COMPARACIÓN DE METODOLOGÍAS EN APLICACIONES WEB. *3C Tecnología, 7*, 4. doi:10.17993/3ctecno.2018.v7n1e25.1-19

Morales Timbela, J. E., & Moya Portilla, S. P. (2015). *Analizar, diseñar y construir un sistema informático híbrido, para el manejo del horario de ingresos y salida del personal de la empresa "Cityventa", mediante identificación de rostros y administración remota de información.* Quito - Ecuador: UNIVERSIDAD POLITÉCNICA SALESIANA. Obtenido de https://dspace.ups.edu.ec/handle/123456789/10115

Muyón, C., & Montaluisa , F. (2020). Métodos de seguridad de la información para proteger la comunicación y los datos de servicios web REST en peticiones

HTTP utilizando JSON Web Token y Keycloak Red Hat Single Sign On. *Associação Ibérica de Sistemas e Tecnologias de Informacao*(E29), 198-213. Obtenido de https://www.proquest.com/scholarly-journals/métodos-de-seguridad-la-información-para-proteger/docview/2394538091/se-2?accountid=12268

Rea-Peñafiel, X. M., Mancero-Menoscal, T. B., Rosero-Rea, D. C., & Imbaquingo-Esparza, D. E. (2020). Web Services REST: una revolución en la forma de acceso a datos. *Revista Ibérica de Sistemas e Tecnologias de Informação*(E32), 41-49. Obtenido de https://www.proquest.com/scholarly-journals/web-services-rest-una-revolución-en-la-forma-de/docview/2452330041/se-2?accountid=12268

Stack Overflow. (15 de Febrero de 2020). Los frameworks web más utilizados entre los desarrolladores de todo el mundo, a principios de 2020. Statista. Recuperado el 14 de Junio de 2021, de https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/

Tibco Spotfire. (2021). *tibco.com*. Obtenido de https://www.tibco.com/es/products/tibco-spotfire

Tito Chulde, P. (2020). *SISTEMAS COMPUTACIONALES;GESTIÓN INFORMÁTICA;GESTIÓN Y CONTROL DE PROCESOS;MICROEMPRESA;SPRING BOOT*. Ibarra - Ecuador: UNIVERSIDAD TECNICA DEL NORTE. Obtenido de http://repositorio.utn.edu.ec/handle/123456789/10366

Vargas Rioja, C. A. (2020). *Implementación de un Sistema de Información Hospitalario (HIS) interoperable basado en HL7 para un Centro Médico de categoría II-1 o superior*. Lima: Pontificia Universidad Católica del Perú. Obtenido de http://creativecommons.org/licenses/by-nc/2.5/pe/

Yépez Llerena, E., & Armijos Guillen, K. (2020). *aplicación de la metodología kanban en el desarrollo del software para generación, validación y actualización de reactivos, integrado al sistema informático de control académico unach.* Riobamba - Ecuador: Universidad Nacional de Chimborazo. Obtenido de http://dspace.unach.edu.ec/handle/51000/6457

## 5.4. Glosario

| Término | Definición |
|---|---|
| Administración de Banco de Datos | Consiste en la prestación de servicios de administración de datos e información técnica de sísmica, pozos, documentos técnicos, información geográfica, geoquímica, gravimetría y magnetometría del sector hidrocarburo. |
| Catalogación | Elemento físico en el cual está contenida la información, ya sea en papel o en versión digital, que entregan las Compañías Operadoras, relacionada con las actividades de exploración y / o explotación de hidrocarburos que desarrollan. Pueden ser Medios Digitales o Medios Impresos. |
| Certificado de Calidad | Son generados a partir de la información que es cargada a la base de datos y debe mostrar los volúmenes cargados por tipo de orden, pozos, tipo de mapa, campaña sísmica y códigos barcode, divididos en peso de verificación y certificación. |
| Dashboard | Es la representación gráfica de los datos en un panel, donde se visualiza información importante, siendo soporte clave en la toma de decisiones. |
| Gestión de Solicitudes de O.T. | Consiste en el registro de los memorándums, seguimiento de sus estados de avance, control de los medios digitales y control de los códigos barcodes. Se ejecución genera una Orden de Trabajo |
| GIS | Abreviado de Sistema de Información Geográfica. |
| Información Heredada | Información Técnica de Exploración & Producción original y heredada de la anterior Administración del Banco de Datos, antes del inicio del Servicio prestado por HALLIBURTON, el cual ha sido asignada con un Código de Barras del Banco de Datos, puede ser información impresa o información en medio digital óptico. Parte de la información antigua ha sido formalmente entregada por la empresa y resguardada en un repositorio de HALLIBURTON. La otra parte de la Información Digital y toda la información impresa se mantienen en el Archivo Técnico. |

| | |
|---|---|
| Información Nueva | Información Técnica de Exploración & Producción entregada por parte de la EMPRESA a HALLIBURTON dentro del período del Servicio de Administración del Banco de Datos, iniciado el 2018. Información aún no asignada con un Código de Barras. |
| Medios Digitales | Medios de almacenamiento de datos usados por el Banco de Datos de EMPRESA, entre los que se encuentran los Medios Digitales Ópticos como los CDs y DVDs y los Medios Digitales Magnéticos como los Discos Duros, Memorias USB y Cintas. |
| Metadata | Hace referencia a un grupo de datos que describen el contenido informativo de un archivo. |
| Metadata Básica | Hace referencia a los atributos básicos de los datos como nombre y peso del archivo |
| Orden de Trabajo (OT) | Solicitud de servicio emitida por LA EMPRESA y atendida por HALLIBURTON como parte de la administración del Banco de Datos de Exploración y Explotación. También referida al número (secuencial) de ticket en el aplicativo, que utiliza el Administrador del Banco de Datos de EMPRESA para realizar una petición de servicio a HALLIBURTON y que facilita su seguimiento. |
| O.T. | Abreviado de Orden de Trabajo |
| Peso Certificado | Está asociado a todos aquellos archivos generados en base a los archivos verificados. Estos archivos son completitud de datos, mejora de calidad de imágenes, actualización de mapas, etc. |
| Peso Verificado | Está asociado a todos los archivos digitales adjuntados a la orden de trabajo o archivos históricos en los repositorios. Para considerar que es un archivo verificado, el archivo original no debe ser modificado y debe ser revisado. |

| | |
|---|---|
| Petrobank | Solución de banco de datos que administra toda la información de Metadata de los datos de Sísmica, Pozos, Documentos técnicos, genera archivos *shapefiles* para la base de datos GIS. Está integrado a aplicaciones de interpretación y procesamiento desde portal web a través de conectores en DSIS – Decisión Integration Server. Proporciona una catalogación en base de datos única y búsqueda facilitada por en el portal web. |
| | |
| Sistema de Gestión de Órdenes de Trabajo del Banco De Datos | Aplicativo que se usa para generar, asignar y controlar las Órdenes de Trabajo del Banco de Datos. La aplicación es clave para la comunicación entre EMPRESA y HALLIBURTON. En HALLIBURTON es usado por los Grupos Funcionales de Recepción y Entrega, Grupo de Sísmica, Grupo de Pozos, Grupo de GIS, Grupo de Archivo Técnico y Grupo de Atenciones. |
| T.D.R. | Abreviado de Términos de Referencia del Servicio de Administración del Banco de Datos. |
| Verificación Física | Revisión comparativa de la cantidad, forma y descripción en los listados de Entrega de la Información Nueva para admitirla como completa y determinar si corresponde o no a una verificación técnica. En los casos de la Información Antigua no se recibirán Listados, de forma que se revisará únicamente la correspondencia entre la versión impresa y la versión digital original de los documentos. |

## Anexos

### Anexo A. Tabla de los principales errores tecnológicos de los softwares heredados.

| Error | Descripción | Funcionalidad |
|---|---|---|
| Secuenciador de Orden de Trabajo | El secuenciador de órdenes de trabajo tiene un comportamiento irregular, en ocasiones no continua la numeración teniendo saltos aleatorios. Esto genera retrabajo en la elaboración de una O.T, atraso en la distribución de tareas a los grupos funcionales y problemas con auditoria interna. | Crear Orden de Trabajo |
| Vista Previa | La aplicación tiene la opción de vista previa antes de crear una orden de trabajo, esta acción incrementa el secuenciador por cada clic realizado en esta opción. | Crear Orden de Trabajo |
| Notificación vía e-mail | La notificación de correos se realizaba desde un correo Gmail, el cliente solicito usar el correo corporativo de la Empresa quien no tenía una configuración de registro M.X. La opción del plugin smtp de drupal no permitía adecuarse a la configuración de correo. | Crear Orden de Trabajo |
| Filtros de Búsqueda Orden de Trabajo | EL sistema no cuenta con filtros de búsqueda general, filtro por columnas y combinación de filtros. | Listado de Orden de Trabajo |
| Registro Manual | Se debe Agregar manualmente el id, orden de trabajo, fecha orden y fecha recepción. Para que se pueda listar las ordenes de trabajo. | Certificado Calidad |
| Exportar | EL sistema no cuenta con la opción de exportar a Excel el listado de Órdenes de Trabajo, Tareas y Reportes. | Orden de Trabajo y Tareas |
| Eliminar Orden de Trabajo | Cuando se elimina una orden de trabajo el contador mantiene su valor, cuando debería ser también eliminado. | Orden de Trabajo |

| | | |
|---|---|---|
| Filtros de Búsqueda Tareas | EL sistema no cuenta con filtros de búsqueda general, filtro por columnas y combinación de filtros. | Listado de Tareas |
| Archivos Asociados | Cuando se crea una orden de trabajo con más de 2 archivos asociados el sistema realiza la creación de varias órdenes de trabajo en proporción a la cantidad de archivos subidos. | Orden de Trabajo |
| Dashboards | El sistema no permite integrarse a la solución de Business Intelligence como Tibco Spotfire | Orden de Trabajo |
| Base de Datos | Compleja estructura, cada campo de texto del formulario de orden de trabajo representaba una tabla de datos en la Base de Datos así como también se duplicaba la tabla adicionando el prefijo “rev”. | Orden de Trabajo |
| Integración de Datos | Se debe correr un script para transferir datos de una base de datos a otra. | Certificados de Calidad y SCC |
| Mantenimiento | Es una aplicación de escritorio, requiere configuraciones previas que deben aplicarse en las 28 Workstation, cada vez que se realiza un cambio o personalización se debe realizar en cada ordenador. | Certificados de Calidad y SCC |
| Rendimiento | Lentitud en el procesamiento de actualización de una O.T. y en el Listado de Ordenes de Trabajo. | Orden de Trabajo |

## Anexo B. Código fuente del backend

- ✓ Entidades

**Ordenes de Trabajo**

**Entidad** | Orders.java

```
package com.halliburton.qc.entities;
import javax.persistence.*;
import org.hibernate.annotations.GenericGenerator;
import java.util.*;
import com.fasterxml.jackson.annotation.JsonIgnore;
import com.halliburton.qc.util.Utilities;
```

```
@Id
@Column(name = "ID")
private String id;
@Column(name = "CODE", insertable=false, updatable=false)
private Integer code;
@Column(name = "ORIGIN")
private String origin;
@Temporal(TemporalType.TIMESTAMP)
@Column(name = "DATECREATE", insertable=false,
updatable=false)
private Date datecreate;
@Column(name = "REGISTERBY")
private String registerby;
@Column(name = "REQUESTEDBY")
private String requestedby;
@Column(name = "MEMO")
private String memo;
@Column(name = "SUBJECT")
private String subject
@Column(name = "OBSERVATION")
private String observation;
@Column(name = "STATE")
private String state;
@Column(name = "SUBSTATE")
private String substate;
@Column(name = "TYPEOT")
private String typeot;
@Column(name = "MANAGEMENT")
private String management;
@Column(name = "COMMENTS")
private String comments;
@Column(name = "COMMENTS_TASK")
private String commentsTask;
@Temporal(TemporalType.TIMESTAMP)
@Column(name = "DATEDUE")
private Date datedue;
@Temporal(TemporalType.TIMESTAMP)
@Column(name = "DATE_UPDATE", insertable=false,
updatable=false)
private Date dateUpdate;
@Column(name = "SIGNEDBY")
private String signedby;
@Column(name = "USERUPDATE")
private String userupdate;
@Column(name = "EDITCATALOGO")
private Boolean editCatalogo;
@Column(name = "EDITDOCUMENT")
private Boolean editDocument;
@Column(name = "EDITSEISMIC")
private Boolean editSeismic;
@Column(name = "EDITWELL")
private Boolean editWell;
@Column(name = "EDITGIS")
private Boolean editGis;
@Column(name = "TYPEOTNEW")
private String typeotnew;
@Column(name = "DETALLE")
private String detalle;
@Column(name = "CODE_ORDER_TRABAJO")
private String codeorderTrabajo;
```

```
@Column(name = "STATETASKRECEPTION")
private String statetaskReception;
@Column(name = "STATETASKWELL")
private String statetaskWell;
@Column(name = "STATETASKARCHIVE")
private String statetaskArchive;
@Column(name = "STATETASKGIS")
private String statetaskGis;
@Column(name = "STATETASKSEISMIC")
private String statetaskSeismic;
@Column(name = "STATETASKATTENTION")
private String statetaskAttention;
@Column(name = "MESOT")
private String mesOt;
@Column(name = "MESCERRADOOT")
private String mescerradoOt;
@Column(name = "EDITOT")
private Boolean editOt;
@Column(name = "MESCERRADORECEPTION")
private String mescerradoReception;
@Column(name = "APPROVEPP")
private String approvePp;
@Column(name = "GRUPOSPP")
private String gruposPp;
@Column(name = "DATEAPPROVEPP")
private Date dateapprovePp;
@Column(name = "DATECERRADOGIS")
private Date datecerradoGis;
@Column(name = "DATECERRADOWELL")
private Date datecerradoWell;
@Column(name = "DATECERRADOSEISMIC")
private Date datecerradoSeismic;
@Column(name = "DATECERRADOARCHIVE")
private Date datecerradoArchive;
@Column(name = "DATECERRADOOT")
private Date datecerradoOt;
@Column(name = "MESCERRADOGIS")
private String mescerradoGis;
@Column(name = "MESCERRADOWELL")
private String mescerradoWell;
@Column(name = "MESCERRADOSEISMIC")
private String mescerradoSeismic;
@Column(name = "MESCERRADOARCHIVE")
private String mescerradoArchive;
@Column(name = "DATECERRADORECEPTION")
private Date datecerradoReception;
@Column(name = "DATECERRADOATTENTION")
private Date datecerradoAttention;
@Column(name = "MESCERRADOATTENTION")
private String mescerradoAttention;
@OneToMany(mappedBy = "orders", cascade = CascadeType.ALL)
@JsonIgnore
private Set<Tasks> tasks = new HashSet<Tasks>();
@OneToMany(mappedBy = "orders", cascade = CascadeType.ALL)
@JsonIgnore
private Set<OrdersGroup> ordersGroups = new
HashSet<OrdersGroup>();
```

```
}
```

| Tareas | |
|---|---|
| **Entidad** | Task.java |

```
package com.halliburton.qc.entities;
import javax.persistence.*;
import org.hibernate.annotations.GenericGenerator;
import java.util.*;
import com.halliburton.qc.util.Utilities;
@Entity
@Table(name = "TASKS")
public class Tasks {
    @Id
    @Column(name = "ID")
    private String id;
    @Column(name = "ORDER_")
    private Long order;
    @Column(name = "ACTIVITY")
    private String activity;
    @Column(name = "PORCENT")
    private Long porcent;
    @Column(name = "owner_task")
    private String ownerTask;
    @Column(name = "COMMENTS_")
    private String comments;
    @Column(name = "ORDER_ID")
    private String orderId;
    @Column(name = "STATUS")
    private Long status;
    @Temporal(TemporalType.DATE)
    @Column(name = "DATE_CREATE")
    private Date dateCreate;
    @Temporal(TemporalType.DATE)
    @Column(name = "DATE_UPDATE")
    private Date dateUpdate;
    @Temporal(TemporalType.DATE)
    @Column(name = "DATE_DELETE")
    private Date dateDelete;
    @Column(name = "USER_CREATE")
    private String userCreate;
    @Column(name = "USER_UPDATE")
    private String userUpdate;
    @Column(name = "USER_DELETE")
    private String userDelete;
    @ManyToOne(optional = false)
    @JoinColumn(name = "owner_task", referencedColumnName = "id",
insertable = false, updatable = false)
    private Usuario usuarioOwnerTask;
    @ManyToOne(optional = false)
    @JoinColumn(name = "ORDER_ID", referencedColumnName
= "id", insertable = false, updatable = false)
    private Orders orders;
    @ManyToOne(optional = false)
    @JoinColumn(name = "ACTIVITY", referencedColumnName
= "ID_GROUPTAREA", insertable = false, updatable = false)
    private Grouptarea grouptarea;
    @Transient
    private Integer code;
    @Transient
    private Date datecreate;
    @Transient
    private String grupoNombre;
    @Transient
    private String actividadNombre;
    @Transient
    private String actividad;
    @Transient
    private String usuarioNombre;
    @Transient
    private String usuarioApellido;
    @Transient
    private String fullName;
    @Transient
    private Date datedue;
    @Transient
    private String idOrders;
}
```

| Metadata | |
|---|---|
| **Entidad** | AssociateFile.java |

```
package com.halliburton.qc.entities;
import javax.persistence.*;
import java.util.*;
import com.halliburton.qc.util.Utilities;
@Entity
@Table(name = "ASSOCIATED_FILES")
public class AssociatedFiles {
```

```
@Id
@Column(name = "ID")
private String id;
@Column(name = "ORDER_ID")
private String orderId;
@Column(name = "TYPE")
private String type;
@Column(name = "NAME")
private String name;
@Column(name = "EXT")
private String ext;
@Column(name = "SIZE")
private Long sizeFile;
@Column(name = "PATH")
private String path;
@Column(name = "MEDIO")
private String medio;
@Column(name = "TYPEBARCODE")
private String typebarcode;
@Column(name = "BARCODE")
private String barcode;
@Column(name = "REGION")
private String region;
@Column(name = "CUENCA")
private String cuenca;
@Column(name = "LOTE")
private String lote;
@Column(name = "YACIMIENTO")
private String yacimiento;
@Column(name = "ESCALA")
private String escala;
@Column(name = "UWI")
private String uwi;
@Column(name = "THEMEDESCRIPTOR")
private String themedescriptor;
@Column(name = "SUBTITLE")
private String subtitle;
@Column(name = "INSTITUTIONAL")
private String institutional;
@Column(name = "ILUSTRATION")
private String ilustration;
@Column(name = "VALIDQC")
private String validqc;
@Column(name = "VALIDOT")
private String validot;
@Column(name = "STATUS")
private Long status;
@Temporal(TemporalType.TIMESTAMP)
@Column(name = "DATE_CREATE")
private Date dateCreate;
@Temporal(TemporalType.TIMESTAMP)
@Column(name = "DATE_UPDATE")
private Date dateUpdate;
@Temporal(TemporalType.TIMESTAMP)
@Column(name = "DATE_DELETE")
private Date dateDelete;
@Column(name = "USER_CREATE")
private String userCreate;
```

```
@Column(name = "CANALNOMINAL3DC")
private String canalNominal3DC;
@Column(name = "CANALAUXILIAR3DC")
private String canalAuxiliar3DC;
@Column(name = "AREATOTAL3DC")
private String areaTotal3DC;
@Column(name = "SISTEMACOORDE3DC")
private String sistemaCoorde3DC;
@Column(name = "REPORTETOPO3DC")
private String reporteTopo3DC;
@Column(name = "REPORTEOPERACIONES3DC")
private String reporteOperaciones3DC;
@Column(name = "FICHASISMICA3DC")
private String fichaSismica3DC;
@Column(name = "DISPAROINICIAL3DC")
private String disparoInicial3DC;
@Column(name = "DISPAROFINAL3DC")
private String disparoFinal3DC;
@Column(name = "CAMPOINICIAL3DC")
private String campoInicial3DC;
@Column(name = "CAMPOFINAL3DC")
private String campoFinal3DC;
@Column(name = "REPORTEOBSERVADOR3DC")
private String reporteObservador3DC;
@Column(name = "LONGITUDREGISTRO3DC")
private String longitudRegistro3DC;
@Column(name = "INTERVALOMUESTREO3DC")
private String intervaloMuestreo3DC;
@Column(name = "CANTIDADDISPARO3DC")
private String cantidadDisparo3DC;
@Column(name = "COORDENADASDR3DC")
private String coordenadasDR3DC;
@Column(name = "COMENTARIO3DC")
private String comentario3DC;
@Column(name = "ANALISISVELOCIDAD")
private String analisisVelocidad;
@Column(name = "INFORMEVERIFICACION")
private String informeVerificacion;
@Column(name = "LONGITUDLINEA")
private String longitudLinea;
@Column(name = "NROCRUCESLINEA")
private String nrocrucesLinea;
@Column(name = "INTERVALOCDPS")
private String intervaloCDPS;
@Column(name = "POZOUWI")
private String pozouwi;
@Column(name = "POZOEQUIPO")
private String pozoequipo;
@Column(name = "POZOPROFTOTAL")
private String pozoproftotal;
@Column(name = "POZOTIPO")
private String pozotipo;
@Column(name = "POZOCROMAT")
private String pozocromat;
@Column(name = "POZOFECHAINICIO")
private String pozofechainicio;
@Column(name = "POZOFECHACOMPLET")
private String pozofechacomplet;
```

```
@Column(name = "USER_UPDATE")
private String userUpdate;
@Column(name = "USER_DELETE")
private String userDelete;
@Column(name = "MEDIOBARCODE")
private String medioBarcode;
@Column(name = "DATATYPE")
private String dataType;
@Column(name = "CERTIGIS")
private Boolean certiGis;
@Column(name = "CERTISEIS")
private Boolean certiSeis;
@Column(name = "CERTIWELL")
private Boolean certiWell;
@Column(name = "CERTIDOC")
private Boolean certiDoc;
@Column(name = "PRINDOC")
private Boolean prinDoc;
@Temporal(TemporalType.TIMESTAMP)
@Column(name = "CERTIDATE")
private Date certiDate;
@Column(name = "COMMENTDOC")
private String commentDoc;
@Column(name = "COMMENTDOCGIS")
private String commentDocgis;
@Column(name = "COMMENTDOCWELL")
private String commentDocwell;
@Column(name = "COMMENTDOCSEIS")
private String commentDocseis;
@Column(name = "COMMENTDOCTEC")
private String commentDoctec;
@Column(name = "TITLEDOCUMENT")
private String titleDocument;
@Column(name = "NUMBERPAGES")
private Double numberPages;
@Temporal(TemporalType.DATE)
@Column(name = "DOCUDATE")
private Date docuDate;
@Column(name = "MATERIA")
private String materia;
@Column(name = "INFOTECNIC")
private String infoTecnic;
@Column(name = "ELLIPSOID")
private String ellipsoid;
@Column(name = "PROJECTION")
private String projection;
@Column(name = "ZONE")
private String zone;
@Column(name = "DATUM")
private String datum;
@Column(name = "LONGWESTESI")
private Float longWestesi;
@Column(name = "LONGWESTEDI")
private Float longWestedi;
@Column(name = "LATSOUTHESI")
private Float latSouthesi;
@Column(name = "LATSOUTHEDI")
private Float latSouthedi;
@Column(name = "CORNERUPLN")
private Float cornerUpln;
@Column(name = "CORNERUPLE")
private Float cornerUple;
@Column(name = "CORNERDOWNLN")
private Float cornerDownln;
@Column(name = "CORNERDOWNLE")
private Float cornerDownle;
@Column(name = "SUMMARY")
private String summary;
@Column(name = "CONTENT")
private String content;
```

```
@Column(name = "POZOREFERENCIA")
private String pozoreferencia;
@Column(name = "POZOELEVACION")
private String pozoelevacion;
@Column(name = "POZOPARAMETROS")
private String pozoparametros;
@Column(name = "POZOSURVEY")
private String pozosurvey;
@Column(name = "POZOCOORDENADAS")
private String pozocoordenadas;
@Column(name = "POZOZONA")
private String pozozona;
@Column(name = "POZOCOORDSUPERN")
private String pozocoordsupern;
@Column(name = "POZOCOORDSUPERE")
private String pozocoordsupere;
@Column(name = "POZOCOORDFONDN")
private String pozocoordfondn;
@Column(name = "POZOCOORDFONDE")
private String pozocoordfonde;
@Column(name = "GISFECHA")
private String gisfecha;
@Temporal(TemporalType.DATE)
@Column(name = "DOCGISFECHA")
private Date docgisFecha;
@Column(name = "EDITDOC")
private Boolean editDoc;
@Column(name = "ORIGINALDOC")
private Boolean originalDoc;
@Temporal(TemporalType.DATE)
@Column(name = "DOCTECFECHA")
private Date doctecFecha;
@Temporal(TemporalType.DATE)
@Column(name = "FECHAINICIOPOZO")
private Date fechainicioPozo;
@Temporal(TemporalType.DATE)
@Column(name = "FECHACOMPLETPOZO")
private Date fechacompletPozo;
@Column(name = "PERFILPOZO")
private Boolean perfilPozo;
@Column(name = "TYPEWELL")
private String typeWell;
@Column(name = "POZOMUESTRARECIPIENTE")
private String pozomuestraRecipiente;
@Column(name = "POZOMUESTRACANTIDAD")
private String pozomuestraCantidad;
@Column(name = "POZOMUESTRATIPO")
private String pozomuestraTipo;
@Column(name = "POZOMUESTRAFORMACION")
private String pozomuestraFormacion;
@Column(name = "POZOMUESTRATOPE")
private String pozomuestraTope;
@Column(name = "POZOMUESTRABASE")
private String pozomuestraBase;
@Column(name = "POZOINFORMES")
private String pozoInformes;
@Column(name = "POZOPERFILESCANEADO")
private String pozoperfilEscaneado;
@Column(name = "POZODUPLICIDADIMAGENES")
private String pozoduplicidadImagenes;
@Column(name = "POZOCONTIENEINSCRIPCIONES")
private String pozocontieneInscripciones;
@Column(name = "POZOCABEZALCOINCIDEPERFIL")
private String pozocabezalcoincidePerfil;
@Column(name = "POZOARCHIVOCURVASIMPRESO")
private String pozoarchivocurvasImpreso;
@Column(name = "POZOARCHIVOPERFILPAPEL")
private String pozoarchivoperfilPapel;
@Column(name = "POZOCONVERTIDOTIF")
private String pozoconvertidoTif;
```

```
@Column(name = "STANDARDOC")
private Boolean standarDoc;
@Column(name = "DIGITALDOC")
private Boolean digitalDoc;
@Column(name = "PRINTDOC")
private String printDoc;
@Column(name = "LOGNAME")
private String logName;
@Column(name = "LOGTYPE")
private String logType;
@Column(name = "LOGSERVICE")
private String logService;
@Column(name = "LOGSOURCE")
private String logSource;
@Column(name = "LOGGINCONTRACTOR")
private String logginContractor;
@Column(name = "TOPDEPTH")
private Integer topDepth;
@Column(name = "BOTTOMDEPTH")
private Integer bottomDepth;
@Column(name = "LOGRUN")
private Integer logRun;
@Temporal(TemporalType.DATE)
@Column(name = "LOGDATE")
private Date logDate;
@Column(name = "CAMPAINGNAME")
private String campaingName;
@Column(name = "REPORTCODE")
private String reportCode;
@Column(name = "IMAGEASSOCIATED")
private String imageAssociated;
@Column(name = "COMPARISONTOPOGRAPHIC")
private Boolean comparisonTopographic;
@Column(name = "TOPOGRAPHICDATA")
private Boolean topographicData;
@Column(name = "LASTCDP")
private Integer lastCdp;
@Column(name = "FIRSTCDP")
private Integer firstCdp;
@Column(name = "MAJORSHOT")
private Double majorShot;
@Column(name = "MINORSHOT")
private Double minorShot;
@Temporal(TemporalType.DATE)
@Column(name = "DATEPROCESS")
private Date dateProcess;
@Column(name = "PROCESSTO")
private String processTo;
@Column(name = "PROCESSBY")
private String processBy;
@Column(name = "PROCESSTYPE")
private String processType;
@Column(name = "NAMELINE")
private String nameLine;
@Column(name = "TYPESEISMIC")
private String typeSeismic;
@Column(name = "BETWEENCDP")
private Double betweenCdp;
@Column(name = "LONGLINE")
private Double longLine;
@Column(name = "LONGTRACE")
private Double longTrace;
@Column(name = "SAMPLINGRATE")
private Double samplingRate;
@Column(name = "AREAPROCESS")
private Double areaProcess;
@Column(name = "NUMELINEAS3DC")
private String numeLineas3DC;
@Column(name = "POZOPERFILESIMPRESOS")
private String pozoperfilesImpresos;
@Column(name = "POZOPERFILESASOCIADOS")
private String pozoperfilesAsociados;
@Column(name = "SISMICANROCRUCES")
private String sismicanroCruces;
@Column(name = "GISMEDIOORIGINAL")
private String gismedioOriginal;
@Column(name = "GISESTADOMAPADIGITAL")
private String gisestadomapaDigital;
@Column(name = "GISESTADOMAPAIMPRESO")
private String gisestadomapaImpreso;
@Column(name = "GISMEDIOREFERENCIA")
private String gismedioReferencia;
@Column(name = "GISTIPOMAPA")
private String gistipoMapa;
@Column(name = "GISESCANEOMAPAIMPRESO")
private String gisescaneomapaImpreso;
@Column(name = "GISINFORMACIONHISTORICA")
private String gisinformacionHistorica;
@Column(name = "SISMICAPESOVERIFICADOANTIGUO")
private Long sismicapesoverificadoAntiguo;
@Column(name = "SISMICAPESOCERTIFICADOANTIGUO")
private Long sismicapesocertificadoAntiguo;
@Column(name = "ARCHIVOSELECTCATALOGO")
private String archivoselectCatalogo;
@Column(name = "ARCHIVOIMPRESOCALIDAD")
private String archivoimpresoCalidad;
@Column(name = "ARCHIVOESCANEOINFOFALTANTE")
private String archivoescaneoinfoFaltante;
@Column(name = "ARCHIVOCONVERSIONPDF")
private String archivoconversionPdf;
@Column(name = "ARCHIVOPDFCOINCIDEIMPRESO")
private String archivopdfcoincideImpreso;
@Column(name = "ARCHIVOFILTROCATALOGO")
private Boolean archivofiltroCatalogo;
@Column(name = "ARCHIVOORIGINALCODIGO")
private String archivooriginalCodigo;
@Column(name = "POZOCORRELATIVOBARCODE")
private String pozocorrelativoBarcode;
@Temporal(TemporalType.DATE)
@Column(name = "DOCSISMICAPROCESADAFECHA")
private Date docsismicaprocesadaFecha;
@Column(name = "SEISMICUPDATE")
private Boolean seismicUpdate;
@Column(name = "NUMPAGARCHIVO")
private String numpagArchivo;
@Column(name = "ORDER_COMPLETO")
private String orderCompleto;
@Column(name = "VERIFGIS")
private Boolean verifGis;
@Column(name = "VERIFSEIS")
private Boolean verifSeis;
@Column(name = "VERIFWELL")
private Boolean verifWell;
@Column(name = "VERIFDOC")
private Boolean verifDoc;
@Column(name = "RECEPTYPEFILE")
private String receptypeFile;
@Column(name = "SURVEYPARTES")
private String surveyPartes;
@Column(name = "POZOMODULONRO")
private String pozomoduloNro;
@Column(name = "POZOMODULONROTABLA")
private String pozomodulonroTabla;
private String pozomoduloexisteInfo;
@Column(name = "POZOMODULOINFOCONSIST")
private String pozomoduloinfoConsist;
}
```

| Certificado | |
|---|---|
| **Entidad** | CertificadoGeneral.java |

```
package com.halliburton.qc.entities;
import java.util.Date;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
@Entity
@Table(name = "CERTIFICADOGENERAL")
public class CertificadoGeneral {

@Id
@Column(name = "ID")
        private String id;

@Column(name = "GRUPOFUNCIONAL")
        private String grupoFuncional;

@Column(name = "NUMCERTI")
        private Integer numCerti;

@Column(name = "ORDERID")
        private String orderId;

@Column(name = "TIPOCERTIFICADO")
        private String tipoCertificado;

@Temporal(TemporalType.DATE)
@Column(name = "DATECREATE")
        private Date dateCreate;

@Temporal(TemporalType.DATE)
@Column(name = "DATEUPDATE")
        private Date dateUpdate;

@Column(name = "USERCREATE")
        private String userCreate;

@Column(name = "USERUPDATE")
        private String userUpdate;

@Column(name = "IDENTIFICADORBARCODECERTI")
        private String identificadorbarcodeCerti;

@Column(name = "POZOCERTI")
        private String pozoCerti;

@Column(name = "SURVEYCERTI")
        private String surveyCerti;

@Column(name = "EXPEDOT")
        private String expedOt;

@Column(name = "EXPEDCERTI")
        private String expedCerti;

@Column(name = "EXPEDUSERREGISTER")
        private String expeduserRegister;

@Column(name = "EXPEDPESOTOTALCARGA")
        private Double expedpesototalCarga;

@Column(name = "EXPEDPESOTOTALVERIF")
        private Double expedpesototalVerif;

@Column(name = "EXPEDPESOTOTALUPDATE")
        private Double expedpesototalUpdate;

@Column(name = "EXPEDUSERUPDATE")
        private String expeduserUpdate;

@Column(name = "CERTISUBTIPO")
        private String certisubTipo;

@Column(name = "SURVEYCERTIPARTE")
        private String surveycertiParte;

}
```

✓ Servicios

**Orden de Trabajo**

**Servicio** OrderService.java

```
package com.halliburton.qc.service;
import java.sql.Date;
import java.util.List;
import java.util.Optional;
import java.util.*;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import com.halliburton.qc.entities.*;
import com.halliburton.qc.repository.*;
import com.halliburton.qc.util.OrdersParam;
import com.querydsl.core.types.Predicate;
@Service
@Transactional
public class OrdersService {
@Autowired
private OrdersRepository repository;

public Optional<Orders> findById(String id) {
                return repository.findById(id);
        }

public void save(Orders orders) {
                repository.save(orders);
        }

public void update(Orders orders) {
                save(orders);
        }

public void deleteById(String id) {
                repository.deleteById(id);
        }

public Page<Orders> findAll(Pageable pageable) {
                return repository.findAll(pageable);
        }

public boolean isOrdersExist(Orders orders) {
                return findById(orders.getId()) != null;
        }

public Page<Orders> findByParams(String code, String registerby,
String requestedby, String subject, String observation, String
signedby, Pageable pageable) {
return repository.findByParams(code, registerby, requestedby,
subject, observation, signedby, pageable);
        }

public Page<Orders> findOrdersByGroupOrOwner(OrdersParam param,
List<String> ids, Pageable pageable) {
Page<Orders> page = null;
        if (param.getOwner() != null &&
!param.getOwner().isEmpty()) {
return repository.findOrdersByListIdOwner(param.getOwner(), ids,
pageable);
        }
        if (param.getListIds() != null &&
!param.getListIds().isEmpty()) {
        return
repository.findOrdersByListIdsGroup(param.getListIds(), ids, pageable);
        }
        page = repository.findAllByOrders(pageable);
                return page;
        }

public Integer findMaxCodeByDateCreateAndTypeOf(Integer datecreate,
String typeot) {
return repository.findMaxCodeByDateCreateAndTypeOf(datecreate,
typeot);
        }

public Page<Orders> findfechaOT(String state, Pageable pageable) {
        return repository.findfechaOT(state, pageable);
}
public Integer findfechaOT2(Integer datecreate) {
        return repository.findfechaROT2(datecreate);
}

public Page<Orders> qfindAll(Predicate predicate, Pageable pageable) {
                return repository.findAll(predicate, pageable);
        }

}
```

| Tarea | |
|---|---|
| **Servicio** | TaskService.java |

```
package com.halliburton.qc.service;
import java.util.ArrayList;
import java.util.Date;
import java.util.Iterator;
import java.util.List;
import java.util.Optional;
import java.util.function.Function;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.PageImpl;
import org.springframework.data.domain.Pageable;
import org.springframework.data.domain.Sort;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import com.halliburton.qc.entities.*;
import com.halliburton.qc.repository.*;
import com.querydsl.core.types.Predicate;
@Service
@Transactional
public class TasksService {
	@Autowired
private TasksRepository repository;

public Optional<Tasks> findById(String id) {
	return repository.findById(id);
	}
public Tasks findId(String id) {
	return repository.getOne(id);
	}
public void save(Tasks tasks) {
	repository.save(tasks);
	}
public void update(Tasks tasks) {
	save(tasks);
	}
public void deleteById(String id) {
	repository.deleteById(id);
	}
public Page<Tasks> findAll(Pageable pageable) {
	return repository.findAll(pageable);
	}
public boolean isTasksExist(Tasks tasks) {
	return findById(tasks.getId()) != null;
	}
public Page<Tasks> findByParams(String comments, String orderId,
Pageable pageable) {
	if (comments != null && !comments.isEmpty() &&
orderId != null && !orderId.isEmpty()) {
	return
repository.findByParamsCommentsAndCorderId(comments, orderId,
pageable);
	} else if ((comments == null || comments.isEmpty()) &&
orderId != null && !orderId.isEmpty()) {
	return repository.findByParamsOrderId(orderId, pageable);
	} else if ((orderId == null || orderId.isEmpty()) &&
comments != null && !comments.isEmpty()) {
	return repository.findByParamsComments(comments,
pageable);
		}
	return new PageImpl<>(new ArrayList<>());
	}
public Integer findTasksAvgyByOrderId(String orderId) {
	return repository.findTasksAvgyByOrderId(orderId);
}

public List<Tasks> findTasksDetailByGroups(List<String> listIds) {
	if (listIds != null && !listIds.isEmpty())
		return repository.findTasksDetailByGroups(listIds);
	else
	return repository.findTasksDetailAll();
}

public Page<Tasks> qfindAll(
	Predicate predicate,	Pageable
pageable){
return repository.findAll(predicate, pageable);
}
}
```

| Metadata | |
|---|---|
| **Servicio** | AssociateFileService.java |

```
package com.halliburton.qc.service;
import java.util.Date;
import java.util.List;
import java.util.Optional;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import com.halliburton.qc.entities.*;
import com.halliburton.qc.repository.*;
import com.querydsl.core.types.Predicate;
@Service
@Transactional
public class AssociatedFilesService {
	@Autowired
private AssociatedFilesRepository repository;

public Optional<AssociatedFiles> findById(String id) {
	return repository.findById(id);
	}

public AssociatedFiles findId(String id) {
	return repository.getOne(id);
	}

public void save(AssociatedFiles associatedFiles) {
	repository.save(associatedFiles);
	}

public void update(AssociatedFiles associatedFiles) {
	save(associatedFiles);
	}

public void deleteById(String id) {
	repository.deleteById(id);
	}

public Page<AssociatedFiles> findAll(Pageable pageable) {
	return repository.findAll(pageable);
	}

public boolean isAssociatedFilesExist(AssociatedFiles
associatedFiles) {
	return findById(associatedFiles.getId()) != null;
	}

public Page<AssociatedFiles> findByParams(
String  orderId,  Pageable pageable){
	return repository.findByParams(orderId, pageable);
}

public Page<AssociatedFiles> findByParamstypebarcode(
String  typeBarcode, String  barcoden,        Pageable pageable){
	return
repository.findByParamstypebarcode(typeBarcode,barcoden,
pageable);
}

public Page<AssociatedFiles> findByParamsmediobarcode(
String  medioBarcode, Pageable pageable){
	return repository.findByParamsmediobarcode(medioBarcode,
pageable);
}

public Page<AssociatedFiles> findByParamstypebarcodeequals(
	String  typeBarcode, String  barcoden,        Pageable
pageable){
	return
repository.findByParamstypebarcodeequals(typeBarcode,barcoden,
pageable);
}

public Page<AssociatedFiles> qfindAll(      Predicate predicate,Pageable
pageable){
	return repository.findAll(predicate, pageable);
}

public Page<AssociatedFiles> findByParamsfind_associatedfile_orderid(
String  orderId, Pageable pageable){
	return
repository.findByParamsfind_associatedfile_orderid(orderId, pageable);
}

}
```

| Certificados | |
|---|---|
| **Servicio** | CertificadoGeneralService.java |

```
package com.halliburton.qc.service;
import java.util.Date;
import java.util.List;
import java.util.Optional;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import com.halliburton.qc.entities.*;
import com.halliburton.qc.repository.*;
@Service
@Transactional
public class CertificadoGeneralService {
	@Autowired
	private CertificadoGeneralRepository repository;
```

```
public Optional<CertificadoGeneral> findById(String id) {
	return repository.findById(id);
	}

public void save(CertificadoGeneral certificadogeneral) {
	repository.save(certificadogeneral);
	}

public void update(CertificadoGeneral certificadogeneral) {
	save(certificadogeneral);
	}

public void deleteById(String id) {
	repository.deleteById(id);
	}

public Page<CertificadoGeneral> findAll(Pageable pageable) {
	return repository.findAll(pageable);
	}

public boolean isletterTransferExist(CertificadoGeneral
certificadogeneral) {
	return findById(certificadogeneral.getId()).isPresent();
	}


}
```

```
public Page<CertificadoGeneral> findByParamsorderId(
	String  orderId, Pageable pageable){
	return repository.findByParamsorderId(orderId, pageable);
	}

public Page<CertificadoGeneral> findByParamscertificadogeneralorderid(
	String  orderId, Pageable pageable){
	return
repository.findByParamscertificadogeneralorderid(orderId, pageable);
}
```

```
}
```

✓ Controladores

**Orders**

**Controlador** AssociateFileController.java

```
package com.halliburton.qc.controller;
import java.util.List;
import java.util.Optional;
import org.slf4j.Logge.*r;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.util.UriComponentsBuilder;
import org.springframework.data.domain.Pageable;
import com.halliburton.qc.entities.*;
import com.halliburton.qc.service.*;
import com.halliburton.qc.util.CustomErrorType;
@RestController
public class OrderfilesController {
   public static final Logger logger = LoggerFactory.getLogger(OrderfilesController.class);
   @Autowired
   OrderfilesService orderfilesService;
```

**Lista de Ordenes de Trabajo**

```
   @CrossOrigin(origins = "*", maxAge = 3600)
   @RequestMapping(value = "/orderfiles/", method = RequestMethod.GET)
   public ResponseEntity<Page<Orderfiles>> listAllOrderfiless(Pageable pageable) {
            Page<Orderfiles> page = orderfilesService.findAll(pageable);
          if (page.getContent().isEmpty()) {
          return new ResponseEntity(HttpStatus.NO_CONTENT);       }
     return new ResponseEntity<Page<Orderfiles>>(page, HttpStatus.OK);    }
```

**Consulta de Orden de Trabajo**

```
@CrossOrigin(origins = "*", maxAge = 3600)
@RequestMapping(value = "/orderfile/{id}", method = RequestMethod.GET)
   public ResponseEntity<?> getOrderfiles(@PathVariable("id") String id) {
      logger.info("Fetching Orderfiles with id {}", id);
      Optional<Orderfiles> orderfiles = orderfilesService.findById(id);
      if (!orderfiles.isPresent()) {
         logger.error("Orderfiles with id {} not found.", id);
         return new ResponseEntity(new CustomErrorType("Orderfiles with id " + id + " not found"), HttpStatus.NOT_FOUND);
      }  return new ResponseEntity<Orderfiles>(orderfiles.get(), HttpStatus.OK);  }
```

**Registrar Orden de Trabajo**

```
   @CrossOrigin(origins = "*", maxAge = 3600)
   @RequestMapping(value = "/orderfile/", method = RequestMethod.POST)
   public ResponseEntity<?> createOrderfiles(@RequestBody Orderfiles orderfiles, UriComponentsBuilder ucBuilder) {
      logger.info("Creating Orderfiles : {}", orderfiles);
      orderfilesService.save(orderfiles);
      HttpHeaders headers = new HttpHeaders();
      headers.setLocation(ucBuilder.path("/api/orderfiles/{id}").buildAndExpand(orderfiles.getId()).toUri());
      return new ResponseEntity<String>(headers, HttpStatus.CREATED); }
```

**Actualizar Orden**

```
   @CrossOrigin(origins = "*", maxAge = 3600)
   @RequestMapping(value = "/orderfile/{id}", method = RequestMethod.PUT)
   public ResponseEntity<?> updateOrderfiles(@RequestBody Orderfiles orderfiles) {
     logger.info("Updating Orderfiles with id {}", orderfiles.getId());
     Optional<Orderfiles> currentOrderfiles = orderfilesService.findById(orderfiles.getId());
     if (!currentOrderfiles.isPresent()) {
       logger.error("Unable to update. Orderfiles with id {} not found.", orderfiles.getId());
       return new ResponseEntity(new CustomErrorType("Unable to upate. Orderfiles with id " + orderfiles.getId() + " not found."); }
       orderfilesService.update(orderfiles);
       return new ResponseEntity<Orderfiles>(orderfiles, HttpStatus.OK); }
}
```

**Tareas**

**Controlador** | TaskController.java

```
package com.halliburton.qc.controller;
import java.util.*;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.util.UriComponentsBuilder;
import com.halliburton.qc.entities.*;
import com.halliburton.qc.service.*;
import com.halliburton.qc.util.CustomErrorType;
@RestController
public class TasksController {  public static final Logger logger = LoggerFactory.getLogger(TasksController.class);
   @Autowired
   TasksService tasksService;
```

**Listar Tareas**

```
   @CrossOrigin(origins = "*", maxAge = 3600)
   @RequestMapping(value = "/tasks/", method = RequestMethod.GET)
   public ResponseEntity<Page<Tasks>> listAllTaskss(Pageable pageable) {Page<Tasks> page = tasksService.findAll(pageable);
      if (page.getContent().isEmpty()) { return new ResponseEntity(HttpStatus.NO_CONTENT);  }
      return new ResponseEntity<Page<Tasks>>(page, HttpStatus.OK);  }
```

**Consulta Tarea**

```
   @CrossOrigin(origins = "*", maxAge = 3600)
   @RequestMapping(value = "/task/{id}", method = RequestMethod.GET)
   public ResponseEntity<?> getTasks(@PathVariable("id") String id) {
      logger.info("Fetching Tasks with id {}", id);
      Optional<Tasks> tasks = tasksService.findById(id);
      if (!tasks.isPresent()) {  logger.error("Tasks with id {} not found.", id);
         return new ResponseEntity(new CustomErrorType("Tasks with id " + id + " not found"), HttpStatus.NOT_FOUND);  }
      return new ResponseEntity<Tasks>(tasks.get(), HttpStatus.OK);    }
```

**Crear Tarea**

```
   @CrossOrigin(origins = "*", maxAge = 3600)
   @RequestMapping(value = "/task/", method = RequestMethod.POST)
   public ResponseEntity<?> createTasks(@RequestBody Tasks tasks, UriComponentsBuilder ucBuilder) {
      logger.info("Creating Tasks : {}", tasks);
      tasksService.save(tasks);
      HttpHeaders headers = new HttpHeaders();
      headers.setLocation(ucBuilder.path("/api/tasks/{id}").buildAndExpand(tasks.getId()).toUri());
      return new ResponseEntity<String>(headers, HttpStatus.CREATED);  }
```

**Actualizar Tarea**

```
   @CrossOrigin(origins = "*", maxAge = 3600)
   @RequestMapping(value = "/task/{id}", method = RequestMethod.PUT)
   public ResponseEntity<?> updateTasks(@PathVariable("id") String id, @RequestBody Tasks tasks) {
      logger.info("Updating Tasks with id {}", id);
      Optional<Tasks> currentTasks = tasksService.findById(id);
      if (!currentTasks.isPresent()) {
         logger.error("Unable to update. Tasks with id {} not found.", id);
         return new ResponseEntity(new CustomErrorType("Unable to upate. Tasks with id " + id + " not found."), HttpStatus.NOT_FOUND);       }
       tasksService.update(tasks);
      return new ResponseEntity<Tasks>(tasks, HttpStatus.OK);    }
```

**Eliminar Tarea**

```
   @CrossOrigin(origins = "*", maxAge = 3600)
   @RequestMapping(value = "/task/{id}", method = RequestMethod.DELETE)
   public ResponseEntity<?> deleteTasks(@PathVariable("id") String id) {  logger.info("Fetching & Deleting Tasks with id {}", id);
      Optional<Tasks> tasks = tasksService.findById(id);
      if (!tasks.isPresent()) {
         logger.error("Unable to delete. Tasks with id {} not found.", id);
         return new ResponseEntity(new CustomErrorType("Unable to delete. Tasks with id " + id + " not found."),
               HttpStatus.NOT_FOUND);       }
      tasksService.deleteById(id);
      return new ResponseEntity<Tasks>(HttpStatus.NO_CONTENT); }
}
```

**Metadata**

| **Controlador** | AssociateFileController.java |
|---|---|

```
package com.halliburton.qc.controller;
import java.util.List;
import java.util.Optional;
import org.slf4j.Logge.*;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.util.UriComponentsBuilder;
import org.springframework.data.domain.Pageable;
import com.halliburton.qc.entities.*;
import com.halliburton.qc.service.*;
import com.halliburton.qc.util.CustomErrorType;
@RestController
public class AssociatedFilesController {
        public static final Logger logger = LoggerFactory.getLogger(AssociatedFilesController.class);
        @Autowired
        AssociatedFilesService associatedFilesService;
```

**Listar Metadata**

```
@CrossOrigin(origins = "*", maxAge = 3600)
@RequestMapping(value = "/associatedFiles/", method = RequestMethod.GET)
        public ResponseEntity<Page<AssociatedFiles>> listAllAssociatedFiless(Pageable pageable) {
        Page<AssociatedFiles> page = associatedFilesService.findAll(pageable);
        if (page.getContent().isEmpty()) {
                return new ResponseEntity(HttpStatus.NO_CONTENT);
                }
                return new ResponseEntity<Page<AssociatedFiles>>(page, HttpStatus.OK);
        }
```

**Listar Metadata por Orden de Trabajo**

```
@CrossOrigin(origins = "*", maxAge = 3600)
@RequestMapping(value = "/find_associatedfile_orderid/", method = RequestMethod.GET)
        public ResponseEntity<Page<AssociatedFiles>> find_associatedfile_orderid(
        @RequestParam(value = "orderId", required = false) String orderId, Pageable pageable) {
        Page<AssociatedFiles> page = associatedFilesService.findByParamsfind_associatedfile_orderid(orderId, pageable);
                if (page.getContent().isEmpty()) {
                        return new ResponseEntity(HttpStatus.NO_CONTENT);
                }
                return new ResponseEntity<Page<AssociatedFiles>>(page, HttpStatus.OK);
        }
```

**Actualizar Metadata**

```
@CrossOrigin(origins = "*", maxAge = 3600)
@RequestMapping(value = "/associatedFileupd/{id}", method = RequestMethod.PUT)
        public ResponseEntity<?> updateAssociatedFiles(@PathVariable("id") String id,
                @RequestBody List<AssociatedFileList> params) {
                for (AssociatedFileList associatedFiles : params) {
                        try {   AssociatedFiles existID = associatedFilesService.findId(associatedFiles.getId());
                                for (AssociatedFiles fielUpd : associatedFiles.getFieldsUpd()) {
                                        logger.info("Updating AssociatedFiles with id {}", associatedFiles.getId());
                                        logger.info("Updating AssociatedFiles Field : Value {}", fielUpd);
                                        Utilities.copyNonNullProperties(fielUpd, existID);
                                        associatedFilesService.update(existID);
                                }
                        } catch (Exception e) {
                                logger.error("Unable to update. AssociatedFiles with id {} not found.", associatedFiles.getId());
                        }
                }
                return new ResponseEntity<String>("", HttpStatus.OK);
        }
```

**Eliminar Metadata**

```
@CrossOrigin(origins = "*", maxAge = 3600)
@RequestMapping(value = "/deleteAssociatedFile/{id}", method = RequestMethod.POST)
	public ResponseEntity<?> deleteAssociatedFiles(@PathVariable("id") String id) {
		logger.info("Fetching & Deleting AssociatedFiles with id {}", id);
		Optional<AssociatedFiles> associatedFiles = associatedFilesService.findById(id);
		if (!associatedFiles.isPresent()) {
			logger.error("Unable to delete. AssociatedFiles with id {} not found.", id);
			return new ResponseEntity(
				new CustomErrorType("Unable to delete. AssociatedFiles with id " + id + " not found."),
					HttpStatus.NOT_FOUND);
		}
		associatedFilesService.deleteById(id);
		return new ResponseEntity<AssociatedFiles>(HttpStatus.NO_CONTENT);
	}
```

**Buscar por Tipo de Barcode**

```
	@CrossOrigin(origins = "*", maxAge = 3600)
	@RequestMapping(value = "/findtypebarcode/", method = RequestMethod.GET)
	public ResponseEntity<Page<AssociatedFiles>> findtypebarcode(
			@RequestParam(value = "typebarcode", required = false) String typebarcode,
			@RequestParam(value = "barcode", required = false) String barcode, Pageable pageable) {
		Page<AssociatedFiles> page = associatedFilesService.findByParamstypebarcode(typebarcode, barcode, pageable);
		if (page.getContent().isEmpty()) {
			return new ResponseEntity(HttpStatus.NO_CONTENT);
		}
		return new ResponseEntity<Page<AssociatedFiles>>(page, HttpStatus.OK);
	}
```

**Buscar por Barcode**

```
	@CrossOrigin(origins = "*", maxAge = 3600)
	@RequestMapping(value = "/findtypebarcodeequals/", method = RequestMethod.GET)
	public ResponseEntity<Page<AssociatedFiles>> findtypebarcodeequals(
			@RequestParam(value = "typebarcode", required = false) String typebarcode,
			@RequestParam(value = "barcode", required = false) String barcode, Pageable pageable) {

		Page<AssociatedFiles> page = associatedFilesService.findByParamstypebarcodeequals(typebarcode, barcode, pageable);
		if (page.getContent().isEmpty()) {
			return new ResponseEntity(HttpStatus.NO_CONTENT);
		}
		return new ResponseEntity<Page<AssociatedFiles>>(page, HttpStatus.OK);
	}
```

**Buscar por Codigo de Medio Digital**

```
	@CrossOrigin(origins = "*", maxAge = 3600)
	@RequestMapping(value = "/findmediobarcode/", method = RequestMethod.GET)
	public ResponseEntity<Page<AssociatedFiles>> findmediobarcode(
			@RequestParam(value = "medioBarcode", required = false) String medioBarcode, Pageable pageable) {
		Page<AssociatedFiles> page = associatedFilesService.findByParamsmediobarcode(medioBarcode, pageable);
		if (page.getContent().isEmpty()) {
			return new ResponseEntity(HttpStatus.NO_CONTENT);
			}
		return new ResponseEntity<Page<AssociatedFiles>>(page, HttpStatus.OK);
	}
}
```

**Certificado**

**Controlador** | CertificadoGeneralController.java

```
package com.halliburton.qc.controller;
import java.util.*;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.util.UriComponentsBuilder;
import org.springframework.data.domain.Pageable;
import com.halliburton.qc.entities.*;
import com.halliburton.qc.service.*;
import com.halliburton.qc.util.CustomErrorType;
@RestController
public class CertificadoGeneralController {
        public static final Logger logger = LoggerFactory.getLogger(CertificadoGeneralController.class);
        @Autowired
        CertificadoGeneralService certificadogeneralService;
```

**Listar Certificados**

```
@CrossOrigin(origins = "*", maxAge = 3600)
@RequestMapping(value = "/certificadogeneral/", method = RequestMethod.GET)
  public ResponseEntity<Page<CertificadoGeneral>> listAllCertificadoGeneral(Pageable pageable) {
        Page<CertificadoGeneral> page = certificadogeneralService.findAll(pageable);
            if (page.getContent().isEmpty()) {
                return new ResponseEntity(HttpStatus.NO_CONTENT);       }
            return new ResponseEntity<Page<CertificadoGeneral>>(page, HttpStatus.OK);         }
```

**Listar Certificados por Orden de Trabajo**

```
  @CrossOrigin(origins = "*", maxAge = 3600)
  @RequestMapping(value = "/certificadogeneral/{id}", method = RequestMethod.GET)
        public ResponseEntity<?> getCertificadoGeneral(@PathVariable("id") String id) {
            logger.info("Fetching certificadogeneral with id {}", id);
            Optional<CertificadoGeneral> certificado = certificadogeneralService.findById(id);
            if (!certificado.isPresent()) {
                logger.error("Orderfiles with id {} not found.", id);
                return new ResponseEntity(new CustomErrorType("certificadogeneral with id "+ id+" not found"), HttpStatus.NOT_FOUND);}
            return new ResponseEntity<CertificadoGeneral>(certificado.get(), HttpStatus.OK);    }
```

**Agregar Certificados**

```
  @CrossOrigin(origins = "*", maxAge = 3600)
  @RequestMapping(value = "/certificadogeneral/", method = RequestMethod.POST)
  public ResponseEntity<?> createCertificadoGeneral(@RequestBody CertificadoGeneral certificado, UriComponentsBuilder ucBuilder) {
            logger.info("Creating certificadogeneral : {}", certificado);
            certificadogeneralService.save(certificado);
            logger.info("Get OrderId certificadogeneral : {}", certificado.getOrderId());
            String orderId = certificado.getOrderId();
            Pageable pageable = null;
            Page<CertificadoGeneral> page = certificadogeneralService.findByParamsorderId(orderId, pageable);
            if (page.getContent().isEmpty()) {
                return new ResponseEntity(HttpStatus.NO_CONTENT);          }
            return new ResponseEntity<Page<CertificadoGeneral>>(page, HttpStatus.OK);         }
```

**Actualizar Certificado**

```
  @CrossOrigin(origins = "*", maxAge = 3600)
  @RequestMapping(value = "/certificadogeneral/", method = RequestMethod.POST)
  public ResponseEntity<?> createCertificadoGeneral(@RequestBody CertificadoGeneral certificado, UriComponentsBuilder ucBuilder) {
            logger.info("Creating certificadogeneral : {}", certificado);
            certificadogeneralService.save(certificado);
            logger.info("Get OrderId certificadogeneral : {}", certificado.getOrderId());
            String orderId = certificado.getOrderId();
            Pageable pageable = null;
            Page<CertificadoGeneral> page = certificadogeneralService.findByParamsorderId(orderId, pageable);
            if (page.getContent().isEmpty()) {
                return new ResponseEntity(HttpStatus.NO_CONTENT);          }
            return new ResponseEntity<Page<CertificadoGeneral>>(page, HttpStatus.OK);         }
}
```

- ✓ Repository

| Orden de Trabajo | |
|---|---|
| **Repository** | OrderRepository.java |

```
package com.halliburton.qc.repository;
import java.util.Calendar;
import java.util.Date;
import java.util.Iterator;
import java.util.List;
import java.util.Optional;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.querydsl.QuerydslPredicateExecutor;
import org.springframework.data.querydsl.binding.QuerydslBinderCustomizer;
import org.springframework.data.querydsl.binding.QuerydslBindings;
import org.bitbucket.gt_tech.spring.data.querydsl.value.operators.ExpressionProviderFactory;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.repository.query.Param;
import org.springframework.data.rest.core.annotation.RepositoryRestResource;
import com.halliburton.qc.entities.*;
import com.querydsl.core.types.dsl.BooleanExpression;

@RepositoryRestResource
public interface OrdersRepository
                extends JpaRepository<Orders, String>, QuerydslPredicateExecutor<Orders>, QuerydslBinderCustomizer<QOrders> {

        @Override
        default public void customize(QuerydslBindings bindings, QOrders root) {
                bindings.bind(root.memo).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                bindings.bind(root.id).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                bindings.bind(root.comments).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                bindings.bind(root.commentsTask).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                // bindings.bind(root.datecreate).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                bindings.bind(root.datecreate).all((path, values) -> {
                        Iterator<? extends Date> it = values.iterator();
                        if ( values.size() == 1 ) {
                                Date begin = it.next();
                                Calendar ini = Calendar.getInstance();
                                ini.setTime(begin);
                                ini.set(Calendar.HOUR, ini.get(Calendar.HOUR)+ 0);
                                Date iniDate = ini.getTime();
                                Calendar fin = Calendar.getInstance();
                                fin.setTime(begin);
                                fin.set(Calendar.HOUR, fin.get(Calendar.HOUR)+ 23);
                                Date finDate = fin.getTime();
                                return Optional.ofNullable(path.between( iniDate.before(finDate) ? iniDate : finDate,
iniDate.after(finDate) ? iniDate : finDate));
                                }else if ( values.size() == 2 ) {
                                Date first = it.next();
                                Date second = it.next();
                                return Optional.ofNullable(path.between( first.before(second) ? first : second, first.after(second) ? first :
second));
                        }
                        else {
                                BooleanExpression e = path.eq(it.next());
                                while ( it.hasNext() && e != null ) {
                                        e = e.or(path.eq(it.next()));
                                }
                                return Optional.ofNullable(e);
                        }
                });
                bindings.bind(root.datedue).all((path, values) -> {
                        Iterator<? extends Date> it = values.iterator();
                        if ( values.size() == 1 ) {
                                Date begin = it.next();
                                Calendar ini = Calendar.getInstance();
                                ini.setTime(begin);
                                ini.set(Calendar.HOUR, ini.get(Calendar.HOUR)+ 0);
                                Date iniDate = ini.getTime();
```

```
                                        fin.setTime(begin);
                                        fin.set(Calendar.HOUR, fin.get(Calendar.HOUR)+ 23);
                                        Date finDate = fin.getTime();

                                        return Optional.ofNullable(path.between( iniDate.before(finDate) ? iniDate : finDate,
iniDate.after(finDate) ? iniDate : finDate));
                                        // return Optional.ofNullable(path.gt(iniDate).and(path.lt(finDate)));
                                }else if ( values.size() == 2 ) {
                                        Date first = it.next();
                                        Date second = it.next();
                                        return Optional.ofNullable(path.between( first.before(second) ? first : second, first.after(second) ? first :
second));
                                }
                                else {
                                        BooleanExpression e = path.eq(it.next());
                                        while ( it.hasNext() && e != null ) {
                                                e = e.or(path.eq(it.next()));
                                        }
                                        return Optional.ofNullable(e);
                                }
                        });
                        bindings.bind(root.management).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.observation).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.origin).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.registerby).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.requestedby).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.signedby).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.state).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.subject).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.substate).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.typeot).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.code).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.dateUpdate).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.userupdate).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.editCatalogo).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.editDocument).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.editWell).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.editSeismic).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.editGis).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.typeotnew).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.detalle).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.statetaskReception).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.statetaskWell).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.statetaskArchive).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.statetaskGis).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.statetaskSeismic).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.statetaskAttention).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
                        bindings.bind(root.mesOt).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));

        }
        @Query("select distinct o from Orders o order by o.datecreate DESC, o.code DESC")
        public Page<Orders> findAllByOrders(Pageable pageable);

        @Query("select distinct o from Orders o " + "inner join o.ordersGroups og " + "where o.id IN (:idsOrder) AND og.groupId in (:ids) "
                        + "order by o.datecreate DESC, o.code DESC")
        public Page<Orders> findOrdersByListIdsGroup(@Param("ids") List<String> ids, @Param("idsOrder") List<String> idsOrder, Pageable
pageable);

@Query("select distinct o from Orders o " + "inner join o.tasks t " + "where o.id IN (:idsOrder) AND t.ownerTask =:ownerTask "

                        + "order by o.datecreate DESC, o.code DESC")
        public Page<Orders> findOrdersByListIdOwner(@Param("ownerTask") String ownerTask, @Param("idsOrder") List<String> idsOrder,
Pageable pageable);

        @Query("select hb from Orders  hb where " + "(hb.code = :code  and hb.registerby = :registerby  ) or "
                        + "(hb.code = :code  and hb.requestedby = :requestedby  ) or "
                        + "(hb.code = :code  and hb.subject = :subject  ) or "
                        + "(hb.code = :code  and hb.observation = :observation  ) or "
                        + "(hb.code = :code  and hb.signedby = :signedby  ) or "
                        + "(hb.code = :code  ) or "
```

```
                                + "(hb.registerby = :registerby  and hb.requestedby = :requestedby  ) or "
                                + "(hb.registerby = :registerby  and hb.subject = :subject  ) or "
                                + "(hb.registerby = :registerby  and hb.observation = :observation  ) or "
                                + "(hb.registerby = :registerby  and hb.signedby = :signedby  ) or "
                                + "(hb.registerby = :registerby  ) or "

                                + "(hb.requestedby = :requestedby  ) or "
                                + "(hb.requestedby = :requestedby  and hb.subject = :subject  ) or "
                                + "(hb.requestedby = :requestedby  and hb.observation = :observation  ) or "
                                + "(hb.requestedby = :requestedby  and hb.signedby = :signedby  ) or "
                                + "(hb.subject = :subject  ) or "

                                + "(hb.subject = :subject  and hb.observation = :observation  ) or "
                                + "(hb.subject = :subject  and hb.signedby = :signedby  ) or "
                                + "(hb.observation = :observation  ) or "

                                + "(hb.observation = :observation  and hb.signedby = :signedby  ) or " + "(hb.signedby = :signedby  ) or "

                                + "(hb.code = :code  and hb.registerby = :registerby  and hb.requestedby = :requestedby  ) or "
                                + "(hb.code = :code  and hb.registerby = :registerby  and hb.requestedby = :requestedby  and hb.subject = :subject  )
or "
                                + "(hb.code = :code  and hb.registerby = :registerby  and hb.requestedby = :requestedby  and hb.subject = :subject
and hb.observation = :observation  ) or "
                                + "(hb.code = :code  and hb.registerby = :registerby  and hb.requestedby = :requestedby  and hb.subject = :subject
and hb.observation = :observation  and hb.signedby = :signedby  ) or "
                                + "(hb.registerby = :registerby  and hb.requestedby = :requestedby  and hb.subject = :subject  ) or "
                                + "(hb.registerby = :registerby  and hb.requestedby = :requestedby  and hb.subject = :subject  and hb.observation =
:observation  ) or "
                                + "(hb.registerby = :registerby  and hb.requestedby = :requestedby  and hb.subject = :subject  and hb.observation =
:observation  and hb.signedby = :signedby  ) or "
                                + "(hb.requestedby = :requestedby  and hb.subject = :subject  and hb.observation = :observation  ) or "
                                + "(hb.requestedby = :requestedby  and hb.subject = :subject  and hb.observation = :observation  and hb.signedby =
:signedby  ) or "
                                + "(hb.subject = :subject  and hb.observation = :observation  and hb.signedby = :signedby )"
                                + "order by hb.datecreate DESC, hb.code DESC")


public Page<Orders> findByParams(@Param("code") String code, @Param("registerby") String registerby,
                                @Param("requestedby") String requestedby, @Param("subject") String subject,

@Param("observation") String observation, @Param("signedby") String signedby, Pageable pageable);
@Query("select coalesce(max(ord.code), 0) from Orders ord where year(ord.datecreate)=:datecreate and ord.typeot=:typeot ")
        public Integer findMaxCodeByDateCreateAndTypeOf(@Param("datecreate") Integer datecreate,
                                @Param("typeot") String typeot);

        @Query("select hb from Orders  hb where" + "(hb.state like %:state%)")
        public Page<Orders> findfechaOT(@Param("state") String state, Pageable pageable);

        @Query("select hb from Orders hb where" + "(year(hb.datecreate)=:datecreate)")
        public Integer findfechaROT2(@Param("datecreate") Integer datecreate);
}
```

| Tarea | |
|---|---|
| **Repository** | TareaRepository.java |

```
package com.halliburton.qc.repository;
import java.util.Date;
import java.util.List;
import java.util.Set;
import org.springframework.data.jpa.repository.EntityGraph;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.querydsl.QuerydslPredicateExecutor;
import org.springframework.data.querydsl.binding.QuerydslBinderCustomizer;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.repository.query.Param;
import org.springframework.data.rest.core.annotation.RepositoryRestResource;
import com.halliburton.qc.entities.*;

@RepositoryRestResource
 public interface TasksRepository extends JpaRepository<Tasks, String>, QuerydslPredicateExecutor<Tasks>{
            @Query("select hb from Tasks  hb where "
             + "(hb.comments = :comments  and hb.orderId = :orderId  )")

             public Page<Tasks> findByParamsCommentsAndCorderId(
                        @Param("comments") String  comments,
                        @Param("orderId") String  orderId,
                        Pageable pageable);
             @Query("select hb from Tasks  hb where " + "(hb.orderId = :orderId   )")

             public Page<Tasks> findByParamsOrderId(@Param("orderId") String  orderId,
                        Pageable pageable);
             @Query("select hb from Tasks  hb where " + "(hb.comments = :comments   )")

             public Page<Tasks> findByParamsComments(
                        @Param("comments") String  comments,
                        Pageable pageable);
             @Query("select COALESCE(avg(tk.porcent),0) from Tasks tk where tk.orderId=:orderId ")
             public Integer findTasksAvgyByOrderId(@Param("orderId")String orderId);
             @Query("select NEW com.halliburton.qc.entities.Tasks("                                    +
"t.porcent,o.code,o.datecreate,o.datedue,g.grupoNombre,ag.actividadNombre,userOwn.usuarioNombre,userOwn.usuarioApellido,t.order, o.id"
                        + ") from Tasks t "
                                    + "inner join t.orders o "
                                    + "inner join t.usuarioOwnerTask userOwn "
                                    + "inner join t.grouptarea gt "
                                    + "inner join gt.funcional g "
                                    + "inner join gt.actividadGruFuncional ag "
                                    + "where userOwn.id in (:listIds) and t.status = 0 "
                                    + "order by o.datecreate DESC, o.code DESC")
             public List<Tasks> findTasksDetailByGroups(@Param("listIds") List<String> listIds);
             @Query("select NEW com.halliburton.qc.entities.Tasks("          +
"t.porcent,o.code,o.datecreate,o.datedue,g.grupoNombre,ag.actividadNombre,userOwn.usuarioNombre,userOwn.usuarioApellido,t.order, o.id"
                        + ") from Tasks t "
                                    + "inner join t.orders o "
                                    + "inner join t.usuarioOwnerTask userOwn "
                                    + "inner join t.grouptarea gt "
                                    + "inner join gt.funcional g "
                                    + "inner join gt.actividadGruFuncional ag "
                                    + "where t.status = 0 "
                                    + "order by o.datecreate DESC, o.code DESC")
             public List<Tasks> findTasksDetailAll();
}
```

| Metadata | |
|---|---|
| **Repository** | AssociateFileRepository.java |

```
package com.halliburton.qc.repository;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.querydsl.QuerydslPredicateExecutor;
import org.springframework.data.querydsl.binding.QuerydslBinderCustomizer;
import org.springframework.data.querydsl.binding.QuerydslBindings;
import java.util.Calendar;
import java.util.Date;
import java.util.Iterator;
import java.util.Optional;
import org.bitbucket.gt_tech.spring.data.querydsl.value.operators.ExpressionProviderFactory;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.repository.query.Param;
import org.springframework.data.rest.core.annotation.RepositoryRestResource;
import com.halliburton.qc.entities.*;
import com.querydsl.core.types.dsl.BooleanExpression;

@RepositoryRestResource
public interface AssociatedFilesRepository extends JpaRepository<AssociatedFiles, String>, QuerydslPredicateExecutor<AssociatedFiles>,
QuerydslBinderCustomizer<QAssociatedFiles>{
@Override
default public void customize(QuerydslBindings bindings, QAssociatedFiles root) {
bindings.bind(QAssociatedFiles.associatedFiles.id).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.barcode).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.cuenca).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.dateCreate).all((path, values) -> {
                                Iterator<? extends Date> it = values.iterator();
                                if ( values.size() == 1 ) {
                                                Date begin = it.next();
                                                Calendar ini = Calendar.getInstance();
                                                ini.setTime(begin);
                                                ini.set(Calendar.HOUR, ini.get(Calendar.HOUR)+ 0);
                                                Date iniDate = ini.getTime();

                                                Calendar fin = Calendar.getInstance();
                                                fin.setTime(begin);
                                                fin.set(Calendar.HOUR, fin.get(Calendar.HOUR)+ 0);
                                                Date finDate = fin.getTime();

return Optional.ofNullable(path.between( iniDate.before(finDate) ? iniDate : finDate, iniDate.after(finDate) ? iniDate : finDate));
                                }else {
                                                BooleanExpression e = path.eq(it.next());
                                                while ( it.hasNext() && e != null ) {
                                                                e = e.or(path.eq(it.next()));
                                                }
                                                return Optional.ofNullable(e);
                                }
                });
bindings.bind(QAssociatedFiles.associatedFiles.dateDelete).all((path, values) ->{
                                Iterator<? extends Date> it = values.iterator();
                                if ( values.size() == 1 ) {
                                                Date begin = it.next();
                                                Calendar ini = Calendar.getInstance();
                                                ini.setTime(begin);
                                                ini.set(Calendar.HOUR, ini.get(Calendar.HOUR)+ 0);
                                                Date iniDate = ini.getTime();
                                                Calendar fin = Calendar.getInstance();
                                                fin.setTime(begin);
                                                fin.set(Calendar.HOUR, fin.get(Calendar.HOUR)+ 0);
                                                Date finDate = fin.getTime();
return Optional.ofNullable(path.between( iniDate.before(finDate) ? iniDate : finDate, iniDate.after(finDate) ? iniDate : finDate));
                                }else {
                                                BooleanExpression e = path.eq(it.next());
                                                while ( it.hasNext() && e != null ) {
                                                                e = e.or(path.eq(it.next()));
                                                }
                                                return Optional.ofNullable(e);
```

```
                }
        });
bindings.bind(QAssociatedFiles.associatedFiles.dateUpdate).all((path, values) ->{
                Iterator<? extends Date> it = values.iterator();
                if ( values.size() == 1 ) {
                        Date begin = it.next();
                        Calendar ini = Calendar.getInstance();
                        ini.setTime(begin);
                        ini.set(Calendar.HOUR, ini.get(Calendar.HOUR)+ 0);
                        Date iniDate = ini.getTime();
                        Calendar fin = Calendar.getInstance();
                        fin.setTime(begin);
                        fin.set(Calendar.HOUR, fin.get(Calendar.HOUR)+ 0);
                        Date finDate = fin.getTime();
return Optional.ofNullable(path.between( iniDate.before(finDate) ? iniDate : finDate, iniDate.after(finDate) ? iniDate : finDate));
                }else {
                        BooleanExpression e = path.eq(it.next());
                        while ( it.hasNext() && e != null ) {
                                e = e.or(path.eq(it.next()));
                        }
                        return Optional.ofNullable(e);
                }
        });
bindings.bind(QAssociatedFiles.associatedFiles.escala).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.ext).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.ilustration).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.institutional).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.lote).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.medio).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.name).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.orderId).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.path).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.region).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.subtitle).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.themedescriptor).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.type).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.typebarcode).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.userCreate).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.userDelete).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.userUpdate).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.uwi).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.validot).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.validqc).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.yacimiento).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.medioBarcode).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.dataType).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
                bindings.bind(QAssociatedFiles.associatedFiles.certiDate).all((path, values) ->{
                Iterator<? extends Date> it = values.iterator();
                if ( values.size() == 1 ) {
                        Date begin = it.next();
                        Calendar ini = Calendar.getInstance();
                        ini.setTime(begin);
                        ini.set(Calendar.HOUR, ini.get(Calendar.HOUR)+ 0);
                        Date iniDate = ini.getTime();
                        Calendar fin = Calendar.getInstance();
                        fin.setTime(begin);
                        fin.set(Calendar.HOUR, fin.get(Calendar.HOUR)+ 23);
                        Date finDate = fin.getTime();
                        return Optional.ofNullable(path.between( iniDate.before(finDate) ? iniDate : finDate,
iniDate.after(finDate) ? iniDate : finDate));
                        }else {
                        BooleanExpression e = path.eq(it.next());
                        while ( it.hasNext() && e != null ) {
                                e = e.or(path.eq(it.next()));
                        }
                        return Optional.ofNullable(e);
                }
        });
bindings.bind(QAssociatedFiles.associatedFiles.titleDocument).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.commentDoc).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
```

```
bindings.bind(QAssociatedFiles.associatedFiles.docuDate).all((path, values) ->{
                Iterator<? extends Date> it = values.iterator();
                        if ( values.size() == 1 ) {
                                Date begin = it.next();
                                Calendar ini = Calendar.getInstance();
                                ini.setTime(begin);
                                ini.set(Calendar.HOUR, ini.get(Calendar.HOUR)+ 0);
                                Date iniDate = ini.getTime();

                                Calendar fin = Calendar.getInstance();
                                fin.setTime(begin);
                                fin.set(Calendar.HOUR, fin.get(Calendar.HOUR)+ 23);
                                Date finDate = fin.getTime();

                                return Optional.ofNullable(path.between( iniDate.before(finDate) ? iniDate : finDate,
iniDate.after(finDate) ? iniDate : finDate));
                                // return Optional.ofNullable(path.gt(iniDate).and(path.lt(finDate)));
                        }else {
                                BooleanExpression e = path.eq(it.next());
                                while ( it.hasNext() && e != null ) {
                                        e = e.or(path.eq(it.next()));
                                }
                                return Optional.ofNullable(e);
                        }
                });
bindings.bind(QAssociatedFiles.associatedFiles.materia).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.infoTecnic).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.ellipsoid).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.projection).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.zone).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.datum).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.commentDocgis).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.commentDocwell).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.commentDocseis).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.commentDoctec).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.latSouthesi).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.content).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.summary).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.logName).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.logType).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.logService).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.logSource).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.logginContractor).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.logDate).all((path, values) ->{
                Iterator<? extends Date> it = values.iterator();
                if ( values.size() == 1 ) {
                        Date begin = it.next();
                        Calendar ini = Calendar.getInstance();
                        ini.setTime(begin);
                        ini.set(Calendar.HOUR, ini.get(Calendar.HOUR)+ 0);
                        Date iniDate = ini.getTime();

                        Calendar fin = Calendar.getInstance();
                        fin.setTime(begin);
                        fin.set(Calendar.HOUR, fin.get(Calendar.HOUR)+ 23);
                        Date finDate = fin.getTime();
                        return Optional.ofNullable(path.between( iniDate.before(finDate) ? iniDate : finDate,
iniDate.after(finDate) ? iniDate : finDate));
                        }else {
                        BooleanExpression e = path.eq(it.next());
                        while ( it.hasNext() && e != null ) {
                                e = e.or(path.eq(it.next()));
                        }
                        return Optional.ofNullable(e);
                }
        });
bindings.bind(QAssociatedFiles.associatedFiles.campaingName).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.reportCode).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.imageAssociated).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
                bindings.bind(QAssociatedFiles.associatedFiles.dateProcess).all((path, values) ->{
```

```
                                Iterator<? extends Date> it = values.iterator();
                                if ( values.size() == 1 ) {
                                        Date begin = it.next();
                                        Calendar ini = Calendar.getInstance();
                                        ini.setTime(begin);
                                        ini.set(Calendar.HOUR, ini.get(Calendar.HOUR)+ 0);
                                        Date iniDate = ini.getTime();

                                        Calendar fin = Calendar.getInstance();
                                        fin.setTime(begin);
                                        fin.set(Calendar.HOUR, fin.get(Calendar.HOUR)+ 23);
                                        Date finDate = fin.getTime();

                                        return Optional.ofNullable(path.between( iniDate.before(finDate) ? iniDate : finDate,
iniDate.after(finDate) ? iniDate : finDate));
                                        // return Optional.ofNullable(path.gt(iniDate).and(path.lt(finDate)));
                                }else {
                                        BooleanExpression e = path.eq(it.next());
                                        while ( it.hasNext() && e != null ) {
                                                e = e.or(path.eq(it.next()));
                                        }
                                        return Optional.ofNullable(e);
                                }
                        });
bindings.bind(QAssociatedFiles.associatedFiles.processTo).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.processBy).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.processType).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.nameLine).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.typeSeismic).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.curveTypes).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.numeLineas3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.formatoGrabacion3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.medioGrabacion3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.canalNominal3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.canalAuxiliar3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.areaTotal3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.sistemaCoorde3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.reporteTopo3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.reporteOperaciones3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path,
values));
bindings.bind(QAssociatedFiles.associatedFiles.fichaSismica3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.disparoInicial3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.disparoFinal3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.campoInicial3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.campoFinal3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.reporteObservador3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.longitudRegistro3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.intervaloMuestreo3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.cantidadDisparo3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.coordenadasDR3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.comentario3DC).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.analisisVelocidad).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.informeVerificacion).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.longitudLinea).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.nrocrucesLinea).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.intervaloCDPS).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozouwi).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozoequipo).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozoproftotal).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozotipo).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozocromat).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozofechainicio).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozofechacomplet).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozoreferencia).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozoelevacion).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozoparametros).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozosurvey).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozocoordenadas).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozozona).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozocoordsupern).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
```

```
bindings.bind(QAssociatedFiles.associatedFiles.pozocoordsupere).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozocoordfondn).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozocoordfonde).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.gisfecha).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.docgisFecha).all((path, values) ->{
                                Iterator<? extends Date> it = values.iterator();
                                if ( values.size() == 1 ) {
                                                Date begin = it.next();
                                                Calendar ini = Calendar.getInstance();
                                                ini.setTime(begin);
                                                ini.set(Calendar.HOUR, ini.get(Calendar.HOUR)+ 0);
                                                Date iniDate = ini.getTime();

                                                Calendar fin = Calendar.getInstance();
                                                fin.setTime(begin);
                                                fin.set(Calendar.HOUR, fin.get(Calendar.HOUR)+ 23);
                                                Date finDate = fin.getTime();

                                                return Optional.ofNullable(path.between( iniDate.before(finDate) ? iniDate : finDate,
iniDate.after(finDate) ? iniDate : finDate));
                                }else {
                                                BooleanExpression e = path.eq(it.next());
                                                while ( it.hasNext() && e != null ) {
                                                                e = e.or(path.eq(it.next()));
                                                }
                                                return Optional.ofNullable(e);
                                }
                });
bindings.bind(QAssociatedFiles.associatedFiles.fechacompletPozo).all((path, values) ->{
                                Iterator<? extends Date> it = values.iterator();
                                if ( values.size() == 1 ) {
                                                Date begin = it.next();
                                                Calendar ini = Calendar.getInstance();
                                                ini.setTime(begin);
                                                ini.set(Calendar.HOUR, ini.get(Calendar.HOUR)+ 0);
                                                Date iniDate = ini.getTime();
                                                Calendar fin = Calendar.getInstance();
                                                fin.setTime(begin);
                                                fin.set(Calendar.HOUR, fin.get(Calendar.HOUR)+ 23);
                                                Date finDate = fin.getTime();
                                                return Optional.ofNullable(path.between( iniDate.before(finDate) ? iniDate : finDate,
iniDate.after(finDate) ? iniDate : finDate));
                                }else {
                                                BooleanExpression e = path.eq(it.next());
                                                while ( it.hasNext() && e != null ) {
                                                                e = e.or(path.eq(it.next()));
                                                }
                                                return Optional.ofNullable(e);
                                }
                });
bindings.bind(QAssociatedFiles.associatedFiles.doctecFecha).all((path, values) ->{
                                Iterator<? extends Date> it = values.iterator();
                                if ( values.size() == 1 ) {
                                                Date begin = it.next();
                                                Calendar ini = Calendar.getInstance();
                                                ini.setTime(begin);
                                                ini.set(Calendar.HOUR, ini.get(Calendar.HOUR)+ 0);
                                                Date iniDate = ini.getTime();
                                                Calendar fin = Calendar.getInstance();
                                                fin.setTime(begin);
                                                fin.set(Calendar.HOUR, fin.get(Calendar.HOUR)+ 23);
                                                Date finDate = fin.getTime();
                                                return Optional.ofNullable(path.between( iniDate.before(finDate) ? iniDate : finDate,
iniDate.after(finDate) ? iniDate : finDate));
                                }else {
                                                BooleanExpression e = path.eq(it.next());
                                                while ( it.hasNext() && e != null ) {
                                                                e = e.or(path.eq(it.next()));
                                                }
                                                return Optional.ofNullable(e);
```

```
                }
        });
bindings.bind(QAssociatedFiles.associatedFiles.fechainicioPozo).all((path, values) ->{
                Iterator<? extends Date> it = values.iterator();
                if ( values.size() == 1 ) {
                        Date begin = it.next();
                        Calendar ini = Calendar.getInstance();
                        ini.setTime(begin);
                        ini.set(Calendar.HOUR, ini.get(Calendar.HOUR)+ 0);
                        Date iniDate = ini.getTime();

                        Calendar fin = Calendar.getInstance();
                        fin.setTime(begin);
                        fin.set(Calendar.HOUR, fin.get(Calendar.HOUR)+ 23);
                        Date finDate = fin.getTime();

                        return Optional.ofNullable(path.between( iniDate.before(finDate) ? iniDate : finDate,
iniDate.after(finDate) ? iniDate : finDate));
                        }else {
                        BooleanExpression e = path.eq(it.next());
                        while ( it.hasNext() && e != null ) {
                                e = e.or(path.eq(it.next()));
                        }
                        return Optional.ofNullable(e);
                }
        });
bindings.bind(QAssociatedFiles.associatedFiles.typeWell).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozomuestraRecipiente).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozomuestraCantidad).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozomuestraTipo).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozomuestraFormacion).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozomuestraTope).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozomuestraBase).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozoInformes).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozoperfilEscaneado).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozoduplicidadImagenes).all((path, values) ->ExpressionProviderFactory.getPredicate(path,
values));
bindings.bind(QAssociatedFiles.associatedFiles.pozocontieneInscripciones).all((path, values) ->ExpressionProviderFactory.getPredicate(path,
values));
bindings.bind(QAssociatedFiles.associatedFiles.pozocabezalcoincidePerfil).all((path, values) ->ExpressionProviderFactory.getPredicate(path,
values));
bindings.bind(QAssociatedFiles.associatedFiles.pozoarchivocurvasImpreso).all((path, values) ->ExpressionProviderFactory.getPredicate(path,
values));
bindings.bind(QAssociatedFiles.associatedFiles.pozoarchivoperfilPapel).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozoconvertidoTif).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.pozoperfilesImpresos).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.sismicanroCruces).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.gismedioOriginal).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.gisestadomapaDigital).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.gisestadomapaImpreso).all((path, values) -> ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.gismedioReferencia).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.gistipoMapa).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.gisescaneomapaImpreso).all((path, values) ->ExpressionProviderFactory.getPredicate(path,
values));
bindings.bind(QAssociatedFiles.associatedFiles.gisinformacionHistorica).all((path, values) ->ExpressionProviderFactory.getPredicate(path,
values));
bindings.bind(QAssociatedFiles.associatedFiles.archivoselectCatalogo).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.archivoimpresoCalidad).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.archivoescaneoinfoFaltante).all((path, values) ->ExpressionProviderFactory.getPredicate(path,
values));
bindings.bind(QAssociatedFiles.associatedFiles.archivoconversionPdf).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.archivopdfcoincideImpreso).all((path, values) ->ExpressionProviderFactory.getPredicate(path,
values));
bindings.bind(QAssociatedFiles.associatedFiles.pozocorrelativoBarcode).all((path, values) ->ExpressionProviderFactory.getPredicate(path, values));
bindings.bind(QAssociatedFiles.associatedFiles.docsismicaprocesadaFecha).all((path, values) ->{
                Iterator<? extends Date> it = values.iterator();
                if ( values.size() == 1 ) {
                        Date begin = it.next();
                        Calendar ini = Calendar.getInstance();
                        ini.setTime(begin);
```

```
                                        ini.set(Calendar.HOUR, ini.get(Calendar.HOUR)+ 0);
                                        Date iniDate = ini.getTime();

                                        Calendar fin = Calendar.getInstance();
                                        fin.setTime(begin);
                                        fin.set(Calendar.HOUR, fin.get(Calendar.HOUR)+ 23);
                                        Date finDate = fin.getTime();

                                        return Optional.ofNullable(path.between( iniDate.before(finDate) ? iniDate : finDate,
iniDate.after(finDate) ? iniDate : finDate));
                                        // return Optional.ofNullable(path.gt(iniDate).and(path.lt(finDate)));
                                }else {
                                        BooleanExpression e = path.eq(it.next());
                                        while ( it.hasNext() && e != null ) {
                                                e = e.or(path.eq(it.next()));
                                        }
                                        return Optional.ofNullable(e);
                                }
                        });
        }
        @Query("select hb from AssociatedFiles  hb where "
      + "(hb.orderId = :orderId  and  hb.status = 0 )")
                public Page<AssociatedFiles> findByParams(
                        @Param("orderId") String  orderId,
                        Pageable pageable);
        @Query("select hb from AssociatedFiles  hb where"
                        + "(hb.typebarcode like %:typeBarcode%  and hb.barcode like %:barcoden% and hb.status = 0)")
        public Page<AssociatedFiles> findByParamstypebarcode(
                        @Param("typeBarcode") String  typeBarcode,
                        @Param("barcoden") String  barcoden,
                        Pageable pageable);
        @Query("select hb from AssociatedFiles  hb where"
                        + "(hb.typebarcode = :typeBarcode  and hb.barcode = :barcoden and hb.status = 0)")
        public Page<AssociatedFiles> findByParamstypebarcodeequals(
                        @Param("typeBarcode") String  typeBarcode,
                        @Param("barcoden") String  barcoden,
                        Pageable pageable);
        @Query("select hb from AssociatedFiles  hb where"
                        + "(hb.medioBarcode like %:medioBarcode%  and hb.status = 0)")
        public Page<AssociatedFiles> findByParamsmediobarcode(
                        @Param("medioBarcode") String  medioBarcode,
                        Pageable pageable);
        @Query("select hb from AssociatedFiles  hb where"
                        + "(hb.orderId = :orderId  and hb.status = 0)")
        public Page<AssociatedFiles> findByParamsfind_associatedfile_orderid(
                        @Param("orderId") String  orderId,
                        Pageable pageable);

}
```

| Certificado | |
|---|---|
| **Repository** | CertificadoGeneralRepository.java |

```
package com.halliburton.qc.repository;

import java.util.Set;
import org.springframework.data.jpa.repository.EntityGraph;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.repository.query.Param;
import org.springframework.data.rest.core.annotation.RepositoryRestResource;
import com.halliburton.qc.entities.*;

@RepositoryRestResource
public interface CertificadoGeneralRepository extends JpaRepository<CertificadoGeneral, String>{

	@Query("select hb from CertificadoGeneral  hb where "
				+ "(hb.orderId = :orderId  )")
	public Page<CertificadoGeneral> findByParamsorderId(
			@Param("orderId") String  orderId,
			Pageable pageable);

	@Query("select hb from CertificadoGeneral  hb where"
				+ "(hb.orderId = :orderId)")
	public Page<CertificadoGeneral> findByParamscertificadogeneralorderid(
				@Param("orderId") String  orderId,
				Pageable pageable);
}
```