

"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

**INSTITUTO TECNOLÓGICO DE VILLAHERMOSA
DIVISION DE ESTUDIOS DE POSGRADO E INVESTIGACION**

**“Plataforma Tecnológica para Explotación de
Información-Técnica Petrolera. Grupo CASE”**

TESIS

PARA OBTENER EL GRADO DE:

MAESTRIA EN TECNOLOGIAS DE LA INFORMACION

PRESENTA:

RODRIGO SANCHEZ PEREGRINO

DIRECTOR DE TESIS

MTC. MIGUEL PEREZ VASCONCELOS

VILLAHERMOSA, TAB.

NOVIEMBRE 2020





“2020, Año de Leona Vicario, Benemérita Madre de la Patria”

Villahermosa, Tabasco, 15/diciembre/2020

DEPARTAMENTO: División de Estudios de Posgrado e Investigación

NO. DE OFICIO: DEPI/331/2020

ASUNTO: AUTORIZACIÓN DE IMPRESIÓN

**C. RODRIGO SÁNCHEZ PEREGRINO
ESTUDIANTE DE LA MAESTRÍA EN
TECNOLOGÍAS DE LA INFORMACION
PRESENTE**

De acuerdo con el fallo emitido por la comisión revisora integrada por los CC.: **M.C.T. Miguel Pérez Vasconcelos, M.S.C Eutimio Sosa Silva, Dra. Rosa Gómez Domínguez y M.G.T.I Fidelio Castillo Romero**, y considerando que cubre todos los requisitos del Reglamento de Titulación en vigor, se da a usted la autorización para que proceda a imprimir su trabajo profesional titulado:

“Plataforma Tecnológica para Explotación de Información-Técnica Petrolera. Grupo CASE.”

Hago de su conocimiento lo anterior para los efectos y fines correspondientes.

A T E N T A M E N T E

*Excelencia en Educación Tecnológica.
Tierra, Tiempo, Trabajo y Tecnología.*



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE VILLAHERMOSA

DIVISION DE ESTUDIOS DE POSGRADO E INVESTIGACION

**MTI. MAXIMILIANO SANTIAGO PÉREZ
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**

c.c.p. Archivo
Dra. Rosa Gómez Domínguez – Coordinadora de la Maestría
MSP/rgd



CONTENIDO.

ABSTRACT	1
INTRODUCCIÓN	2
1. MARCO REFERENCIAL	3
1.1 ANTECEDENTES	3
1.2 CON TEXTO DEL PROBLEMA	3
1.3 PLANTEAMIENTO DEL PROBLEMA	5
1.4 JUSTIFICACIÓN	6
1.5 OBJETIVOS	7
1.5.1 OBJETIVO GENERAL	7
1.5.2 OBJETIVOS PARTICULARES	7
1.6 DELIMITACIONES	7
1.7 INDUSTRIA PETROLERA	8
1.7.1 POZO PETROLERO	10
1.7.2 REFORMA ENERGÉTICA	11
1.8 PLATAFORMA TECNOLÓGICA	11
1.8.1 WEB API	12
2. ESTADO DEL ARTE	14
2.1 CONTECH	16
2.1.1 SCHNEIDER	16
3. DESARROLLO DEL PROYECTO	17
3.1 MODELO DE ANÁLISIS	17
3.1.1 RECOPIACIÓN DE REQUISITOS	18
A. REQUISITOS FUNCIONALES	18
B. REQUISITOS NO FUNCIONALES	18
3.1.2 METODOLOGÍA	19
3.1.3 TECNOLOGÍA	19
A. SQL SERVER	19
B. .NET CORE	20
C. ANGULAR	21
D. ESTRUCTURA DEL SOFTWARE	22
E. FACTIBILIDAD OPERACIONAL	22
F. RECURSOS HUMANOS	23
G. RECURSOS TECNOLÓGICOS	23
H. PLAN DE TRABAJO	24
3.1.4 MODELADO DEL ANÁLISIS	25
A. CASOS DE USOS	25
B. ESPECIFICACIONES DE CASOS DE USOS	28
3.1.5 MODELADO DE DATOS	29
A. MODELADO DE DATOS RELACIONAL EN BASE DE DATOS	30
B. ESQUEMAS	34
C. TABLAS	34
D. STORED PROCEDURES	37
3.2 MODELADO DE COMPONENTES	37
A. CLASES	38
B. SEGURIDAD	40
3.3 FRONT-END	41
A. COMPONENTES	41
B. SERVICIOS	42
C. SEGURIDAD	42

3.4 PRUEBAS.....	43
A. BASE DE DATOS.....	43
B. WEB API.....	49
C. FRONT-END.....	52
4. RESULTADO FINAL.....	54
CONCLUSIONES.....	57
BIBLIOGRAFÍAS.....	58

ÍNDICE DE FIGURAS.

FIGURA 1. GRAFICA DE NÚMERO DE USUARIOS DE SMARTPHONE EN LOS ÚLTIMOS AÑOS.....	4
FIGURA 2. ESQUEMA DE LA SITUACIÓN ACTUAL.....	6
FIGURA 3. ESQUEMA CONCEPTUAL DE LA PLATAFORMA.....	8
FIGURA 4. RESERVA DE PETRÓLEO EN EL MUNDO.....	9
FIGURA 5. LOGO DE CONTECH.....	16
FIGURA 6. LOGO DE SCHNEIDER ELECTRONIC.....	16
FIGURA 7. MODELO DE ANALISIS.....	17
FIGURA 8. ESTRUCTURA DEL PROYECTO.....	22
FIGURA 9. CRONOGRAMA DE ACTIVIDADES.....	24
FIGURA 10. CASO DE USO: ING. CAMPO.....	25
FIGURA 11. COMPORTAMIENTO DE ING. CAMPO.....	26
FIGURA 12. CASO DE USO: LABORATORIO.....	26
FIGURA 13. COMPORTAMIENTO DE LABORATORIO.....	27
FIGURA 14. CASO DE USO: GEOLOGÍA.....	27
FIGURA 15. COMPORTAMIENTO DE GEOLOGÍA.....	28
FIGURA 16. BASE DE DATOS PART.1 BASE DE DATOS PART2.....	30
FIGURA 17. BASE DE DATOS PART. 3.....	31
FIGURA 18. BASE DE DATOS PART 4.....	31
FIGURA 19. BASE DE DATOS PART. 5.....	32
FIGURA 20. BASE DE DATOS PART. 6.....	32
FIGURA 21. BASE DE DATOS PART. 7.....	33
FIGURA 22. BASE DE DATOS PART 8.....	33
FIGURA 23. CONTROLADORES.....	38
FIGURA 24. REPOSITORIOS.....	38
FIGURA 25. ENTITIES.....	39

FIGURA 26.	COMPONENTES	41
FIGURA 27.	TABLA SERVICES	42
FIGURA 28.	GUARDS.....	42
FIGURA 29.	SQL MANAGEMENT STUDIO.....	43
FIGURA 30.	SCRIPT SELECT * FROM CATALOGOS.POZOS.....	44
FIGURA 31.	SCRIPT SELECT * FROM REPORTE.DATOSPOZO.....	44
FIGURA 32.	SCRIPT SELECT * FROM REPORTE.LECTURAS	44
FIGURA 33.	VISUAL ESTUDIO	49
FIGURA 34.	MÉTODO GET A DATOSPOZO.....	50
FIGURA 35.	METODO GET A ESTAPASPOZO	51
FIGURA 36.	METODO GET A PROYECTOINVERSION	51
FIGURA 37.	RESULTADOS DE CONSULTAS A APIS.....	51
FIGURA 38.	COMPONENTE MAPA	52
FIGURA 39.	COMPONENTE LISTA DE POZOS	52
FIGURA 40.	COMPONENTE SERVICIO	53
FIGURA 41.	LOGIN	54
FIGURA 42.	MAPA	54
FIGURA 43.	LISTA DE POZOS.....	54
FIGURA 44.	AGREGAR POZO.....	55
FIGURA 45.	REPORTES DIARIOS	55
FIGURA 46.	FIGURA AGREGAR REPORTES DIARIOS	55
FIGURA 47.	SERVICIOS.....	56
FIGURA 48.	AGREGAR SEVICIOS	56

ÍNDICE DE TABLAS.

TABLA 1.	TABLA ESQUEMAS	34
TABLA 2.	TABLAS DE LA BASE DATOS	36
TABLA 3.	RESULTADOS SCRIPTS	45
TABLA 4.	RESULTADOS DE SCRIPTS CON TABLAS RELACIONADAS.....	49
TABLA 5.	RESULTADOS DE SERVICIOS.....	53

ABSTRACT.

The following document shows the process for the realization of a project for the exploitation of petroleum technical information which shows the research processes that were carried out in the petroleum area, analyzes each of the technologies available to reach the expected results, As well as seeking to optimize the implementation of the investigated techniques so that the user has a better experience with the platform, as well as analyzes the implementation of a collaborative technological platform with Api-Rest technology, the results of said investigation are shown with physical results already implemented. for technology.

KEYWORDS:

API, REST, OIL, TECHNOLOGY, PLATAFORM

Resumen

El siguiente documento muestra el proceso para la realización de un proyecto para la explotación de información técnica petrolera el cual muestra los procesos de investigación que fueron realizados en el área petrolera, analiza cada una de las tecnologías al alcance para poder llegar a los resultados esperados, así como buscar optimizar la implementación de las técnicas investigadas para que el usuario tenga una mejor experiencia con la plataforma de igual manera de analiza la implementación de una plataforma tecnológica colaborativa con tecnología Api-Rest, se muestra resultados de dicha investigación con resultados físicos ya implementados para la tecnología.

KEYWORDS:

API, REST, PETROLEO, PLATAFORMA, TECNOLOGIA

INTRODUCCIÓN.

La industria petrolera es una de las más importante en el país, ya que nos encontramos en uno de los países más ricos en petróleo en el mundo, esta industria cada año genera miles y miles de datos en información petrolera los cuales son manejados de diferentes formas dentro de la industria.

A través de los años la implementación de la tecnología para la explotación de datos fue creciendo, esto debido a que las empresas guardaban y manejaban sus datos a través de archivos físicos, o documentos electrónicos de un solo formato, es por eso que existían inconsistencias al manejar dicha información y se perdía información histórica.

El manejo de datos dentro de la industria petrolera es muy importante ya que la explotación de esa información influye en la toma de decisiones dentro de algunas de las actividades que se realizan dentro de esa área, estas decisiones tienen gran impacto en el desempeño en los resultados que se obtienen al realizar dichas actividades, es por eso que poder realizar una plataforma para la explotación de información técnica petrolera sería gran ayuda para la industria ya que se obtendrían grandes beneficios dentro de la misma.

La realización de una plataforma tecnología debe definirse junto con la tecnología que se va a usar para que funcione de manera correcta, una tecnología que están en tendencia es la tecnología “web api” que consiste en crear una arquitectura de software donde se pueda implementar dicha arquitectura para que funcione correctamente.

Estas arquitecturas están básicamente definidas en cliente-servidor, pero estas pueden ser personalizadas para que se adapte al tipo de solución a realizar, estas soluciones pueden ser de “n” capas, pero una de las más utilizadas dentro de la industria del desarrollo está compuesta por tres capas, estas pueden ser divididas dentro de la estructura de la plataforma depende a sus necesidades.

1. MARCO REFERENCIAL.

1.1 ANTECEDENTES.

Anteriormente las aplicaciones móviles no tenían un gran uso en el área empresarial o recopilación de datos, el uso de móviles estaba más destinado a solo medio de comunicación como llamadas y mensajes de texto, con la llegada de los smartphones al mercado, también se popularizo el uso de aplicaciones móviles y las empresas comenzaron a invertir en el uso de esta tecnología, ya que era más practico tener a la mano un celular que una computadora portátil.

En los últimos años la tecnología ha dado un gran salto en el mundo empresarial, tanto que actualmente las empresas buscan actualizar sus procesos al uso de nuevas tecnologías como aplicaciones móviles, páginas web, aplicaciones de escritorio entre otros.

El uso de tecnologías de la información para las empresas se convirtió en algo indispensable para poder entrar al mundo de las actualizaciones, procesos que antes se llevaban a cabo de manera manual, ahora se actualizan con procesos automatizados, esto hace que se pueda controlar más el flujo de la información y optimiza los procesos.

La empresa CASE se encarga de realizar consultorías de tecnologías de la información a diferentes empresas, la cual busca ofrecer soluciones informáticas para los procesos empresariales.

El proyecto SEOPEX ofrece actualizar un proceso para la explotación de información técnica petrolera que actualmente se realiza de manera manual con tecnología obsoleta, esto implica perdida de la información, retrasos en los procesos e inconsistencia en la captura de datos.

1.2 CON TEXTO DEL PROBLEMA.

La mayoría de las empresas tiene que realizar muchas actividades para poder llevar a cabo sus funciones, estas actividades pueden ser manuales, automatizados o digitalizados.

Actualmente existen algunas empresas que aún lo realizan manualmente teniendo así problemas para hacer correctamente sus funciones, como también problemas en la recopilación de datos.

Las empresas entraron ahora en una etapa de actualización en la que tienen que automatizar sus procesos con el uso de la tecnología para poder llevar sus funciones correctamente y poder competir con otras empresas.

El propósito de actualizar estos procesos en una empresa es que estos sean más fáciles de llevar a cabo, con el uso de la tecnología que utilizamos diariamente; ya que esto ayudaría al usuario a familiarizarse más con la plataforma.

De acuerdo a un artículo publicación en la página web gestiopolis da mención que actualmente hay diferentes factores que hacen que una gran cantidad de empresas fracasen, los años más difíciles dentro de un grupo lucrativo con a los 5 años desde sus inicios y algunas ni tan siquiera puede cumplir su primer año y solo el 5 por ciento puede trascender mas de los primeros 5 años esto debido a la falta de renovación y actualización de sus tecnologías.

Su investigación muestra que hoy en día una empresa debe absolutamente tener un sitio web para su credibilidad, su marketing y su manera de comunicarse.

En la figura 1 se muestra que número de usuarios de una computadora portátil o dispositivos móviles ha incrementado exponencialmente en los últimos años, por eso las empresas actualizan sus procesos a estas plataformas ya que el usuario está familiarizado con ellas y son más fáciles de utilizar.

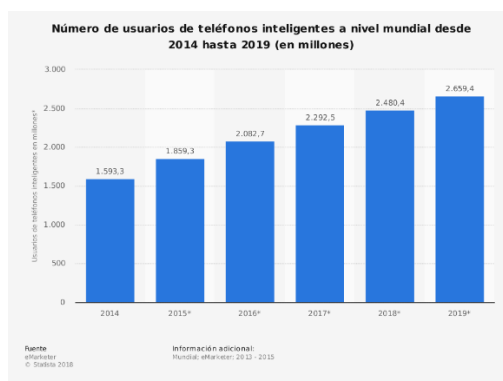


Figura 1. Grafica de número de usuarios de smartphone en los últimos años.

1.3 PLANTEAMIENTO DEL PROBLEMA.

Actualmente la mayoría de las empresas ya cuentan con sistemas automatizados de sus procesos con el uso de la tecnología actual, por la que si una empresa aun realiza sus procesos manualmente se ve obligada a actualizarse

La industria petrolera se encuentra en una etapa en el que busca actualizar sus procesos de perforación y producción para tener un mejor control sobre esto. El país esta en cambio en el cual las empresas petroleras buscan ganar contratos de perforación y producción, pero los procesos que son utilizados para esto ya son obsoletos.

Esto surge en el campo al momento de que la empresa gana un contrato para perforación, tiene la responsabilidad de documentar todo el proceso por el cual se lleva a cabo.

La manera en que hacen este proceso actualmente con lleva en manejar documentos de hojas de cálculo que son llenados manualmente por los ingenieros de campo, estos tienen un formato estandarizado por la empresa Pemex, este mismo se llena de información al momento de empezar el proceso de perforación.

La hoja de cálculo es transferida mediante una memoria usb, un correo electrónico o medios de comunicación como bluetooth al personal de análisis y captura de datos como son los geólogos, personal de laboratorio e ingeniería de pozo.

Estos generan un reporte diario con la información dada por el personal de campo, estos reportes se usan para el análisis de datos, toma de decisiones y visualización del estado mecánico.

Ya realizado los reportes estos son guardados en alguna carpeta electrónica en alguna computadora de la compañía.

Este proceso ya descrito trae inconsistencia en la captura de datos ya que es muy fácil que esta información se pierda porque no cuenta con un sistema de seguridad, además la información no siempre está disponible ya que solo se maneja de manera local en la computadora, esto genera periodos de inactividad laboral y perdida de información histórica.

El proyecto se enfoca en trabajar con estas problemáticas, poniendo a disposición un sistema automatizado para el análisis y captura de datos mediante una plataforma tecnológica.



Figura 2. Esquema de la situación actual.

En la figura 2. Podemos ver la situación actual que presenta la empresa, como el personal de campo lleva los reportes en hojas de cálculos mediante usb, correo electrónico o compartir información mediante línea telefónica a el personal en el análisis y captura de datos, estos se encargan de llenar los campos requeridos en el formato y gracias a esto generan reporte diario de operaciones, estado mecánico del pozo, graficas de avance entre otros.

1.4 JUSTIFICACIÓN.

La problemática identificada en la empresa se encuentra que no cuenta con un mecanismo ordenado para el registro de datos de pozo ya que actualmente dependen de un archivo único en formato Excel para el manejo de los datos y esto trae inconsistencias en la captura de datos, también se encuentran periodos de inactividad por bloqueo de archivos y pérdida de información histórica.

El proyecto implementara el uso de una plataforma colaborativa que sustituya el registro de datos en un archivo por registros en una base de datos para facilitar el flujo de la información.

Esto con el objetivo de facilitar todo el proceso que engloba las operaciones a pozos y así poder realizar un mejor análisis sobre el seguimiento operativo de pozos, se podrá asegurar la integridad de la información con un sistema de seguridad dentro de la plataforma, se resguardará la información histórica del pozo y se generará reportes y graficas operativas.

Con la plataforma se agilizará el registro de datos de pozos ya que no dependerá de un archivo único a como se hace actualmente, gracias a todo esto se tomará menos tiempo para el análisis de datos y se podrá tener mejor razonamiento a la hora de la toma de decisiones.

1.5 OBJETIVOS.

1.5.1 OBJETIVO GENERAL.

Crear una solución tecnológica que facilite el flujo de generación, recopilación y preservación de la información técnica petrolera de operaciones a pozos, para facilitar la toma de decisiones y gestión del conocimiento técnico.

1.5.2 OBJETIVOS PARTICULARES.

- Realizar un análisis sobre el seguimiento operativo de pozos.
- Identificar y determinar la metodología para el desarrollo del sistema.
- Crear una arquitectura de software que permita la comunicación entre plataformas tecnológicas.
- Seleccionar la tecnología para crear el proyecto.
- Implementar la arquitectura de software(backend).
- Diseñar y crear una base de datos de acuerdo a los requerimientos del sistema.
- Implementar una aplicación móvil(frontend).
- Elaboración de manuales técnicos y manual de usuario.
- Capacitación los usuarios y administrador del sistema.

1.6 DELIMITACIONES.

En las características generales de la empresa se encuentra la implementación y el uso de una plataforma colaborativa esto gracias a que la estructura de software y el padrón de diseño a utilizar en este proyecto podrá ser usado en un smartphone con sistema operativo Android, en alguna

computadora con acceso a internet mediante una página web, y en un pc con sistema operativo Windows.

Se sustituirá el registro de datos en un archivo único que actualmente se hace en formato Excel por una base de datos que se crea en sql server para así poder tener información histórica y puntual del pozo.

Al tener todos los datos en una base de datos, se podrá tener mas disponible para el usuario y todos estos datos se procesarán para hacer gráficas y relaciones, y esto ayudara a la toma de decisiones y a la interpretación de toda la información relacionada al pozo.

La seguridad de la información se manejará mediante usuarios y los permisos que estos tengan, esto depende al puesto que el usuario maneje en la empresa y también los pozos que este tenga asignado.

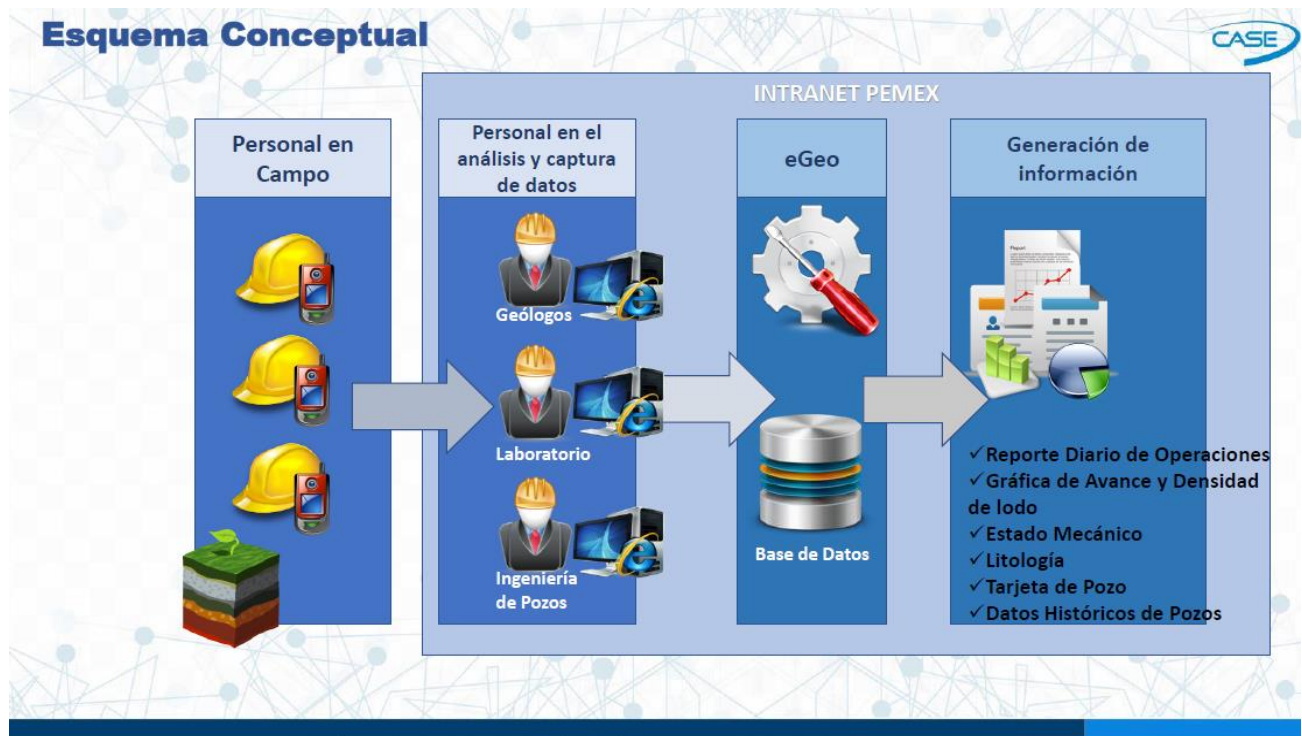


Figura 3. Esquema conceptual de la plataforma.

1.7 INDUSTRIA PETROLERA

En la industria del petrolero podemos encontrar procesos globales, los cuales pueden ser transporte del crudo, refinación del mismo y su extracción (esto con maquinarias ductos o buques) y el proceso de mercadotecnia que lleva el producto del petróleo.

Normalmente los productos que suelen ocupar un gran volumen en la industria son los combustibles y gasolinas, el petróleo crudo es su materia prima, esto con otros químicos se convierten en fertilizantes, pesticidas, plásticos y disolventes,

La industria del petróleo se encuentra dividida normalmente en 3 fases

1. Upstream el cual se encarga de la exploración, extracción y producción del petróleo.
2. Midstream que se encarga del transporte, almacenamiento y los procesos que este requiera.
3. Downstream el refina, vende y distribuye el producto terminado.

En la categoría final se incluyen las medias generales.

El petróleo es un producto que es esencial para diferentes industrias en el mundo por lo tanto es de mucha importancia mantener una civilización industrializada, por lo que se considera una industria que se encuentra en estado crítico en la mayoría de las naciones. El petróleo conserva un porcentaje alto en el uso de energía en el mundo, en el continente europeo con un 32 por ciento y en Asia hasta el 53 por ciento, en algunas otras regiones geográficas su peso en energía del petróleo se puede encontrar con un 44 por ciento en Sudamérica y América central, con un 41 por ciento en África y en Norteamérica con un 40 por ciento.

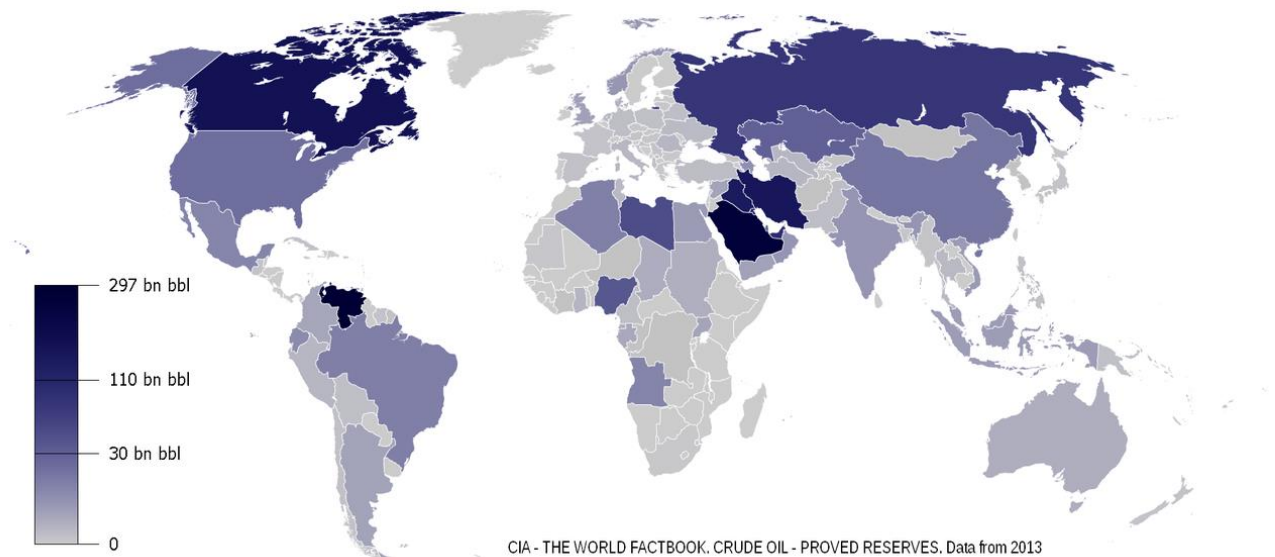


Figura 4. Reserva de petróleo en el mundo

En México se encuentran 6,464.17 millones de barriles de reserva de petróleo esto según datos calculados en 2018 según el sistema de información de hidrocarburos.

La comisión nacional de hidrocarburos en México ocupó el sexto lugar en el mayor productor de petróleo en el año 2006, en el cual se obtuvo una producción de 3,71 millones de barriles por día de los cuales 3,25 millones de barriles por día fueron de petróleo en estado crudo. Pemex en un comunicado realizado en octubre del 2012 informó que ya se habían encontrado dos nuevos

yacimientos de petróleo en el golfo de mexico, mas especifico en el estado de Tamaulipas, esas reservas aseguran la estabilidad económica y en hidrocarburos por unos 30 años mas.

1.7.1 POZO PETROLERO.

Un pozo petrolero es una obra realizada por la ingeniería que se ocupar de poner en contacto un yacimiento de petróleo o hidrocarburos con la superficie. Es una perforación realizada en el subsuelo con diferentes herramientas como barrenas de varios diámetros y con tuberías de revestimiento que se van colocando en diversas profundidades para la explotación del yacimiento. La perforación es un proceso el cual consiste en realizar una perforación mediante la rotación de la sarta de perforación y realizando un empuje hacia el fondo. La perforación rotatoria es cuando se realiza un agujero en el subsuelo por medio de movimientos rotatorios y una fuerza de empuje de la barrena sobre la roca convirtiéndola así en recortes, este movimiento se puede generar y ser aplicado en la superficie mediante maquinas rotatorias y se transmite por medio de la sarta de perforación o también en forma hidráulica con la acción de un motor de fondo que se encuentra conectado con la barrena.

Todo esto según el artículo: Depósitos: Definición y tipos de pozos petroleros de Energy Press

Pozo Exploratorio

Un pozo exploratorio es aquel que es perforado en zonas donde antes no se había encontrado gas o petróleo, básicamente es perforado en un campo nuevo o en una nueva formación en algún campo – existente.

Pozos Productores

Son los que permiten extraer fluidos de las formaciones que ya son productoras, mientras que los secos (no productores) una vez terminado no producen ni gas ni petróleo en cantidades grandes para ser rentable económicamente.

Pozos de Desarrollo

Son los pozos perforados para explorar, drenar y extraer las reservas que se encuentren en un yacimiento, con el objetivo principal de perforar un pozo de desarrolla para aumenta la producción que se encuentra en el campo, por esto se perforan dentro de un área probada y explorada mas sin embargos algunos suelen resultar ser secos.

Pozo de Avanzada

Después de realizar la perforación de un pozo exploratorio en un área que aún no ha sido explorada y resulta ser productor, se realiza la perforación de los pozos de avanzada esto con el objetivo de establecer límites en el yacimiento.

1.7.2 REFORMA ENERGÉTICA.

La reforma energética consiste en mitigar algunos efectos con la creación de mercados que sean competitivos en toda la línea de valor de electricidad e hidrocarburos, también se van estableciendo nuevos esquemas que invitan a la participación de empresas privadas y las empresas que sean productivas en el estado.

La propuesta y proceso de esta reforma energética empezó a inicios del mes de diciembre del año 2013 con la modificación que diferentes artículos de la constitución sin embargo son hasta el 2015 que empieza a consolidarse al realizar la licitación por primera vez en contratos de exploración y extracción de hidrocarburos en subastas realizadas al público.

Según las estimaciones de petróleos mexicanos (Pemex) al desarrollar un potencial en la industria de extracción y exploración se necesitaba al menos unos 60,000 millones de dólares al año, más sin embargo Pemex solo contaba con un presupuesto al año de alrededor de 27,000 millones de dólares.

La reforma energética, Armando Rodríguez, 16 de agosto de 2018

1.8 PLATAFORMA TECNOLÓGICA.

En la informática una plataforma tecnológica es aquella que nos funciona como base para hacer funcionar diversos componentes del hardware o bien módulos de software con los que pueda ser compatible, este sistema está determinado por un estándar en el cual es precisa una arquitectura de software o hardware (con algunos entornos de aplicaciones), al definir esto se puede establecer tipos de arquitectura, lenguajes de programación, sistemas operativos o algún interfaz de usuario compatible.

Ejemplos de estas plataformas pueden ser IBM-PC, que se encuentran con base a la arquitectura I386 (x86), IA64 o AMD64 (x86-64), Macintosh, que contiene la arquitectura Gecko y PowerPC y SPARC. Hay algunos programas multiplataforma que permiten ser ejecutados en diversas plataformas sin perder compatibilidad. También existen emuladores los cuales son programas que permiten ejecutar desde una plataforma programas de que fueron hechas para otras plataformas emulando su funcionamiento, como por ejemplo UAE, que emula el hardware VICE, un emulador muy conocido como Commodore 9 bits de mediados de los 80.

Al referirse al termino plataforma cruzada nos referimos a la capacidad que tiene un hardware o un software para correr de modo que parezca otra plataforma emulando su comportamiento, muchas aplicaciones de Mac o Windows ahora están produciendo programas compatibles a nivel binario lo cual permite a los usuarios cambiar de plataforma sin que cambie el formato de sus datos.

Esta plataforma cruzada se hace cada vez más importante al grado que las redes que se encuentran el área local se puede perfeccionar al integrar las máquinas de diferentes plataformas como por ejemplo el ambiente de programación de una plataforma cruzada que permite a un desarrollador crear programas para muchas plataformas al mismo tiempo

En informática una plataforma es, en una simple definición (y siendo muy creativos), determinado hardware o software o simultáneamente con el cual una aplicación puede ser compatible y este le permita ser ejecutado.

Una plataforma es por ejemplo un sistema operativo, es como un gran software que es base para que se puedan ejecutar algunas aplicaciones que sean compatibles con este, también podría decirse que son plataformas a nivel hardware para que los lenguajes de programación y sus librerías en tiempo de ejecución puedan ser ejecutados.

En la actualidad existen muchos programas que pueden ser multiplataforma ya que pueden ser ejecutados en diferentes sistemas o plataformas o también ya existen diversos emuladores que permiten ejecutar un programa que no sea compatible con el sistema.

1.8.1 WEB API

De acuerdo al artículo “web api” de la revista “bbva actualidad” una web api rest se define como una interfaz que se pueda comunicar entre sistemas que use HTTP para poder obtener datos o crear algún tipo de operación al utilizar dichos datos en todos los formatos posibles, como json y xml. Es una alternativa en apogeo a otros protocolos que utilizan el intercambio de datos como SOAP (Simple Object Access Protocol), que son muy capaces pero muy complejos a la hora de ser desarrollados. A veces es mejor una solución más sencilla de manipulación de datos como REST

Características:

- Protocolo sin estado de cliente/servidor, cada petición realizada por el protocolo HTTP contiene toda la información que es necesaria para ejecutarlo lo que ayuda a que ni el servidor ni el cliente necesite recordar algún estado que haya aparecido antes para satisfacerla. Mas sin embargo algunas aplicaciones HTTP realizadas concretamente contienen memoria cache en la que se configura lo que se conoce como protocolo de cliente, cache, servidor que no tiene estado, hay la posibilidad de poder definir algunas respuestas a las peticiones HTTP personalizables como cacheables esto con el objetivo que el cliente pueda en un futuro ejecutar la misma respuesta a peticiones que puedan ser idénticas, aunque exista no significa que sea la opción más recomendada.

En un sistema REST las operaciones que son más significativas y la especificaciones HTTP son cuatro: GET (leer y consultar los datos), POST (crear un recurso), PUT (actualizar un recurso) y DELETE (elimina un recurso).

Los objetivos en REST normalmente se manipulan a través de la URI. La URI es un identificador único con el cual se puede manipular cada uno de los recursos, este nos permite acceder a la información de los recursos y realizar algunas de las operaciones disponibles en el mismo mediante los verbos HTTP.

Debe haber una interfaz uniforme para la transferencia de datos en un sistema basado en rest, este puede aplicar acciones concretas y definidas como (POST, GET, PUT Y DELETE) los cuales pueden ser ejecutados directamente en los recursos, esto permite la existencia de una interfaz uniforme en el cual esta sistematizada la información.

El sistema de capas no es más que la arquitectura entre los componentes, estos pueden ser personalizados a las necesidades del desarrollador y llevan a cabo una función concreta dentro del sistema REST.

2. ESTADO DEL ARTE.

La tecnología API REST se desarrolló en el año 2000, descrito en la tesis de Roy Fielding, padre de la especificación HTTP. Un servicio REST no especifica una arquitectura software, sino es un grupo de limitaciones y restricciones con las que podemos crear un estilo de arquitectura software personalizado, la cual podremos usar para crear aplicaciones web con el protocolo HTTP.

Hoy en día muchas de las empresas han adoptado esta arquitectura debido a su fácil mantenimiento a nivel back-end y front-end, así como el excelente rendimiento que tienen las aplicaciones con esta arquitectura. Un servicio API REST va orientado a aquellas empresas con aplicaciones que pueden ser lanzadas en diferentes plataformas, como por ejemplo una aplicación web, una aplicación móvil o una aplicación de escritorio, esta tecnología es muy conveniente ya que no importa la tecnología con la que se aplicara el front-end ya que este solo recibe información en formato JSON gracias a la arquitectura back end.

Aquí podemos ver algunas de las empresas que ya implementan esta tecnología.

Facebook.



Facebook es una compañía de estados unidos que ofrece servicio de redes sociales mediante aplicaciones web y aplicaciones móviles con sede en Menlo Park, California, su website fue lanzado el 4 de febrero del 2004 por su fundador Mark Zuckerberg junto con otros estudiantes de la universidad de harvard y sus compañeros de habitación Eduardo Saverin, Andrew McCollum, Dustin Moskovits y Chris Hughes, se encuentra disponible en el idioma español desde el año 2008, esta es una plataforma que funciona sobre una infraestructura basada principalmente en Linux utilizando un conjunto de tecnologías LAMP.

Fuente: <https://es.wikipedia.org/wiki/Facebook>

Esta compañía a través de su pagina web ofrece un conjunto de interfaces de programación de aplicaciones (API) que los desarrolladores pueden utilizar para aprovechar la tecnología y los datos de esta red social así como extender sus aplicaciones y sitios web. Las principales interfaces de programación de aplicaciones que ofrece Facebook incluyen la Ads API, Credits API, Chat API y Graph API.

Google



Google es una compañía que principalmente es subsidiaria de la multinacional estadounidense Alphabet Inc., la cual su especialidad son los productos y servicios relacionados con Internet, software, dispositivos electrónicos y otras tecnologías. El producto principal de Google es el motor de búsqueda de contenido en Internet mediante índices, del mismo nombre, aunque también ofrece otros productos y servicios como el correo electrónico llamado Gmail, sus servicios de mapas Google Maps, Google Street View y Google Earth, el sitio web de vídeos YouTube y otras utilidades web como Google Libros o Google Noticias, Google Chrome y la red social Google+,

Fuente: <https://es.wikipedia.org/wiki/Google>

Las API de Google es un conjunto de API desarrollada por Google, las cuales permiten la comunicación e integración de los Servicios de Google con otros servicios. Ejemplos incluyen las API de Búsqueda, Gmail, Traductor o Maps. Las aplicaciones de terceros pueden usar esas API para extender la funcionalidad de sus servicios.

Spotify



Spotify es una aplicación que es multiplataforma sueca, empleada para la reproducción de música vía streaming. Mediante aplicaciones web y aplicaciones móviles. Cuenta con un modelo de negocio premium, un servicio gratuito con funciones básicas y con publicidad, pero que contienen características adicionales, como una mejor calidad de audio, a través de una suscripción de pago. Permite escuchar en su modo radio buscando por artista, álbum o listas de reproducción creadas por los propios usuarios.

Fuente: <https://es.wikipedia.org/wiki/Spotify>

Spotify ofrece como las demás empresas mencionadas un conjunto de interfaces de programación de aplicaciones (API) la cual se puede encontrar en su website para desarrolladores, estas API's se pueden utilizar de forma libre, solo se necesita registrarse en su página oficial, esto con la intención de crecer su número de aplicaciones y el alcance a los usuarios.

2.1 CONTECH.

Contech es una empresa con sede en México que ha estado 22 años en el mercado, brinda soluciones para la automatización y monitoreo remoto de variables de procesos

Esta empresa empezó sus operaciones en el año 1991, sus socios indicaron que es una compañía que se basa en la aportación de última tecnología con la adquisición, transmisión y control de datos en sus procesos con controles de variables como flujo, presión, nivel et.



Figura 5. Logo de contech.

2.1.1 SCHNEIDER.

Esta empresa utiliza un portafolio de productos y soluciones en las que aborda la certificación de pozos, manejan servicios son control y automatización con instrumentos específicos que se pueden utilizar en yacimientos, esto para la medición de sus variables de entorno.



Figura 6. Logo de Schneider electronic

Su misión es aportar energía y eficiencia para prolongar la vida, el progreso y la sostenibilidad para todos. Se consideran pioneros en digitalización y transformación de energía y los procesos, y convierten las grandes ideas de nuestros clientes en realidades. Cumplen con los más altos estándares de responsabilidad y ética.

3. DESARROLLO DEL PROYECTO.

Desarrollar una plataforma tecnológica que permita la gestión de los datos que conforman la información técnica petrolera que se genera día a día en pozos petroleros para así tener una mejor administración de la información y poder mejorar el rendimiento de los procedimientos que se emplean.

Para poder generar mejor los procedimientos se necesita un análisis de las diferentes áreas en las que se crea esta información, profundizar todo el procedimiento que se lleva a cabo y así poder crear un software funcional que se pueda adaptarse a las necesidades del cliente.

El proyecto implementará el uso de una plataforma colaborativa que sustituya el registro de datos en un archivo por registros en una base de datos para facilitar el flujo de la información.

Esto con el objetivo de facilitar todo el proceso que engloba las operaciones a pozos y así poder realizar un mejor análisis sobre el seguimiento operativo de pozos, se podrá asegurar la integridad de la información con un sistema de seguridad dentro de la plataforma, se resguardará la información histórica del pozo, generando reportes y graficas operativas.

3.1 MODELO DE ANÁLISIS.

Se realiza un modelo de análisis para determinar los actores y visualizar de manera abstracta todo el proceso que contiene el proyecto para poder ver los resultados finales

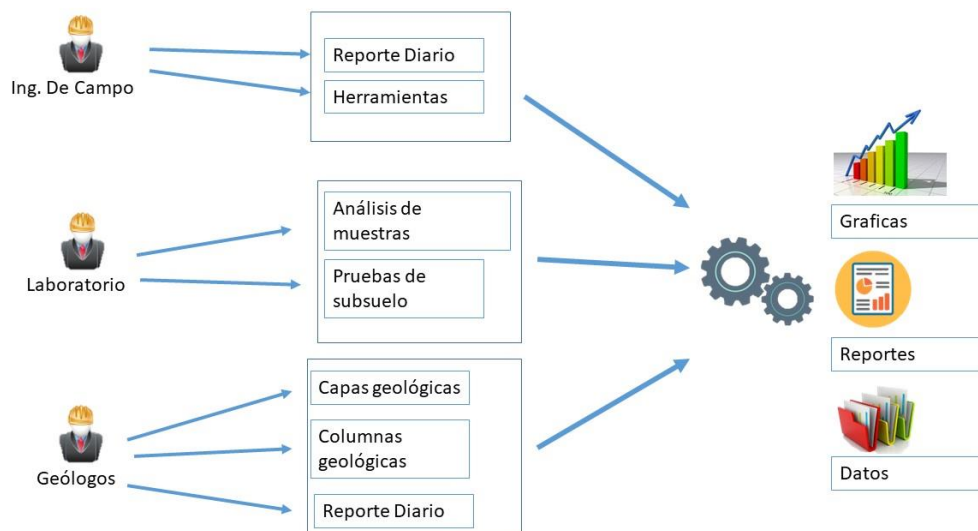


Figura 7. Modelo de analisis

Al identificar los procesos y visualizar el modelo de análisis podemos determinar que el proceso de este proyecto conlleva a tres actores, los cuales son:

- Ingeniero de campo
- Geólogo
- Laboratorio

Con los cuales se puede ver una participación colaborativa para poder obtener los resultados requeridos, el ingeniero de campo realiza reportes diarios los cuales van relacionados con el seguimiento operativo del pozo, también puede integrar las herramientas con las cuales se está trabajando, el geólogo de acuerdo a los reportes diarios debe ingresar información sobre la capa geológica sobre la cual se encuentran y especificar la columna geológica, el laboratorio debe recopilar muestras de aceite, subsuelo, gases etc. Y realizar un informe el cual debe de ser guardado en el sistema.

3.1.1 RECOPIACIÓN DE REQUISITOS.

Se realiza una reingeniería de requisitos dependiendo de las necesidades del proyecto, especificando puntualmente los procedimientos que requiere el sistema, esto de acuerdo a las especificaciones al cliente y a la información obtenida en la realización de la descripción del proyecto junto con el cliente final.

A. REQUISITOS FUNCIONALES.

- La plataforma debe administrar toda la información capturada por los administrativos del pozo.
- Podrá generar un reporte diario.
- El laboratorio podrá generar informe de muestras.
- Los geólogos podrán ingresar capas y columnas geológicas de acuerdo a un catalogo
- El Ingeniero del pozo podrá visualizar graficas del seguimiento operativo.
- Visualizar estado mecánico del pozo
- Guardara un registro histórico en la base de datos.
- Generará graficas con la información obtenida.
- La base de datos será relacional con SQL Server
- Generará credenciales.

B. REQUISITOS NO FUNCIONALES.

- El back-end será realizado específicamente en el lenguaje C#
- El lenguaje front-end podrá ser elegido por el cliente
- La base de datos se realizará en Sql Server
- La documentación generada será en formato xlsx.
- Deberá implementarse un servidor.

3.1.2 METODOLOGIA

Scrum es una metodología para el desarrollo agil de algún software, permite trabajar en un grupo de trabajo amplio y tener una comunicación constante con el cliente implementando buenas practicas.

Se decide trabajar con esta metodología por el marco de trabajo en el cual se puede implementar, al ser un software de desarrollo de una aplicación web está sujeto a cambios, esto quiere decir que se debe tener una comunicación directa con el cliente y con el equipo de trabajo realizando reuniones diarias de avances e indicando tareas directamente al equipo mediante un tablero Kanban el cual consiste en desglosar las tareas 3 fases básicas que son:

- Por hacer
- Haciendo
- Hecho

En estas fases se van ingresando tareas conforme indique su título y el scrum master podrá ingresar más tareas al equipo de trabajo y este deberá revisarlas.

De igual manera se le puede anexar al tablero fases conforme al proyecto requiera como fase de revisión, fase de testeo, fase de implementación etc.

3.1.3 TECNOLOGÍA

A. SQL SERVER

Microsoft SQL Server es un sistema de gestión de base de datos relacional, desarrollado por la empresa Microsoft.

El lenguaje que se utiliza en líneas de comando es Transact-SQL que se implementa del estándar ANSI, este lenguaje facilita la lógica de negocios por lo que es de las más utilizadas en esta área, permite tareas básicas como son crear tablas y definir relaciones entre ellas así como implementar funciones, procedimientos almacenados y triggers.

Se optó por utilizar Sql Server por la tecnología que Microsoft maneja para la administración de la base de datos ya que sus funcionalidades se adaptan a las necesidades del problema teniendo así mejor optimización del sistema.

La empresa en la que se realizara el proyecto cuenta con un amplio conocimiento de la tecnología ya que ha desarrollado aplicaciones web y móviles utilizando la misma, por lo que en su recomendación esta opción es muy adaptable al proyecto ya que puede tener un amplio crecimiento.

La herramienta grafica que usa es SQL Server Management Studio esta es una aplicación de software lanzada por primera vez con Microsoft SQL Server 2005 que se usa para configurar, administrar y administrar todos los componentes dentro de Microsoft SQL Server . Es el sucesor del Enterprise Manager en SQL 2000 o antes. La herramienta incluye editores de scripts y herramientas gráficas que trabajan con objetos y características del servidor. Esto facilita al usuario la visualización y administración de los datos.

La lógica de negocios del sistema es usada por Procedimiento Almacenados(Stored Procedure) que pueden ser instrucciones de tipo Transact-SQL o referencias a un método de Common Runtime Language (CLR) de .NET. Dichas instrucciones se encuentran almacenadas de forma física con un nombre dentro de la base de datos.

B. .NET CORE

Net Core es un framework informático que es considerado Open Source para los sistemas operativos Linux, Windows y dispositivos Mac, es sucesor del framework de .NET. este esta desarrollado por Microsoft principalmente con una licencia MIT.

Con la tecnología .net core crearemos las diferentes api's para que el software sea multiplataforma y así poder conectar entre sistemas y visualizar la información de una forma precisas en cualquier tecnología en donde se creara el front-end.

La tecnología de NET Core puede admitir cuatro escenarios multiplataforma:

- Desarrollo de aplicaciones web con net core
- Aplicaciones de consola.
- Librería de clases.
- Windows Presentation Foundation que permite el desarrollo de aplicaciones de escritorio con interfaces gráficas deWindows.

Al llegar NetCore 3.0 se puede agregar soporte para el desarrollo de aplicaciones de escritorios realizadas con Windows Forms Y Windows Presentation Foundation.

Con el uso de NetCore se puede admitir el uso de paquetes NuGet que a diferencia del framework .Net, que utiliza Windows Update, este utiliza su propio sistema de gestión de paquetes para recibir actualizaciones.

Igual a como el framework de .NET implementa su infraestructura con un lenguaje común (CLI) a través de su entorno en tiempo de ejecución de lenguaje común (CLR) y el framework de libreria de clases (FCL) NetCore implementa en su propio entorno CoreCLR y CoreFx respectivamente.

A la implementación CLI del sistema de ejecución virtual (VES), CoreCLR es un sistema en tiempo de ejecución y máquina virtual para la ejecución de código que es gestionado.

NetCore contiene CoreRT, el cual suministra una compilación anticipada en código nativo optimizado y un compilador justo a tiempo llamado RyuJit..

C. ANGULAR

Angular es un framework para JavaScript (que también puede ser utilizado con TypeScript) muy potente y adecuado para el desarrollo de aplicaciones web modernas con una complejidad mediana o alta, las aplicaciones comúnmente desarrolladas en este framework son de tipo SPA que son comúnmente denominadas aplicaciones de una sola página o Aplicaciones web progresivas (PWA)

Este framework nos da una base para el desarrollo de aplicaciones flexibles, robustas y de fácil mantenimiento, que promueve además las mejores prácticas y un estilo de codificación homogéneo y de gran singularidad.

Normalmente angular nos da una base para el desarrollo de la parte del Front End con la programación en JavaScript, este también contiene técnicas de desarrollo para la parte del Back-End con la implementación del angular universal o Server Side Redering.

El desarrollo en Angular se implementa con TypeScript (aunque de igual manera se podría desarrollar con el lenguaje Javascript, pero toda su documentacion y recomendaciones se basan en usar TypeScript), el cual es un complemento del lenguaje Javascript que ofrece muchas herramientas adicionales al lenguaje, como el tipado estático o los decoradores.

Angular es un framework que se especializa en el desarrollo de aplicaciones SPA, Single Page Application que son aplicaciones de una sola página. En este tipo de aplicaciones se muestra en una sola página y las interfaces son simplemente vistas que se van cambiando con el routing desarrollado. con ello se obtiene una mejor experiencia de usuario, muy parecida a la de aplicaciones de escritorio.

Para logra esto el framework angular ofrecer un sistema de routing que es altamente configurable y muy potente, el cual se puede implementar fácilmente con una estrategia de Lazy Loading. Asi se pueden desarrollar aplicaciones muy optimizadas que pueden cargar muy rápido desde el primer acceso hasta cuando se navega por las otras páginas.

D. ESTRUCTURA DEL SOFTWARE

Al seleccionar la tecnología para el desarrollo del proyecto la estructura del software nos quedaría de la siguiente manera:

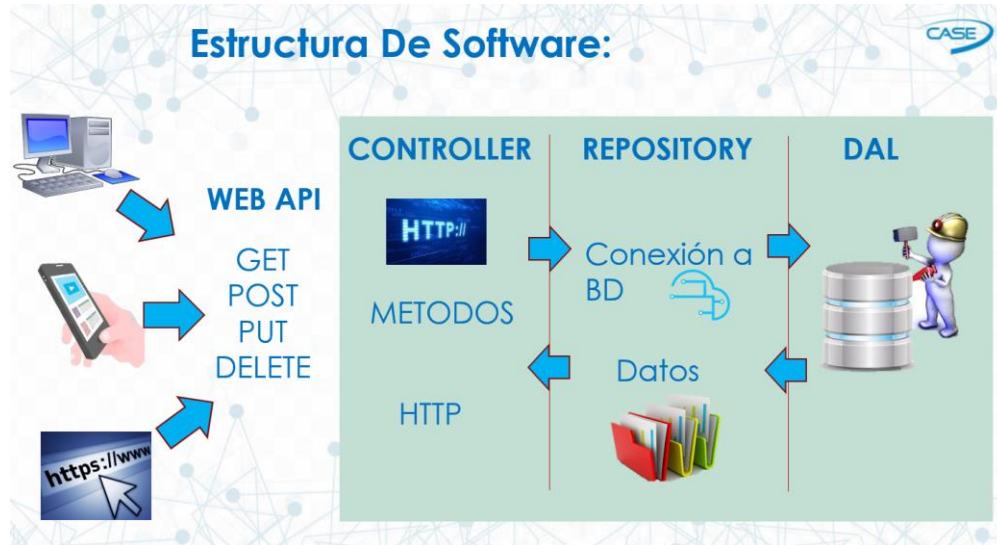


Figura 8. Estructura del proyecto

Esta estructura es de alta confiabilidad con cada uno de sus componentes y está diseñada para implementar cualquier componente dentro de ella sin que se modifique ninguno de los otros objetos que la integran, esto reduce tiempos de mantenimientos, así como poder modificar sus objetos sin que se tenga que compilar completamente el software. Por lo tanto, podremos tener una plataforma de alta estabilidad y confiable para el usuario.

E. FACTIBILIDAD OPERACIONAL.

CANTIDAD	DESCRIPCION	PRECIO POR UNIDA	TOTAL
5	Computadoras de escritorio HP con 8gb de RAM, 1tb de disco duro, procesador Intel i5	10,000.00	50,000.00
1	Servicio de internet de 10mb	1,000.00	1,000.00
1	Servicio de dominio web y hosting	1,000.00	1,000.00
5	SmartPhone con sistema operativo Android, 3gb de RAM, 16gb de almacenamiento	6,000.00	30,000.00

Total: \$82,000.00

Proyecto Total: \$341,500.00

En la factibilidad operacional tenemos descritos las necesidades para que el proyecto pueda funcionar de manera correcta.

F. RECURSOS HUMANOS.

CANTIDA D	CARGO/PUESTO	PRECIO POR HORA	HORAS	TOTAL
1	Analista	180.00	100	18,000
1	Diseñador	200.00	100	20,000
1	Administrador BD	400.00	150	60,000
1	Ing. Software	250.00	130	32,500
1	Programador	400.00	150	60,000

Total: \$190,500.00

En la tabla podemos visualizar los recursos humanos utilizados para la creación del proyecto, así como el precio por hora, la cantidad de horas a trabajar y la cantidad total a pagar a cada uno.

G. RECURSOS TECNOLÓGICOS.

CANT.	DESCRIPCION	PRECIO POR UNIDAD	TOTAL
1	Laptop HP con 16 gb de memoria ram, 2 tb de disco duro, procesador Intel i7, memoria de video de 4gb	25,000.00	25,000.00
1	Monitor LED de 32 pulgadas	6,000.00	6,000.00
1	Conexión a internet de 20mb	1,000.00	1,000.00
1	Impresora HP con conexión wifi	5,000.00	5,000.00
1	Licencia Visual Studio Enterprise	32,000.00	32,000.00

Total: \$69,000.00

En la tabla podemos encontrar los recursos a utilizar para la realización del proyecto así como la descripción y precio por unidad de cada uno de ellos.

H. PLAN DE TRABAJO.

Para la creación del proyecto tenemos el siguiente cronograma de actividades.

Actividades, Meses	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Realizar un análisis sobre el seguimiento operativo de pozos.	█	█	█	█	█	█																				
Crear una arquitectura de software que permita la interoperabilidad entre plataformas tecnológicas						█	█	█	█	█																
Seleccionar Tecnología Para Crear el proyecto										█	█	█														
Implementar la arquitectura de software(backend)											█	█	█	█	█											
Diseñar la base de datos															█	█	█	█	█							
crear la base de datos																				█	█	█				
Implementar una aplicación móvil																						█	█	█	█	█

Figura 9. Cronograma de actividades

De acuerdo con nuestro cronograma de actividades tenemos un total de 26 meses para la realización e implementación del proyecto donde los primeros 6 meses nos indica la realización de un análisis sobre el seguimiento operativo de pozos donde gracias a los resultados podemos obtener un proceso más detallado que se lleva a cabo al realizar un seguimiento operativo.

Como segundo argumento tenemos la creación de una arquitectura que permita la interoperabilidad entre sistemas esto para que al software se le puedan ir implementando componentes sin la necesidad de recompilar todo el proyecto, de acuerdo a un análisis se optó por la arquitectura Api Rest que descompone el software en 3 capas:

- Base de datos,
- Back-End.
- Front-End.

Estas capas son totalmente independientes lo que nos permite manipular los datos de una forma más segura.

Al implementar esta arquitectura seleccionamos cuidadosamente la tecnología a usar en el proyecto, la cual pudiera adaptarse perfectamente y nos pudiera dar los resultados esperados, conforme a las capas se seleccionaron las siguientes tecnologías:

- SQL SERVER
- .NET CORE
- ANGULAR

Estas tecnologías trabajando de manera conjunta nos darán los resultados esperados en el proyecto.

Para la creación de la arquitectura empezamos con la base de datos que la crearemos en sql server e implementaremos procedimientos almacenados para la lógica de negocios, en la parte del back-end utilizaremos .net core para la implementación de las apis y para el consumo de estas utilizaremos la tecnología angular.

3.1.4 MODELADO DEL ANÁLISIS.

Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo, la relación y la generalización son relaciones. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar como reacciona una respuesta a eventos que se producen en el mismo.

De acuerdo al análisis realizado se crearon casos de usos para cada uno de los actores que participaran en la plataforma

A. CASOS DE USOS.

Tenemos los siguientes casos de uso:

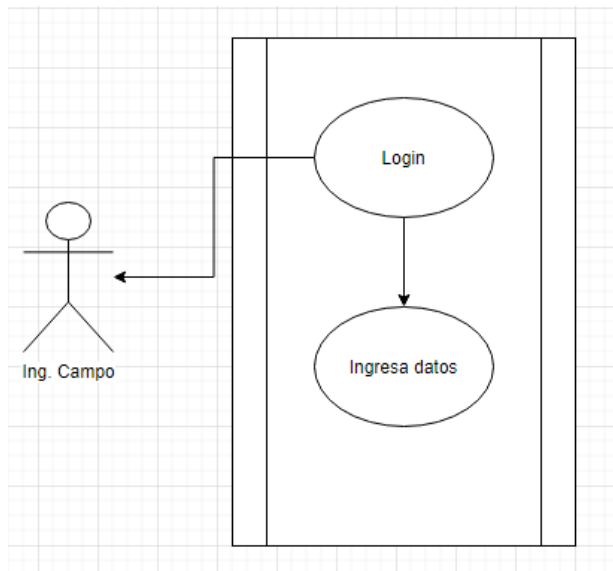


Figura 10. caso de uso: Ing. Campo

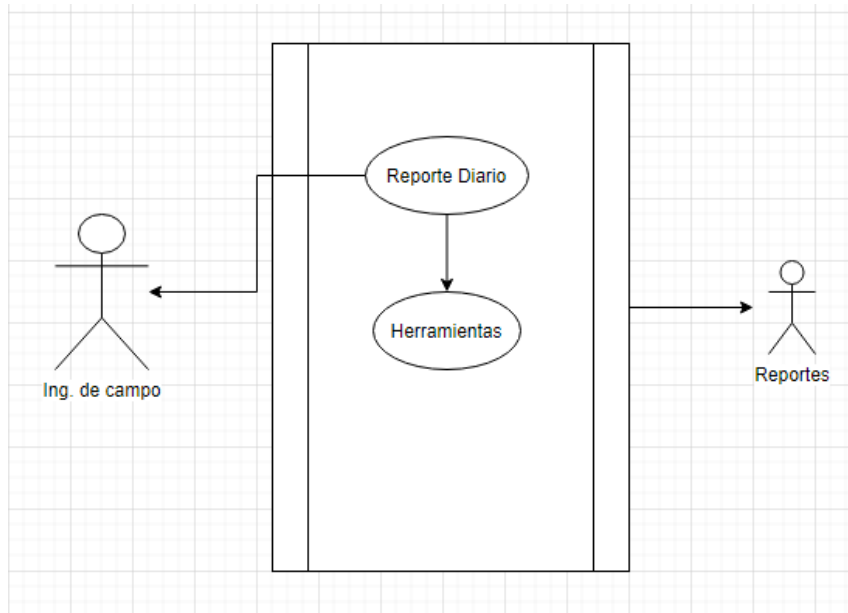


Figura 11. comportamiento de ing. campo

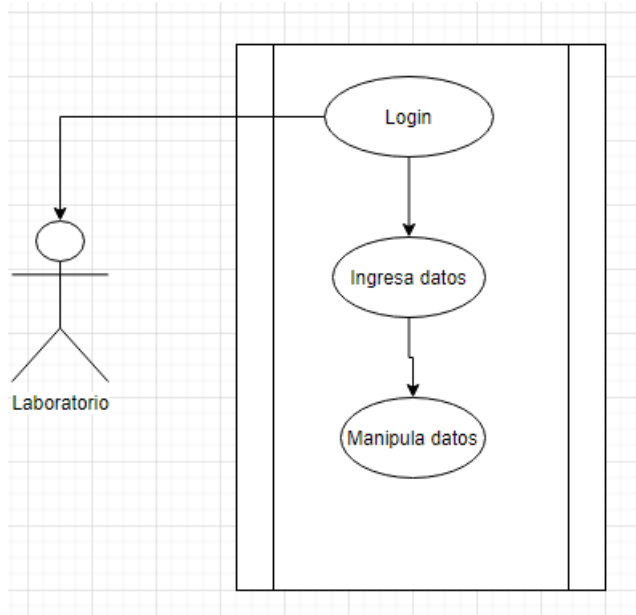


Figura 12. caso de uso: laboratorio

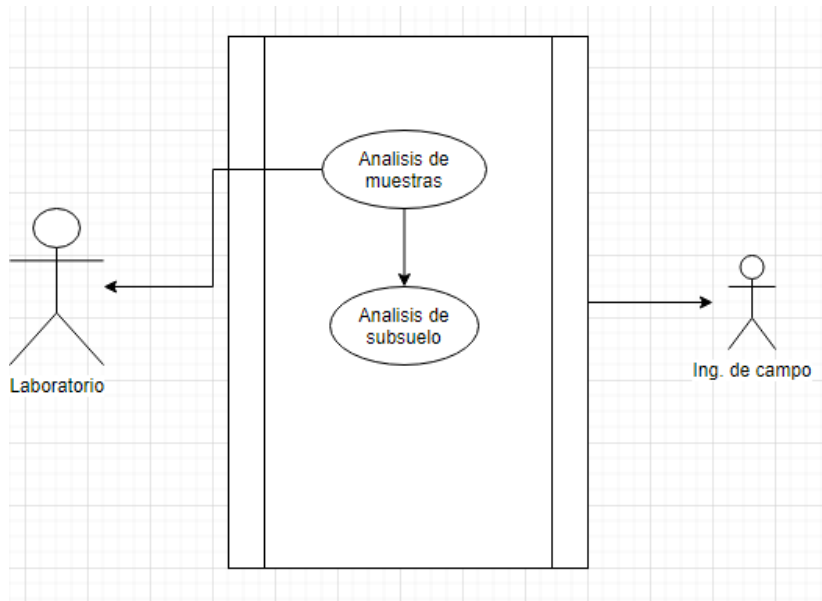


Figura 13. comportamiento de laboratorio

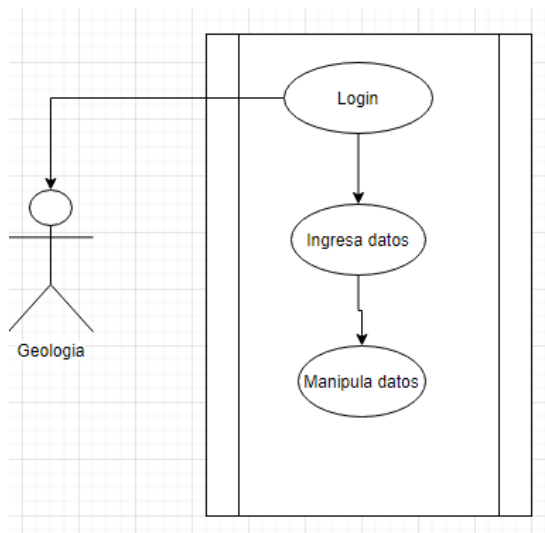


Figura 14. caso de uso: Geología

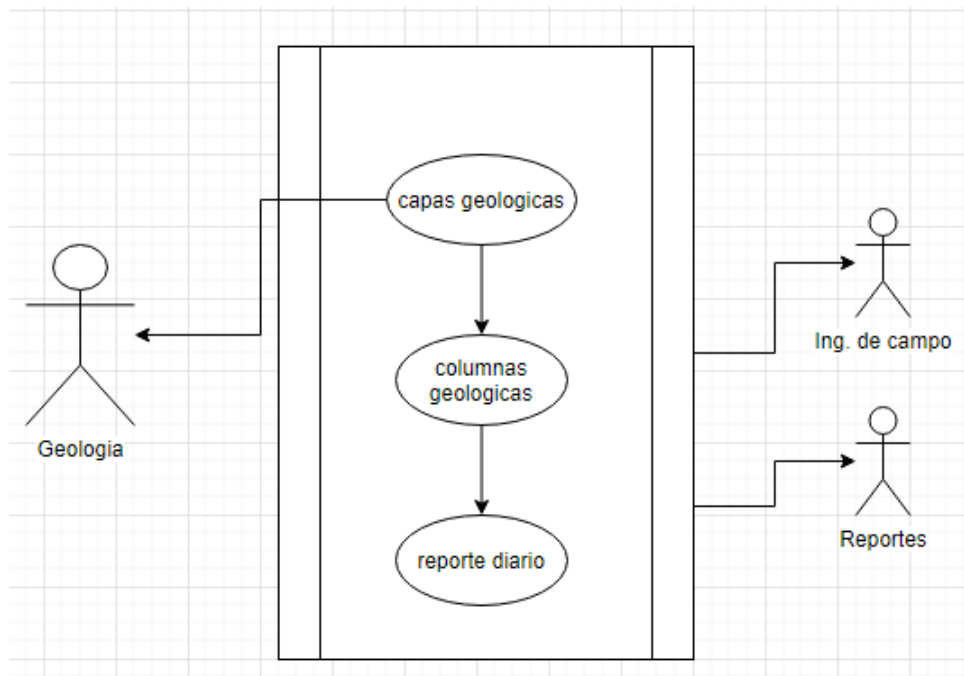


Figura 15. Comportamiento de geología

B. ESPECIFICACIONES DE CASOS DE USOS.

Para la creación de casos de usos conforme al análisis ya realizado se utilizaron 3 actores que los que formarán parte de la plataforma tecnológica, estos actores podrán visualizar, manipular y actualizar datos dependiendo de su actuación en el sistema, especificaremos cada caso de uso por actores.

Tenemos como primer actor el ingeniero de campo, como primera instancia debe loguearse con sus credenciales en el sistema cuando el administrador de sistemas haya creado su perfil con sus datos correspondientes, este deberá ingresar datos diarios a la plataforma dependiendo del turno en que este se encuentre, al ingresar los datos en la plataforma estos serán manipulados por los otros actores.

El geólogo de igual manera podrá loguearse con sus credenciales en la plataforma podrá ingresar datos de su área y de acuerdo a los datos ingresados por el ingeniero de campo estos serán manipulados para poder observar los reportes correspondientes.

El laboratorio tiene una participación importante en la plataforma ya que este actor más que nada es para manipular los datos que el geólogo y el ingeniero de campo hayan ingresado, este se encargara de los estudios correspondientes para obtener los resultados deseados y de ser así se podrá observar mediante gráficas y reportes.

Al especificar cada uno de estos actores podemos observar que se trata de una plataforma colaborativa donde los datos dependen de otros actores para que puedan ser correctos, cada uno deberá mantener una comunicación constante entre ellos para poder utilizar la plataforma correctamente.

Al obtener los resultados podremos observar graficar y reportes que facilitaran a la toma de decisiones de los actores en los trabajos de campos.

Se manejarán diferentes perfiles para el manejo de seguridad del sistema donde cada uno de los actores tendrá un rol para desempeñar su papel en la plataforma, estos roles son definidos por el administrador de sistema y de igual manera seleccionara el correcto para cada uno de sus áreas correspondientes, estos roles servirán para mantener seguros los datos ya que de acuerdo a ellos se podrán colocar permisos para manipular, ingresar y actualizar datos.

Sin embargo, al crear el perfil de los actores se incorpora un rol por defecto para la visualización de datos dependiendo a su área, para poder cambiar este rol deberá comunicarse con el administrados de sistemas y mostrar sus motivos por el cual se deberá cambiar sus permisos en su rol, de ser aceptado el cambio el administrados podrá cambiar sus permisos y este podrá loguearse con su mismo usuario, pero con sus permisos cambiados.

3.1.5 MODELADO DE DATOS.

El modelado de datos es una manera de estructurar y organizar los datos para que se puedan utilizar fácilmente por las bases de datos. Los datos no estructurados se pueden encontrar en los documentos de procesamiento de texto, mensajes de correo electrónico, archivos de audio o vídeo, y programas de diseño.

El modelado de datos no quiere estos datos "crudos" sino que el modelado de datos quiere que todos los datos se presenten en un paquete bonito, limpio para el procesamiento de una base de datos. Así que, en cierto modo, el modelado de datos se refiere a cómo se ven los datos.

En la realización del modelado de datos tomamos en cuenta todos los posibles factores que podrían generarse en el uso de la información para guardar datos, así que se realizó un análisis exhaustivo para seleccionar que datos pudieran ir en la base de datos y no perder ningún dato histórico de la plataforma.

A. MODELADO DE DATOS RELACIONAL EN BASE DE DATOS

A continuación, se muestra el diagrama entidad relación de nuestra base de datos.

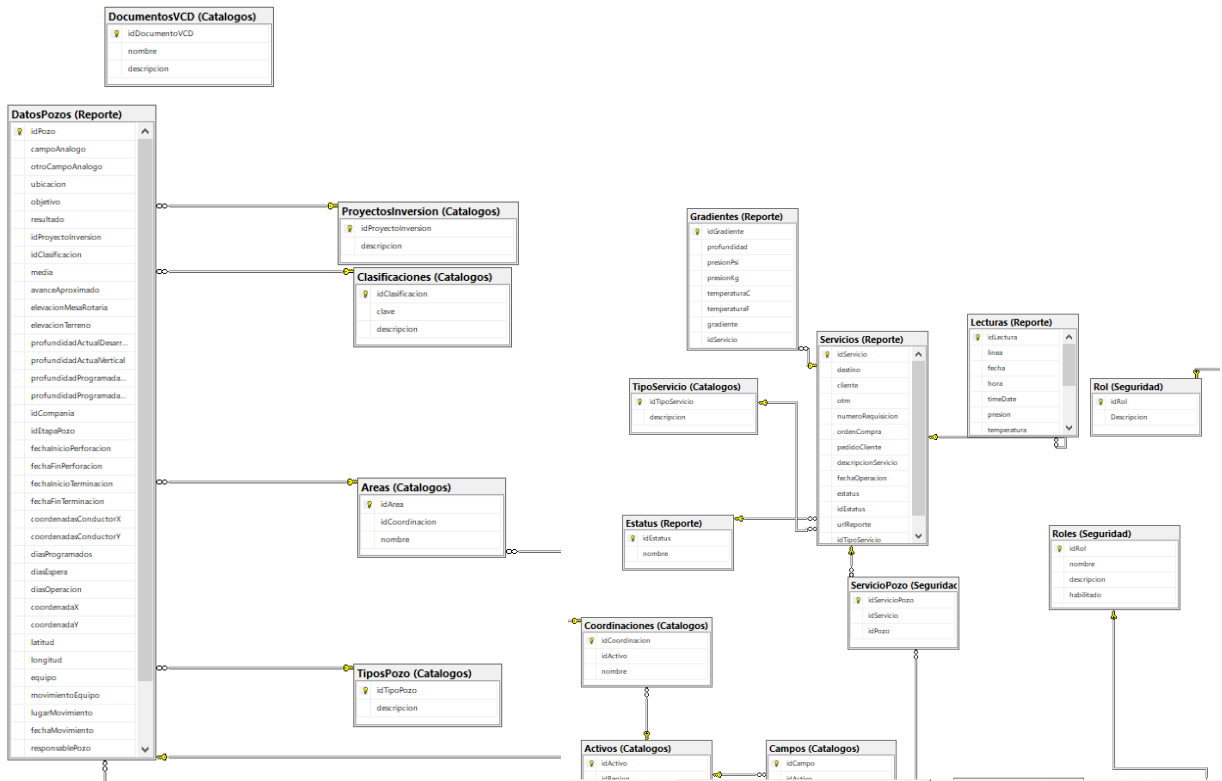


Figura 16. Base de datos part.1

Base de datos part2

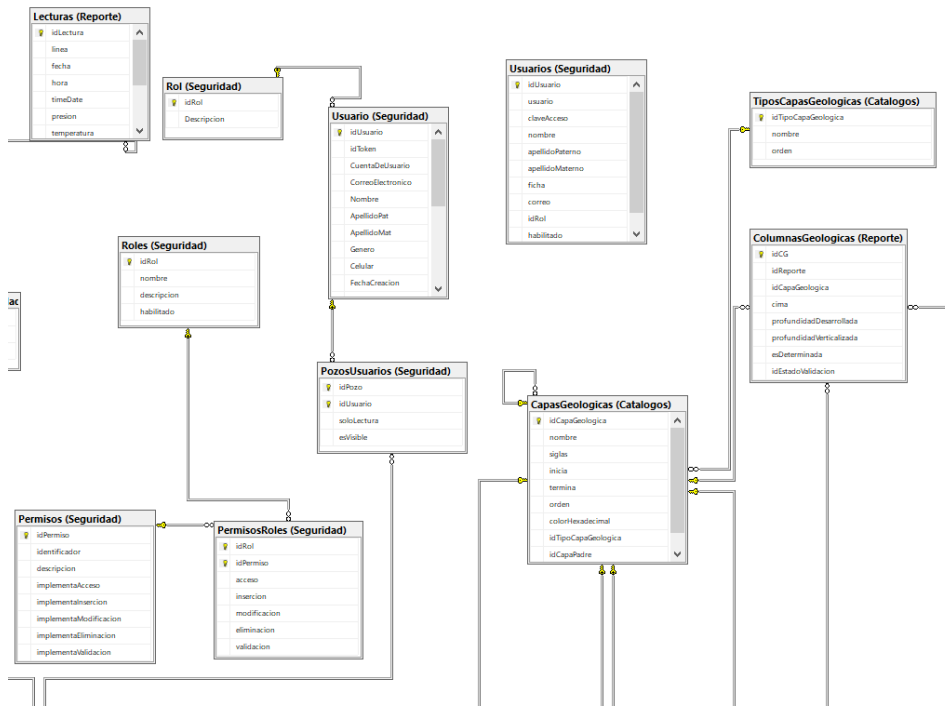


Figura 17. Base de datos part. 3

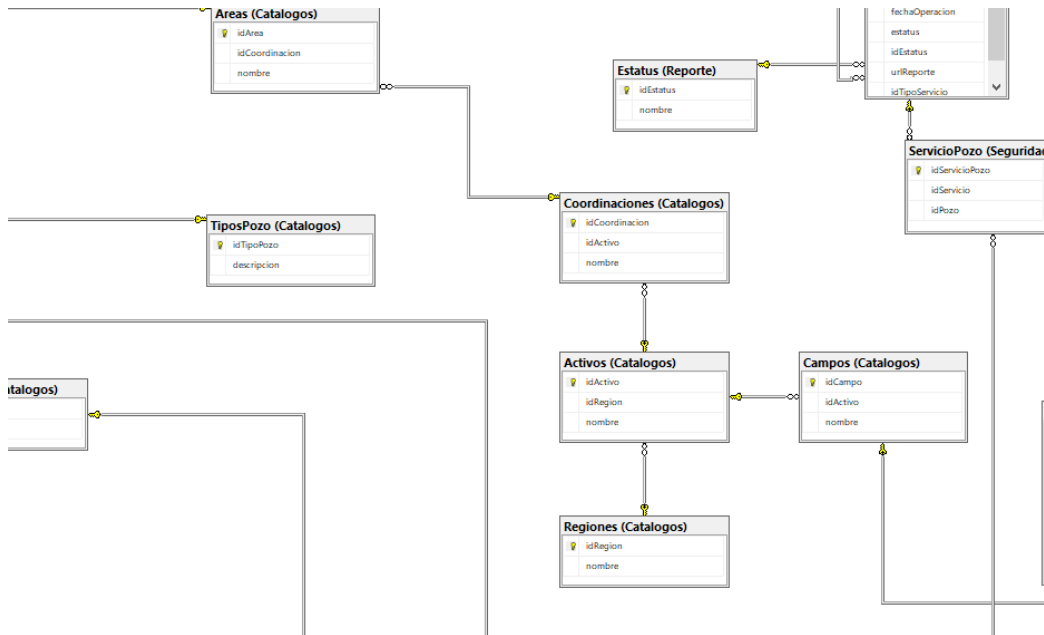


Figura 18. Base de datos part 4

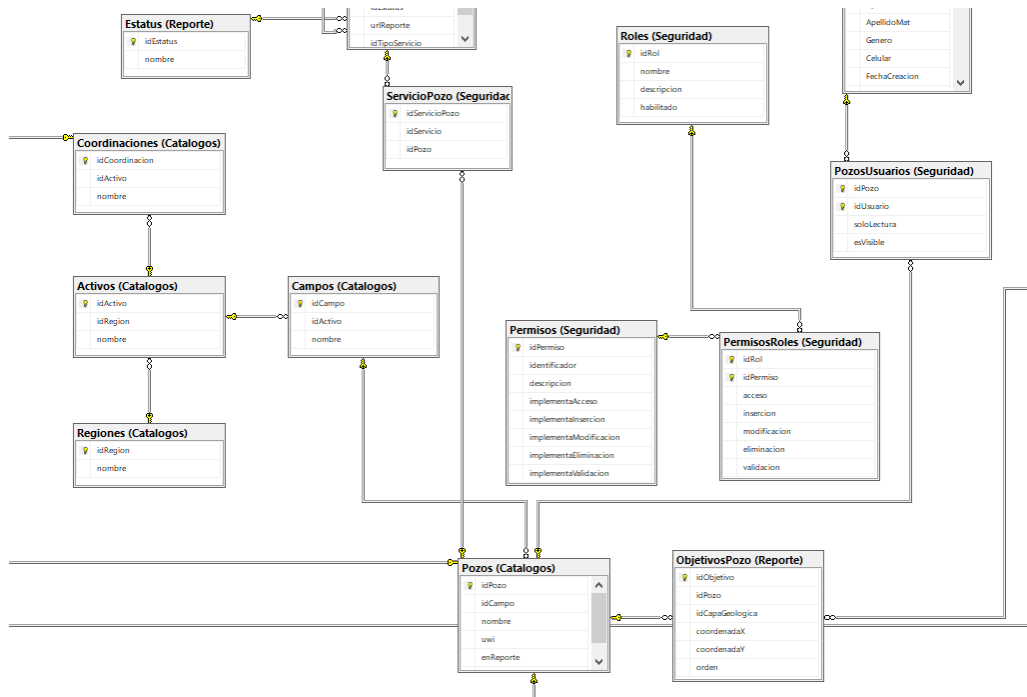


Figura 19. base de datos part. 5

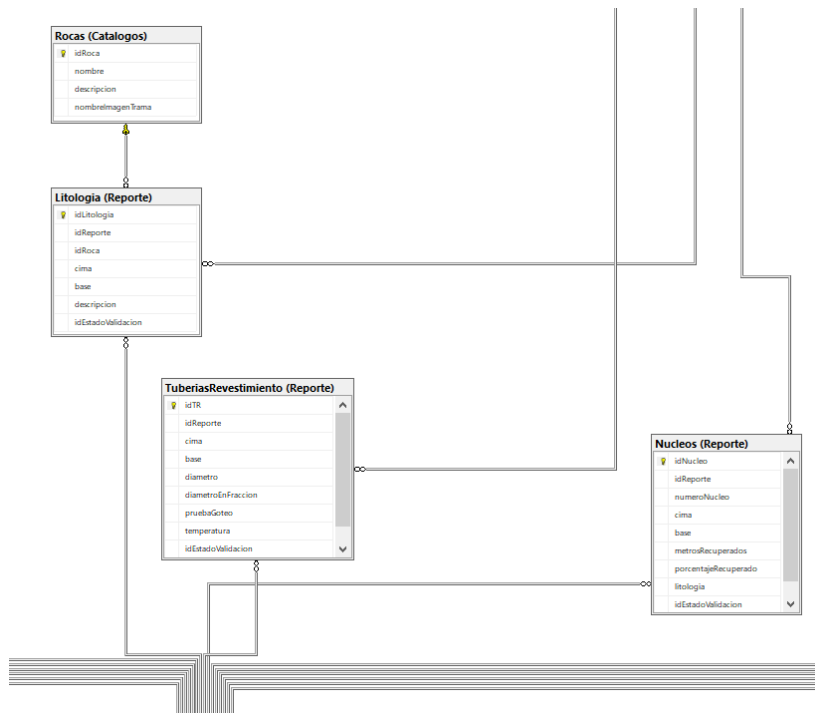


Figura 20. Base de datos part. 6

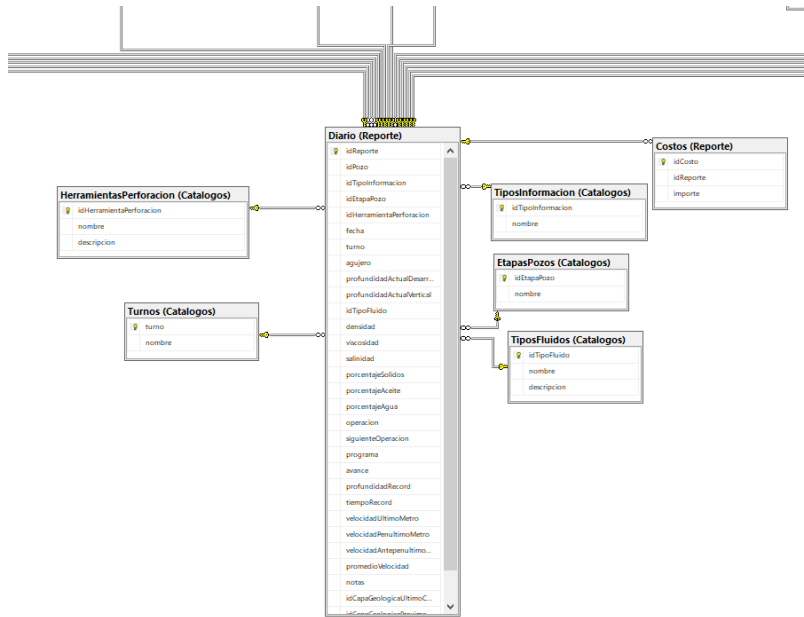


Figura 21. base de datos part. 7

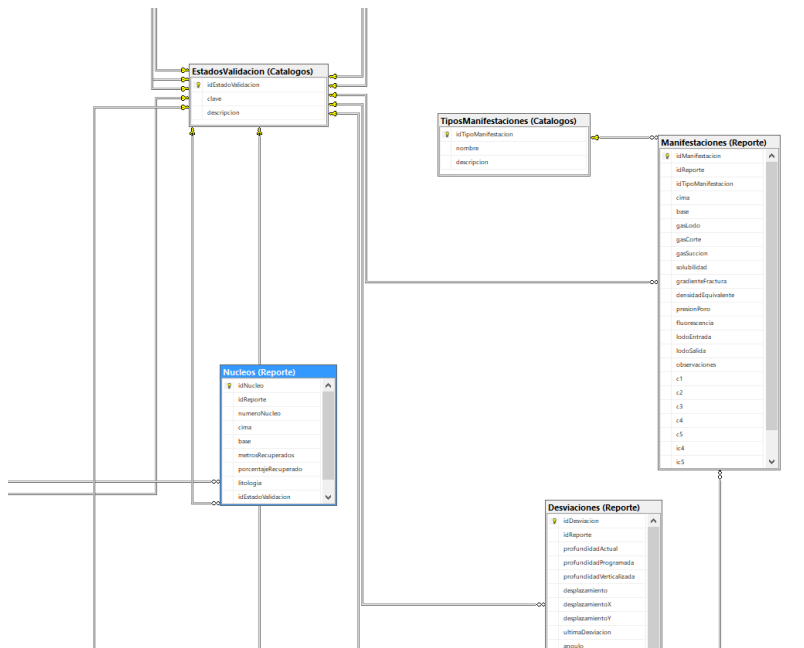


Figura 22. Base de datos part 8

B. ESQUEMAS

Un esquema de base de datos representa la configuración lógica de todo o parte de una base de datos relacional. Puede existir de dos formas: como representación visual y como un conjunto de fórmulas conocidas como restricciones de integridad que controlan una base de datos. Estas fórmulas se expresan en un lenguaje de definición de datos, tal como SQL. Como parte de un diccionario de datos, un esquema de base de datos indica cómo las entidades que conforman la base de datos se relacionan entre sí, incluidas las tablas, las vistas, los procedimientos almacenados y mucho más.

Se especificación los siguientes esquemas en la base de datos:

Nombre:	Descripción
Catálogos	En este esquema se integran todas las tablas con datos relacionados a algún catalogo en específico
Reportes	En el esquema reportes se pueden observar todas las tablas que llevan datos para la realización de algún reporte
Seguridad	Dentro del esquema de seguridad podemos observar todas aquellas tablas relacionadas a la seguridad de la base de datos

Tabla 1.

Tabla esquemas

C. TABLAS

Al realizar el esquema de base de datos se especificaron las tablas que contendría cada una de ellas de acuerdo a su función dentro de la plataforma, a cada una de las tablas se le especificaron sus campos correspondientes y se le implemento las llaves foráneas que contendría para las relaciones con otras tablas dentro de la base de datos.

A continuación, tenemos las tablas realizadas en la base de datos:

N.	ESQUEMA	NOMBRE DE LA TABLA
1	SEGURIDAD	PERMISOS
2	SEGURIDAD	ROLES
3	SEGURIDAD	PERMISO-ROLES
4	SEGURIDAD	USUARIOS
5	SEGURIDAD	SERVICIO-POZOS
6	REPORTE	COLUMNAS-GEOLOGICAS
7	REPORTE	COSTOS

8	REPORTE	DATOS-POZOS
9	REPORTE	DESVIACIONES
10	REPORTE	DIARIO
11	REPORTES	ESTATUS
12	REPORTES	ESTUDIOS
13	REPORTES	GRADIENTES
14	REPORTES	INCIDENCIAS
15	REPORTES	INTERVALOS
16	REPORTES	LECTURAS
17	REPORTES	LITOLOGIAS
18	REPORTES	MANIFESTACIONES
19	REPORTES	NUCLEOS
20	REPORTES	OBJETIVOS
21	REPORTES	PECES
22	REPORTES	REGISTROS-GEOFISICOS
23	REPORTES	SERVICIOS
24	REPORTES	TIPO-GRADIENTES
25	REPORTES	TUBERIAS-REVESTIMIENTOS
26	CATALOGOS	ACTIVOS
27	CATALOGOS	AREAS
28	CATALOGOS	CAMPOS
29	CATALOGOS	CAPAS-GEOLOGICAS
30	CATALOGOS	CLASIFICACIONES
31	CATALOGOS	COMPAÑIAS

32	CATALOGOS	COORDINACIONES
33	CATALOGOS	ESTADO-VALIDACION
34	CATALOGOS	ETAPAS-POZO
35	CATALOGOS	HERRAMIENTAS-PERFORACION
36	CATALOGOS	MACROPERAS
37	CATALOGOS	POZOS
38	CATALOGOS	PROYECTOS-INVERSION
39	CATALOGOS	REGIONES
40	CATALOGOS	ROCAS
41	CATALOGOS	SEVERIDAD-INCIDENCIAS
42	CATALOGOS	TIPOS-CAPASGEOLOGICAS
43	CATALOGOS	TIPOS-SERVICIO
44	CATALOGOS	TIPOS-ESTUDIOS
45	CATALOGOS	TIPOS-FLUIDOS
46	CATALOGOS	TIPOS-INFORMACION
47	CATALOGOS	TIPO-INTERVALOS
48	CATALOGOS	TIPOS-MANIFESTACIONES
49	CATALOGOS	TIPOS-MUESTRAS
50	CATALOGOS	TIPOS-POZOS

Tabla 2. Tablas de la base datos

D. STORED PROCEDURES

Para la lógica de negocios de la plataforma se optó por utilizar procedimientos almacenados donde se realiza el CRUD (CREATE, READ, UPDATE, DELETE) para cada una de las tablas ya descritas, esto nos ayuda a la optimización de la plataforma ya que cada una de las sentencias SQL son ejecutadas directamente en el motor de la base de datos por lo que obtenemos las siguientes ventajas:

- Optimización de la base de datos
- Reducir tiempos
- Mejor manipulación de las tablas
- Lógica de negocios en el motor de base de datos
- Visualización específica de errores

Todo esto ayuda a que la plataforma manipule directamente los datos en el motor de base de datos, así podemos reducir tiempos y ver específicamente si tenemos algún error, por lo que se reduce el tiempo y costo de mantenimientos para que la plataforma sea más óptima para el usuario y el cliente final.

3.2 MODELADO DE COMPONENTES.

Después de analizar e implementar la base de datos, se analiza la tecnología a implementar en el Back-End para la comunicación y control de los datos, por lo que se seleccionó Api-Rest para controlar la estructura de la plataforma.

Esta arquitectura contendrá 3 capas:

- ENTITIES
- CONTROLLER
- REPOSITORY

Donde entities será la capa que contendrá cada uno de los modelos de las tablas especificadas en la base de datos para tener un mejor control de los datos de entrada y salida que contendrá la plataforma, esta se comunicara con la capa Repository que es la que contiene las conexiones con la base de datos y se especifican en funciones los procedimientos almacenados a ejecutar, al ejecutar el procedimiento el repository tiene que especificar que tipo de modelo se va a utilizar para traer los datos.

La capa controller especifica cada uno de los recursos que se pueden manipular mediante los verbos HTTP y tendrá conexión directa con las otras dos capas de la plataforma ya que se puede especificar el tipo de modelo que va a traer la petición y de igual manera el tipo que la petición debe de devolver al usuario que la está solicitando

A. CLASES

Para la realización del Web Api de la plataforma tecnológica se especificaron 3 clases de tipo Controller para la manipulación de los recursos que tendrá disponible el Front-End las cuales se crearon de acuerdo con el esquema de seguridad especificado en la base de datos, por lo que se implementaron las siguientes clases:

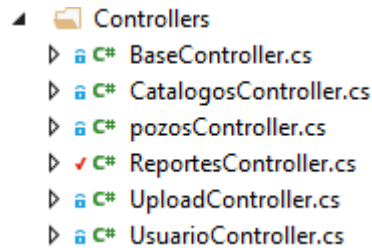


Figura 23. Controladores

Dentro de cada una de las clases de encuentras los recursos que podrán ser manipulados mediante una URI específica. Estos controladores se comunican directamente con las clases de nuestro repository, donde podemos encontrar las siguientes clases:

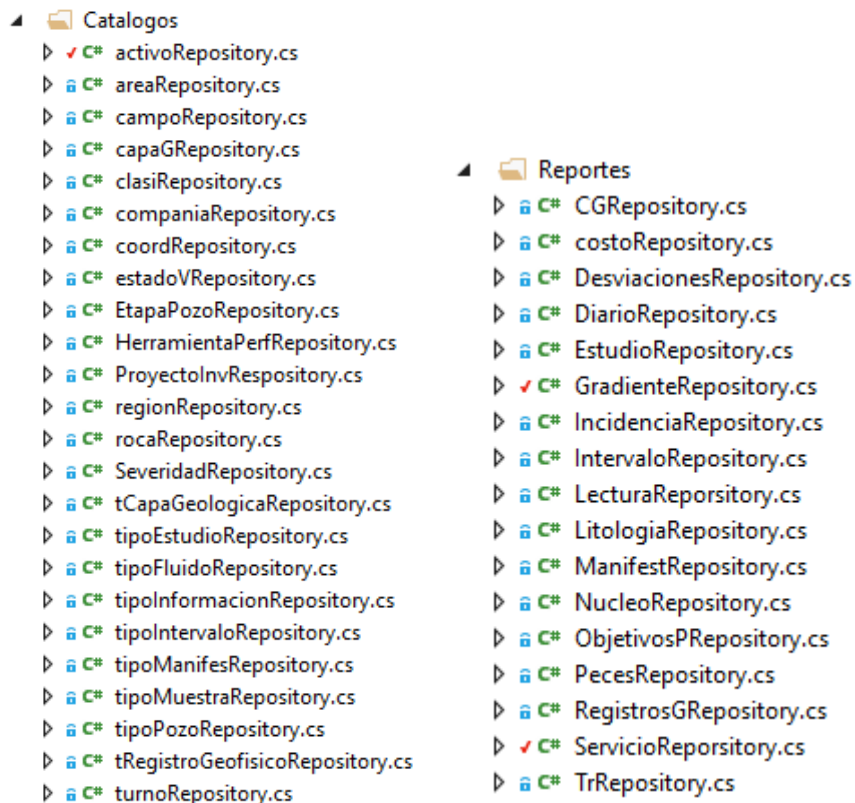


Figura 24. Repositorios

Estos fueron realizados de acuerdo a las tablas de la base de datos para así tener un nombre más específico de que contiene cada una de las clases. Estas a su vez son comunicadas con las entidades que son los modelos de los datos. En la capa entities podemos encontrar las siguientes clases:

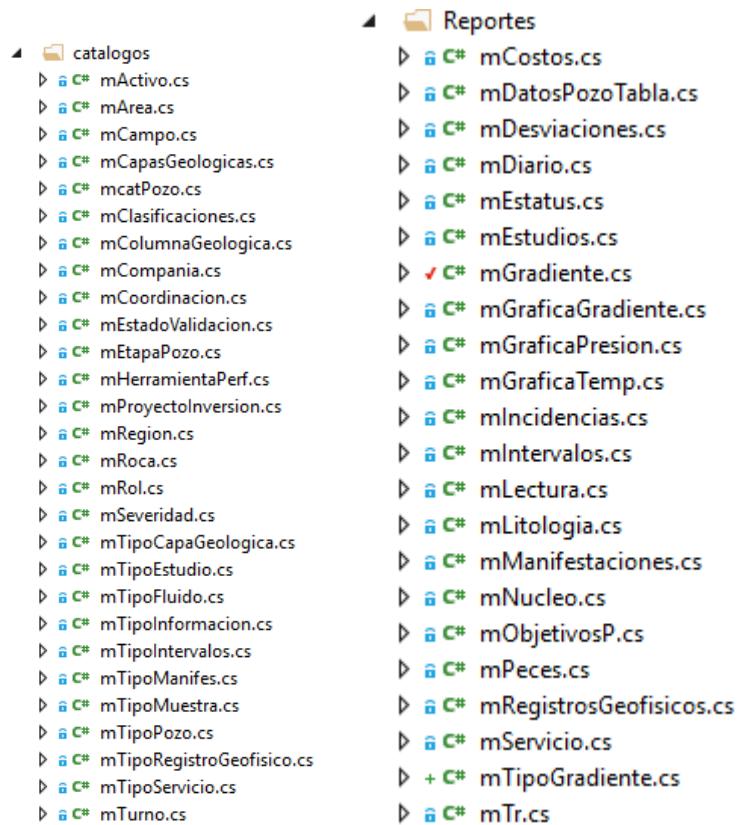


Figura 25. entities

La inicial “m” en cada una de las clases indica que es un modelo de datos que contiene todos los campos específicos con su respectivo tipo de dato.

B. SEGURIDAD

Para la seguridad de los recursos del Back-End y que estos se encuentren protegidos a las peticiones http, se seleccionó la seguridad por web tokens.

JWT (JSON Web Token) es un estándar de seguridad para las Api Rest

En un JWT se define un módulo el cual va a propagar entre la aplicación de forma segura la identidad de un determinado usuario, además podrá asignarle una serie de privilegios.

Estos privilegios pueden estar codificados en objetos que son de tipo JSON que están dentro de un payload el cual es el cuerpo del mensaje pero de forma encriptada y va firmada digitalmente.

Esto nos ayuda a proteger que un usuario no registrado y no logueado pueda acceder a nuestros recursos y de igual manera no podrá ver nuestros datos.

Para la seguridad de nuestros usuarios utilizamos contraseñas encriptadas, con la cual el usuario podrá ver en texto plano su contraseña, pero alguien que consuma algún recurso o directamente el administrador de sistemas vera solamente la contraseña encriptada con diferentes caracteres que este no podrá descifrar a menos que contenga la contraseña del usuario para ello utilizaremos encriptación SHA.

De acuerdo a la página TICsite el hasg SHA1 es un algoritmo que genera firmas, es una cadena de letras y números que es el resultado de una cadena de origen, lo principal es que no pueda verse la información en texto plano pero pueda ser decodificado en otra área para poder ver su información, este cálculo es de una sola vía así que es muy difícil de decodificar al menos que se tengas la semilla con la que fue encriptada.

Los hashes SHA1 son utilizados mayormente para almacenar de forma segura las firmas de las contraseñas, así en vez de guardar directamente la contraseña del usuario en texto plano este es guardado de forma encriptada para mayor seguridad, así se trasmite por la red y solo teniendo la semilla podrá ser decodificada.

También se utiliza un sistema de roles para restringir o dar permisos al usuario sobre la manipulación de los datos, estos son agregados por el administrador de sistema y el usuario solo tendrá los permisos que este le asigne para así poder proteger los datos que se encuentren en el sistema.

3.3 FRONT-END

Al haber creado un Api-Rest se debe seleccionar la tecnología adecuada que se adapte perfectamente al proyecto para el consumo de recursos, el manejo de datos y la visualización de la interfaz que tendrá el usuario, se opta por consumir dichos recursos en una aplicación web.

Una aplicación web es aquella donde se puede manipular datos y estos pueden ser visualizados de forma dinámica por el usuario, mediante una interfaz ya definida, a diferencia de una página web que solo contiene datos estáticos, esta tendrá diferentes funciones donde se pondrá cambiar los datos en la interfaz de forma dinámica, dichas funciones pueden ser:

- Login
- Insertar datos
- Actualizar datos
- Eliminar datos
- Gráficas

Una de las ventajas al ser usado en una aplicación web es que puede correr en cualquier computador con una conexión a internet y un navegador instalado (internet Explorer, Chrome, Firefox etc)

Para crear la aplicación web se utilizará angular que es un framework desarrollado por google que facilita el desarrollo de una aplicación web mediante componentes, estos se comunican entre sí y ayudan al desarrollador a implementar un patrón de diseño estándar para el desarrollo.

A. COMPONENTES

Un componente en angular es un paquete de clases que contiene una hoja de estilos, un archivo typescrip y un archivo html en el que prácticamente Se desarrollan cada una de las páginas que contendrá nuestra aplicación web, estas páginas serán administradas mediante un archivo de rutas e implementadas mediante un módulo.

Tenemos los siguientes componentes:

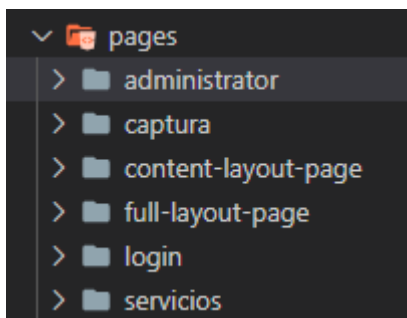


Figura 26. componentes

B. SERVICIOS

Los servicios en angular son aquellos que contienen las conexiones con la URL de los recursos de las Apis, estos son divididos dependiendo el tipo de recurso que se va a utilizar y son consumidos por lo componentes para el manejo de los datos y la visualización de los mismo.

Se agregaron los siguientes servicios a la aplicación:

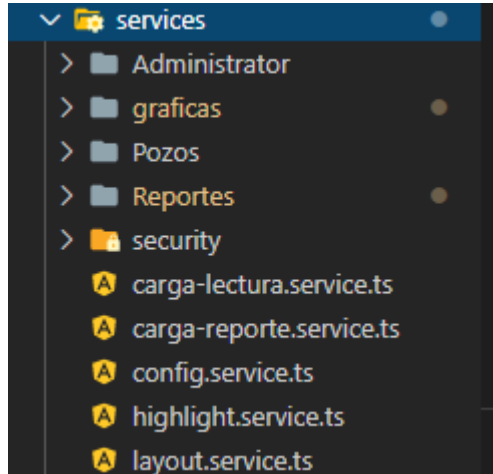


Figura 27. Tabla services

Estos son administrados en la carpeta services en nuestro desarrollo para identificarlo de manera más precisa al momento de realizar algún cambio.

C. SEGURIDAD

La seguridad de nuestra aplicación será manejada mediante GUARDS que son servicios que pueden proteger directamente un componente para que a este solo se pueda tener acceso si se cumple la condición que se establece, de igual manera se manejara la seguridad por token, que este será entregado por el Back-End y será guardado en el local storage de la aplicación.

Cada vez que se haga una petición http a nuestro Api, este evaluara el token traído por el Front-End para verificar que aun este vigente en la aplicación de no ser así el usuario será redirigido a la página de login para que vuelva a iniciar sesión y se le proporcione un token vigente para que pueda seguir utilizando la aplicación.

Tenemos los siguientes guards en nuestra aplicación.

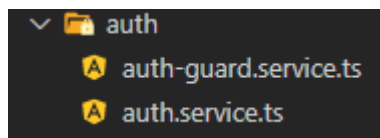


Figura 28. Guards

3.4 PRUEBAS.

Para la realización de pruebas del proyecto se utilizaron algunas herramientas ya incluidas en las herramientas de desarrollo y algunas que son independiente de estas, el fin de realizar pruebas a este proyecto se basa en que la plataforma sea estable para el usuario, y este pueda confiar que su información estará disponible y segura dentro de la aplicación.

Las pruebas se pueden realizar sobre 3 capas dentro de nuestra arquitectura:

- Base de datos
- Api Rest
- Front-End

Se debe de realizar las diferentes pruebas dentro de cada uno de estos niveles para poder identificar problemas al momento de que la aplicación esté funcionando y estos puedan ser corregidos de manera inmediata para que el usuario final no detecte ningún fallo dentro de la aplicación.

A. BASE DE DATOS

El primer nivel en el que se realizan las pruebas es a nivel de base de datos, este es muy importante ya que si llega a fallar esta capa en nuestra aplicación los datos no podrán estar disponibles para el usuario final y por lo tanto la aplicación no tendría funcionamiento.

Para la realización de las pruebas se utilizará la siguiente herramienta:

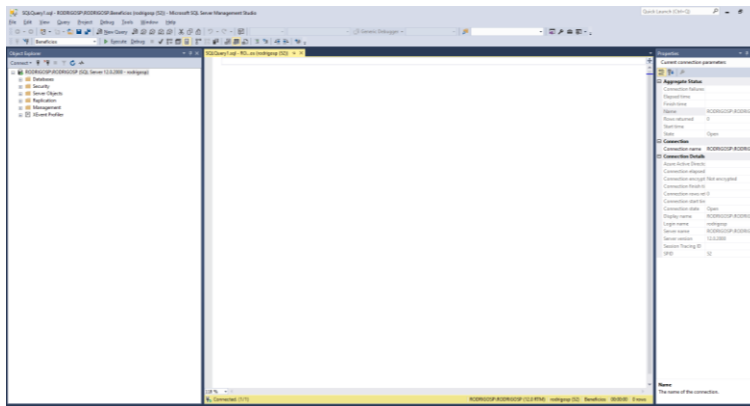


Figura 29. SQL Management Studio

SQL Management Studio es una aplicación de software lanzada por primera vez con Microsoft SQL Server 2005 que se usa para configurar, administrar y administrar todos los componentes dentro de Microsoft SQL Server . Es el sucesor del Enterprise Manager en SQL 2000 o antes. La herramienta incluye editores de scripts y herramientas gráficas que trabajan con objetos y características del servidor.

Las pruebas se realizan mediante scripts directamente en la base de datos, los scripts son pequeños fragmentos de código que pueden ser ejecutado directamente en el motor de base de datos mediante una línea de comando o una herramienta de administración.

Al realizar estos scripts se obtienen diferentes resultados, uno de los principales es que el script pudo ser ejecutado correctamente, después de esto se analiza el tiempo que se demoró el script en ser realizado y por último el resultado que se obtiene al ser ejecutado.

A continuación, se muestran algunos scripts realizados con sus resultados:

Row	Time	Message	Severity	Action
1	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
2	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
3	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
4	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
5	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
6	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
7	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
8	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
9	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
10	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information

Figura 30. Script Select * from Catalogos.Pozos

Row	Time	Message	Severity	Action
1	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
2	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
3	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
4	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
5	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
6	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
7	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
8	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
9	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
10	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information

Figura 31. Script Select * from Reporte.DatosPozo

Row	Time	Message	Severity	Action
1	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
2	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
3	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
4	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
5	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
6	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
7	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
8	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
9	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information
10	2010-01-14 10:00:00.000	SQL Server Enterprise Manager started.	0	Information

Figura 32. Script Select * from Reporte.Lecturas

Al realizar los anteriores Scripts obtuvimos los siguientes resultados de cada uno de ellos

Script	Schema	Tabla	Filas	Tiempo	Resultado
SELECT	CATALOGOS	POZOS	46	00:00:00	Successfully
SELECT	REPORTE	DATOS POZO	44	00:00:00	Successfully
SELECT	REPORTE	Lecturas	9901	00:00:01	Successfully

Tabla 3. resultados scripts

Stored Procedures:

Procedimiento: [Catalogos].[spObtenerPozos]

Query:

```
ALTER PROCEDURE [Catalogos].[spObtenerPozos]
AS
BEGIN --exec Catalogos.spObtenerPozos
SELECT p.idPozo
      ,p.idCampo
      ,c.nombre nombreCampo
      ,c.idActivo
      ,a.nombre nombreActivo
      ,a.idRegion
      ,r.nombre nombreRegion
      ,p.nombre nombrePozo
      ,p.uwi
      ,dp.campoAnalogo
      ,dp.otroCampoAnalogo
      ,dp.ubicacion
      ,dp.objetivo
      ,dp.resultado
      ,pn.idProyectoInversion
      ,pn.descripcion dscProyectoInversion
      ,cl.idClasificacion
      ,cl.descripcion dscClasificacion
      ,dp.media
      ,dp.avanceAproximado
      ,dp.elevacionMesaRotaria
      ,dp.elevacionTerreno
      ,dp.profundidadActualDesarrollada
      ,dp.profundidadActualVertical
      ,dp.profundidadProgramadaDesarrollada
      ,dp.profundidadProgramadaVertical
      ,cp.idCompania idCompaniaPerforadora
      ,cp.descripcion dscCompania
      ,ISNULL(dp.idEtapaPozo, 1) idEtapaPozo
      ,ISNULL(cep.nombre, 'Perforación') nombreEtapaPozo
      ,dp.fechaInicioPerforacion
      ,dp.fechaFinPerforacion
      ,dp.fechaInicioTerminacion
      ,dp.fechaFinTerminacion
      ,dp.diasProgramados
```

```

,dp.diasOperacion
,dp.diasEspera
,dp.equipo
,dp.movimientoEquipo
,dp.lugarMovimiento
,dp.fechaMovimiento
,dp.responsablePozo
,dp.limiteTecnico
,dp.tiranteAgua
,dp.idArea
,ar.nombre nombreArea
,co.idCoordinacion
,co.nombre nombreCoordinacion
,dp.coordenadaX
,dp.coordenadaY
    ,dp.latitud
    ,dp.longitud
,dp.idTipoPozo
,tp.descripcion descripcionTipoPozo
,p.enReporte,
    CAST(
        CASE
            WHEN EXISTS(SELECT * FROM Seguridad.ServicioPozo where
idPozo = p.idPozo)
                THEN 1
            ELSE 0
            END AS int) as tieneDatos
FROM Catalogos.Pozos p
LEFT JOIN Catalogos.Campos c ON p.idCampo = c.idCampo
LEFT JOIN Catalogos.Activos a ON c.idActivo = a.idActivo
LEFT JOIN Catalogos.Regiones r ON a.idRegion = r.idRegion
LEFT JOIN Reporte.DatosPozos dp ON p.idPozo = dp.idPozo
LEFT JOIN Catalogos.EtapasPozos cep ON dp.idEtapaPozo = cep.idEtapaPozo
LEFT JOIN Catalogos.TiposPozo tp ON tp.idTipoPozo = dp.idTipoPozo
LEFT JOIN Catalogos.ProyectosInversion pn ON pn.idProyectoInversion =
dp.idProyectoInversion
LEFT JOIN Catalogos.Clasificaciones cl ON cl.idClasificacion =
dp.idClasificacion
LEFT JOIN Catalogos.Companias cp ON cp.idCompania = dp.idCompania
LEFT JOIN Catalogos.Areas ar ON dp.idArea = ar.idArea
LEFT JOIN Catalogos.Coordinaciones co ON ar.idCoordinacion =
co.idCoordinacion
END

```

Procedimiento: [Reporte].[spObtenerCGPorPozo]

QUERY:

```
ALTER PROCEDURE [Reporte].[spObtenerCGPorPozo]
@idPozo INT
AS
BEGIN
    SELECT d.idReporte ,
           d.idPozo ,
           d.idTipoInformacion ,
           d.idEtapaPozo ,
           d.fecha ,
           d.turno ,
           d.agujero ,
           cg.idCG ,
           cg.idCapaGeologica ,
           ccg.nombre nombreCapaGeologica,
           ccg.colorHexadecimal ,
           ccg.siglas siglasCapaGeologica,
           cg.cima ,
           cg.profundidadDesarrollada ,
           cg.profundidadVerticalizada ,
           cg.esDeterminada,
           cg.idEstadoValidacion
    FROM Reporte.ColumnasGeologicas cg
         INNER JOIN Reporte.Diario d ON cg.idReporte = d.idReporte
         INNER JOIN Catalogos.CapasGeologicas ccg ON cg.idCapaGeologica =
ccg.idCapaGeologica
    WHERE d.idPozo = @idPozo
    ORDER BY d.fecha, d.turno, cg.cima
END
```

Procedimiento: [Reporte].[spObtenerRegistrosGeofisicosPorPozo]

Query:

```
USE [DB_A296F9_eGeo]
GO
/***** Object: StoredProcedure [Reporte].[spObtenerRegistrosGeofisicosPorPozo]    Script
Date: 25/06/2020 04:25:55 p. m. *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [Reporte].[spObtenerRegistrosGeofisicosPorPozo]
@idPozo INT
AS
BEGIN --exec [Reporte].[spObtenerRegistrosGeofisicosPorPozo] 3
    SELECT d.idReporte ,
           d.idPozo ,
           d.idTipoInformacion ,
           d.idEtapaPozo ,
           d.fecha ,
           d.turno ,
           d.agujero ,
           rg.idRegistroGeofisico ,
           rg.cima ,
           rg.base ,
           c.idCompania,
           c.descripcion descripcionCompania,
           rg.idEstadoValidacion,
           rg.idTipoRegistroGeofisico,
           cg.nombre nombreTipoRegistro
    FROM Reporte.RegistrosGeofisicos rg
         INNER JOIN Reporte.Diario d ON rg.idReporte = d.idReporte
         INNER JOIN Catalogos.Companias c ON c.idCompania = rg.idCompania
         INNER JOIN Catalogos.TiposRegistroGeofisico cg ON
cg.idTipoRegistroGeofisico = rg.idTipoRegistroGeofisico
    WHERE d.idPozo = @idPozo
    ORDER BY d.fecha, d.turno, rg.cima, rg.base
END
```

Resultados:

Script	Schema	Tablas	Tiempo	Resultado
[Catalogos] . [spObtenerPozos]	CATALOGOS	12	00:00:01	Successfully
[Reporte] . [spObtenerCGPorPozo]	REPORTE	3	00:00:01	Successfully
[Reporte] . [spObtenerRegistrosGeofisicosPorPozo]	REPORTE	4	00:00:01	Successfully

Tabla 4. Resultados de scripts con tablas relacionadas

(justificación de los procedimientos)

Al analizar los resultados y ver que todos fueron ejecutados correctamente se llegó a la conclusión de que la base de datos está correctamente creada y sus tablas bien definidas por que lo que opta por finalizar las pruebas en la base de datos como exitosas.

B. WEB API

Para la realización de pruebas dentro del Back-End Se utilizó La herramienta de desarrollo “Visual Estudio” específicamente su función de “debug” dentro de sus herramientas. Esta función permite a los desarrolladores ejecutar directamente una parte del código en su momento de ejecución, o realizar detenciones directamente en una parte del código para ir ejecutando las funciones por fragmentos y así poder localizar directamente algún problema ocurrido.

El IDE “Visual Studio” nos trae diferentes herramientas de desarrollo y para la realización de pruebas directamente en el código.

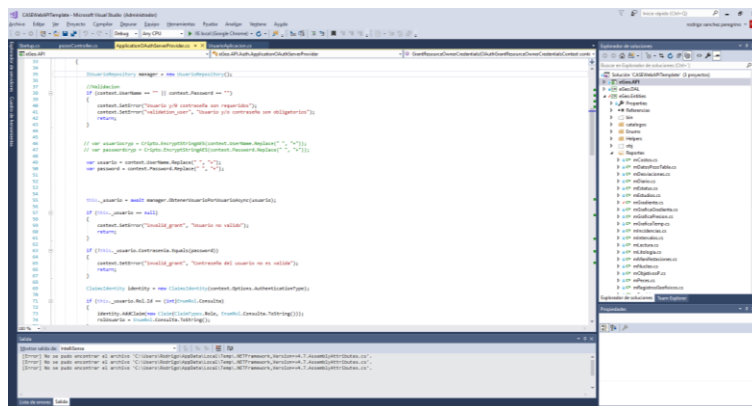


Figura 33. Visual Estudio

Microsoft Visual Studio es un entorno de programación integrada(IDE) que esta desarrollado para las plataformas Linux, Windows y MacOs, en el se puede manejar diversos lenguajes de programación

tales con c#, visual basic, f#, java, Python, ruby, php etc. Al igual se pueden crear entornos de desarrollo web como Asp .net MVC, Django, etc. Con los cuales se utilizar diferentes herramientas que nos ofrece Windows Azure en forma del editor Monaco.

De igual manera para las pruebas y para los resultados finales se realiza un “response” Básico para las peticiones http, un response es la respuesta que trae el servidor al realizar una petición http, este “response” básico esta construido con 3 valores por defecto los cuales son:

- Success: Boolean. Esta traerá el valor true en el caso de que la petición haya sido realizada correctamente y false en el caso contrario.
- Menssage: String. Este valor devolverá un mensaje directamente del servidor.
- Errors[]]. Este valor devolverá un arreglo de errores en el caso de que alguno ocurra.

Para poder ver los resultados de manera mas puntuales podemos utilizar 2 herramientas:

Swagger: es un marco de software de código abierto que es respaldado por un gran ecosistema de herramientas que facilita a los desarrolladores a diseñar, construir, documentar y consumir servicios web RestFul . Si bien la mayoría de los usuarios identifican Swagger con la herramienta Swagger UI, el conjunto de herramientas Swagger incluye documentación automatizada que se va desarrollando conforme a se crean las apis ofreciéndonos una interfaz grafica con dichos servicios.

PostMan: se crea como una herramienta que especialmente nos permite crear peticiones sobre Apis de una forma muy sencilla y controlada para asi poder, de esta manera, probarlas. Todo esta basado en una extensión de Google Chrome. El usuario de Postman puede ser un desarrollador que esté probando el funcionamiento de sus apis su comportamiento y poder administrar los diferentes servicios de una manera organizada.

Para las pruebas se realizan diferentes peticiones http a diferentes URI dentro del controlador del back end, a continuación, veremos algunas de las pruebas realizadas y sus resultados:

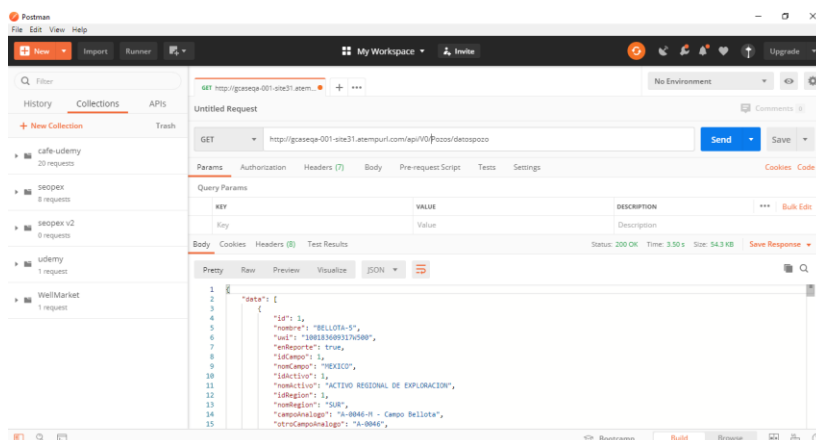


Figura 34. método get a datospozo

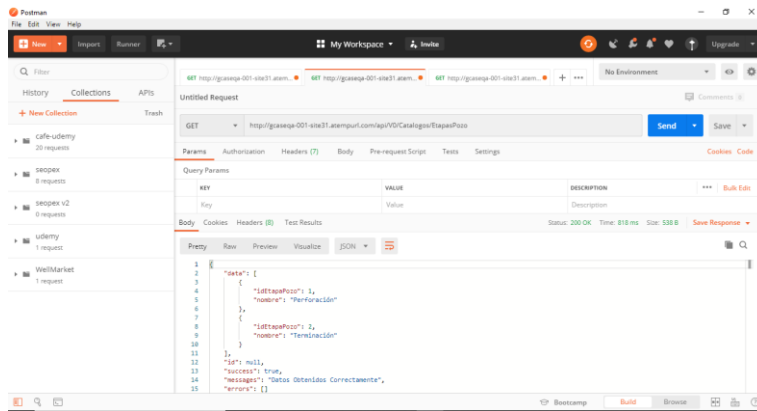


Figura 35. metodo get a estapaspozo

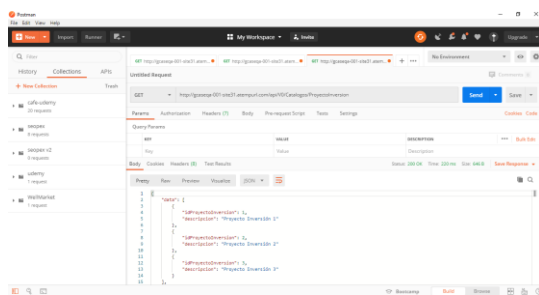


Figura 36. metodo get a proyectoinversion

Al realizar las siguientes peticiones http obtuvimos los siguientes resultados:

Método	Uri	StatusCode	Tiempo	Size	Resultado
GET	DatosPozo	200	3.50S	54.3kb	Success
GET	EtapaPozo	200	818ms	538B	Success
GET	ProyectoInversion	200	220ms	646B	Success

Figura 37. Resultados de consultas a Apis

Al analizar los resultados obtenidos se llega a la conclusión de que se implementó correctamente la programación dentro del back-end y la plataforma esta lista para poder realizar pruebas dentro del front-end.

C. FRONT-END

Para las realizaciones de pruebas en el Front-End es más estética ya que toda la lógica de negocios esta implementada dentro del back-end, sin embargo también se realizan pruebas para verificar que estén bien implementadas las peticiones http, pruebas en el servidor para visualización de la página, y funcionalidades básicas de la mismas.

Las herramientas a utilizar para realizar estas pruebas no es más que las herramientas de desarrollo de un navegador web, en el cual nos vamos a enfocar en los siguientes datos:

- Consola de comandos
- LocalStorage
- SessionStorage

Estos datos nos darán los resultados para verificar que este bien implementados los servicios, se realizaron las siguientes pruebas:

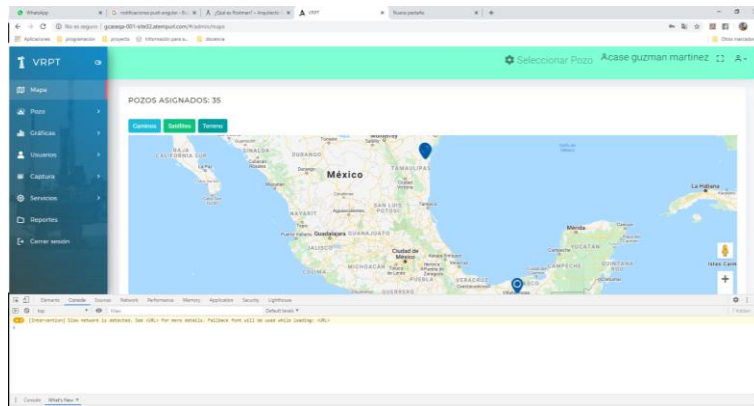


Figura 38. Componente Mapa

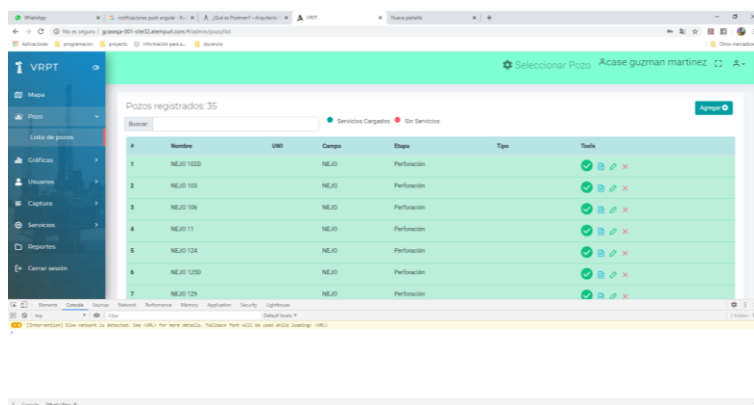


Figura 39. Componente Lista de pozos

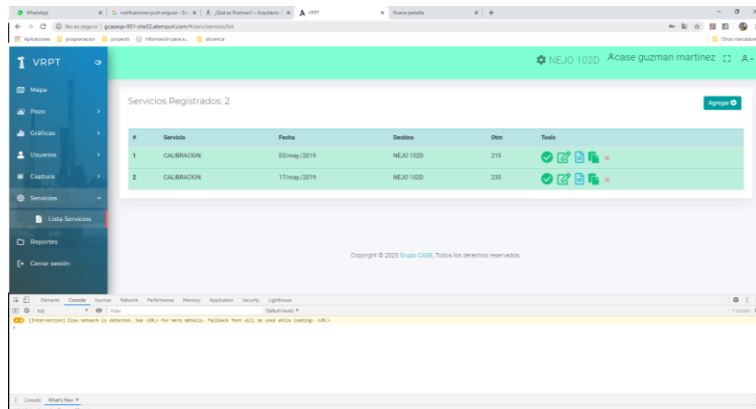


Figura 40. componente servicio

Al realizar estas pruebas obtuvimos los siguientes resultados:

Componente	Consola	LocalStorage	Resultado
MAPA	SIN ERRORES	LONGITUD,LATITUD	SUCCESS
LISTA DE POZOS	SIN ERRORES	IDPOZOS	SUCCESS
SERVICIO	SIN ERRORES	IDSERVICIO	SUCCESS

Tabla 5. Resultados de servicios

Después de realizar las anteriores pruebas se llegó a la conclusión de que la pagina web del Fron-End está correctamente implementada y sus componentes son funcionales, además sus servicios traen los datos correctos y los datos son guardados correctamente en el LocalStorage de la aplicación.

localStorage y sessionStorage son propiedades que acceden al objeto Storage y tienen la función de almacenar datos de manera local, la diferencia entre éstas dos es que localStorage almacena la información de forma indefinida o hasta que se decida limpiar los datos del navegador y sessionStorage almacena información mientras la pestaña donde se esté utilizando siga abierta, una vez cerrada, la información se elimina.

4. RESULTADO FINAL

Después de realizar todas las pruebas correspondientes se muestra el resultado final de “PLATAFORMA TECNOLÓGICA PARA EXPOTACION DE INFORMACION TECNICA PETROLERA GRUPO CASE” donde obtuvimos una plataforma implementada con una arquitectura web api, por lo que la aplicación web quedo de la siguiente manera:

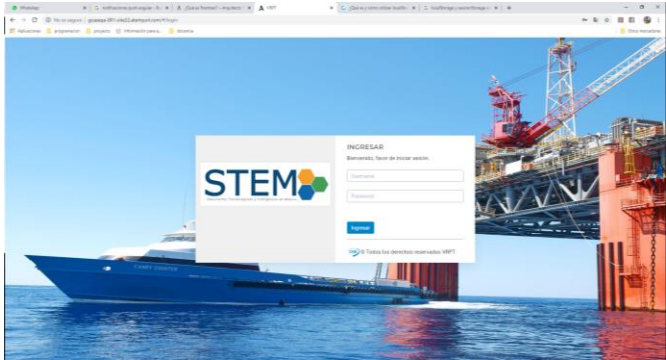


Figura 41. login

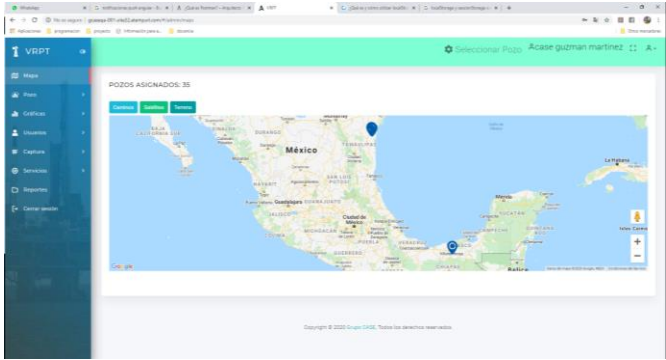


Figura 42. mapa

#	Nombre	UW	Campo	Etapa	Tipo	Hubs
1	NEJO 1000	NEJO		Perforación		📍 🔄 🗑️
2	NEJO 100	NEJO		Perforación		📍 🔄 🗑️
3	NEJO 106	NEJO		Perforación		📍 🔄 🗑️
4	NEJO 11	NEJO		Perforación		📍 🔄 🗑️
5	NEJO 104	NEJO		Perforación		📍 🔄 🗑️
6	NEJO 120	NEJO		Perforación		📍 🔄 🗑️
7	NEJO 109	NEJO		Perforación		📍 🔄 🗑️
8	NEJO 103	NEJO		Perforación		📍 🔄 🗑️
9	NEJO 104	NEJO		Perforación		📍 🔄 🗑️
10	NEJO 105	NEJO		Perforación		📍 🔄 🗑️
11	NEJO 1000	NEJO		Perforación		📍 🔄 🗑️
12	NEJO 1000	NEJO		Perforación		📍 🔄 🗑️

Figura 43. lista de pozos

VRPT | Seleccíon Pozo | Acase guzman martinez

Pozos

REGIÓN: [dropdown] ACTIVO: [dropdown] CAMPO: [dropdown] TIPO POZO: [dropdown]

NOMBRE DEL POZO: [input] UMI: [input] CAMPO ANALÓGICO: [input] OTRO CAMPO ANALÓGICO: [input]

UBICACIÓN: [input] OBJETIVO: [input] RESULTADO: [input] PROYECTO DE INVERSIÓN: [input]

CLASIFICACIÓN: [dropdown] MEDIA: [input] ANÁLISIS APRESA: [input] ELEVACIÓN MESA ROTATORIA: [input] ELEVACIÓN TERRENO: [input]

PROFUNDIDAD PROGRAMADA: [input] PROFUNDIDAD ACTUAL: [input] COMPENSA PROFUNDIDAD: [input] FECHA INICIO PERFORACIÓN: [input]

FECHA FIN PERFORACIÓN: [input] FECHA FIN TERMINACIÓN: [input] FECHA FIN TERMINACIÓN: [input] ASES PROGRAMADOS: [input]

DIAS ESPERA: [input] DIAS OPERACIÓN: [input] EQUIPO: [input] MOVIMIENTO DE EQUIPO: [input] LOCAL DE MOVIMIENTO: [input]

FECHA DE MOVIMIENTO: [input] RESPONSABLE DEL POZO: [input] LÍMITE TÉCNICO: [input] TIRANTE DE AGUA: [input] ÁREA: [input]

COORDENADAS UTM: [input] [input] [input] [input] [input] [input]

Figura 44. agregar pozo

VRPT | Seleccíon Pozo | Acase guzman martinez

Reportes Diarios Registrados: 8

#	Fecha	Turno	Etapa	Agujero	Profundidad(metros)	Demanda(galones)	Tools
1	18/sep./2016	Mañana	Perforación	1	800/0	0	[icon] [icon] [icon]
2	21/sep./2017	Mañana	Perforación	1	1300/1300	0	[icon] [icon] [icon]
3	12/may./2014	Mañana	Perforación	1	1800/0	0	[icon] [icon] [icon]
4	28/mar./2012	Mañana	Perforación	2	700/700	0	[icon] [icon] [icon]
5	13/mar./2012	Mañana	Perforación	1	500/500	1.025	[icon] [icon] [icon]
6	01/mar./2012	Mañana	Perforación	1	220/0	1.215	[icon] [icon] [icon]
7	20/feb./2012	Mañana	Perforación	1	300/0	1.41	[icon] [icon] [icon]
8	01/feb./2012	Mañana	Perforación	1	100/0	1.802	[icon] [icon] [icon]

Copyright © 2016 Grupo OAS. Todos los derechos reservados.

Figura 45. reportes diarios

VRPT | Seleccíon Pozo | Acase guzman martinez

Reportes Diario

FECHA: [input] TURNO: [input] AGUJERO: [input] PROFUNDIDAD ACTUAL: [input] PROFUNDIDAD ACTUAL: [input]

[input] [input] [input] [input] [input] [input]

[input] [input] [input] [input] [input] [input]

Copyright © 2016 Grupo OAS. Todos los derechos reservados.

Figura 46. figura agregar reportes diarios

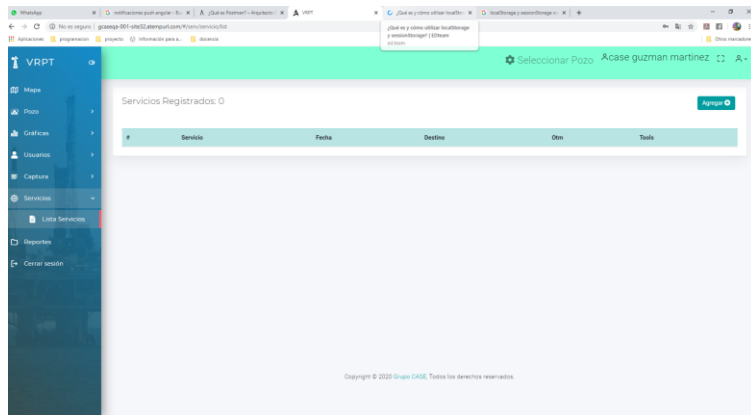


Figura 47. servicios

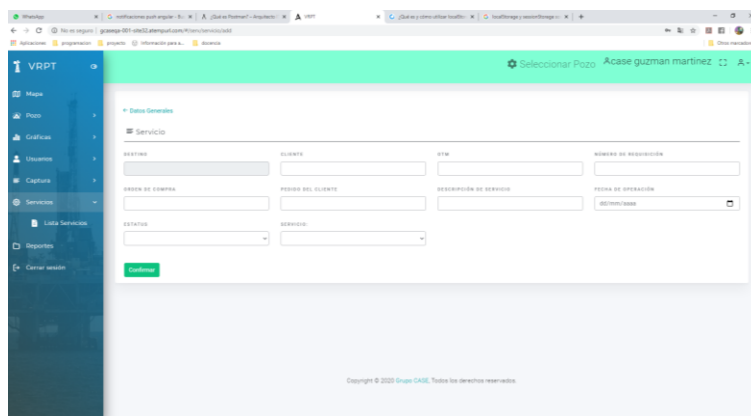


Figura 48. agregar sevicios

Estas son algunas paginas que se encuentran dentro de la plataforma, debido a que la plataforma es bastante amplia se limita a motrar estas imagenes para obtener informacion mas detallada hablar con el administrador de sistemas para explicar e informar de forma mas puntuales algunas funcionalidades que se pueden encontrar dentro de la plataforma.

CONCLUSIONES

Las empresas que innoven e incursionen en este tipo de plataforma tecnológica para la explotación de información técnica petrolera, mejorarán su desempeño en el manejo de los datos visualizados principalmente en forma de gráficos lineales, apoyando la toma de decisiones y mejorando su desempeño.

De manera generar al implementar esta plataforma tecnológica se obtienen las siguientes ventajas:

- Seguridad de los datos que se obtiene al implementar un sistema de seguridad basada en tokens garantizando la integridad y permanencia de los datos.
- Graficas que presentan el comportamiento de las variables más importantes como son presión y temperatura.
- Datos históricos en actividades de producción y perforación de un pozo petrolero, estas bases de datos estarán alojadas en tiempo real con tecnología de vanguardia (nube).

La arquitectura Api Rest empleada dentro de la plataforma permite visualizar exactamente donde se encuentra algún error dentro de la plataforma, utilizar componentes adaptables ya que al ser una arquitectura escalable se le pueden agregar componentes de otras plataformas, reducir costos de mantenimientos puntualizando los errores, así mismo esta arquitectura permite la escalabilidad del proyecto agregando módulos que se van requiriendo sin que se afecte la estructura de la plataforma.

Con la implementación de la metodología ágil Scrum se permitió una mejor comunicación con el cliente logrando el desarrollo funcional en cada uno de los procesos definidos, así como la entrega del proyecto de acuerdo a la calendarización establecida.

Con la realización de este proyecto se adquiere experiencia en el área de base de datos con la tecnología SQLSERVER gracias a la implementación de lógica de negocios el cual hace que la plataforma sea más rápida a la hora de realizar consultas a los datos.

De igual manera se adquirieron conocimientos en el área de “back-end” con la tecnología net core 2.0 de Microsoft el cual nos permitió la creación de Apis para que la plataforma ejecute solamente código de servidor en su respectiva área, esto con la finalidad de tener una arquitectura bien definida.

Por el lado del cliente se maneja la tecnología angular con la cual se obtienen conocimientos de maquetación de aplicaciones web, así como lógica del lado del cliente, al usar esta tecnología permite al cliente ejecutar código solamente en su respectiva área teniendo así una correcta estructura.

BIBLIOGRAFÍAS.

- Paola Britos y Ramón García Martínez (2018) –Modelado de Objetos-
<http://laboratorios.fi.uba.ar/lsi/rgm/articulos/R-ITBA-26-objetos.pdf>
- Banking As Service y Desarrolladores (2016) –Api Rest, Que es y Cuáles Son Sus Ventajas- <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>
- Pedro Ángel Romero (2006) -Arquitectura de Software, Esquemas y Servicios Ciudad de La Habana, Cuba
- Cecilio Álvarez Caules (2018) -Arquitecturas REST y sus niveles-
<https://www.arquitecturajava.com/arquitecturas-rest-y-sus-niveles/>
- TechBizDesing (2018) -¿Por qué fracasan hoy el 70% de los proyectos de software?-
<https://www.techbizdesign.com/biz/fracaso-proyectos-software/>
- Ester Ribas (2018) –Que es Api Rest y Porque Debes Integrarlas a tu Negocio-
<https://www.iebschool.com/blog/que-es-api-rest-integrar-negocio-business-tech/>
- Yenisleidy Fernández Romero (2018) -Patrón Modelo-Vista-Controlador.
<http://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15/10>
- JavaScript MDN Web Docs. (2020) –JavaScript-
<https://developer.mozilla.org/es/docs/Web/JavaScript>
- Erich Gamma Richard Helm Ralph Johnson John Vlissides (2015)
-Patrones de Diseño-
- Christopher Alexander (1977)
libro -pattern language-