

UNIVERSIDAD PERUANA UNIÓN
FACULTAD DE INGENIERIA Y ARQUITECTURA
Escuela Profesional de Ingeniería de Sistemas



Una Institución Adventista

**Desarrollo de un sistema web utilizando Angular y Entity
Framework asp.net Core para la gestión de horas máquina de
trabajo en E&R Contratistas Generales SRL**

Tesis para obtener el Título Profesional de Ingeniero de Sistemas

Por:

Marco Antonio Choquehuanca Quispe

Asesor:

Mg. David Mamani Pari

Juliaca, diciembre de 2020

DECLARACIÓN JURADA DE AUTORÍA DEL INFORME DE TESIS

David Mamani Pari, de la Facultad de Ingeniería y Arquitectura, Escuela Profesional de Ingeniería de Sistemas, de la Universidad Peruana Unión.

DECLARO:

Que el presente informe de investigación titulado: **"DESARROLLO DE UN SISTEMA WEB UTILIZANDO ANGULAR Y ENTITY FRAMEWORK ASP.NET CORE PARA LA GESTIÓN DE HORAS MÁQUINA DE TRABAJO EN E&R CONTRATISTAS GENERALES SRL"** constituye la memoria que presenta el Bachiller **Marco Antonio Choquehuanca Quispe** para obtener el título de Profesional de Ingeniero de Sistemas, cuya tesis ha sido realizada en la Universidad Peruana Unión bajo mi dirección.

Las opiniones y declaraciones en este informe son de entera responsabilidad del autor, sin comprometer a la institución.

Y estando de acuerdo, firmo la presente declaración en Juliaca, a los 29 días del mes de junio del año 2021



Ing. David Mamani Pari
Asesor



ACTA DE SUSTENTACIÓN DE TESIS

En Puno, Juliaca, Villa Chullunquiani, a 27 día(s) del mes de diciembre del año 2020, siendo las 28:00 horas, se reunieron en el Salón de Grados y Títulos de la Universidad Peruana Unión, Filial Juliaca, bajo la dirección del Señor Presidente del jurado: Mtro. Leonnir Henry Centurión Tula, el secretario: Mg. Abel Ángel Sullón Macalupe y los demás miembros: Mtro. Roel Dante Espinoza Spaza y el asesor Ing. David Mamani Pari

con el propósito de administrar el acto académico de sustentación de la tesis titulada: Desarrollo de un sistema web utilizando Angular y Entity Framework asp.net Core para la gestión de horas máquina de trabajo en E.R.R. Contratistas Generales SRL de el(los)/la(las) bachilleres: a) Marco Antonio Choquehuanca Quispe b)

conducente a la obtención del título profesional de Ingeniero de Sistemas (Nombre del Título Profesional)

con mención en

El Presidente inició el acto académico de sustentación invitando al (los)/a(la)(las) candidato(a)s hacer uso del tiempo determinado para su exposición. Concluida la exposición, el Presidente invitó a los demás miembros del jurado a efectuar las preguntas, y aclaraciones pertinentes, las cuales fueron absueltas por el(los)/la(las) candidato(a)s. Luego, se produjo un receso para las deliberaciones y la emisión del dictamen del jurado.

Posteriormente, el jurado procedió a dejar constancia escrita sobre la evaluación en la presente acta, con el dictamen siguiente:

Candidato (a): Marco Antonio Choquehuanca Quispe

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	
Aprobado	14	C	aceptable	Bueno

Candidato (b):

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	

(*) Ver parte posterior

Finalmente, el Presidente del jurado invitó al(los)/a(la)(las) candidato(a)s a ponerse de pie, para recibir la evaluación final y concluir el acto académico de sustentación procediéndose a registrar las firmas respectivas.

Presidente

Secretario

Asesor

Miembro

Miembro

DEDICATORIA

Con aprecio y gratitud a mis padres Remigio Choquehuanca Llacho y Valentina Quispe Ccamercco por su apoyo, motivación y confianza, que fueron necesarios para el logro de este trabajo de investigación.

Todo este trabajo ha sido posible gracias a ellos.

AGRADECIMIENTOS

El logro de esta meta académica es razón de mostrar mi sincero agradecimiento a quienes me acompañaron durante el camino de la obtención de conocimientos tecnológicos que se manifiesta en el informe de mi proyecto de investigación.

A Dios por la bendición que me dio de formarme académicamente en una institución superior donde primero está los valores y principios.

A toda mi familia, por el apoyo incondicional y motivación constante.

Al Mg. David Mamani Pari por su asesoría, guía y esfuerzo en la elaboración de este trabajo de investigación.

ÍNDICE

DECLARACIÓN JURADA DE AUTORÍA DEL INFORME DE TESIS.....	ii
ACTA DE SUSTENTACIÓN.....	¡Error! Marcador no definido.
DEDICATORIA	iv
AGRADECIMIENTOS	v
ÍNDICE.....	vi
ÍNDICE DE TABLAS	x
ÍNDICE DE FIGURAS.....	xiii
ÍNDICE DE ANEXOS.....	xvi
ABREVIATURAS Y ACRÓNIMOS	xvii
RESUMEN	xviii
CAPÍTULO I. El Problema.....	19
1.1. Descripción del Problema.....	19
1.2. Justificación de la Investigación.....	21
1.3. Presunción Filosófica	23
1.4. Objetivos	24
1.4.1. Objetivo general	24
1.4.1. Objetivos específicos.....	24
CAPÍTULO II: Marco Teórico.....	25
2.1. Revisión de la Literatura	25
2.1.1. Artículos relacionados	25
2.1.2. Tesis relacionadas.....	30

2.2. Marco Teórico	33
2.2.1. Marco de Desarrollo.Net	33
2.2.2. Asp.Net Core	33
2.2.2.1. Ventajas de ASP.NET Core.....	34
2.2.3. Framework Web Api	35
2.2.4. C#.....	36
2.2.5. SQL Server.....	37
2.2.5.1. Consultas rápidas.	37
2.2.5.2. Mejor seguridad.....	38
2.2.5.3. Mayor disponibilidad.....	38
2.2.5.4. Motor de base de datos mejorado.	39
2.2.5.5. Acceso a datos más amplio.	39
2.2.6. Angular.....	40
2.2.6.1. Módulo.	40
2.2.6.2. Componentes.	41
2.2.6.3. Servicios e inyección de dependencia.....	42
2.2.7. Metodología Extreme Programming (XP).....	43
2.2.7.1. Ventajas.	47
2.2.7.2. Aplicaciones de XP.	48
2.2.8. Entity Framework.....	48
2.2.8.1. Características.....	49
2.2.8.2 Elementos.	51

2.2.9. Sistemas de gestión y control	52
2.2.9.1. Reconocimiento de entidad.....	52
2.2.9.2. Gestión de máquinas de construcción.....	52
2.2.9.3. Productividad.....	53
2.2.9.4. Rendimiento.	53
2.2.9.5. Ubicaciones y actividades.....	53
CAPÍTULO III. Materiales y Métodos.....	55
3.1. Descripción del Lugar de Ejecución.....	55
3.2. Materiales e Insumos.....	55
3.3. Arquitectura de Solución.....	56
3.4. Metodología	57
3.4.1 Tipo de investigación	57
3.4.2. Desarrollo de la investigación según la metodología XP	57
3.4.2.1. Uso de metodología XP Personalizado	57
3.4.2.2. Planificación.....	58
3.4.2.1.1. Identificación de requerimientos:.....	58
3.4.2.1.2. Historia de usuarios	60
3.4.2.3. Diseño	80
3.4.3.2.1. Prototipos.....	81
3.4.3.2.2. Diagrama de Clases	85
3.4.2.4. Codificación	85
3.4.2.3.1. Base de datos	85

3.4.2.3.2. ASP .Net Core y Entity Framework.....	86
3.4.2.3.3. Servicio Web Api y solicitudes Http Angular.....	87
3.4.2.3.4. Módulos Angular.....	89
3.4.2.3.5. Componentes Angular.....	90
3.4.2.3.6. Código fuente de la lógica del módulo Parte Diario.....	91
3.4.2.5. Pruebas del sistema.....	93
3.4.2.4.1. Pruebas Unitarias.....	93
3.4.2.4.2. Casos de Prueba de Aceptación.....	98
3.4.2.4.3. Pruebas de Aceptación con Selenium.....	111
3.4.2.4.4. Pruebas de concurrencia con Jmeter.....	121
CAPÍTULO IV. Resultados.....	128
4.1. Especificar los requerimientos para el sistema web de horas máquina en la empresa.....	128
4.2. Desarrollar el sistema web para la gestión de horas máquina.....	128
4.3. Desplegar y realizar pruebas de funcionalidad del sistema web de horas máquina.....	129
CAPÍTULO V. Conclusiones y Recomendaciones.....	130
5.1 Conclusiones.....	130
5.2. Recomendaciones.....	131
REFERENCIAS.....	132
ANEXOS.....	136

ÍNDICE DE TABLAS

Tabla 1 Artículos Relacionados.	26
Tabla 2 Tesis Relacionadas.	30
Tabla 3. Materiales del proyecto.	55
Tabla 4. Crear Operario.	60
Tabla 5. Elaborar Diseño y Formulario Nuevo Operario.	61
Tabla 6. Modelo de Base de Datos Operario y Mapeo.	61
Tabla 7. Crear Datos Operario en la Base de Datos.	62
Tabla 8. Editar Operario.	62
Tabla 9. Interface Editar Operario.	63
Tabla 10. Guardar Operario Editado.	63
Tabla 11. Listar Operario.	64
Tabla 12. Diseño e Interfaz Lista Operario.	65
Tabla 13. Mostrar Lista de Operario.	65
Tabla 14. Crear Máquina.	66
Tabla 15. Diseño e Interfaz Crear Máquina.	66
Tabla 16. Modelo de Base de Datos Máquina y Mapeo.	67
Tabla 17. Insertar Máquina.	67
Tabla 18. Editar Máquina.	68
Tabla 19. Interface Editar Máquina.	68
Tabla 20. Guardar Máquina Editado.	69
Tabla 21. Listar Máquina.	69
Tabla 22. Diseño e Interfaz de Lista Máquina.	70
Tabla 23. Listar Máquina.	70
Tabla 24. Crear Parte Diario.	71
Tabla 25. Diseño e Interfaz Crear Parte Diario.	72

Tabla 26. Modelo Base de Datos y Mapeo.....	72
Tabla 27. Guardar Nuevo Parte Diario.	73
Tabla 28. Editar Parte Diario.....	73
Tabla 29. Interfaz Editar Parte Diario.....	74
Tabla 30. Interfaz Editar Parte Diario.....	74
Tabla 31. Listar Parte Diario.....	75
Tabla 32. Diseño e Interfaz Lista Partes Diarios.	76
Tabla 33. Lista de Parte Diario.....	76
Tabla 34. Gestión de Obras.	76
Tabla 35. Listar Obras.....	77
Tabla 36. Nueva Obra.	78
Tabla 37. Editar Obra	78
Tabla 38. Gestión de Consumo.	78
Tabla 39. Listar Consumo.	79
Tabla 40. Nuevo Consumo.....	80
Tabla 41. Editar Consumo.....	80
Tabla 42. Prueba 1 – Historia de usuario 1: Crear operario.....	99
Tabla 43. Prueba 2 – Historia de usuario 2: Editar operario.	100
Tabla 44. Prueba 3 – Historia de usuario 3: Listar operario.....	101
Tabla 45. Prueba 4 – Historia de usuario 4: Crear máquina.	102
Tabla 46. Prueba 5 – Historia de usuario 5: Editar máquina.....	103
Tabla 47. Prueba 6 – Historia de usuario 6: Listar máquina.	104
Tabla 48. Prueba 7 – Historia de usuario 7: Crear parte diario.....	105
Tabla 49. Prueba 8 – Historia de usuario 8: Editar parte diario.	106
Tabla 50. Prueba 9 – Historia de usuario 9: Listar parte diario.....	107
Tabla 51. Listar, crear y editar obras.	108
Tabla 52. Listar, crear y editar consumos.....	109

Tabla 53. Características de un Parte Diario.....	121
---------------------------------------------------	-----

ÍNDICE DE FIGURAS

Figura 1: Arquitectura de solución.	56
Figura 2: Metodología XP personalizado.....	58
Figura 3: Nuevo Operario.....	81
Figura 4: Gestión de Operario.....	82
Figura 5: Nueva Máquina.	82
Figura 6: Gestión de Máquinas.	83
Figura 7: Nuevo Parte Diario.....	83
Figura 8: Gestión de Parte Diario.....	84
Figura 9: Diagrama de clases gestión de horas máquina.	85
Figura 10: Entidad Relación Gestión de Horas Máquina.	86
Figura 11: Instalación de Entity Framework.	87
Figura 12: Comando Scaffolding para mapear los modelos.	87
Figura 13: Controlador operario, Web Api y sus protocolos.....	88
Figura 14: Servicio operario Http angular.	88
Figura 15: Módulo raíz AppModule angular.....	89
Figura 16: Módulo OperarioModule angular.	90
Figura 17: Componente OperarioComponent angular.	90
Figura 18: Código fuente del registro de Parte Diario.	91
Figura 19: Código fuente de editar Parte Diario.	92
Figura 20: Código fuente de lista de los partes diarios.	93
Figura 21: Prueba unitaria registrar operario en Postman.	94
Figura 22: Prueba unitaria editar operario en Postman.....	94
Figura 23: Prueba unitaria listar operario en Postman.....	95
Figura 24: Prueba unitaria registrar máquina en Postman.....	95
Figura 25: Prueba unitaria editar máquina en Postman.....	96

Figura 26: Prueba unitaria listar máquina en Postman.	96
Figura 27: Prueba unitaria registrar parte diario en Postman.	97
Figura 28: Prueba unitaria editar un parte diario en Postman.....	97
Figura 29: Prueba unitaria listar parte diario en Postman.	98
Figura 30: Formulario para crear un nuevo operario.....	99
Figura 31: Formulario para editar un operario.	100
Figura 32: Vista de registros de los operarios.	101
Figura 33: Formulario para crear una nueva máquina.	102
Figura 34: Formulario para editar un operario.	103
Figura 35: Vista de registros de las máquinas.....	104
Figura 36: Formulario para crear un nuevo parte diario.	105
Figura 37: Formulario para editar un parte diario.	106
Figura 38: Vista de registros de los partes diario.	107
Figura 39: Vista de registros de las obras.	108
Figura 40: Formulario de nueva Obra.....	109
Figura 41: Formulario de editar obra.	109
Figura 42: Vista de registros de los consumos.....	110
Figura 43: Formulario de nuevo consumo.	111
Figura 44: Formulario de editar consumo.....	111
Figura 45: Script lista de partes diarios.....	112
Figura 46: Resultado de script listar parte diarios en Selenium.	112
Figura 47: Script nuevo parte diario.....	113
Figura 48: Resultado de nuevo parte diario en Selenium.	113
Figura 49: Script editar parte diario.....	114
Figura 50: Resultado de editar parte diario en Selenium.	114
Figura 51: Script lista operarios.	115
Figura 52: Resultado de lista de operarios en Selenium.....	115

Figura 53: Script nuevo operario.....	116
Figura 54: Resultado de nuevo operario en Selenium.	116
Figura 55: Script editar operario.....	117
Figura 56: Resultado editar operario en Selenium.	117
Figura 57: Script listar máquina.	118
Figura 58: Resultado de listar máquina en Selenium.....	118
Figura 59: Script nueva máquina.	119
Figura 60: Resultado de nueva máquina.	119
Figura 61: Script editar máquina.	120
Figura 62: Resultado de editar máquina en Selenium.	120
Figura 63: Grupo de Hilos de lista de partes diarios.	122
Figura 64: Resultados de peticiones de lista de partes diarios.....	122
Figura 65: Informe Agregado de lista de partes diarios.....	123
Figura 66: Resultados de peticiones de nuevos partes diarios.....	123
Figura 67: Informe Agregado de nuevos partes diarios.	124
Figura 68: Grupo de Hilos del operario.....	124
Figura 69: Resultado de peticiones del operario.	125
Figura 70: Informe Agregado del operario.	125
Figura 71: Grupo de Hilos de la máquina.	126
Figura 72: Resultados de peticiones de la máquina.....	126
Figura 73: Informe Agregado de la máquina.	127

ÍNDICE DE ANEXOS

Anexo 1: Mapic.....	136
Anexo 2: Escenario de prueba de la usabilidad	137
Anexo 3: Resultados de pruebas de la usabilidad.	142
Anexo 4: Reportes del sistema web.....	145

ABREVIATURAS Y ACRÓNIMOS

- EF: Entity Framework
- XP: Programción Extrema
- IoT : Internet de las cosas
- API: Interfaz de Programación de Aplicaciones
- JSON: Notación de Objetos de JavaScript
- HTTP: Protocolo de Transferencia de Hipertexto
- SQL: Lenguaje de Consulta Estructurada
- OWIN: Interfaz Web abierta para el estándar .Net
- UDF: Funciones definidas por el usuario
- OLTP: Online Transaction Processing
- ORM: Objeto Mapeador Relacional
- NER: Reconocimiento de Entidad Nombrada

RESUMEN

En la empresa Contratistas Generales E&R dedicada al servicio de alquiler de máquinas de construcción se presentan deficiencias en el área administrativa donde registran todas las fichas diarias horas máquina de cada operador y hay problemas de redundancia, también tienen un excesivo tiempo al ordenar y dar reportes de las máquinas según los operadores y obras. Frente a este problema la finalidad de esta investigación es el desarrollo de un sistema de gestión de horas máquina de trabajo.

Para el desarrollo del proyecto de investigación se utilizó la metodología XP que cuenta con las fases de planificación donde se identificaron los requerimientos e historias de usuario; en el diseño se realizó un bosquejo y diseños de interfaz. En la parte del código, se utilizó la plataforma de desarrollo Visual Studio, las herramientas de .Net Core, Entity Framework, Sql Server y Angular; por último se realizó las pruebas de funcionalidad del sistema con Selenium obteniendo como resultado scrips automatizados; también se realizó las pruebas de rendimiento con Jmeter, listando más de 200 registros y creando Partes Diarios con una muestra de 70 usuarios, se obtuvo tiempos de respuesta a cada petición de los usuarios representada en milisegundos y el rendimiento en segundos, los resultados fueron; lista de partes diarios su media 968ms, mín 547ms, máx 1416ms y rendimiento 45,6sec; nuevo parte diario su media 47ms, mín 20ms, máx 217ms y rendimiento 71,2sec.

Palabras Clave: Sistema Web, Angular, Entity Framework, Asp.Net Core.

CAPÍTULO I. El Problema

1.1. Descripción del Problema

La empresa E&R Contratistas Generales brinda el servicio de alquiler de máquinas a empresas que realizan obras de construcción públicas y privadas. También realiza contratos de propias obras de construcción en lo cual alquila y usa sus mismas máquinas. Dicha empresa realiza una ficha diaria de horario de cada máquina según reporte de su operador, debido a que el importe económico de cada máquina es por horas de trabajo.

La situación problemática de la empresa actualmente está en la redundancia o incongruencias de los registros ingresados en Microsoft Excel, generando falta de control y gestión de las fichas entregadas por los operadores, además en el área administrativa ocasiona un exceso de esfuerzo al ordenar y dar reportes del trabajo de las máquinas.

A parte de la pérdida económica por el trabajo excesivo en el área administrativa, hay descontento en la falta de información oportuna (horas de trabajo realizadas) de sus partes diarios entregados que dará respaldo a su remuneración en planilla. Se muestra clara falta de organización de las fichas diarias de las máquinas, por lo que se pretende mejorar mediante un sistema web que brinde mejor organización y reportes requeridos por la empresa.

Roque (2018) describe que en el área de administración recibe el reporte de un proceso documentado y registro de producción manual, por la cantidad de máquinas de juego; este procedimiento tiene muchas entradas de datos y dificultad en el manejo de la gestión, acarreado tiempos excesivos por cada horario de trabajo del personal.

Según Roldan (2018) menciona que en la empresa de Pastas Comarrico SAS existen falencias en el ingreso de datos en el protocolo de producción y selección de datos de las máquinas, causando perdidas de formatos registrados manualmente.

La gestión de información en las empresas es importante en el medio mundial, si hablamos de un sistema que gestione datos con el propósito de lograr objetivos tomando en cuenta el estado de la información de la sociedad, obtenemos un control automatizado de tareas individuales, en esta etapa la gestión de la información proporciona una transformación permanente del entorno empresarial e identifica un negocio exitoso, es decir, el incremento de los ingresos primarios de la generalidad de las empresas por el cual muestra el incremento de las tecnologías de información a comienzos del ciclo XXI. Así nosotros podemos asumir que la gestión de información es el hoy y el mañana de la economía. El crecimiento de la economía tiende a la gestión de información que se lleva a través de las redes electrónicas de un control de información. (Eroshkin et al., 2017)

En las empresas industriales y textiles peruanas hay un grado de desorden en la gestión y control de máquinas, Enrique y Fernández describen la ineficiencia de máquinas en una empresa textil, un conteo de paros de dichas máquinas que se detienen sin un debido control disminuyendo el rendimiento del producto y descontento a clientes. Logrando un control de las máquinas al instalar un sistema web, identificando las causas en detener la producción y en consecuencia disminuyendo esos espacios. (Enrique & Fernández, 2015)

En la constructora Tafca Ltda. existen maquinarias cuyo trabajo es seleccionado y se establece un tiempo para la marcha de la obra de construcción, pero cambian continuamente las asignaciones de las tareas según sea necesario para la obra; estos cambios en el trabajo de las máquinas, se realizan con la experiencia en obras pasadas. Al comienzo se tiene una asignación inicial y que es guardada en un documento escrito

pero las modificaciones realizadas no son registradas en el documento, esto manifiesta que no vemos ningún tipo de gestión actual y las máquinas trabajan en tareas que no le fueron asignadas, lo cual puede provocar, atrasos en tareas diferentes, también hay pérdida de materiales y hasta maquinarias. (Brito & Mayne, 2008)

Los casos de estudio de los autores mencionados muestran una relación en sus necesidades; buscan el control y gestión de información de las máquinas, en todos los casos se realizan registros manuales que provocan problemas y consecuencias similares que ya se han descrito. De la misma forma ocurre en la empresa E&R Contratistas Generales que muestra la importancia del caso de estudio de este proyecto.

1.2. Justificación de la Investigación

La empresa Contratistas Generales E&R, al mantener una gestión de partes diarios en Excel, registrando incongruencia en horas de las máquinas y operarios; ha ocasionado una administración incorrecta de los datos, mostrando disconformidad de los operarios en las horas de trabajo o alquiler de las máquinas y recurriendo a un sobreesfuerzo del administrador. En consecuencia, se necesita mejorar la gestión de las horas máquina de manera efectiva, con herramientas o estrategias que faciliten el registro de partes diarios con veracidad y orden.

Se ve la necesidad de buscar herramientas y estrategias para el registro de horas de trabajo de las máquinas en las empresas de construcción, porque facilita la gestión de las máquinas ya sea de contrato o alquiler; aunque el registro de parte diarios parezca simple, influencia en el inventario de máquinas, contrato de obras y la contabilidad según los operarios.

Por la mejora continua en el área de administración de la empresa Contratistas Generales E&R, se justifica el desarrollo de un sistema para la gestión de horas máquina como propuesta para el registro de partes diarios y dar facilidad información de los datos.

La importancia de desarrollar un sistema para la empresa Contratistas Generales E&R, es de mejorar en el área administrativa, en la rapidez de registro de los partes diarios de cada máquina, organizando y gestionando las horas de trabajo de cada máquina. Otros puntos en el que ayudará, será en reportes de horas máquina, manejo de máquinas por operador y máquinas de trabajo en la obra. Todo esto mejorará la atención al cliente y así tener una mejor organización en la información de las obras de construcción, en consecuencias a la población y la comunidad a través de la ejecución de las obras de manera eficiente y comprometida. Por otra parte, ayudará al control del mantenimiento de máquinas y también en el cobro de servicios por el alquiler de dichas máquinas.

El sistema web Contratistas Generales se desarrolló con asp.net Core que tiene una comunidad activa y de código abierto, que se desenvuelve en la plataforma de .NET con herramientas y librerías precisamente para elaborar las aplicaciones web. También se usará Entity framework para el mapeo de entidades del sistema y desarrollo del software orientado a datos y por último usaremos angular para las vistas, donde solo se cargue el código requerido al presentar la vista que solicitan.

Enrique & Fernández (2015) Enfatiza en los beneficios de una aplicación tecnológica de control y gestión, al aplicar una sección en el monitoreo de máquinas textiles. Dicho resultado tubo mejoras a la anotación de paros breves de las máquinas en beneficio de los empleados.

Dehai (2019) muestra la importancia de la gestión de equipos con recolección de datos en las máquinas térmicas, que se componen de múltiples unidades por lo que hay muchos dispositivos que necesitan ser revisados. El ejemplo tradicional de gestión de información requiere la cooperación en varios empleados y múltiples departamentos y tienen la desventaja de una gran carga de trabajo y perdida estadística de datos. El

objetivo del sistema de información del equipo. Es gestionar las tareas diarias del equipo en una manera ordenada usando efectivamente los datos de entrada, dar evaluación integrada de los equipos, personal y repuestos, comprendiendo con precisión el estado operativo del equipo. El sistema de gestión integra eficazmente varios módulos, haciendo que sea eficiente el trabajo del equipo.

1.3. Presunción Filosófica

El incremento de la tecnología y uso de ella en la gestión de la información, es una experiencia que llevé durante toda mi estadía académica en la escuela de Ingeniería de Sistemas. Las habilidades y conocimientos inculcados por docentes profesionales siempre estuvieron presente los valores y principios cristianos, el cual reconozco que Dios es el quien otorga sabiduría y entendimiento.

La legítima educación no da por menos el valor del intelecto científico, pero considera que la autoridad es mayor a la información, la generosidad al poder y el carácter al pensamiento intelectual. Este mundo no necesita personas de gran conocimiento de carácter bondadoso, se precisa personas cuya aptitud este dirigido por principios fijos. (Elena G. de White)

Dios es el escultor de la ciencia. En busca de conocimiento las mentes abren grandes campos de discernimiento e investigación, instruyéndonos para ver las obras de la creación. La desinformación puede intentar dar mayor valor a la incredulidad apelando a la ciencia; a pesar de todo, la ciencia muestra evidencias de la sabiduría y la potestad de Dios. Con el conocimiento adecuado de la ciencia y la palabra de Dios muestran veracidad respaldándose así mismo. (Elena G. de White).

En cuanto la persona acepta que la fuente de todo conocimiento está en Dios, dejando guiar nuestra mente y conducta atreves de sus Escritos, podemos estar seguros a esta promesa: “Yo honraré a los que me honran”. 1 Samuel 2:30.

1.4. Objetivos

1.4.1. Objetivo general

Desarrollar un sistema web para la gestión de horas máquina de trabajo utilizando Angular y Entity Framework asp.net Core en la Empresa E&R Contratistas Generales SRL.

1.4.1. Objetivos específicos

Especificar los requerimientos para el sistema web de horas máquina en la empresa.

Desarrollar el sistema web para la gestión de horas máquina.

Desplegar y realizar pruebas de funcionalidad del sistema web de horas máquina.

CAPÍTULO II: Marco Teórico

2.1. Revisión de la Literatura

2.1.1. Artículos relacionados

Tabla 1

Artículos Relacionados.

Autor	Año	Título	Objetivos	Método	Resultados
Chu Dehai	2019	(Dehai, 2019)“Development and Research of Equipment Information Management System of Thermal Power Plant”	(Dehai, 2019)“Desarrollar y la construcción de sistemas y utiliza El sistema de gestión de información del equipo adopta una arquitectura cliente / servidor para facilitar El intercambio y la transmisión. El sistema se divide en cliente, servidor y cliente móvil.”	(Dehai, 2019)“Servidor de PC adoptó un programa C # y el cliente de telefonía móvil adoptó el lenguaje del escaneo bidimensional códigos por teléfonos móviles, realizando transmisión en línea de datos en el sitio y procesamiento en tiempo real de servicios, y mejorar la eficiencia de procesamiento de asuntos laborales.”	El sistema de gestión integra eficazmente varios módulos, haciendo que la gestión del equipo trabajar juntos de manera eficiente. Adquisición de información del equipo y estado operativo a través del escaneo bidimensional códigos por teléfonos móviles, realizando transmisión en línea de datos en el sitio y procesamiento en tiempo real de servicios, y mejorar la eficiencia de procesamiento de asuntos laborales.
Douglas Burdick Ronald Fagin Phokion G. Kolaitis Lucian Popa Wang-Chiew Tan	2019	(Burdick et al., 2019)“Expressive Power of Entity-Linking Frameworks”	Crear enlaces entre registros que representan entidades del mundo real que están relacionadas de cierta manera. La vinculación de entidades ha sido reconocida como un problema computacional fundamental que	Teorema de una condición suficiente para la trazabilidad	En este artículo, presentamos y exploramos un enfoque unificador de la entidad. enlace. Este enfoque, que se basa en la noción de un enlace de entidad marco y la noción de ciertos enlaces en dicho marco, proporciona un

		ha sido investigado por varias comunidades de investigación diferentes	formalismo único para modelar diferentes escenarios de vinculación de entidades y para comparar ellos usan ciertos enlaces como una medida de su poder expresivo.
Viacheslav E. Wolfengagen Larisa Yu Ismailova Sergey V. Kosikov	2018	(Wolfengagen et al., 2018)“Modelo de conversión de objetos de datos para definir el mapeo de relación de objeto.”	(Wolfengagen et al., 2018)“posible construir un mapeo relacional de objetos en orden parcial con red semántica”
Cheng, Chieh Feng Rashidi, Abbas Davenport, Mark A. Anderson, David V.	2017	(Cheng et al., 2017)“Activity analysis of construction equipment using audio signals and support vector machines”	En este artículo, proponemos un sistema innovador basado en audio para el análisis de actividad (y seguimiento) de equipos pesados de construcción. Audio signals and support vector machines Se ha comparado el rendimiento de diferentes máquinas con el algoritmo de señales de audio de dichas máquinas, por lo cual se propone el marco presentado, novedoso ya que es el primer plan de investigación que intenta introducir el audio como una fuente alternativa de información para reconocer y registrar actividades de construcción en un sitio de trabajo.

Adela Vásquez-García, Dora Ma. Sangerman-Jarquín y Luis Reyes Muro	2017	(Vásquez García et al., 2017)“Developmapplication to evaluate agricultural crops through the MAP method”	(Vásquez García et al., 2017)“Desarrollar unaanálisis de aplicación web, con una interfazpolíticas MAP” amigable, cuenta con una base de datos amigable de confianza para la gestión de datos, este sistema analiza cultivos agrícolas con la metodología de análisis política”	(Vásquez García et al., 2017)“La aplicación web desarrollada se denominó sistema de evaluación a través de la MAP (SIEMAP) facilitará a los usuarios registrar datos de cultivos agrícolas que se deseen revisar”
S.Yu. Eroshkina, N.A. Kamenevaa, D.V. Kovkovb, A.I. Sukhorukova	2017	(Eroshkin et al., 2017)“Conceptual system in the modern information management”	Se propone la sistematización del aparato conceptual y categorico que se revela en el ejemplo del uso de sistemas de información en la gestión de acuerdo con su jerarquía. Se justifica la creación de un aparato unificado conceptual y categorico de gestión de información.	La estructura jerárquica de los sistemas de información en la gestión, que cubre todo espectro de tareas de automatización de la empresa desde la gestión estratégica hasta los niveles de control tecnológico procesos de producción (construcción) y producción (construcción) de gestión de productos (característica) en todo el ciclo vital. Estos sistemas pueden basarse en diferentes enfoques metodológicos, lo que hace que la estructura propuesta sea bastante flexible.
Joe Cheri Ross Aditya Joshi Pushpak Bhattacharyya	2016	(Ross et al., 2016)“A Framework that Uses the Web	(Ross et al., 2016)“Introducir un marco para la identificación de clase de entidad con nombre entity recognition para una entidad dada, usando(NER)”	Presentamos un marco para realizar esta tarea de identificación. Nuestro marco ayuda a configurar una plataforma

<p>Santiago Videla Carito Guziolowski Federica Eduati Sven Thiele Martin Gebser Jacques Nicola Julio Saez-Rodriguez Torsten Schaub Anne Siegel</p>	<p>2015</p>	<p>for Named Entity Class Identification: Case Study for Indian Classical Music Forums” (Videla et al., 2015)“Learning Boolean logic models of signaling networks with ASP”</p>	<p>la web cuando se conocen los límites de la entidad. El marco propuesto será beneficioso para dominios especializados donde existen desafíos de datos y etiquetas de clase. (Videla et al., 2015)“Este artículo aborda el problema que consiste en aprender de un conjunto de conocimientos previos que describe las interacciones causales y las actividades de fosforilación en un estado lógico pseudoestable, modelos lógicos booleanos de respuesta inmediata temprana en redes de transducción de señalización”</p>	<p>para la identificación de clase de entidad con nombre de entidades que hacen uso de recursos web, ignorando el contexto de las entidades.</p>
<p>Philadelphia University, Department of Software Engineering, Jerash, Jordan</p>	<p>2015</p>	<p>“An Approach to Modeling Web Services Datatype Descriptions”</p>	<p>Producir descripciones de tipo de datos vagas, personalizadas e inconsistentes por diferentes desarrollos de servicios web técnicas dentro de los lenguajes de descripción de servicios web autogenerados.</p>	<p>servicios web Los servicios web son importantes para desarrollar e integrar aplicaciones basadas en web, incluido el comercio electrónico aplicaciones. Las descripciones de los tipos de datos de los servicios web aún enfrentan muchos problemas, como la producción vaga, personalizada.</p>

2.1.2. Tesis relacionadas

Tabla 2

Tesis Relacionadas.

Autor	Año	Título	Objetivos	Método	Resultados
Diego Armando Quelal Enríquez	2019	(Quelal, 2019) "Desarrollo del sistema académico aplicando la herramienta asp.net Core 2 para la escuela Eufrasia Pelletier"	(Quelal, 2019) "Realizar una aplicación web para el procedimiento académico en la Escuela Eufrasia Pelletier, utilizando la herramienta multiplataforma ASP.NET CORE 2"	(Quelal, 2019) "Extreme Programming"	(Quelal, 2019) "ASP.NET CORE asegura la función de los módulos y muestra una interfaz eficaz con el usuario. Muestra la ventaja de tener frameworks integrados que ayudan al desarrollo de la aplicación"
César Augusto Otoya Florián	2015	(Otoya Florián, 2015) "Implementación de una aplicación móvil para el monitoreo de contenido y disponibilidad de servicio web, servicio ftp y páginas web"	(Otoya Florián, 2015) "Desarrollar una aplicación web que permita el reconocimiento y seguimiento de los servicios web, páginas web y servicios FTP y le brinde la información de los datos analizados a la aplicación móvil"	(Otoya Florián, 2015) "EL producto no se utilizará una metodología establecida, sin embargo, se utilizarán algunas herramientas que permitirán organizar y establece los lineamientos del proyecto"	(Otoya Florián, 2015) "Se logró el desarrollo de una aplicación móvil que permita dar seguimiento servicios de carácter dinámico. La estructura utilizada permite que el usuario pueda en cada pestaña desarrollar una actividad específica, donde la primera pestaña "Servicios" permite crear, eliminar y editar servicios de información, presenta datos estadísticos que dan recorrido al desempeño del usuario"

Christian Pérez Ortiz	2015	(Ortiz, 2015)“Análisis, diseño e implementación de una guía gastronómica para la administración y ubicación de restaurantes en entorno web”	(Ortiz, 2015)“Analizar, diseñar e implementar una guía gastronómica para la ubicación de restaurantes mediante la web, la cual permita un análisis automatizado de las críticas de los comensales.”	(Ortiz, 2015)“Para la gestión del proyecto a elaborar, se utilizarán ciertos principios del PMBOK”	(Ortiz, 2015)“Se ha desarrollado una arquitectura para la plataforma web que le permitirá a esta ser sostenible y escalable en el tiempo ya que, a través de las distintas vistas de arquitectura, se entiende cómo debe moverse el sistema lógicamente y, conjuntamente, agrupado en componentes o módulos”
Gustavo Villar García	2015	(García, 2015) “Implementación de un sistema de gestión de la investigación para especialidad de Ingeniería Informática de la PUCP”	(García, 2015)“Ejecutar un sistema que gestione la investigación para la sección de Ingeniería Informática de la PUCP.”	(García, 2015)“El desarrollo de la aplicación Web tomará algunas de las prácticas de la metodología Extreme Programming (XP) que se pueden aplicar para el proyecto”	(García, 2015)“El sistema web desarrollado brinda la interfaz para administrar cómodamente la información de las investigaciones. Los módulos principales del sistema logran la organización de personas, grupos, proyectos y conclusiones. Así los investigadores llevan un registro de sus funciones y avance.”

José Luis Enrique Antaurco, Ecmias Eduardo Fernández Galvéz	2015	(Enrique & Fernández, 2015) “Implementación de un sistema web para el control de paros de las máquinas textiles en la fábrica de tejidos San Carlos S.A.C.”	(Enrique & Fernández, 2015) “Implementación de un sistema capaz de aumentar la eficiencia de las maquinas textiles, garantizando una adecuada supervisión de paros en la Fábrica de Tejido San Carlos S.A.C”	(Enrique & Fernández, 2015) “Buenas prácticas establecidas PMBOK y metodología Scrum”	(Enrique & Fernández, 2015)“Se logró desarrollar e implementar un sistema que permita el reconocimiento de paros mayores con sus causas y su respectivo procedimiento; como asimismo la implementación del componente de monitoreo de las maquinarias en tiempo existente. Se ha logrado reducir el tiempo de registro de espacios detenidos de las máquinas artesanales”
-------------------------------------------------------------	------	----------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2.2. Marco Teórico

2.2.1. Marco de Desarrollo.Net

.Net es una plataforma open source, multiplataforma y gratuita para construir varias clases de aplicaciones en los lenguajes de software como C#, F# y Visual Basic de tal manera que su código se aplicará de forma nativa en cualquier sistema operativo relacionado; .NET tiene diferentes formas de implementarse para el trabajo pesado, entre esas implementaciones tenemos .NET Core, .NET framework y Xamarin / Mono.

Los proveedores de servicios web (desarrolladores) dependen de las técnicas, herramientas, marcos de desarrollo de servicios web para implementar y publicar sus servicios web. Hay muchos servicios de desarrollo web técnicas que se basan en diferentes plataformas como: la plataforma .NET de Microsoft, Oracle (EE). Microsoft Visual Studio se utiliza para construir servicios web basados en la plataforma .Net; el equipo de desarrollo de software para desarrollar e implementar servicios en .NET se llama Windows Communication Foundation. (Hanna, 2016)

2.2.2. Asp.Net Core

ASP.NET Core, es un escenario multiplataforma de gran utilidad de código abierto, con alto beneficio en la elaboración de aplicaciones en relación a la nube, servicios web, asimismo aplicaciones IoT (internet de las cosas) y backends móviles, también el manejo al desarrollar con las herramientas preferidas de Windows, MacOS y Linux. ASP.NET Core se puede ejecutar en la nube de Microsoft y servicios de las varias plataformas que se muestran en su página. (Roth et al., 2019)

Existe una composición apegada entre las herramientas para programar de Visual Studio y el lenguaje de C# que tiene buen fundamento a una programación orientada a objetos. Con el lenguaje y herramientas mencionadas se realizó una estructura jerárquica

en la navegación que va de inicio de sesión, registro de cultivos y módulos financieros.
(Vásquez García et al., 2017)

2.2.2.1. Ventajas de ASP.NET Core.

Multiplataforma.

Una aplicación tiene diferentes tipologías y asp.net Core tiene las herramientas precisas para usar el mismo código en varias plataformas. Hoy por hoy, una aplicación en asp.net Core se puede efectuar en: Windows, Linux y macOS; brindando facilidad de uso en ellos donde ya no se recrea algún cambio al implementar el arreglo o la APIs.

Código abierto.

La fundación .Net subvenciona a este marco de trabajo, dicho así su código fuente es accesible en GitHub. Al ser una herramienta de código abierto, ASP .NET Core dispone que la elaboración sea más rápida y clara, para la comunidad comprometida este pendiente de los cambios.

Desarrollo flexible.

Para crear una aplicación su desarrollo se efectúa con una implementación autocontenida y la construcción referente a un marco. La construcción referente a un marco, solamente requiere la instalación de las dependencias y su solución con dicha herramienta en el sistema. Una implementación autocontenida se compila según la versión de la herramienta que va parejo con las dependencias y la solución, se puede elaborar distintas intenciones.

Modular.

.Net core es un sistema modular y a diferencia del framework .Net estándar, estos módulos deben incluirse uno a uno. Estas dependencias se definen en el archivo de

proyecto .csproj y deben descargarse utilizando otro símbolo del sistema. Esto es necesario si agrega o elimina dependencias durante el desarrollo, pero no es estrictamente necesario inmediatamente después de crear una nueva aplicación porque el nuevo comando lo ejecuta automáticamente. (Chiaretta, 2018)

Asp.Net Core se puede publicar en NuGet, que son paquetes de modelos ensamblados de manera concisa aún más que las funciones generales, Asp.Net Core tiene características que proporcionan una estructura en la fabricación de la aplicación con mejor seguridad y forma los paquetes NuGet, reduciendo costos ahorrando en el mantenimiento y brinda comodidad al desarrollador. (Quelal, 2019)

Servidores web.

Las aplicaciones hechas con Asp.Net Core se pueden usar en un entorno .NET Core o .NET Framework, administrador de internet Information Services y Information Server Express correspondiente. (Anderson, 2017)

2.2.3. Framework Web Api

Con la llegada de dispositivos se necesita un punto final web para dar servicio a cualquier tipo de contenidos (JSON, XML, Imágenes, PDF) a cualquier tipo de cliente. Cualquier código que solicite una petición HTTP es un cliente potencial para un punto final web. El framework Web Api se ofrece para dar solución a la demanda de capas de los servidores capaces de presentar la interfaz RESTful e interactuar con cualquier cliente HTTP sin suposiciones y restricciones. El marco Web Api es un conjunto alternativo de clases para crear puntos finales HTTP diseñados para ser conscientes de la sintaxis y semántica HTTP completas. El framework web api ofrece una interfaz de programación casi idéntica a ASP.NET MVC; incorpora controladores, rutas y modelos, pero los ejecuta dentro de un entorno de ejecución completamente nuevo. Con la API web asp.net, el punto de crear un marco web desacoplado del servidor web comenzó a echar raíces, y

esto llevó a la definición de la Interfaz Web Abierta para el estándar .Net (OWIN), donde especifica las reglas para que un servidor web y una aplicación web interoperen. Con OWIN, el segundo objetivo original de asp.net, la unión fuerte y ajustada entre el alojamiento web y la aplicación web fue descartado como obsoleto (Esposito, 2018).

2.2.4. C#.

C# es un lenguaje de programación con orientación a objetos que comenzó con Anders Hejlsberg, quien también creó el lenguaje de Delphi y Turbo Pascal. El propósito fue mejorar los lenguajes de C y C++.

Algunas de las características del lenguaje de programación C# son: Su código se puede tratar íntegramente como un objeto. Su sintaxis es muy similar a la del JAVA. Es un lenguaje orientado a objetos y a componentes. Armoniza la productividad del Visual Basic con el poder y la flexibilidad del C++. Ahorramos tiempo en la programación ya que tiene una librería de clases muy completa y bien diseñada.

C# se hizo parte de la plataforma de .Net, Visual Basic utiliza esta plataforma, pero se creó por otras razones. Visual Basic se hizo con el motivo de facilitar el aprendizaje para los principiantes y C# se agregó para realizar las operaciones que Visual Basic no proporcionaba como métodos virtuales, sucesión, sobrecarga de módulos. (Informática, 2015).

C# en iniciación del desarrollo de software da cabida para desarrollar módulos muy factibles ya que se podía utilizar un ambiente dividido por lo que se puede decir que tiene:

El código fuente se puede utilizar en distintas plataformas.

Flexibilidad a que desarrolladores de C y C++ migren a este nuevo lenguaje.

Cavidad para distintos países.

Cifra aplicaciones de distinto tamaño: como sistemas operativos de gran tamaño y funciones pequeñas de menor tamaño.

Costo económico en las aplicaciones según su proceso y memoria.(Quelal, 2019)

2.2.5. SQL Server

Es un sistema de estructura de Datos relacional de importancia para guardar, procesar y analizar datos es SQL Server. Se ejecuta en Transact - SQL (T-SQL), viene a ser un grupo de extensiones de programación, que añaden varias funciones al estándar SQL, como declarar variables, transacciones, manejo de errores y procesamiento de filas. SQL Server también permite procedimientos almacenados que se definirán. Las funciones son un tipo especial de procedimientos almacenados. Aceptan parámetros, realizan algún tipo de acción y devolver un resultado. Las funciones hacen todo esto sin efectos secundarios, las consultas que contienen (UDF) Funciones Definidas por el Usuario son ampliamente utilizadas, ya que permiten que las consultas se escriban utilizando una combinación de imperativos construcciones de lenguaje y SQL, aumentando así el poder expresivo de SQL; además fomentan la modularidad, y facilitar la comprensión de las consultas. (Gubalova & Medvedova, 2018)

2.2.5.1. Consultas rápidas.

Cuando los usuarios quieren datos, los quieren lo más rápido que pueda dárselos. Microsoft SQL Server 2016 incluye varias opciones para habilitar más rápido consultas. Las tablas optimizadas para memoria ahora admiten aún más rápido en OLTP (online transaction processing), con un mejor rendimiento como resultado de nuevas operaciones paralelizadas. Para el análisis al cargar datos, se puede tomar ventaja de índices de almacén agrupados en columnas, optimizadas a archivos logrando consultas que son

hasta cien veces más rápidas. El motor de base de datos no solo es mejor y más rápido en SQL Server 2016, pero las mejoras en el motor de Analysis Services también ofrecen mayor rapidez rendimiento tanto para modelos multidimensionales como tabulares. Cuanto más rápido tú puede entregar datos a sus usuarios, más rápido podrán usar esos datos para hacer mejores decisiones para su organización.(Varga et al., 2016)

2.2.5.2. Mejor seguridad.

SQL Server presenta tres características de seguridad principales: siempre seguridad encriptada, seguridad a nivel de fila y enmascaramiento dinámico de datos. En la seguridad encriptada proporciona manejo a los administradores para cifrar datos confidenciales dentro de una aplicación, no tiene que revelar el cifrado en la base de datos o servidor.(Mukherjee, 2019)

Seguridad a nivel de fila le permite configurar tablas de modo que los usuarios vean solo las filas dentro del tabla a la que les concede acceso. Esta característica limita qué filas se devuelven al usuario, independientemente de la aplicación que estén usando, aplicando automáticamente un predicado a la consulta.(Varga et al., 2016)

El enmascaramiento dinámico de datos evita que los usuarios eliminen el acceso a los datos para visualizarlos. Esto es una característica de seguridad complementaria y la necesidad de ser utilizado con otras funciones de seguridad como auditoría, cifrado o seguridad a nivel de fila.(Mukherjee, 2019)

2.2.5.3. Mayor disponibilidad.

En un mundo que siempre está en línea, mantener el tiempo de actividad y optimizar las operaciones de mantenimiento para sus aplicaciones, esta misión crítica es más importante que nunca. En SQL Server las capacidades de los grupos de disponibilidad AlwaysOn continúan evolucionando, lo que le permite proteger los datos de

manera más fácil y flexible y con un mayor rendimiento en sistemas de almacenamiento modernos. Además, los grupos de disponibilidad AlwaysOn y las instancias de clúster de conmutación por error AlwaysOn ahora tienen mayor seguridad, confiabilidad y escalabilidad, tiene más opciones para administrar mejor los clústeres y el almacenamiento.(Varga et al., 2016)

2.2.5.4. Motor de base de datos mejorado.

En versiones anteriores de SQL Server, Microsoft se ha centrado en áreas específicas de mejora. En SQL Server 2005, el motor de almacenamiento era nuevo. En SQL Server 2008, el énfasis estaba en la consolidación de servidores. Ahora, en SQL Server 2016, puede encontrar una funcionalidad mejorada en todo el motor de base de datos. Microsoft administra más de un millón de base de datos en SQL Server, existe Microsoft Azure SQL Database que puede responder más rápidamente a las oportunidades para mejorar el producto y validar esas mejoras de manera integral antes de agregar funciones a la versión local de SQL Server.(Varga et al., 2016)

2.2.5.5. Acceso a datos más amplio.

A medida que el costo de almacenar datos continúa disminuyendo y la cantidad de formatos de datos comúnmente utilizados por las aplicaciones continúa cambiando, necesita la capacidad tanto de administrar el acceso a los datos históricos de manera relacional como de integrar sin problemas los datos relacionales con datos semiestructurados y no estructurados. Actualmente SQL Server incluye varias características nuevas que respaldan este entorno en evolución al brindar acceso a una variedad más amplia de datos. La introducción de tablas temporales de la base de datos le permite mantener datos históricos, gestionar de forma transparente aquellos cambios y recuperar fácilmente los valores de los datos en un momento determinado. Además, SQL Server le permite importar datos de notación de objetos JavaScript (JSON) en el

almacenamiento relacional, exportar datos relacionales como estructuras JSON e incluso analizar, agregar o filtrar datos JSON.(Varga et al., 2016)

2.2.6. Angular

Angular es una plataforma y marco front-end de código abierto basado en TypeScript dirigido por el equipo de Angular en Google que se utiliza para crear las aplicaciones web del lado del cliente. Angular en sí está escrito en TypeScript, que es un superconjunto de JavaScript y se utiliza para el desarrollo front-end.(Leo, 2020)

Las aplicaciones creadas con Angular se separan en plantillas HTML, clases de componentes escritas para gestionar las plantillas y los servicios que contienen la lógica empresarial de la aplicación. Este enfoque modular, facilita el desarrollo rápido y lógico de las aplicaciones y facilita sus pruebas. (Korva, 2016)

Angular sigue una arquitectura basada en componentes, su vista y lógica empresarial. Tales componentes deben ser agregado a las declaraciones en un módulo principal, luego asignado a la vista, es decir, archivo .html y es renderizado por el navegador. Angular proporciona muchas funciones integradas, como inyección de dependencia, validación de formularios, internacionalización, capacidades fuera de línea, mecanismos de seguridad, API HTTP simplificada y tipo seguridad (TypeScript). Por lo tanto, es una buena opción para aplicaciones escalables, aunque tiene su flujo de trabajo específico. Al principio, este flujo de trabajo puede verse como una desventaja sobre otros marcos, pero este tipo de ecosistema maduro es increíblemente importante lanzando un software estable (Danielecki, 2019).

2.2.6.1. Módulo.

En Angular un módulo es la agrupación de componentes, directivas, pipes y servicios, que pertenece a la aplicación. En caso de que esté desarrollando un sitio web,

la sección de encabezado, pie de página, izquierda, centro y derecha formar parte de un módulo. Se puede cambiar las características del módulo en NgModule; el cual se genera de forma predeterminada al crear el proyecto con el comando `–cli`, que se encuentra en `app.module.ts`. (TutorialsPoint, 2019)

Tenemos módulos angulares o podemos decir NgModules que mantiene el modularidad de las aplicaciones angulares. Cada aplicación Angular tiene al menos un módulo como módulo raíz. Convencionalmente, se denomina AppModule. Normalmente, una aplicación contiene muchos módulos funcionales. En angular, cualquier módulo es una clase con el `@NgModule` decorador. (Leo, 2020)

2.2.6.2. Componentes.

Un componente contiene la lógica a desarrollar con los datos de una clase, el componente está asociado a un elemento html en una plantilla, dedicado a una tarea en particular. Para esto, normalmente también necesitará tener alguna lógica vinculada a esa plantilla, para completar datos y reaccionar a eventos. Sus componentes se organizarán de forma jerárquica, como el DOM. Un componente raíz tienen componentes secundarios, cada uno de ellos también tendrá hijos, etc. (Devica, 2016)

Cada aplicación angular tiene un componente raíz, es decir, AppComponent que es bootstrapped dentro del módulo principal. En angular, los componente definen una clase con los datos y la lógica que se desarrolla para la aplicación; mostrando en la plantilla HTML el entorno de destino. (Leo, 2020)

Con esa mirada de alto nivel a los componentes de Angular en nuestro haber, veamos dos propiedades que se pueden pasar al `@Component` decorator para implementar el flujo de datos hacia abajo y hacia arriba: "inputs" y "outputs" (Aden, 2018).

Existen directivas en la plantilla que realizan la lógica del componente y la conexión de datos y DOM; son dos tipos de conexión para los datos (data binding).

Enlace de datos se usan llaves `{{}}`; este proceso se llama interpolación. Enlace de eventos cuando un usuario interactúa con una aplicación en forma de un movimiento de teclado, un clic del mouse o un mouseover, genera un evento.

Estos eventos deben manejarse para realizar algún tipo de acción. Aquí es donde entra en escena la vinculación de eventos (TutorialPoint & Wall, 2019).

2.2.6.3. Servicios e inyección de dependencia.

Los desarrolladores continuamente repiten su código en varias partes de la página. Por ejemplo, los componentes comparten sus datos entre ellos para sus conexiones. Esto se logra con la ayuda de servicios, podemos ingresar a métodos y propiedades en todos los componentes. El módulo inyectable se importa desde `@angular/core`. Contiene el método `@Injectable` y una clase con sus respectiva función de servicio; primero debemos adjuntar el servicio en el archivo principal `app.module.ts`, posteriormente crear el servicio adjuntado y así se pueda usar en cualquier componente, utilizando su función.(TutorialPoint & Wall, 2019)

Enrutamiento.

El enrutamiento básicamente significa navegar entre páginas. Se ven muchos sitios con enlaces que dirigen a una nueva página, esto se puede lograr mediante el enrutamiento. Aquí las páginas que se refieren será en forma de componentes.(TutorialPoint & Wall, 2019)

2.2.7. Metodología Extreme Programming (XP)

XP es un modelo de desarrollo de software que comprende principalmente en las mejoras de calidad del software y la capacidad de resultados a las necesidades cambiantes del cliente. Es un tipo de metodología de desarrollo de software ágil, Aboga por entregas constantes en pequeños ciclos de desarrollo. Estas versiones están destinadas a mejorar la productividad y calidad del software mediante la introducción de ciertas medidas que se centran en iniciar ciertos puntos de control en los que se pueden adoptar y cumplir los requisitos del cliente.(Yadav et al., 2019)

El primer proyecto de XP inició un 6 de marzo de 1996. XP es uno de muchos procesos ágiles conocidos, que ha demostrado tener mucho éxito en empresas, industrial de toda parte del mundo.(Wells, 2013)

Wells (2013) afirma que la XP tiene éxito por las entregas de avance al cliente en vez de ver una fecha muy lejana, las entregas son correspondiente a la necesidad del uso del software. XP demuestra una respuesta cómoda al cambio de las necesidades de los clientes y los desarrolladores.

Como lo hace notar Wells (2013) la metodología XP realiza el trabajo en equipo. Aumentando la productividad gracias al entorno colaborador creado entre los clientes y desarrolladores. Este entorno se organiza sólo, buscando resolver la necesidad de manera eficiente y asertiva.

XP usa la práctica de Historias de Usuario donde el interesado describe en tarjetas la necesidad de la aplicación, las características que la aplicación debe realizar, pueden ser requerimientos funcionales o no funcionales, todo en presencia del desarrollador. (Quelal, 2019).

En el desarrollo se encuentra la retroalimentación que va dirigido al interesado y a la misma aplicación; que consta en la revisión del diseño para asegurar su funcionamiento según lo establecido. La comunicación es importante para definir si cumple o debe corregirse; así mismo, causar una retroalimentación ágilmente. (García, 2015).

XP brinda sencillez a los cambios del cliente, pero también influye a los desarrolladores para mejorar su reutilización de código, mediante ello dimitir a un lado las peticiones del cliente según lo que esperaba en los posteriores días. Logrando comodidad en la organización y bajo costo económico. (García, 2015).

Está claro que XP es una metodología de desarrollo de sistemas muy eficaz. Sin embargo, de experiencia personal es un desafío formar un equipo que pueda adoptar todas estas prácticas para trabajar de manera efectiva juntos. Teniendo en cuenta que XP se centra en las personas, a veces tal metodología puede no ser apropiado porque el cliente comercial no está preparado para sentarse en la oficina con los desarrolladores para proporcionar retroalimentación constante. De lo contrario, cuando todos los miembros del equipo están ansiosos por trabajar, XP hace que proyecte un viaje agradable, asegurando que se entregue una solución correcta a tiempo (Bulavko, 2017).

Bulavko (2017) Menciona que generalmente todas las metodologías de desarrollo de software siguen una línea enfoque en el que cada fase se completa en orden y, una vez finalizada, se inicia la siguiente. Sin embargo, XP combina todas las fases en el transcurso del ciclo de vida del desarrollo de la aplicación y cada persona involucrada lleva a cabo fases en varios órdenes. Este enfoque es muy valorado por la flexibilidad que aporta. Estas son las fases del ciclo de vida que XP ha desarrollado:

Planificación: en esta etapa, el equipo de desarrollo y los clientes se reúnen para crear historias de usuarios.

Diseñar: crear soluciones breves a los problemas descritos.

Codificación: conversión de diseños en código real considerando la propiedad colectiva del código.

Pruebas: probar y eliminar errores, así como garantizar que se realicen las pruebas de aceptación del cliente.

Escuchar: obtener comentarios sobre componentes específicos y discutir el potencial mejoras con el equipo para maximizar la eficiencia.

Yadav et al. (2019) menciona 12 prácticas básicas que siempre son implementado. Estas prácticas garantizan una mejor legibilidad del código y comprensibilidad y también ayuda a mejorar comunicación entre el desarrollador y los clientes para que se hace un mejor producto:

Juego de planificación: Esto involucra al desarrollo y planeación de las publicaciones de los planes para el proyecto y las entrevistas entre los interesados y desarrolladores. Entre las partes interesadas asegurar que las consultas estén resueltas y que tanto el desarrollador y el interesado tiene un conocimiento adecuado del flujo del proyecto y funcionamiento del sistema.

Versión pequeña: Los desarrolladores se organizan mediante funciones, una vez completo se entrega al interesado para una retroalimentación de parte suya. Está hecho ayuda a disminuir errores y riesgos.

Metáfora: es un documento que describe el funcionamiento del sistema y expresa la visión del proyecto en evolución que definiría el alcance y propósito del sistema. Las metáforas son directamente derivadas de los principios y estándares del proyecto arquitectura y requisitos.

Pruebas: XP en pruebas frecuentes; también se realizan pruebas de aceptación. Verifica los requisitos tal como los entiende el desarrollador y si cumple o no con los requisitos del usuario. También se realizan pruebas de integración que ayudan a probar el sistema cuando se integran diferentes funciones del sistema en una unidad. Inicialmente se ejecuta en códigos pequeños y luego todo el sistema.

Refactorización: Es una forma para que el código fuente se sostenga en una forma fácil de comprender con mejor presencia y un código fácil. En esta práctica se reestructura el código; no ha métodos largos y código repetido.

Programación de dos: Como el nombre lo dice dos programadores trabajan en una función, mientras uno desarrolla el código de la función el otro verifica el código escrito. Mejora la confianza y seguridad del código escrito y determina que la eficiencia general se reduce si dos diferentes personalidades están trabajando juntas en un módulo específico en una ubicación diferente. Por otro lado, los programadores con los rasgos similares tienden a comprender y comunicarse fácilmente.

Propiedad colectiva: Se refiere a que el equipo completo de desarrolladores debe conocer el código del sistema; de esta manera podrán hacer cambios del código en el momento que necesiten hacerlo, también se le conoce como código abierto.

Trabajo de 40 horas: Las horas de trabajo aseguran la calidad del sistema y el equipo debe establecerlas. Lo importante es reconocer el tiempo acordado por todos los miembros del equipo para el número de horas que debe dedicar durante una semana. Después de cada semana, se revisará el trabajo realizado.

On-Site Cliente: Disponible para responder a todas las preguntas de los desarrolladores, resolver disputas y establecer a pequeña escala prioridades todo el tiempo.

Estándares de codificación: Para garantizar la propiedad colectiva para que cualquier programador pueda cambiar el código en el sistema en cualquier momento, se implementa la práctica de propiedad colectiva. Los estándares de codificación ayudan a mantener un código ordenado y legible.

2.2.7.1. Ventajas.

Las empresas de desarrollo de software tienen la máxima ventaja que son sus costos, tiempo y presentación de sus aplicaciones. XP elimina las actividades improductivas que permite a los desarrolladores centrarse en la codificación.

La simplicidad del código. Con diferentes iteraciones con el tiempo, se realiza la refactorización del código, es decir, los desarrolladores están restringidos a escribir código en la forma más simple y después de cada iteración, el código se vuelve más simple.

XP **reduce los riesgos** relacionados con la programación y fracaso del proyecto. El cliente obtiene lo que él o ella quiere al final.

Comentarios constantes después de cada iteración de clientes ayuda a los programadores a proceder en la dirección correcta.

XP **ayuda a aumentar la satisfacción** de los empleados y retención. El desglose del proyecto en los subcomponentes y la retroalimentación constante ayudan a los empleados para completar el proyecto dentro del plazo sin horas extraordinarias.

Este enfoque **crea software de trabajo más rápido** como las pruebas del sistema continuamente garantizan la detección de todos los errores y el uso de pruebas de validación aprobadas por el cliente para determinar la finalización exitosa de un bloque de codificación asegura la implementación de solo lo que el cliente quiere y nada más.

2.2.7.2. Aplicaciones de XP.

Las prácticas de XP se utilizan ampliamente en el transcurso del desarrollo de la aplicación. Esto se hace para asegurar la organización del software; capacidad de respuesta mientras se reduce la sobrecarga del desarrollo garantizando reuniones frecuentes entre el cliente y el desarrollador, XP también ha sido útil en software global desarrollo. Cuando los clientes y los desarrolladores no están relacionados, la retroalimentación entre ellos no es lo suficientemente oportuna que puede convertirse en un obstáculo. Entonces, XP se practica cuando el resultado adoptado en un canal de comunicación es de acceso directo.(Yadav et al., 2019)

Yadav et al. (2019) explica que la XP implementa prácticas de ingeniería inversa mientras se desarrolla el modelo de software. Los desarrolladores comprenden el código fuente del sistema antiguo y entregan sus características para el cliente. El cliente valida el conjunto de especificaciones y los desarrolladores revisan nuevamente las características y comience a construir el sistema de acuerdo con las prácticas de XP. En el diseño de algunos modelos de software, hay desarrolladores independientes para la fase de ingeniería inversa y para implementación que es la fase de ingeniería avanzada, hay desarrolladores independientes presentes.

Yadav et al. (2019) afirma que, en el diseño de algunos modelos de software, hay desarrolladores independientes para la fase de ingeniería inversa y para la implementación, que es la fase de ingeniería directa, están presentes desarrolladores independientes. El grupo de ingeniería inversa comparte documentos y especificaciones con el grupo de ingeniería avanzada y se realizan consultas entre los dos grupos.

2.2.8. Entity Framework

Según TutorialPoint (2019) Entity Framework(EF) se manifestó en 2008, mostrando la interacción de aplicaciones .Net con base de datos relacionales. EF es un

Objeto Mapeador Relacional (ORM), que mapea los objetos de la aplicación de una base de datos relacional, sus tablas y atributos. EF es un marco de código libre para ADO.NET. y ayuda a los programadores en códigos redundantes y conservar la información utilizada en el sistema.

Mueller (2013) sostiene que EF proporciona lo necesario para tener tipos de modelos para interactuar con el desarrollador. Entity Framework es en realidad una tecnología Microsoft ActiveX Data Object .NET (ADO.NET), que crea diseños difíciles de realizar donde el desarrollador puede usar esos códigos de manera entendible. No es necesario traducir entre los niveles de abstracción: Entity Framework realiza esa tarea por usted. Entity Framework es una completa solución que funciona con cualquier fuente de datos, incluso archivos planos y bases de datos jerárquicas.

En Stack Overflow (2020) se afirma que EF reduce significativamente el código a escribir para la obtención de datos, Microsoft recomienda este ORM para la obtención de datos.

2.2.8.1. Características

TutorialPoint, (2019) menciona las características básicas de Entity Framework (EF).

- EF es accesible como código hacia los desarrolladores.
- La tecnología de EF ya no depende a nuevas versiones de .NET.
- El desarrollador puede trabajar con diferente base de datos, mejorando el código en relación de entidades y objetos.
- Compatibilidad con Linq to Entiites, para consultas de las entidades.

- EF de cifra consultas de entidad de forma parametrizada a datos específicos hacia el lector.
- Posee un contenedor que da seguimiento a cambios, administrando las identidades.
- Trabaja con un paradigma visual de diseño en relación a sus clases.
- EF tiene instrucciones de procedimientos almacenados.

Mientras que Dykstra & Anderson (2014) describen dos características de Entity Framework que son especialmente valiosas cuando se implementa en el entorno de la nube: resistencia de la conexión (automática reintentos para errores transitorios) e interceptación de comandos (captura las consultas SQL enviadas a la base de datos para registrarlos o modificarlos).

Habilitar la resistencia de la conexión: Los servidores de bases de datos están conectados directamente en el mismo centro de datos. Incluso si una web en la nube con el servidor de datos en la nube esté alojada en el mismo centro de datos, hay más redes conexiones entre ellos que pueden tener problemas, como balanceadores de carga. Muchos o la mayoría de los problemas de conexión cuando accede a un servicio en la nube son transitorios, es decir, se resuelven en un corto período de tiempo. Puede proporcionar una experiencia mucho mejor a sus usuarios si maneja los errores transitorios volviendo a intentar automáticamente, haciendo que la mayoría de ellos sean invisibles cliente. La función de resistencia de la conexión en Entity Framework automatiza ese proceso de reintentando consultas SQL fallidas.

Habilitar la interceptación de comandos: Para probar la función de resistencia de la conexión, necesita una forma de interceptar consultas que Entidad Framework envía a SQL Server y reemplaza la respuesta de SQL Server con un tipo de excepción

que suele ser transitorio. También puede utilizar la interceptación de consultas para implementar una práctica recomendada para las aplicaciones en la nube: registrar la latencia y el éxito o fracaso de todas las llamadas a servicios externos, como servicios de base de datos. EF6 proporciona una API de registro que puede facilitar el registro.

2.2.8.2 Elementos.

Mueller (2013) afirma que EF realiza tres tareas específicas que se encuentran en archivos XML y son:

Modelo conceptual: Este modelo capta como se ve la base de datos según la aplicación y es la que tiene más comunicación con los desarrolladores. Las clases nuevas se comunican con la base de datos, si realizamos cambios en la base de datos o en la aplicación los cambios se reflejan en los dos. Un modelo conceptual también incorpora el concepto de un espacio de nombres, tal como lo hacen sus aplicaciones. El espacio de nombres de EF realiza las mismas funciones que el espacio de nombres en su aplicación. En el corazón de este modelo se encuentran al determinar la entidad y las relaciones utilizadas de la base de datos para construir la vista.

Modelo de almacenamiento: este modelo establece como el administrador de datos visualiza la base de datos mediante Visual Studio; también se conoce como un modelo lógico. Según TutorialPoint (2019) este modelo abarca en formato XML toda la estructura de la base de datos.

Asignaciones entre modelos: Se sabe que hay dos modelos que se utilizan con EF: el conceptual modelo presenta la base de datos desde la aplicación y el modelo de almacenamiento presenta la base de datos del administrador. Estos dos modelos son necesariamente diferentes. Para que los dos modelos funcionen juntos, EF necesita un mapeo de modelos, donde se describe como se realizó la relación de los anteriores modelos. El propósito general de la parte de mapeo de modelos de EF es la traducción y

construcción de los elementos que salen del modelo conceptual hacia el modelo de almacenamiento. Cuando se realiza este mapeo permite localizar la base de datos y realizar un mínimo esfuerzo en las tareas del código. Así se muestra que la relación entre estos tres modelos es fundamental además de automático. Según TutorialPoint (2019) este elemento se almacena en un archivo XML donde la lógica de este mapeo se da a la interacción de los modelos conceptual y de almacenamiento con las entidades escritas y los datos de las tablas cada una según corresponde.

2.2.9. Sistemas de gestión y control

2.2.9.1. Reconocimiento de entidad.

Los enfoques existentes para el reconocimiento de entidad nombrada (NER) combinan la identificación de límite de entidad y la identificación de clase de entidad nombrada. Existe una gran cantidad de enfoques de aprendizaje supervisado para NER. La mayoría de estos enfoques se basan en un conjunto de datos anotado de un dominio similar para capacitar al sistema. Hasta donde sabemos, el marco propuesto es el primero de su tipo en la identificación de clase de entidad con nombre cuando se conocen los límites de la entidad (Ross et al., 2016).

2.2.9.2. Gestión de máquinas de construcción.

En el negocio de la construcción, especialmente para proyectos de infraestructura civil, una gran parte de los gastos generales del proyecto se asignan a diversos costos asociados con máquinas pesados. Como resultado, el seguimiento y monitoreo continuo de las tareas realizadas por equipos pesados de construcción es vital para los gerentes de proyectos y el personal del sitio de trabajo. Los enfoques actuales para el monitoreo automatizado de equipos de construcción incluyen la ubicación y métodos de seguimiento de acciones (Cheng et al., 2017).

2.2.9.3. Productividad.

Es un hecho desafortunado que la industria de la construcción sufre de menores tasas de productividad en comparación con la mayoría de las industrias manufactureras. El tiempo productivo en la industria de la construcción es solo del 43%, en comparación con el 88% en la fabricación. Un factor importante que contribuye a este problema es que los proyectos de construcción son únicos y generalmente es muy difícil encontrar proyectos, operaciones completamente similares. Como consecuencia, en contraste con las industrias manufactureras que incluyen procesos altamente repetitivos, una sola técnica de gestión de proyectos o un solo conjunto de medidas de rendimiento fijas rara vez están disponibles en la industria de la construcción (Cheng et al., 2017).

2.2.9.4. Rendimiento.

Por lo tanto, el primer paso para mejorar la productividad dentro de la industria de la construcción es desarrollar técnicas eficientes para supervisar el rendimiento y la producción de los recursos clave que sea lo suficientemente flexible como para manejar las condiciones ampliamente variables que surgen en diferentes sitios de trabajo. Dado que una gran parte de los costos totales del proyecto generalmente se derivan de los costos de alquilar, poseer, arrendar, mantener equipos pesados de construcción, nos enfocaremos en reconocer y rastrear actividades de construcción de equipo pesado. Tenga en cuenta que más allá del rendimiento y el estudio de la productividad, el monitoreo de los equipos de construcción es útil para el control y seguimiento de emisiones, la gestión de seguridad y el análisis, reducción de los tiempos de inactividad (Cheng et al., 2017).

2.2.9.5. Ubicaciones y actividades.

Las prácticas comunes para reconocer y monitorear actividades en las obras de construcción se realizan mediante la recopilación manual de datos y observaciones

directas. Se sabe que este proceso requiere mucho tiempo y trabajo. En los últimos años se han introducido una serie de métodos para reconocer y rastrear automáticamente las ubicaciones y acciones de equipos pesados de construcción. Estos métodos incluyen el uso de sensores activos (GPS (Sistema de posicionamiento global), acelerómetros, etiquetas RFID (identificación por radiofrecuencia), etc.) o sensores pasivos (procesamiento de videos usando algoritmos de visión por computadora) (Cheng et al., 2017).

CAPÍTULO III. Materiales y Métodos

3.1. Descripción del Lugar de Ejecución

El proyecto se desarrolló en el área administrativa de la empresa Contratistas Generales E&R, ubicada en la ciudad de Juliaca – San Román – Puno.

3.2. Materiales e Insumos

Tabla 3.

Materiales del proyecto.

	Descripción	Versión	Tipo de recurso	Licencia
	1 Laptop HP AMD Ryzen	5	Equipo	
	Visual Studio Community 2019	16.6	Entorno de desarrollo	Open-source
Materiales	Asp .Net Core	3.0	Framework	Open-source
para el	Entity Framework	3.1.2	Framework ORM	Open-source
desarrollo	Angular Cli	8.3.14	Framework	Open-source
del sistema	Bootstrap	4	Librería	Open-source
web	Postman	7.35.0	Herramienta de pruebas API	Open-source
	Sql Server 2017	14.0	Base de Datos	Open-source
	Internet		Proveedor de servicios	
	Balsamiq Wireframes	4.1.9	Herramienta de diseño	Open-source

Fuente: Elaboración propia.

3.3. Arquitectura de Solución

Para la arquitectura de solución del proyecto mostramos en la imagen el componente central WS Gestión Máquina indicando que trabajamos la parte del backend con .Net Core, Entity Framework.

Seguidamente al lado izquierdo muestra a que base de Datos se envía la consultas para el sistema; se utilizó Sql Server.

Finalmente mostramos la parte derecha de la imagen donde el usuario visualizará y tendrá las funcionalidades del sistema; donde utilizamos el framework Angular con Bootstrap y realizará la conexión mediante Http, enviará mensajes Json y autenticación JWT(Json Web Token).

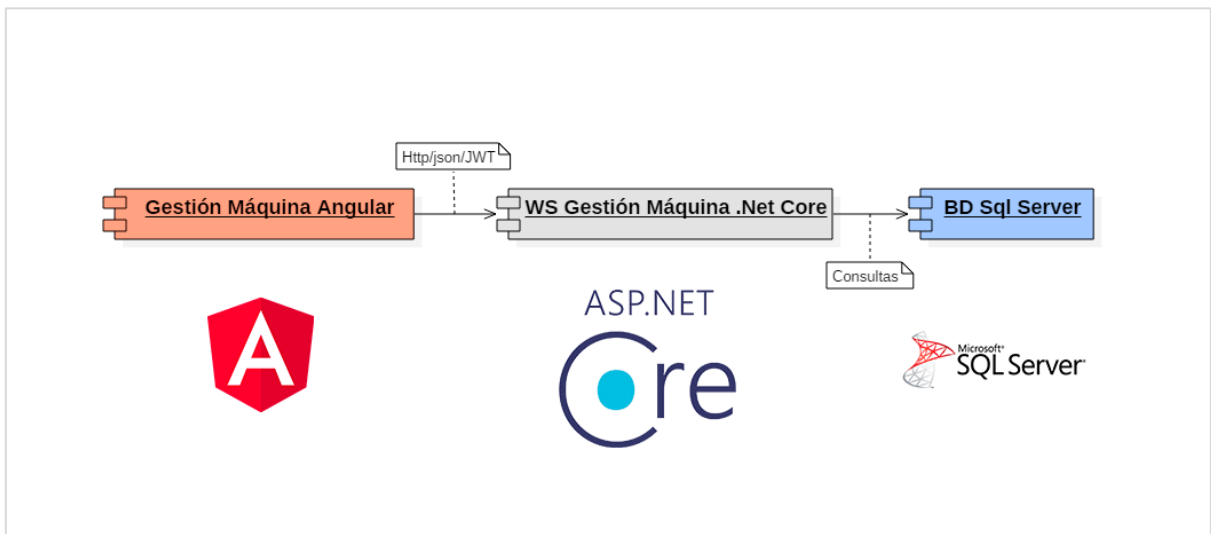


Figura 1: Arquitectura de solución.

Fuente: Elaboración propia.

3.4. Metodología

3.4.1 Tipo de investigación

El presente proyecto de investigación es de tipo propositiva. Correa & Wilson (2014) manifiesta que una investigación es propositiva por cuanto hay una necesidad o vacío dentro de una institución, una vez recopilado los datos se realizará una propuesta de sistema de evaluación de desempeño para superar el problema identificado.

Al recopilar datos del área administrativa se ve la necesidad o el problema en la gestión de horas máquina en la empresa Contratistas Generales, por lo cual se considera que la investigación es propositiva, siendo su objetivo fundamental desarrollar un sistema web horas máquina para dar una solución al problema.

3.4.2. Desarrollo de la investigación según la metodología XP

3.4.2.1. Uso de metodología XP Personalizado

El desarrollo de la investigación se hizo en las fases primordiales de la programación XP de planificar, diseño, codificación y pruebas.

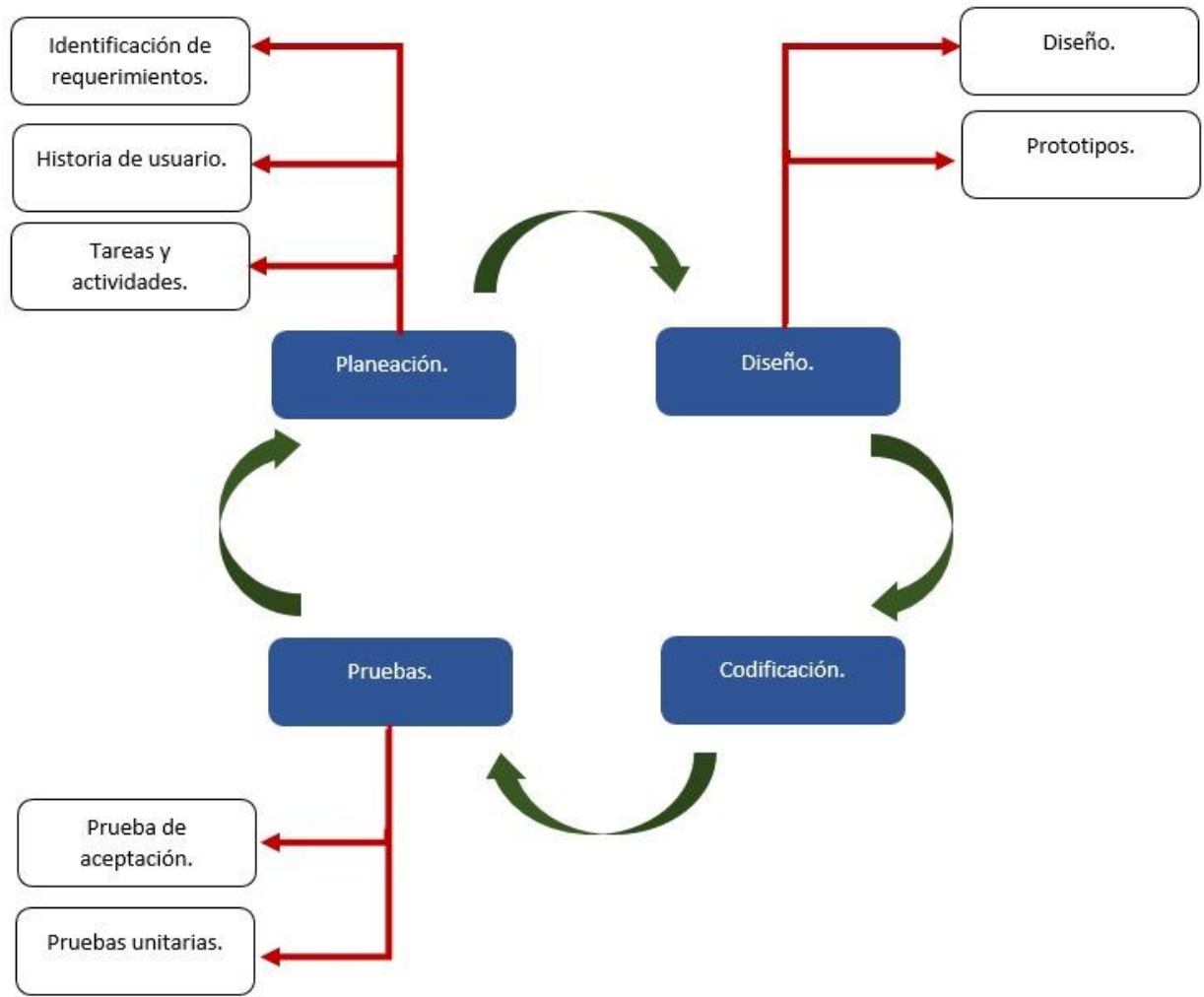


Figura 2: Metodología XP personalizado
 Fuente: Elaboración propia.

3.4.2.2. Planificación.

Al definir el objetivo de la investigación realizamos las entrevistas necesarias al usuario del área administrativa de la empresa, para aclarar dudas y elaborar la primera fase de la metodología, identificar los requisitos de parte de los interesados e historias de usuario.

3.4.2.1.1. Identificación de requerimientos:

Operario

Registrar operarios de Contratistas Generales E&R.

Modificar operarios en caso de registro erróneo.

Lista de Operarios con su paginación.

Maquinas

Registrar máquinas

Modificar máquinas

Lista de máquinas

Parte Diario

Son las horas de trabajo de las máquinas con su respectivo operario y a la obra que pertenece.

Registro de un parte diario con sus respectivas obra, máquina y operario.

Modificar un parte diario.

Listar un Parte Diario.

Obra

Registrar obra.

Modificar obra.

Lista de obra.

Consumo

Registrar consumo

Modificar consumo

Lista de consumo

3.4.2.1.2. *Historia de usuarios*

Historia de Usuario 1: Crear operario.

Tabla 4.

Crear Operario.

Historia de Usuario	
Desarrollo del sistema web utilizando ASP.NET CORE para el área de administración de Contratistas Generales.	
Número: 1	Usuario: Administrador
Nombre: Crear operario	
Prioridad: Alta	Riesgo de desarrollo: Baja
Puntos Estimados: 1	Iteraciones asignadas: 1
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: El usuario podrá crear nuevos operarios en el sistema para lo cual se realizará las siguientes tareas: Se realizará el diseño del formulario para un nuevo operario. Crear un modelo de datos y el mapeo con Entity Framework, pruebas de inserción de operario en el web api. Crearemos la parte lógica de inserción de datos, guardando la información en la base de datos.	
Observaciones: El administrador tendrá el acceso a este módulo del sistema.	

Fuente: Elaboración propia.

Tabla 5.

Elaborar Diseño y Formulario Nuevo Operario.

Tarea de Ingeniería	
N° de la tarea: 1	N° de la Historia: 1
Nombre: Elaborar diseño y formulario nuevo operario	
Tipo: Desarrollo	Puntos Estimados: 0.5
Inicio: 15/06/2020	Fin: 15/06/2020
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: Se realizará el diseño e interface nuevo operario	

Fuente: Elaboración propia.

Tabla 6.

Modelo de Base de Datos Operario y Mapeo.

Tarea de Ingeniería	
N° de la tarea: 2	N° de la Historia: 1
Nombre: Modelo de base de datos operario y mapeo	
Tipo: Desarrollo	Puntos Estimados: 1
Inicio: 15/06/2020	Fin: 16/06/2020
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: Crear modelo operario en la base de datos, mapear y realizar prueba de inserción.	

Fuente: Elaboración propia.

Tabla 7.

Crear Datos Operario en la Base de Datos.

Tarea de Ingeniería	
N° de la tarea: 2	N° de la Historia: 1
Nombre: Crear datos operario en la base de datos	
Tipo: Desarrollo	Puntos Estimados: 1
Inicio: 16/06/2020	Fin: 17/06/2020
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: Realizar la parte lógica de la inserción, validaciones e inserción de datos del operario.	

Fuente: Elaboración propia.

Historia de usuario 2: Editar operario.

Tabla 8.

Editar Operario.

Historia de Usuario	
Desarrollo del sistema web utilizando ASP.NET CORE para el área de administración de Contratistas Generales.	
Número: 2	Usuario: Administrador
Nombre: Editar operario	
Prioridad: Alta	Riesgo de desarrollo: Baja
Puntos Estimados: 1	Iteraciones asignadas: 1
Programador: Marco Antonio Choquehuanca Quispe	

Descripción: El usuario podrá editar un operario ya registrado, para lo cual se realizará las siguientes tareas:

Actualizar formulario editar operario.

Crear prueba editar operario en web api.

Lógica de edición operario y actualizar en la base de datos.

Observaciones: El administrador tendrá el acceso a este módulo del sistema.

Fuente: Elaboración propia.

Tabla 9.

Interface Editar Operario.

Tarea de Ingeniería

N° de la tarea: 1

N° de la Historia: 2

Nombre: Interface editar operario

Tipo: Desarrollo

Puntos Estimados:0.5

Inicio: 17/06/2020

Fin: 17/06/2020

Programador: Marco Antonio Choquehuanca Quispe

Descripción: Elaborar formulario reactivo editar operario

Fuente: Elaboración propia.

Tabla 10.

Guardar Operario Editado.

Tarea de Ingeniería

N° de la tarea: 2

N° de la Historia: 2

Nombre: Guardar operario editado

Tipo: Desarrollo

Puntos Estimados:1

Inicio: 18/06/2020

Fin: 18/06/2020

Programador: Marco Antonio Choquehuanca Quispe

Descripción: Crear prueba editar operario web api, editar operario parte lógica y guardar en la base de datos.

Fuente: Elaboración propia.

Historia de Usuario 3: Listar Operario.

Tabla 11.

Listar Operario.

Historia de Usuario

Desarrollo del sistema web utilizando ASP.NET CORE para el área de administración de Contratistas Generales.

Número: 3

Usuario: Administrador

Nombre: Listar Operario

Prioridad: Alta

Riesgo de desarrollo: Baja

Puntos Estimados: 1

Iteraciones asignadas:1

Programador: Marco Antonio Choquehuanca Quispe

Descripción: El usuario podrá visualizar la lista de operarios ya registrado, para lo cual se realizará las siguientes tareas:

Diseño e interfaz de lista operarios.

Crear prueba lista operarios web api.

Lógica de lista operarios con paginación.

Observaciones: El administrador tendrá el acceso a este módulo del sistema.

Fuente: Elaboración propia.

Tabla 12.

Diseño e Interfaz Lista Operario.

Tarea de Ingeniería	
N° de la tarea: 1	N° de la Historia: 3
Nombre: Diseño e interfaz lista operario	
Tipo: Desarrollo	Puntos Estimados: 0.5
Inicio: 19/06/2020	Fin: 19/06/2020
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: Elaborar diseño e interfaz lista operario	

Fuente: Elaboración propia.

Tabla 13.

Mostrar Lista de Operario.

Tarea de Ingeniería	
N° de la tarea: 2	N° de la Historia: 3
Nombre: Mostrar lista de operario	
Tipo: Desarrollo	Puntos Estimados: 1
Inicio: 19/06/2020	Fin: 20/06/2020
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: Realizar prueba lista operario web api, parte lógica y lista de operarios registrados en la interfaz.	

Fuente: Elaboración propia.

Historia de usuario 4: Crear Máquina

Tabla 14.

Crear Máquina.

Historia de Usuario	
Desarrollo del sistema web utilizando ASP.NET CORE para el área de administración de Contratistas Generales.	
Número: 4	Usuario: Administrador
Nombre: Crear máquina	
Prioridad: Alta	Riesgo de desarrollo: Baja
Puntos Estimados: 1	Iteraciones asignadas: 1
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: El usuario podrá crear una máquina en el sistema, para lo cual se realizará las siguientes tareas:	
Elaborar diseño e interfaz para crear nueva máquina.	
Crear prueba insertar máquina y mapeo del modelo.	
Lógica de insertar máquina e insertar en base de datos.	
Observaciones: El administrador tendrá el acceso a este módulo del sistema.	

Fuente: Elaboración propia.

Tabla 15.

Diseño e Interfaz Crear Máquina.

Tarea de Ingeniería	
N° de la tarea: 1	N° de la Historia: 4
Nombre: Diseño e interfaz crear máquina	
Tipo: Desarrollo	Puntos Estimados: 0.5
Inicio: 20/06/2020	Fin: 20/06/2020
Programador: Marco Antonio Choquehuanca Quispe	

Descripción: Elaborar formulario reactivo nueva máquina

Fuente: Elaboración propia.

Tabla 16.

Modelo de Base de Datos Máquina y Mapeo.

Tarea de Ingeniería

N° de la tarea: 2

N° de la Historia: 4

Nombre: Modelo de base de datos máquina y mapeo

Tipo: Desarrollo

Puntos Estimados:1

Inicio: 21/06/2020

Fin: 21/06/2020

Programador: Marco Antonio Choquehuanca Quispe

Descripción: Realizar modelo de datos máquina y mapeo del modelo

Fuente: Elaboración propia.

Tabla 17.

Insertar Máquina.

Tarea de Ingeniería

N° de la tarea: 3

N° de la Historia: 4

Nombre: Insertar máquina

Tipo: Desarrollo

Puntos Estimados:1

Inicio: 22/06/2020

Fin: 22/06/2020

Programador: Marco Antonio Choquehuanca Quispe

Descripción: Realizar lógica de insertar máquina y guardarlo en la base de datos.

Fuente: Elaboración propia.

Historia de Usuario 5: Editar Máquina

Tabla 18.

Editar Máquina.

Historia de Usuario	
Desarrollo del sistema web utilizando ASP.NET CORE para el área de administración de Contratistas Generales.	
Número: 5	Usuario: Administrador
Nombre: Editar máquina	
Prioridad: Alta	Riesgo de desarrollo: Baja
Puntos Estimados: 1	Iteraciones asignadas: 1
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: El usuario podrá editar una máquina ya registrada, para lo cual se realizará las siguientes tareas:	
Actualizar interfaz editar máquina de forma reactiva.	
Crear prueba editar máquina en web api.	
Realizar lógica de edición máquina y guardar en la base de datos.	
Observaciones: El administrador tendrá el acceso a este módulo del sistema.	

Fuente: Elaboración propia.

Tabla 19.

Interface Editar Máquina.

Tarea de Ingeniería 1	
N° de la tarea: 1	N° de la Historia: 5
Nombre: Interface editar máquina	
Tipo: Desarrollo	Puntos Estimados: 0.5
Inicio: 23/06/2020	Fin: 23/06/2020
Programador: Marco Antonio Choquehuanca Quispe	

Descripción: Actualizar interfaz reactivo editar máquina

Fuente: Elaboración propia.

Tabla 20.

Guardar Máquina Editado.

Tarea de Ingeniería 2

N° de la tarea: 2

N° de la Historia: 5

Nombre: Guardar máquina editado

Tipo: Desarrollo

Puntos Estimados:1

Inicio: 23/06/2020

Fin: 24/06/2020

Programador: Marco Antonio Choquehuanca Quispe

Descripción: Realizar prueba web api de editar máquina y guardar máquina en la base de datos.

Fuente: Elaboración propia.

Historia de Usuario 6: Listar Máquina

Tabla 21.

Listar Máquina.

Historia de Usuario

Desarrollo del sistema web utilizando ASP.NET CORE para el área de administración de Contratistas Generales.

Número: 6

Usuario: Administrador

Nombre: Listar máquina

Prioridad: Alta

Riesgo de desarrollo: Baja

Puntos Estimados: 1

Iteraciones asignadas:1

Programador: Marco Antonio Choquehuanca Quispe

Descripción: El usuario podrá visualizar máquinas registradas, para lo cual se realizará las siguientes tareas:

Diseño e interfaz de lista de máquinas.

Realizar prueba de lista de máquinas web api y mostrar lista en la interfaz.

Observaciones: El administrador tendrá el acceso a este módulo del sistema.

Elaboración: Fuente propia.

Tabla 22.

Diseño e Interfaz de Lista Máquina.

Tarea de Ingeniería	
N° de la tarea: 1	N° de la Historia: 6
Nombre: diseño e interfaz de lista máquina	
Tipo: Desarrollo	Puntos Estimados: 0.5
Inicio: 24/06/2020	Fin: 24/06/2020
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: Elaborar formulario reactivo lista de máquinas	

Fuente: Elaboración propia.

Tabla 23.

Listar Máquina.

Tarea de Ingeniería	
N° de la tarea: 2	N° de la Historia: 6
Nombre: Listar máquinas	
Tipo: Desarrollo	Puntos Estimados: 1
Inicio: 25/06/2020	Fin: 25/06/2020

Programador: Marco Antonio Choquehuanca Quispe

Descripción: Prueba web api de listado de máquinas y mostrarlo en la interfaz con su paginación.

Fuente: Elaboración propia.

Historia de Usuario 7: Crear Parte Diario

Tabla 24.

Crear Parte Diario.

Historia de Usuario

Desarrollo del sistema web utilizando ASP.NET CORE para el área de administración de Contratistas Generales.

Número: 7

Usuario: Administrador

Nombre: Crear parte diario

Prioridad: Alta

Riesgo de desarrollo: Baja

Puntos Estimados: 1

Iteraciones asignadas: 2

Programador: Marco Antonio Choquehuanca Quispe

Descripción: El usuario podrá crear un parte diario en el sistema, para lo cual se realizará las siguientes tareas:

Crear diseño e interfaz de nuevo parte diario.

Realizar modelo de datos parte diario y mapeo del modelo.

Realizar lógica de insertar parte diario y guardarlo en la base de datos.

Observaciones: El administrador tendrá el acceso a este módulo del sistema.

Fuente: Elaboración propia.

Tabla 25.

Diseño e Interfaz Crear Parte Diario.

Tarea de Ingeniería	
N° de la tarea: 1	N° de la Historia: 7
Nombre: Diseño e interfaz crear parte diario	
Tipo: Desarrollo	Puntos Estimados: 1
Inicio: 26/06/2020	Fin: 26/06/2020
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: Diseño y elaboración de interfaz de parte diario en un formulario reactivo, paginación y buscador.	

Fuente: Elaboración propia.

Tabla 26.

Modelo Base de Datos y Mapeo.

Tarea de Ingeniería	
N° de la tarea: 2	N° de la Historia: 7
Nombre: Modelo base de datos y mapeo	
Tipo: Desarrollo	Puntos Estimados: 1
Inicio: 29/06/2020	Fin: 29/06/2020
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: Elaborar prueba web api de insertar parte diario, y mapeo del modelo parte diario.	

Fuente: Elaboración propia.

Tabla 27.

Guardar Nuevo Parte Diario.

Tarea de Ingeniería	
N° de la tarea: 3	N° de la Historia: 7
Nombre: Guardar nuevo parte diario	
Tipo: Desarrollo	Puntos Estimados: 1
Inicio: 30/06/2020	Fin: 30/06/2020
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: Realizar lógica de insertar parte diario y guardar en la base de datos	

Fuente: Elaboración propia.

Historia de Usuario 8: Editar Parte Diario

Tabla 28.

Editar Parte Diario.

Historia de Usuario	
Desarrollo del sistema web utilizando ASP.NET CORE para el área de administración de Contratistas Generales.	
Número: 8	Usuario: Administrador
Nombre: Editar parte diario	
Prioridad: Alta	Riesgo de desarrollo: Baja
Puntos Estimados: 1	Iteraciones asignadas: 1
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: El usuario podrá editar un parte diario ya registrado en el sistema, para lo cual se realizará las siguientes tareas: Actualizar interfaz editar parte diario en el formulario reactivo.	

Crear prueba web api editar parte diario.

Realizar lógica de editar parte diario y guardarlo en la base de datos.

Observaciones: El administrador tendrá el acceso a este módulo del sistema.

Fuente: Elaboración propia.

Tabla 29.

Interfaz Editar Parte Diario.

Tarea de Ingeniería

N° de la tarea: 1

N° de la Historia: 8

Nombre: Interfaz editar parte diario

Tipo: Desarrollo

Puntos Estimados: 2

Inicio: 1/07/2020

Fin: 1/07/2020

Programador: Marco Antonio Choquehuanca Quispe

Descripción: Actualizar interfaz editar parte diario en dicho formulario reactivo.

Fuente: Elaboración propia.

Tabla 30.

Interfaz Editar Parte Diario.

Tarea de Ingeniería

N° de la tarea: 2

N° de la Historia: 8

Nombre: Guardar parte diario editado

Tipo: Desarrollo

Puntos Estimados: 1

Inicio: 2/07/2020

Fin: 2/07/2020

Programador: Marco Antonio Choquehuanca Quispe

Descripción: Realizar prueba web api de editar parte diario y guardarlo en la base de datos.

Fuente: Elaboración propia.

Historia de Usuario 9: Listar Parte Diario

Tabla 31.

Listar Parte Diario.

Historia de Usuario

Desarrollo del sistema web utilizando ASP.NET CORE para el área de administración de Contratistas Generales.

Número: 9

Usuario: Administrador

Nombre: Listar parte diario

Prioridad: Alta

Riesgo de desarrollo: Baja

Puntos Estimados: 1

Iteraciones asignadas: 2

Programador: Marco Antonio Choquehuanca Quispe

Descripción: El usuario podrá visualizar partes diarios ya registrados en el sistema, para lo cual se realizará las siguientes tareas:

Elaborar diseño e interfaz de un nuevo formulario reactivo parte diario.

Realizar prueba web api lista de partes diarios.

Realizar lógica de lista de partes diarios y mostrarlo en la interfaz.

Observaciones: El administrador tendrá el acceso a este módulo del sistema.

Fuente: Elaboración propia.

Tabla 32.

Diseño e Interfaz Lista Partes Diarios.

Tarea de Ingeniería	
N° de la tarea: 1	N° de la Historia: 9
Nombre: Diseño e interfaz lista partes diarios	
Tipo: Desarrollo	Puntos Estimados: 1
Inicio: 3/07/2020	Fin: 3/07/2020
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: Elaborar diseño e interfaz de partes diarios, buscador y paginación	

Fuente: Elaboración propia.

Tabla 33.

Lista de Parte Diario.

Tarea de Ingeniería	
N° de la tarea: 2	N° de la Historia: 9
Nombre: Lista de parte diario	
Tipo: Desarrollo	Puntos Estimados: 1
Inicio: 4/07/2020	Fin: 4/07/2020
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: Realizar prueba web api lista de parte diario y mostrar lista en interfaz.	

Historia de usuario 10: Gestión de Obras

Tabla 34.

Gestión de Obras.

Historia de Usuario

Desarrollo del sistema web utilizando ASP.NET CORE para el área de administración de Contratistas Generales.

Número: 9

Usuario: Administrador

Nombre: Gestión de Obras

Prioridad: Media

Riesgo de desarrollo: Baja

Puntos Estimados: 1

Iteraciones asignadas: 3

Programador: Marco Antonio Choquehuanca Quispe

Descripción: El usuario podrá visualizar, crear y editar obras:

Realizar el listado de obras, entidad relación, interface y lógica de la operación.

Realizar nueva obra, interface, lógica de la operación.

Realizar editar obra, interface, lógica de la operación.

Observaciones: El administrador tendrá el acceso a este módulo del sistema.

Tabla 35.

Listar Obras.

Tarea de Ingeniería

N° de la tarea: 1

N° de la Historia: 10

Nombre: Listar Obras

Tipo: Desarrollo

Puntos Estimados: 1

Inicio: 20/10/2020

Fin: 20/10/2020

Programador: Marco Antonio Choquehuanca Quispe

Descripción: Realizar entidad relación, interface y operación lógica de listar obra.

Tabla 36.

Nueva Obra.

Tarea de Ingeniería	
N° de la tarea: 2	N° de la Historia: 10
Nombre: Nueva Obra	
Tipo: Desarrollo	Puntos Estimados: 1
Inicio: 20/10/2020	Fin: 20/10/2020
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: Realizar entidad relación, interface y operación lógica nueva obra.	

Tabla 37.

Editar Obra

Tarea de Ingeniería	
N° de la tarea: 3	N° de la Historia: 10
Nombre: Editar Obra	
Tipo: Desarrollo	Puntos Estimados: 1
Inicio: 20/10/2020	Fin: 20/10/2020
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: Realizar entidad relación, interface y operación lógica editar obra.	

Historia de Usuario 11: Gestión de Consumo

Tabla 38.

Gestión de Consumo.

Historia de Usuario

Desarrollo del sistema web utilizando ASP.NET CORE para el área de administración de Contratistas Generales.

Número: 9

Usuario: Administrador

Nombre: Gestión de Consumo

Prioridad: Media

Riesgo de desarrollo: Baja

Puntos Estimados: 1

Iteraciones asignadas: 3

Programador: Marco Antonio Choquehuanca Quispe

Descripción: El usuario podrá visualizar, crear y editar consumos:

Realizar el listado de consumo, entidad relación, interface y lógica de la operación.

Realizar nuevo consumo, interface, lógica de la operación.

Realizar editar consumo, interface, lógica de la operación.

Observaciones: El administrador tendrá el acceso a este módulo del sistema.

Tabla 39.

Listar Consumo.

Tarea de Ingeniería

N° de la tarea: 1

N° de la Historia: 10

Nombre: Listar Consumo

Tipo: Desarrollo

Puntos Estimados:1

Inicio: 20/10/2020

Fin: 20/10/2020

Programador: Marco Antonio Choquehuanca Quispe

Descripción: Realizar entidad relación, interface y operación lógica listar consumo.

Tabla 40.

Nuevo Consumo.

Tarea de Ingeniería	
N° de la tarea: 2	N° de la Historia: 10
Nombre: Nuevo Consumo	
Tipo: Desarrollo	Puntos Estimados: 1
Inicio: 20/10/2020	Fin: 20/10/2020
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: Realizar entidad relación, interface y operación lógica nuevo consumo.	

Tabla 41.

Editar Consumo.

Tarea de Ingeniería	
N° de la tarea: 3	N° de la Historia: 10
Nombre: Editar Consumo	
Tipo: Desarrollo	Puntos Estimados: 1
Inicio: 20/10/2020	Fin: 20/10/2020
Programador: Marco Antonio Choquehuanca Quispe	
Descripción: Realizar entidad relación, interface y operación lógica editar consumo.	

3.4.2.3. Diseño

En la fase de diseño elaboramos el diseño rápido para el sistema mediante un bosquejo a mano y prototipos, donde se mostrará los módulos requeridos del sistema los cuales deben ser sencillos para los usuarios y ejecutarlos en breve tiempo para los desarrolladores.

Los prototipos lo realizaremos en Balsamiq Wireframes, crearemos soluciones breves a según a los requerimientos e historias de usuarios identificados.

3.4.3.2.1. Prototipos

Tenemos tres módulos principales, operario, máquina y parte diario.

The image shows a wireframe of a web browser window titled "Gestión Horas Maquina". The address bar contains "https://www.operario.com". The main content area displays a form titled "Nuevo Operario" with the following fields and controls:

- Nombre: Text input field
- Apellido Paterno: Text input field
- Apellido Materno: Text input field
- Dni: Text input field
- Fecha de nacimiento: Date input field with slashes and a calendar icon
- Genero: Dropdown menu with "Masculino" selected
- Teléfono: Text input field

At the bottom of the form are two buttons: "Guardar" (highlighted in blue) and "Cancelar".

Figura 3: Nuevo Operario.

Fuente: Elaboración propia.

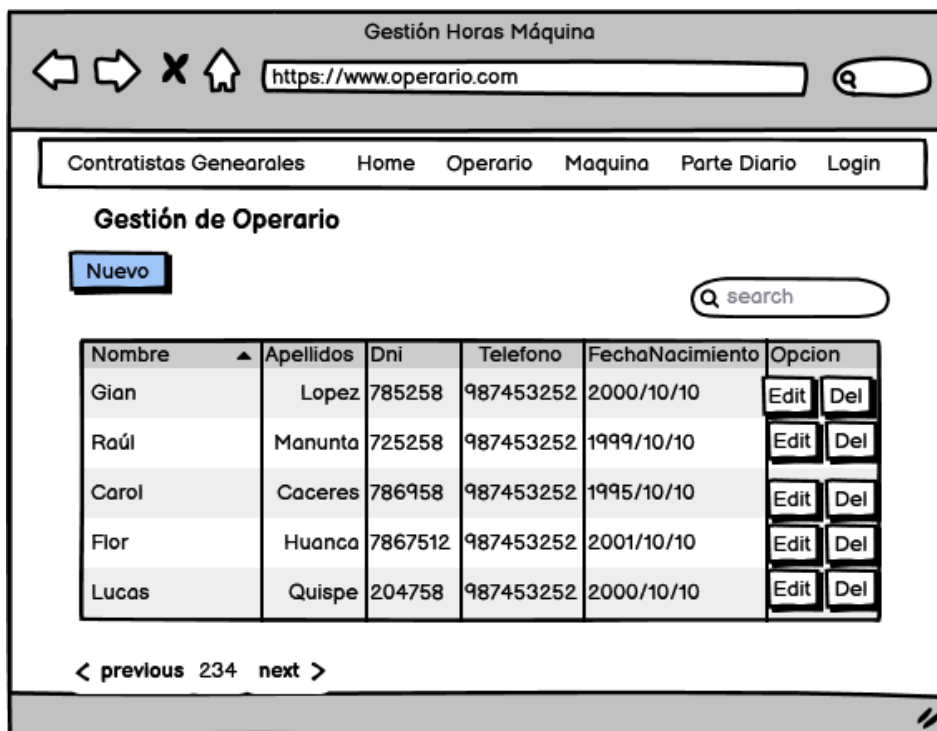


Figura 4: Gestión de Operario.

Fuente: Elaboración propia.

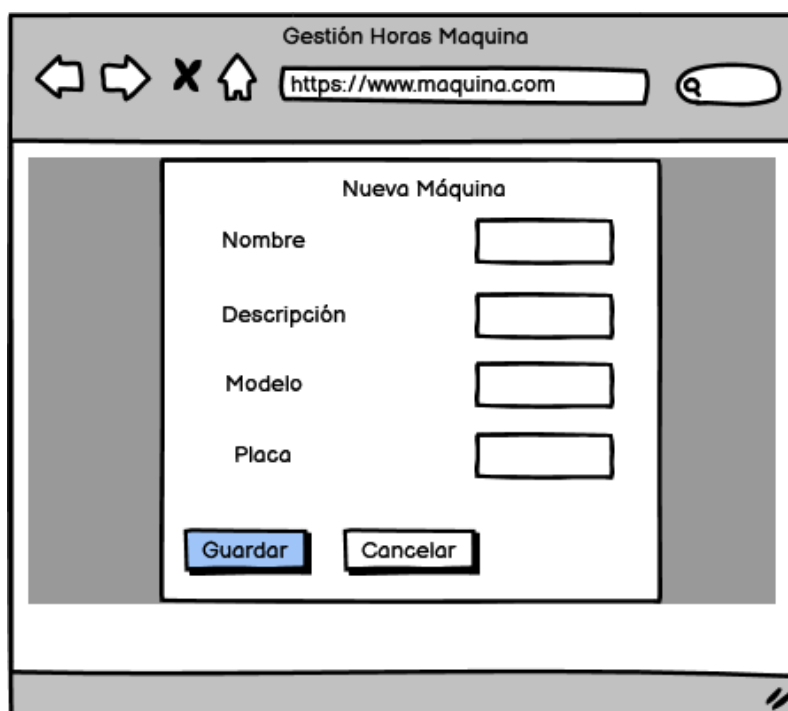


Figura 5: Nueva Máquina.

Fuente: Elaboración propia.

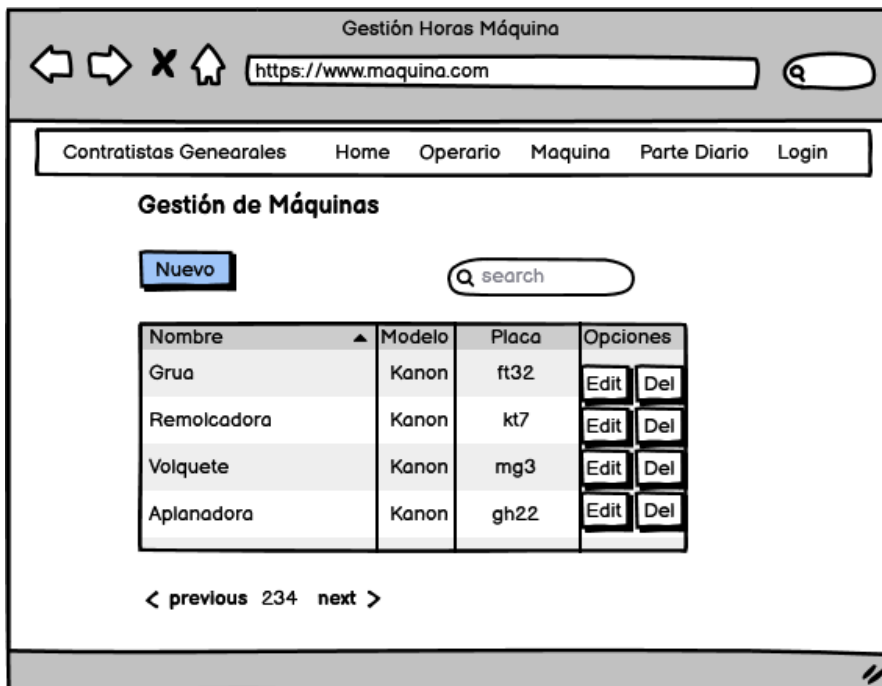


Figura 6: Gestión de Máquinas.

Fuente: Elaboración propia.

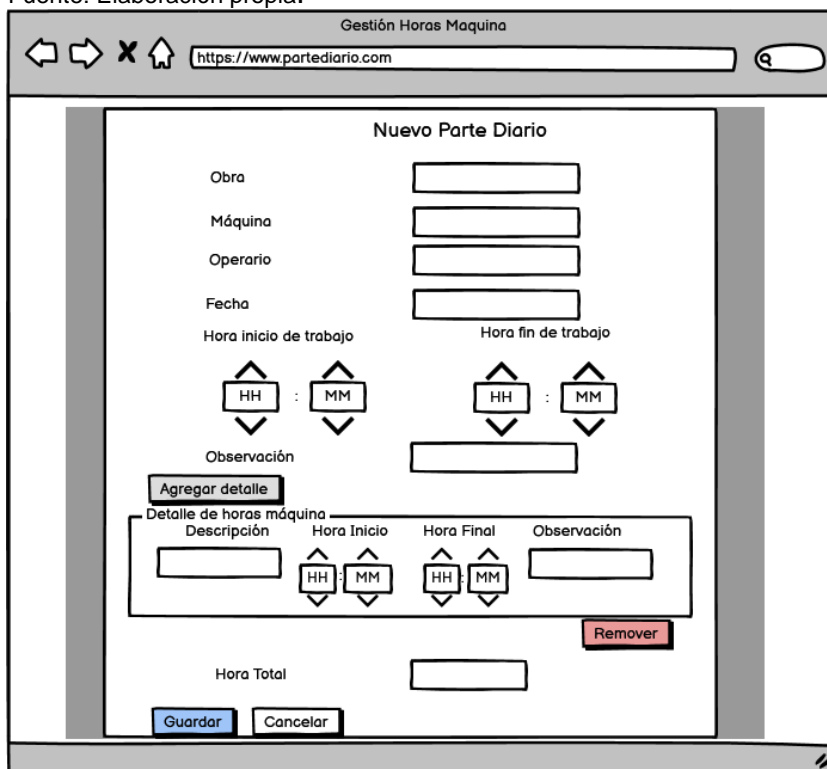


Figura 7: Nuevo Parte Diario.

Fuente: Elaboración propia.

Gestión Horas Máquina

[←](#) [→](#) [X](#) [🏠](#)

[Contratistas Genearales](#) [Home](#) [Operario](#) [Maquina](#) [Parte Diario](#) [Login](#)

Gestión de Parte Diario

[Imprimir](#)

[Nuevo](#)

Maquina	Operario ▲	Fecha	Hora inicio operario	Hora fin operario	Hora total máquina	Opciones
Grua	Carlos Gutierrez	2020/08/	10:00:00	16:00:00	03:00:00	Edit Del
Remolcadora	Lucas Alarcon	2020/08/	10:00:00	16:00:00	03:00:00	Edit Del
Volquete	Mari Chacon	2020/08/	10:00:00	16:00:00	03:00:00	Edit Del
Cisterna	Felix Condori	2020/08/	10:00:00	16:00:00	03:00:00	Edit Del

< previous 234 next >

Figura 8: Gestión de Parte Diario.

Fuente: Elaboración propia.

3.4.3.2.2. Diagrama de Clases

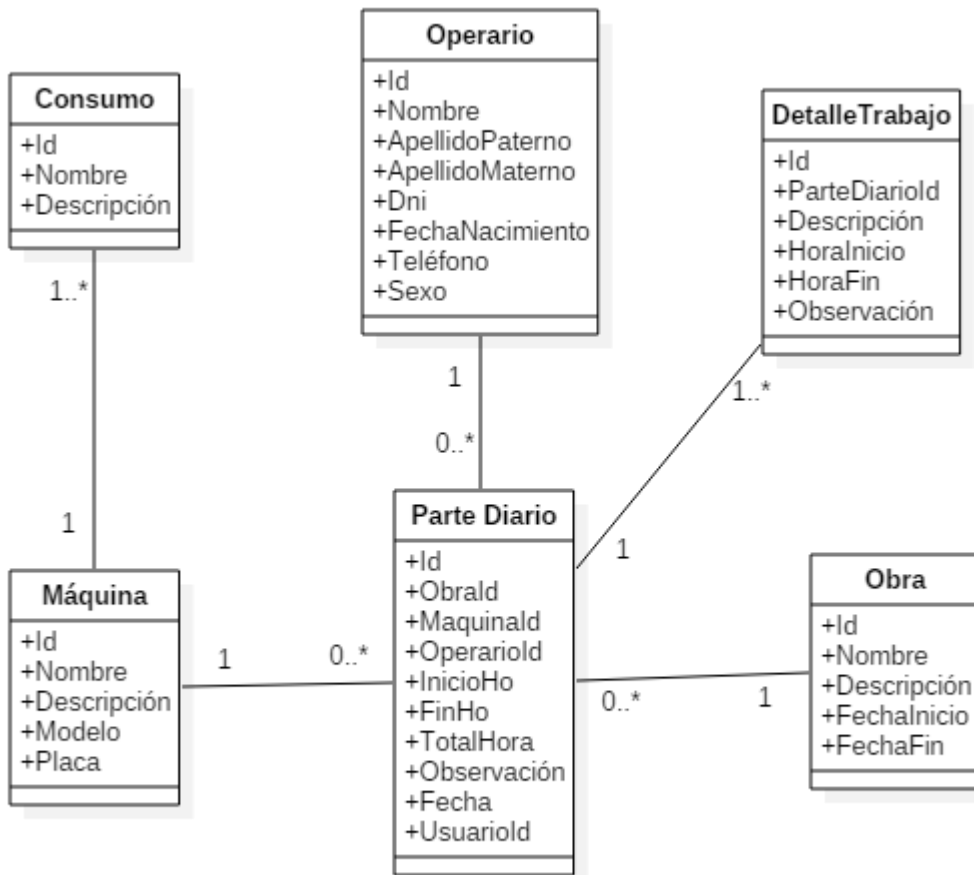


Figura 9: Diagrama de clases gestión de horas máquina.

Fuente Elaboración propia.

3.4.2.4. Codificación

En esta fase se convertirá todo lo diseñado en código real y funcional, donde se construirá una arquitectura de software, modelo de la base de datos, codificar la parte lógica del sistema y elaborar las interfaces propuestas.

3.4.2.3.1. Base de datos

El sistema de base de datos a utilizar es SQL Server 2017 creando tablas y definir relaciones entre ellas, así mostramos la entidad relación en la figura:

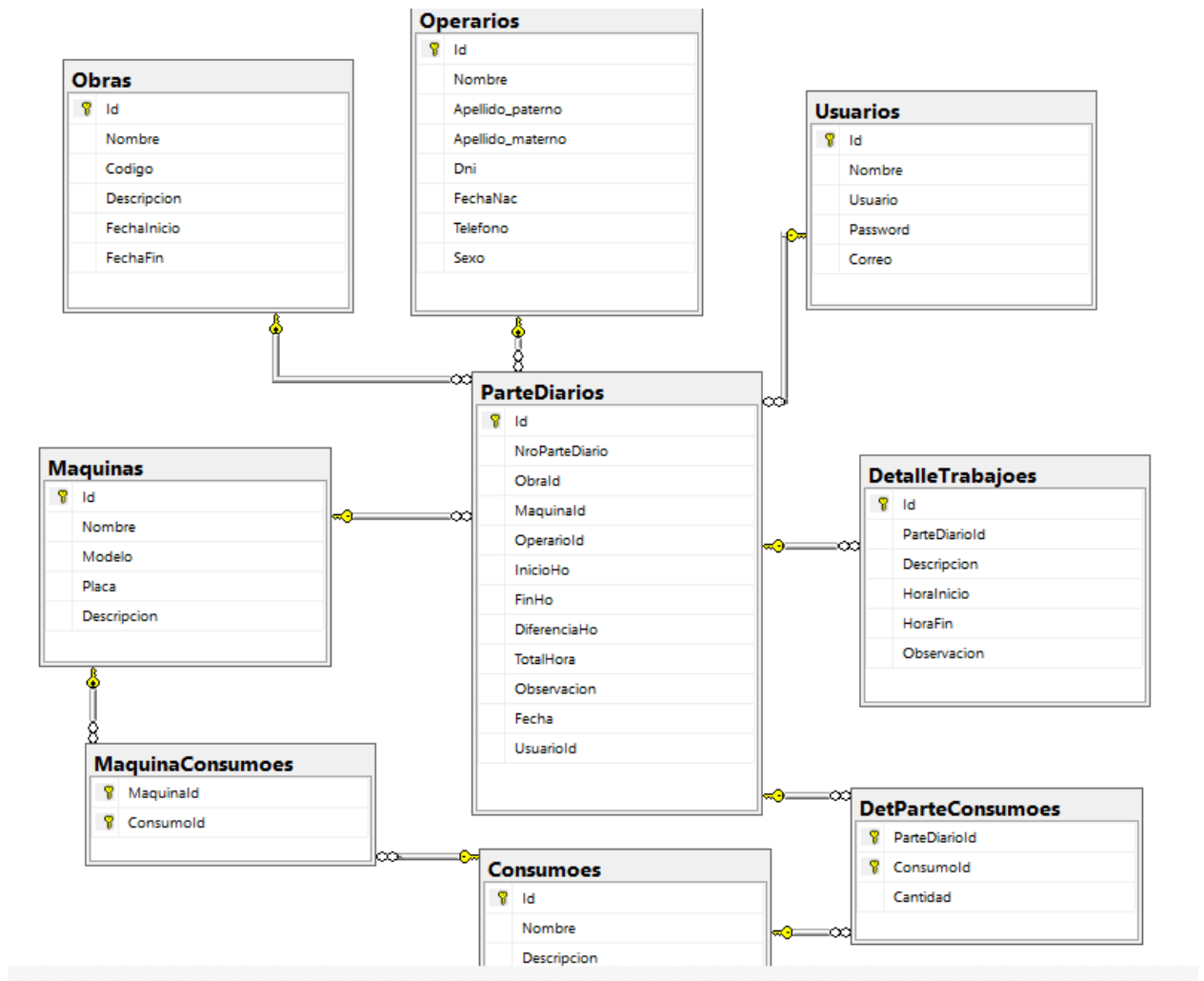


Figura 10: Entidad Relación Gestión de Horas Máquina.

Fuente: Elaboración propia.

3.4.2.3.2. ASP .Net Core y Entity Framework

El lenguaje de programación utilizado es C# parte de la plataforma de desarrollo .Net, se usará el IDE de Visual Studio 2019 con el framework ASP.NET Core 3, Entity Framework. Entity Framework es un ORM que da acceso a los datos de la base de datos SQL Server. En las siguientes figuras mostramos la instalación de Entity Framework y los comandos para generar los modelos del proyecto.

Estas opciones se encuentran al dar click derecho en el proyecto y seleccionar *Administración de Paquetes NuGet*. Instalar Microsoft.EntityFrameworkCore.SqlServer y Microsoft.EntityFrameworkCore.Tools.

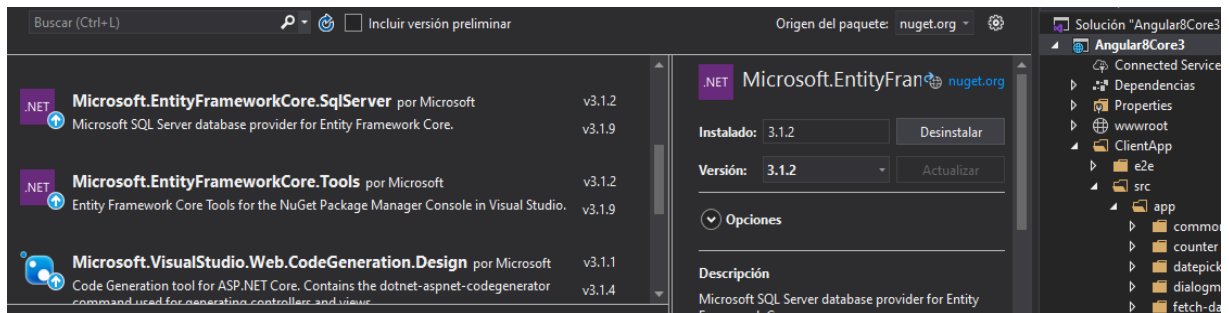


Figura 11: Instalación de Entity Framework.

Fuente: Elaboración propia.

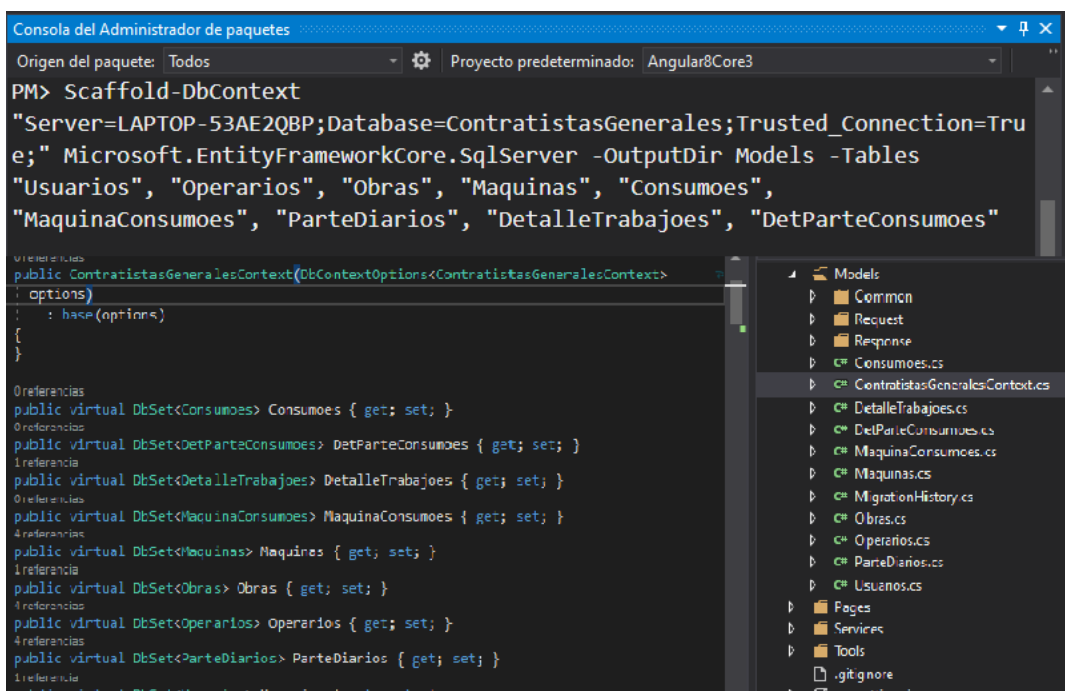


Figura 12: Comando Scaffolding para mapear los modelos.

Fuente: Elaboración propia.

3.4.2.3.3. Servicio Web Api y solicitudes Http Angular

Web Api se encarga de proveer datos desde angular hacia el servidor, en formato JSON ya sea códigos de estatus, string o arreglos, en esta parte del código se ve rutas consultas a la base de datos, también se ve los verbos o protocolos de Http que envían y reciben información. En las figuras 11 y 12 se muestra el controlador operario y la conexión con el servicio Http en angular.

```

7 using Angular8Core3.Models.Request;
8 using Angular8Core3.Models.Response;
9 using Microsoft.AspNetCore.Mvc;
10
11 namespace Angular8Core3.Controllers
12 {
13     [Route("api/[controller]")]
14     [ApiController]
15     public class OperariosController : ControllerBase
16     {
17         [HttpGet]
18         public IActionResult Get()...
19
20         [HttpPost]
21         public IActionResult Add(OperarioRequest oModel)...
22
23         [HttpPut]
24         public IActionResult Edit(OperarioRequest oModel)...
25
26         [HttpDelete("{Id}")]
27         public IActionResult Delete(int Id)...
28     }
29 }

```

Buscar en Explorador de soluciones (Ctrl+)

- Controllers
 - NewFolder
 - DetalleTrabajosController.cs
 - MaquinasController.cs
 - ObrasController.cs
 - OperariosController.cs
 - ParteDiarioController.cs
 - UsuariosController.cs
 - WeatherForecastController.cs
- Models
 - Common
 - Request
 - Response
 - Consumoes.cs
 - ContratistasGeneralesContext
 - DetalleTrabajos.cs
 - DetParteConsumoes.cs
 - MaquinaConsumoes.cs
 - Maquinas.cs
 - MigrationHistory.cs

Explorador de soluciones Team Explorer

Figura 13: Controlador operario, Web Api y sus protocolos.

Fuente: Elaboración propia.

```

1 import { Injectable } from '@angular/core';
2 import { HttpClient, HttpHeaders } from '@angular/common/http';
3 import { Observable } from 'rxjs';
4 import { Response } from '../models/response';
5 import { Operario } from '../models/operario';
6
7 const httpOption = {
8     headers: new HttpHeaders({
9         'Content-Type': 'application/json'
10     })
11 };
12
13 @Injectable({
14     providedIn: 'root'
15 })
16 export class OperarioService {
17
18     url: string = 'https://localhost:44319/api/Operarios';
19
20     constructor(private _http: HttpClient) { }
21
22     //Reliza la petición
23     getOperarios(): Observable<Response> {
24         return this._http.get<Response>(this.url)
25     }
26 }

```

Figura 14: Servicio operario Http angular.

Fuente: Elaboración propia.

3.4.2.3.4. Módulos Angular

Cada proyecto en angular tiene al menos un módulo donde se agrupa componentes, directivas, pipes y servicios. El módulo raíz convencionalmente se denomina AppModule. En nuestro sistema tenemos varios módulos para organizar nuestro sistema web, donde creamos el módulo operario, máquina, parte diario.

```
exports: [],
imports: [
  BrowserModule.withServerTransition({ appId: 'ng-cli-universal' }),
  HttpClientModule,
  FormsModule,
  NgbModule,
  ReactiveFormsModule,
  NgxPaginationModule,

  OperarioModule,
  MaquinaModule,
  PartediarioModule,

  RouterModule.forRoot([
    { path: '', component: HomeComponent, pathMatch: 'full', canActivate: [AuthGuard] },
    { path: 'counter', component: CounterComponent, canActivate: [AuthGuard] },
    { path: 'fetch-data', component: FetchDataComponent, canActivate: [AuthGuard] },
    { path: 'login', component: LoginComponent },
  ]),

],
providers: [
  { provide: HTTP_INTERCEPTORS, useClass: JwtInterceptor, multi: true }
],
bootstrap: [AppComponent]
})
export class AppModule { }
```

Figura 15: Módulo raíz AppModule angular.

Fuente: Elaboración propia.

```

10 import { NgxPaginationModule } from 'ngx-pagination';
11 import { DatePickerPopupModule } from '../datepicker-popup/datepicker-popup.module';
12
13 @NgModule({
14   declarations: [
15     OperarioComponent,
16     DialogComponent,
17     DeleteComponent
18   ],
19   imports: [
20     CommonModule,
21
22     DatePickerPopupModule,
23     BrowserModule,
24     NgbModule,
25     FormsModule,
26     ReactiveFormsModule,
27     NgxPaginationModule,
28     OperarioRoutingModule
29   ],
30   exports: [OperarioComponent],
31   bootstrap: [OperarioComponent],
32   entryComponents: [DialogComponent, DeleteComponent]
33 })
34 export class OperarioModule { }
35

```

Figura 16: Módulo OperarioModule angular.

Fuente: Elaboración propia.

3.4.2.3.5. Componentes Angular

Cada componente está asociado con un HTML que se visualizará en un entorno y cada componente se separa según a la lógica funcional, definiendo una clase que contiene datos. En la figura mostraremos los archivos asociados en el componente operario y estructura principal.

```

1 import { Component, OnInit, Output, EventEmitter, Input } from '@angular/core';
2 import { OperarioService } from '../services/operario.service';
3 import { DialogComponent } from '../dialog/dialog.component';
4 import { NgbModal } from '@ng-bootstrap/ng-bootstrap';
5 import { Operario } from '../models/operario';
6 import { DeleteComponent } from '../common/delete/delete.component';
7
8 @Component({
9   selector: 'app-operario',
10  templateUrl: './operario.component.html',
11  styleUrls: ['./operario.component.css']
12 })
13 export class OperarioComponent implements OnInit {
14
15   public lst: any[];
16   totalRecords: Number;
17   page: Number = 1;
18
19   @Input() public operario;
20   @Output() passEntry: EventEmitter<any> = new EventEmitter();
21
22   constructor(private serviceOperario: OperarioService, public dialog: NgbModal)
23   {
24
25

```

Figura 17: Componente OperarioComponent angular.

Fuente: Elaboración propia.

3.4.2.3.6. Código fuente de la lógica del módulo Parte Diario

En las figuras anteriores visualizamos las partes primordiales de construcción del sistema, ahora mostraremos el código fuente de Parte Diario, un módulo principal del sistema gestión horas máquina, el código muestra la parte lógica del registro, edición y listado; teniendo en cuenta los formatos, transacciones, consultas linq y otras herramientas.

```
public IActionResult Add(ParteDiarioRequest oModel)
{
    Respuesta oRespuesta = new Respuesta();

    try
    {
        using (ContratistasGeneralesContext db = new ContratistasGeneralesContext())
        {
            using (var transaction = db.Database.BeginTransaction())
            {
                try
                {
                    PartesDiarios objpartedia = new PartesDiarios
                    {
                        ObraId = oModel.ObraId,
                        MaquinaId = oModel.MaquinaId,
                        OperarioId = oModel.OperarioId,
                        InicioHo = oModel.InicioHo,
                        FinHo = oModel.FinHo,
                        DiferenciaHo = oModel.DiferenciaHo,
                        TotalHora = oModel.TotalHora,
                        Observacion = oModel.Observacion,
                        Fecha = oModel.Fecha
                    };

                    db.PartesDiarios.Add(objpartedia);
                    db.SaveChanges();

                    foreach (var modelDetTrabajo in oModel.DetalleTrabajos)
                    {
                        var detTrabajo = new Models.DetalleTrabajos();
                        detTrabajo.Descripcion = modelDetTrabajo.Descripcion;
                        detTrabajo.HoraInicio = modelDetTrabajo.HoraInicio;
                        detTrabajo.HoraFin = modelDetTrabajo.HoraFin;
                        detTrabajo.Observacion = modelDetTrabajo.Observacion;
                        detTrabajo.ParteDiarioId = objpartedia.Id;
                        db.DetalleTrabajos.Add(detTrabajo);
                        db.SaveChanges();
                    }
                    transaction.Commit();
                    oRespuesta.Exito = 1;
                }
            }
        }
    }
}
```

Figura 18: Código fuente del registro de Parte Diario.

Fuente: Elaboración propia.

```

public IActionResult Edit(ParteDiarioRequest oModel)
{
    Respuesta oRespuesta = new Respuesta();

    try
    {
        using (ContratistasGeneralesContext db = new ContratistasGeneralesContext())
        {
            using (var transaction = db.Database.BeginTransaction())
            {
                try
                {
                    ParteDiarios oPartedia = db.ParteDiarios.Find(oModel.Id);

                    oPartedia.ObraId = oModel.ObraId;
                    oPartedia.MaquinaId = oModel.MaquinaId;
                    oPartedia.OperarioId = oModel.OperarioId;
                    oPartedia.InicioHo = oModel.InicioHo;
                    oPartedia.FinHo = oModel.FinHo;
                    oPartedia.DiferenciaHo = oModel.DiferenciaHo;
                    oPartedia.TotalHora = oModel.TotalHora;
                    oPartedia.Observacion = oModel.Observacion;
                    oPartedia.Fecha = oModel.Fecha;
                    db.Entry(oPartedia).State = Microsoft.EntityFrameworkCore.EntityState.Modified;

                    db.SaveChanges();
                    foreach (var modelDetTrabajo in oModel.DetalleTrabajoes)
                    {
                        if (modelDetTrabajo.Id == 0)
                        {
                            var detTrabajo = new Models.DetalleTrabajoes();
                            detTrabajo.Id = modelDetTrabajo.Id;
                            detTrabajo.Descripcion = modelDetTrabajo.Descripcion;
                            detTrabajo.HoraInicio = modelDetTrabajo.HoraInicio;
                            detTrabajo.HoraFin = modelDetTrabajo.HoraFin;
                            detTrabajo.Observacion = modelDetTrabajo.Observacion;
                            detTrabajo.ParteDiarioId = oPartedia.Id;
                            db.AddRange(detTrabajo);
                            db.SaveChanges();
                        } else if ( modelDetTrabajo.Id != 0)
                        {
                            var detTrabajo = new Models.DetalleTrabajoes();
                            detTrabajo.Id = modelDetTrabajo.Id;
                            detTrabajo.Descripcion = modelDetTrabajo.Descripcion;
                            detTrabajo.HoraInicio = modelDetTrabajo.HoraInicio;
                            detTrabajo.HoraFin = modelDetTrabajo.HoraFin;
                            detTrabajo.Observacion = modelDetTrabajo.Observacion;
                            detTrabajo.ParteDiarioId = oPartedia.Id;
                            db.Update(detTrabajo);
                            db.SaveChanges();
                        }
                    }
                }
            }
        }
    }
}

```

Figura 19: Código fuente de editar Parte Diario.

Fuente: Elaboración Propia

```

[HttpGet]
0 referencias
public IActionResult Get()
{
    Respuesta oRespuesta = new Respuesta();
    try
    {
        using (ContratistasGeneralesContext db = new ContratistasGeneralesContext())
        {
            var dateparted = DateTime.Now.ToString("yyyy-MM-dd");
            var lst = db.ParteDiarios.Select(d => new
            {
                d.Id,
                d.ObraId,
                obraNombre = d.Obra.Nombre,
                d.MaquinaId,
                maquinaNombre = d.Maquina.Nombre,
                d.OperarioId,
                operarioNombre = d.Operario.Nombre + " " + d.Operario.ApellidoPaterno + " " + d.Operario.ApellidoMaterno,
                InicioHo = d.InicioHo.ToString("HH:mm:ss"),
                FinHo = d.FinHo.ToString("HH:mm:ss"),
                d.TotalHora,
                d.Observacion,
                d.Fecha,
                DetalleTrabajoes = d.DetalleTrabajoes.Select(f => new {
                    f.Id,
                    f.Descripcion,
                    HoraInicio = f.HoraInicio.ToString("HH:mm:ss"),
                    HoraFin = f.HoraFin.ToString("HH:mm:ss"),
                    f.Observacion,
                    f.ParteDiarioId
                })

            }).ToList()

            }).OrderByDescending(d => d.Id).ToList();
            oRespuesta.Exito = 1;
            oRespuesta.Data = lst;
        }
    }
    catch (Exception ex)

```

Figura 20: Código fuente de lista de los partes diarios.

Fuente: Elaboración propia.

3.4.2.5. Pruebas del sistema

3.4.2.4.1. Pruebas Unitarias

Cuando trabajamos con Apis o servicios del software necesitamos realizar unos test correspondientes, por lo cual usaremos la herramienta de Postman. En las figuras mostraremos las partes que una prueba unitaria debe tener, definir los requisitos, ejecución del test y resultados del test.

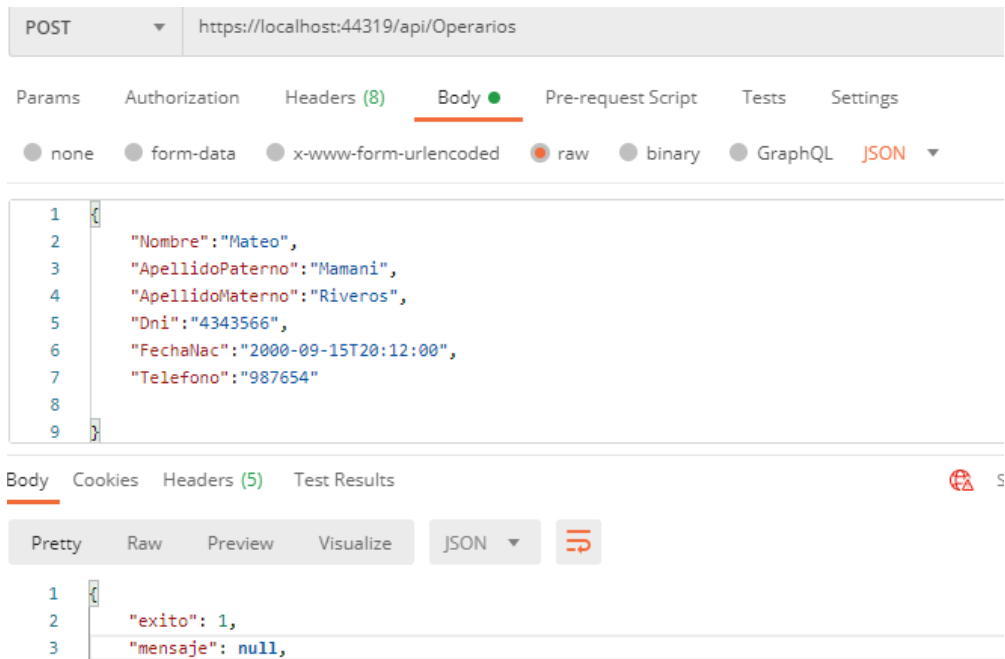


Figura 21: Prueba unitaria registrar operario en Postman.

Fuente: Elaboración propia.

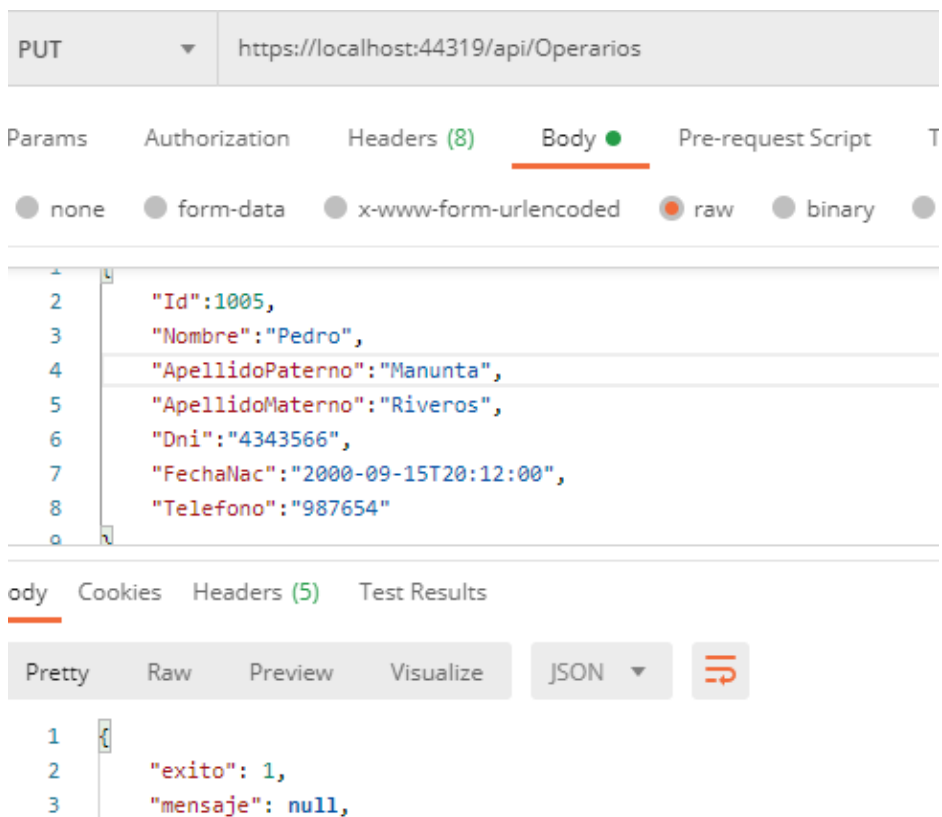


Figura 22: Prueba unitaria editar operario en Postman.

Fuente: Elaboración propia.

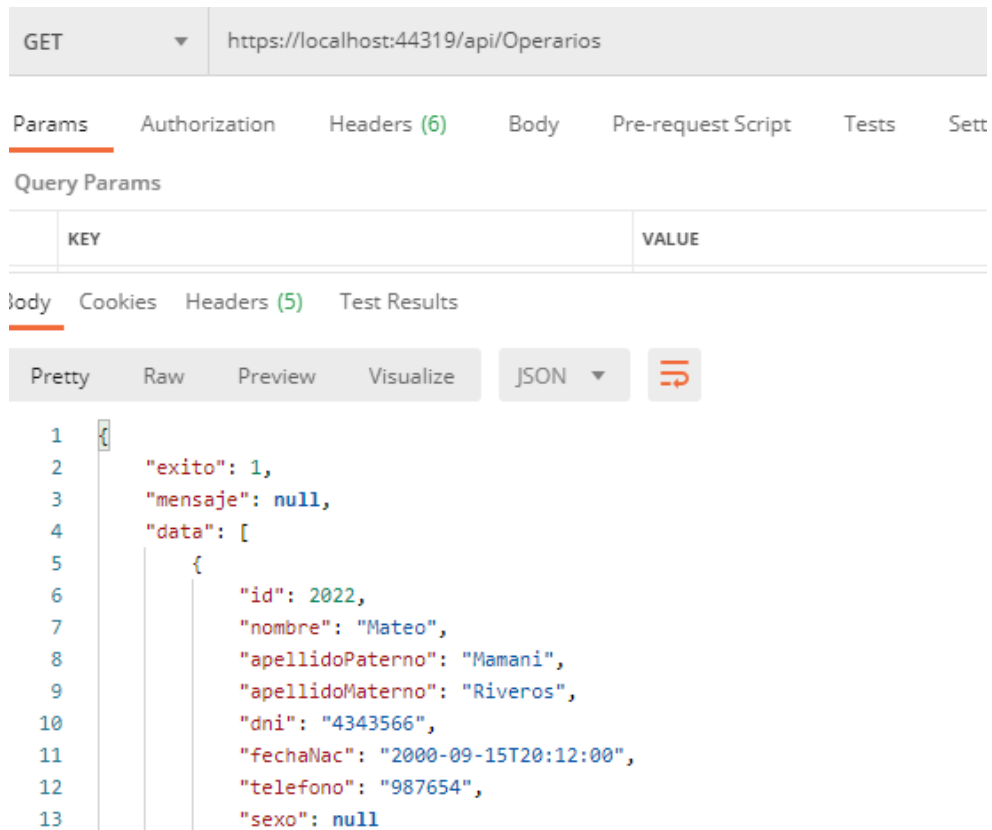


Figura 23: Prueba unitaria listar operario en Postman.

Fuente: Elaboración propia.

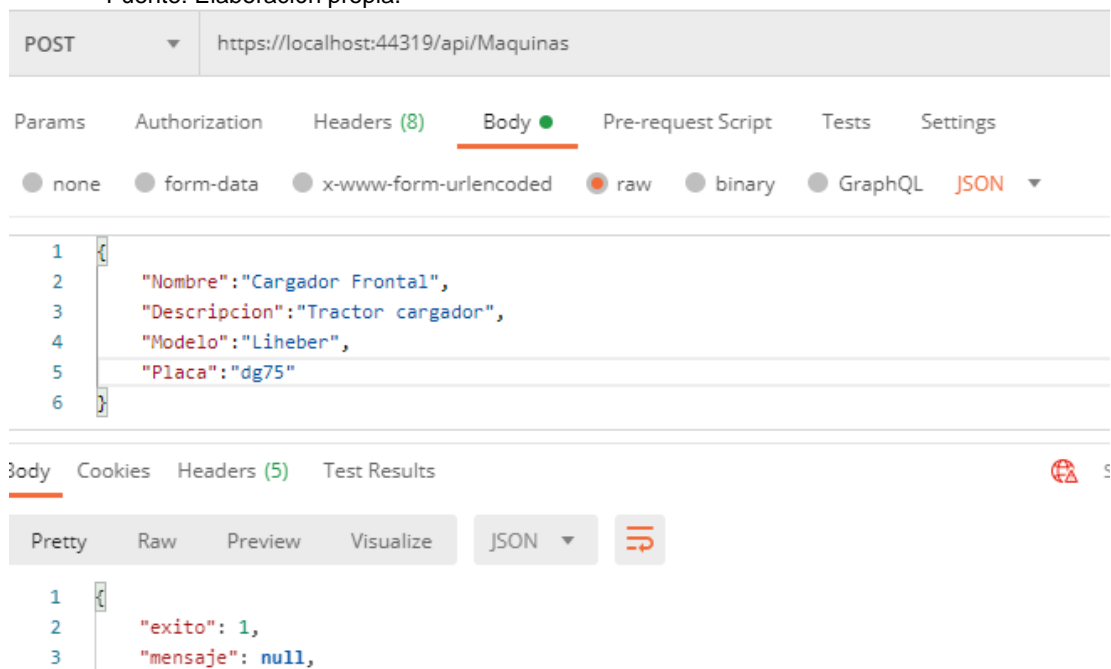


Figura 24: Prueba unitaria registrar máquina en Postman

Fuente: Elaboración propia.

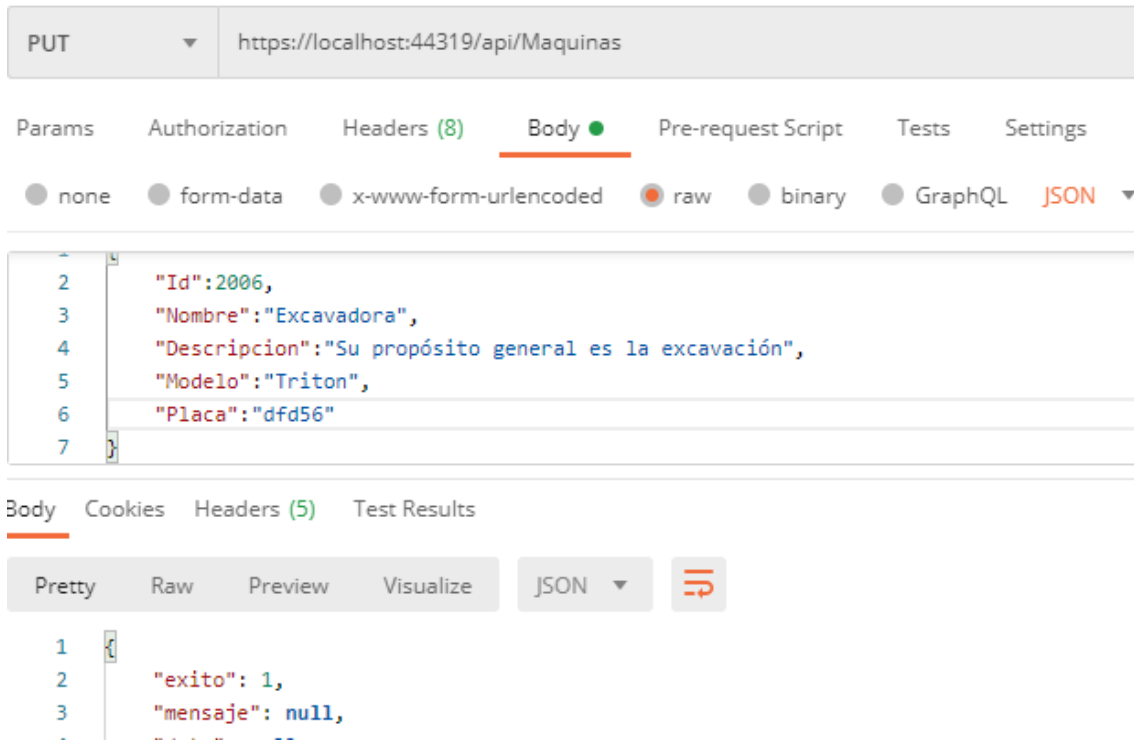


Figura 25: Prueba unitaria editar máquina en Postman.

Fuente: Elaboración propia.

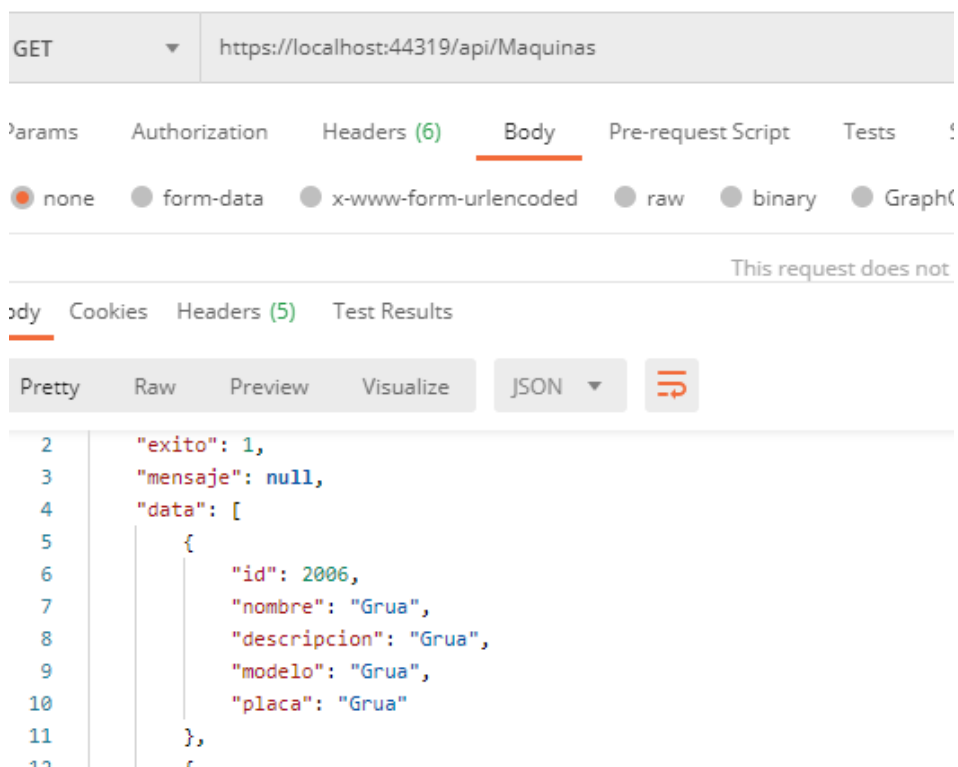


Figura 26: Prueba unitaria listar máquina en Postman.

Fuente: Elaboración propia.

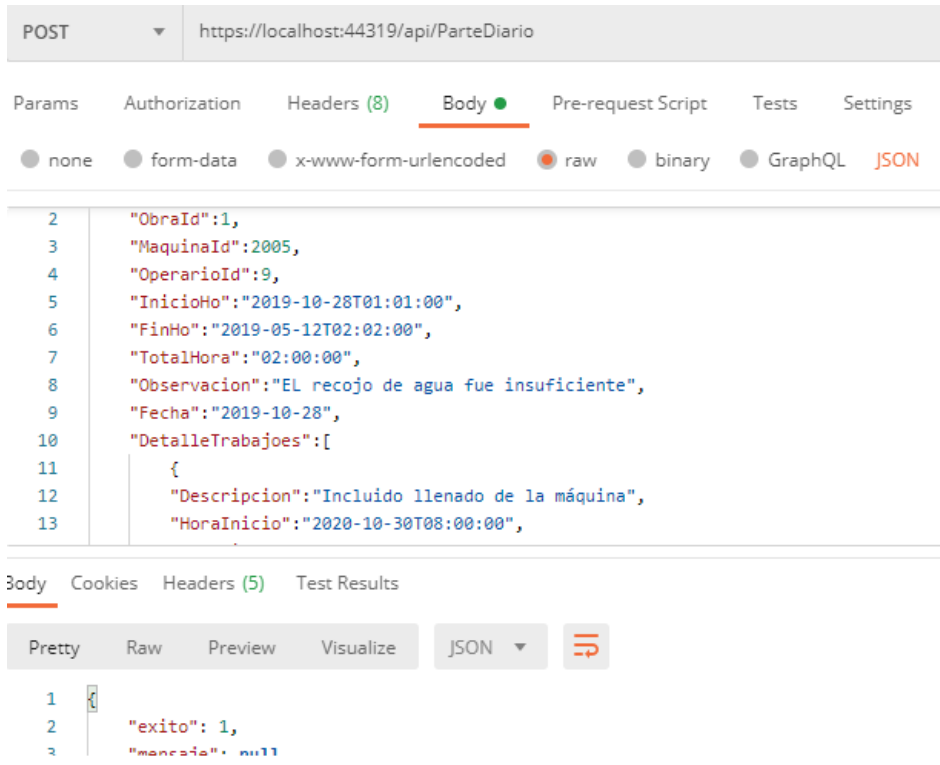


Figura 27: Prueba unitaria registrar parte diario en Postman.

Fuente: Elaboración propia.

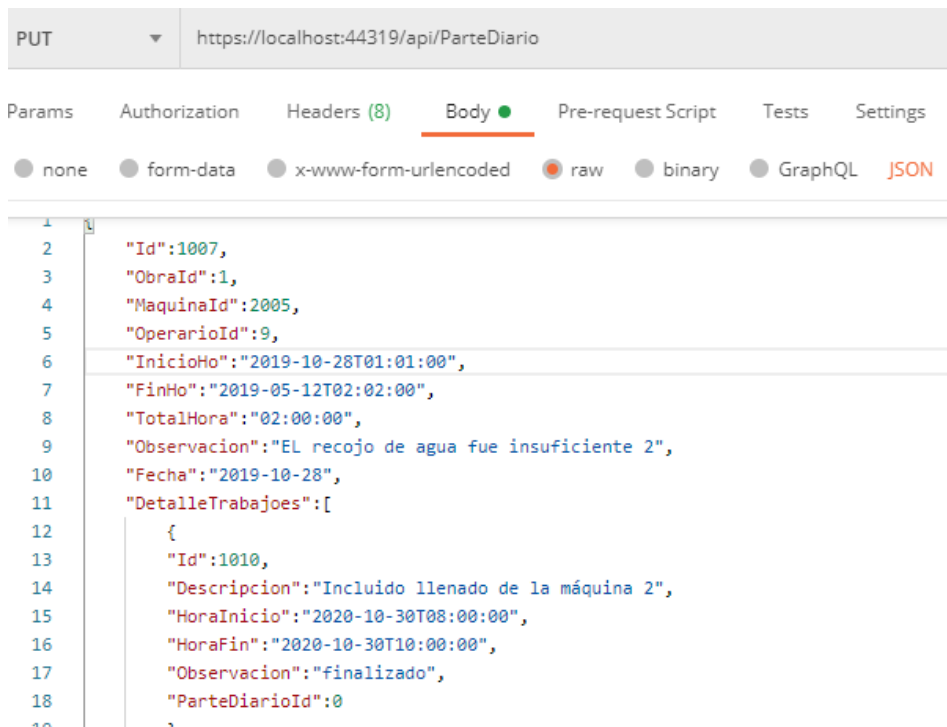


Figura 28: Prueba unitaria editar un parte diario en Postman.

Fuente: Elaboración propia.

```
1 {
2   "exito": 1,
3   "mensaje": null,
4   "data": [
5     {
6       "id": 1008,
7       "obraId": 1,
8       "obraNombre": "Carretera salida Cuzco km 38",
9       "maquinaId": 2005,
10      "maquinaNombre": "Cisterna",
11      "operarioId": 9,
12      "operarioNombre": "Camilo Llacho Quispe",
13      "inicioHo": "01:01:00",
14      "finHo": "02:02:00",
15      "totalHora": "02:00:00",
16      "observacion": "EL recojo de agua fue insuficiente",
17      "fecha": "2019-10-28T00:00:00",
18      "detalleTrabajoes": [
```

Figura 29: Prueba unitaria listar parte diario en Postman.

Fuente: Elaboración propia.

3.4.2.4.2. Casos de Prueba de Aceptación

Las pruebas de aceptación es la comprobación correcta de la funcionalidad del sistema web, la cual se hizo con el administrador de la empresa Contratistas Generales, quien es el encargado de registro de partes diarios. Se probó la funcionalidad de registro, edición y listado del Operario; también el registro, edición y listado de máquinas; finalmente el registro, edición y listado de los Partes Diarios.

Tabla 42.

Prueba 1 – Historia de usuario 1: Crear operario.

Caso de Prueba de Aceptación	
Número: 1	Historia de Usuario: 1 Crear operario
Nombre:	Registrar datos del operario.
Descripción:	En el formulario nuevo operario, ingresar los datos del operario y guardar.
Condición al ejecutar:	El botón guardar se activa cuando los datos necesarios se llenen.
Entrada:	El usuario llena los campos del formulario nuevo operario.
Resultado Esperado:	Al dar click en guardar, los datos llenados crean un nuevo operario.
Evaluación:	Resultado satisfactorio.

Fuente: Elaboración propia.

The image shows a web application interface for managing operators. A modal window titled "Nuevo Operario" is open, displaying a form with the following fields: "Nombre:", "Apellido Paterno:", "Apellido Materno:", "Dni:", "Fecha de nac.:" (with a date picker set to "yyyy-mm-dd"), and "Telefono:". At the bottom of the form are "Guardar" and "Cancelar" buttons. In the background, a table lists existing operators with columns for "Operario Id", "Nombre", "Dni", "Fecha de nac.", and "Telefono". The table contains data for operators from 2021 down to 2015. To the right of the table, there are "Opciones" (Edit and Delete) buttons for each row.

Figura 30: Formulario para crear un nuevo operario.

Fuente: Elaboración propia.

Tabla 43.

Prueba 2 – Historia de usuario 2: Editar operario.

Caso de Prueba de Aceptación	
Número: 2	Historia de Usuario: 2 Editar operario
Nombre: Editar datos del operario.	
Descripción: En el formulario editar operario, modificar datos del operario y guardar.	
Condición al ejecutar: El botón guardar se activa si los campos necesarios estén llenos.	
Entrada: El usuario modifica los campos en el formulario editar operario.	
Resultado Esperado: Al dar click en guardar se actualizará los datos del operario.	
Evaluación: Resultado satisfactorio.	

Fuente: Elaboración propia.

The image shows a web application interface for managing operators. A modal window titled "Editar Operario" is open, displaying a form with the following fields:

- Nombre: Federico
- Apellido Paterno: Ramos
- Apellido Materno: Zenteno
- Dni: 9874523
- Fecha de nac.: 2000-10-11
- Telefono: 987412

At the bottom of the modal are two buttons: "Guardar" (highlighted in blue) and "Cancelar".

In the background, a table lists operators with columns for "Operario Id", "Nombre", "Apellido", "Dni", "Telefono", and "Fecha de nac.". Each row has "Edit" and "Delete" buttons.

Operario Id	Nombre	Apellido	Dni	Telefono	Fecha de nac.	Acciones	
2021	Federico	Ramos	9874523	987412	2000-10-11	Edit Delete	
2020	Mateo	Zenteno				Edit Delete	
2019	Lucas					Edit Delete	
2018	David					Edit Delete	
2017	Yilver					Edit Delete	
2016	Darwin					Edit Delete	
2015	Emerson	Cho	Qui	987452	9874252	2003-09-24	Edit Delete
2014	William 3	Calisaya	Calisaya	4343566	987654	2000-01-06	Edit Delete
2012	Roy	Vela	Torres	4343566	987654	1995-01-27	Edit Delete
2011	Vilma	Aguirre	Roman	78412152	98562147	2000-01-06	Edit Delete

Figura 31: Formulario para editar un operario.

Fuente: Elaboración propia.

Tabla 44.

Prueba 3 – Historia de usuario 3: Listar operario.

Caso de Prueba de Aceptación	
Número: 3	Historia de Usuario: 3 Listar operario
Nombre: Listar registros del operario.	
Descripción: En la tabla de gestión operarios muestra los registros de los operarios.	
Condición al ejecutar: Seleccionar la pestaña de operario.	
Entrada: El usuario selecciona la pestaña de operario, en la pantalla principal.	
Resultado Esperado: Se visualizará la lista de registros de operarios con sus opciones.	
Evaluación: Resultado satisfactorio.	

Fuente: Elaboración propia.

Operario Id	Nombre	Apellido Paterno	Apellido Materno	Dni	Teléfono	Fecha Nacimiento	Opciones
2021	Federico	Ramos	Zenteno	9874523	987412	2000-10-11	Edit Delete
2020	Mateo	Mamani	Riveros	4343566	987654	2000-09-15	Edit Delete
2019	Lucassnombre	Mamani	Calisaya	4343566	987654	2000-09-15	Edit Delete
2018	David	Hancocallo	Chuma	9874252	987423	1993-03-18	Edit Delete
2017	Yilver	Ticona	Perez	9856328	987528695	1994-12-14	Edit Delete
2016	Darwin	Ocho	Agui	768765	98563254	2003-09-24	Edit Delete
2015	Emerson	Cho	Qui	987452	9874252	2003-09-24	Edit Delete
2014	William 3	Calisaya	Calisaya	4343566	987654	2000-01-06	Edit Delete
2012	Roy	Vela	Torres	4343566	987654	1995-01-27	Edit Delete
2011	Vilma	Aguirre	Roman	78412152	98562147	2000-01-06	Edit Delete

< Previous 1 2 3 Next >

Figura 32: Vista de registros de los operarios.

Fuente: Elaboración propia.

Tabla 45.

Prueba 4 – Historia de usuario 4: Crear máquina.

Caso de Prueba de Aceptación	
Número: 4	Historia de Usuario: 4 Crear máquina
Nombre: Registrar datos de la máquina.	
Descripción: En el formulario nueva máquina, ingresar los datos de la máquina y guardar.	
Condición al ejecutar: El botón guardar se activa cuando los datos necesarios se llenen.	
Entrada: El usuario llena los campos del formulario nueva máquina.	
Resultado Esperado: Al dar click en guardar, los datos llenados crean una máquina.	
Evaluación: Resultado satisfactorio.	

Fuente: Elaboración propia.

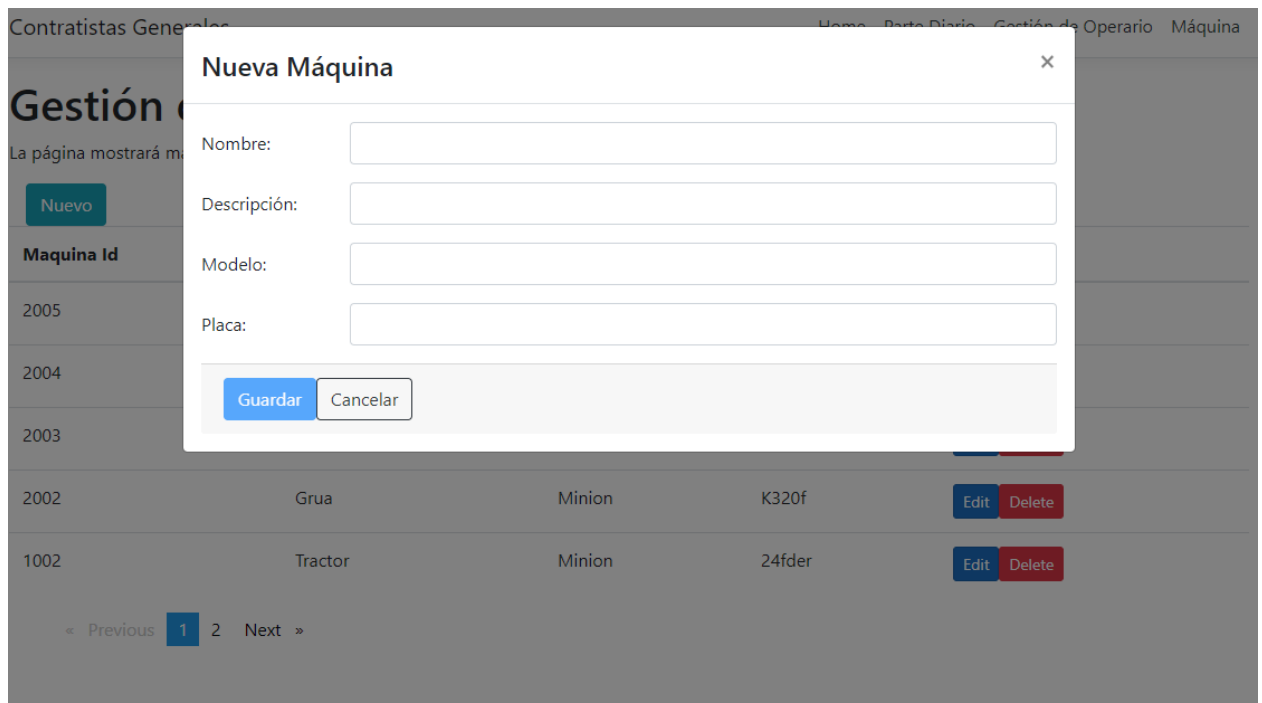


Figura 33: Formulario para crear una nueva máquina.

Fuente: Elaboración propia.

Tabla 46.

Prueba 5 – Historia de usuario 5: Editar máquina.

Caso de Prueba de Aceptación	
Número: 5	Historia de Usuario: 5 Editar máquina
Nombre: Editar datos de la máquina.	
Descripción: En el formulario editar máquina, modificar datos de la máquina y guardar.	
Condición al ejecutar: El botón guardar se activa si los campos necesarios estén llenos.	
Entrada: El usuario modifica los campos en el formulario editar máquina.	
Resultado Esperado: Al dar click en guardar se actualizará los datos de la máquina.	
Evaluación: Resultado satisfactorio.	

Fuente: Elaboración propia.

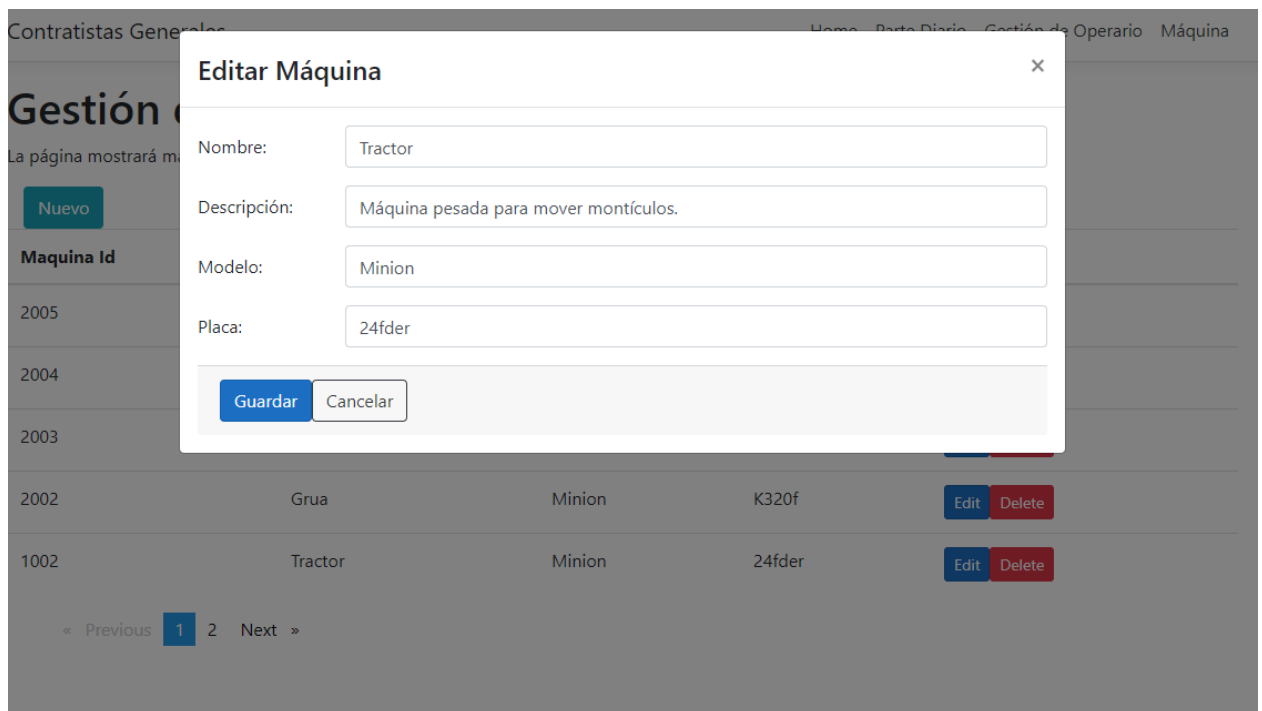


Figura 34: Formulario para editar un operario.

Fuente: Elaboración propia.

Tabla 47.

Prueba 6 – Historia de usuario 6: Listar máquina.

Caso de Prueba de Aceptación

Número: 6	Historia de Usuario: 6 Listar máquina
-----------	---------------------------------------

Nombre: Listar registros de la máquina.

Descripción: En la tabla de gestión máquinas muestra los registros de las máquinas.

Condición al ejecutar: Seleccionar la pestaña de Máquina.

Entrada: El usuario selecciona la pestaña Máquina, en la pantalla principal.

Resultado Esperado: Se visualizará la lista de registros de las máquinas con sus opciones.

Evaluación: Resultado satisfactorio.

Fuente: Elaboración propia

Contratistas Generales Home Parte Diario Gestión de Operario Máquina

Gestión de Máquinas

La página mostrará máquinas, creará y actualizará

[Nuevo](#)

Maquina Id	Nombre	Modelo	Placa	Opciones
2005	Cisterna	wert	2fed	Edit Delete
2004	Aplanadora	Canon	hnbfw1	Edit Delete
2003	Volquete	Kalix	hbfbw3	Edit Delete
2002	Grua	Minion	K320f	Edit Delete
1002	Tractor	Minion	24fder	Edit Delete

« Previous [1](#) 2 Next »

Figura 35: Vista de registros de las máquinas.

Fuente: Elaboración propia.

Tabla 48.

Prueba 7 – Historia de usuario 7: Crear parte diario.

Caso de Prueba de Aceptación	
Número: 7	Historia de Usuario: 7 Crear parte diario
Nombre: Registrar datos del parte diario.	
Descripción: En el formulario nuevo parte diario, ingresar los datos del parte diario.	
Condición al ejecutar: El botón guardar se activa cuando los datos necesarios se llenen.	
Entrada: El usuario llena los campos del formulario nuevo parte diario.	
Resultado Esperado: Al dar click en guardar, los datos llenados crean un nuevo parte diario.	
Evaluación: Resultado satisfactorio.	

Fuente: elaboración propia.

The image shows a web form titled "Nuevo Parte Diario" with a close button (X) in the top right corner. The form is divided into three sections: "1 general", "2 detalle", and "3 save". The "1 general" section is currently active and contains the following fields: "Obra:" (text input), "Máquina:" (text input), "Operario:" (text input), "Fecha:" (date picker with "yyyy-mm-dd" placeholder), "Hora inicio de trabajo:" (time picker with "HH" and "MM" boxes and up/down arrows), "Hora fin de trabajo:" (time picker with "HH" and "MM" boxes and up/down arrows), and "Observaciones:" (text area). A blue "Continue" button is located at the bottom left of the form. On the right side of the form, there is a vertical sidebar with a header "Opciones" and several "Edit" buttons.

Figura 36: Formulario para crear un nuevo parte diario.

Fuente: Elaboración propia.

Tabla 49.

Prueba 8 – Historia de usuario 8: Editar parte diario.

Caso de Prueba de Aceptación	
Número: 8	Historia de Usuario: 8 Editar parte diario
Nombre: Editar datos del parte diario.	
Descripción: En el formulario editar parte diario, modificar datos del parte diario y guardar.	
Condición al ejecutar: El botón guardar se activa si los campos necesarios estén llenos.	
Entrada: El usuario modifica los campos del formulario editar parte diario.	
Resultado Esperado: Al dar click en guardar se actualizará los datos del parte diario.	
Evaluación: Resultado satisfactorio.	

Fuente: Elaboración propia.

The image shows a web application interface for editing a daily part. The main window is titled 'Editar Parte Diario' and has a close button (X) in the top right corner. Below the title, there are three tabs: '1 general', '2 detalle', and '3 save'. The 'detalle' tab is currently selected. In the 'detalle' tab, there is a button 'Agregar detalle' at the top left. Below it is a table with four columns: 'Descripción', 'Hora Inicio', 'Hora Final', and 'Observación'. The 'Descripción' column contains the text 'Salida Cabanillas'. The 'Hora Inicio' column shows '10 : 00' with up and down arrows for adjustment. The 'Hora Final' column shows '12 : 00' with up and down arrows. The 'Observación' column is empty. Below the table, there is a red 'Remove' button and a blue 'Continue' button.

Figura 37: Formulario para editar un parte diario.

Fuente: Elaboración propia.

Tabla 50.

Prueba 9 – Historia de usuario 9: Listar parte diario

Caso de Prueba de Aceptación

Número: 9	Historia de Usuario: 9 Listar parte diario
-----------	--------------------------------------------

Nombre: Listar registros de los partes diarios.

Descripción: En la tabla de gestión de parte diario muestra los registros de los partes diarios.

Condición al ejecutar: Seleccionar la pestaña de Parte Diario.

Entrada: El usuario selecciona la pestaña de Parte Diario, en la pantalla principal.

Resultado Esperado: Se visualizará la lista de registros de partes diarios con sus opciones.

Fuente: Elaboración propia.

Contratistas Generales Home Parte Diario Gestión de Operario Máquin

Gestión de Partes Diarios

.a página mostrará partes diarios,creará y actualizará

[Nuevo](#)

Maquina	Operario	Fecha	Hora inicio operario	Hora fin operario	Hora total máquina	Opciones
Cisterna	Pedro Manunta Riveros	2019-10-28	01:01:00	02:02:00	02:30:00	Edit Delete
Volquete	Camilo Llacho Quispe	2020-10-14	08:00:00	16:00:00	02:00:00	Edit Delete
Cisterna	Federico Ramos Zenteno	2019-10-28	01:01:00	02:02:00	02:00:00	Edit Delete
Remolcadora	Pamela Tízón Humayasi	2019-03-14	09:00:00	16:00:00	00:40:00	Edit Delete
Grua	Roy Vela Torres	2003-09-24	10:30:00	17:00:00	03:15:00	Edit Delete
Tractor	Pedro Manunta Riveros	2001-09-13	08:10:00	16:00:00	02:11:00	Edit Delete

« Previous [1](#) 2 Next »

Figura 38: Vista de registros de los partes diario.

Fuente: Elaboración propia.

Tabla 51.

Listar, crear y editar obras.

Caso de Prueba de Aceptación	
Número: 10	Historia de Usuario: 10 Gestión de Obras
Nombre: Listar, crear y editar obras.	
Descripción: Se puede crear, editar y listar obras en el sistema.	
Condición al ejecutar: Seleccionar la pestaña obras.	
Entrada: El usuario selecciona la pestaña de Obras, en la pantalla principal.	
Resultado Esperado: Mostrará el listado de obras y los botones de nuevo y editar.	

Fuente: Elaboración propia.

Gestión de Obras

La página mostrará obras, creará y actualizará

Nuevo		<input type="text" value="Buscar"/> <input type="text" value="Search..."/>		
Obra Id	Nombre	Fecha de Inicio	Fecha de Fin	Opciones
2002	Construir reservorio en el poblado de Cabanillas	2020-05-13T00:00:00	2020-11-17T00:00:00	Edit Delete
1002	Mantenimiento de la carretera coporaque km 1122	2020-08-15T00:00:00	2020-12-10T00:00:00	Edit Delete
5	Mejoramiento de los servicios en la institución educativa Horacio Zeballos Gamez Juliaca - San Roman - Puno	2020-08-20T00:00:00	2021-01-06T00:00:00	Edit Delete
1	Carretera salida Cuzco km 38	2019-04-03T00:00:00	2020-01-04T00:00:00	Edit Delete
« Previous 1 2 Next »				

Figura 39: Vista de registros de las obras.

Elaboración propia.

Nueva Obra [X]

Nombre:

Descripción:

Fecha Inicio:

Fecha Fin:

Figura 40: Formulario de nueva Obra.

Fuente: Elaboración propia.

Editar Obra [X]

Nombre:

Descripción:

Fecha Inicio:

Fecha Fin:

Figura 41: Formulario de editar obra.

Fuente: Elaboración propia.

Tabla 52.

Listar, crear y editar consumos.

Número: 11

Historia de Usuario: 11 Gestión de Consumo

Nombre: Listar, crear y editar consumos.

Descripción: Se puede crear, editar y listar obras en el sistema.

Condición al ejecutar: Seleccionar la pestaña consumo.

Entrada: El usuario selecciona la pestaña de consumo, en la pantalla principal.

Resultado Esperado: Mostrará el listado de consumos y los botones de nuevo y editar.

Fuente: Elaboración propia.

Contratistas Generales Home Parte Diario Operario Máquina Obra Consumo Vista Cerrar Sesión

Gestión de Consumos

La página mostrará consumos, creará y actualizará


[Nuevo](#)

Obra Id	Nombre	Descripción	Opciones
6	Grasa	Material de mantenimiento	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
5	Aceite de transporte	Material de mantenimiento	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
4	Petroleo Gls.	Combustible de maquinaria	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	Aceite Hidráulico	Material de mantenimiento	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	Aceite motor	Material de mantenimiento	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

« Previous **1** 2 Next »

Figura 42: Vista de registros de los consumos.

Fuente: Elaboración propia.



Nuevo Consumo

Nombre:

Descripción:

Guardar Cancelar

Figura 43: Formulario de nuevo consumo.

Fuente: Elaboración propia.



Editar Consumo

Nombre:

Descripción:

Guardar Cancelar

Grasa Material de mantenimiento Edit Delete

Figura 44: Formulario de editar consumo.

Fuente: Elaboración propia.

3.4.2.4.3. Pruebas de Aceptación con Selenium

Selenium son conjuntos de herramientas para facilitar la obtención de pruebas en una aplicación web donde graba pruebas de forma automática, esta herramienta crea scripts que realizan acciones que se pueden editar y usarlas cuando se necesite ejecutar los test funcionales. Se generó los scripts automatizados de los módulos principales del sistema de gestión de máquinas, crear, editar un parte diario, operario y máquina.

```

0 referencias
28 public void Listadepartediario() {
29     driver.Navigate().GoToUrl("https://localhost:44319/");
30     driver.Manage().Window.Size = new System.Drawing.Size(1382, 744);
31     driver.FindElement(By.LinkText("Parte Diario")).Click();
32     driver.FindElement(By.Name("filterOper")).Click();
33     driver.FindElement(By.Name("filterOper")).SendKeys("com");
34 }
35 }
36

```

Figura 45: Script lista de partes diarios.

Fuente: Elaboración propia.

The screenshot displays the Selenium IDE interface. On the left, a search bar contains 'Search tests...' and a list of tests includes 'listar parte diario'. The main area shows a table of commands for the selected test, with a URL of 'https://localhost:44319'.

	Command	Target	Value
1	open	/	
2	set window size	1382x744	
3	click	linkText=Parte Diario	
4	click	name=filterOper	
5	type	name=filterOper	rem

Below the table are input fields for Command, Target, Value, and Description. At the bottom, the 'Log' tab is active, showing the following entries:

Log	Reference
2. setWindowSize on 1382x744 OK	15:41:54
3. click on linkText=Parte Diario OK	15:41:54
4. click on name=filterOper OK	15:42:02
5. type on name=filterOper with value rem OK	15:42:03
'listar parte diario' completed successfully	15:42:04

Figura 46: Resultado de script listar parte diarios en Selenium.

Fuente: Elaboración propia.


```

28 public void NuevoParteDiario() {
29     driver.Navigate().GoToUrl("https://localhost:44319/");
30     driver.Manage().Window.Size = new System.Drawing.Size(1382, 744);
31     driver.FindElement(By.LinkText("Parte Diario")).Click();
32     driver.FindElement(By.Id("nuevoPDiario")).Click();
33     {
34         var element = driver.FindElement(By.Id("nuevoPDiario"));
35         Actions builder = new Actions(driver);
36         builder.MoveToElement(element).Perform();
37     }
38     {
39         var element = driver.FindElement(By.TagName("body"));
40         Actions builder = new Actions(driver);
41         builder.MoveToElement(element, 0, 0).Perform();
42     }
43     driver.FindElement(By.CssSelector(".form-group:nth-child(1) > .col-sm-10 > .form-control")).Click();
44     driver.FindElement(By.CssSelector(".form-group:nth-child(1) > .col-sm-10 > .form-control")).SendKeys("ca");
45     driver.FindElement(By.CssSelector("#ngb-typeahead-2-0 > ngb-highlight")).Click();
46     driver.FindElement(By.CssSelector(".form-group:nth-child(2) > .col-sm-10 > .form-control")).Click();
47     driver.FindElement(By.CssSelector(".form-group:nth-child(2) > .col-sm-10 > .form-control")).SendKeys("gr");
48     driver.FindElement(By.CssSelector("#ngb-highlight")).Click();
49     driver.FindElement(By.CssSelector(".col-sm-10 > .ng-invalid")).Click();
50     driver.FindElement(By.CssSelector(".form-group:nth-child(3) > .col-sm-10 > .form-control")).SendKeys("da");
51     driver.FindElement(By.CssSelector("#ngb-typeahead-1-3 .ngb-highlight")).Click();
52     driver.FindElement(By.Name("dp")).Click();
53     driver.FindElement(By.Name("dp")).SendKeys("2021-02-17");
54     driver.FindElement(By.CssSelector(".form-group:nth-child(1) .ngb-tp-input-container:nth-child(1) > .ngb-tp-input")).SendKeys("05");
55     driver.FindElement(By.CssSelector(".form-group:nth-child(1) .ngb-tp-input-container:nth-child(3) > .ngb-tp-input")).SendKeys("30");
56     driver.FindElement(By.CssSelector(".form-group:nth-child(2) .ng-invalid .ngb-tp-input-container:nth-child(1) > .ngb-tp-input")).SendKeys("12");
57     driver.FindElement(By.CssSelector(".form-group:nth-child(2) .ng-invalid .ngb-tp-input-container:nth-child(3) > .ngb-tp-input")).SendKeys("00");
58     driver.FindElement(By.CssSelector(".btn:nth-child(7)")).Click();
59     driver.FindElement(By.Id("descripcion0")).Click();
60     driver.FindElement(By.Id("descripcion0")).SendKeys("transporte material");
61     driver.FindElement(By.CssSelector(".form-group:nth-child(2) > app-timepicker-dynamic .ngb-tp-input-container:nth-child(1) > .ngb-tp-input")).SendKeys("05");
62     driver.FindElement(By.CssSelector(".form-group:nth-child(2) > app-timepicker-dynamic .ngb-tp-input-container:nth-child(3) > .ngb-tp-input")).SendKeys("30");
63     driver.FindElement(By.CssSelector(".form-group:nth-child(3) .ngb-tp-input-container:nth-child(1) > .ngb-tp-input")).SendKeys("07");
64     driver.FindElement(By.CssSelector(".form-group:nth-child(3) .ngb-tp-input-container:nth-child(3) > .ngb-tp-input")).SendKeys("15");
65     driver.FindElement(By.LinkText("Agregar detalle")).Click();
66     driver.FindElement(By.Id("descripcion1")).Click();
67     driver.FindElement(By.Id("descripcion1")).SendKeys("armazon de fierro");
68     driver.FindElement(By.CssSelector(".card:nth-child(3) .form-group:nth-child(2) .ngb-tp-input-container:nth-child(1) > .ngb-tp-input:nth-child(2)")).SendKeys("00");
69     driver.FindElement(By.CssSelector(".card:nth-child(3) .form-group:nth-child(2) .ngb-tp-input-container:nth-child(3) > .ngb-tp-input:nth-child(2)")).SendKeys("00");
70     driver.FindElement(By.CssSelector(".card:nth-child(3) .form-group:nth-child(3) .ngb-tp-input-container:nth-child(1) > .ngb-tp-input:nth-child(2)")).SendKeys("12");
71     driver.FindElement(By.CssSelector(".card:nth-child(3) .form-group:nth-child(3) .ngb-tp-input-container:nth-child(3) > .ngb-tp-input:nth-child(2)")).SendKeys("00");
72     driver.FindElement(By.CssSelector(".btn:nth-child(4)")).Click();
73     driver.FindElement(By.CssSelector(".card-footer > .btn-primary")).Click();

```

Figura 47: Script nuevo parte diario.

Fuente: Elaboración propia.

33	send keys	css= card:nth-child(3) .form-group:nth-child(2) .ngb-tp-input-container:nth-child(3) > .ngb-tp-input:nth-child(2)	uu
34	send keys	css= card:nth-child(3) .form-group:nth-child(3) .ngb-tp-input-container:nth-child(1) > .ngb-tp-input:nth-child(2)	12
35	send keys	css= card:nth-child(3) .form-group:nth-child(3) .ngb-tp-input-container:nth-child(3) > .ngb-tp-input:nth-child(2)	00
36	click	css= btn:nth-child(4)	
37	click	css= card-footer > .btn-primary	

Command: //

Target:

Value:

Description:

Log	Reference
33. sendKeys on css= card:nth-child(3) .form-group:nth-child(2) .ngb-tp-input-container:nth-child(3) > .ngb-tp-input:nth-child(2) with value 00 OK	
34. sendKeys on css= card:nth-child(3) .form-group:nth-child(3) .ngb-tp-input-container:nth-child(1) > .ngb-tp-input:nth-child(2) with value 12 OK	
35. sendKeys on css= card:nth-child(3) .form-group:nth-child(3) .ngb-tp-input-container:nth-child(3) > .ngb-tp-input:nth-child(2) with value 00 OK	
36. click on css= btn:nth-child(4) OK	
37. click on css= card-footer > .btn-primary OK	
'nuevo parte diario' completed successfully	

Figura 48: Resultado de nuevo parte diario en Selenium.

Fuente: Elaboración propia.

```

public void Editarpartediario() {
    driver.Navigate().GoToUrl("https://localhost:44319/");
    driver.Manage().Window.Size = new System.Drawing.Size(1382, 744);
    driver.FindElement(By.LinkText("Parte Diario")).Click();
    driver.FindElement(By.CssSelector("tr:nth-child(1) .btn-primary")).Click();
    driver.FindElement(By.CssSelector(".form-group:nth-child(2) > .col-sm-10 > .form-control")).Click();
    driver.FindElement(By.CssSelector(".form-group:nth-child(2) > .col-sm-10 > .form-control")).SendKeys("\n");
    driver.FindElement(By.CssSelector(".form-group:nth-child(2) > .col-sm-10 > .form-control")).SendKeys("gr");
    driver.FindElement(By.CssSelector(".ngb-highlight")).Click();
    driver.FindElement(By.CssSelector(".btn-outline-secondary")).Click();
    driver.FindElement(By.CssSelector(".ngb-dp-week:nth-child(3) > .ngb-dp-day:nth-child(3) > .btn-light")).Click();
    driver.FindElement(By.CssSelector(".btn:nth-child(7)")).Click();
    driver.FindElement(By.CssSelector(".card:nth-child(3) .form-group:nth-child(2) .ngb-tp-input-container:nth-child(3) > .ngb-tp-input:nth-child(2)")).Click();
    driver.FindElement(By.CssSelector(".card:nth-child(3) .form-group:nth-child(2) .ngb-tp-input-container:nth-child(3) > .ngb-tp-input:nth-child(2)")).SendKeys("\n");
    driver.FindElement(By.CssSelector(".ng-touched > fieldset .ngb-tp-minute > .ngb-tp-input")).SendKeys("05");
    driver.FindElement(By.CssSelector(".btn:nth-child(4)")).Click();
    driver.FindElement(By.CssSelector(".card-footer > .btn-primary")).Click();
}

```

Figura 49: Script editar parte diario.

Fuente: Elaboración propia.

The screenshot shows the Selenium IDE interface during a test run. The top bar indicates the test is 'Executing' and the URL is 'https://localhost:44319'. The test name is 'editar parte diario'. Below this, a table lists the commands being executed:

Step	Command	Target	Value
14	type	css=.ng-touched > fieldset .ngb-tp-minute > .ngb-tp-input	05
15	click	css=.btn:nth-child(4)	
16	click	css=.card-footer > .btn-primary	

Below the table, the 'Command' field is set to 'click', the 'Target' is 'css=.card-footer > .btn-primary', and the 'Value' is empty. The 'Description' field is also empty. At the bottom left, it shows 'Runs: 1 Failures: 0'. The log at the bottom shows the following entries:

Log	Reference	Time
13. type on css=.card:nth-child(3) .form-group:nth-child(2) .ngb-tp-input-container:nth-child(3) > .ngb-tp-input:nth-child(2) OK		15:32:01
14. type on css=.ng-touched > fieldset .ngb-tp-minute > .ngb-tp-input with value 05 OK		15:32:02
15. click on css=.btn:nth-child(4) OK		15:32:03
16. click on css=.card-footer > .btn-primary OK		15:32:04
'editar parte diario' completed successfully		15:32:05

Figura 50: Resultado de editar parte diario en Selenium.

Fuente: Elaboración propia.

```

public void Listaroperario() {
    driver.Navigate().GoToUrl("https://localhost:44319/");
    driver.Manage().Window.Size = new System.Drawing.Size(1382, 744);
    driver.FindElement(By.LinkText("Operario")).Click();
    {
        var element = driver.FindElement(By.LinkText("Operario"));
        Actions builder = new Actions(driver);
        builder.MoveToElement(element).Perform();
    }
    {
        var element = driver.FindElement(By.TagName("body"));
        Actions builder = new Actions(driver);
        builder.MoveToElement(element, 0, 0).Perform();
    }
    driver.FindElement(By.Name("filterOper")).Click();
    driver.FindElement(By.Name("filterOper")).SendKeys("lui");
}
}

```

Figura 51: Script lista operarios.

Fuente: Elaboración propia.

3	click	linkText=Operario	
4	mouse over	linkText=Operario	
5	mouse out	linkText=Operario	
6	click	name=filterOper	
7	type	name=filterOper	lui

Command	<input type="text"/>	<input type="button" value="//"/>	<input type="button" value="↗"/>
Target	<input type="text"/>	<input type="button" value="📍"/>	<input type="button" value="🔍"/>
Value	<input type="text"/>		
Description	<input type="text"/>		

Log	Reference	
6. click on name=filterOper	OK	16:20:23
7. type on name=filterOper with value lui	OK	16:20:24
'listar operario' completed successfully		16:20:25

Figura 52: Resultado de lista de operarios en Selenium.

Fuente: Elaboración propia.

```

{
    var element = driver.FindElement(By.TagName("body"));
    Actions builder = new Actions(driver);
    builder.MoveToElement(element, 0, 0).Perform();
}
driver.FindElement(By.CssSelector(".form-group:nth-child(1) .form-control")).Click();
driver.FindElement(By.CssSelector(".form-group:nth-child(1) .form-control")).SendKeys("Luis Miguel");
driver.FindElement(By.CssSelector(".form-group:nth-child(2) .form-control")).SendKeys("Feria");
driver.FindElement(By.CssSelector(".form-group:nth-child(3) .form-control")).SendKeys("Lazaro");
driver.FindElement(By.CssSelector(".form-group:nth-child(4) .form-control")).Click();
driver.FindElement(By.CssSelector(".form-group:nth-child(4) .form-control")).SendKeys("75468542");
driver.FindElement(By.CssSelector(".btn-outline-secondary")).Click();
{
    var element = driver.FindElement(By.CssSelector(".btn-outline-secondary"));
    Actions builder = new Actions(driver);
    builder.MoveToElement(element).Perform();
}
{
    var element = driver.FindElement(By.TagName("body"));
    Actions builder = new Actions(driver);
    builder.MoveToElement(element, 0, 0).Perform();
}
driver.FindElement(By.CssSelector(".custom-select:nth-child(2)")).Click();
{
    var dropdown = driver.FindElement(By.CssSelector(".custom-select:nth-child(2)"));
    dropdown.FindElement(By.XPath("//option[. = '2011']")).Click();
}
driver.FindElement(By.CssSelector(".custom-select:nth-child(2)")).Click();
driver.FindElement(By.CssSelector(".custom-select:nth-child(2)")).Click();
{
    var dropdown = driver.FindElement(By.CssSelector(".custom-select:nth-child(2)"));
}

```

Figura 53: Script nuevo operario.

Fuente: Elaboración propia.

The screenshot shows the Selenium IDE interface. On the left, there is a progress bar and a summary: "Runs: 1 Failures: 0". On the right, there are input fields for "Target", "Value", and "Description". Below this, a table displays the test log with two columns: "Log" and "Reference". The log shows three entries: a successful type action at 15:48:49, a successful click action at 15:48:50, and a final status message "'Nuevo operario' completed successfully" at 15:48:51.

Log	Reference
30. type on css=.ng-untouched:nth-child(1) with value 976845850 OK	15:48:49
31. click on css=.card-footer > .btn-primary OK	15:48:50
'Nuevo operario' completed successfully	15:48:51

Figura 54: Resultado de nuevo operario en Selenium.

Fuente: Elaboración propia.

```

public void EditarMquina() {
    driver.Navigate().GoToUrl("https://localhost:44319/");
    driver.Manage().Window.Size = new System.Drawing.Size(1382, 744);
    driver.FindElement(By.LinkText("Máquina")).Click();
    {
        var element = driver.FindElement(By.LinkText("Máquina"));
        Actions builder = new Actions(driver);
        builder.MoveToElement(element).Perform();
    }
    {
        var element = driver.FindElement(By.TagName("body"));
        Actions builder = new Actions(driver);
        builder.MoveToElement(element, 0, 0).Perform();
    }
    driver.FindElement(By.CssSelector("tr:nth-child(1) .btn-primary")).Click();
    driver.FindElement(By.CssSelector(".form-group:nth-child(2)")).Click();
    driver.FindElement(By.CssSelector(".form-group:nth-child(2)")).SendKeys("\n");
    driver.FindElement(By.CssSelector(".form-group:nth-child(2)")).SendKeys("Máquina de carga contrapesada en la parte trasera");
    driver.FindElement(By.CssSelector(".col-sm-10:nth-child(6) > input")).Click();
    driver.FindElement(By.CssSelector(".card-footer > .btn-primary")).Click();
}

```

Figura 55: Script editar operario.

Fuente: Elaboración propia.

8	type	css=.form-group:nth-child(2)	
9	type	css=.form-group:nth-child(2)	Máquina de carga contrapesada en la parte trasera
10	click	css=.col-sm-10:nth-child(6) > input	
11	click	css=.card-footer > .btn-primary	

Command	<input type="text"/>	<input type="button" value="//"/>	<input type="button" value="📄"/>
Target	<input type="text"/>	<input type="button" value="📍"/>	<input type="button" value="🔍"/>
Value	<input type="text"/>		
Description	<input type="text"/>		

Log	Reference	
10. click on css=.col-sm-10:nth-child(6) > input	OK	18:20:03
11. click on css=.card-footer > .btn-primary	OK	18:20:04
'Editar Máquina' completed successfully		18:20:04

Figura 56: Resultado editar operario en Selenium.

Fuente: Elaboración propia.

```

0 referencias
public void Listarmquina() {
    driver.Navigate().GoToUrl("https://localhost:44319/");
    driver.Manage().Window.Size = new System.Drawing.Size(1382, 744);
    driver.FindElement(By.LinkText("Máquina")).Click();
    driver.FindElement(By.CssSelector("li:nth-child(8) span:nth-child(2)")).Click();
    driver.FindElement(By.Name("search")).Click();
    driver.FindElement(By.Name("search")).SendKeys("vol");
}

```

Figura 57: Script listar máquina.

Fuente: Elaboración propia.

The screenshot shows the Selenium IDE interface. At the top, the URL is set to `https://localhost:44319`. Below this is a table of commands and their results:

	Command	Target	Value
1	<code>open</code>	<code>/</code>	
2	<code>set window size</code>	<code>1382x744</code>	
3	<code>click</code>	<code>linkText=Máquina</code>	
4	<code>click</code>	<code>name=search</code>	
5	<code>type</code>	<code>name=search</code>	<code>com</code>

Below the table, there are input fields for Command, Target, Value, and Description. At the bottom, a log shows the execution results:

Log	Reference	
1. open on / OK		18:23:31
2. setWindowSize on 1382x744 OK		18:23:31
3. click on linkText=Máquina OK		18:23:31
4. click on name=search OK		18:23:38
5. type on name=search with value com OK		18:23:39
'listar máquina' completed successfully		18:23:40

Figura 58: Resultado de listar máquina en Selenium.

Fuente: Elaboración propia.

```

public void NuevaMaquina() {
    driver.Navigate().GoToUrl("https://localhost:44319/");
    driver.Manage().Window.Size = new System.Drawing.Size(1382, 744);
    driver.FindElement(By.LinkText("Máquina")).Click();
    driver.FindElement(By.CssSelector(".btn")).Click();
    driver.FindElement(By.CssSelector(".form-group:nth-child(1) .form-control")).Click();
    driver.FindElement(By.CssSelector(".form-group:nth-child(1) .form-control")).SendKeys("Montacarga");
    driver.FindElement(By.CssSelector(".form-group:nth-child(2) .form-control")).SendKeys("Máquina de carga ");
    driver.FindElement(By.CssSelector(".form-group:nth-child(3) .form-control")).Click();
    driver.FindElement(By.CssSelector(".form-group:nth-child(3) .form-control")).SendKeys("cat");
    driver.FindElement(By.CssSelector(".form-group:nth-child(4) .form-control")).Click();
    driver.FindElement(By.CssSelector(".form-group:nth-child(4) .form-control")).SendKeys("TYG76");
    driver.FindElement(By.CssSelector(".ng-untouched:nth-child(2) > input")).Click();
    driver.FindElement(By.CssSelector(".col-sm-10:nth-child(4) > input")).Click();
    driver.FindElement(By.CssSelector(".card-footer > .btn-primary")).Click();
}

```

Figura 59: Script nueva máquina.

Fuente: Elaboración propia.

	Command	Target	Value
9	type	css=.form-group:nth-child(3) .form-control	cat
10	click	css=.form-group:nth-child(4) .form-control	
11	type	css=.form-group:nth-child(4) .form-control	TYG76
12	click	css=.ng-untouched:nth-child(2) > input	
13	click	css=.col-sm-10:nth-child(4) > input	

Command //

Target

Value

Description

Runs: 1 Failures: 0

Log	Reference	
13. click on css=.col-sm-10:nth-child(4) > input	OK	18:09:01
14. click on css=.card-footer > .btn-primary	OK	18:09:02
'nueva máquina' completed successfully		18:09:03

Figura 60: Resultado de nueva máquina.

Fuente: Elaboración propia.

```

public void EditarMquina() {
    driver.Navigate().GoToUrl("https://localhost:44319/");
    driver.Manage().Window.Size = new System.Drawing.Size(1382, 744);
    driver.FindElement(By.LinkText("Máquina")).Click();
    {
        var element = driver.FindElement(By.LinkText("Máquina"));
        Actions builder = new Actions(driver);
        builder.MoveToElement(element).Perform();
    }
    {
        var element = driver.FindElement(By.TagName("body"));
        Actions builder = new Actions(driver);
        builder.MoveToElement(element, 0, 0).Perform();
    }
    driver.FindElement(By.CssSelector("tr:nth-child(1) .btn-primary")).Click();
    driver.FindElement(By.CssSelector(".form-group:nth-child(2)")).Click();
    driver.FindElement(By.CssSelector(".form-group:nth-child(2)")).SendKeys("\n");
    driver.FindElement(By.CssSelector(".form-group:nth-child(2)")).SendKeys("Máquina de carga contrapesada en la parte trasera");
    driver.FindElement(By.CssSelector(".col-sm-10:nth-child(6) > input")).Click();
    driver.FindElement(By.CssSelector(".card-footer > .btn-primary")).Click();
}

```

Figura 61: Script editar máquina.

Fuente: Elaboración propia.

9	type	css= form-group:nth-child(2)	Máquina de carga contrapesada en la parte trasera
10	click	css= .col-sm-10:nth-child(6) > input	
11	click	css= .card-footer > .btn-primary	

Command	<input type="text"/>	<input type="button" value="//"/>	<input type="button" value="↗"/>
Target	<input type="text"/>	<input type="button" value="↖"/>	<input type="button" value="🔍"/>
Value	<input type="text"/>		
Description	<input type="text"/>		

Log	Reference	
10. click on css=.col-sm-10:nth-child(6) > input OK		18:20:03
11. click on css=.card-footer > .btn-primary OK		18:20:04
'Editar Máquina' completed successfully		18:20:04

Figura 62: Resultado de editar máquina en Selenium.

Fuente: Elaboración propia.

3.4.2.4.4. Pruebas de concurrencia con Jmeter

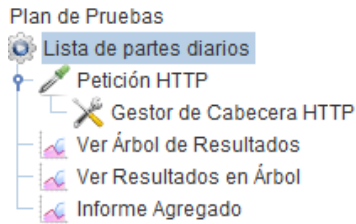
Jmeter es una herramienta Open Source que usamos para evaluar el rendimiento del sistema web y así conocer el tiempo de respuesta del sitio. Es necesario calcular valores de tiempos a la concurrencia de varios usuarios y varias peticiones a la vez. Definimos las peticiones http en función al registro de Partes Diarios.

Tabla 53.

Características de un Parte Diario.

Características de un Parte Diario	
Datos Generales	Detalle de parte diario
Obra: Mantenimiento de la carretera Coporaque km 1122.	Detalle de trabajo 1
Máquina: Oruga pala cargadora	Descripción 1: Recorrido máquina
Fecha: 2021-02-15	Hora inicio: 08:10:00
Hora inicio de trabajo: 8:00:00	Hora fin: 10:00:00
Hora fin de trabajo: 11:00:00	Observación: finalizada
Observación: Ninguna	Descripción 2: Transporte de montículo
Total: 03:00:00 horas	Hora inicio: 10:00:00
	Hora fin: 11:00:00
	Observación: finalizada

En el Grupo de Hilos de la figura 63 se configuró con 70 usuarios concurrentes que realizaron la petición de listar los partes diarios, que vienen siendo más de 200 registros. También se muestra el periodo de subida que viene siendo 1 segundo y el número de bucles que es 1.



Grupo de Hilos

Nombre:

Comentarios:

Acción a tomar después de un error de Muestreador

Continuar
 Comenzar siguiente iteración
 Parar Hilo

Propiedades de Hilo

Número de Hilos:

Periodo de Subida (en segundos):

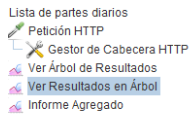
Contador del bucle: Sin fin

Same user on each iteration
 Retrasar la creación de Hilos hasta que se necesiten
 Planificador

Figura 63: Grupo de Hilos de lista de partes diarios.

Fuente: Elaboración propia.

La figura 64 muestra el resultado de cada petición para listar partes diarios que realizaron los 70 usuarios, el estado muestra que cada petición fue exitosa, el máximo tiempo de petición aproximadamente es de 958 milisegundos y la desviación no supera a la media, lo que indica que cada petición no tuvo ningún retraso.



Muestra #	Tiempo de co...	Nombre del hilo	Etiqueta	Tiempo de Mues...	Estado	Bytes	Sent Bytes	Latency	Connect Time[...
44	16.26.23.110	Lista de partes...	Petición HTTP	1100	✓	116281	167	610	40
45	16.26.23.059	Lista de partes...	Petición HTTP	1221	✓	116281	167	549	53
46	16.26.23.047	Lista de partes...	Petición HTTP	1233	✓	116281	167	628	63
47	16.26.23.156	Lista de partes...	Petición HTTP	1126	✓	116281	167	972	46
48	16.26.23.128	Lista de partes...	Petición HTTP	1152	✓	116281	167	550	36
49	16.26.23.142	Lista de partes...	Petición HTTP	1138	✓	116281	167	984	49
50	16.26.23.698	Lista de partes...	Petición HTTP	587	✓	116281	167	556	27
51	16.26.23.726	Lista de partes...	Petición HTTP	560	✓	116281	167	521	24
52	16.26.23.437	Lista de partes...	Petición HTTP	849	✓	116281	167	797	47
53	16.26.23.074	Lista de partes...	Petición HTTP	1216	✓	116281	167	430	32
54	16.26.23.738	Lista de partes...	Petición HTTP	555	✓	116281	167	507	19
55	16.26.23.746	Lista de partes...	Petición HTTP	547	✓	116281	167	503	18
56	16.26.23.639	Lista de partes...	Petición HTTP	654	✓	116281	167	601	34
57	16.26.23.619	Lista de partes...	Petición HTTP	674	✓	116281	167	626	43
58	16.26.23.577	Lista de partes...	Petición HTTP	717	✓	116281	167	685	57
59	16.26.23.521	Lista de partes...	Petición HTTP	775	✓	116281	167	726	31
60	16.26.23.584	Lista de partes...	Petición HTTP	712	✓	116281	167	656	50
61	16.26.23.654	Lista de partes...	Petición HTTP	645	✓	116281	167	596	27
62	16.26.23.598	Lista de partes...	Petición HTTP	702	✓	116281	167	649	62
63	16.26.23.510	Lista de partes...	Petición HTTP	791	✓	116281	167	732	32
64	16.26.23.623	Lista de partes...	Petición HTTP	678	✓	116281	167	622	38
65	16.26.23.670	Lista de partes...	Petición HTTP	633	✓	116281	167	579	38
66	16.26.23.556	Lista de partes...	Petición HTTP	748	✓	116281	167	687	61
67	16.26.23.548	Lista de partes...	Petición HTTP	760	✓	116281	167	699	41
68	16.26.23.494	Lista de partes...	Petición HTTP	815	✓	116281	167	744	23
69	16.26.23.685	Lista de partes...	Petición HTTP	634	✓	116281	167	627	37
70	16.26.23.704	Lista de partes...	Petición HTTP	615	✓	116281	167	610	22

Scroll automatically? Child samples? **No. de Muestras 70** **Última Muestra 615** **Media 968** **Desviación 247**

Figura 64: Resultados de peticiones de lista de partes diarios.

Fuente: Elaboración propia.

El informe agregado es muy importante para los resultados de las peticiones porque nos muestra el resumen de todas las peticiones como vemos en la figura 65, lo que necesitamos saber es la media que es 958 ms, el máximo tiempo que es de 1415 ms y nuestro rendimiento de todas las peticiones son 45.6 segundos.

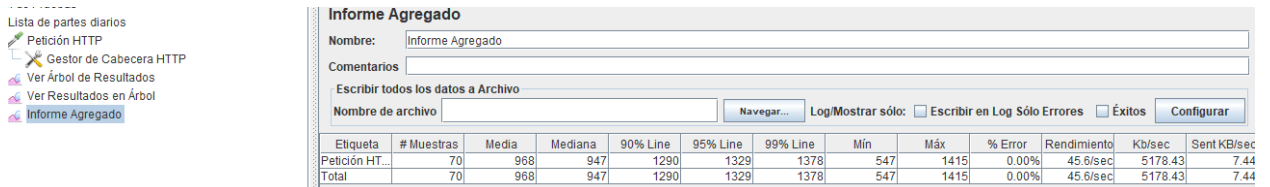


Figura 65: Informe Agregado de lista de partes diarios.

Fuente: Elaboración propia.

La figura 66 se muestra el resultado de cada petición para un nuevo parte diario que realizaron los 70 usuarios, el estado muestra que cada petición fue exitosa, el máximo tiempo de petición aproximadamente es de 47 ms y la desviación no supera a la media, lo que indica que cada petición no tuvo ningún retraso.

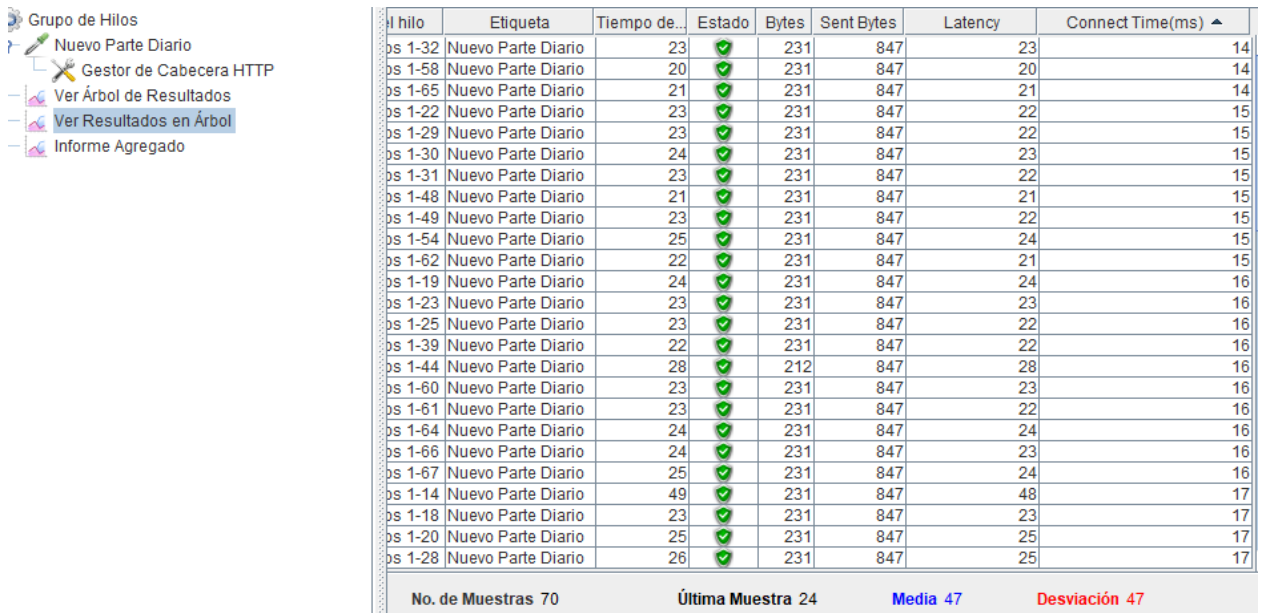


Figura 66: Resultados de peticiones de nuevos partes diarios.

Fuente: Elaboración propia.

El informe agregado muestra el resumen de todas las peticiones como vemos en la figura 67, lo que necesitamos saber es la media que es 47 ms, el máximo tiempo que es de 217 ms y nuestro rendimiento de todas las peticiones son 71.2 segundos.

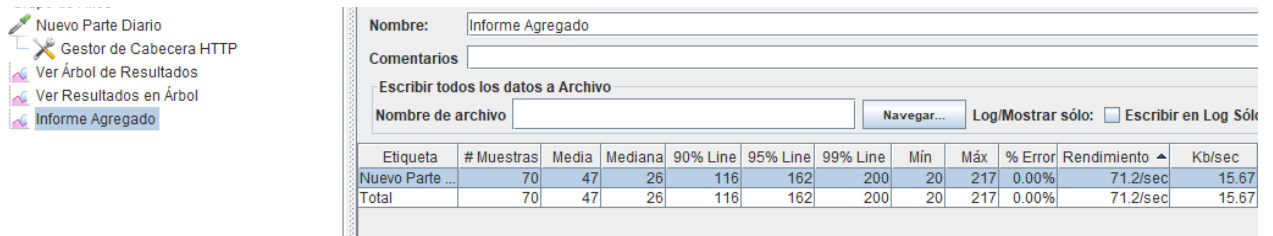


Figura 67: Informe Agregado de nuevos partes diarios.

Fuente: Elaboración propia.

En el Grupo de Hilos de la figura 68 se configuró con 10 usuarios concurrentes que realizaron la petición de listar, crear y editar operarios, también se muestra el periodo de subida que viene siendo 1 segundo y el número de bucles que es 1.

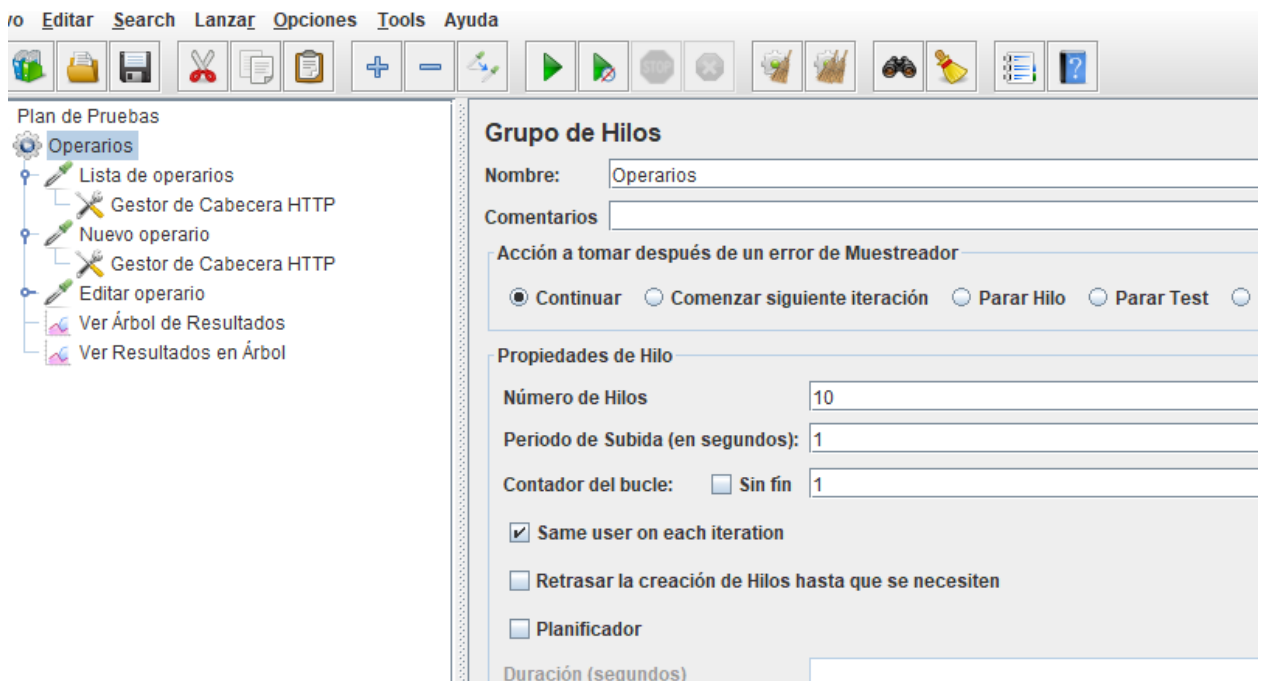


Figura 68: Grupo de Hilos del operario.

Fuente: Elaboración propia.

La figura 69 se muestra el resultado de cada petición para el operario que realizaron los 10 usuarios, el estado muestra que cada petición fue exitosa, el máximo tiempo de petición aproximadamente es de 36 ms.

de c...	Nombre del...	Etiqueta	Tiempo de M...	Estado	Bytes	Sent Bytes	Latency	Connect Tim...
32.880	Operarios 1-2	Editar operario	8	✓	231	362	7	0
32.916	Operarios 1-3	Lista de oper...	28	✓	7789	165	28	22
32.945	Operarios 1-3	Nuevo opera...	10	✓	212	353	10	0
32.955	Operarios 1-3	Editar operario	8	✓	231	362	8	0
33.012	Operarios 1-4	Lista de oper...	33	✓	7956	165	33	22
33.045	Operarios 1-4	Nuevo opera...	13	✓	231	353	13	0
33.059	Operarios 1-4	Editar operario	9	✓	231	362	8	0
33.113	Operarios 1-5	Lista de oper...	23	✓	8104	165	22	19
33.136	Operarios 1-5	Nuevo opera...	8	✓	231	353	8	0
33.145	Operarios 1-5	Editar operario	6	✓	212	362	6	0
33.214	Operarios 1-6	Lista de oper...	22	✓	8290	165	22	17
33.236	Operarios 1-6	Nuevo opera...	10	✓	231	353	9	0
33.246	Operarios 1-6	Editar operario	16	✓	231	362	15	0
33.314	Operarios 1-7	Lista de oper...	24	✓	8457	165	23	17
33.338	Operarios 1-7	Nuevo opera...	7	✓	231	353	7	0
33.346	Operarios 1-7	Editar operario	6	✓	231	362	6	0
33.413	Operarios 1-8	Lista de oper...	23	✓	8605	165	23	19
33.436	Operarios 1-8	Nuevo opera...	11	✓	231	353	11	0
33.447	Operarios 1-8	Editar operario	6	✓	212	362	6	0
33.513	Operarios 1-9	Lista de oper...	23	✓	8791	165	22	18

Figura 69: Resultado de peticiones del operario.

Fuente: Elaboración propia.

El informe agregado muestra el resumen de todas las peticiones como vemos en la figura 70, lo que necesitamos saber es la media que es 36 ms para listar y 15 ms para editar y crear; el máximo tiempo que es de 156 ms para listar, 85 ms para crear y 39 ms para editar; nuestro rendimiento 10.8 s para listar, 12.8 para crear y 14.1 para editar.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
Lista de op...	10	50	36	64	64	156	28	156	0.00%	10.8/sec	103.85	1.74
Nuevo oper...	10	28	15	84	84	85	8	85	0.00%	12.8/sec	2.86	4.40
Editar oper...	10	19	15	38	38	39	9	39	0.00%	14.1/sec	3.18	4.99
Total	30	32	20	64	84	156	8	156	0.00%	31.5/sec	105.73	9.02

Figura 70: Informe Agregado del operario.

Fuente: Elaboración propia.

En el Grupo de Hilos de la figura 69 se configuró con 10 usuarios concurrentes que realizaron la petición de listar, crear y editar máquinas, también se muestra el periodo de subida que viene siendo 1 segundo y el número de bucles que es 1.

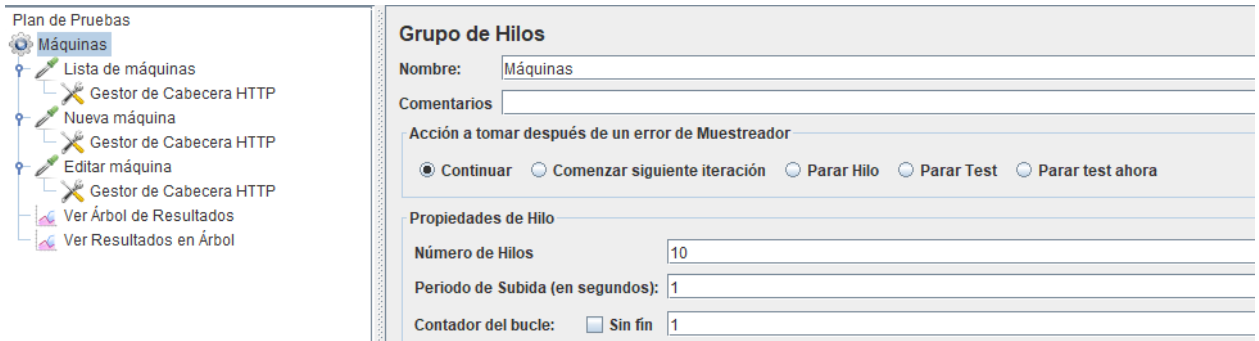


Figura 71: Grupo de Hilos de la máquina.

Fuente: Elaboración propia.

La figura 66 se muestra el resultado de cada petición para un nuevo parte diario que realizaron los 70 usuarios, el estado muestra que cada petición fue exitosa, el máximo tiempo de petición aproximadamente es de 47 ms y la desviación no supera a la media, lo que indica que cada petición no tuvo ningún retraso.

Nombre del...	Etiqueta	Tiempo de M...	Estado	Bytes	Sent Bytes	Latency	Connect Tim...
17.482 Máquinas 1-2	Editar máqui...	38	✓	212	488	38	0
17.544 Máquinas 1-3	Lista de máq...	23	✓	4354	164	23	18
17.568 Máquinas 1-3	Nueva máqui...	9	✓	231	488	8	0
17.577 Máquinas 1-3	Editar máqui...	37	✓	212	488	37	0
17.645 Máquinas 1-4	Lista de máq...	23	✓	4529	164	23	19
17.669 Máquinas 1-4	Nueva máqui...	6	✓	212	488	6	0
17.676 Máquinas 1-4	Editar máqui...	36	✓	212	488	36	0
17.744 Máquinas 1-5	Lista de máq...	20	✓	4723	164	20	16
17.765 Máquinas 1-5	Nueva máqui...	6	✓	212	488	6	0
17.772 Máquinas 1-5	Editar máqui...	36	✓	231	488	36	0
17.844 Máquinas 1-6	Lista de máq...	25	✓	4917	164	25	21
17.869 Máquinas 1-6	Nueva máqui...	11	✓	231	488	10	0
17.880 Máquinas 1-6	Editar máqui...	38	✓	231	488	38	0
17.945 Máquinas 1-7	Lista de máq...	17	✓	5111	164	17	14
17.963 Máquinas 1-7	Nueva máqui...	9	✓	212	488	9	0
17.973 Máquinas 1-7	Editar máqui...	39	✓	212	488	39	0
18.044 Máquinas 1-8	Lista de máq...	22	✓	5305	164	22	18
18.066 Máquinas 1-8	Nueva máqui...	7	✓	212	488	7	0
18.073 Máquinas 1-8	Editar máqui...	34	✓	212	488	34	0
18.144 Máquinas 1-9	Lista de máq...	23	✓	5518	164	23	18

Figura 72: Resultados de peticiones de la máquina.

Fuente: Elaboración propia.

El informe agregado muestra el resumen de todas las peticiones como vemos en la figura 73, lo que necesitamos saber es la media que es 83 ms para listar, 17 ms para crear y 147 para editar; el máximo tiempo que es de 279 ms para listar, 130 ms para

crear y 343 ms para editar; nuestro rendimiento 10.6 s para listar,10.9 para crear y 11.7 para editar.

Informe Agregado

Nombre: Informe Agregado

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
ista de m...	10	110	83	188	188	279	33	279	0.00%	10.6/sec	68.99	1.69
lueva máq...	10	41	17	102	102	130	9	130	0.00%	10.9/sec	2.46	5.20
ditar máq...	10	187	147	341	341	343	63	343	0.00%	11.7/sec	2.49	5.59
otal	30	112	83	279	324	343	9	343	0.00%	27.6/sec	64.13	10.26

Figura 73: Informe Agregado de la máquina.

Fuente: Elaboración propia.

CAPÍTULO IV. Resultados

Los resultados de la investigación del “Desarrollo de un sistema web utilizando Angular y Entity Framework asp.net Core para la gestión de horas máquina de trabajo en E&R Contratistas Generales SRL.” se describió según sus objetivos específicos.

4.1. Especificar los requerimientos para el sistema web de horas máquina en la empresa.

En la especificación de requerimientos para el sistema web de horas máquina se ha recopilado la información mediante reuniones con el usuario que realiza los registros de partes diarios en el área administrativa; los partes diarios comprenden las horas máquina de trabajo de los operarios. Así obtuvimos los requisitos del sistema, estimando fechas para el proceso de las historias de usuario.

Después de las reuniones con el usuario del área administrativa tenemos como resultado la identificación de requerimientos dividiéndolos en tres módulos principales; operarios, máquinas y partes diarios; también se definieron cada historia del usuario dando prioridad con los módulos correspondientes y finalmente se realizó un bosquejos y prototipos para el sistema web para la gestión de horas máquina.

4.2. Desarrollar el sistema web para la gestión de horas máquina.

Después de los resultados al tener definido los requerimientos del sistema para la gestión de horas máquina empezamos con la codificación, que significa convertir los requerimientos, historias de usuario y prototipos en un sistema web funcional. El desarrollo de esta aplicación web se realizó con el entorno de desarrollo Visual Studio 2019, se usó los frameworks de ASP .Net Core, Entity Framework y Angular. Primeramente, realizamos el modelo de base de datos, seguidamente mapeamos las tablas con Entity Framework, al tener los modelos en el proyecto se construyó el backend

y se realizó el código web Api. Finalmente se hizo la conexión Http con angular, módulos y componentes, desarrollando el código fuente en cada módulo para recibir y enviar los datos.

Los resultados que obtuvimos en el desarrollo del sistema web, son los módulos de operario, máquina, parte diario, obra y consumo, donde cada módulo funciona correctamente cumpliendo los requerimientos del proyecto de investigación.

4.3. Desplegar y realizar pruebas de funcionalidad del sistema web de horas máquina.

Antes de hacer las pruebas que dan la aceptación realizamos las pruebas unitarias en la parte del backend, cada prueba correspondiente en cada módulo según sus historias de usuario, para lo cual usamos la herramienta llamada Postman. Finalmente realizamos la prueba de aceptación en Selenium y pruebas de concurrencia en Jmeter; que se crearon conforme a las historias de usuario; el usuario identificó un escenario para confirmar el correcto funcionamiento de los módulos del sistema.

Al pasar por las pruebas de aceptación se realizó pruebas de usabilidad en un escenario real anexo 2, en la obra, mejoramiento de los servicios en la institución educativa Horacio Zeballos Gamez de Juliaca Provincia de San Roman Puno. Registrando operarios, máquinas y partes diarios como se muestra en el anexo 3.

Los resultados de las pruebas de usabilidad fueron satisfactorios, como muestran los resultados de las pruebas de concurrencia en Jmeter, donde se visualiza los tiempos de cada muestra, valor de la media, rendimiento y la desviación, en la figura 65 se muestra la lista de partes diarios donde su media es 968ms, mín 547ms, máx 1416ms y rendimiento 45,6sec; nuevo parte diario su media 47ms, mín 20ms, máx 217ms y rendimiento 71,2sec; el tiempo que empleó los usuarios para las peticiones fue rápido y sencillo, por los formularios dinámicos y los campos autocomplete.

CAPÍTULO V. Conclusiones y Recomendaciones

5.1 Conclusiones

Después de los resultados concluimos que se desarrolló de un sistema web utilizando Angular y Entity Framework asp.net Core para la gestión de horas máquina de trabajo en E&R Contratistas Generales SRL., todo el proyecto se realizó con la metodología XP donde en sus fases se identificó los requerimientos, desarrollo del sistema y las pruebas de usuario.

Se identificó 5 módulos principales, 9 historias de usuario y se dividió en 3 iteraciones, todo en conjunto con el usuario del área de administración. Se desarrolló prototipos para el entendimiento del usuario y dar su aceptación o corrección en los requerimientos e historias de usuario.

Se desarrolló el sistema con los frameworks ASP.Net Core, Entity Framework y Angular. Los requerimientos funcionales realizados son la gestión de los módulos operario, máquina, parte diario, obra y consumo; el sistema permite el registro, edición y visualización de los módulos mencionados y un reporte general del uso de las máquinas que se muestra en el anexo 3.

Se realizó las pruebas de aceptación con Selenium y concurrencia en Jmeter, las pruebas de usabilidad se dieron en un escenario de trabajo real que se presenta en el anexo 2. Los resultados fueron satisfactorios al registrar toda la información del escenario propuesto se visualiza en el anexo 3; el usuario empleó menos tiempo de lo acostumbrado al hacer los registros en Excel y su funcionalidad es amigable sencilla.

5.2. Recomendaciones

En la parte de la programación se recomienda una aplicación en Xamari, para equipos móviles y así optimizar la rapidez de registros horas de trabajo máquinas, lo cual no se pudo realizar por presupuesto y capacitaciones de usuarios.

REFERENCIAS

- Aden, D. (2018). *Angular 2 Components: Inputs and Outputs*.
<https://www.sitepoint.com/angular-2-components-inputs-outputs/>
- Brito, G., & Mayne, J. (2008). *Pontificia Universidad Católica de Valparaíso Facultad de ingeniería*. 0–54.
- Bulavko, A. (2017). *Extreme Programming (XP) Article*. April.
- Burdick, D., Fagin, R., Kolaitis, P. G., Popa, L., & Tan, W. C. (2019). Expressive power of entity-linking frameworks. *Journal of Computer and System Sciences*, 100, 44–69.
<https://doi.org/10.1016/j.jcss.2018.09.001>
- Cheng, C. F., Rashidi, A., Davenport, M. A., & Anderson, D. V. (2017). Activity analysis of construction equipment using audio signals and support vector machines. *Automation in Construction*, 81, 240–253.
<https://doi.org/10.1016/j.autcon.2017.06.005>
- Chiaretta, S. (2018). .Net Core - Modular. In *Front-end Development with ASP.NET Core, Angular, and Bootstrap* (p. <https://books.google.com.pe/books?hl=es&lr=&id=W7Y>).
- Correa, C., & Wilson, B. (2014). MAESTRÍA EN ADMINISTRACIÓN DE NEGOCIOS; SISTEMA DE EVALUACIÓN; GOBIERNO AUTÓNOMO DESCENTRALIZADO; CANTÓN SUSCAL; CAÑAR. *Sereal Untuk*, 51(1), 51.
- Danielecki, D. (2019). *Security First approach in development of Single-Page Application based on Angular*.
- Dehai, C. (2019). Development and Research of Equipment Information Management System of Thermal Power Plant. *DEStech Transactions on Engineering and Technology Research*, eeec. <https://doi.org/10.12783/dtetr/eeec2018/26885>
- Devica, S. (2016). Become a ninja with Angular. In *Pengaruh Harga Diskon Dan Persepsi Produk Terhadap Nilai Belanja Serta Perilaku Pembelian Konsumen* (Vol. 7, Issue 9).
- Dykstra, T., & Anderson, R. (2014). *Getting Started with Entity Framework 6 Database First using MVC 5 | Microsoft Docs*. <https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/database-first-development/setting-up->

database

- Enrique, J., & Fernández, E. (2015). *IMPLEMENTACIÓN DE UN SISTEMA WEB PARA EL CONTROL DE PAROS DE LAS MÁQUINAS TEXTILES EN LA FÁBRICA DE TEJIDOS SAN CARLOS S.A.C PRESENTADA*.
- Eroshkin, S. Y., Kameneva, N. A., Kovkov, D. V., & Sukhorukov, A. I. (2017). Conceptual System in the Modern Information Management. *Procedia Computer Science*, 103, 609–612. <https://doi.org/10.1016/j.procs.2017.01.079>
- Esposito, D. (2018). The Web Api Framework. In *Programing ASP.NET Core* (p. <https://books.google.com.pe/books?hl=es&lr=&id=5Ld>).
- García, G. V. (2015). *PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ FACULTAD DE CIENCIAS E INGENIERÍA Implementación de un sistema de gestión de la investigación para la especialidad de Ingeniería Informática de la PUCP*.
- Gubalova, J., & Medvedova, P. (2018). User-defined financial functions for MS SQL server. *International Journal of Advanced Computer Science and Applications*, 9(9), 19–25. <https://doi.org/10.14569/ijacsa.2018.090903>
- Hanna, S. (2016). An approach to modeling web services datatype descriptions. *Journal of Theoretical and Applied Electronic Commerce Research*, 11(2), 64–82. <https://doi.org/10.4067/S0718-18762016000200006>
- Korva, J. (2016). *Developing a web application with Angular 2*. https://www.theseus.fi/bitstream/handle/10024/121905/Korva_Jukka.pdf?sequence=1
- Leo, M. (2020). *A Beginner's Guide to Angular Architecture for Front-End Development*. https://www.academia.edu/37389712/A_Beginners_Guide_to_Angular_Architecture_for_Front_End_Development
- Mueller, J. (2013). Microsoft ADO.NET Entity Framework. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 9).
- Mukherjee, S. (2019). Popular SQL Server Database Encryption Choices. *International Journal of Computer Trends and Technology*, 66(1), 14–19. <https://doi.org/10.14445/22312803/ijctt-v66p103>

- Ortiz, C. P. (2015). *Análisis, Diseño e Implementación de una Guía Gastronómica para la administración y ubicación de restaurantes en entorno web*. 105.
- Otoya Florián, C. A. (2015). Implementación de una aplicación móvil para el monitoreo de contenido y disponibilidad de servicios Web, servicio FTP y páginas Web. *Pontificia Universidad Católica Del Perú*.
<http://tesis.pucp.edu.pe/repositorio/handle/123456789/6043>
- Quelal, D. (2019). “*Desarrollo Del Sistema Académico Aplicando La Herramienta Asp.Net Core 2 Para La Escuela Eufrosia Pelletier.*”
[http://repositorio.utn.edu.ec/bitstream/123456789/9027/1/04 IS 509 TRABAJO DE GRADO.pdf](http://repositorio.utn.edu.ec/bitstream/123456789/9027/1/04_ISC_509_TRABAJO_DE_GRADO.pdf)
- Roldan Ortiz, M. (2018). *Desarrollo de un sistema de información para sistematizar la recolección y clasificación de información de las máquinas y las prensas del proceso de Producción de la empresa Pastas Comarrico SAS*.
- Roque Rojas, V. D. (2018). *SISTEMA DE INFORMACIÓN PARA EL CONTROL DE PRODUCCIÓN DE MÁQUINAS ACEPTADORAS DE MONEDAS EN LA EMPRESA KTP E.I.R.L. – LIMA*.
[https://repositorio.utelesup.edu.pe/bitstream/UTELESUP/966/1/ROJAS ROQUE VICTOR DARWIN.pdf](https://repositorio.utelesup.edu.pe/bitstream/UTELESUP/966/1/ROJAS_ROQUE_VICTOR_DARWIN.pdf)
- Ross, J. C., Joshi, A., & Bhattacharyya, P. (2016). A Framework that Uses the Web for Named Entity Class Identification: Case Study for Indian Classical Music Forums. *Computacion y Sistemas*, 20(3), 505–513. <https://doi.org/10.13053/CyS-20-3-2464>
- Roth, D., Anderson, R., Luttin, S., & Microsoft. (2019). *Introduction to ASP.NET Core | Microsoft Docs*. <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-3.0>
- Stack Overflow. (2020). *Learning Entity Framework*. Introducing .NET 4.0.
https://doi.org/10.1007/978-1-4302-2456-3_8
- TutorialPoint. (2019). *The Entity Framework*. Applied WPF 4 in Context.
https://doi.org/10.1007/978-1-4302-3471-5_6
- TutorialPoint, & Wall, L. (2019). *Angular 7*. 2.
<https://doi.org/10.1017/CBO9781107415324.004>

- TutorialsPoint. (2019). *Learn Angular 4*. <https://doi.org/10.1017/CBO9781107415324.004>
- Varga, S., Cherry, D., & D'Antoni, J. (2016). *Introducing Microsoft SQL Server 2016*.
- Vásquez García, A., Sangerman- Jarquín, D. M., & Reyes Muro, L. (2017). Desarrollo de una aplicación web para evaluar cultivos agrícolas a través del método de la MAP. *Revista Mexicana de Ciencias Agrícolas*, 8(8), 1813.
<https://doi.org/10.29312/remexca.v8i8.704>
- Videla, S., Guziolowski, C., Eduati, F., Thiele, S., Gebser, M., Nicolas, J., Saez-Rodriguez, J., Schaub, T., & Siegel, A. (2015). Learning Boolean logic models of signaling networks with ASP. *Theoretical Computer Science*, 599, 79–101.
<https://doi.org/10.1016/j.tcs.2014.06.022>
- Wells, D. (2013). *Extreme Programming: A gentle introduction*.
<http://www.extremeprogramming.org/>
- Wolfengagen, V. E., Ismailova, L. Y., & Kosikov, S. V. (2018). Model of conversion of data objects for defining the object-relation mapping. *Procedia Computer Science*, 123, 541–546. <https://doi.org/10.1016/j.procs.2018.01.082>
- Yadav, K. S., Yasvi, M. A., & Shubhika. (2019). Review On Extreme Programming-XP. *International Conference on Robotics, Smart Technology and Electronics Engineering, At Delhi, April*.
<https://www.researchgate.net/publication/332465869%0D>

ANEXOS

Anexo 1: Mapic

Variable Fáctica	Dimensiones	Indicadores
1. Gestión horas máquina de trabajo	1.1 Gestión de Operarios	1.1.1 Operarios registrados y visualizados por usuario administrativo.
	1.2 Gestión de Informes horas máquinas	
	1.3 Gestión de Máquinas	1.1.2 Máquinas registradas y visualizadas por usuario administrativo
		1.1.3 Partes Diarios registrados y visualizados por usuario administrativo.
Variable Temática	Ejes Temáticos	Sub Ejes Temáticos
2. Entity Framework Asp .Net Core	2.1 Teoría de Entity Framework	2.1.1 ORM definición 2.2.1 Ventajas de Asp.Net Core
	2.2 Teoría de Asp.Net Core	2.3.1 Módulos
3. Angular	2.3 Teoría de Angular	2.3.2 Componentes 2.3.3 Servicios e Inyección de dependencias
Variable Propositiva	Ejes Propositivos	Sub Ejes Propositivos
2. Desarrollar Sistema Web	4.1 Metodología:	4.1.1 Planificación
	Programación Extrema	4.1.2 Diseño
		4.1.3 Codificación
		4.1.4 Pruebas

Anexo 2: Escenario de prueba de la usabilidad

Partes diarios de horas máquina de diferentes operarios y máquinas en la obra
 “Mejoramiento de los servicios de educación secundaria en la institución educativa
 Horacio Zeballos Gamez de Juliaca Provincia de San Roman Puno”.

PARTE DIARIO

EJECUCIÓN DE OBRAS CIVILES
 ALQUILER DE MAQUINARIAS Y TRANSPORTES DE
 MATERIALES DE CONSTRUCCIÓN EN GENERAL
 Celular: 951 587493 - 947677489 - 989 487704

CONTROL DIARIO DE TRABAJO

OBRA: _____
 MAQUINARIA: Excavadora MODELO: _____ PLACA: _____
 OPERADOR: Segundo Sanchez Fernandez FECHA: 19/06/2019

Nº	01907	MAÑANA			TARDE			Total Horas
		Desde	Hasta	Hrs.	Desde	Hasta	Hrs.	
Horas Trabajadas	Maquina	9:00 am 1:00 pm						

HOROMETRO	INICIO	FINAL	DIFERENCIA

Nº	DESCRIPCIÓN DE TRABAJO	HR. INICIO	HR. FINAL	PARCIAL	FIRMA	OBSERVACIONES
1	Demolicion A-1	9:00 am	9:50 am			
2	Demolicion A-2	10:00 am	11:00 am			
4	Demolicion A-3	11:00 am	12:00 pm			
4	Demolicion A-4	12:00 pm	1:00 pm			
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

CONSUMO		
Gasolina Gls.	Acetite Motor	A. de Hidraulico
Petroleo Gls.	A. de Transp.	Grasa

OBSERVACIONES:

PARTE DIARIO

EJECUCIÓN DE OBRAS CIVILES
 ALQUILER DE MAQUINARIAS Y TRANSPORTES DE
 MATERIALES DE CONSTRUCCIÓN EN GENERAL
 Celular: 951 587493 - 947677489 - 989 487704

CONTROL DIARIO DE TRABAJO

OBRA: _____
 MAQUINARIA: Druse - Pala Cereabra MODELO: _____ PLACA: _____
 OPERADOR: Ceballos Homaco Perez FECHA: 12-06-2019

Nº	01908	MAÑANA			TARDE			Total Horas
		Desde	Hasta	Hrs.	Desde	Hasta	Hrs.	
Horas Trabajadas	Maquina	11 am 4 pm.						

HOROMETRO	INICIO	FINAL	DIFERENCIA

Nº	DESCRIPCIÓN DE TRABAJO	HR. INICIO	HR. FINAL	PARCIAL	FIRMA	OBSERVACIONES
1	Carga de material demolido A-4	11:00 am	12:00 pm			
2	Carga de material demolido A-2	12:30 pm	1:45 pm			
5	Carga de material demolido A-3	2:20 pm	3:00 pm			
7	Carga de material demolido A-4	3:30 pm	4:00 pm			
8						
9						
10						
11						
12						
13						
14						
15						
16						

CONSUMO		
Gasolina Gls.	Acetite Motor	A. de Hidraulico
Petroleo Gls.	A. de Transp.	Grasa

OBSERVACIONES:

PARTE DIARIO



EJECUCIÓN DE OBRAS CIVILES
ALQUILER DE MAQUINARIAS Y TRANSPORTES DE
MATERIALES DE CONSTRUCCIÓN EN GENERAL
Celular: 951 587493 - 947677489 - 989 487704



CONTROL DIARIO DE TRABAJO

OBRA: _____
MAQUINARIA: Camión Voltrados MODELO: _____ PLACA: _____
OPERADOR: Luis Tuti - Tama FECHA: 14-06-2019

Nº	01909	MAÑANA			TARDE			Total Horas
		Desde	Hasta	Hrs.	Desde	Hasta	Hrs.	
Horas Trabajadas	Maquina	11:30 am	4:30 pm					

HOROMETRO	INICIO	FINAL	DIFERENCIA
-----------	--------	-------	------------

Nº	DESCRIPCIÓN DE TRABAJO	HR. INICIO	HR. FINAL	PARCIAL	FIRMA	OBSERVACIONES
1	Traslado material demolido A-1	11:30 am	12:30 am			
2	Traslado material demolido A-2	1:00 pm	2:00 pm			
4	Traslado material demolido A-3	3:00 pm	3:30 pm			
5	Traslado material demolido A-4	4:00 pm	4:30 pm			
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

CONSUMO

Gasolina Gls.	Aceite Motor	A. de Hidráulico
Petroleo Gls.	A. de Transp.	Grasa

OBSERVACIONES:

PARTE DIARIO



EJECUCIÓN DE OBRAS CIVILES
ALQUILER DE MAQUINARIAS Y TRANSPORTES DE
MATERIALES DE CONSTRUCCIÓN EN GENERAL
Celular: 951 587493 - 947677489 - 989 487704



CONTROL DIARIO DE TRABAJO

OBRA: _____
MAQUINARIA: Retroexcavadora MODELO: _____ PLACA: _____
OPERADOR: Cristian Mazon Maquera FECHA: 14-06-2019

Nº	01910	MAÑANA			TARDE			Total Horas
		Desde	Hasta	Hrs.	Desde	Hasta	Hrs.	
Horas Trabajadas	Maquina	11:00 am	6:00 pm					

HOROMETRO	INICIO	FINAL	DIFERENCIA
-----------	--------	-------	------------

Nº	DESCRIPCIÓN DE TRABAJO	HR. INICIO	HR. FINAL	PARCIAL	FIRMA	OBSERVACIONES
1	Excavación para zapatas M-1	11:00 am	1:00 pm			
2		2:00 pm	4:00 pm			
4	Excavación para zapatas M-2	4:00 pm	6:00 pm			
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

CONSUMO

Gasolina Gls.	Aceite Motor	A. de Hidráulico
Petroleo Gls.	A. de Transp.	Grasa

OBSERVACIONES:

Una hora de almuerzo 1 pm - 2 pm

PARTE DIARIO



EJECUCIÓN DE OBRAS CIVILES
ALQUILER DE MAQUINARIAS Y TRANSPORTES DE
MATERIALES DE CONSTRUCCIÓN EN GENERAL
Celular: 951 587493 - 947677489 - 989 487704



CONTROL DIARIO DE TRABAJO

OBRA:

MAQUINARIA: Oruga-Excavadora MODELO: PLACA:
OPERADOR: David Huallpa Castillo FECHA: 14/06/19

Nº	01911	MAÑANA			TARDE			Total Horas
		Desde	Hasta	Hrs.	Desde	Hasta	Hrs.	
Horas Trabajadas	Maquina	11:00am	6:00pm					

HOROMETRO	INICIO	FINAL	DIFERENCIA

Nº	DESCRIPCIÓN DE TRABAJO	HR. INICIO	HR. FINAL	PARCIAL	FIRMA	OBSERVACIONES
1	Excavación de zanjas M-2	11:00am	1:00pm			
2		2:00pm	4:00pm			
2						
4	Excavación de zanjas M-1	4:00pm	6:00pm			
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

CONSUMO		
Gasolina Gls.	Aceite Motor	A. de Hidráulico
Petroleo Gls.	A. de Transp.	Grasa

OBSERVACIONES:

1 hora a/muerto

PARTE DIARIO



EJECUCIÓN DE OBRAS CIVILES
ALQUILER DE MAQUINARIAS Y TRANSPORTES DE
MATERIALES DE CONSTRUCCIÓN EN GENERAL
Celular: 951 587493 - 947677489 - 989 487704



CONTROL DIARIO DE TRABAJO

OBRA:

MAQUINARIA: MALLADORA DE CONCRETO MODELO: PLACA:
OPERADOR: Edwin Quispe Luna FECHA: 14/06/2019

Nº	01912	MAÑANA			TARDE			Total Horas
		Desde	Hasta	Hrs.	Desde	Hasta	Hrs.	
Horas Trabajadas	Maquina	4:00 PM	5:00 PM					

HOROMETRO	INICIO	FINAL	DIFERENCIA

Nº	DESCRIPCIÓN DE TRABAJO	HR. INICIO	HR. FINAL	PARCIAL	FIRMA	OBSERVACIONES
1	CIMENTO CORALDO M-2	4:00pm	4:30pm			
2	M-2 ZANJAS					
2	CIMENTO CORALDO M-2	4:30 PM	5:00 PM			
4	M-1 ZANJAS					
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

CONSUMO		
Gasolina Gls.	Aceite Motor	A. de Hidráulico
Petroleo Gls.	A. de Transp.	Grasa

OBSERVACIONES:

PARTE DIARIO



EJECUCIÓN DE OBRAS CIVILES
ALQUILER DE MAQUINARIAS Y TRANSPORTES DE
MATERIALES DE CONSTRUCCIÓN EN GENERAL
Celular: 951 587493 - 947677489 - 989 487704



CONTROL DIARIO DE TRABAJO

OBRA: _____
MAQUINARIA: Mercedes de concreto MODELO: _____ PLACA: _____
OPERADOR: Daniel Mendoza Luna FECHA: 20-06-19

Nº	01913	MAÑANA			TARDE			Total Horas
		Desde	Hasta	Hrs.	Desde	Hasta	Hrs.	
Horas Trabajadas	Maquina	8 am	9 am					
HOROMETRO		INICIO		FINAL		DIFERENCIA		
Nº	DESCRIPCIÓN DE TRABAJO	HR. INICIO	HR. FINAL	PARCIAL	FIRMA	OBSERVACIONES		
1	Cemento cascado merca	8:00 am	8:30 am					
2	M-1 zapatas							
2	Cemento cascado merca	8:30 am	9:00 am					
4	M-2 zapatas							
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
CONSUMO								
Gasolina Gls.		Aceite Motor		A. de Hidráulico				
Petroleo Gls.		A. de Transp.		Grasa				

OBSERVACIONES:

PARTE DIARIO



EJECUCIÓN DE OBRAS CIVILES
ALQUILER DE MAQUINARIAS Y TRANSPORTES DE
MATERIALES DE CONSTRUCCIÓN EN GENERAL
Celular: 951 587493 - 947677489 - 989 487704



CONTROL DIARIO DE TRABAJO

OBRA: Mejoramiento de la redacción de educación secundaria en la I.E. Horacio Tebballos Garmen Joloca - San Antonio - Peru
MAQUINARIA: Camion Volador MODELO: _____ PLACA: _____
OPERADOR: Eduardo Quipe Tipula FECHA: 20/06/2019

Nº	01914	MAÑANA			TARDE			Total Horas
		Desde	Hasta	Hrs.	Desde	Hasta	Hrs.	
Horas Trabajadas	Maquina	8:30 am	10:30 am					
HOROMETRO		INICIO		FINAL		DIFERENCIA		
Nº	DESCRIPCIÓN DE TRABAJO	HR. INICIO	HR. FINAL	PARCIAL	FIRMA	OBSERVACIONES		
1	Traslado material	8:30 am	9:30 am					
2	Extracción M-1							
2	Traslado material extracción M-2	10:00 am	10:50 am					
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
CONSUMO								
Gasolina Gls.		Aceite Motor		A. de Hidráulico				
Petroleo Gls.		A. de Transp.		Grasa				

OBSERVACIONES:

PARTE DIARIO



EJECUCIÓN DE OBRAS CIVILES
ALQUILER DE MAQUINARIAS Y TRANSPORTES DE
MATERIALES DE CONSTRUCCIÓN EN GENERAL
Celular: 951 587493 - 947677489 - 989 487704

CONTROL DIARIO DE TRABAJO

OBRA:

MAQUINARIA: Oruga para Cascaños MODELO: PLACA:

OPERADOR: Wily Cerezo LOMA FECHA: 20-06-2019

Nº	01915	MAÑANA			TARDE			Total Horas
		Desde	Hasta	Hrs.	Desde	Hasta	Hrs.	
Horas Trabajadas	Maquina	8:00 am	10:30 am					

HOROMETRO	INICIO	FINAL	DIFERENCIA

Nº	DESCRIPCIÓN DE TRABAJO	HR. INICIO	HR. FINAL	PARCIAL	FIRMA	OBSERVACIONES
1	Traslado de tierra	8 am	9 am			
2	excavación M-1					
2	Traslado de tierra	9:30 am	10:30 am			
4	excavación M-2					
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

CONSUMO

Gasolina Gls.	Aceite Motor	A. de Hidráulico
Petroleo Gls.	A. de Transp.	Grasa

OBSERVACIONES:

PARTE DIARIO



EJECUCIÓN DE OBRAS CIVILES
ALQUILER DE MAQUINARIAS Y TRANSPORTES DE
MATERIALES DE CONSTRUCCIÓN EN GENERAL
Celular: 951 587493 - 947677489 - 989 487704

CONTROL DIARIO DE TRABAJO

OBRA: MEJORAMIENTO DE LOS SERVICIOS DE EDUCACION SECUNDARIA DE CA

I.E. MANALO ZEBALLOS GAMER JUCIAEN - S. ROMAN - PUNO

MAQUINARIA: MOTOCARRO DE CONCRETO MODELO: 4.0 HP PLACA: N2400

OPERADOR: MARIO VILA CASANOVA FECHA: 24/06/2019

Nº	01916	MAÑANA			TARDE			Total Horas
		Desde	Hasta	Hrs.	Desde	Hasta	Hrs.	
Horas Trabajadas	Maquina	10:00 am	11:00 am					

HOROMETRO	INICIO	FINAL	DIFERENCIA

Nº	DESCRIPCIÓN DE TRABAJO	HR. INICIO	HR. FINAL	PARCIAL	FIRMA	OBSERVACIONES
1	CONCRETO SOBRECIMENTOS	10:00 am	10:30 am			
2	M-1					
2	CONCRETO SOBRECIMENTOS	10:30 am	11:00 am			
4	M-2					
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

CONSUMO

Gasolina Gls.	Aceite Motor	A. de Hidráulico
Petroleo Gls.	A. de Transp.	Grasa

OBSERVACIONES:

Anexo 3: Resultados de pruebas de la usabilidad.

Se registro los operarios que llenaron las fichas de cada parte diario.

Contratistas Generales Home Parte Diario Gestión de Operario Máquina

Gestión de Operarios

La página mostrará opearios,creará y actualizará

[Nuevo](#)

Operario Id	Nombre	Apellido Paterno	Apellido Materno	Dni	Teléfono	Fecha Nacimiento	Opciones
2032	Segundo	Sanchez	Fernandez	87455587	987441400	1979-11-28	Edit Delete
2031	Celso	Hannco	Perez	65874477	967785442	1991-11-05	Edit Delete
2030	Luis	Tunti	Toma	45774411	988763255	1986-02-12	Edit Delete
2029	Clinton	Marin	Maquera	76325418	998774410	1992-10-11	Edit Delete
2028	David	Huallpa	Castillo	20458463	974587458	1983-05-22	Edit Delete
2027	Edwin	Quispe	Luna	68574102	968574120	1980-08-06	Edit Delete
2026	Daniel	Mendoza	Lima	57462314	965471285	1988-03-29	Edit Delete
2025	Willy	Cerezo	Loma	78463211	945632171	1992-06-18	Edit Delete
2024	Eduardo	Quispe	Tipula	85472163	945325750	1990-11-20	Edit Delete
2023	Mario	Vilca	Ccasani	44571269	984756231	1994-11-23	Edit Delete

« Previous **1** 2 3 4 Next »

Se registro las máquinas que se encuentran en los partes diarios.

Contratistas Generales Home Parte Diario Gestión de Operario Máquina

Gestión de Máquinas

La página mostrará máquinas,creará y actualizará

[Nuevo](#)

Maquina Id	Nombre	Modelo	Placa	Opciones
2014	Mezcladora de concreto	9.0 Hp honda	n2400	Edit Delete
2013	Oruga excavadora	Cat	EI - 8248	Edit Delete
2012	Retroexcavadora	Cat	EO - 6242	Edit Delete
2011	Oruga pala cargadora	Cat	FO - 1752	Edit Delete
2010	Camión Volcador	Ford	BB-8784	Edit Delete

« Previous **1** 2 3 Next »

Cada registro de parte diario se realizó en este formulario, en las fichas del anexo 2 se muestra varios detalles de trabajo; en el sistema al seleccionar **agregar detalle** se puede agregar cuantos detalles de trabajo necesites. Con esta funcionalidad se registro todos los partes diarios de la ficha.

Editar Parte Diario

Obra: Mejoramiento de los servicios en la institución educativa Horacio Zeballos Gamez Juli

Máquina: Mezcladora de concreto

Operario: Mario Vilca Ccasani

Fecha: 2019-06-24

Hora inicio de trabajo: 10 : 00

Hora fin de trabajo: 11 : 00

Observaciones:

Agregar detalle

Descripción	Hora Inicio:	Hora Final:	Observación
Concreto sobre cimiento	10 : 00	10 : 30	
Remover			
Concreto sobre cimiento	10 : 30	11 : 00	
Remover			

Total: 01:00:00

Guardar **Cancelar**

En cada registro de partes diarios se mostró sencillez al llenar el formulario y se mostró rapidez en el llenado de los campos, los autocompletados y la parte dinámica de los detalles de trabajo. El resultado es el registro de todas las fichas.

Contratistas Generales Home Parte Diario Gestión de Operario Máquina

Gestión de Partes Diarios

La página mostrará partes diarios, creará y actualizará

[Nuevo](#)

Máquina	Operario	Fecha	Hora inicio operario	Hora fin operario	Hora total máquina	Opciones
Mezcladora de concreto	Mario Vilca Ccasani	2019-06-24	10:00:00	11:00:00	01:00:00	Edit Delete
Camión Volcador	Eduardo Quispe Tipula	2019-06-20	08:30:00	10:50:00	01:50:00	Edit Delete
Oruga pala cargadora	Willy Cerezo Loma	2019-06-20	08:00:00	10:30:00	02:00:00	Edit Delete
Mezcladora de concreto	Daniel Mendoza Lima	2019-06-20	08:00:00	09:00:00	01:00:00	Edit Delete
Mezcladora de concreto	Edwin Quispe Luna	2019-06-19	16:00:00	17:00:00	01:00:00	Edit Delete
Oruga excavadora	David Huallpa Castillo	2019-06-19	11:00:00	18:00:00	06:00:00	Edit Delete
Retroexcavadora	Clinton Marin Maquera	2019-06-19	11:00:00	18:00:00	06:00:00	Edit Delete
Camión Volcador	Luis Tunti Toma	2019-06-19	11:30:00	16:30:00	03:20:00	Edit Delete
Oruga pala cargadora	Celso Hannco Perez	2019-06-19	11:00:00	16:00:00	03:25:00	Edit Delete
Excavadora	Segundo Sanchez Fernandez	2019-06-19	09:00:00	13:00:00	03:40:00	Edit Delete

« Previous **1** 2 3 Next »

Anexo 4: Reportes del sistema web

Contratistas Generales

Home Parte Diario Gestión de Operario Máquina Vista Cerrar Sesión

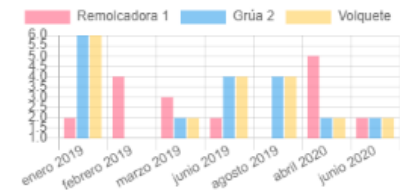
Máquina

Parte Diario

Operario

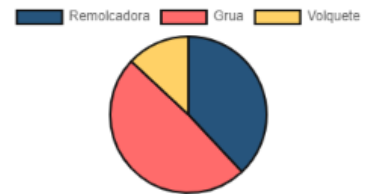
Uso de máquinas

Mostramos una barra con las máquinas más solicitadas



Horas de máquinas

Mostramos el porcentaje de horas de trabajo de las máquinas



Cuadro general

Mostramos la línea del tiempo de uso de las máquinas del año

