



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

IMPLEMENTACIÓN DE UN APLICATIVO MÓVIL Y WEB PARA LA
GESTIÓN ADMINISTRATIVA DE LA EMPRESA CONSTSA USANDO LA
METODOLOGÍA MOBILE-D

RAMIREZ NOVILLO PAULO CESAR
INGENIERO DE SISTEMAS

MACHALA
2021



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

IMPLEMENTACIÓN DE UN APLICATIVO MÓVIL Y WEB PARA
LA GESTIÓN ADMINISTRATIVA DE LA EMPRESA CONSTSA
USANDO LA METODOLOGÍA MOBILE-D

RAMIREZ NOVILLO PAULO CESAR
INGENIERO DE SISTEMAS

MACHALA
2021



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

TRABAJO TITULACIÓN
PROPUESTAS TECNOLÓGICAS

IMPLEMENTACIÓN DE UN APLICATIVO MÓVIL Y WEB PARA LA GESTIÓN
ADMINISTRATIVA DE LA EMPRESA CONSTSA USANDO LA METODOLOGÍA
MOBILE-D

RAMIREZ NOVILLO PAULO CESAR
INGENIERO DE SISTEMAS

CÁRDENAS VILLAVICENCIO OSCAR EFRÉN

MACHALA, 28 DE SEPTIEMBRE DE 2021

MACHALA
2021

IMPLEMENTACIÓN DE UN APLICATIVO MÓVIL Y WEB PARA LA GESTIÓN ADMINISTRATIVA DE LA EMPRESA CONSTSA USANDO LA METODOLOGÍA MOBILE-D

INFORME DE ORIGINALIDAD

3%

INDICE DE SIMILITUD

2%

FUENTES DE INTERNET

0%

PUBLICACIONES

2%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1

1library.co

Fuente de Internet

<1%

2

www.es.paessler.com

Fuente de Internet

<1%

3

Submitted to Instituto Tecnológico de Costa Rica

Trabajo del estudiante

<1%

4

Submitted to Universidad de Cundinamarca

Trabajo del estudiante

<1%

5

Submitted to Universitat Politècnica de València

Trabajo del estudiante

<1%

6

repositorio.unesum.edu.ec

Fuente de Internet

<1%

7

Submitted to Escuela Politecnica Nacional

Trabajo del estudiante

<1%

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

El que suscribe, RAMIREZ NOVILLO PAULO CESAR, en calidad de autor del siguiente trabajo escrito titulado IMPLEMENTACIÓN DE UN APLICATIVO MÓVIL Y WEB PARA LA GESTIÓN ADMINISTRATIVA DE LA EMPRESA CONSTSA USANDO LA METODOLOGÍA MOBILE-D, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

El autor declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 28 de septiembre de 2021



RAMIREZ NOVILLO PAULO CESAR
0703828384

DEDICATORIA

El presente trabajo está dedicado de manera muy especial para mis padres César Ramírez Quezada y Mónica Novillo Vásquez, quienes me han brindado su apoyo en todo momento, enseñándome a no darme por vencido en alcanzar mis metas respaldándome en todas mis decisiones.

También se lo dedicó con un enorme orgullo a mi esposa e hijo por ser el pilar de mi vida, por nunca dejar de creer en mí respaldándome en todo momento, brindándome la fuerza y empuje para poder lograr mi meta, recalcándome siempre lo orgullosos que están de mí.

A mi abuelo quien en vida fue Marco Bolívar Novillo López, por apoyarme en todo momento y por su inmenso cariño que me brindaba día a día, enseñándome a ser un hombre honesto y de bien.

Por último, pero no menos importante a mis hermanos, tíos, primos, amigos y compañeros que estuvieron a lo largo de esta hermosa travesía, cada uno dejando una pequeña huella ayudándome cada día a ser mejor.

Sr. Ramírez Novillo Paulo César

AGRADECIMIENTO

Agradezco en primer lugar a Dios por permitirme lograr con éxito mi etapa profesional, a mis padres por ser mis guías en este camino, apoyándome con sus consejos y predicando con el ejemplo.

También quiero agradecer inmensamente a mi esposa por brindarme todo su cariño y respaldo para cumplir mi sueño, por no dejarme nunca solo y por acompañarme en todos los acontecimientos que viví.

A mi hijo por ser mi razón de ser, mi faro, mi luz, por brindarme siempre las fuerzas para seguir adelante haciéndome feliz a cada instante con su inocencia, el ser por quien busco mejorar cada día.

También agradezco a mis amigos y compañeros, que estuvieron a mi lado de diferentes formas compartiendo alegrías y momentos amargos, pero siempre brindándome su apoyo o aportando con su granito de arena.

Finalmente agradezco a la Universidad Técnica de Machala por permitirme formar parte de esta hermosa institución, por sus excelentes docentes quienes buscan siempre brindar todo su conocimiento y a mi tutor, el Ing. Oscar Cárdenas Villavicencio quien con su paciencia y experiencia logró guiarme en la elaboración de este proyecto.

Sr. Ramírez Novillo Paulo César

RESUMEN

Los smartphones en los últimos años han ocupado en las personas un lugar significativo y a la vez limitado a su propio entorno al momento de ejecutar sus actividades diarias. El cambio tecnológico que se ha ido proporcionando a brindado la posibilidad de hacer uso de plataformas de código abierto permitiendo a los desarrolladores realizar aplicativos móviles con facilidad. A su vez las empresas han ido optando por realizar una migración paulatina a la tecnología, en específico aplicativos móviles y web, en base a sus procesos y actividades que se realizan de manera cotidiana para conseguir la productividad y eficiencia de las mismas. A pesar de existir diferentes tipos de herramientas que permiten el desarrollo móvil y web, es necesario realizar una planificación de desarrollo apegado a las actividades y reglamentos de las empresas para así garantizar la calidad del software. La metodología de desarrollo móvil “Mobile-D” permite ejecutar sus fases y diferentes etapas enfocadas a la productividad del aplicativo, permitiendo desarrollar el sistema en un periodo corto de tiempo y garantizando el funcionamiento y calidad del aplicativo. Actualmente la empresa de construcción ConstSA dedicada a la elaboración de obras tipo urbanización radicada en la provincia de El Oro, hace uso de herramientas ofimáticas y elementos físicos para llevar un registro de las evidencias generadas en la fase de construcción de los proyectos de viviendas siendo el administrador el encargado de llevar los procesos como registro de asistencia diaria de obreros, registro diario de progreso de obras, asignación de actividades diarias y generación de reportes mensuales sobre el estado actual que poseen las obras a las que ha sido asignado. Registros que, al ser llevados de forma manual están propensos a errores humanos en este caso por parte de los encargados de obra, a la vez se vuelve tedioso el proceso diario de registro de la información. Por tal motivo se presentó una solución para la problemática, tomando en consideración la recolección de información a nivel de requerimientos proporcionada por las partes interesadas y el modelo de negocio por parte de la empresa. Como resultado se dio la propuesta tecnológica de la implementación de un aplicativo híbrido móvil y web para los procesos de gestión administrativa de la empresa constructora ConstSA, estableciendo su fundamentación en la metodología de desarrollo móvil

“Mobile-D”, haciendo uso de los frameworks de desarrollo multiplataforma para el cliente y el servidor, basados en el lenguaje JavaScript como son: React Native, Expo y NodeJS, implementando su arquitectura en la nube con la ayuda de los servicios de Firebase, así como su base de datos en tiempo real; permitiendo la obtención rápida de información y seguridad en el sistema. Garantizando la calidad del software mediante la ayuda de la herramienta para generación de pruebas unitarias o globales en base a componentes, funciones, declaraciones, sintaxis de métodos y líneas de código, en entornos JavaScript como lo es Jest. Logrando así que la empresa pueda realizar sus procesos y actividades de manera eficaz, ágil y optimizada, permitiendo aumentar su rentabilidad y productividad.

Palabras claves: Mobile-D, React Native, NodeJS, Expo, Gestión de Empleados, Desarrollo Híbrido, Desarrollo multiplataforma.

ABSTRACT

Smartphones in recent years have occupied a significant place in people and at the same time limited to their own environment when executing their daily activities. The technological change that has been provided has provided the possibility of making use of open-source platforms allowing developers to make mobile applications with ease. Simultaneously, companies have chosen to make a gradual migration to technology, specifically mobile and web applications, based on their processes and activities that are carried out on a daily basis to achieve productivity and efficiency. Despite the existence of different types of tools that allow mobile and web development, it is necessary to carry out a development planning attached to the activities and regulations of the companies in order to guarantee the quality of the software. Currently the construction company ConstSA dedicated to the development of urbanizations, works located in the province of El Oro, makes use of office tools and physical elements to record the evidence generated in the construction phase of housing projects, the administrator being in charge of carrying processes such as daily worker attendance record, daily work progress record, activity assignment and generation of monthly reports on the status of the works to which it has been assigned. Records that being kept manually are prone to human errors in this case by construction managers, at the same time the daily information recording process becomes tedious. For this reason, a solution for the problem was presented, taking into consideration the collection of information at the level of requirements provided by the stakeholders and the business model by the company. As a result, the technological proposal was presented for the implementation of a hybrid mobile and web application for the employee management processes of the company ConstSA, establishing its foundation in the mobile development methodology "Mobile-D", development methodology "Mobile-D", making use of the frameworks of cross-platform development for the client and the server, based on the JavaScript language such as: React-Native, Expo and NodeJS, implementing its architecture in the cloud with help of Firebase services, as well as its real-time database; allowing a rapid obtaining of information and security in the system. Guaranteeing the quality of the software through the help of the tool for generating unit or global tests

based on components, functions, declarations, method syntax and lines of code, in JavaScript environments such as Jest. Thus, the company can carry out its processes and activities in an efficient, agile and optimized way, allowing it to increase its profitability and productivity.

Keywords: Mobile-D, React Native, NodeJS, Expo, Employee Management, Hybrid Development, Multiplatform Development.

INTRODUCCIÓN

La tecnología cumple un papel fundamental dentro de los procesos y actividades de una empresa, cada vez más empresas optan por reformar la manera en que se llevan a cabo dichos procesos, con el fin de optimizar su productividad y agilizar sus actividades. Hoy en día los smartphones están al alcance de todos permitiendo a las personas realizar sus actividades cotidianas en base a una aplicación móvil, ofreciendo una alternativa para sus labores del día, brindando eficiencia y ahorro de tiempo.[1]

Las empresas buscan la manera en apegar sus actividades en base a un sistema que sea implementado acorde a sus necesidades y reglamentos. Por ende, los dispositivos tecnológicos que existen en la actualidad, permiten desarrollar y ejecutar sistemas complejos de alto nivel facilitando su portabilidad. El desarrollo de software móvil aumenta de manera acelerada debido a las tendencias de desarrollo híbrido o multiplataforma, generando así que las empresas hagan uso de estos sistemas en sus smartphones o computadores, permitiendo que los usuarios obtengan la información cuando sea necesaria. [2]

Actualmente ConstSA, empresa dedicada a la construcción, emplea sus tareas cotidianas de registro de progreso de actividades en herramientas ofimáticas y elementos físicos que son usados por parte de empleados que están a cargo de los procesos generales en la etapa de construcción. Dichos registros deben ser generados diariamente con el fin de al finalizar cada mes poder obtener un reporte de las actividades en general. A su vez el responsable de obra tiene el deber de llevar un registro diario de la asistencia de los empleados, permitiendo obtener al finalizar la obra la cantidad de días trabajados por cada empleado, generando el pago total por el servicio acorde a los días registrados.

Estos procesos al ser realizados de forma manual y con elementos físicos, tienden a generar errores o confusiones a lo largo del tiempo de ejecución, ya que no existe un centro de acopio de recolección, ordenamiento y análisis de los registros y reportes generados por los responsables de obra. Adicional la información no puede ser accesible de manera instantánea ni compartida; y los

empleados deben esperar la presencia del encargado para recibir las actividades que serán realizadas en el día.

Debido a la problemática existente, nace la necesidad de la realización de un aplicativo móvil y web que administre la gestión de obras y empleados en base a los distintos proyectos de la empresa, con la ayuda de las herramientas basadas en JavaScript como: React Native y Expo del lado del cliente y NodeJS del lado del servidor, empleando un trabajo en conjunto con los servicios de Firebase y su base de datos en tiempo real, haciendo uso de la metodología de desarrollo móvil “Mobile-D” implementando sus fases de desarrollo para la obtención de un software de calidad.

A continuación, se detalla una estructura por capítulos de la propuesta del proyecto tecnológico:

Capítulo I: En este primer apartado se especifica la situación de la problemática del proyecto, su caso de estudio y su diseño de propuesta para la solución del mismo, por lo que la recolección de información debe ser detallada definiendo un esquema de requerimientos, y fundamentando el uso de herramientas y tecnología.

Capítulo II: En este segundo apartado se diseñan los diagramas y modelos que sean requeridos para la implementación del prototipo; además se define el patrón de diseño de las interfaces y las pruebas necesarias para comprobar la ejecución y funcionalidad del sistema.

Capítulo III: En este apartado final se recolecta información de todas las pruebas realizadas al sistema, siguiendo las recomendaciones de la metodología en la ejecución de pruebas con Jest, determinando la eficacia y eficiencia del sistema.

CONTENIDO

1.	CAPITULO I: DIAGNOSTICO DE NECESIDADES Y REQUISITOS.....	15
1.1.	DESCRIPCIÓN DEL CONTEXTO Y HECHOS DE INTERÉS.....	15
1.2.	ESTABLECIMIENTOS DE REQUERIMIENTOS.....	16
1.3.	JUSTIFICACIÓN DE REQUERIMIENTOS.....	16
2.	CAPITULO II: DESARROLLO DEL PROTOTIPO	17
2.1.	DEFINICIÓN DEL PROTOTIPO TECNOLÓGICO	17
2.1.1.	Arquitectura de Hardware.....	18
2.1.2.	Arquitectura de Hardware a Implementar.....	18
2.1.3.	Arquitectura de Software	19
2.1.4.	Arquitectura de Software a Implementar	19
2.1.5.	Capa de presentación o visualización	20
2.1.6.	Capa lógica o de negocio	21
2.1.7.	Capa de datos	21
2.2.	FUNDAMENTACIÓN TEÓRICA DEL PROTOTIPO.....	21
2.2.1.	Metodología de desarrollo de software.....	22
2.2.1.1.	Metodologías Tradicionales	22
2.2.1.2.	Metodologías Ágiles	23
2.2.1.3.	Metodologías Móviles	23
2.2.1.4.	Metodología Mobile-D	23
2.2.2.	Servidor	24
2.2.2.1.	Servidor Web	24
2.2.2.2.	NodeJS Framework	25
2.2.3.	Cliente.....	26
2.2.3.1.	React-Native Framework.....	26
2.2.3.1.1.	Nativo.....	27
2.2.3.1.2.	JavaScript/JSX.....	27
2.2.3.2.	Componentes de React-Native.....	27
2.2.3.3.	Expo Framework.....	28
2.2.4.	Jest	28
2.2.5.	Cloud computing	28
2.2.5.1.	Firebase.....	29

2.2.5.2.	Base de datos Firebase en tiempo real.....	29
2.2.6.	Herramientas CASE.....	30
2.2.6.1.	Microsoft Visio.....	30
2.2.6.2.	Microsoft Project	30
2.2.6.3.	Enterprise Architect	31
2.2.6.4.	Balsamiq Mockups.....	31
2.2.6.5.	StarUML	31
2.3.	OBJETIVOS DEL PROTOTIPO.....	31
2.3.1.	Objetivo General	31
2.3.2.	Objetivos Específicos.....	31
2.4.	DISEÑO DEL PROTOTIPO.....	32
2.4.1.	Fase de exploración.....	32
2.4.1.1.	Alcance del proyecto	32
2.4.1.2.	Análisis de factibilidad	33
2.4.1.2.1.	Factibilidad operativa	33
2.4.1.2.2.	Factibilidad técnica.....	34
2.4.1.2.3.	Factibilidad económica	35
2.4.1.3.	Identificación de interesados o “stakeholders”	36
2.4.1.4.	Requerimientos.....	36
2.4.1.5.	Requerimientos Funcionales.....	37
2.4.1.6.	Requerimientos No Funcionales.....	37
2.4.2.	Fase de inicialización.....	38
2.4.2.1.	Historias de usuario	38
2.4.2.2.	Gestión de Cronograma	44
2.4.2.3.	Configuración del Ambiente del Desarrollo	47
2.4.2.3.1.	Instalación de Visual Studio Code.....	47
2.4.2.3.2.	Instalación de Android Studio	47
2.4.2.3.3.	Instalación de React Native	47
2.4.2.3.4.	Instalación de NodeJS	48
2.4.2.3.5.	Instalación de Expo	48
2.4.2.3.6.	Instalación de Firebase	48
2.4.2.3.7.	Instalación de Postman.....	49
2.4.2.4.	Diseño de la Aplicación	49

2.4.2.5.	Diseño de Base de Datos.....	50
2.4.2.6.	Esquema de Navegabilidad	56
2.4.2.7.	Diseño de Interfaces.....	57
2.4.2.7.1.	Diseño de Interfaz Web	57
2.4.2.7.2.	Diseño de Interfaz Móvil	60
2.4.2.8.	Diagramas de caso de uso.....	63
2.4.3.	Producción y Estabilización.....	65
2.4.3.1.	Estándares de codificación	65
2.4.3.2.	Estructura de los directorios.....	65
2.5.	EJECUCIÓN Y/O EMBALAJE DEL PROTOTIPO.....	66
2.5.1.	Inicio de Sesión	67
2.5.2.	Pantalla Principal	68
2.5.3.	Registro Obras	68
2.5.4.	Registro Empleados.....	69
2.5.5.	Visualizar Obras	70
3.	CAPÍTULO III. EVALUACIÓN DEL PROTOTIPO.....	72
3.1.	Plan de Evaluación	72
3.2.	Resultados de Evaluación.....	72
3.3.	CONCLUSIONES.....	75
3.4.	RECOMENDACIONES.....	76
	BIBLIOGRAFÍA.....	78
	ANEXOS	83
	ANEXO A: FORMATO DIGITAL DE REGISTRO DE ACTIVIDADES DIARIAS	83
	ANEXO B: FORMATO DIGITAL DE REGISTRO DE ASISTENCIA.....	83
	ANEXO C: ANÁLISIS DE RIESGOS.....	84

ÍNDICE DE TABLAS

Tabla 1: Requisitos funcionales del sistema	32
Tabla 2: Requerimientos no funcionales del sistema	33
Tabla 3: Factibilidad operativa	33
Tabla 4: Factibilidad técnica.....	34
Tabla 5: Factibilidad económica	35
Tabla 6: Matriz de Interesados	36
Tabla 7: Requerimientos Funcionales	37
Tabla 8: Requerimientos No Funcionales	38
Tabla 9: Historia de usuario N.-1	38
Tabla 10: Historia de usuario N.-2	39
Tabla 11: Historia de usuario N.-3	40
Tabla 12: Historia de usuario N.-4	40
Tabla 13: Historia de usuario N.-5	41
Tabla 14: Historia de usuario N.- 6	41
Tabla 15: Historia de usuario N.-7	42
Tabla 16: Historia de usuario N.-8	43
Tabla 17: Historia de usuario N.-9	43
Tabla 18: Historia de usuario N.-10	44
Tabla 19: Gestión de Riesgos.....	84

ÍNDICE DE FIGURAS

Figura 1: Arquitectura Hardware del Sistema	19
Figura 2: Modelo de Tres capas	20
Figura 3: Base teórica	22
Figura 4: Ciclo de vida Mobile-D.....	24
Figura 5: Componentes básicos de React-Native	26
Figura 6: Sincronización de datos en tiempo real con diferentes dispositivos...30	
Figura 7: Cronograma de actividades	46
Figura 8: Diseño general del sistema	49
Figura 9: Estructura no relacional de Usuarios	50
Figura 10: Estructura no relacional de Obras	51
Figura 11: Estructura no relacional de asistencia de obreros.....	51
Figura 12: Estructura no relacional de pagos a obreros	52
Figura 13: Diseño de Base de Datos NoSQL 1	53
Figura 14: Diseño de Base de Datos NoSQL 2	54
Figura 15: Diseño de Base de Datos NoSQL 3	55
Figura 16: Diagrama de Navegación Administrador.....	56
Figura 17: Diseño Login del sistema	57
Figura 18: Diseño de página principal	57
Figura 19: Menú lateral.....	58
Figura 20: Diseño de registro de obra	58
Figura 21: Diseño registro de empleado	59
Figura 22: Diseño de Administración de Obras	59
Figura 23: Diseño de Login	60
Figura 24: Diseño de pantalla principal	60
Figura 25: Diseño registro de obra	61
Figura 26: Diseño registro de empleado	62
Figura 27: Diseño administrar obras.....	62
Figura 28: Caso de uso general del sistema.....	63
Figura 29: Caso de uso inicio de sesión	64
Figura 30: Caso de uso visualización de obras	64
Figura 31: Caso de uso visualización de empleados	65
Figura 32: Estructura de los directorios.....	66

Figura 33: Ejecución del sistema web - Inicio de Sesión	67
Figura 34: Ejecución de app en Android - Inicio de Sesión	67
Figura 35: Ejecución del sistema web - Pantalla Principal.....	68
Figura 36: Ejecución de sistema web - Registro de Obra	68
Figura 37: Ejecución de app en Android– Registro de obra	69
Figura 38: Ejecución de sistema web - Registro de Empleado	69
Figura 39: Ejecución de app en Android - Registro de Empleado	70
Figura 40: Pantalla de administración de obras en la web	70
Figura 41: Ejecución de app en Android - Visualizar obras	71
Figura 42: Prueba unitaria a vista Inicio	72
Figura 43: Prueba unitaria a vista Obras.....	73
Figura 44: Prueba unitaria a vista Usuarios	73
Figura 45: Prueba unitaria a vista Registro de Obra	74
Figura 46: Resultado final de Test realizados a diferentes vistas	74
Figura 48: Formato digital de registro de actividades diarias.....	83
Figura 47: Formato digital de registro de asistencia	83

1. CAPITULO I: DIAGNOSTICO DE NECESIDADES Y REQUISITOS

1.1. DESCRIPCIÓN DEL CONTEXTO Y HECHOS DE INTERÉS

El auge de la tecnología ha ido en constante crecimiento en el área de la construcción, cada vez es más evidente el uso de estas tecnologías donde el único beneficiario es la propia empresa que lo adopta. Esto facilita y optimiza los procesos que deben ejecutarse de forma cotidiana a lo largo del día, alcanzando un mayor nivel de competitividad y mejores resultados en rentabilidad.

Dentro de los procesos de la fase de puesta en marcha de la construcción, la empresa ConstSA desarrolla su registro de actividades por parte de un encargado o responsable de la obra (**ver anexo A**), por lo general un Ingeniero Civil, donde este está sujeto a recolectar de forma diaria, información que respalde la ejecución y progreso de las distintas obras durante el tiempo de realización del proyecto.

El encargado deberá entregar al finalizar el día un informe sobre el cumplimiento, progreso y novedades que se hayan generado a lo largo de la actividad laboral diaria, adicional a esto debe llevar un control sobre la asistencia de los diferentes obreros de cada obra de la cual sea responsable (**ver anexo B**), esto se realiza hasta finalizar el proyecto de construcción. El fin de poder obtener un reporte final sobre la asistencia laboral de cada obrero es el poder determinar la cantidad de días laborados, para así poder generar el pago por los servicios prestados.

Todos estos procesos apoyan su ejecución en herramientas ofimáticas (Excel y Word) y elementos físicos (Hoja y lapicero), Volviendo la tarea tediosa y abriendo la posibilidad de que pueda existir un error humano en los registros diarios. Sin embargo, la empresa no posee con una plataforma o aplicativo que gestione la información generada en el día a día, dificultando la obtención de un reporte general de manera rápida para apoyarse en la toma de decisiones.

Como trabajo de titulación se ha realizado la propuesta para buscar satisfacer la problemática planteada, mediante el diseño e implementación de un

aplicativo móvil y web utilizando como base la metodología de desarrollo móvil “Mobile-D”.

1.2. ESTABLECIMIENTOS DE REQUERIMIENTOS

Dentro de la empresa CONSTSA se manejan procesos y actividades de forma manual y física, en lo que respecta a la asignación de actividades y control de los empleados en sus proyectos. Para la recolección de los requerimientos del sistema se tuvo diálogos con los responsables de obras y con los obreros, que son los intérpretes principales dentro del sistema.

Adicionalmente se entrevistó a la persona encargada de recolectar de forma diaria los reportes generados por los responsables de obra, con el fin de saber que procesos se llevan a cabo con dicha información, obteniendo los siguientes parámetros.

- Control de progreso de obras por mes en base a los informes realizados en el día a día. Cada mes se debe generar un reporte sobre la(s) obra(s) que se requieran con el fin de obtener el progreso de las mismas.
- Control de progreso de obras por semana. Cada semana se debe generar un reporte de la(s) obra(s) que se requieran con el propósito de evaluar el progreso.
- Control de asistencia de obreros. Es necesario llevar un registro de asistencia de obreros con el fin de obtener los días totales laborados.
- Control de pagos a obreros. Al finalizar la obra general se debe realizar el pago a los empleados en base a los días laborados.

Cabe destacar que dentro de los requisitos existe la gestión de roles y permisos para los administradores y obreros; además los requisitos funcionales y no funcionales se detallarán en la fase de análisis de la metodología a usar.

1.3. JUSTIFICACIÓN DE REQUERIMIENTOS

El sistema híbrido (Web y Móvil) para la asignación de actividades y control de empleados, tiene como propósito agilizar y optimizar los procesos que se llevan

a cabo en la fase de construcción de viviendas. Ya que originalmente estos se llevan de forma manual o natural donde el responsable o encargado de la obra es quien designa a diario las tareas que deben realizar sus obreros a cargo, así como el tiempo en que deben cumplir dichas actividades. Adicional se deben generar informes de manera semanal y mensual con el fin de poder medir el progreso de las obras que se están realizando, esto ayuda a la toma de decisiones de la empresa permitiendo aumentar su productividad y rentabilidad.

Para poder llevar un control, progreso y observaciones de las construcciones a realizar se requiere de un sistema portable y robusto, disponible a cualquier instante con una infraestructura que brinde mecanismos para el transporte rápido de la información con la capacidad de soporte de afluencia diaria concurrencia de distintos empleados, independiente de la plataforma que use.

Dentro de los roles posibles para los usuarios existen dos, de tipo administrador y de tipo obrero, haciendo que el sistema sea altamente administrable. Todos los requerimientos serán desarrollados siguiendo las últimas tendencias en las herramientas de desarrollo híbrido o multiplataforma, del lado del “Back-End” haciendo uso de una API Rest para la obtención y asignación de información con tecnología de NodeJS, y del “Front-End” usando el framework React-Native y Expo para la visualización de la información. Esto siguiendo la metodología “Mobile-D” debido a la clara definición de sus fases de desarrollo con sus respectivas pruebas, permitiendo obtener un software de calidad.

2. CAPITULO II: DESARROLLO DEL PROTOTIPO

2.1. DEFINICIÓN DEL PROTOTIPO TECNOLÓGICO

El presente proyecto tecnológico tiene como finalidad la automatización de los procesos en el área de construcción de la empresa CONSTSA. Dicho proceso posee una estructura bien definida, donde consta de dos partes; La API Rest para la manipulación de la información y la interfaz para la visualización e interacción de los usuarios con el sistema. A continuación, se detallan los procesos y herramientas para el funcionamiento del aplicativo.

2.1.1. Arquitectura de Hardware

Está basada en visualizar la arquitectura que posee el proyecto, es decir, los equipamientos y herramientas que se emplean para brindar soporte y funcionamiento al sistema.

2.1.2. Arquitectura de Hardware a Implementar

El sistema a desarrollar requiere de un servidor para el manejo de información y peticiones por parte del sistema, en base al análisis se determinó que la afluencia de información es en cantidades pequeñas por lo que se optó utilizar los servicios de Firebase, un servicio en la nube gratuito y de paga, donde proporciona herramientas para la integración del sistema.

Del lado del servidor se encuentra la API, que se encarga del almacenamiento como la obtención de la información, aquí se hace uso de la herramienta “Realtime Database” de Firebase que nos permite almacenar información en manera de árbol (json).

Del lado del Cliente se hace uso de la herramienta Expo, que permite la ejecución del sistema en plataforma Web y Móvil, haciendo posible la visualización de la información.

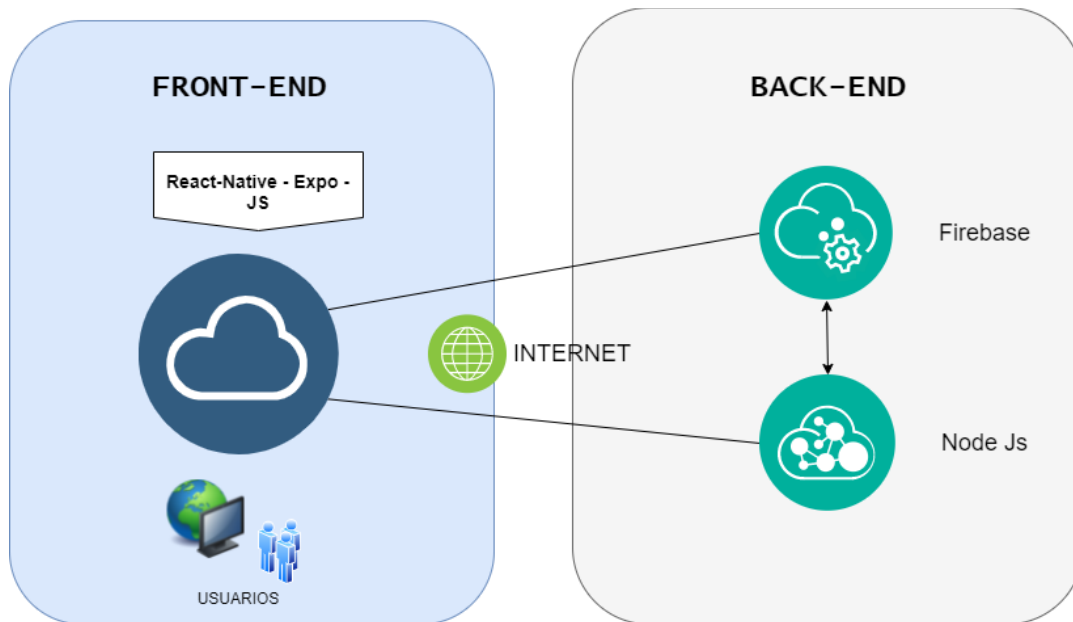


Figura 1: Arquitectura Hardware del Sistema

Fuente: Elaboración propia

Como se aprecia en la **Figura 1**, el sistema hace uso de los servicios de firebase y este a su vez se enlaza con la API en Node Js para el manejo de información, permitiendo así a los clientes visualizar el contenido.

2.1.3. Arquitectura de Software

La arquitectura de software es considerada la estructura de los elementos de software que están involucrados en el sistema, así como las propiedades para la interacción de dichos elementos. [3][4]

2.1.4. Arquitectura de Software a Implementar

Para el sistema propuesto se ha optado por emplear una distribución a nivel de capas, con el propósito de hacer escalable el sistema en un futuro. De este modo se aplicará el modelo de tres capas separando la lógica de datos, la información o lógica de negocio y la presentación al usuario.[5]



Figura 2: Modelo de Tres capas

Fuente: Elaboración propia

En el modelo de la **Figura 2** se observa que el ambiente de ejecución del sistema fue pensado en la nube, haciendo uso de Firebase y sus herramientas como “Realtime Database”, “Autenticación”, “Funciones”, entre otras; permitiendo obtener un ambiente de desarrollo óptimo dentro de una misma plataforma.[6]

2.1.5. Capa de presentación o visualización

Dentro de esta capa se encuentra inmerso todo lo correspondiente a la interfaz o diseño que visualizarán los usuarios; también llamado “frontend”, en esta capa los usuarios podrán interactuar con las funcionalidades del sistema.[7]

El contenido de las interfaces debe ser consistente con la información que se requiera, así como deben satisfacer los requerimientos del usuario, por lo que no se debe excluir información que haya sido proporcionada por el usuario final.

2.1.6. Capa lógica o de negocio

En esta capa se llevan a cabo los procesos que el sistema brindará a los usuarios, es decir, se emplearán los mecanismos para enlazar la capa de presentación y la capa de datos, permitiendo a los usuarios poder obtener la información según su requerimiento.[8]

Representa el corazón del sistema ya que es la encargada de enlazar las demás capas para su correcto funcionamiento. Recolecta la información que haya sido proporcionada por el usuario encapsulándola y enviándola a la capa de datos.

2.1.7. Capa de datos

En la capa de datos se emplea la estructura y almacenamiento de la información, es la encargada de proporcionar los datos cuando el usuario(s) lo requieran, esto se lleva a cabo mediante modelos que obtienen la información de la base de datos.

En esta capa se establecen las consultas a la base de datos, ya sea registrando u obteniendo información del sistema cuando un usuario lo requiera.

2.2. FUNDAMENTACIÓN TEÓRICA DEL PROTOTIPO

En el presente proyecto se emplean fuertes bases de fundamentación permitiendo obtener un software sólido y adaptable para la empresa. A continuación, se especifica los componentes y tecnologías que se aplican en el sistema.

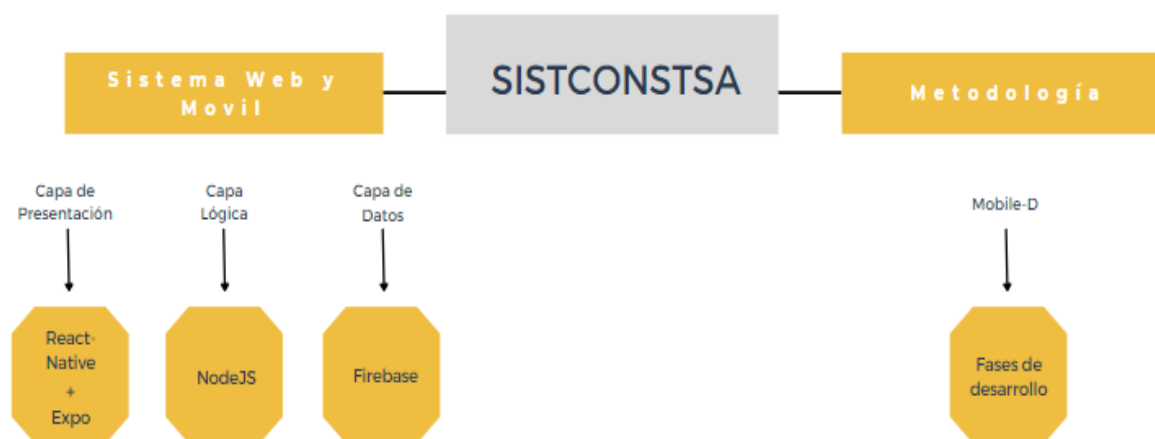


Figura 3: Base teórica

Fuente: Elaboración propia

En la **Figura 3** se observa cómo está distribuido el sistema en base a la fundamentación teórica. El aplicativo será desarrollado con una arquitectura de tres capas, usando React Native y Expo en el frontend, NodeJS para el manejo de información y firebase como almacenamiento de datos. Por otro lado, el sistema será desarrollado en base a la metodología móvil Mobile-D implementando sus fases de desarrollo en el proyecto.

2.2.1. Metodología de desarrollo de software

Las metodologías de desarrollo brindan un camino o guía de procesos a seguir para obtener un software de calidad; dentro de este enfoque estructurado se detallan los modelos, reglas, sugerencias de diseño y patrones que se deben llevar a cabo dentro del ciclo de vida del desarrollo del proyecto.[9]

2.2.1.1. Metodologías Tradicionales

Las metodologías tradicionales son normalmente conocidas por emplear un enfoque a la planificación, definiendo los detalles de los procesos y actividades a realizar; Este tipo de enfoque es más eficiente dentro de una organización de gran tamaño optimizando tiempo y recursos de la organización.[10][11]

Uno de los modelos más conocidos es el modelo de cascada el cual enfoca su desarrollo de forma secuencial y sistemática, empleando más énfasis en la disciplina de sus etapas que con: análisis de requisitos, diseño, codificación, pruebas y mantenimiento.

2.2.1.2. Metodologías Ágiles

Las metodologías ágiles son particularmente útiles para proyectos en los que los requisitos no se comprenden completamente desde el principio, es necesario crear nuevas tecnologías y el cliente desea participar de cerca en el proceso de desarrollo.[12]

Estas metodologías proponen una particularidad de flexibilidad durante el desarrollo del proyecto, subdividiendo el proyecto general en pequeños proyectos, dándole peso a la comunicación del equipo donde estos son altamente colaborativos y adaptables a los cambios.

2.2.1.3. Metodologías Móviles

Una de las claves dentro del desarrollo móvil es su corta duración, esto se debe a la feroz competencia en el campo y el desarrollo de nuevas características de software y hardware. Por lo general muchas aplicaciones nacen como un prototipo y con el tiempo van evolucionando, salvo algunas excepciones estas son las normas a seguir en el desarrollo móvil.[13][14]

2.2.1.4. Metodología Mobile-D

Mobile-D nació de la unión de distintos recursos ya conocidos donde se emplean los mismos principios de flexibilidad, centrándose en la funcionalidad del aplicativo en lugar de la documentación y elige la participación en forma directa por parte del cliente antes que una negociación.[15]

El enfoque de la metodología esta optimizado para un equipo de desarrollo de no más de diez personas que trabajan en un mismo espacio con el objetivo de entregar un aplicativo móvil completamente funcional en un corto periodo de tiempo.

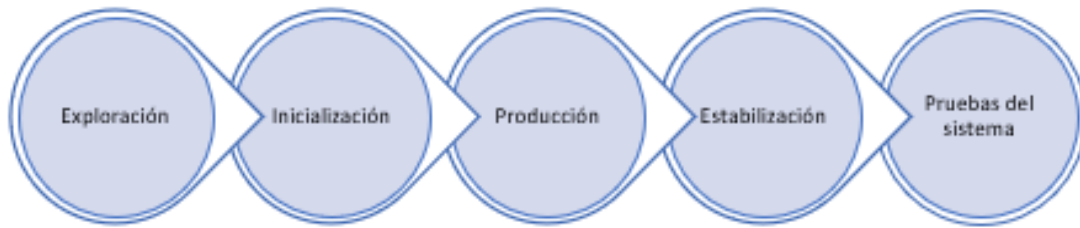


Figura 4: Ciclo de vida Mobile-D

Fuente: Elaboración propia

Como se observa en la **Figura 4**, las fases forman un proceso secuencial donde cada fase consta de diferentes etapas con el objetivo de conseguir ciclos de desarrollo rápidos. [16][17]

2.2.2. Servidor

Un servidor es una computadora o sistema que proporciona recursos, datos, servicios o programas a otras computadoras, conocidas como clientes, a través de una red. En teoría, siempre que las computadoras comparten recursos con las máquinas cliente, se consideran servidores. Hay muchos tipos de servidores, incluidos servidores web, servidores de correo y servidores virtuales.

Los servidores suelen tener una gran potencia informática y un gran espacio de almacenamiento. Como resultado, el servidor puede procesar y almacenar de manera compleja los datos del cliente, lo que reduce los costos de procesamiento y ahorra espacio en la memoria.[18]

2.2.2.1. Servidor Web

Un servidor web (también conocido como servidor HTTP) es un tipo de servidor que se utiliza para entregar (distribuir) contenido web a través de una intranet o Internet. En una red informática, un servidor web transfiere documentos a una computadora o cliente. Por ejemplo, una página web a un navegador.[19]

Es uno de los servidores más conocidos en el ambiente tecnológico. Un servidor web es un tipo especial de servidor de aplicaciones que aloja programas y datos solicitados por los usuarios a través de Internet o una intranet. Los servidores web responden a las solicitudes de los navegadores

que se ejecutan en los equipos cliente de páginas web u otros servicios basados en web. Los servidores web comunes incluyen servidores web Apache, servidores de Microsoft Internet Information Services (IIS) y servidores Nginx. [20]

2.2.2.2. NodeJS Framework

NodeJS es un framework de E/S asíncrono sin bloqueo basado en eventos que utiliza el motor JavaScript V8 de Google. Se utiliza para desarrollar aplicaciones JavaScript intensivas que se ejecutan tanto en el lado del cliente como en el lado del servidor. Como resultado, se beneficia de la reutilización del código y no del cambio de contexto.[21][22]

NodeJS es un entorno de ejecución multiplataforma de código abierto para desarrollar aplicaciones de red y del lado del servidor. Las aplicaciones se pueden ejecutar en OS X, Microsoft Windows y Linux. Las principales características de NodeJS son las siguientes:

- **Asíncrono y controlado por eventos:** Todas las API de NodeJS son asíncronas, es decir, sin bloqueo. Básicamente, significa que un servidor nunca espera a que una API devuelva datos. El servidor pasa a la siguiente API después de llamarlo y un mecanismo de notificación de Eventos ayuda al servidor a obtener una respuesta de la llamada API anterior.
- **Rapidez:** Al estar construido sobre el motor JavaScript V8 de Google Chrome, la biblioteca NodeJS es muy rápida en la ejecución de código. Permitiendo realizar simultaneas tareas sin complicidad.
- **Un solo subproceso, pero altamente escalable:** NodeJS utiliza un modelo de un solo subproceso con bucle de eventos. El mecanismo de eventos ayuda al servidor a responder sin bloqueos y hace que el servidor sea altamente escalable en comparación con los servidores tradicionales que crean subprocesos limitados para manejar solicitudes.
- **Sin almacenamiento en búfer:** Las aplicaciones NodeJS nunca almacenan en búfer ningún dato. Estas aplicaciones simplemente generan los datos en trozos.

2.2.3. Cliente

Un cliente es un proceso que permite a un usuario realizar una solicitud y reenviarla a un servidor, a esto se conoce como “frontend”.[23] Los clientes suelen manejar todas las funciones relacionadas con la manipulación y visualización de información, en base a esto suelen ser creados en plataformas que permiten crear interfaces gráficas de usuario (GUI) y acceder a servicios distribuidos en cualquier lugar de Internet.[24]

2.2.3.1. React-Native Framework

React-Native es un framework de JavaScript para escribir aplicaciones móviles reales y de representación nativa para iOS y Android, también está enfocado a la web. Se basa en React, la biblioteca JavaScript de Facebook para crear interfaces de usuario, pero en lugar de apuntar solo al navegador, apunta a plataformas móviles y navegadores.[25]

React-Native está construido para brindar al desarrollador la misma experiencia para aplicaciones web que ReactJS, sin paso de compilación y los cambios aparecen instantáneamente. Lo que se destaca de React-Native es que la mayoría de los otros frameworks móviles utilizan lenguajes de programación como Objective-C, Java, etc. Mientras tanto, React-Native le da al desarrollador web la oportunidad de trabajar fácilmente con aplicaciones móviles nativas.

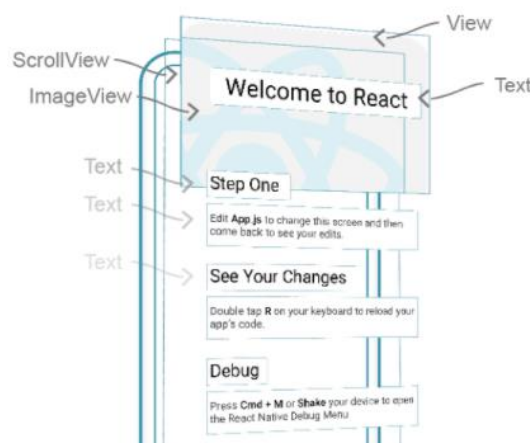


Figura 5: Componentes básicos de React-Native

Fuente: www.reactnative.dev

En la Figura 5 se observa parte de los componentes básicos de React Native para el diseño y creación de interfaces de pantalla para la interacción de los usuarios. Entre los cuales están “View”, “ScrollView”, “ImageView” y “Text”.

2.2.3.1.1. Nativo

Las aplicaciones nativas son aplicaciones que se ajustan para plataformas específicas y tienen acceso completo a la API que se comunica directamente con la unidad de software. Las aplicaciones nativas a menudo se mencionan en el desarrollo móvil porque las aplicaciones móviles a menudo se escriben para una plataforma en específico por lo general Android y iOS.

2.2.3.1.2. JavaScript/JSX

JavaScript es un lenguaje de programación muy común en el mundo del desarrollo web. En React-Native, el desarrollador puede optar por utilizar lenguaje JavaScript o la biblioteca JSX, pero React-Native recomienda a los desarrolladores que utilicen JSX.

JSX es JavaScript con una extensión de sintaxis que permite una mezcla entre JavaScript y etiquetas XML. JSX es un lenguaje de programación orientado a objetos que está diseñado para ejecutarse en navegadores web. Dado que React-Native deseaba que los desarrolladores que están acostumbrados a desarrollar para la web tuvieran la misma sensación al usar JSX. JSX optimiza el código para que se ejecute más rápido en comparación a que si estuviera escrito directamente en JavaScript.

2.2.3.2. Componentes de React-Native

Los componentes permiten separar la interfaz de usuario en piezas independientes, reutilizables y pensar en cada pieza de forma aislada. Conceptualmente, los componentes son como las funciones de JavaScript. Aceptan entradas arbitrarias (llamadas “props”) y devuelven a React-Native elementos que describen lo que debe aparecer en la pantalla. Entre los principales componentes de React-Native están los siguientes:

- **View:** View es el componente más fundamental en React-Native ya que es soportado por las propiedades de flexbox y el estilo del layout. Este

componente determina como la pantalla de la aplicación debería ser mostrada. View es el equivalente a un div en la Web.

- **ListView:** Este componente que es diseñado para mostrar información en formar vertical, posee la propiedad de refrescar cuando su data cambia.
- **Navigator:** Este componente brinda la posibilidad de realizar un seguimiento del lugar de la aplicación en la que el usuario se encuentra, de manera que es más fácil pasar de una vista a otra.
- **Modal:** Este componente brinda de manera básica la posibilidad de poder mostrar contenido en una vista flotante a la vista principal.

2.2.3.3. Expo Framework

Expo es un framework y una plataforma para aplicaciones universales de React. Es un conjunto de herramientas y servicios creados alrededor de React-Native y plataformas nativas que lo ayudan a desarrollar, construir, implementar e iterar rápidamente en iOS, Android y aplicaciones web desde la misma base de código JavaScript / TypeScript.

Expo hace uso de su propio SDK el cual es una librería nativa y de JavaScript que da acceso a funcionalidades básicas de los dispositivos como uso de cámara, contactos, galería, entre otros, evitando codificar en lenguaje nativo para cada plataforma.

2.2.4. Jest

Jest es un marco de prueba de JavaScript diseñado para garantizar la corrección de cualquier base de código JavaScript. Le permite escribir pruebas con una API accesible, familiar y rica en funciones que le brinda resultados rápidamente.[26]

2.2.5. Cloud computing

Es un modelo para permitir un acceso de red conveniente y bajo demanda a un grupo compartido de recursos informáticos configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que se pueden aprovisionar y liberar rápidamente con un mínimo esfuerzo de administración o interacción del proveedor de servicios.[27][28]

2.2.5.1. Firebase

Firestore es una plataforma de desarrollo conocida originalmente por su base de datos en tiempo real optimizada para sincronizar datos, a menudo entre las máquinas de los usuarios o los teléfonos inteligentes y el almacenamiento centralizado en la nube. Está diseñado para hacer la vida más fácil a los desarrolladores al manejar gran parte del empuje y extracción de datos.[29]

2.2.5.2. Base de datos Firebase en tiempo real

Firestore Realtime Database es una base de datos NoSQL basada en la nube que sincroniza datos entre todos los clientes en tiempo real y proporciona funcionalidad sin conexión. Los datos se almacenan en la base de datos en tiempo real como JSON, y todos los clientes conectados comparten una instancia y reciben automáticamente actualizaciones con los datos más recientes.[30]

Firestore Realtime Database permite crear aplicaciones enriquecidas al permitir el acceso seguro a la base de datos directamente desde el código del lado del cliente. Los datos se conservan localmente, e incluso cuando están fuera de línea, los eventos en tiempo real continúan activándose, lo que brinda al usuario final una experiencia receptiva. Cuando el dispositivo recupera la conexión, Firestore Realtime Database sincroniza los cambios de datos locales con las actualizaciones remotas que ocurrieron mientras el cliente estaba fuera de línea, fusionando cualquier conflicto automáticamente.

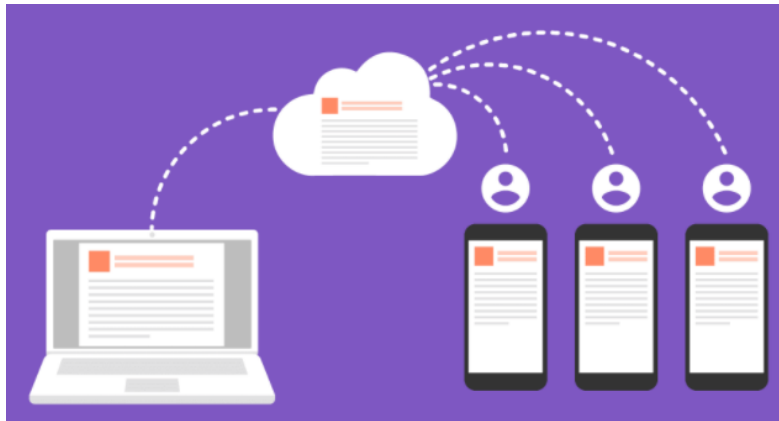


Figura 6: Sincronización de datos en tiempo real con diferentes dispositivos

Fuente: www.firebase.google.com

2.2.6. Herramientas CASE

La herramienta CASE cubre cada etapa del proceso de software y las operaciones generales que se aplican a todo el proceso. CASE es una combinación de bloques de construcción que comienzan en el nivel de hardware y software del sistema operativo y terminan con herramientas individuales.[31]

2.2.6.1. Microsoft Visio

Microsoft Visio es uno de los softwares más populares que permite la creación de diagramas, la visualización de datos y el modelado de procesos en una interfaz familiar. Visio viene con una variedad de plantillas y formas integradas que permiten crear prácticamente cualquier diagrama de cualquier complejidad.[32]

2.2.6.2. Microsoft Project

Project es una potente aplicación que permite planificar y gestionar una amplia gama de proyectos. Desde cumplir con plazos y presupuestos cruciales hasta seleccionar los recursos adecuados, permite aumentar la producción y obtener mejores resultados utilizando el conjunto de funciones que ofrece.[33]

2.2.6.3. Enterprise Architect

Enterprise Architect es una herramienta integral de análisis y diseño UML, que cubre todos los aspectos del ciclo de desarrollo de software, desde la recopilación de requisitos, pasando por el análisis, el diseño de modelos, las pruebas, el control de cambios y el mantenimiento hasta la implementación, con total trazabilidad.[34]

2.2.6.4. Balsamiq Mockups

Balsamiq Mockups es una herramienta de gran ayuda en la creación de mockups, facilitando la creación de bocetos o diseños que sean necesarios dentro del proceso de desarrollo de software.[35]

2.2.6.5. StarUML

StarUML es una plataforma de modelado de software que admite UML (Lenguaje de modelado unificado). Está basado en la versión 1.4 de UML y proporciona once tipos diferentes de diagrama, y acepta la notación UML 2.0. Apoya activamente el enfoque MDA (Model Driven Architecture) al admitir el concepto de perfil UML.[36]

2.3. OBJETIVOS DEL PROTOTIPO

2.3.1. Objetivo General

Desarrollar un aplicativo web y móvil para los procesos de gestión de empleados de una empresa constructora mediante el uso de la metodología Mobile-D y el framework React-Native, optimizando los procesos de control y asignación de actividades a empleados.

2.3.2. Objetivos Específicos

- Aplicar las fases de la metodología Mobile-D para establecer un marco de trabajo.
- Definir los requerimientos funcionales y no funcionales mediante reuniones de las partes interesadas.
- Implementar prototipos de diseños del funcionamiento del sistema haciendo uso de las herramientas CASE.
- Validar pruebas de funcionalidad de la aplicación web y móvil.

2.4. DISEÑO DEL PROTOTIPO

El diseño de la aplicación web y móvil está fundamentado en la metodología Mobile-D, por lo que a continuación se detallará las fases y etapas del ciclo de vida de la misma por medio de la planificación y el modelado del aplicativo.

2.4.1. Fase de exploración

Dentro de esta etapa se lleva a cabo un previo análisis de lo necesario para la inicialización del sistema, concerniente a lo más importante dentro del desarrollo de un software.[37]

2.4.1.1. Alcance del proyecto

En esta fase se detallan las funcionalidades del sistema, especificando de manera resumida las actividades que se deben realizar de menor a mayor prioridad, permitiendo obtener lo esencial para cumplir con el objetivo del proyecto.

Tabla 1: Requisitos funcionales del sistema

Requisitos Funcionales del Sistema	Permitir el ingreso al sistema con credenciales mediante un Login.
	Permitir la gestión de obras y asociación de los empleados.
	Permitir la gestión de los empleados.
	Permitir la asignación de actividades a los empleados.
	Permitir la generación de reportes sobre empleados y obras.
	Permitir generar el pago a un empleado.

Fuente: Elaboración propia

En la **Tabla 1** se observa una descripción general de los requisitos funcionales del sistema a implementar.

Tabla 2: Requerimientos no funcionales del sistema

Limitaciones Funcionales del Sistema	Se carece de un super usuario para el manejo de todo el sistema.
	Las actividades ya vienen establecidas, no se podrá crear nuevas actividades.
	Los empleados solo podrán ser registrados por un administrador.
	Mantener seguro datos sensibles de los usuarios.

Fuente: Elaboración propia

En la **Tabla 2** se detallan las limitaciones que posee el sistema, las mismas que podrán ser implementadas en futuras actualizaciones del sistema si así se requiere.

2.4.1.2. Análisis de factibilidad

El análisis de factibilidad es determinar mediante un estudio que tan viable es el desarrollo del proyecto a implementar y si se cuenta con la infraestructura y herramientas para llevarlo a cabo.

En el presente análisis de factibilidad se pretende determinar que tan realizable resulta la implementación del aplicativo web y móvil para la gestión de empleados, en base a las necesidades de la empresa.

2.4.1.2.1. Factibilidad operativa

Dentro de esta fase se realiza un análisis de los recursos productivos y necesarios que deben ser implementados para la funcionalidad total del proyecto. Es decir, verificar que existan las condiciones necesarias para llevar a cabo el desarrollo del proyecto.

Tabla 3: Factibilidad operativa

Factibilidad operativa

No.	Actividad	Priorización		
		Alta	Media	Baja

1	Compra de un plan en Firebase para el uso de servicios en la plataforma.	X		
2	Se cuenta con los conocimientos para el desarrollo del sistema.	X		
3	Necesidad de implementar capacitaciones sobre funcionamiento del sistema.		X	
4	El sistema sigue estrictamente los reglamentos de la empresa.	X		

Fuente: Elaboración propia

En la **Tabla 3** se observa las actividades que han sido consideradas como una parte fundamental para llevar a cabo la implementación del sistema, determinando su grado de priorización.

2.4.1.2.2. Factibilidad técnica

En este apartado se detallan los recursos para el funcionamiento y ejecución del sistema. Se han analizado los recursos tanto software como hardware que influyen en el proyecto de manera directa, a la vez estos permiten efectuar las actividades o procesos que requiera el proyecto.

Tabla 4: Factibilidad técnica

Factibilidad técnica			
Recursos hardware		Recursos software	
Cantidad	Descripción	Cantidad	Descripción
1	HDD de 4TB	1	Firestore
		1	Dominio
		1	IP pública
		2	Frameworks: React-Native, NodeJS

Fuente: Elaboración propia

En la **Tabla 4** se presentan los recursos tanto hardware como software que son necesarios para el proyecto. Se observa que en mayor cantidad son los recursos de software, y esto se debe ya que la arquitectura del proyecto está basada en la nube.

2.4.1.2.3. Factibilidad económica

Se establece el análisis de los recursos económicos que implica la realización del proyecto. En este caso se tomará en cuenta el valor de los equipos tecnológicos que se requieran. Cabe mencionar que el desarrollo del sistema conlleva un gasto a nivel personal, por lo que no se consideró el costo de codificación y pruebas de funcionalidad, ya que forman parte del trabajo de titulación.

Tabla 5: Factibilidad económica

Factibilidad Económica					
Tipo / Concepto	Parámetro / Descripción	Cantidad	Unid.Med.	P.Unitario (\$)	Total (\$)
Software					\$0.00
Suite Ofimática	Elaboración de documentación e informe.	1	U	\$0.00	\$0.00
Balsamiq Mockups	Diseño de los prototipos de interfaces de usuario.	1	U	\$0.00	\$0.00
Visual Studio Code	Codificación de la aplicación.	1	U	\$0.00	\$0.00
Nox Emulador	Emulador de aplicaciones Android.	1	U	\$0.00	\$0.00
Hardware					\$898.00
Laptop	Windows 10, Intel corei7, 1TB disco duro, 16GB RAM	1	U	\$890.00	\$890.00
Unidad de Almacenamiento	Almacenamiento de datos, registros, plantillas, entre otros archivos necesarios.	1	U	\$8.00	\$8.00
Dispositivo móvil	Dispositivo móvil para realización de pruebas.	1	U	\$0.00	\$0.00

Factibilidad Económica					
Tipo / Concepto	Parámetro / Descripción	Cantidad	Unid.Med.	P.Unitario (\$)	Total (\$)
Total					\$898.00

Fuente: Elaboración propia

En la **Tabla 5** se describe los recursos económicos que han sido considerados para la puesta en marcha del proyecto, obteniendo un total de \$898.00 que es el valor del equipo tecnológico que ya se posee.

2.4.1.3. Identificación de interesados o “stakeholders”

Los stakeholders son todos los implicados ya sea de forma directa o indirecta en la aprobación y toma de decisiones en la ejecución de un proyecto, permitiendo el desarrollo operacional del sistema a implementar.[38]

Tabla 6: Matriz de Interesados

Nombre	Rol en el Proyecto	Profesión	Responsabilidad
Ing. Oscar Cárdenas	Docente / Gerente de proyecto	Ing. De Sistemas	Funcionalidad del sistema y revisión de entregables
Paulo Ramírez	Desarrollador	Estudiante	Planificación, diseño, implementación y pruebas
CONSTSA	Cliente	Empresa constructora	Revisión de entregables y funcionalidad del sistema

Fuente: Elaboración propia

En la **Tabla 6** se detalla la matriz de interesados conformada por el Ing. Oscar Cárdenas Gerente del Proyecto, la empresa Constsa quien es el beneficiario del proyecto, y como último mi persona, encargada del desarrollo e implementación del sistema.

2.4.1.4. Requerimientos

Los requisitos son una especificación de lo que se debe implementar. Son descripciones de cómo debería comportarse el sistema, o de una propiedad o

atributo del sistema. Pueden ser una restricción en el proceso de desarrollo del sistema.[39]

2.4.1.5. Requerimientos Funcionales

Describen lo que los desarrolladores deben implementar para permitir que los usuarios realicen sus tareas (requisitos del usuario), satisfaciendo así los requisitos comerciales. Esta alineación entre los tres niveles de requisitos es esencial para el éxito del proyecto.

Tabla 7: Requerimientos Funcionales

REQUERIMIENTOS FUNCIONALES (RF)	
CÓDIGO	DESCRIPCIÓN
RF-01	Página de bienvenida permitiendo la autenticación de usuarios.
RF-02	Gestión de obras (administrador)
RF-03	Gestión de empleados (administrador)
RF-04	Visualización de perfil de usuario
RF-05	Visualización de actividades
RF-06	Registro de asistencia de obreros
RF-07	Asignación de actividades a obreros (administrador)
RF-08	Registro de pago a obreros
RF-09	Generación de reporte de obras
RF-10	Generación de reporte de pagos a empleados

Fuente: Elaboración propia

En la **Tabla 7** se detallan los requerimientos funcionales del sistema necesarios para la interacción de los usuarios y su correcto funcionamiento. Los mismos que han sido analizados y recolectados en base a reuniones con las partes interesadas.

2.4.1.6. Requerimientos No Funcionales

Describen características o propiedades importantes del sistema. Estos incluyen la disponibilidad, usabilidad, seguridad, rendimiento y muchas otras características del sistema.[40]

Tabla 8: Requerimientos No Funcionales

REQUERIMIENTOS NO FUNCIONALES (RNF)	
CÓDIGO	DESCRIPCIÓN
RNF-01	El sistema debe ser usable y de fácil manejo para usuarios con poca experiencia.
RNF-02	Debe cargar la información de manera rápida a todos los usuarios que estén conectados.
RNF-03	El sistema debe ser para web y Android, aun con smartphones de baja gama.
RNF-04	La información de cada usuario debe estar cifrada y segura.
RNF-05	Su estructura de base de datos no relacional debe estar bien definida.

Fuente: Elaboración propia

En la **Tabla 8** se detallan los requerimientos no funcionales con los que debe contar el sistema en su implementación.

2.4.2. Fase de inicialización

Dentro de esta fase se realizan las historias de usuario y el cronograma de actividades, en este caso por iteraciones, con el fin de analizar los requerimientos de las partes interesadas del proyecto, permitiendo así crear un proyecto viable que cumpla con las necesidades planteadas con anterioridad.

2.4.2.1. Historias de usuario

Una historia de usuario es una descripción breve y simple de una característica contada desde la perspectiva del actor que desea la nueva funcionalidad, generalmente un usuario o cliente del sistema.[41]

Tabla 9: Historia de usuario N.-1

Historia de usuario	
Número: 1	Usuarios: Administrador y obrero
Nombre de historia: Ingreso al sistema	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta

Requerimiento funcional: RF-01	Iteración asignada: 1
Desarrollador responsable: Paulo César Ramírez Novillo	
Descripción: Los dos tipos de usuarios deben ingresar al sistema mediante su correo electrónico y contraseña; una vez correcto al acceso se debe mostrar la pantalla principal de sistema.	
Observaciones: Se deberá mostrar un mensaje de alerta en caso de que las credenciales sean incorrectas.	

Fuente: Elaboración propia

El aplicativo debe permitir la autenticación de los usuarios con sus respectivas credenciales, si la acción es correcta se mostrará la pantalla principal con las opciones de menú según el rol del usuario.

Tabla 10: Historia de usuario N.-2

Historia de usuario	
Número: 2	Usuarios: Administrador
Nombre de historia: Gestión de obras	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Requerimiento funcional: RF-02	Iteración asignada: 1
Desarrollador responsable: Paulo César Ramírez Novillo	
Descripción: El usuario, en este caso administrador podrá realizar las operaciones CRUD de las obras que presente la empresa.	
Observaciones: Al momento de registrar una obra se deberá verificar que la fecha de inicio no sea menos que la fecha actual del registro.	

Fuente: Elaboración propia

El administrador podrá realizar el registro, modificación y eliminación de una obra dentro del sistema, a su vez no podrá ingresar dentro de la fecha de inicio de la obra un valor menor a la fecha actual en que se dé el registro. El administrador deberá asignar a un responsable de la obra.

Tabla 11: Historia de usuario N.-3

Historia de usuario	
Número: 3	Usuarios: Administrador
Nombre de historia: Gestión de empleados	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Requerimiento funcional: RF-03	Iteración asignada: 1
Desarrollador responsable: Paulo César Ramírez Novillo	
Descripción: El usuario, en este caso administrador podrá realizar las operaciones CRUD de los empleados que presente la empresa.	
Observaciones: Al momento de registrar un empleado obrero se deberá almacenar la información bancaria, con el fin de poder realizar los pagos.	

Fuente: Elaboración propia

El usuario administrador tendrá el privilegio de poder registrar, modificar y eliminar a los diferentes empleados dentro de la empresa. Es necesario poder registrar la información bancaria de los empleados de tipo obrero para poder realizar los pagos de los días trabajados.

Tabla 12: Historia de usuario N.-4

Historia de usuario	
Número: 4	Usuarios: Administrador y Obrero
Nombre de historia: Visualización de perfil de usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Requerimiento funcional: RF-04	Iteración asignada: 1
Desarrollador responsable: Paulo César Ramírez Novillo	

Descripción: Los diferentes usuarios tendrán la posibilidad de visualizar su perfil con toda su información correspondiente.

Observaciones: Los usuarios podrán modificar su información en caso de ser necesario.

Fuente: Elaboración propia

Los usuarios de los diferentes tipos tienen la opción de menú para visualizar su perfil con toda su información correspondiente. A la vez los usuarios tendrán la posibilidad de modificar su información si así se requiera.

Tabla 13: Historia de usuario N.-5

Historia de usuario	
Número: 5	Usuarios: Administrador y Obrero
Nombre de historia: Visualización de actividades	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Requerimiento funcional: RF-05	Iteración asignada: 1
Desarrollador responsable: Paulo César Ramírez Novillo	
Descripción: Los administradores podrán visualizar las actividades de las obras de los cuales son responsables; por otro lado, los obreros solo podrán visualizar las actividades de las obras a las que fueron asignados.	
Observaciones: Las actividades se deberán mostrar en orden cronológico con sus respectivas fechas.	

Fuente: Elaboración propia

Los usuarios deben tener acceso a las actividades de las obras, con el fin de poder organizarse según el cronograma programado y a la vez observar el tiempo de ejecución para cada actividad a realizar.

Tabla 14: Historia de usuario N.- 6

Historia de usuario	
Número: 6	Usuarios: Administrador
Nombre de historia: Registro de asistencia de obreros	

Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Requerimiento funcional: RF-06	Iteración asignada: 1
Desarrollador responsable: Paulo César Ramírez Novillo	
Descripción: Los administradores deberán realizar el registro de asistencia de los obreros según las obras a las que han sido asignados, esto de manera diaria con el fin de poder obtener la cantidad de días trabajados por cada obrero.	
Observaciones: La asistencia no podrá registrarse fuera del horario laboral.	

Fuente: Elaboración propia

Una de las cosas fundamentales es el poder tener a disposición un registro de la asistencia de los obreros en base a cada obra, con el fin de poder llevar un control sobre los obreros que han laborado de forma constante a lo largo de la obra, además esto beneficia en la obtención de los días trabajados en la obra finalizada con el propósito de realizar el pago a cada obrero.

Tabla 15: Historia de usuario N.-7

Historia de usuario	
Número: 7	Usuarios: Administrador
Nombre de historia: Asignación de actividades a obreros	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Requerimiento funcional: RF-07	Iteración asignada: 1
Desarrollador responsable: Paulo César Ramírez Novillo	
Descripción: Los administradores podrán realizar la asignación de actividades a los diferentes obreros según la obra a la que estén asignados.	
Observaciones: Las actividades son creadas al registrar una obra, no se pueden añadir nuevas actividades.	

Fuente: Elaboración propia

La asignación de actividades permite a los administradores llevar un control sobre el progreso y cumplimiento de las tareas en el día a día por parte de los obreros en las diferentes obras de la empresa. Las actividades para cada obra

son generadas al momento de registrar dicha obra, no se podrán asignar nuevas actividades durante la duración de la obra.

Tabla 16: Historia de usuario N.-8

Historia de usuario	
Número: 8	Usuarios: Administrador
Nombre de historia: Registro de pagos a obreros	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Requerimiento funcional: RF-08	Iteración asignada: 1
Desarrollador responsable: Paulo César Ramírez Novillo	
Descripción: Los administradores podrán realizar al finalizar la obra el pago a los diferentes empleados, en base a los días laborados.	
Observaciones: El registro de pago a obreros por parte de los administradores no podrá ser ejecutado fuera del horario laboral.	

Fuente: Elaboración propia

Los registros de pagos a obreros son realizados una vez finalizada la obra a la que han sido asignados, los mismos que deberán ser generados por parte del administrador o encargado de la obra. Estos se generan en base a los días que ha laborado cada obrero, y no podrán ser generados fuera del horario laboral.

Tabla 17: Historia de usuario N.-9

Historia de usuario	
Número: 9	Usuarios: Administrador
Nombre de historia: Generación de reporte de obras	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Requerimiento funcional: RF-09	Iteración asignada: 1
Desarrollador responsable: Paulo César Ramírez Novillo	

Descripción: Los administradores tendrán la capacidad de poder generar un reporte de la(s) obra(s) que se requieran.

Observaciones: Deberá tener la posibilidad de generar reporte en base a fechas.

Fuente: Elaboración propia

Los reportes de obra cumplen un papel fundamental dentro del sistema, pues estos reflejan el estado de las obras que han sido registradas en el sistema. Se podrá obtener una información detalladas de las obras que se requieran pudiendo generar reportes por fechas, observando el progreso de las obras como las actividades que se han cumplido.

Tabla 18: Historia de usuario N.-10

Historia de usuario	
Número: 10	Usuarios: Administrador
Nombre de historia: Generación de reporte de pagos a obreros	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Requerimiento funcional: RF-10	Iteración asignada: 1
Desarrollador responsable: Paulo César Ramírez Novillo	
Descripción: Los administradores tendrán la capacidad de poder generar reportes de los pagos sobre el o los obreros.	
Observaciones: Deberá tener la posibilidad de generar reportes en base a fechas.	

Fuente: Elaboración propia

Los reportes de pagos a obreros brindan a la empresa la posibilidad de almacenar un registro de pagos a los obreros en base a las obras que se han realizado. A su vez permiten a la empresa llevar un control de los gastos generados por obreros. El reporte deberá ser generado en base a fechas.

2.4.2.2. Gestión de Cronograma

En el cronograma se detallan las actividades que deben ser ejecutas para llevar a cabo el proceso de desarrollo de aplicativo web y móvil, aquí se detallan las actividades para la recolección de requisitos, procesos para

análisis de factibilidad, prototipos de interfaces y las pruebas de funcionalidad del sistema.

A continuación, se presenta el cronograma de actividades con sus respectivas fechas de ejecución, el cronograma se presenta mediante un diagrama de Gantt:

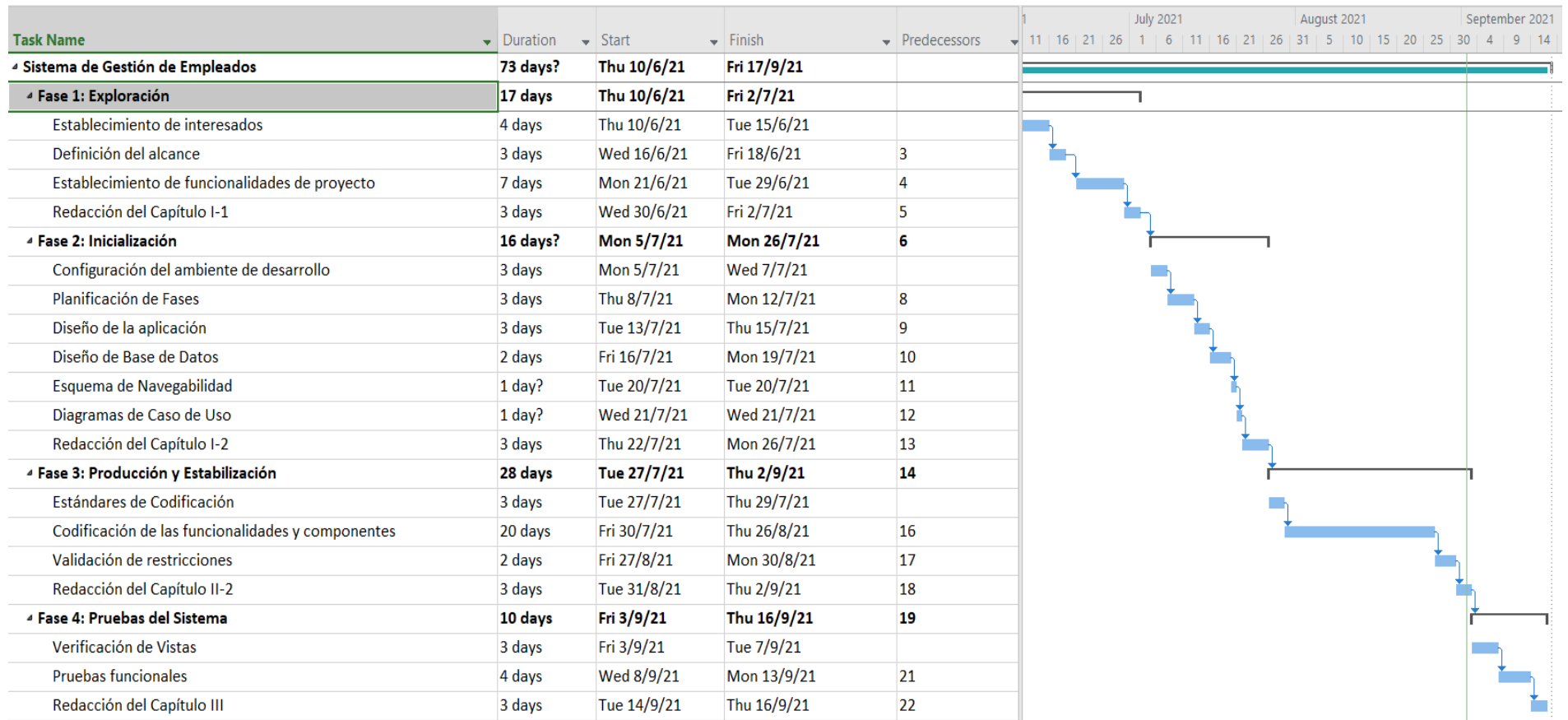


Figura 7: Cronograma de actividades

Fuente: Elaboración propia

Como se observa en la **Figura 7** las fases de la metodología Mobile-D se dan de manera secuencial, estableciendo un tiempo de duración de 73 días para la finalización del desarrollo del aplicativo web y móvil, sin tomar en consideración los tiempos de sustentación.

2.4.2.3. Configuración del Ambiente del Desarrollo

En este apartado que es orientado netamente al desarrollador del aplicativo, ya que su función es de crear un ambiente idóneo para llevar a cabo el desarrollo.

- Tipo de Proyecto: Aplicación Híbrida
- Framework para la aplicación móvil: React-Native, Expo
- Framework para la aplicación web: React-Native, Expo

Para iniciar con la preparación del ambiente de desarrollo es necesario la instalación de las siguientes herramientas: Visual Studio Code, React-Native, NodeJS, Expo, Firebase, Android Studio, Postman.

2.4.2.3.1. Instalación de Visual Studio Code

Visual Studio Code es un editor de texto propiedad de Microsoft donde permite el desarrollo en diferentes lenguajes de programación, así como de base de datos. Nos permitirá crear el código para el aplicativo web y móvil y a la vez para el desarrollo de la API. Visual Studio Code posee terminales integradas, facilitando la ejecución de comandos dentro del proyecto.

2.4.2.3.2. Instalación de Android Studio

Android Studio permite desarrollar aplicaciones Android de manera nativa haciendo uso del lenguaje Java o Kotlin. Android Studio viene incluido con distintas herramientas como un IDE, las herramientas SDK y las APIs.

La instalación de Android Studio permitirá poder desarrollar aplicaciones híbridas mediante el uso de React-Native, ya que basa su funcionamiento en las herramientas SDK permitiendo escribir código JavaScript en React-Native para luego ser convertido a lenguaje Java para generar el apk final.

2.4.2.3.3. Instalación de React Native

React Native está basado en librerías de JavaScript por lo que permite desarrollar interfaces de usuario para la interacción entre el usuario y el

sistema. React Native permite el desarrollo híbrido o multiplataforma, generando a partir de un mismo código las interfaces para Android, iOS y Web, React Native está elaborado con los componentes de ReactJS por lo que permite su ejecución en la Web. Permite el desarrollo con JavaScript por lo que es un lenguaje interpretado permitiendo observar los cambios en tiempo de ejecución.

El beneficio de este Framework es el poder optimizar el tiempo de desarrollo, limitando a un solo proceso de programación del sistema y obteniendo como resultado un aplicativo móvil y web.

2.4.2.3.4. Instalación de NodeJS

NodeJS es un framework basado en JavaScript orientado para el desarrollo backend de un aplicativo. En este caso se hará uso de NodeJS para la creación de la API que será la encargada de interactuar con la base de datos en los requerimientos o solicitudes de información por parte de los clientes. Esto brindará seguridad al sistema distribuyendo los procesos de información con la API.

2.4.2.3.5. Instalación de Expo

Expo es un framework para React Native donde simplifica el uso de los componentes y API para el desarrollo de aplicaciones en Android, iOS y Web. Expo brinda la posibilidad de poder crear, ejecutar y publicar un proyecto de React Native mediante su plataforma. Expo ha creado distintos componentes para los usuarios donde permite desarrollar las aplicaciones híbridas de forma más nativa, y simplificando líneas de código en la programación. Adicional Expo cuenta con un entorno de ejecución en la nube donde se puede crear y ejecutar componentes y ser ejecutados Android, iOS y Web.

2.4.2.3.6. Instalación de Firebase

La instalación de los servicios de Firebase permitirá hacer uso de sus componentes y APIs en la nube. Firebase cumple un papel fundamental dentro del proyecto ya que la estructura del aplicativo está basada en sus servicios, haciendo uso del almacenamiento en la nube, autenticación de usuario,

ejecución de funciones y base de datos en tiempo real. Firebase interactúa con la API desarrollada con NodeJS para el manejo de la información.

2.4.2.3.7. Instalación de Postman

Postman es una herramienta gratuita que nos permite realizar peticiones de tipo http (GET, PUT, UPDATE, DELETE, etc) de forma local, con el fin de verificar si su ejecución y resultado es el correcto. Será de vital uso dentro del proyecto ya que permitirá realizar pruebas de funcionamiento de la API desarrollada con NodeJS, poniendo a prueba la ejecución de peticiones, así como el resultado obtenido de las mismas.

2.4.2.4. Diseño de la Aplicación

El sistema cuenta con una infraestructura en la nube, realizando peticiones al servidor a través de internet, pudiendo ser visualizada desde la web o móvil.

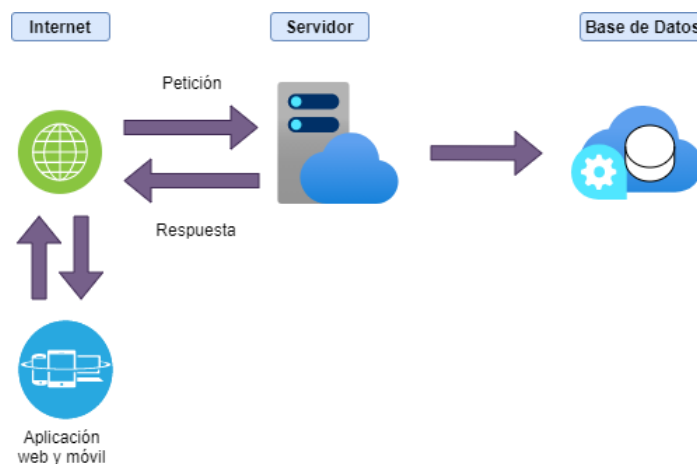


Figura 8: Diseño general del sistema

Fuente: Elaboración propia

En la **Figura 8** se detalla el diseño del funcionamiento del aplicativo, basando su estructura en la nube con la ayuda de los servicios de Firebase. El aplicativo interactúa con los distintos clientes, ya sean dispositivos móviles o navegadores web, mediante el uso de Internet, el cliente realizará la petición al servidor en la nube y este a su vez interactúa con la base de datos que está en la nube.

2.4.2.5. Diseño de Base de Datos

Para este proyecto se emplea la base de datos en la nube de Firebase, una base de datos NoSQL donde se almacena la información en formato JSON. Permitiendo el acceso en tiempo real a los distintos usuarios a través de múltiples dispositivos, brindando rapidez en la obtención de información, a su vez permite la sincronización en modo offline, una vez que se recupere la conexión se actualizará la información al instante.

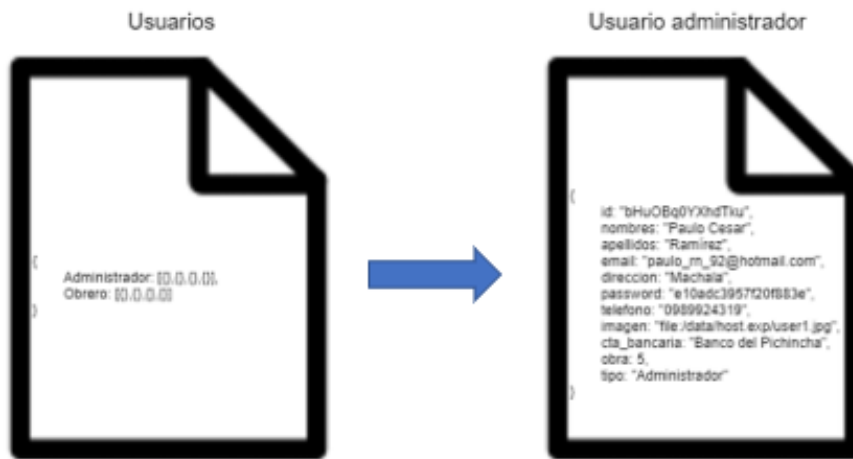


Figura 9: Estructura no relacional de Usuarios

Fuente: Elaboración propia

En la **Figura 9** se muestra la estructura no relacional que posee los usuarios, la estructura es de tipo árbol en formato JSON, siendo Usuarios la raíz principal y Administrador y Obrero los nodos de la raíz, a su vez estos nodos contienen objetos donde se almacena la información de cada usuario, registrando información como nombres, apellidos, teléfono, email, entre otros.

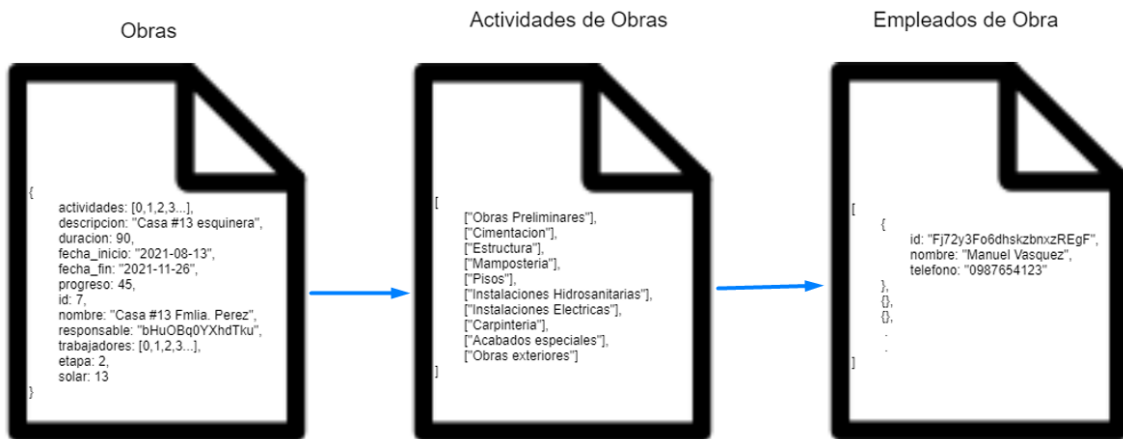


Figura 10: Estructura no relacional de Obras

Fuente: Elaboración propia

En la **Figura 10** se observa la estructura que poseen las obras, cada obra tiene un identificador y su respectiva información, siendo una de los más importantes el nodo actividades, este posee un arreglo de objetos donde están registradas las actividades de dicha obra, además de las fechas y tiempos de ejecución de cada proceso. Otro nodo es el de empleados donde este es un arreglo de objetos que contiene la información de cada empleado de la obra.



Figura 11: Estructura no relacional de asistencia de obreros

Fuente: Elaboración propia

En la **Figura 11** se observa la estructura que posee el registro de asistencia de obreros, el cual posee un identificador, la fecha en que se hace el registro, el identificador del obrero, el identificador de la obra y el responsable de realizar el registro.

Pagos

```
{
  [
    {
      id: 5,
      id_obra: "bFghy0SalowxY",
      días_laborados: 90,
      monto: 2500,
      responsable: "esFrd0lkOpasdFR"
    },
    [
      [
        [
          .
        ]
      ]
    ]
  ]
}
```

Figura 12: Estructura no relacional de pagos a obreros

Fuente: Elaboración propia

Dentro de la **Figura 12** se observa la estructura que posee el registro de pagos a obreros, contando con un identificador, los días laborados, el identificador del obrero, el monto y como ultimo campo el responsable de realizar el registro.

```

{
  "obras" : {
    "7" : {
      "actividades" : [ {
        "Obras Preliminares" : [ {
          "Descapote" : {
            "Localizacion y Replanteo" : {
              "fechas" : {
                "fin" : "16-08-2021",
                "inicio" : "14-08-2021"
              }
            }
          }
        }, {
          "Cimentacion" : [ {
            "Zapatas" : {
              "Vigas de cimentacion" : {
                "Concreto ciclopeo" : {
                  "fechas" : {
                    "fin" : "24-08-2021",
                    "inicio" : "16-08-2021"
                  }
                }
              }
            }
          }, {
            "Estructura" : [ {
              "Columna" : {
                "Viga Aerea" : {
                  "Placa losa entrepiso" : {
                    "Escaleras" : {
                      "Cubierta" : {
                        "fechas" : {
                          "fin" : "17-09-2021",
                          "inicio" : "24-08-2021"
                        }
                      }
                    }
                  }
                }, {
                  "Mamposteria" : [ {
                    "Mamposteria" : {
                      "fechas" : {
                        "fin" : "29-09-2021",
                        "inicio" : "17-09-2021"
                      }
                    }
                  }
                }
              }
            }
          }, {
            "Pisos" : [ {

```

Figura 13: Diseño de Base de Datos NoSQL 1

Fuente: Elaboración propia

En la **Figura 13** se observa la estructura que posee la base de datos, es de tipo árbol en formato JSON, podemos notar parte de la estructura de las obras, observando las actividades de dicha obra.

```

    "Pisos" : {
  }, {
    "fechas" : {
      "fin" : "11-10-2021",
      "inicio" : "29-09-2021"
    }
  }
], {
  "Instalaciones Hidrosanitarias" : [ {
    "Instalaciones Hidrosanitarias" : {
  }, {
    "fechas" : {
      "fin" : "19-10-2021",
      "inicio" : "11-10-2021"
    }
  }
]
}, {
  "Instalaciones Electricas" : [ {
    "Instalaciones Electricas" : {
  }, {
    "fechas" : {
      "fin" : "26-10-2021",
      "inicio" : "19-10-2021"
    }
  }
]
}, {
  "Carpinteria" : [ {
    "Carpinteria" : {
  }, {
    "fechas" : {
      "fin" : "06-11-2021",
      "inicio" : "26-10-2021"
    }
  }
]
}, {
  "Acabados especiales" : [ {
    "Acabados especiales" : {
  }, {
    "fechas" : {
      "fin" : "18-11-2021",
      "inicio" : "06-11-2021"
    }
  }
]
}, {
  "Obras exteriores" : [ {
    "Obras exteriores" : {
  }, {
    "fechas" : {
      "fin" : "26-11-2021",
      "inicio" : "18-11-2021"
    }
  }
]
} ],
"descripcion" : "Casa # 15 Etapa 1",
"duracion" : 90,

```

Figura 14: Diseño de Base de Datos NoSQL 2

Fuente: Elaboración propia

En la **Figura 14** se observa la continuación de la estructura de obra, se puede visualizar parte de las actividades de la obra con sus respectivas fechas.

```

"fechaFin": "2021-11-24",
"fechaInicio": "2021-09-17",
"id": 1,
"nombre": "Casa Solar 15",
"progreso": 0,
"responsable": "thub8qYUhdTvd8fcCTHafjpac",
"trabajadores": [
  {
    "id": "Fy72yIFo6dhaicIhns8EgFq712",
    "nom": "Riguel Carrón",
    "telefono": "0997654321"
  },
  {
    "id": "Fy72yIFo6dhaicIhns8EgFq712",
    "nom": "Andrés Galarrá",
    "telefono": "0997654321"
  },
  {
    "id": "Fy72yIFo6dhaicIhns8EgFq712",
    "nom": "Manuel Vargas",
    "telefono": "0997654321"
  },
  {
    "id": "Fy72yIFo6dhaicIhns8EgFq712",
    "nom": "Pedro Guzmán"
  }
],
"ubicacion": "Eropa 1 Solar 15"
},
"12": {
},
"13": {
},
},
"usuario": {
"Administrador": {
"thub8qYUhdTvd8fcCTHafjpac": {
"apellido": "ramirez",
"direccion": "machala",
"email": "psulo_cn_02@hotmail.com",
"id": "thub8qYUhdTvd8fcCTHafjpac",
"imagen": "file:///data/user/0/boot.ewp.exponent/cache/ExperienceData/8254/paslocmsA212AppExp/leador/ImagePictore/196wbd6-047e-43f1-858c-00e14f9eb7c.jpg",
"nombre": "psulo",
"obra": 1,
"password": "123456",
"telefono": "0999024129",
"tipo": "Administrador"
},
"Fy72yIFo6dhaicIhns8EgFq712": {
},
"Fy72yIFo6dhaicIhns8EgFq713": {
},
"Fy72yIFo6dhaicIhns8EgFq714": {
},
"Fy72yIFo6dhaicIhns8EgFq715": {
},
"Obrero": {
"Fy72yIFo6dhaicIhns8EgFq712": {
"apellido": "obrero 1",
"direccion": "buquiba",
"email": "proba@hotmail.com",
"id": "Fy72yIFo6dhaicIhns8EgFq712",
"imagen": "file:///data/user/0/boot.ewp.exponent/cache/ExperienceData/8254/paslocmsA212AppExp/leador/ImagePictore/196wbd6-047e-43f1-858c-00e14f9eb7c.jpg",
"nombre": "proba",
"obra": "3",
"password": "123456",
"telefono": "0999024129",
"tipo": "Obrero"
},
"0172yIFo6dhaicIhns8EgFq711": {
},
"8172yIFo6dhaicIhns8EgFq712": {
},
"1172yIFo6dhaicIhns8EgFq713": {
},
"2172yIFo6dhaicIhns8EgFq714": {
},
"3172yIFo6dhaicIhns8EgFq715": {
}
}
}
}

```

Figura 15: Diseño de Base de Datos NoSQL 3

Fuente: Elaboración propia

En la **Figura 15** se observa la estructura de los usuarios, existiendo dos tipos de usuario, administrador y obrero, podemos notar que cada usuario posee un identificador y los campos con su respectiva información personal

2.4.2.6. Esquema de Navegabilidad

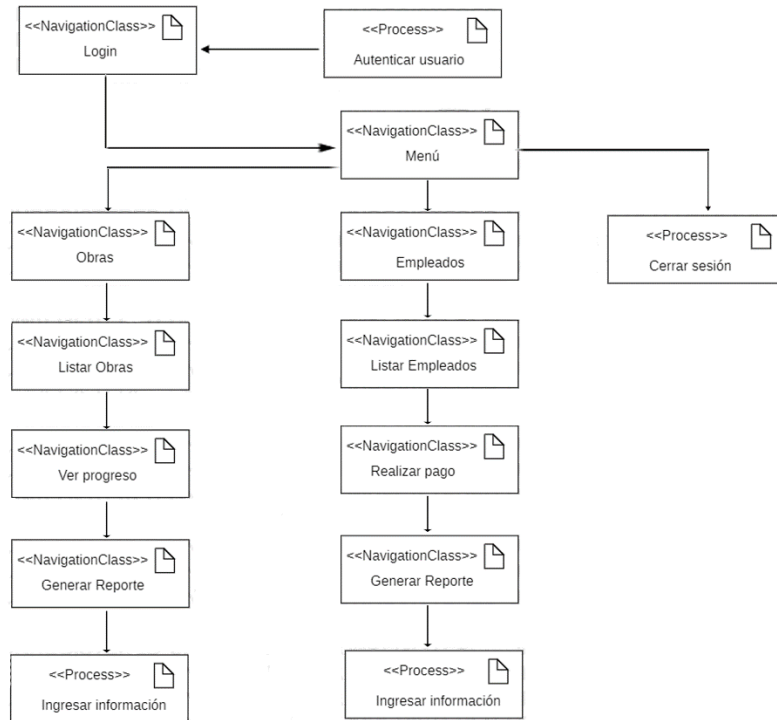


Figura 16: Diagrama de Navegación Administrador

Fuente: Elaboración propia

Como se observa en la **Figura 16** el usuario administrador tiene el acceso al listado de obras y empleados, a la vez podrá generar reportes de obras y de obreros. El administrador podrá ver el progreso de las obras a las que este ha sido asignado y tendrá la capacidad de registrar pagos a obreros.

2.4.2.7. Diseño de Interfaces

2.4.2.7.1. Diseño de Interfaz Web



Figura 17: Diseño Login del sistema

Fuente: Elaboración propia

En la **Figura 17** se puede observar la pantalla de Login en un navegador Web donde el usuario deberá ingresar sus credenciales (email y contraseña) para obtener acceso al menú con las distintas opciones.

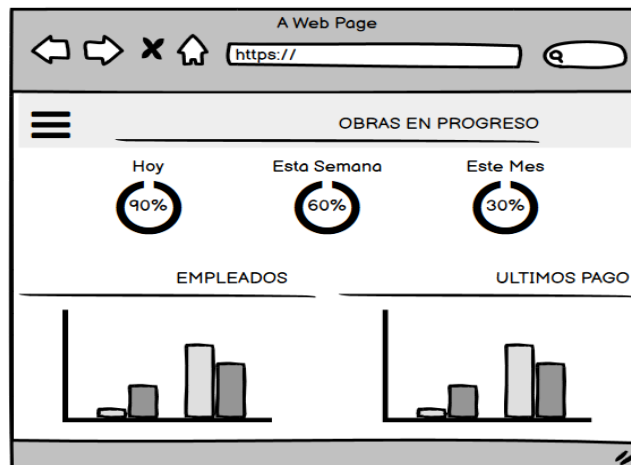


Figura 18: Diseño de página principal

Fuente: Elaboración propia

En la **Figura 18** se observa la pantalla principal del sistema en un navegador Web, en esta se presenta los progresos generales de obras ya sea en el día, semana o mes. De igual manera se detallan mediante gráficos estadísticos los

últimos pagos que se han realizado a obreros, y muestra la cantidad de empleados por obra.

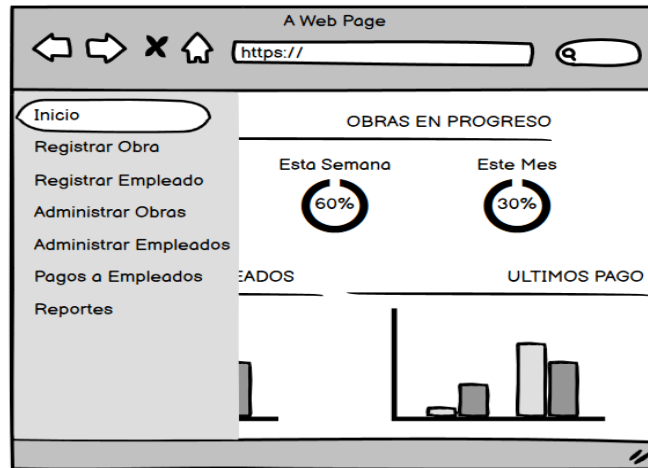


Figura 19: Menú lateral

Fuente: Elaboración propia

En la **Figura 19** se muestra el panel de menú lateral dentro de un navegador Web con las distintas opciones para un usuario administrador, como son: Registro de Obra, Registro de Empleados, Administrar Obras, Administrar Empleados, Pagos a Empleados y Reportes.

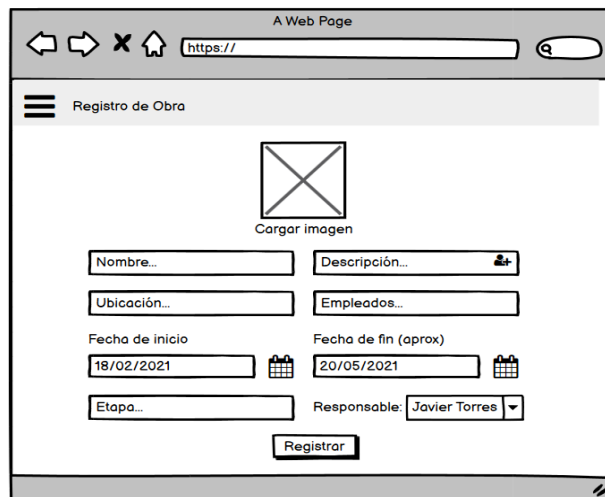


Figura 20: Diseño de registro de obra

Fuente: Elaboración propia

En la **Figura 20** se detalla el formulario para el registro de obra por parte de un administrador en un navegador Web. Se muestran los campos como nombre,

ubicación, selección de empleados, fecha de inicio, fecha de fin, responsable, entre otros.

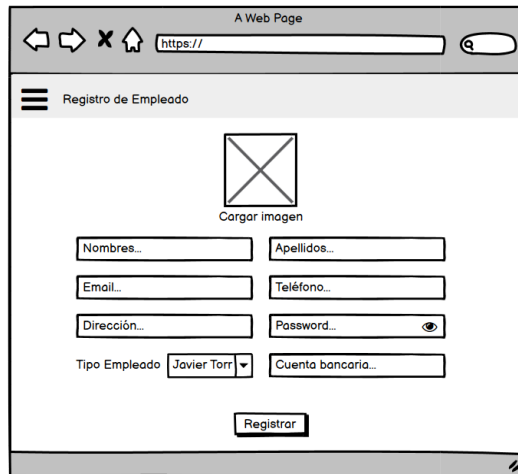


Figura 21: Diseño registro de empleado

Fuente: Elaboración propia

En la **Figura 21** se detalla el formulario para el registro de empleado por parte de un administrador en un navegador Web. Se muestran los campos como nombres, apellidos, email, teléfono, dirección, contraseña, tipo de empleado y cuneta bancaria.

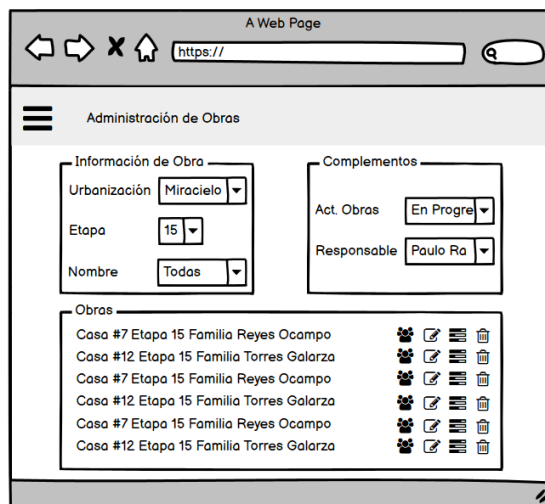


Figura 22: Diseño de Administración de Obras

Fuente: Elaboración propia

En la **Figura 22** se detalla la pantalla de administración de obras, donde el administrador podrá filtrar las obras en base a la urbanización, etapa y nombre. A la vez podrá filtrar sobre las obras que aún no inicia, están en progreso o ya

finalizadas. El administrador podrá visualizar los obreros de cada obra asignada, así como las actividades con sus respectivas fechas, además podrá editar o eliminar una obra en caso de ser estrictamente necesario.

2.4.2.7.2. Diseño de Interfaz Móvil



Figura 23: Diseño de Login

Fuente: Elaboración propia

En la **Figura 23** se observa la pantalla de Login en el diseño para móvil donde el usuario deberá autenticar sus credenciales (email y contraseña) para obtener acceso a las distintas opciones del menú.

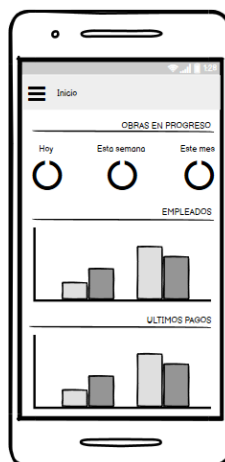


Figura 24: Diseño de pantalla principal

Fuente: Elaboración propia

En la **Figura 24** se observa la pantalla principal del sistema en su diseño para móvil, se puede visualizar los progresos generales de obras ya sea en el día, semana o mes. Al igual que se detallan mediante gráficos estadísticos los últimos pagos que se han realizado a obreros, y la cantidad de empleados por obra.

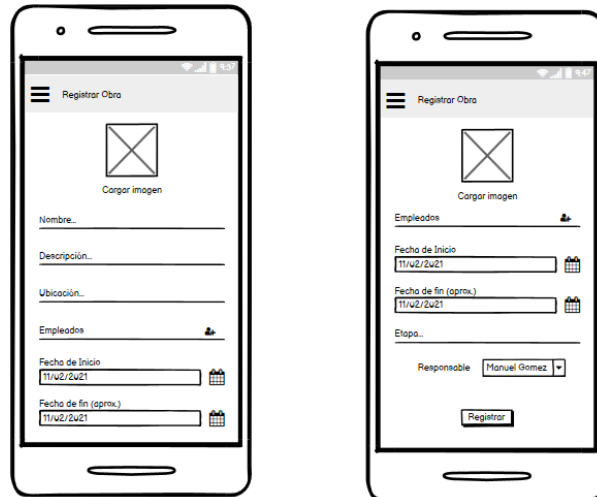


Figura 25: Diseño registro de obra

Fuente: Elaboración propia

En la **Figura 25** se detalla la pantalla en su versión móvil para el registro de obras por parte de los administradores del sistema. Se muestran los campos como nombre, ubicación, selección de empleados, fecha de inicio, fecha de fin, responsable, entre otros.

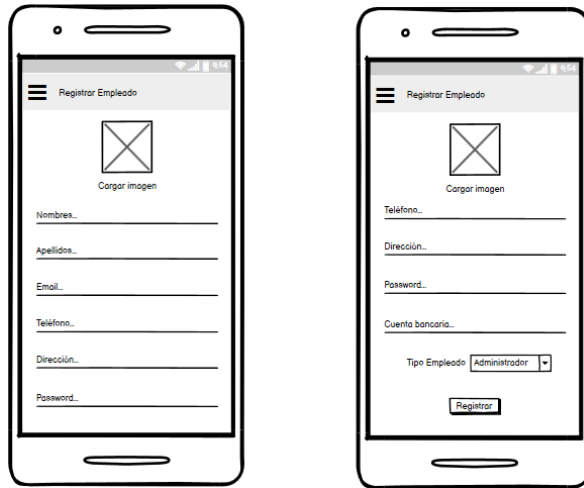


Figura 26: Diseño registro de empleado

Fuente: Elaboración propia

En la **Figura 26** se presenta el formulario en su versión móvil, para el registro de empleados por los administradores. Se muestran los campos de nombres, apellidos, email, teléfono, dirección, contraseña, tipo de empleado, etc.



Figura 27: Diseño administrar obras

Fuente: Elaboración propia

En la **Figura 27** se detalla la pantalla de administración de obras en su versión móvil, donde el administrador podrá filtrar las obras en base a la urbanización, etapa y nombre. A la vez podrá filtrar sobre las obras que aún no inician, están en progreso o ya finalizadas. El administrador podrá visualizar los obreros de cada obra asignada, así como las actividades con sus respectivas fechas,

además podrá editar o eliminar una obra en caso de ser estrictamente necesario.

2.4.2.8. Diagramas de caso de uso

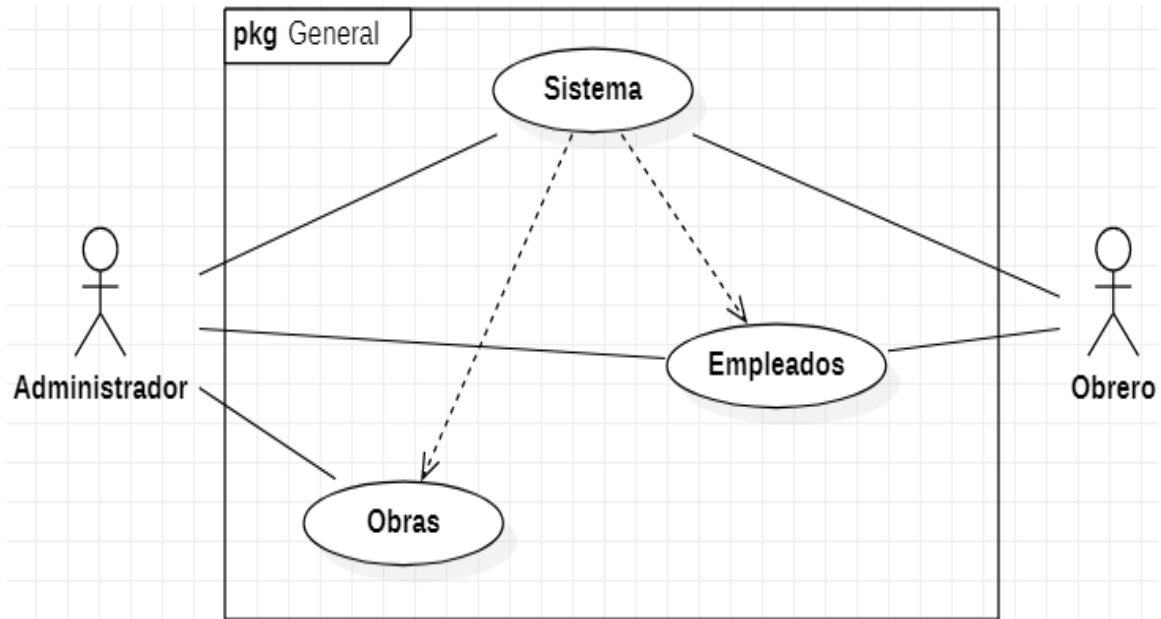


Figura 28: Caso de uso general del sistema

Fuente: Elaboración propia

En la **Figura 28** se muestran los procesos que se dan por parte de los administradores y obreros dentro del sistema. Los administradores y obreros son usuarios donde interactúan directamente con el sistema.

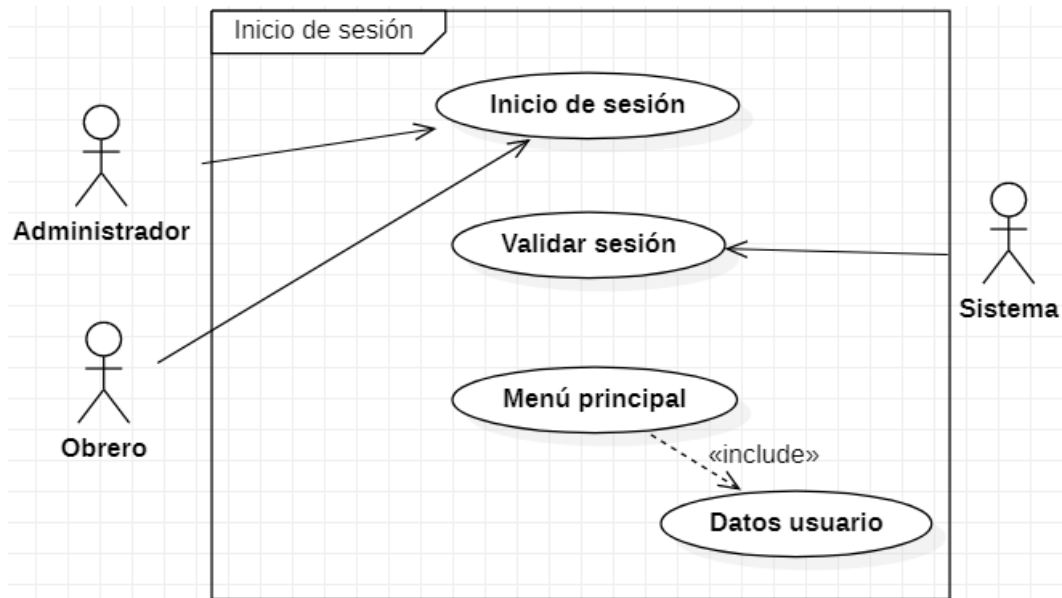


Figura 29: Caso de uso inicio de sesión

Fuente: Elaboración propia

El proceso de inicio de sesión es ejecutado por los administradores y obreros, donde se espera la respuesta del sistema una vez validada la información. Si la respuesta es correcta el usuario tendrá acceso al sistema.

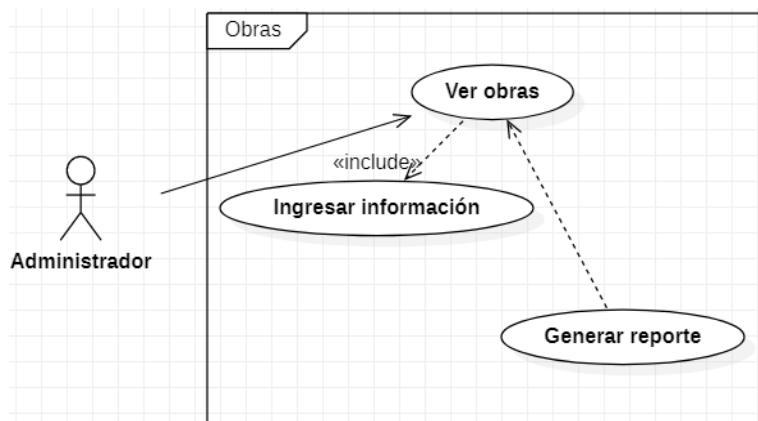


Figura 30: Caso de uso visualización de obras

Fuente: Elaboración propia

En la **Figura 30**, el administrador es el usuario que cuenta con los privilegios para visualizar las obras a las que ha sido asignado. Adicional a esto podrá ingresar información ya sea un nuevo registro o modificando los ya existentes, a su vez tendrá la capacidad de generar reportes de las obras.

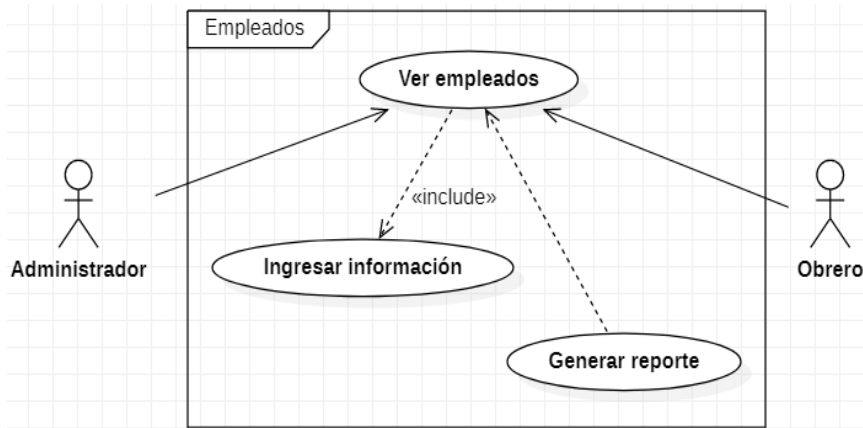


Figura 31: Caso de uso visualización de empleados

Fuente: Elaboración propia

En la **Figura 31**, los usuarios de tipo administrador podrán visualizar a los obreros de las distintas obras de las que son responsables, además de poder registrar información y generar reportes. Por otro lado, los obreros solo podrán visualizar a los empleados de la(s) obra(s) que han sido asignados.

2.4.3. Producción y Estabilización

En esta etapa se abarca la puesta en marcha de la funcionalidad del sistema web y móvil, asegurando el cumplimiento de los requerimientos recolectados para así asegurar un software de calidad.

2.4.3.1. Estándares de codificación

- **Interfaces:** Para los nombres de interfaces se usará el primer carácter en mayúscula del objeto de la clase.
- **Vistas:** El nombre para las vistas será en letras minúsculas.
- **Servicios:** Para los servicios se usará letras en minúscula junto a la separación de la palabra *service*.

2.4.3.2. Estructura de los directorios

En este apartado se detalla básicamente la estructura que posee el sistema

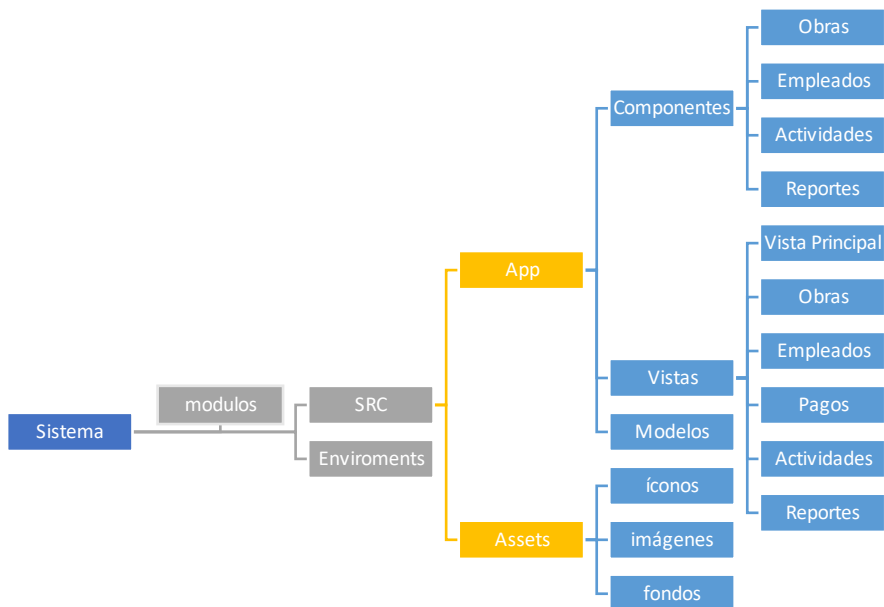


Figura 32: Estructura de los directorios

Fuente: Elaboración propia

La **Figura 32** muestra la estructura de directorios en la cual está constituido el sistema. Podemos notar que el directorio SRC contiene todos los archivos del sistema como son los componentes, vistas y modelos. Por otro lado, el directorio Assets contiene la información relacionada con imágenes, iconos o fondos que sean necesarios en el sistema.

2.5. EJECUCIÓN Y/O EMBALAJE DEL PROTOTIPO

El prototipo del sistema está basado en la representación visual del aplicativo a implementar, por lo tanto, se detallarán los módulos resultantes ya implementados con la ayuda de herramientas tecnológicas y la arquitectura anterior mencionada (**punto 2.1.4**), satisfaciendo el objetivo del proyecto.

2.5.1. Inicio de Sesión

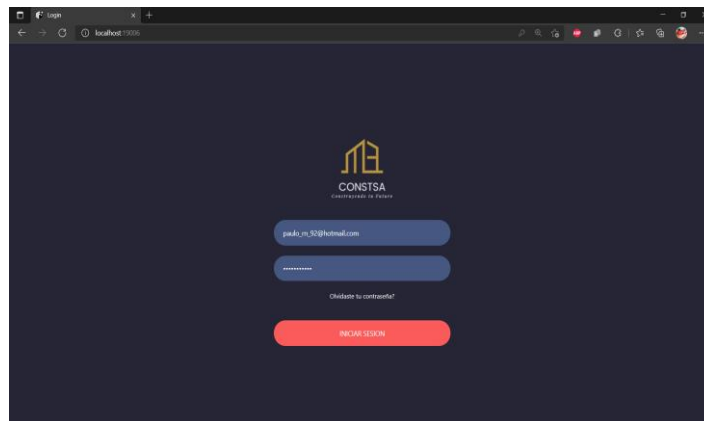


Figura 33: Ejecución del sistema web - Inicio de Sesión

Fuente: Elaboración propia

En la **Figura 33** se observa la ejecución de la página de Login del sistema web, el usuario deberá ingresar sus credenciales, en caso de ser correctos se mostrará la pantalla principal.

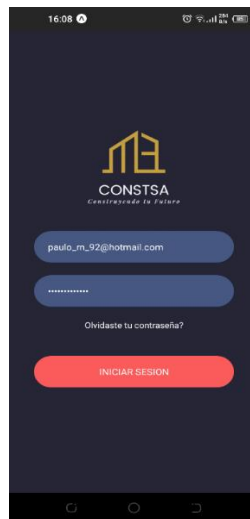


Figura 34: Ejecución de app en Android - Inicio de Sesión

Fuente: Elaboración propia

En la **Figura 34** se presenta la ejecución del aplicativo en un dispositivo Android, el usuario deberá ingresar su correo y contraseña para autenticación, en caso de ser correctas se mostrará la pantalla principal.

2.5.2. Pantalla Principal

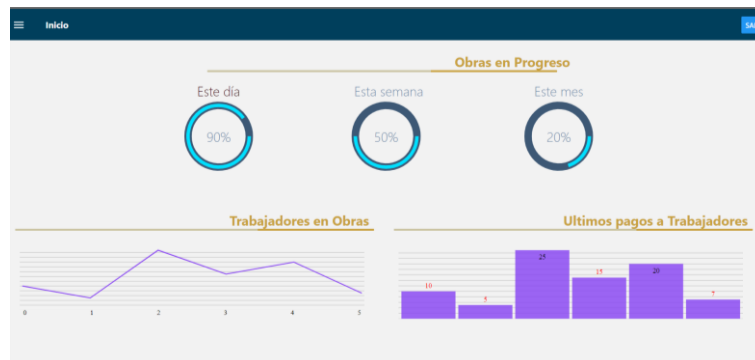


Figura 35: Ejecución del sistema web - Pantalla Principal

Fuente: Elaboración propia

En la **Figura 35** se muestra la pantalla principal del sistema web, donde se puede observar el progreso de las obras en general, Los trabajadores que están en obras y lo últimos pagos que se han realizado.

2.5.3. Registro Obras

The 'Registro de Obra' form includes fields for: 'Nombre' (Casa Folia Abad), 'Descripción' (Casa), 'Ubicación' (Tonos del Mar), 'Empleados' (with a plus icon), 'Fecha de inicio' (12/09/2021), 'Fecha de fin (aprox)' (25/12/2021), and 'Responsable' (paulo ramirez). A red 'REGISTRAR EMPLEADO' button is at the bottom.

Figura 36: Ejecución de sistema web - Registro de Obra

Fuente: Elaboración propia

En la **Figura 36** se muestra la pantalla web para el registro de obras por parte de un administrador, se muestran los campos como nombre, ubicación, responsable, empleados, entre otros.

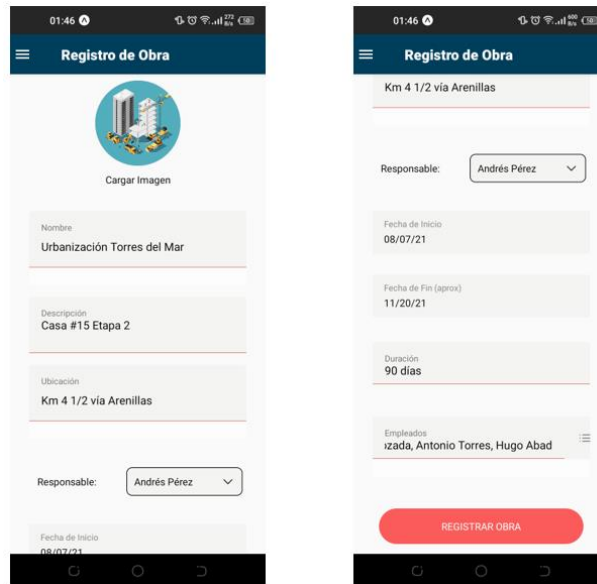


Figura 37: Ejecución de app en Android– Registro de obra

Fuente: Elaboración propia

En la **Figura 37** se muestra la aplicación en Android con la pantalla para el registro de obras, se muestran los campos como nombre, ubicación, responsable, empleados, entre otros.

2.5.4. Registro Empleados

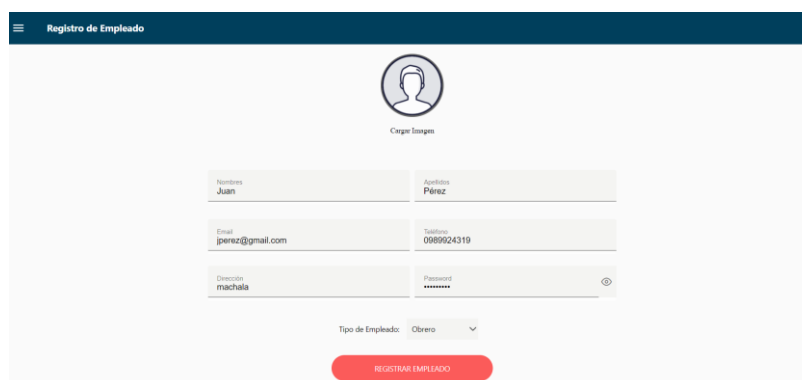


Figura 38: Ejecución de sistema web - Registro de Empleado

Fuente: Elaboración propia

En la **Figura 38** se observa la pantalla web para registro de empleados por parte de un usuario administrador, en él se detallan los campos como nombre, email, teléfono, entre otros.

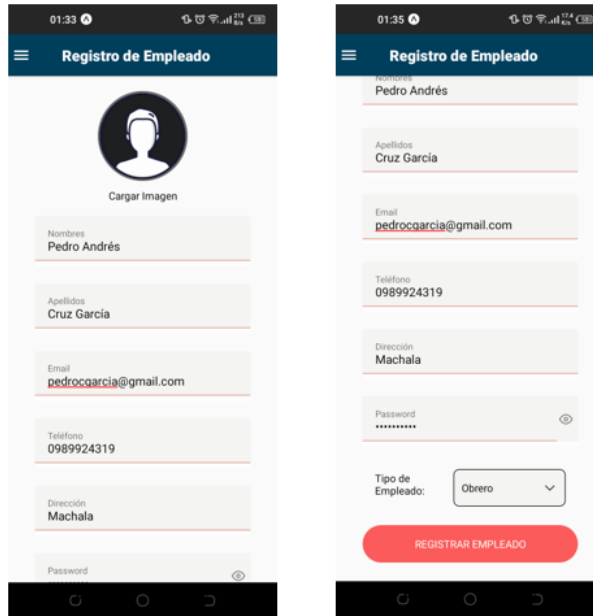


Figura 39: Ejecución de app en Android - Registro de Empleado

Fuente: Elaboración propia

En la **Figura 39** se observa el aplicativo móvil en Android con pantalla para registro de empleados, en él se detallan los campos de información personal.

2.5.5. Visualizar Obras

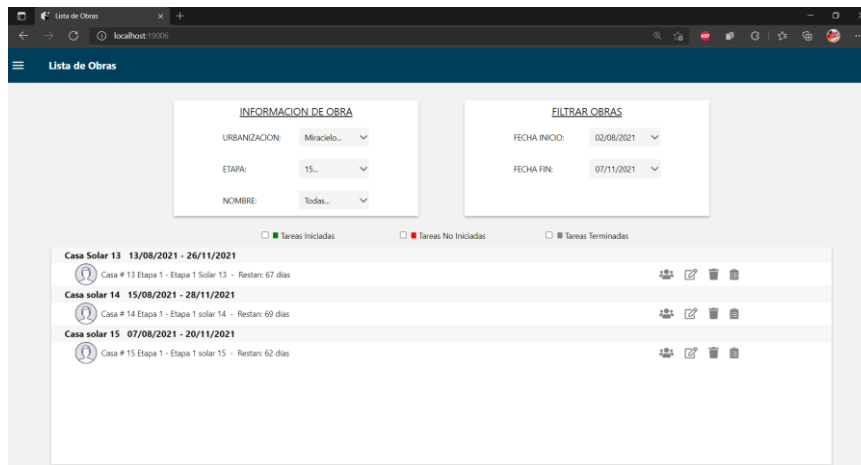


Figura 40: Pantalla de administración de obras en la web

Fuente: Elaboración propia

En la **Figura 40** se muestra la lista de obras disponibles por la empresa, se puede filtrar la visualización de obras en base a la información como por fechas.



Figura 41: Ejecución de app en Android - Visualizar obras

Fuente: Elaboración propia

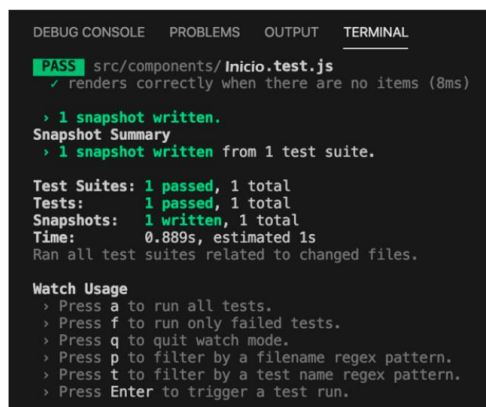
En la **Figura 41** se muestra la pantalla de administración de obras, donde existen opciones que permiten filtrar los datos. Adicional se puede editar y eliminar obras.

3. CAPÍTULO III. EVALUACIÓN DEL PROTOTIPO

3.1. Plan de Evaluación

Para evaluar el sistema móvil y web se hará uso de las recomendaciones de la metodología Mobile-D. Se usará Jest que es una librería para aplicaciones JavaScript, donde permite realizar pruebas unitarias de los archivos del sistema. Para este proyecto se realizará pruebas sobre los componentes utilizados y la lógica de desarrollo como: líneas de código, funciones, declaraciones de variables entre otros. En esta ocasión se evaluará cada vista del sistema.

3.2. Resultados de Evaluación



```
DEBUG CONSOLE  PROBLEMS  OUTPUT  TERMINAL
PASS src/components/Inicio.test.js
  ✓ renders correctly when there are no items (8ms)

  › 1 snapshot written.
Snapshot Summary
  › 1 snapshot written from 1 test suite.

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  1 written, 1 total
Time:        0.889s, estimated 1s
Ran all test suites related to changed files.

Watch Usage
  › Press a to run all tests.
  › Press f to run only failed tests.
  › Press q to quit watch mode.
  › Press p to filter by a filename regex pattern.
  › Press t to filter by a test name regex pattern.
  › Press Enter to trigger a test run.
```

Figura 42: Prueba unitaria a vista Inicio

Fuente: Elaboración propia

En la **Figura 42** se observa la ejecución de la prueba unitaria realizada a la vista Inicio, se realizó una suite de test donde aprobó satisfactoriamente.

```
DEBUG CONSOLE  PROBLEMS  OUTPUT  TERMINAL
[PASS] src/components/Obras.test.js
✓ renders correctly when there are no items (7ms)

> 1 snapshot written.
Snapshot Summary
> 1 snapshot written from 1 test suite.

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  1 written, 1 total
Time:       0.713s, estimated 1s
Ran all test suites related to changed files.

Watch Usage
> Press a to run all tests.
> Press f to run only failed tests.
> Press q to quit watch mode.
> Press p to filter by a filename regex pattern.
> Press t to filter by a test name regex pattern.
> Press Enter to trigger a test run.
```

Figura 43: Prueba unitaria a vista Obras

Fuente: Elaboración propia

En la **Figura 43** se observa la ejecución de la prueba unitaria realizada a la vista Obras, se analizó estructura de componentes, sintaxis y funciones.

```
DEBUG CONSOLE  PROBLEMS  OUTPUT  TERMINAL
[PASS] src/components/Usuarios.test.js
✓ renders correctly when there are no items (7ms)

> 1 snapshot written.
Snapshot Summary
> 1 snapshot written from 1 test suite.

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  1 written, 1 total
Time:       0.889s, estimated 1s
Ran all test suites related to changed files.

Watch Usage
> Press a to run all tests.
> Press f to run only failed tests.
> Press q to quit watch mode.
> Press p to filter by a filename regex pattern.
> Press t to filter by a test name regex pattern.
> Press Enter to trigger a test run.
```

Figura 44: Prueba unitaria a vista Usuarios

Fuente: Elaboración propia

En la **Figura 44** se observa la ejecución de la prueba unitaria realizada a la vista Usuarios, se realizó test de funciones, líneas de código y sintaxis.

```

DEBUG CONSOLE  PROBLEMS  OUTPUT  TERMINAL
PASS src/components/RegistroObra.test.js
✓ renders correctly when there are no items (8ms)
> 1 snapshot written.
Snapshot Summary
> 1 snapshot written from 1 test suite.

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  1 written, 1 total
Time:       0.889s, estimated 1s
Ran all test suites related to changed files.

Watch Usage
> Press a to run all tests.
> Press f to run only failed tests.
> Press q to quit watch mode.
> Press p to filter by a filename regex pattern.
> Press t to filter by a test name regex pattern.
> Press Enter to trigger a test run.

```

Figura 45: Prueba unitaria a vista Registro de Obra

Fuente: Elaboración propia

En la **Figura 45** se observa la ejecución de la prueba unitaria realizada a la vista Registro de Obras, se realizó test del componente de ingreso de datos, de componentes de la interfaz, funciones y sintaxis.

```

-----|-----|-----|-----|-----|-----
File    | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----|-----|-----|-----|-----|-----
All files |    99 |    100 |    100 |    100 |
  Usuarios.js |    99 |    100 |    100 |    100 |
  Obras.js |    98 |    100 |    100 |    100 |
  RegistroObra.js |   100 |    100 |    100 |    100 |
  RegistroObreros.js |   100 |    100 |    100 |    100 |
-----|-----|-----|-----|-----|-----
Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        1.651 s
Ran all test suites matching /jest.spec.js/i.

```

Figura 46: Resultado final de Test realizados a diferentes vistas

Fuente: Elaboración propia

En la **Figura 46** se observa el resultado final presentado por la herramienta Jest en el análisis de los componentes de las diferentes vistas. Se observa que todos los test se generaron con éxito sin presencia de fallos en el uso de componentes. Por otro lado, también se presenta la valoración obtenida en base a los tests de declaraciones de variables, de ejecución de funciones, líneas de código usadas, entre otras características que ofrece la herramienta.

En base a las pruebas realizadas se determina que el sistema ha sido desarrollado de manera correcta y satisfactoria en base a la metodología de

desarrollo Mobile-D. Haciendo un correcto uso de los componentes del framework utilizado en el desarrollo del aplicativo.

3.3. CONCLUSIONES

- Se desarrolló un aplicativo web y móvil totalmente funcional satisfaciendo la problemática en la gestión de empleados y actividades de la empresa de construcción ConstSA, alcanzando la correcta implementación por medio de una arquitectura basada en la nube y un framework de desarrollo multiplataforma, usando la metodología de desarrollo móvil “Mobile-D”.
- “Mobile-D” es una metodología que facilita la implementación de aplicaciones móviles, basándose en pequeñas iteraciones logrando obtener el aplicativo en un periodo corto de tiempo, mediante la implementación de cada una de sus fases.
- La recopilación de requerimientos tiene un impacto primordial en la definición del alcance del proyecto, estableciendo la funcionalidad del sistema y las limitaciones a las que debe regirse.
- Las herramientas CASE cumplen un rol primordial en la construcción del sistema para la gestión de empleados y actividades, permitiendo previsualizar futuros escenarios dentro del aplicativo, brindando la posibilidad de plantear posibles soluciones.
- Las pruebas de evaluación unitarias realizadas a la plataforma web y móvil mediante el uso de la herramienta Jest, garantiza el correcto uso de los componentes empleados en el desarrollo, obteniendo un sistema completamente funcional para la correcta administración de la información.

3.4. RECOMENDACIONES

Acorde a mi experiencia en programación haciendo uso de distintos lenguajes de programación ya sea del lado del servidor o cliente, encuentro necesario el poder recomendar lo siguiente:

- Implementar una metodología de desarrollo ágil dependiendo del proyecto a implementar, garantizando un software de calidad en un periodo corto de tiempo.
- Hacer siempre uso de herramientas garantizadas para el desarrollo del sistema, ya sea del lado del cliente como del servidor, permitiendo gestionar correctamente los recursos disponibles como su relación y adaptación con otras plataformas.
- Priorizar el uso de frameworks con diseño incluido con el fin de reducir los tiempos en la creación de las interfaces, permitiendo darle un mayor enfoque al desarrollo funcional del sistema.
- Es necesario basar los sistemas en arquitecturas flexibles y robustas que permitan la concurrencia de usuarios sin ningún inconveniente, asegurando una rápida obtención de la información.
- La realización de pruebas unitarias sobre un aplicativo de software garantiza el correcto funcionamiento del mismo, permitiendo reducir y corregir los errores que se presenten en la ejecución del sistema.

BIBLIOGRAFÍA

- [1] O. E. Cárdenas Villavicencio, M. P. Zea Ordóñez, M. R. Valarezo Pardo, and R. A. Ramón Ramón, “Comparativa de tendencias de desarrollo de software móvil,” *3C TIC Cuad. Desarro. Apl. a las TIC*, vol. 10, no. 1, Mar. 2021, doi: 10.17993/3ctic.2021.101.123-147.
- [2] A. Carrión, J. Luis, P. Leiva, M. Ríos, and C. Villavicencio, “PROTOTIPO DE APLICACIÓN MÓVIL COMO RECURSO EDUCATIVO EN LA ENSEÑANZA-APRENDIZAJE DE INFORMÁTICA PROTOTYPE MOBILE APPLICATION AS AN EDUCATIONAL RESOURCE IN COMPUTER TEACHING-LEARNING.” [Online]. Available: www.grupocieg.org.
- [3] A. Alebrahim, *Bridging the Gap between Requirements Engineering and Software Architecture*. Wiesbaden: Springer Fachmedien Wiesbaden, 2017.
- [4] V. Cortellessa, R. Eramo, and M. Tucci, “From software architecture to analysis models and back: Model-driven refactoring aimed at availability improvement,” *Inf. Softw. Technol.*, vol. 127, Nov. 2020, doi: 10.1016/j.infsof.2020.106362.
- [5] J. Herrera de la Barrera, J. M. Martínez Buendía, and L. Martínez García, “Software de sostenibilidad turística para el cumplimiento de la NTS colombiana,” *Tur. y Soc.*, vol. 28, pp. 187–205, Dec. 2020, doi: 10.18601/01207555.n28.09.
- [6] P. Megantoro, B. A. Pramudita, P. Vigneshwaran, A. Yurianta, and H. A. Winarno, “Real-time monitoring system for weather and air pollutant measurement with HTML-based UI application,” *Bull. Electr. Eng. Informatics*, vol. 10, no. 3, Jun. 2021, doi: 10.11591/eei.v10i3.3030.
- [7] Z. E. Mamani Rodríguez, L. Del Pino Rodríguez, and J. C. Gonzales Suarez, “Arquitectura basada en Microservicios y DevOps para una ingeniería de software continua,” *Ind. Data*, vol. 23, no. 2, Dec. 2020, doi: 10.15381/idata.v23i2.17278.
- [8] M. A. Sánchez, “Revisión de elementos conceptuales para la

- representación de las arquitecturas de referencias de software Review of conceptual elements at representation of software reference architectures,” *Rev. Cuba. Ciencias Informáticas*, vol. 13, no. 1, pp. 143–157, 2019, [Online]. Available: <http://rcci.uci.cu>.
- [9] J. P. Zumba, “Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software,” *INNOVA Res. J.*, vol. 3, no. 10, pp. 20–33, Oct. 2018, doi: 10.33890/innova.v3.n10.2018.651.
- [10] C. Tecnológica, S.- Romero, M. Antonio, ; Escudero-López, and N. Ezequiel, “PDF generado a partir de XML-JATS4R por Redalyc Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto Propuesta de Metodología Híbrida y Base de Documentación para el Desarrollo de Software Actual.”
- [11] J. R. Molina Ríos, M. P. Zea Ordóñez, M. J. Contento Segarra, and F. G. García Zerda, “COMPARACIÓN DE METODOLOGÍAS EN APLICACIONES WEB,” *3C Tecnol. innovación Apl. a la pyme*, vol. 7, no. 1, Mar. 2018, doi: 10.17993/3ctecno.2018.v7n1e25.1-19.
- [12] J. Manuel *et al.*, “Portada Julio Ezequiel Pérez Carbajo Logotipo y Título de la revista.” [Online]. Available: <http://www.upo.es/MoleQla>.
- [13] J. R. Molina Ríos *et al.*, “*MMS*,” *Metodología para el Diseño y Desarrollo de Aplicaciones Móviles*. Editorial Científica 3Ciencias, 2021.
- [14] P. Huerta-Riveros, H. Gaete-Feres, and L. Pedraja-Rejas, “PLANIFICACIÓN MÓVIL: DISEÑO PARA UNA INSTITUCIÓN DE EDUCACIÓN SUPERIOR,” *Interciencia*, vol. 45, pp. 84–89, Feb. 2020.
- [15] J. R. Molina Ríos, J. A. Honores Tapia, N. Pedreira-Souto, and H. P. Pardo León, “Comparativa de metodologías de desarrollo de aplicaciones móviles,” *3C Tecnol. innovación Apl. a la pyme*, vol. 10, no. 2, pp. 73–93, Jun. 2021, doi: 10.17993/3ctecno/2021.v10n2e38.73-93.
- [16] M. Botto-Tobar, J. León-Acurio, A. Díaz Cadena, and P. Montiel Díaz, Eds., *Advances in Emerging Trends and Technologies*, vol. 1066. Cham: Springer International Publishing, 2020.

- [17] G. P. P, R. Hamsini, and G. R. Smitha, "Agile Development Methodology and Testing for Mobile Applications - A Survey," *Int. J. New Technol. Res.*, vol. 2, no. 9, 2016.
- [18] Y.-S. Jheng, R. Tso, C.-M. Chen, and M.-E. Wu, "Password-Based Authenticated Key Exchange from Lattices for Client/Server Model," 2018.
- [19] N. Palma Pérez, "Solución informática para la selección del servidor web durante la migración a código abierto," *Rev. Cuba. Ciencias Informáticas*, vol. 14, pp. 49–69, 2020.
- [20] F. E. Ríos Pérez, F. Polanco Carrillo, and V. Moreno Vega, "Servidor web empotrado en un FPGA para configurar un Controlador Maestro del Sistema Inteligente de Tráfico Cubano," *Rev. Cuba. Ciencias Informáticas*, vol. 11, pp. 16–28, 2017.
- [21] A. S. Miró, "Desarrollo de una aplicación de Web Mapping con teselas vectoriales en la plataforma NODEJS Development in NODEJS platform of a Web Mapping application with vector tiles," vol. 28, pp. 6–17.
- [22] A. Sterling, "NodeJS and Angular Tools for JSON-LD," Jan. 2019, doi: 10.1109/ICOSC.2019.8665625.
- [23] V. Borrayo and D. Arias, "Sistema servidor-cliente seguro para la verificación de autenticidad de archivos por medio de algoritmos digest HASH reestadarizados," Sep. 2017, doi: 10.1109/CONESCAPAN.2017.8277599.
- [24] F. A. Acevedo-Cardozo, L. C. Calderón-Soto, J. C. González-Castillo, and H. R. Vargas-Torres, "Comunicación de sistemas eléctricos basados en la Norma IEC 61850: Desarrollo de caso usando sampled values en servicios Cliente-Servidor," *Rev. UIS Ing.*, vol. 18, no. 2, Jan. 2018, doi: 10.18273/revuin.v18n2-2019021.
- [25] P. So, "React Native," in *Decoupled Drupal in Practice*, Berkeley, CA: Apress, 2018.
- [26] M. Thakkar, "Unit Testing Using Jest," in *Building React Apps with Server-Side Rendering*, Berkeley, CA: Apress, 2020.

- [27] M. J. Parra Royón, "Servicios de minería de datos en Cloud Computing," Jul. 2019, Accessed: Aug. 31, 2021. [Online]. Available: <http://hdl.handle.net/10481/56592>.
- [28] "Editorial Note: Multimedia Service and Applications with Mobile Cloud Computing," *Multimed. Tools Appl.*, vol. 76, no. 16, pp. 16873–16873, Aug. 2017, doi: 10.1007/s11042-017-4972-7.
- [29] Y. Guedira and J. Rouillard, "Multimodal Interaction Framework Based on Firebase Real-Time Database," 2021.
- [30] G. T. Le, N. M. Tran, and T. V. Tran, "IoT System for Monitoring a Large-Area Environment Sensors and Control Actuators Using Real-Time Firebase Database," 2021.
- [31] R. A. Burnashev, A. V. Gubajdullin, and A. I. Enikeev, "Specialized Case Tools for the Development of Expert Systems," 2018.
- [32] R. Mercurio and B. Merrill, *Beginning Microsoft 365 Collaboration Apps*. Berkeley, CA: Apress, 2021.
- [33] D. Chatfield, *Microsoft Project 2013 Step by Step*. 2013.
- [34] J. Molina Ríos, M. Valarezo Pardo, J. Honores Tapia, B. Romero Machare, and C. Jumbo Parrales, *Del análisis al diseño*. Editorial Científica 3Ciencias, 2019.
- [35] B. Silver, "Life's too short for bad software!," 2021. <https://balsamiq.com/> (accessed Aug. 21, 2021).
- [36] StarUML, "StarUML documentation," 2020. <https://docs.staruml.io/> (accessed Aug. 23, 2021).
- [37] C. Guevara-Vega, J. Hernández-Rojas, M. Botto-Tobar, I. García-Santillán, A. Basantes Andrade, and A. Quiña-Mera, "Automation of the Municipal Inspection Process in Ecuador Applying Mobile-D for Android," 2020.
- [38] J. A. Orozco Toro and J. E. Alzate Sanz, "Stakeholders, actores estratégicos en la construcción de marca," *Anagramas - Rumbos y*

sentidos la Comun., vol. 16, no. 32, Jun. 2018, doi:
10.22395/anqr.v16n32a6.

- [39] E. Pineda Ballesteros, F. R. Tellez Acuña, and J. Medina Cruz, "Requerimientos de Software: Prototipado, software heredado y análisis de documentos," *Ing. y Desarro.*, vol. 37, no. 2, Mar. 2020, doi: 10.14482/inde.37.2.1053.
- [40] C. Garita, G. Ortiz, and J. Mora-Mora, "Análisis de requerimientos para un sistema nacional de monitoreo de puentes," *Rev. Technol. en Marcha*, Oct. 2018, doi: 10.18845/tm.v31i4.3965.
- [41] D. E. Lucero Guevara and N. B. La Serna Palomino, "Automatización de requisitos: Historias de usuario generadas a partir de un modelo orientado a objetivos basado en el framework i*," *Interfases*, no. 011, 2018, doi: 10.26439/interfases2018.n011.2953.

ANEXOS

ANEXO A: FORMATO DIGITAL DE REGISTRO DE ACTIVIDADES DIARIAS

Fuente: Empresa ConstSA

Registro de Actividades diarias			
Nombre:	Torres del Mar	N.- Obra:	#00129
Ubicación:	km 4 1/2 via Arenillas	Fecha:	10/8/2021
Etapa Actual:	Estructura		
Actividades Realizadas		Observaciones	
Encofrado para infraestructura de columna		Ninguna	
Cortar varillas y armar estructura		Ninguna	
Elaborar abrazaderas para encofrado		Ninguna	
Registro de Actividades diarias			
Nombre:	Torres del Mar	N.- Obra:	#00129
Ubicación:	km 4 1/2 via Arenillas	Fecha:	11/8/2021
Etapa Actual:	Estructura		
Actividades Realizadas		Observaciones	
Encofrado para infraestructura de columna		Ninguna	
Vaciado de concreto		Ninguna	
Relleno de encofrados		Ninguna	

Figura 47: Formato digital de registro de actividades diarias

Fuente: Empresa ConstSA

ANEXO B: FORMATO DIGITAL DE REGISTRO DE ASISTENCIA

INFORMACION DE OBRA				
Nombre:	Torres del Mar	N.- Obra:	#00129	
Ubicación:	km 4 1/2 via Arenillas	Fecha:	10/8/2021	
REGISTRO DE ASISTENCIA				
	Juan Torres	Nestor Chica	Miguel Carrión	Byron Segura
Asistencia	X	X	X	X
INFORMACION DE OBRA				
Nombre:	Torres del Mar	N.- Obra:	#00129	
Ubicación:	km 4 1/2 via Arenillas	Fecha:	11/8/2021	
REGISTRO DE ASISTENCIA				
	Juan Torres	Nestor Chica	Miguel Carrión	Byron Segura
Asistencia	X	X		X

Figura 48: Formato digital de registro de asistencia

ANEXO C: ANÁLISIS DE RIESGOS

Tabla 19: Gestión de Riesgos

N.-	RIESGO	EFECTO	RESPUESTA
R-1	Poca disponibilidad del cliente.	Los requerimientos y las historias de usuario no son identificados en el tiempo estimado.	Establecer las preguntas adecuadas para la realización de pocas reuniones, pero concisas.
R-2	El cliente modifica o agrega nuevos requerimientos en etapas finales.	Se debe rediseñar la base de datos o las interfaces de usuario.	Validar los diseños de las interfaces y funcionalidades del aplicativo ya sea por reuniones o medios de comunicación.
R-3	Falta de disponibilidad o conocimiento de las herramientas de desarrollo o diseño.	Se dificulta el diseño o desarrollo de la aplicación.	Cambiar a una herramienta conocida que permita realizar las mismas funcionalidades que la anterior.
R-4	Fallos de seguridad en la implementación del código	Logeo de personal inadecuado al sistema	Usar módulos para brindar seguridad, así como protocolos.
R-5	Problemas con el manejo del aplicativo	Falta de capacitaciones para el manejo de software	Brindar capacitaciones sobre el uso del sistema antes probar en la empresa

Fuente: Elaboración propia