

Asistente de cálculo de prefactibilidad para construcción de vivienda.

Memoria de Proyecto Final de Máster

Máster Universitario en Desarrollo de sitios y aplicaciones web

Área de Informática, Multimedia y Telecomunicación

Autor: Diana Alexandra Morocho Puchaicela

Consultor: Carles Arnal Castello

Profesor: César Pablo Córcoles Briongos

07 de junio de 2021

Créditos



Esta obra está sujeta a una licencia de Reconocimiento-
NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Dedicatoria

“No puedo cambiar la dirección del viento, pero puedo ajustar mis velas para alcanzar mi destino”

- Jimmy Dean -

En este momento de alegría, inclino mi rostro para agradecer a Dios, por acompañarme con ternura en los momentos de dificultad y mostrarme tu rostro en los gestos delicados de tu creación.

Quiero dedicar el esfuerzo de este trabajo, al motor de mis sueños y razón de mi vida, mis pequeños hijos Pablo Andrés y Leonel David, de quienes aprendo cada día el valor de las pequeñas cosas y la fortaleza para enfrentarse al mundo como guerreros.

Mi eterna gratitud a mis padres, María y Teófilo, a quienes tanto amo y admiro.

Finalmente, a mis familiares y amigos que de una u otra forma me alientan a la consecución de mis sueños.

Abstract

El presente documento, refleja la memoria del trabajo de fin de máster en el que se especifica las fases del desarrollo de una aplicación web para el cálculo de prefactibilidad de construcción de vivienda para la empresa de arquitectura y construcción WETCH.

El desarrollo de la aplicación está basado en una arquitectura SPA (Single Page Application) utilizando como plataforma de desarrollo el framework Angular y la utilización de las guías de diseño, usabilidad y experiencia de usuario adquiridas en el máster.

Es así que, previo a la codificación del software con la ayuda de la aplicación de métodos y técnicas de investigación científica, se introdujo en el contexto del cálculo de prefactibilidad en el área de la arquitectura, proceso que es desarrollado en la fase preliminar de la construcción de cualquier proyecto de inversión arquitectónico de vivienda y que permitió determinar los requerimientos de software necesarios.

Luego, para la construcción del sistema, se aplicó las fases de análisis, diseño, implementación y pruebas, así como la utilización de determinados artefactos propios de cada fase como: prototipos, diagramas, pruebas, etc.

Finalmente, se realiza una discusión y socialización de los resultados obtenidos, frente a los objetivos planteados, y con ello, la formulación de premisas a modo de conclusiones y oportunidades de proyectos futuros.

Palabras clave: Memoria, Trabajo de Final de Máster, Aplicación web, Single Page Applications, Angular, Prefactibilidad de construcción, Ecuador.

Abstract (english version)

This document reflects the report of the master's thesis in which the phases of the development of a web application are specified for calculating the budget and pre-feasibility of housing construction for the architecture and construction company WETCH.

The development of the application is based on a SPA (Single Page Application) architecture using the Angular framework as a development platform and the use of the design, usability and user experience guides acquired in the master.

Thus, prior to the coding of the software with the help of the application of scientific research methods and techniques, it was introduced in the context of the calculation of budget and pre-feasibility in the area of architecture, a process that is developed in the preliminary phase of the construction of any architectural investment project of housing and that allowed to determine the necessary software requirements.

Then, for the construction of the system, the analysis, design, implementation and testing phases were applied, as well as the use of certain artifacts typical of each phase such as: prototypes, diagrams, tests, etc.

Finally, a discussion and socialization of the results obtained is carried out, against the objectives set, and with it, the formulation of premises as conclusions and opportunities for future projects.

Keywords: Memory, Master's Final Project, Web Application, Single Page Applications, Angular, Construction Pre-feasibility, Ecuador.

Agradecimiento

Es mi deber expresar un sincero agradecimiento a:

La Universidad Oberta de Catalunya, en la titulación del Master Universitario en Desarrollo de sitios y aplicaciones web donde forjé mis conocimientos y a mis maestros a quienes le debo mi formación profesional.

A los consultores y profesores, Carles Arnal Castello y César Pablo Córcoles Briongos, al compartir sus conocimientos, tiempo y colaboración desinteresada durante el transcurso de las tutorías para dirigir mi trabajo de fin de máster.

Al Arq. Wilson Eduardo Tapia, gerente general de la empresa de arquitectura y construcción WETCH, por proveerme de los requerimientos, lineamientos técnicos y facilidades logísticas para el desarrollo del software.

Finalmente, a mis familiares y amigos que siempre me acompañan especialmente en la adversidad.

Índice

CAPÍTULO I. INTRODUCCIÓN.....	12
1. Introducción	12
2. Descripción	13
3. Objetivos	15
3.1. Principales.....	15
3.2. Secundarios	15
4. Metodología	16
4.1. Métodos	16
4.2. Técnicas.....	16
4.3. Metodología de desarrollo de software.....	16
5. Planificación.....	18
5.1. Entregas.....	18
5.2. Hitos	18
5.3. Diagrama de Gantt.....	19
6. Presupuesto.....	21
CAPÍTULO II. ANÁLISIS.....	22
1. Marco teórico	22
1.1. Single Page Application (SPA)	22
1.2. Normativa de construcción en Ecuador.....	24
2. Contenidos.....	25
3. Requerimientos.....	26
3.1. Índice de valoración para historias de usuario	26
3.2. Historias de usuario aplicadas	27
3.3. Requerimientos Funcionales	30
3.4. Requerimientos No Funcionales.....	30
4. Perfiles de usuario	31
5. Análisis de mercado	33
5.1. Contexto social y de mercado.....	33
5.2. Productos similares.....	33
5.3. Estudio de mercado	33
CAPÍTULO III. DISEÑO	35
1. Arquitectura de la aplicación	35
1.1. Back-end.....	35

1.2. Front-end.....	36
2. Plataforma de desarrollo	37
2.1. Software	37
3. Usabilidad/UX	38
3.1. Formas de interacción	38
3.2. Sitemap	38
4. Prototipos.....	40
5. Diagrama de clases	44
6. Diagrama de casos de uso.....	45
CAPÍTULO IV. IMPLEMENTACIÓN.....	46
1. Proceso de desarrollo.....	46
1.1. Plan de iteraciones	46
1.2. Tarjetas CRC	47
1.3. Ejecución del plan de iteraciones	48
CAPÍTULO V. VALIDACIÓN.....	65
1. Tests	65
1.1. Pruebas unitarias.....	65
1.2. Prueba de rendimiento	67
1.3. Prueba de validación de usuario	70
2. Requisitos de implantación.....	71
2.1. Requisitos de Back-end	71
2.2. Requisitos de Front-end.....	72
3. Instrucciones de implantación	73
3.1. Implantación del Back-end.....	73
3.2. Implantación del Front-End.....	76
CAPÍTULO VI. DISCUSIÓN	78
1. Proyección a futuro.....	78
2. Conclusiones	79
Anexo 1. Entregables del proyecto.....	80
Anexo 2. Librerías utilizadas	81
Anexo 3. Ciclo de vida NgRx.....	83
Anexo 4. Libro de estilo	84
Anexo 5. Glosario	86
Anexo 6. Bibliografía.....	87
Anexo 7. Vita	89

Figuras y tablas

Índice de figuras

Figura 1. Nomenclatura de actividades – Diagrama de Gantt.....	19
Figura 2. Diagrama de Gantt – Planificación.....	20
Figura 3. Arquitectura Angular	23
Figura 4. Criterios de la Norma Ecuatoriana de la Construcción.....	24
Figura 5. Arquitectura de la aplicación.....	35
Figura 6. Screen Sizes and Breackpoints	38
Figura 7. Sitemap de la aplicación web.....	39
Figura 8. Wireframe mobile - página principal.....	40
Figura 9. Wireframe mobile - página de prefactibilidad.....	41
Figura 10. Wireframe mobile - página de prefactibilidad.....	41
Figura 11. Wireframe desktop – página principal.....	42
Figura 12. Wireframe desktop – página de prefactibilidad	43
Figura 13. Diagrama de clases de la aplicación.....	44
Figura 14. Diagrama de casos de uso.....	45
Figura 15. Instalación Docker.....	48
Figura 16. Instalación Composer	49
Figura 17. Creación de proyecto Laravel.....	49
Figura 18. Inicio de servicio Docker	49
Figura 19. Configuración de base de datos.....	50
Figura 20. Módulo de autenticación Laravel.....	50
Figura 21. Ejecución de Laravel.....	50
Figura 22. Instalación de npm.....	51
Figura 23. Instalación de Angular CLI	51
Figura 24. Creación del proyecto Angular.....	51
Figura 25. Angular Material	52
Figura 26. Angular Flex Layout	52
Figura 27. Fontawesome.....	52
Figura 28. NgRx.....	52
Figura 29. Effects	52
Figura 30. NgRx Store Localstorage	53
Figura 31. Formulario - Registro de nuevo usuario	53
Figura 32. Correo electrónico de validación de cuenta.....	54
Figura 33. Formulario – Login	54
Figura 34. Formulario – Reestablecer contraseña	54
Figura 35. Correo electrónico de reseteo de contraseña.....	55
Figura 36. Formulario – Resetear Contraseña	55
Figura 37. Lista de áreas de vivienda – Vista Mobile	56
Figura 38. Lista de áreas de vivienda – Vista Desktop	56

Figura 39. Formulario – Área de vivienda – Vista Mobile	56
Figura 40. Formulario – Área de vivienda – Vista Desktop	56
Figura 41. Lista de ambientes – Vista Mobile.....	57
Figura 42. Lista de ambientes – Vista Desktop.....	57
Figura 43. Formulario – Ambiente – Vista Desktop.....	57
Figura 44. Formulario – Ambiente – Vista Mobile	58
Figura 45. Lista de proyectos – Vista Desktop.....	58
Figura 46. Lista de proyectos – Vista Mobile.....	59
Figura 47. Formulario – Proyecto – Vista Mobile	59
Figura 48. Formulario – Proyecto – Vista Desktop.....	59
Figura 49. Lista de estudios de prefactibilidad – Vista Desktop.....	60
Figura 50. Estudio de prefactibilidad, área de construcción - Vista Desktop.....	60
Figura 51. Estudio de prefactibilidad, plan de necesidades – Vista Desktop	61
Figura 52. Resultados del estudio de prefactibilidad – Vista Desktop.....	61
Figura 53. Vista dashboard del usuario Administrador – Vista Desktop.....	62
Figura 54. Vista dashboard del usuario Planificador – Vista Desktop.....	62
Figura 55. Vista dashboard del usuario cliente – Vista Desktop	63
Figura 56. Formulario de contacto para clientes – Vista Desktop	63
Figura 57. Formulario de comunicaciones recibidas - Vista Desktop.....	64
Figura 58. Administración de usuarios – Vista Desktop.....	64
Figura 59. Prueba unitaria – Login test	65
Figura 60. Prueba unitaria – Verify User	66
Figura 61. Ejecución exitosa de pruebas unitarias.....	66
Figura 62. Test de rendimiento 1 de 7.....	67
Figura 63. Test de rendimiento 2 de 7.....	67
Figura 64. Test de rendimiento 3 de 7.....	67
Figura 65. Test de rendimiento 4 de 7.....	67
Figura 66. Test de rendimiento 3 de 7.....	68
Figura 67. Test de rendimiento 4 de 7.....	68
Figura 68. Test de rendimiento 7 de 7.....	68
Figura 69. Puntaje obtenido luego de la optimización	70
Figura 70. Validación de usuario. Gerente Wetch.....	70
Figura 71. Creación de base de datos.	73
Figura 72. Importación de la estructura de la db	73
Figura 73. Creación de subdominio	73
Figura 74. Cliente FTP	74
Figura 75. Cliente FTP	74
Figura 76. Permiso al directorio storage.....	74
Figura 77. Configuración del archivo .htaccess.....	75
Figura 78. Configuración del archivo .env.....	75
Figura 79. Habilitación de claves SSH	75
Figura 80. Acceso mediante terminal.....	76
Figura 81. Api rest en servidor	76

Figura 82. Construcción de la SPA versión compilada para producción.....	76
Figura 83. Subida de archivos al servidor.	77
Figura 84. Configuración de acceso .htaccess.....	77
Figura 85. Acceso a la aplicación en producción	77
Figura 86. Ciclo de vida NgRx.....	83
Figura 87. Logotipo de la empresa Wetch.....	84
Figura 88. Paleta de colores corporativa.....	84
Figura 89. Tipografía corporativa	84
Figura 90. Imagen de pie de página.....	85

Índice de tablas

Tabla 1: Resumen de entregas del TFM.....	18
Tabla 2. Hitos de la planificación del TFM.....	19
Tabla 3. Presupuesto del proyecto.....	21
Tabla 4. Índice de valoración del campo “Prioridad de negocio”	26
Tabla 5. Índice de valoración del campo “Riesgo de desarrollo”	26
Tabla 6. Índice de valoración del campo “Puntos estimados”	26
Tabla 7. Historia de usuario 1. Insumos para el cálculo de prefactibilidad de construcción de vivienda	27
Tabla 8. Historia de usuario 2. Cálculo de prefactibilidad de construcción de vivienda	28
Tabla 9. Historia de usuario 3. Insumos para el cálculo de prefactibilidad de construcción de vivienda.....	28
Tabla 10. Historia de usuario 4. Cálculo de prefactibilidad de construcción de vivienda	29
Tabla 11. Historia de usuario 5. Tipos de usuario	29
Tabla 12. Requerimientos funcionales de la aplicación	30
Tabla 13. Requisitos no funcionales de la aplicación.....	30
Tabla 14. Perfiles de usuario de la aplicación.....	32
Tabla 15. Plan de iteraciones.....	46
Tabla 16. Tarjeta CRC – User.....	47
Tabla 17. Tarjeta CRC – Tipo de área de vivienda	47
Tabla 18. Tarjeta CRC – Ambiente	47
Tabla 19. Tarjeta CRC – Proyecto	47
Tabla 20. Tarjeta CRC – Tipo Acabado.	48
Tabla 21. Tarjeta CRC – Prefactibilidad.....	48
Tabla 22. Métricas obtenidas de la plataforma PageSpeeds Insights.....	67

CAPÍTULO I. INTRODUCCIÓN

1. Introducción

En el último Informe Nacional del Ecuador para la Tercera Conferencia de las Naciones Unidas sobre Vivienda y Desarrollo Urbano Sostenible HABITAT III (2015), se menciona que el Ecuador, al 2010, cuenta con una población de 14'483.499 habitantes de los cuales el 62,7% vive en zonas urbanas, con una proyección de 64% al 2020, denotando un gran crecimiento desde 1970 que era del 39,5%, provocando un desarrollo urbano desordenado. (pág. 12)

Es así que, el gobierno ecuatoriano con la finalidad de promover el crecimiento urbano sostenible, a través de sus ministerios y subsecretarías, determinaron la normativa que regula los procesos de construcción en el territorio nacional, que permite hacer cumplir con las exigencias básicas de seguridad y calidad, siendo esta una oportunidad donde empresa de arquitectura y construcción WETCH ha enfocado su nicho de negocio en una población predominantemente urbana para ofrecerle sus servicios profesionales en el ámbito de dirección de proyectos de vivienda.

Con estas premisas, el presente trabajo de fin de máster, permite aplicar los conocimientos obtenidos durante el transcurso del postgrado y colaborar con la empresa WETCH, mediante el desarrollo de una aplicación web que automatiza el proceso de cálculo de prefactibilidad en proyectos de construcción de vivienda.

A lo largo de la lectura de este documento, encontrará información técnica y relevante en el ámbito de la arquitectura, pero principalmente sobre el desarrollo web, destacando todos los aspectos necesarios para el análisis, desarrollo, implementación y pruebas, así como la aplicación de métodos y técnicas de investigación científica utilizados.

2. Descripción

La vivienda, como respuesta a las necesidades básicas de cobijo, alojamiento y protección del ser humano, cuenta con el respaldo normativo internacional, siendo así, que en el documento de la Declaración Universal de Derechos Humanos, en su artículo 25 señala que: “Toda persona tiene derecho a un nivel de vida adecuado que le asegure, así como a su familia, la salud y el bienestar, y en especial la alimentación, el vestido, la vivienda, la asistencia médica y los servicios sociales necesarios” (Organización de Naciones Unidas, 1948, pág. 7), de ahí que, es deber de todos los estados miembros, el proveer de las estrategias para que las personas ejerzan sus derechos, como procura de la calidad de vida.

En el Ecuador, el ejercicio del derecho a la vivienda, está regulado por normas constitucionales y legales, que son promovidas por el Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), teniendo como finalidad entre otras, la de regular los procesos que permitan cumplir con las exigencias básicas de seguridad y calidad en todo tipo de edificaciones. Por lo tanto, es el MIDUVI, que a través de la Subsecretaría de Hábitat y Asentamientos Humanos dispone con obligatoriedad y cumplimiento a nivel nacional que: “todos los proyectos arquitectónicos y los procesos de construcción deberán observar las condiciones o parámetros establecidos en la Norma Ecuatoriana de la Construcción y las regulaciones locales, expedidas por los distintos Gobiernos Autónomos Descentralizados Municipales” (Ministerio de Desarrollo Urbano y Vivienda, 2021, pág. 1)

Con este antecedente, en el ejercicio de los derechos, cualquier persona natural o jurídica que desea poner en marcha un proyecto de construcción de vivienda, debe ser asesorado por profesionales de la construcción, quienes ajustados a la Norma Ecuatoriana de Construcción (NEC) y respetando el presupuesto inicial inversión así como los requerimientos de los ambientes que desea implementar, proponen un proyecto arquitectónico, que en su fase inicial, contempla el proceso de cálculo de prefactibilidad, lo que implica una actualización constante de rubros, que infieren directamente en el análisis de costos directos para la obtención de precios unitarios en una construcción, como es el caso de estimaciones de mano de obra, material puesto en obra, etc.

Es así, que el presente trabajo de fin de master, presenta el desarrollo de una aplicación web que automatiza el proceso de cálculo de prefactibilidad en proyectos de construcción de vivienda, como herramienta tecnológica para la empresa de arquitectura y construcción WETCH.

El proyecto de software, permite realizar los cálculos aproximados de inversión en un proyecto de construcción de vivienda (prefactibilidad), basado en la gestión de rubros y sus tipos, áreas de construcción, ambientes, acabados, tipos de vivienda, entre otros. La gestión de la información está disponible en una vista directa al cliente, lo que le permitirá involucrarse directamente con la ejecución de su proyecto y consecuentemente con su asesor profesional de la construcción.

El producto de software del presente trabajo de fin de master, será una aplicación web responsiva con el enfoque mobile-first, que cuente con una arquitectura Single Page Applications (SPA), utilizando como plataforma de desarrollo el framework Angular, apoyado con herramientas modernas para el maquetado en el front-end como HTML5 y CSS3 y, contemplando los estándares actuales de programación web. Mientras tanto, en el back-end se consumirá los datos mediante una API-REST desarrollada con php y Laravel como framework, todas estas herramientas permiten aplicar efectivamente los conocimientos obtenidos a lo largo del máster.

3. Objetivos

Los objetivos de investigación para el presente trabajo de titulación de fin de máster son:

3.1. Principales

- Diseñar e implementar una aplicación web para automatizar el proceso de cálculo de prefactibilidad en proyectos de construcción de vivienda.

3.2. Secundarios

- Determinar los requerimientos de software para la aplicación web, mediante el análisis del proceso de cálculo de prefactibilidad en proyectos de construcción de vivienda urbana.
- Diseñar una aplicación web responsive bajo el enfoque mobile-first, utilizando como base los requerimientos obtenidos del análisis y la aplicación de las guías de diseño, usabilidad y experiencia de usuario aprendidas en el máster.
- Implementar los módulos para el cálculo de prefactibilidad en proyectos de construcción de vivienda bajo la arquitectura Single Page Application.
- Elaborar un plan de verificación a la aplicación web, mediante la ejecución de pruebas de funcionalidad y rendimiento.

4. Metodología

Para la ejecución del proyecto es conveniente y necesaria la adopción de los siguientes métodos científicos, que viabilizan la obtención de información relevante y fiable, para ello se utilizó

4.1. Métodos

- **Método Científico:** “método de investigación usado principalmente en la producción de conocimiento en las ciencias” (Castán, 2014). La aplicación de este método tendrá mayor protagonismo en el desarrollo de la construcción de software, así como en la redacción de la memoria de trabajo de fin de máster, ya que este método permite contar con un trabajo planificado y sustentable en los conocimientos de la ciencia, además de viabilizar la detección y corrección de errores durante el proceso.
- **Método deductivo:** Cuyo estudio parte de lo general a lo particular, permite comprender y desarrollar las fases de planificación, diseño y desarrollo del software, partiendo del conocimiento de los aspectos generales en el área de la arquitectura, hasta abordar detalladamente cada aspecto técnico utilizado en el cálculo de prefactibilidad para proyectos de vivienda, que son la base para la determinación de requerimientos de software.
- **Método inductivo:** Estudio que parte de lo particular a lo general, este singular método es de gran aplicabilidad continua durante la fase de ejecución pruebas de funcionalidad e interfaz de usuario, donde como punto de partida se prueban los componentes pequeños hasta determinar la ejecución correcta del módulo general.

4.2. Técnicas

- **Análisis de información:** En este estudio, la técnica de análisis de información está enfocada en recopilar toda la información actual y relevante a acerca de los procesos de construcción arquitectónicos y sus variables, con el propósito de automatizar el proceso de cálculo de prefactibilidad de construcción de vivienda.
- **Observación Directa:** Mediante la observación, se constatará el manejo actual de los procesos preliminares a la propuesta de un proyecto arquitectónico en la empresa WETCH, con la finalidad de constatar los perfiles e historias de usuario.
- **Búsqueda de información científica:** Esta técnica permite sustentar teóricamente el desarrollo del presente trabajo de fin de máster.

4.3. Metodología de desarrollo de software

Dadas las características del software que se desea construir, se propone aplicar de modo simplificado la metodología de desarrollo ágil Programación Extrema (XP), por las siguientes razones:

- **Equipo de desarrollo:** Para ejecutar este proyecto, el equipo de desarrollo está limitado al número de integrantes, ya que se redujo a un desarrollador y el jefe de departamento de cálculo arquitectónico, y la metodología XP propone la programación en parejas e incluir al usuario final como parte del equipo de desarrollo.
- **Modelo:** Este proyecto requiere trabajar con entregas de software en el menor tiempo posible cumpliendo paulatinamente un ciclo de vida en cada fase. La metodología XP propone un modelo iterativo e incremental con funcionamiento en cascada, es por esto que son posibles las entregas de software durante la ejecución de cada iteración.
- **Velocidad:** El trabajo final de máster exige la realización del software en un plazo máximo de 3 meses, así, XP propone el desarrollo ágil, velocidad de reacción para la implementación y los cambios con reducida documentación.

Seguidamente, se detalla las fases de la metodología XP y los elementos que aplica este proyecto.

FASE DE ANÁLISIS O PLANEACIÓN. En esta fase, se interactúa con el cliente para recopilar información y definir los requerimientos del sistema. (Ver capítulo II de este documento).

FASE DE DISEÑO. En esta fase se determina la arquitectura de la aplicación, plataforma de desarrollo, experiencia de usuario, modelos conceptuales y tarjetas simplificadas CRC. (Ver capítulo III de este documento).

FASE DE IMPLEMENTACIÓN O CODIFICACIÓN. En esta fase se configura el entorno de desarrollo y se codifican los algoritmos con la tecnología y arquitectura propuestas, además se ejecutan iteraciones para presentar los entregables a lo largo de las PEC's (Ver capítulo IV de este documento)

FASE DE VALIDACIÓN O PRUEBAS. En esta fase, se ejecutan las pruebas para determinar fallos o errores en la aplicación y corregirlos. (Ver capítulo V de este documento)

5. Planificación

La planificación del trabajo final de máster está determinada en 4 fases, cada una pertenece a las diferentes PECs programadas en el curso, en las cuales constan las actividades entregables con sus correspondientes fechas.

5.1. Entregas

Entregas	Actividad Principal	Fecha inicio	Fecha fin
PEC 1.	Propuesta y definición formal del proyecto	17/02/2021	02/03/2021
PEC 2.	Fundamentación teórica, análisis y diseño de software.	03/03/2021	31/03/2021
PEC 3.	Fase de codificación de software. Primera versión.	01/04/2021	08/05/2021
PEC 4.	Entrega final de software, documentación y publicación.	10/05/2021	07/06/2021

Tabla 1: Resumen de entregas del TFM.
 Fuente: Entorno Virtual de Aprendizaje de la Universidad Oberta de Catalunya

5.2. Hitos

En la siguiente tabla, se presentan los hitos más importantes de la planificación del presente trabajo de fin de master, aquí se estima la duración de las actividades en días y horas, cada día representa 5 horas de trabajo empleado. Se debe tomar en cuenta, que para ejecutar el proyecto se ha considerado el trabajo de los días sábado como imprevistos.

Entregas	Actividad	Duración/día	Duración/hora	Inicio	Fin
PEC 1.	Elección del tema	3	15	17/02/2021	19/02/2021
	Propuesta formal del tema y sustentación.	2	10	20/02/2021	22/02/2021
	Redacción de las páginas preliminares de la memoria.	1	5	23/02/2021	23/02/2021
	Fundamentación teórica TFM: Introducción, descripción, objetivos	3	15	24/02/2021	26/02/2021
	Fundamentación teórica TFM: Metodología, Planificación.	3	15	27/02/2021	02/03/2021
PEC 2.	Fundamentación teórica TFM: Marco teórico	4	20	03/03/2021	06/03/2021
	Definición de requerimientos funcionales.	3	15	08/03/2021	10/03/2021
	Definición de roles de usuario	2	10	11/03/2021	12/03/2021
	Diseño de la arquitectura de la aplicación	4	20	13/03/2021	17/03/2021
	Definición del modelo de	5	20	18/03/2021	23/03/2021

	datos				
	Diseño de prototipos/UX	3	15	24/03/2021	26/03/2021
	Documentación de la memoria de trabajo	4	20	27/03/2021	31/03/2021
PEC 3.	Correcciones al diseño	1	5	01/04/2021	01/04/2021
	Configuración del entorno	2	10	03/04/2021	05/04/2021
	Instalación de paquetes y librerías	2	10	06/04/2021	07/04/2021
	Estructuración de código	1	5	08/04/2021	08/04/2021
	Primera iteración: Estructuración del backend	2	10	09/04/2021	10/04/2021
	Segunda iteración: Creación y prueba de la API-REST.	4	20	12/04/2021	15/04/2021
	Tercera iteración: Estructuración front end	2	10	16/04/2021	17/04/2021
	Cuarta iteración: Navegación principal y desarrollo del módulo de usuarios.	10	50	19/04/2021	29/04/2021
	Quinta iteración: Codificación del módulo de prefactibilidad	5	25	30/04/2021	05/05/2021
	Documentación de la memoria de trabajo	2	10	06/05/2021	08/05/2021
	PEC 4.	Correcciones de codificación	5	25	10/05/2021
Ejecución de pruebas		5	25	15/05/2021	20/05/2021
Documentación de la memoria de trabajo		4	20	21/05/2021	25/05/2021
Presentación académica		3	15	26/05/2021	28/05/2021
Presentación pública		3	15	29/05/2021	01/06/2021
Presentación de video		3	15	02/06/2021	04/06/2021
Autoinforme de evaluación		2	10	05/06/2021	07/06/2021

Tabla 2. Hitos de la planificación del TFM

5.3. Diagrama de Gantt

En el siguiente diagrama de Gantt se muestra las actividades planificadas de acuerdo a lo propuesto en los hitos de la planificación. Además, se ha considerado programar las actividades en cuatro categorías según las entregas de la asignatura de Trabajo de fin máster.

Considere la siguiente nomenclatura para la planificación de tareas en el diagrama de Gantt.

Tarea  Resumen 

Figura 1. Nomenclatura de actividades – Diagrama de Gantt
Elaborado por: Diana Morocho.

Asistente de cálculo de prefactibilidad para construcción de vivienda
 Máster Universitario en Desarrollo de sitios y aplicaciones web
 Diana Alexandra Morocho Puchaicela

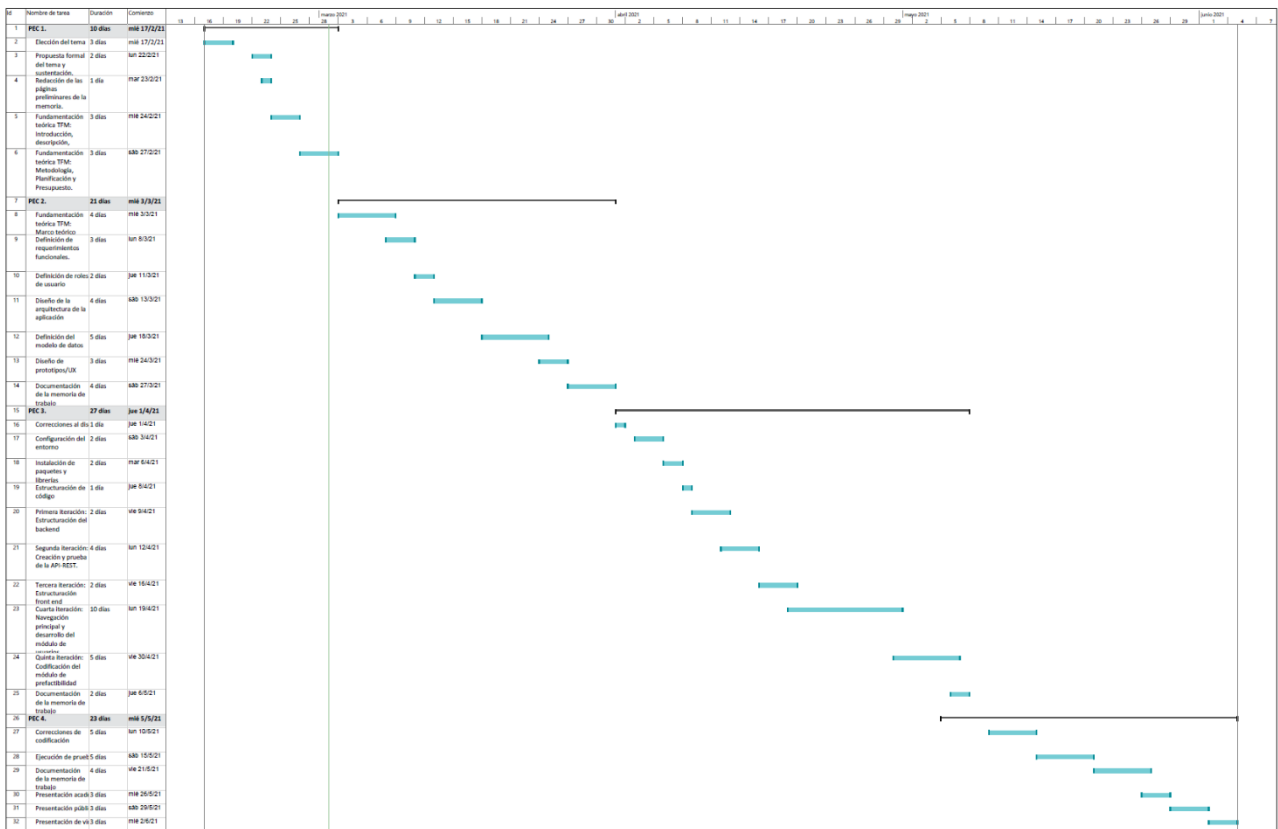


Figura 2. Diagrama de Gantt – Planificación.
 Elaborado por: Diana Morocho.

6. Presupuesto

Para el cálculo del presupuesto, se consideró categorizar los recursos en fases del desarrollo del software, bienes, servicios e imprevistos, que son acumulados para obtener la totalidad de la inversión en dólares americanos.

Código	Concepto	Unidad	Cantidad	P.U	Importe
001	Talento humano para el análisis de software, incluye: Requerimientos, Experiencia de Usuario.	Horas	120	\$5.00	\$600.00
002	Talento humano para el diseño del software, incluye: documentación formal del modelo de datos, diseño de experiencia de usuario, diseño de identidad gráfica, árbol de navegación, lineamientos para el diseño responsive, creación de guía de estilo.	Horas	100	\$5.00	\$500.00
003	Talento humano para la implementación del sitio web que incluye, la codificación de los módulos: usuarios, rubros, tipos de vivienda, acabados, tipos de acabados, cálculo de prefactibilidad, pruebas de funcionalidad.	Horas	155	\$5.00	\$775.00
004	Implantación del sitio web, incluye: migración de datos, pruebas de funcionalidad en producción.	Horas	30	\$5.00	\$150.00
005	Servicio de dominio y alojamiento web.	-	1	\$150.00	\$150.00
006	Servicios adicionales: internet, llamadas telefónicas, entregables impresos.	-	1	\$100.00	\$100.00
SUBTOTAL					\$2275.00
IVA 12%					\$273.00
TOTAL					\$2548.00

Tabla 3. Presupuesto del proyecto.
Elaborado por: Diana Morocho.

CAPÍTULO II. ANÁLISIS

1. Marco teórico

En la actualidad, con el crecimiento urbano de las poblaciones del mundo, la arquitectura y el urbanismo enfrentan grandes retos en innovación de proyectos de construcción, tanto para prever la calidad urbana y calidad de vida para los residentes.

En Ecuador, el crecimiento urbano no es la excepción, desde 1970 este factor incrementó de tal manera que, de acuerdo al último censo realizado, más del 60% de la población habita en zonas urbanas, convirtiendo estas cifras en un factor de oportunidad para los profesionales de la construcción que buscan el crecimiento ordenado de las ciudades de acuerdo a la normativa vigente.

Es así que, previo a la ejecución de un proyecto de construcción de vivienda urbana, dentro del ciclo de vida de un proyecto arquitectónico, se encuentra el proceso de cálculo de prefactibilidad, que permiten contextualizar una obra de construcción de acuerdo a los requerimientos del usuario con el área de construcción disponible en una localidad determinada. En la actualidad, debido a la falta de aplicaciones que automaticen estos procesos de arquitectura, los estudios preliminares se realizan mediante el uso de hojas de cálculo, de aquí que surge la necesidad de agilizar el proceso elevándolo a herramientas modernas, accesibles a la web que permitan vincular los proyectos de construcción con los clientes.

Por lo tanto, en la lectura de los siguientes apartados, encontrará los mecanismos que ofrece la tecnología Single Page Application como herramienta moderna y estructurada para el desarrollo de aplicaciones rápidas y fluidas, cuya elección para la ejecución del presente proyecto de cálculo de prefactibilidad de construcción se fundamenta en la posibilidad del consumo de datos mediante API-REST. Además, se presenta una síntesis de la normativa vigente en el Ecuador para proyectos de construcción de vivienda.

1.1. Single Page Application (SPA)

Según la comunidad de Desarrolladores de Google, una aplicación de una sola página o SPA “es una aplicación web o sitio web que carga todos los recursos necesarios para navegar por el sitio en la primera carga de la página” (Google Analytics, 2021), que se relaciona con otro concepto propuesto en el artículo científico de Madhuri, donde menciona que: “una SPA está conformada por componentes individuales que se pueden reemplazar o actualizar de forma independiente, sin recargar toda la página, por lo que no es necesario volver a cargar la página completa en todas las acciones del usuario” (Madhuri, Balkrishna, & Anushree, 2015, pág. 1). Estas definiciones permiten concluir que una de las principales ventajas de usar una aplicación basada en SPA, es la posibilidad de actualizar la data de la página completa y su

navegabilidad, sin necesidad de recargar la página completa en cada acción de usuario, así como lo exige el ciclo de vida de una página web tradicional basada en la solicitud – respuesta HTTP¹.

Un framework para el diseño de SPA's es Angular, que presenta una arquitectura basada en conceptos fundamentales como: Módulos, Componentes, Servicios e inyección de dependencias y Enrutamiento, que permite organizar internamente componentes mediante módulos. (Angular, 2021). En la figura 3, se puede visualizar la interacción del usuario con un módulo de Angular, cuyo servicio le permite realizar la gestión de la data mediante el consumo de una API a una base de datos externa.

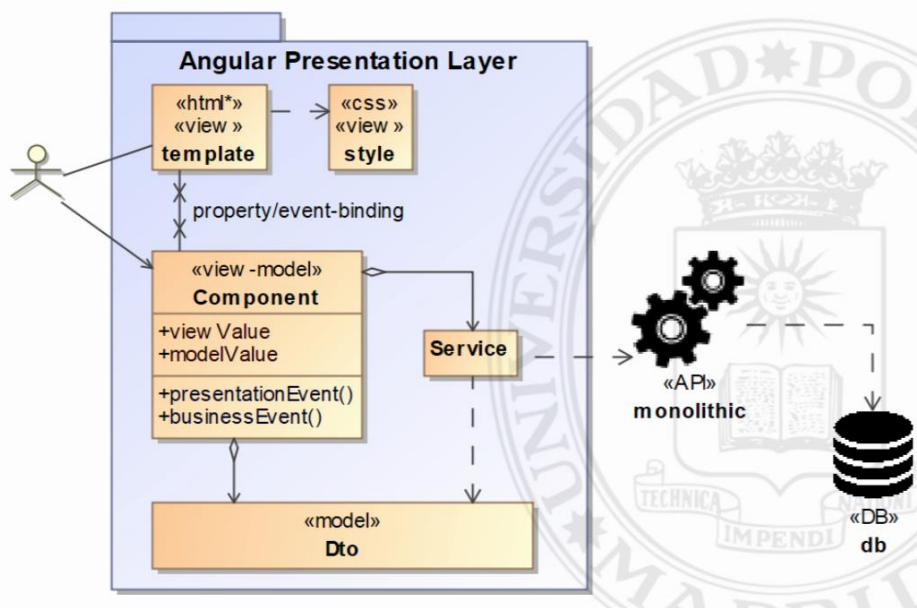


Figura 3. Arquitectura Angular

Fuente: Bernal J. (2020). Arquitectura y patrones web. Recuperado de: <https://github.com/miw-upm/apaw/releases/tag/v1.4.0-Release>

La documentación oficial de Angular menciona varios beneficios del uso de SPA's, sin embargo, para efectos de la construcción del presente proyecto, se menciona los principales:

- Mejoran la experiencia de usuario, evitando recargar desde el servidor cada interacción del usuario con el sitio web.
- La aplicación se siente más rápida puesto que utiliza menor cantidad de recursos como ancho de banda.
- La aplicación es mucho más sencilla de implementar en producción, gracias a su mecanismo de versionamiento.
- Agiliza la lógica de negocio al procesarse en el lado del cliente.
- Disminuye la cantidad de peticiones request en el servidor.

¹ **HTTP**. De sus siglas en inglés: "Hypertext Transfer Protocol", es el nombre de un protocolo el cual nos permite realizar una petición de datos y recursos, como pueden ser documentos HTML. (MDN Web Docs, 2021)

1.2. Normativa de construcción en Ecuador.

La Norma Ecuatoriana de la Construcción (NEC) tiene como finalidad regular los procesos que permiten cumplir con las exigencias básicas de seguridad y calidad en todo tipo de edificaciones como consecuencia de las características del proyecto, la construcción, el uso y el mantenimiento (Ministerio de Desarrollo Urbano y Vivienda, 2021). En la figura 4, se visualiza la estructura de los criterios principales que ampara la normativa.

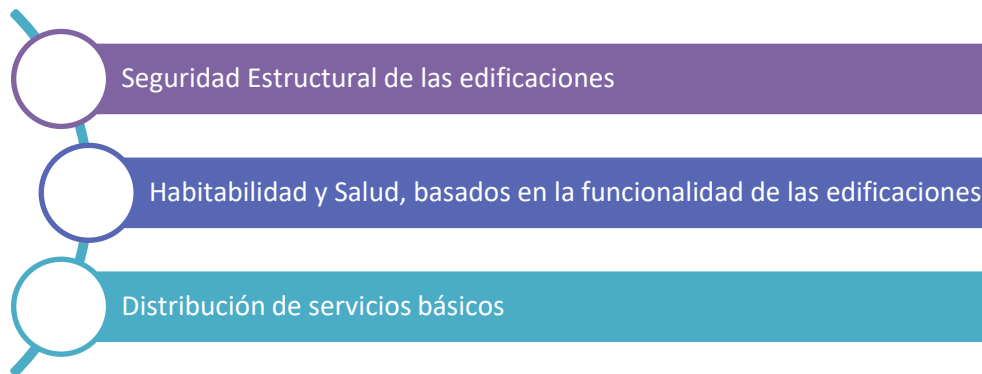


Figura 4. Criterios de la Norma Ecuatoriana de la Construcción
Elaborado por: Diana Morocho

Por lo tanto, de las premisas anteriores, cualquier proyecto de construcción de vivienda urbana está obligado a ajustarse a la NEC y las regulaciones locales expedidas por los Gobiernos Autónomos Descentralizados Municipales.

En la fase inicial del ciclo de vida de un proyecto arquitectónico de construcción de vivienda urbana, se realizan los estudios de prefactibilidad y presupuesto, que involucra la interrelación de los múltiples factores técnicos y métodos que inciden en la determinación del costo total de una construcción. Así, los estudios de prefactibilidad permiten determinar si los proyectos son o no son técnica, financiera, económica, social, y jurídicamente viable. (Salazar, 2011)

Estudios de prefactibilidad

La prefactibilidad como parte importante de la fase de proyectos, permite pasar la idea a algo más concreto, en otras palabras, permite orientar a quienes tienen que tomar decisiones sobre la posibilidad de que dicho proyecto, dada una serie de condiciones sea viable. (Salazar, 2011)

Un estudio de factibilidad debe contemplar varios aspectos, sin embargo, para el desarrollo de la aplicación web de esta memoria, se han tomado los parámetros del análisis de las condiciones del medio, que implica determinar la disponibilidad de insumos, disponibilidad de cantidad, calidad y costos de servicios básicos, servicios habitacionales, disponibilidad de materias primas, disponibilidad de mano de obra, requerimientos de la planta, disponibilidad de los requerimientos, entre otras.

2. Contenidos

En este documento encontrará las fases de la construcción del software mediante el apoyo de la metodología de desarrollo ágil Extreme Programming (XP) y los artefactos desarrollados en cada paso. Es necesario mencionar que la aplicación web requiere calcular la prefactibilidad en un proyecto de construcción de vivienda y para ello la aplicación cuenta con los siguientes módulos:

- **MÓDULO DE PREFACTIBILIDAD:** Para calcular la prefactibilidad de un proyecto de vivienda, es necesario gestionar las entidades: Área de vivienda, Tipo de Vivienda, Item Prefactibilidad, para mostrar resumen de costos y total de componentes.
- **MÓDULO DE PROYECTO:** Para calcular la prefactibilidad de un proyecto de vivienda, es necesario gestionar las entidades: Encabezado, Área de construcción, Dimensiones adicionales, Plan de Necesidades, Item de plan de necesidades, Área, Tipo de Área, Acabado, Tipo de Acabado, Ambiente, Tipo de ambiente, Tipo de Edificación, Unidades de Vivienda, de tal manera que permita generar una vista de datos generales, terreno, construcción, estudios, inversión total y esquema.
- **MÓDULO DE USUARIO:** En este módulo se determinan tipos de usuario.

En cuanto al apartado 3 de este capítulo, se plantea el análisis de los requerimientos de software, donde se propone la recolección de información relevante al desarrollo mediante el planteamiento de historias de usuario, categorizadas de acuerdo un índice de valoración. De aquí, también se obtienen los perfiles de usuario, donde se detallan las acciones de cada uno y su relación con la aplicación web.

3. Requerimientos

3.1. Índice de valoración para historias de usuario

Para la extracción de los requerimientos en la metodología XP, se aplica el artefacto “Historias de usuario”, que “son documentos usados para la especificación de los requisitos”. (Echeverry & Delgado, 2021). En el formato de historia de usuario utilizada, se aplicó un índice de valoración para los campos “Prioridad de Negocio”, “Riesgo de Desarrollo”, “Puntos estimados”, e “iteración asignada”, que permite realizar una estimación medible y controlable del proyecto.

PRIORIDAD DE NEGOCIO		
La prioridad de negocio, es el grado de prioridad para el desarrollo de la historia de usuario. (Leteiler & Penadéz, 2006) El índice de estimación se define en tres valoraciones cualitativas		
Alto	Medio	Bajo
Significa que la historia de usuario se encuentra entre las primeras que se debe desarrollar (etapa inicial del desarrollo). Implica mayor preferencia de desarrollo	Significa que la historia de usuario puede desarrollarse con un nivel de prioridad media. Implica menor preferencia en el orden de desarrollo.	Significa que esta historia de usuario está ubicada entre las historias que pueden codificarse en la etapa final del desarrollo. Implica preferencia mínima.

Tabla 4. Índice de valoración del campo “Prioridad de negocio”
Elaborado por: Diana Morocho

RIESGO DE DESARROLLO		
El riesgo de desarrollo, se define en el riesgo que afronta el equipo de desarrollo para satisfacer los requerimientos del cliente. (Leteiler & Penadéz, 2006)		
Alto	Medio	Bajo
Implica que la historia de usuario debe satisfacer los requerimientos del cliente con un porcentaje igual o mayor al 80%. [80%-100%]	Implica que la historia de usuario debe satisfacer los requerimientos del cliente con un porcentaje igual o superior al 45%. [45%-79%]	Implica que la historia de usuario satisfaga los requerimientos del cliente con un porcentaje menor al 45%. [0%-44%]

Tabla 5. Índice de valoración del campo “Riesgo de desarrollo”
Elaborado por: Diana Morocho

PUNTOS ESTIMADOS																				
El riesgo de desarrollo, se define en el riesgo que afronta el equipo de desarrollo para satisfacer los requerimientos del cliente. (Leteiler & Penadéz, 2006)																				
Puntos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Horas de desarrollo	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
Nro. Semana	Semana 1				Semana 2				Semana 3				Semana 4							

Tabla 6. Índice de valoración del campo “Puntos estimados”
Elaborado por: Diana Morocho

3.2. Historias de usuario aplicadas

Las historias de usuario se aplicaron al personal que labora en la gerencia de la empresa de arquitectura y construcción Wetch, quienes tienen una relación directa con el flujo de procesos para el cálculo de prefactibilidad en un proyecto de construcción de vivienda. El proceso de recolección consistió en entablar un diálogo directo con las personas involucradas, así como la observación directa de las acciones realizadas al proceso.

Si bien el cliente no fue quien escribió personalmente las historias de usuario, fue el quien diseñó su contenido y dirigió la redacción de las mismas debido a que no poseía los conocimientos necesarios en formato para elaborarlas y mucho menos estimarlas.

A continuación, se muestra las historias de usuario que se obtuvieron del levantamiento de información.

HISTORIA DE USUARIO		
Nro. 1	Nombre de historia: Insumos para el cálculo de prefactibilidad de construcción de vivienda	
Fecha: 13 de marzo de 2021		Usuario: Wilson Tapia
Prioridad de Negocio: Alto (Alto/Medio/Bajo)	Riesgo de desarrollo: Alto (Alto/Medio/Bajo)	Puntos estimados: 15
<p>Descripción: El arquitecto, con la ayuda de las características iniciales del área de construcción y la idea de proyecto, se prepara para realizar el cálculo de prefactibilidad de construcción de vivienda, que requiere determinar previamente los factores que influyen en el cálculo, que pueden incluir entre otros:</p> <ul style="list-style-type: none"> • Rubro: Corresponde a una actividad comercial productiva, orientada a la elaboración de bienes inmuebles y prestación de servicios de construcción. Ejemplos de rubro son: colocación de mesa de granito, desbanque manual de tierra, colocación de cornisas de cielo raso, etc. Los rubros se clasifican por categorías, por ejemplo: De un sistema contra incendios, se podría realizar instalación de gabinetes, tuberías, sensores, etc. Por lo tanto, se codifica cada rubro por tipo. Un rubro contiene los parámetros: unidad de medida, precio unitario, precio de mano de obra, precio operativo y precio de equipamiento, que, sumados entre sí, se obtiene el precio total por rubro. • Costos: Una obra de construcción es un proceso productivo donde se ensamblan y se transforman los materiales en productos terminados. En una construcción, existen diferentes tipos de costos entre los que se puede mencionar: Costos directos, gastos generales, costos indirectos y costos comerciales, donde cada uno se calcula de diferente manera. 		
<p>Observaciones:</p> <ul style="list-style-type: none"> • Los costos directos se relacionan con la compra de materiales y productos manufacturados. Utilización de personas y equipos para realizar labores de colocación, transporte, ensamblaje, etc. Se calculan por planos • Los gastos generales son los sueldos u honorarios a profesionales que coordinan y dirigen el proceso de construcción, así como auxiliares. Se calculan de acuerdo a la experiencia del constructor y la oferta del mercado. • Costos indirectos, elaboración de diseños o estudios técnicos, impuestos asociados a la actividad constructora, servicios de imposición gubernamental. Se calcula por tarifas • Costos comerciales: Intereses del capital y costos asociados. Relacionados con las comisiones de ventas. Se calculan por tarifas y determinantes del mercado. 		

Tabla 7. Historia de usuario 1. Insumos para el cálculo de prefactibilidad de construcción de vivienda.
Elaborado por Diana Morocho

HISTORIA DE USUARIO		
Nro. 2	Nombre de historia: Cálculo de prefactibilidad de construcción de vivienda	
Fecha: 13 de marzo de 2021		Usuario: Wilson Tapia
Prioridad de Negocio: Medio (Alto/Medio/Bajo)	Riesgo de desarrollo: Alto (Alto/Medio/Bajo)	Puntos estimados: 15
<p>Descripción:</p> <ul style="list-style-type: none"> De acuerdo al contexto de las construcciones en el Ecuador, la empresa Wetch, considera en el cálculo de prefactibilidad los costos directos, gastos generales y costos indirectos El cálculo dependerá de los rubros que se requieran puesto que son diferentes para cada proyecto arquitectónico. Ejemplo de aplicación: Para preparar el área de construcción, un lote de terreno posee condiciones donde previamente se debe realizar un desbanque con maquinaria y el aplanamiento de terreno, el cálculo se realiza con las unidades requeridas por metro cuadrado. El presupuesto de una obra final se obtiene de la suma de los costos involucrados. Adicionalmente, es importante considerar las tarifas de impuestos que en el Ecuador están fijadas al 12% 		
<p>Observaciones:</p> <ul style="list-style-type: none"> El cálculo de los rubros a utilizar varía de acuerdo a las características de cada proyecto. 		

Tabla 8. Historia de usuario 2. Cálculo de prefactibilidad de construcción de vivienda
Elaborado por Diana Morocho

HISTORIA DE USUARIO		
Nro. 3	Nombre de historia: Insumos para el cálculo de prefactibilidad de construcción de vivienda	
Fecha: 14 de marzo de 2021		Usuario: Wilson Tapia
Prioridad de Negocio: Medio (Alto/Medio/Bajo)	Riesgo de desarrollo: Medio (Alto/Medio/Bajo)	Puntos estimados: 20
<p>Descripción:</p> <p>Es necesario determinar de manera preliminar las variables que intervienen en el análisis.</p> <ul style="list-style-type: none"> Área: Hace referencia a las características físicas del proyecto, es decir, el área de construcción, este factor debe estar categorizado por tipos: áreas mínimas, normales, grandes y de lujo. Ambiente: Entendiendo al ambiente como el espacio de construcción deseado por el cliente para desarrollar una actividad específica. Ejemplo: Garaje, cocina, baño social, etc. Permite determinar mediante una sumatoria ambientes el área total de construcción. Costo de suelo urbano: Precio del costo de terreno destinado para el área de construcción. Acabado de construcción: Son todos aquellos revestimientos que se colocan sobre una superficie de obra negra, es decir, materiales que se colocan sobre pisos, muros, plafones, azoteas, etc., en la construcción. En la empresa, se consideran 5 tipos de acabado: acabado de lujo, primera, segunda, tercera y cuarta calidad. 		
<p>Observaciones:</p> <ul style="list-style-type: none"> Los ambientes deben ser categorizados por tipo de área, ejemplo: El área de construcción de un garaje en condiciones mínimas no posee las mismas magnitudes que un garaje de lujo. El plan de necesidades en la prefactibilidad, se entiende como el conjunto de ambientes deseados y las características asociadas en su cálculo. De acuerdo a los expertos de la construcción y las normas técnicas establecidas, los ambientes de construcción deben poseer un área mínima dada en metros cuadrados. 		

Tabla 9. Historia de usuario 3. Insumos para el cálculo de prefactibilidad de construcción de vivienda
Elaborado por Diana Morocho

HISTORIA DE USUARIO		
Nro. 4	Nombre de historia: Cálculo de prefactibilidad de construcción de vivienda	
Fecha: 14 de marzo de 2021		Usuario: Wilson Tapia
Prioridad de Negocio: Medio (Alto/Medio/Bajo)	Riesgo de desarrollo: Medio (Alto/Medio/Bajo)	Puntos estimados: 40
<p>Descripción:</p> <ul style="list-style-type: none"> En primera instancia se calcula el área de construcción disponible de acuerdo a las características del proyecto. Que se obtiene de restar las áreas de retiro frontal, posterior, lateral derecho e izquierdo. Una vez determinada el área de construcción final, se valida con las áreas de los requerimientos de los ambientes deseados. Por ejemplo, si un cliente requiere de la construcción de 2 dormitorios, se deberá calcular el producto del área del dormitorio por dos. Una vez ingresados todos los ambientes deseados se realiza la suma de todas las áreas obteniendo el subtotal, factor de área de circulación y paredes y área total. De acuerdo a la naturaleza del proyecto, es el cliente el que propone el tipo de edificación y unidades de vivienda para el estudio. 		
<p>Observaciones:</p> <ul style="list-style-type: none"> Del estudio de prefactibilidad la información útil para la empresa es: área del terreno por unidad de vivienda, área total del terreno, costo del terreno por metro cuadrado, área de ocupación del suelo, coeficiente de ocupación del suelo y costo total del terreno. Se deben incluir en el cálculo los datos de construcción, como: unidades de vivienda, área de construcción por unidad de vivienda, número de plantas de edificación, costo de construcción, costo de dirección técnica y costo total de construcción. Adicionalmente, se incluye costos indirectos como los relacionados con los estudios técnicos requeridos, costos de los estudios en función del área de construcción, y costo total de los estudios y tarifas gubernamentales. 		

Tabla 10. Historia de usuario 4. Cálculo de prefactibilidad de construcción de vivienda
Elaborado por Diana Morocho

HISTORIA DE USUARIO		
Nro. 5	Nombre de historia: Tipos de usuario	
Fecha: 14 de marzo de 2021		Usuario: Wilson Tapia
Prioridad de Negocio: Medio (Alto/Medio/Bajo)	Riesgo de desarrollo: Medio (Alto/Medio/Bajo)	Puntos estimados: 40
<p>Descripción:</p> <p>Se identifica a los siguientes usuarios:</p> <ul style="list-style-type: none"> Gerente General: Persona encargada de la administración de los proyectos de construcción. Arquitecto: Profesional de la construcción que realiza los estudios de prefactibilidad. 		
<p>Observaciones:</p> <ul style="list-style-type: none"> El gerente general, es quien designa los proyectos arquitectónicos para el desarrollo de los estudios. 		

Tabla 11. Historia de usuario 5. Tipos de usuario
Elaborado por Diana Morocho

3.3. Requerimientos Funcionales

Un requerimiento es una característica que el sistema debe cumplir para ser aceptado por el cliente. En la aplicación para la empresa Wetch se propone desarrollar las siguientes características útiles.

REQUERIMIENTOS FUNCIONALES			
Referencia	Requerimiento Funcional	Visibilidad	Historia de usuario
La aplicación web permite			
RF - 1	El acceso a la aplicación mediante los roles de usuario: administrador, planificador, cliente.	Evidente	5
RF - 2	Al usuario administrador, realizar las operaciones de CRUD para los demás roles de usuario.	Evidente	5
RF - 3	Al usuario administrador y arquitecto, realizar las acciones de CRUD para el área de vivienda	Evidente	1, 2
RF - 4	Al usuario administrador y arquitecto, realizar las acciones de CRUD para prefactibilidad, área, tipo de área, acabado, tipo de acabado.	Evidente	3, 4
RF - 5	Calcular el área útil, área de ocupación del suelo, coeficiente de ocupación, pisos y costo de terreno.	Oculto	4
RF - 6	Calcular costo directo, costo indirecto como requisitos determinantes en el cálculo de prefactibilidad	Oculto	2
RF - 7	Generar una vista de datos generales, terreno, construcción, estudios, inversión total y esquema.	Evidente	1, 3
RF - 8	A los usuarios, acceder al sistema mediante un perfil de usuario que permita su identificación.	Evidente	5
RF - 9	Al usuario arquitecto, agregar los datos del área de construcción donde se ejecutará el proyecto.	Evidente	4

Tabla 12. Requerimientos funcionales de la aplicación
Elaborado por: Diana Morocho

3.4. Requerimientos No Funcionales

Referencia	Requerimientos no funcionales
RNF - 1	La aplicación web está desarrollada con herramientas no privativas.
RNF - 2	El diseño de interfaces de usuario de la aplicación debe desarrollarse lo más intuitivo como sea posible.
RNF - 3	El diseño web de la aplicación debe ser responsivo, basado en el enfoque mobile first.
RNF - 4	Debe poseer algún grado de escalabilidad para facilitar su crecimiento en el futuro.

Tabla 13. Requisitos no funcionales de la aplicación
Elaborado por: Diana Morocho

4. Perfiles de usuario

“Existen muchas técnicas de investigación de usuarios que se pueden usar durante todo el ciclo de vida del proyecto, ya sea para comprender mejor a los usuarios o para probar su comportamiento en las versiones de un sitio.” (Unger & Chandler, 2009) Si se parte de esta premisa, serán las historias de usuario del apartado de requerimientos los que permitirán determinar los grupos de usuario y su intervención durante el desarrollo del software.

Dicho de otra manera, para determinar los perfiles de usuario que se muestran en la tabla 14, durante la recolección de información en la historia de usuario, se tomó como base la aplicación de los 5 pasos básicos que propone la publicación de Russ Unger en su libro de Diseño de experiencia de usuario, 2009:

1. Definir los grupos de usuario principales.
2. Planificación de la participación del usuario
3. Realizar la investigación
4. Validar las definiciones de los grupos de usuarios
5. Generar requisitos de usuario.

PERFILES DE USUARIO				
Perfil	Función en el sistema	Trato Personalidad Comunicación	Expectativas para mejorar la experiencia de usuario	Usuario
Cliente	Usuario que requiere contratar al profesional de la construcción para la realización de los estudios de prefactibilidad	Es autónomo, le interesa involucrar sus intereses con la ejecución del proyecto de construcción. Propone ideas en función del presupuesto de inversión disponible. Le importa hacer escuchar su opinión.	Permitirle simular el estudio de prefactibilidad de una obra.	Patricia Galarza, 55 años. Docente de educación superior. Casada
Administrador	Es el usuario encargado de la administración general y absoluta del sistema, permitiéndole administrar las funciones relacionadas con el módulo de usuarios, roles, privilegios, gestión de insumos para prefactibilidad.	Le interesa el crecimiento personal y empresarial en el ámbito de la construcción. Personalidad con habilidades comunicativas y excelentes relaciones interpersonales. Está involucrado en diferentes proyectos. Gestiona su equipo de trabajo para asignarle tareas. Es digital y matemático.	Enfocar los esfuerzos en mostrarle la mayor parte de acciones que se realizan en el sistema. Proveerle de una interfaz intuitiva para reconocer rápidamente la ubicación de las cosas.	Wilson Tapia 48 años Gerente general de Wetch Arquitecto y docente de educación superior. Casado Emprendedor

Asistente de cálculo de prefactibilidad para construcción de vivienda

Máster Universitario en Desarrollo de sitios y aplicaciones web

Diana Alexandra Morocho Puchaicela

Arquitecto	Usuario que gestiona los insumos para realizar el cálculo de prefactibilidad.	Presenta habilidades particulares con la proyección, matemáticas y el dibujo. Dispone de una libreta para hacer sus anotaciones. Es flexible. Está en continuo aprendizaje. Es experto y reconocido en su profesión.	Facilitarle la búsqueda de los insumos para realizar los cálculos de prefactibilidad. Mostrarle su ruta de ubicación actual en el sistema. Proveerle de accesos directos a las funciones más utilizadas.	Paulo Román 36 años Arquitecto Soltero
------------	---	--	--	---

Tabla 14. Perfiles de usuario de la aplicación.

Elaborado por Diana Morocho

5. Análisis de mercado

5.1. Contexto social y de mercado

Es importante mencionar que el desarrollo del presente proyecto surge debido a la necesidad existente de automatizar el proceso de cálculo de prefactibilidad como parte del ciclo de vida de un proyecto arquitectónico, ya que en la actualidad las pocas herramientas de similar función no se ajustan a los lineamientos definidos por las normas de construcción del Ecuador y únicamente están disponibles para ser usadas como aplicación de escritorio, de ahí que es importante migrar la información a la web, donde el cliente tiene la posibilidad de acceder a la información.

Si bien la plataforma de la competencia que se menciona a continuación sirve de inspiración, los lineamientos planteados en este proyecto de software están enfocados en la actividad que realiza la empresa Wetch y sus necesidades.

5.2. Productos similares

Obras 6.0 Plus

En el siguiente párrafo, se muestra una descripción oficial del software emitida por Wilson Jaramillo, quien promueve su uso con fines de aprendizaje y parte del material de una capacitación, donde expresa que:

“El software Obras, es una herramienta informática diseñada para fortalecer el desempeño de los profesionales, sean Ingenieros Civiles, Arquitectos u otros relacionados a la planificación, diseño y construcción de obras civiles, permite: elaborar presupuestos de obras; anexo de medición de cantidades de obra por ejecutar; mantener una base de datos de especificaciones técnicas, análisis de precios unitarios, materiales, mano de obra y equipo; elaborar el cronograma valorado de actividades; elaborar la fórmula para reajuste de precios; elaboración de planillas, anexo de medición de cantidades de obra ejecutada; estudio de desagregación tecnológica; libro de obra; avalúo de construcciones usadas y ponderación de proyectos.” (Jaramillo, 2018)

Quercusoft

Es un software de escritorio para la elaboración de presupuestos de obra. Cuenta con características para realizar listados de análisis de precios unitarios para el cálculo de presupuestos. En su sitio oficial se menciona que es usado en varios países de América Latina y cuenta con dos licenciamientos, uno gratuito y otro de pago. (Quercusoft, 2021)

5.3. Estudio de mercado

El mercado de usuarios de la aplicación web, engloba a todas las personas que se relacionan con los procesos y servicios que provee la empresa de arquitectura y construcción Wetch, tanto clientes como el

personal técnico, sin embargo, se piensa a futuro implementar un espacio donde más profesionales de la construcción puedan crear una cuenta de usuario y disponer de la herramienta para realizar los cálculos.

La diferencia entre el software de este estudio comparado con el software Obras 6.0 y Quercusoft, radica en las siguientes razones:

- Obras 6.0 es una herramienta informática de aprendizaje, como un producto que está disponible desde el año 2009 y a pesar de actualizar continuamente los datos mediante versiones su plataforma está orientada a su uso como aplicación de escritorio, lo que limita el alcance de su uso y actualización permanente de información.
- Quercusoft y Obras 6.0 tiene un enfoque de uso como software de escritorio, mientras que la herramienta Wetch está planificada para su producción en entornos web.
- La herramienta Wetch, propone actualizar los procesos y datos en función las propuestas de la Norma Ecuatoriana de la Construcción, esto debido a que cada proyecto varía de acuerdo a sus necesidades.
- Un potencial de las herramientas de la competencia, es que, al ser software con más experiencia en el mercado, le ha permitido crecer en funcionalidades para su uso.

CAPÍTULO III. DISEÑO

1. Arquitectura de la aplicación

Debido a la naturaleza de la aplicación y los cambios permanentes que realiza la normativa ecuatoriana a la que se rige los cálculos de prefactibilidad de construcción, se consideró a la flexibilidad como una característica potencial para el crecimiento eficiente de la aplicación, por ello, se plantea una arquitectura de software en el enfoque Single Page Application con sus principales elementos: Componentes, Vistas, Modelos, Servicios apoyado del patrón de arquitectura de datos Redux y su conexión con el backend.

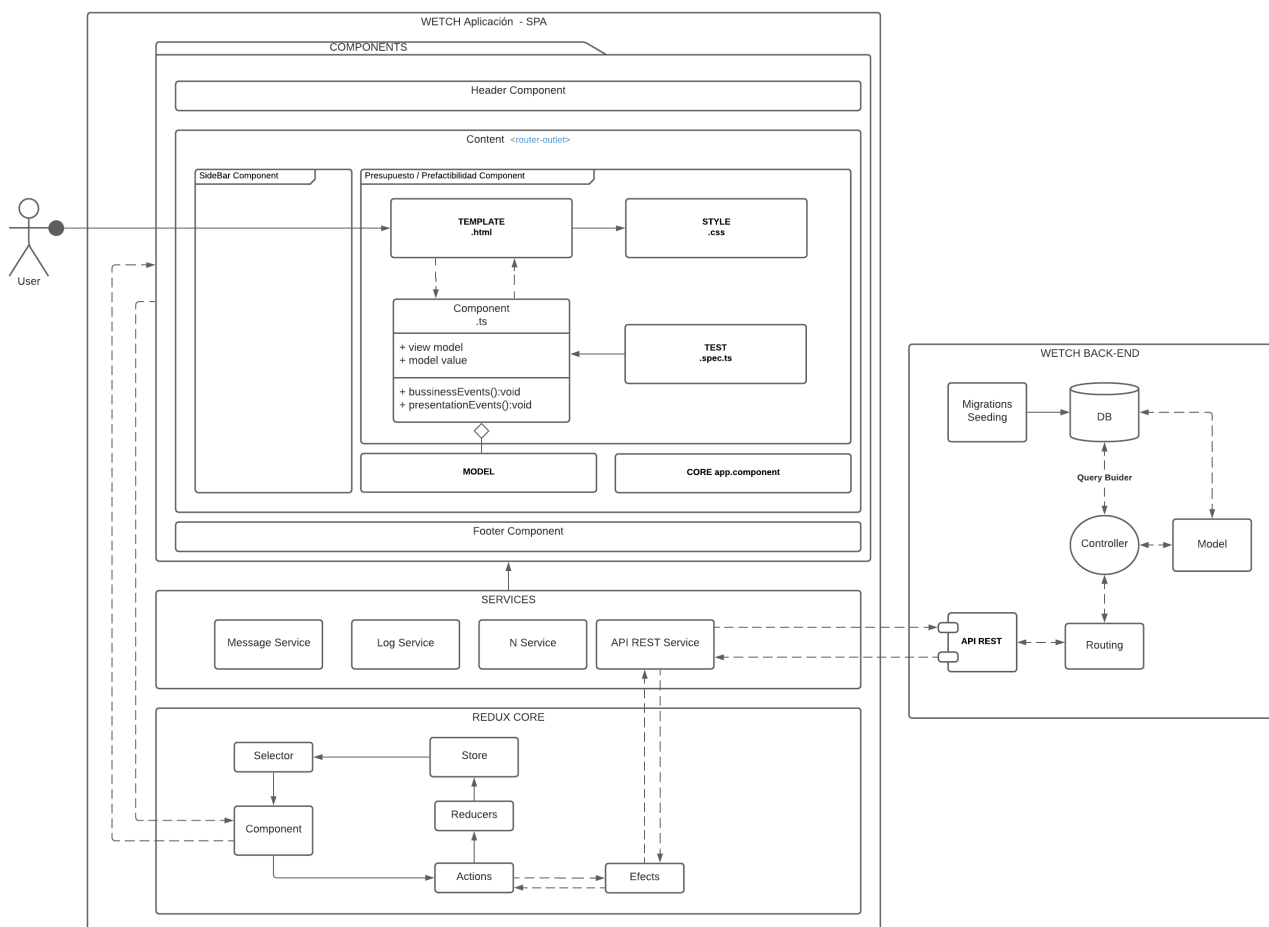


Figura 5. Arquitectura de la aplicación.
Elaborado por Diana Morocho

1.1. Back-end

La comunicación con el backend será mediante la consulta a una API-REST desarrollada con Laravel, un framework PHP moderno, que basa su estructura en clases bajo el patrón arquitectónico Hexagonal y las capas de Dominio, Aplicación e Infraestructura.

Laravel cuenta con un ORM² potente que facilita el acceso a la base de datos Mysql para que puedan ser gestionados como simples objetos y mediante el procesamiento de controladores sea servido al front-end mediante el protocolo de comunicación RESTFull³.

La API será de acceso público una vez dado de alta el usuario en el sistema y así gestionar eficientemente las peticiones

1.2. Front-end

El front-end es el encargado de realizar las diversas llamadas al back-end de acuerdo a la interacción de las acciones realizadas por el usuario en sus interfaces. Está desarrollada con el framework Angular y la generación de templates con lenguaje de marcado HTML⁴ y CSS⁵.

En la figura 5., se puede visualizar la interrelación de los principales elementos del framework Angular (Componentes, Vistas, Modelos y Servicios) como plataforma para generar aplicaciones escalables, mediante el uso de sus bibliotecas para proporcionar las características de enrutamiento, administración de formularios, comunicación cliente-servidor, etc.

Adicionalmente, mediante la integración de la librería NgRx⁶ se cuenta con el patrón redux que a decir de Carlos Azaustre en su artículo para explicar los conceptos básicos 2021, este permite:

- Gestión del estado global y local de la aplicación, que a decir en otras palabras se trata de lo datos que se pueden recibir a través de un API Rest o Web service, e incluso al estado de la interfaz de usuario.
- El estado se almacena en un único store, lo que facilita el flujo de la información y la gestión de datos entre componentes.
- El estado es solo de lectura. Se puede acceder a él únicamente para representarlo en la vista, sin embargo, si se desea modificar se deberá realizar mediante una acción.
- Cambios con funciones puras. Para especificar los cambios en el store se utilizan funciones llamadas reducers. Que no son más que funciones que reciben una acción y el estado inicial para generar un nuevo estado.

² **ORM**. Es un modelo de programación que permite mapear las estructuras de una base de datos relacional. (Muro, 2021)

³ **RESTFull**. Tipo de conexión encargada de manejar datos a través de operaciones CRUD (Get, Post, Update, Delete). (Lopez, 2020)

⁴ **HTML**. Lenguaje de Marcado de Hipertexto, es el código que se utiliza para estructurar y desplegar una página web y sus contenidos. (MDN Web Docs, 2021)

⁵ **CSS**. Es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. (MDN Web Docs, 2021)

⁶ **NgRx**. Es un framework para construcción de aplicaciones reactivas en Angular. (NgRx , 2021)

2. Plataforma de desarrollo

Con la finalidad de optimizar los costos en el desarrollo, se consideró priorizar el uso de herramientas libres, gratuitas y con versiones estables. A continuación, se detalla los recursos de software utilizados.

2.1. Software

FRONT-END

- **Angular 11.2.6:** Es un marco de diseño de aplicaciones y plataforma de desarrollo para la creación de aplicaciones tipo Single Page Application. (Angular, 2021)
- **Typescript 3.2.2:** Es un lenguaje de código abierto que se basa en JavaScript (Typescript, 2021). Es el lenguaje principal para el desarrollo de aplicaciones Angular.
- **Sass 3.6.4:** Lenguaje de extensión CSS (Sass, 2021), para generar el formato del contenido html en la página web.
- **NgRx 11.0.2:** NgRx Store proporciona administración de estado reactivo para aplicaciones Angular inspiradas en Redux. (NgRx Community , 2021)
- **Angular Material 12.0.0:** Librería de estilos basada en la guía de diseño de Material Design para integrarse perfectamente con Angular. (Angular Material, 2021)
- **Fontawesome 5.15.2:** Framework de iconos vectoriales y estilos css. (Fontawesome, 2021)

BACK-END

- **MySql 5.7:** Es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation. (MySql, 2021)
- **Docker 20.10.5:** Es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. (Docker, 2021)
- **Composer 2.0.11:** Es un sistema de gestión de paquetes para programar en PHP el cual provee los formatos estándar necesarios para manejar dependencias y librerías de PHP. (Composer, 2021)
- **Php 7.2.5:** Es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web. (Php, 2021)
- **Laravel 8:** Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP. (Laravel , 2021)

3. Usabilidad/UX

Con el diseño UX (User Experience) se pretende garantizar que los visitantes a un sitio web se relacionen con sus elementos de tal manera que puedan interactuar correctamente y consecuentemente tengan una buena experiencia de uso.

3.1. Formas de interacción

El diseño de interacción comprende el planteamiento de varios pasos, sin embargo, para este estudio, se ha considerado importante mencionar únicamente los relacionados con la navegabilidad de sitio desde los diferentes dispositivos. Por lo tanto, para el diseño de interfaces se han definido los siguientes puntos de quiebre para la visualización del sitio desde cualquier pantalla. (Ver Figura 6.)

Screen Sizes

	Mobile Devices <= 600px	Tablet Devices > 600px	Desktop Devices > 992px	Large Desktop Devices > 1200px
Class Prefix	.s	.m	.l	.xl
Container Width	90%	85%	70%	70%
Number of Columns	12	12	12	12

Figura 6. Screen Sizes and Breackpoints
Fuente: Materialize Css. Recuperado de: <https://materializecss.com/grid.html>

Adicionalmente, se prevé que, para mejorar la experiencia de usuario, se realice en las iteraciones la intervención de los usuarios con el sistema, con el fin de recoger opiniones y punto de vista, respecto de las funciones disponibles y su accesibilidad.

3.2. Sitemap

En la Figura 7. se puede visualizar el mapa del sitio web donde se ha considerado las principales páginas. El acceso empieza por la página de inicio denominada Home, de aquí que se puede vincular al apartado Quienes Somos para mostrar la información empresarial.

Desde el Home, también es posible acceder a la creación de proyecto y sus correspondientes funciones para realizar el cálculo de prefactibilidad con la gestión de las entidades: cliente, acabado de construcción, área de construcción y rubros de construcción.

Adicionalmente, se dispone de un apartado para la gestión del usuario, aquí encontrará las funciones de configuración de proyectos, así como la personalización de su cuenta de perfil.

Es importante mencionar, que en el siguiente diagrama se han omitido las páginas de CRUD de algunas entidades, para simplificar el esquema.

Asistente de cálculo de prefactibilidad para construcción de vivienda
Máster Universitario en Desarrollo de sitios y aplicaciones web
Diana Alexandra Morocho Puchaicela

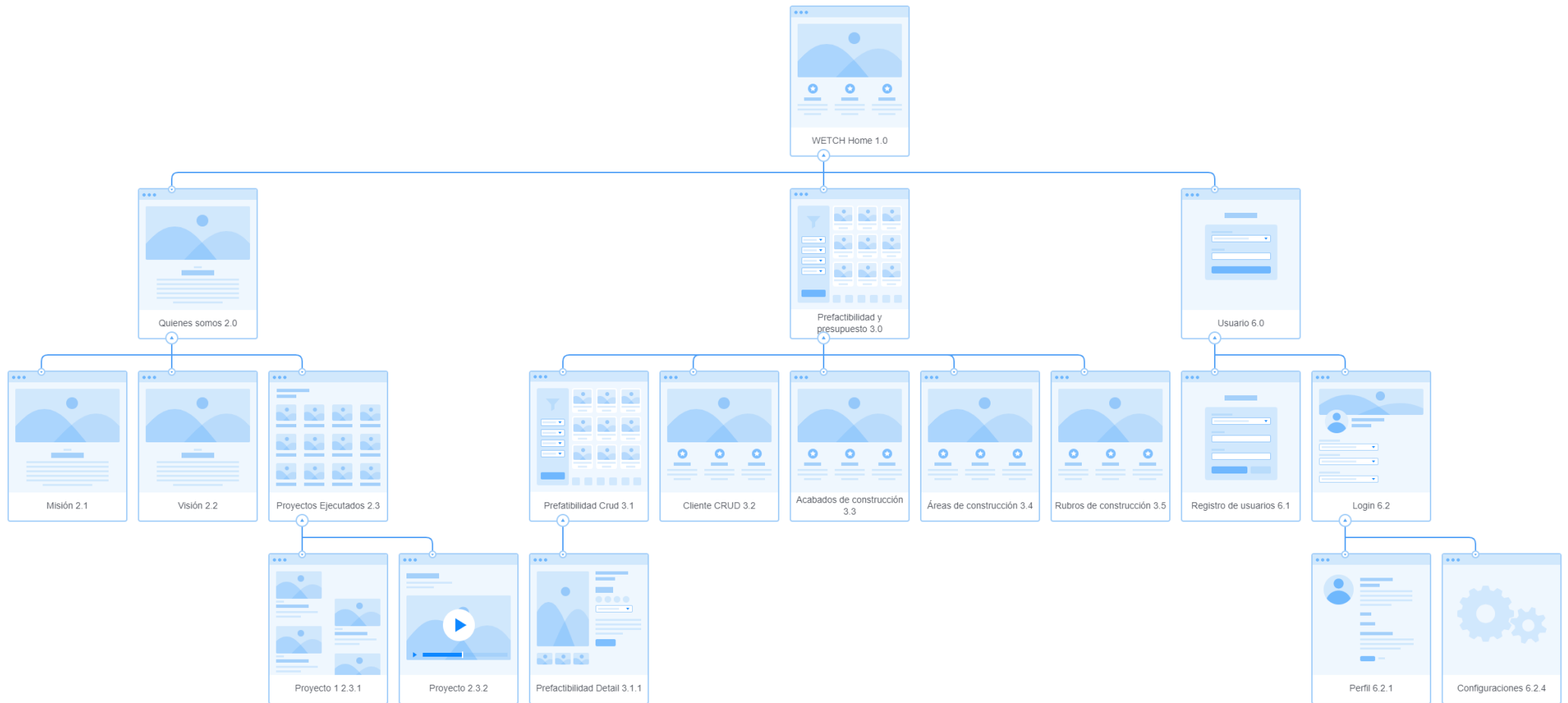


Figura 7. Sitemap de la aplicación web
Elaborado por: Diana Morocho

4. Prototipos

Las siguientes imágenes representan el diseño de Wireframes⁷ como prototipos de baja calidad para las principales funcionalidades del software, partiendo de la aplicación de la técnica Mobile First⁸ del diseño web responsivo.

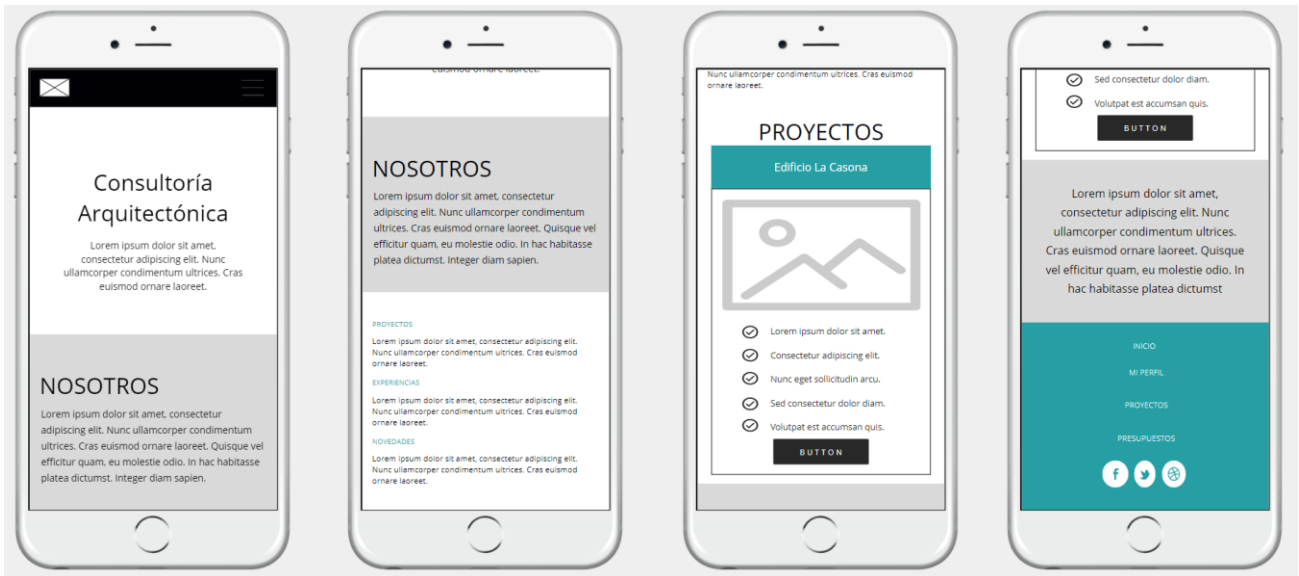


Figura 8. Wireframe mobile - página principal.
Elaborado por Diana Morocho.

⁷ Wireframe. Es una representación visual de la estructura y funcionalidad de una página web o pantalla de aplicación móvil.” (Lucidchart, 2021)

⁸ Mobile First.



Figura 9. Wireframe mobile - página de prefactibilidad.
 Elaborado por Diana Morocho.

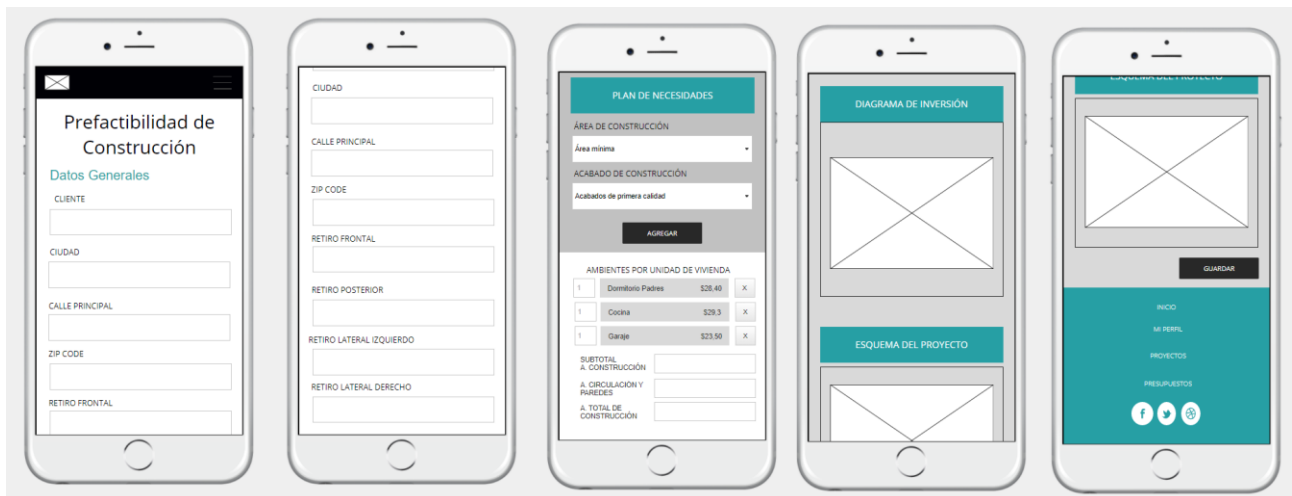


Figura 10. Wireframe mobile - página de prefactibilidad.
 Elaborado por Diana Morocho



Figura 11. Wireframe desktop – página principal
Elaborado por Diana Morocho

INICIO PUBLICACIONES MIS PRESUPUESTOS PERFIL

Prefactibilidad de Construcción

Datos Generales

CLIENTE CIUDAD
CALLE PRINCIPAL CALLE DE INTERSECCIÓN ZIP CODE
RETIRO FRONTAL RETIRO POSTERIOR RETIRO LATERAL IZQ. RETIRO LATERAL DER.

Plan de Necesidades

Área mínima Acabados de primera calidad
TIPOS DE ÁREAS

Ambientes por unidad de vivienda

Cantidad	Ambiente	Area (m2)	Subtotal (m2)	
1	Dormitorio padres	14.2	28.4	<input type="button" value="X"/>
1	Cocina	29.3	29.3	<input type="button" value="X"/>

SUBTOTAL AREA DE CONSTRUCCION
AREA DE CIRCULACION Y PAREDES
AREA TOTAL DE CONSTRUCCION

Esquema del proyecto



Diagrama de inversión

PROYECTOS

- Features
- Promo
- Download

CONTACT

- Find Us
- FAQ
- Help

FOLLOW USSUBSCRIBE

Figura 12. Wireframe desktop – página de prefactibilidad
Elaborado por Diana Morocho

5. Diagrama de clases

En la siguiente imagen se muestra la estructura de la aplicación web mediante el diagrama de clases, que incluye las clases con sus atributos y métodos específicos, sin embargo, para simplificar la lectura no se han incluido los métodos relacionados con el CRUD⁹ de entidades.

En la imagen, se incluyen las entidades para la gestión de Usuarios y Prefactibilidad asociadas a sus insumos para la gestión como áreas de construcción, ambientes, tipos de área de vivienda, rubros y costos indirectos útiles para el cálculo.

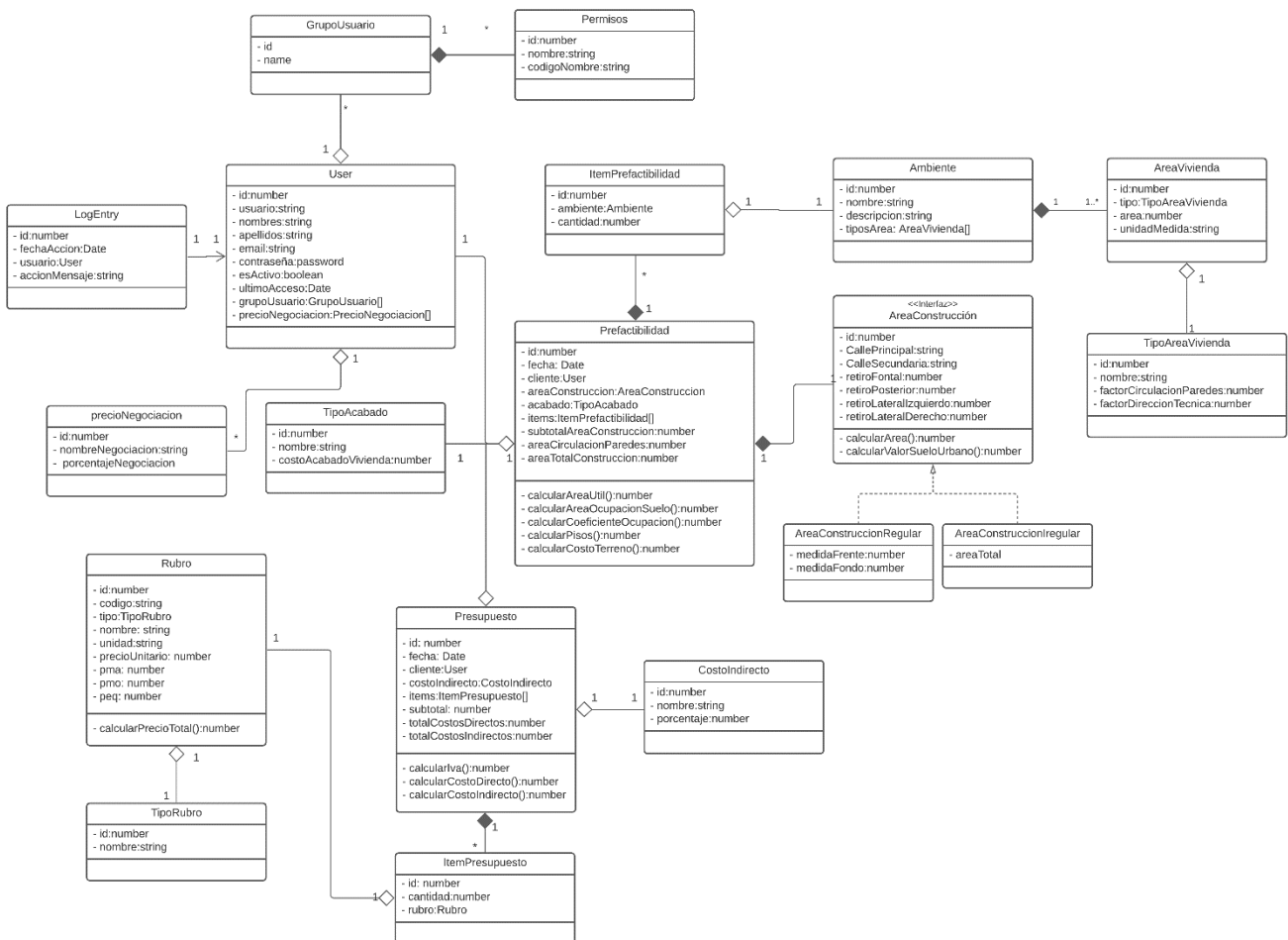


Figura 13. Diagrama de clases de la aplicación
 Elaborado por Diana Morocho.

⁹ CRUD. Acrónimo de “Crear, Leer, Actualizar y Borrar” como funciones de persistencia en un software.

6. Diagrama de casos de uso.

Para mejorar la comprensión del comportamiento del software Wetch, se aplicó el uso del artefacto de casos de uso, puesto que “un caso de uso es una secuencia de interrelaciones entre un sistema y alguien o algo que usa alguno de sus servicios” (Ceria, 2012) y por lo tanto define una funcionalidad concreta, entre tanto las historias de usuario solo muestran la silueta de la tarea a realizarse.

Por otra parte, el diagrama de casos de uso es un elemento de esquematización de UML y, la metodología XP puede adoptar las prácticas del UML, lo que en la actualidad se denomina Modelamiento Ágil.

Es importante mencionar que los casos de uso son de uso alternativo y no reemplazan a las historias de usuario, pero para este estudio resultó útil para analizar, elaborar, comprender el funcionamiento y mejorar la calidad del sistema.

La representación gráfica del uso de requisitos de negocio para el sistema Help Desk se muestra en el siguiente diagrama general de casos de uso. (Ver Figura 15.)

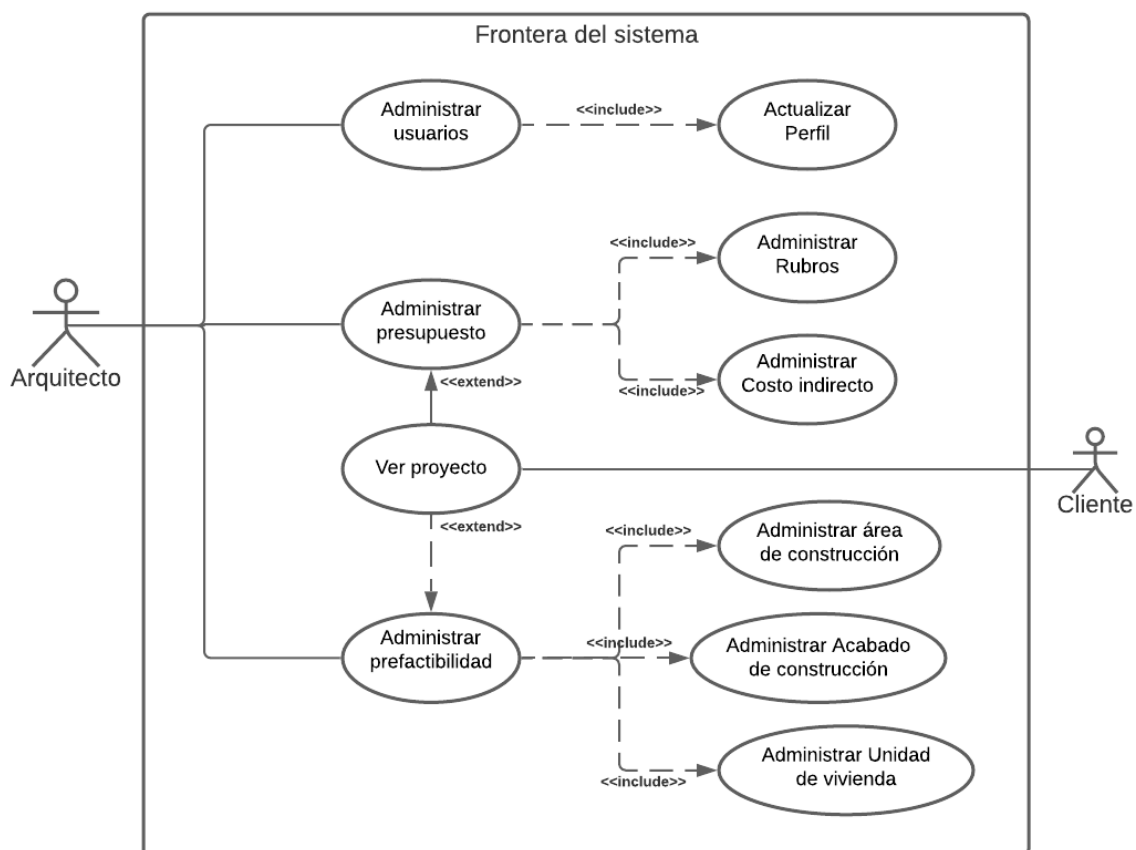


Figura 14. Diagrama de casos de uso.
Elaborado por Diana Morocho

CAPÍTULO IV. IMPLEMENTACIÓN

1. Proceso de desarrollo

Al considerar XP como metodología para el desarrollo de la aplicación, es importante considerar los principios y buenas prácticas que propone esta disciplina de desarrollo. Para ello, en la fase de implementación se estableció la codificación de funcionalidades por prioridad para cada historia de usuario mediante un plan de iteraciones, además, de contar con el diseño de tarjetas CRC como estrategia para modelar la base de datos.

1.1. Plan de iteraciones

El plan de iteraciones es una planificación incremental del desarrollo de software basado en las historias de usuario, con la finalidad de generar ciclos de desarrollo que permitan entregar funcionalidades útiles del sistema. En la siguiente tabla, se muestra la planificación de iteraciones asociadas a los requerimientos funcionales del sistema.

PLAN DE ITERACIONES			
Referencia	Requerimiento	Historia de usuario	Función a desarrollar
PE-1	RNF-1	-	Configuración: Instalación y configuración de herramientas de desarrollo.
PE-2	RF-1 RF-2	5	Administración de usuarios: Registro y acceso al sistema, roles de usuario y validación de cuentas
PE-3	RF-4 RF-9	3	Insumos de prefactibilidad: Administración de tipos de área de vivienda, acabados de construcción, ambientes.
PE-4	RF-9	3	Proyectos: Gestión de proyectos asociando a cada uno una lista de estudios de prefactibilidad.
PE-5	RF-7	3, 4	Prefactibilidad: Cálculo de prefactibilidad.
PE-6	RF-5 RF-6	1, 2	Configuraciones generales: Vistas de perfil de usuario

Tabla 15. Plan de iteraciones.
Elaborado por: Diana Morocho

1.2. Tarjetas CRC

Una de las principales piezas de diseño empleada en el desarrollo del proyecto, fueron las tarjetas CRC¹⁰, que sirvieron como base para el modelado de la base de datos. Cada tarjeta CRC se convirtió en un objeto con responsabilidades en métodos públicos y sus colaboradores en llamadas a otras clases.

TARJETA CRC NRO. 1	
Clase: User	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> - Registrar datos de acceso al sistema. - Ingresar al sistema. - Crear cuenta de usuario. - Recuperar contraseña - Validar cuenta. - Gestionar clases de insumos para el cálculo de prefactibilidad (rubro, costo indirecto, tipo de área de vivienda, ambiente) - Gestionar estudios de prefactibilidad. 	<ul style="list-style-type: none"> - Model_has_permissions - Model_has_rolés - Password - Personal_Access_Token - Roles - Roles_has_permissions

Tabla 16. Tarjeta CRC – User
Elaborado por: Diana Morocho

TARJETA CRC NRO. 2	
Clase: Tipo_Area_Vivienda	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> - Gestionar los tipos de área de vivienda. - Registrar los datos de factor de circulación de paredes y factor de dirección técnica. 	<ul style="list-style-type: none"> - User - Roles_has_permissions

Tabla 17. Tarjeta CRC – Tipo de área de vivienda
Elaborado por: Diana Morocho

TARJETA CRC NRO. 3	
Clase: Ambiente	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> - Registra los datos de ambientes de construcción en el sistema. - Agrupa los ambientes de vivienda por áreas de construcción y tipo de área de vivienda. 	<ul style="list-style-type: none"> - User - Roles_has_permissions - Tipo_Area_Vivienda - Area_Construccion

Tabla 18. Tarjeta CRC – Ambiente
Elaborado por: Diana Morocho

TARJETA CRC NRO. 4	
Clase: Proyecto	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> - Gestionar la creación de proyectos en el sistema. - Agrega estudios de presupuesto y prefactibilidad. 	<ul style="list-style-type: none"> - User - Roles_has_permissions - Prefactibilidad - Presupuesto

Tabla 19. Tarjeta CRC – Proyecto
Elaborado por: Diana Morocho

TARJETA CRC NRO. 5

¹⁰ **CRC.** Colaborador – Responsabilidad – Clase. Tarjetas para identificar las clases orientadas a objetos que son relevantes en los incrementos de software.

Clase: Tipo_Acabado	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> - Gestionar los tipos de acabados de construcción. - Registrar los datos relacionados con los costos de acabados de la construcción. 	<ul style="list-style-type: none"> - User - Roles_has_permissions

Tabla 20. Tarjeta CRC – Tipo Acabado.
Elaborado por: Diana Morocho

TARJETA CRC NRO. 6	
Clase: Prefactibilidad	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> - Realizar el cálculo de estudios de prefactibilidad asociando el tipo de acabado, tipo de área y ambiente. - Determinar el subtotal de área de construcción, circulación, estructura y paredes. - Asociar los estudios de prefactibilidad a proyectos. 	<ul style="list-style-type: none"> - User - Roles_has_permissions - Item_prefactibilidad - Ambiente - Tipo_Area_Vivienda - Tipo_Acabado

Tabla 21. Tarjeta CRC – Prefactibilidad
Elaborado por: Diana Morocho

1.3. Ejecución del plan de iteraciones

ITERACIÓN PE – 1. CONFIGURACIÓN

La primera iteración, consistió en realizar la configuración de las herramientas para el fron-end y backend. A continuación, se describe de forma general la configuración de la tecnología que se utilizó en el proyecto.

BACK-END

- Ingresar a la página oficial de Docker disponible en: <https://www.docker.com/>, para descargar e instalar el gestor de contenedores sobre el que se configurará Laravel.

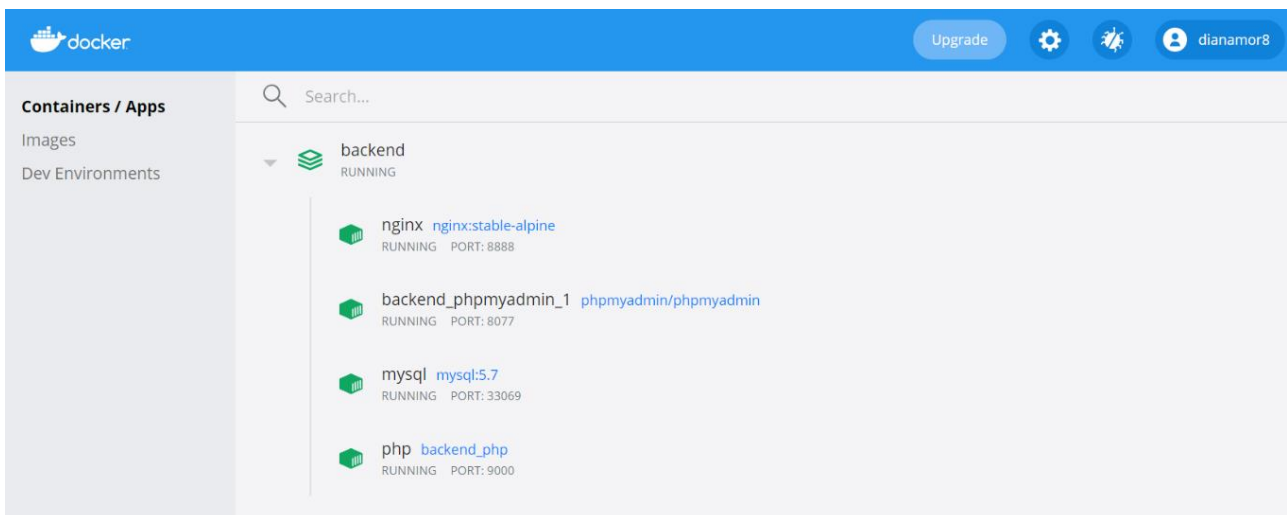


Figura 15. Instalación Docker
Elaborado por: Diana Morocho

- La instalación de dependencias de PHP se gestionará mediante Composer, para ello, se debe acceder al sitio de descargas oficial, disponible en: <https://getcomposer.org/download/>, versión: v2.0.9. Además, Previo a la instalación de Laravel, verifique la versión de Composer instalada en el equipo.

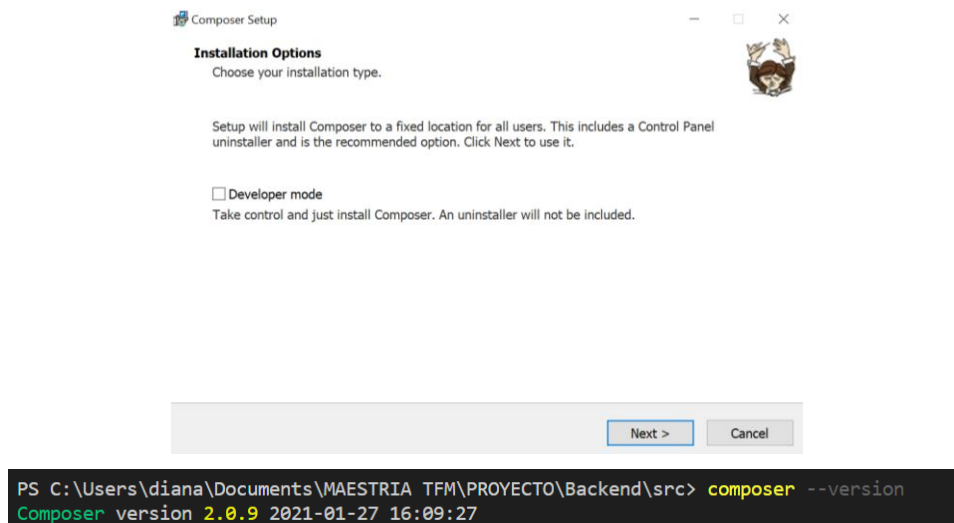


Figura 16. Instalación Composer
Elaborado por: Diana Morocho

- Con la ubicación de la consola en el directorio raíz del proyecto, se realiza la instalación de Laravel vía Composer. Aquí se crean tres directorios: mysql, nginx, src y los archivos: nginx/default.conf, src/Docker.compose.yml y Dockerfile

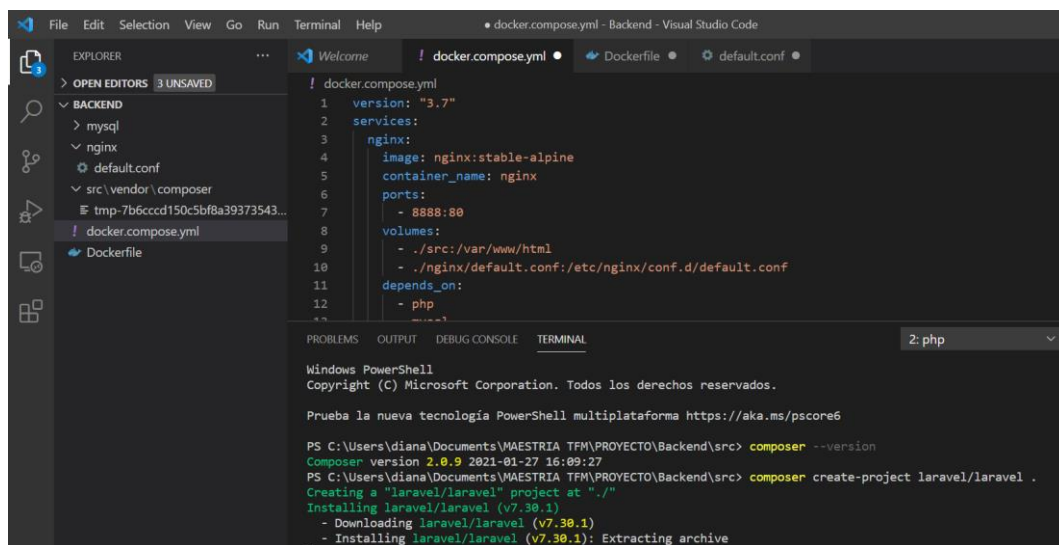


Figura 17. Creación de proyecto Laravel.
Elaborado por: Diana Morocho

- Finalmente, se inicializa el contenedor Docker con las dependencias configuradas.

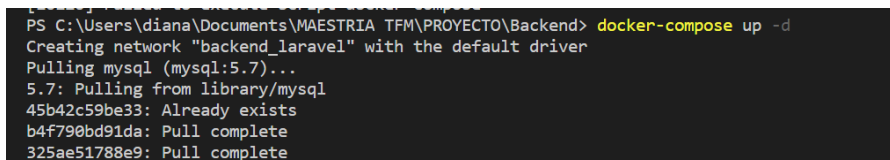
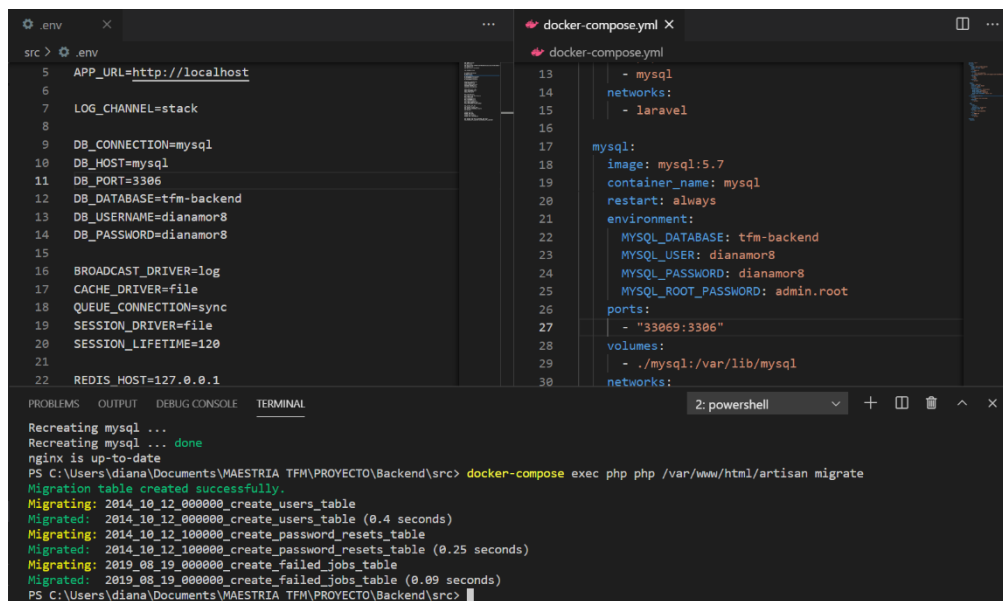


Figura 18. Inicio de servicio Docker
Elaborado por: Diana Morocho

- Para poner en marcha el servidor de Laravel, en la documentación oficial, se menciona que se debe realizar la configuración de la base de datos en el archivo .env y desplegar la primera migración de datos.



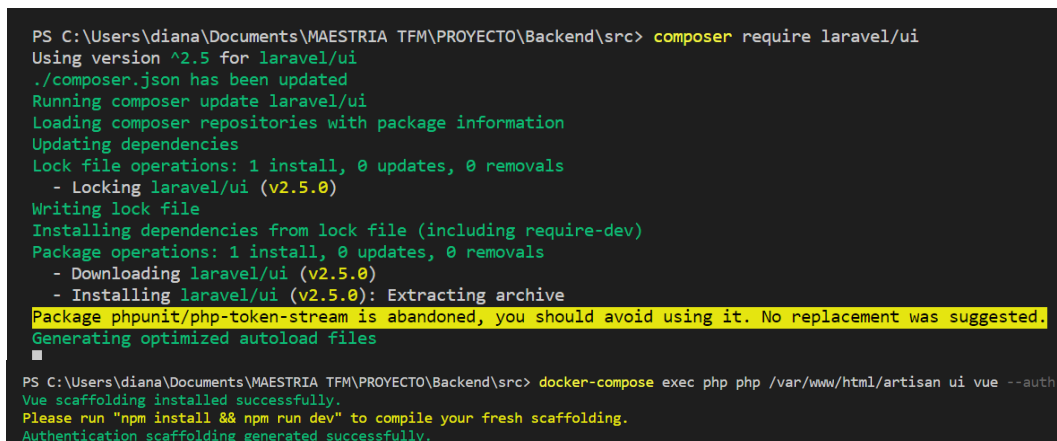
```
src > .env
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8
9 DB_CONNECTION=mysql
10 DB_HOST=mysql
11 DB_PORT=3306
12 DB_DATABASE=tfm-backend
13 DB_USERNAME=dianamor8
14 DB_PASSWORD=dianamor8
15
16 BROADCAST_DRIVER=log
17 CACHE_DRIVER=file
18 QUEUE_CONNECTION=sync
19 SESSION_DRIVER=file
20 SESSION_LIFETIME=120
21
22 REDIS_HOST=127.0.0.1

docker-compose.yml
13 - mysql
14 networks:
15 - laravel
16
17 mysql:
18 image: mysql:5.7
19 container_name: mysql
20 restart: always
21 environment:
22 MYSQL_DATABASE: tfm-backend
23 MYSQL_USER: dianamor8
24 MYSQL_PASSWORD: dianamor8
25 MYSQL_ROOT_PASSWORD: admin.root
26 ports:
27 - "33069:3306"
28 volumes:
29 - ./mysql:/var/lib/mysql
30 networks:
```

```
Recreating mysql ...
Recreating mysql ... done
nginx is up-to-date
PS C:\Users\diana\Documents\MAESTRIA TFM\PROYECTO\Backend\src> docker-compose exec php php /var/www/html/artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (0.4 seconds)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (0.25 seconds)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (0.09 seconds)
PS C:\Users\diana\Documents\MAESTRIA TFM\PROYECTO\Backend\src>
```

Figura 19. Configuración de base de datos.
Elaborado por: Diana Morocho

- El framework posee el módulo de Autenticación, para integrar el módulo Auth, se ejecuta:



```
PS C:\Users\diana\Documents\MAESTRIA TFM\PROYECTO\Backend\src> composer require laravel/ui
Using version ^2.5 for laravel/ui
./composer.json has been updated
Running composer update laravel/ui
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
- Locking laravel/ui (v2.5.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
- Downloading laravel/ui (v2.5.0)
- Installing laravel/ui (v2.5.0): Extracting archive
Package phpunit/php-token-stream is abandoned, you should avoid using it. No replacement was suggested.
Generating optimized autoload files
PS C:\Users\diana\Documents\MAESTRIA TFM\PROYECTO\Backend\src> docker-compose exec php php /var/www/html/artisan ui vue --auth
Vue scaffolding installed successfully.
Please run "npm install && npm run dev" to compile your fresh scaffolding.
Authentication scaffolding generated successfully.
```

Figura 20. Módulo de autenticación Laravel
Elaborado por: Diana Morocho

- Levantar nuevamente el contenedor Docker y acceder a la dirección <http://localhost:8888/> para verificar la instalación correcta del entorno

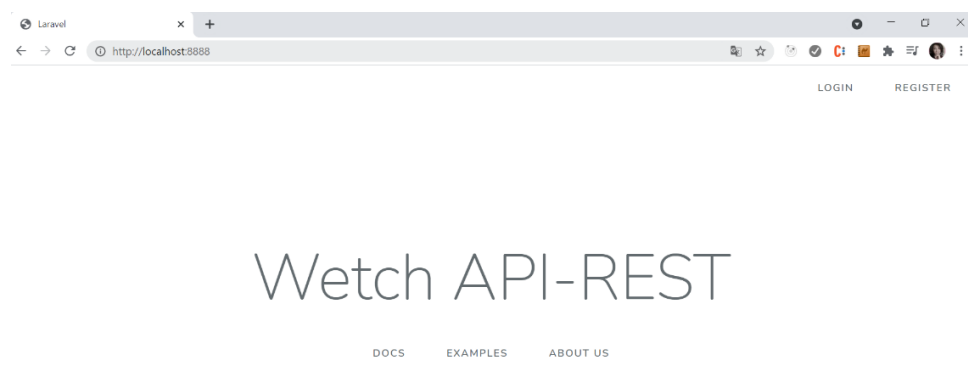


Figura 21. Ejecución de Laravel
Elaborado por: Diana Morocho

FRONT-END

- Acceder con una terminal al directorio raíz, para instalar y actualizar el gestor de paquetes npm.

```
PS C:\Users\diana\Documents\MAESTRIA TFM\PROYECTO\Front-end> Set-ExecutionPolicy Unrestricted -Scope CurrentUser -Force
PS C:\Users\diana\Documents\MAESTRIA TFM\PROYECTO\Front-end> npm install -g npm-windows-upgrade

C:\WINDOWS\system32>npm-windows-upgrade
npm-windows-upgrade v6.0.1
? Which version do you want to install? 7.7.6
Checked system for npm installation:
According to PowerShell: C:\Program Files\nodejs
According to npm: C:\Users\diana\AppData\Roaming\npm
Decided that npm is installed in C:\Program Files\nodejs
Upgrading npm... -

Upgrade finished. Your new npm version is 7.7.6. Have a nice day!
```

Figura 22. Instalación de npm
Elaborado por: Diana Morocho

- Instalar la interfaz de comandos Angular CLI.

```
PS C:\Users\diana\Documents\MAESTRIA TFM\PROYECTO\Front-end> npm install -g @angular/cli
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer supported

changed 242 packages, and audited 243 packages in 19s

22 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Figura 23. Instalación de Angular CLI
Elaborado por: Diana Morocho

- Generar un nuevo proyecto Angular.

```
PS C:\Users\diana\Documents\MAESTRIA TFM\PROYECTO\Front-end> ng new wetch-app
? Do you want to enforce stricter type checking and stricter bundle budgets in the workspace?
This setting helps improve maintainability and catch bugs ahead of time.
For more information, see https://angular.io/strict Yes
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? SCSS [ https://sass-lang.com/documentation/syntax#scss
]
CREATE wetch-app/angular.json (3729 bytes)
CREATE wetch-app/package.json (1199 bytes)
CREATE wetch-app/README.md (1017 bytes)
CREATE wetch-app/tsconfig.json (783 bytes)
CREATE wetch-app/tslint.json (3185 bytes)
CREATE wetch-app/.editorconfig (274 bytes)
CREATE wetch-app/.gitignore (631 bytes)
CREATE wetch-app/.browserslistrc (703 bytes)
CREATE wetch-app/karma.conf.js (1426 bytes)
CREATE wetch-app/tsconfig.app.json (287 bytes)
CREATE wetch-app/tsconfig.spec.json (333 bytes)
CREATE wetch-app/src/favicon.ico (948 bytes)
CREATE wetch-app/src/index.html (294 bytes)
CREATE wetch-app/src/main.ts (372 bytes)
CREATE wetch-app/src/polyfills.ts (2830 bytes)
CREATE wetch-app/src/styles.scss (80 bytes)
CREATE wetch-app/src/test.ts (753 bytes)
CREATE wetch-app/src/assets/.gitkeep (0 bytes)
CREATE wetch-app/src/environments/environment.prod.ts (51 bytes)
CREATE wetch-app/src/environments/environment.ts (662 bytes)
CREATE wetch-app/src/app/app-routing.module.ts (245 bytes)
CREATE wetch-app/src/app/app.module.ts (393 bytes)
```

Figura 24. Creación del proyecto Angular.
Elaborado por: Diana Morocho

- En la configuración de librerías para gestionar la interfaz de usuario, se incluye a Angular Material.

```
PS C:\Users\diana\Documents\MAESTRIA TFM\PROYECTO\Front-end\wetch-app> ng add @angular/material
i Using package manager: npm
✓ Found compatible package version: @angular/material@11.2.6.
✓ Package information loaded.
✓ Package successfully installed.
? Choose a prebuilt theme name, or "custom" for a custom theme: Indigo/Pink [ Preview: https://material.angular.io/th
me=indigo-pink ]
? Set up global Angular Material typography styles? Yes
? Set up browser animations for Angular Material? Yes
UPDATE package.json (1265 bytes)
✓ Packages installed successfully.
UPDATE src/app/app.module.ts (907 bytes)
UPDATE angular.json (3893 bytes)
UPDATE src/index.html (576 bytes)
UPDATE src/styles.scss (221 bytes)
```

Figura 25. Angular Material
Elaborado por: Diana Morocho

- Para el diseño responsivo, se ha incluido las propiedades de Flexbox mediante la librería de Angular flex layout.

```
PS C:\Users\diana\Documents\MAESTRIA TFM\PROYECTO\Front-end\wetch-app> npm i @angular/flex-layout @angular/cdk --save
added 1 package, and audited 1636 packages in 10s
94 packages are looking for funding
run `npm fund` for details
found 0 vulnerabilities
```

Figura 26. Angular Flex Layout
Elaborado por: Diana Morocho

- Fontawesome permite incluir un paquete de iconos, para el desarrollo del presente proyecto, se ha utilizado el paquete de iconos gratuitos.

```
PS C:\Users\diana\Documents\MAESTRIA TFM\PROYECTO\Front-end\wetch-app> ng add @fortawesome/angular-fontawesome@0.8.2
i Using package manager: npm
✓ Package information loaded.
✓ Package successfully installed.
? Choose Font Awesome icon packages you would like to use: Free Solid Icons, Free Regular Icons, Free Brands Icons
UPDATE package.json (1597 bytes)
UPDATE src/app/app.module.ts (824 bytes)
✓ Packages installed successfully.
PS C:\Users\diana\Documents\MAESTRIA TFM\PROYECTO\Front-end\wetch-app> █
```

Figura 27. Fontawesome
Elaborado por: Diana Morocho

- La aplicación del patrón redux es gestionada por la librería NgRx, que se instala con el comando:

```
PS C:\Users\diana\Documents\MAESTRIA TFM\PROYECTO\Front-end\wetch-app> npm install @ngrx/store --save
added 1 package, removed 1 package, and audited 1644 packages in 7s
94 packages are looking for funding
run `npm fund` for details
found 0 vulnerabilities
```

Figura 28. NgRx
Elaborado por: Diana Morocho

- Adicionalmente, para gestionar las múltiples acciones asíncronas se incluye el módulo effects de la librería NgRx.

```
PS C:\Users\diana\Documents\MAESTRIA TFM\PROYECTO\Front-end\wetch-app> npm install @ngrx/effects --save
added 1 package, removed 1 package, and audited 1646 packages in 11s
94 packages are looking for funding
run `npm fund` for details
found 0 vulnerabilities
```

Figura 29. Effects
Elaborado por: Diana Morocho

- Finalmente, para el almacenamiento del estado en localStorage, se agrega la librería ngx-store-localstorage.

```
PS C:\Users\diana\Documents\MAESTRIA TFM\PROYECTO\Front-end\wetch-app> npm install ngx-store-localstorage --save
added 2 packages, removed 1 package, and audited 1648 packages in 9s
94 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
```

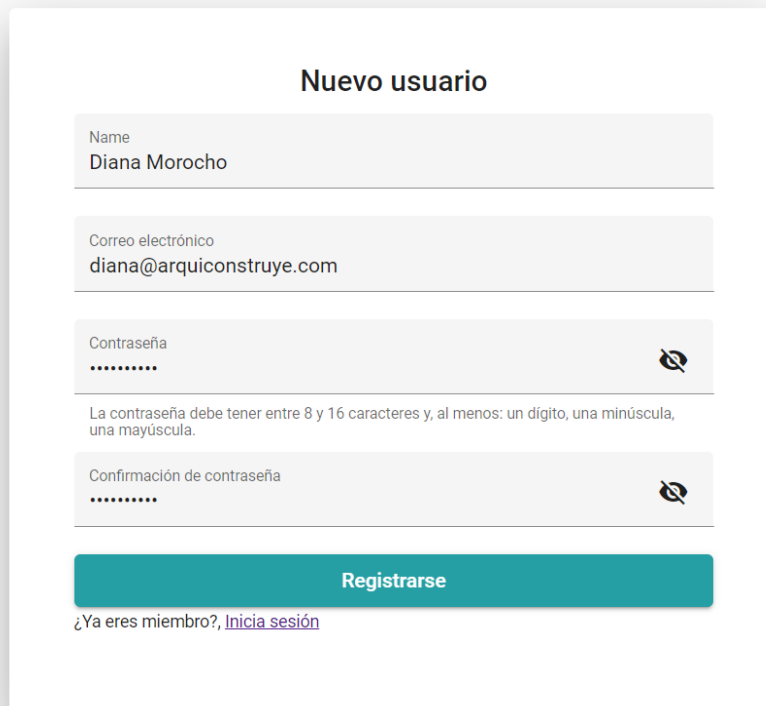
Figura 30. NgRx Store Localstorage
Elaborado por: Diana Morocho

ITERACIÓN PE – 2. ADMINISTRACIÓN DE USUARIOS.

El desarrollo de esta iteración permite restringir el acceso al sistema a usuarios autenticados y acceder a las funciones determinadas para cada rol de usuario.

En esta iteración se realizaron varias validaciones que permiten obtener las credenciales de acceso, por ejemplo: registrar correos electrónicos válidos, generar contraseñas con niveles de seguridad.

- **REGISTRO DE USUARIOS:** Se realiza el registro de nuevos usuarios ingresando un nombre, correo electrónico válido y contraseña. Aquí, al finalizar el registro, se enviará un mensaje vía correo electrónico para confirmar la cuenta.



Formulario de registro de nuevo usuario con los siguientes campos:

- Nombre:** Diana Morocho
- Correo electrónico:** diana@arquiconstruye.com
- Contraseña:** [oculto] (icono de ojo)
- Confirmación de contraseña:** [oculto] (icono de ojo)

Nota: La contraseña debe tener entre 8 y 16 caracteres y, al menos: un dígito, una minúscula, una mayúscula.

Botón: **Registrarse**

¿Ya eres miembro?, [Inicia sesión](#)

Figura 31. Formulario - Registro de nuevo usuario
Elaborado por: Diana Morocho

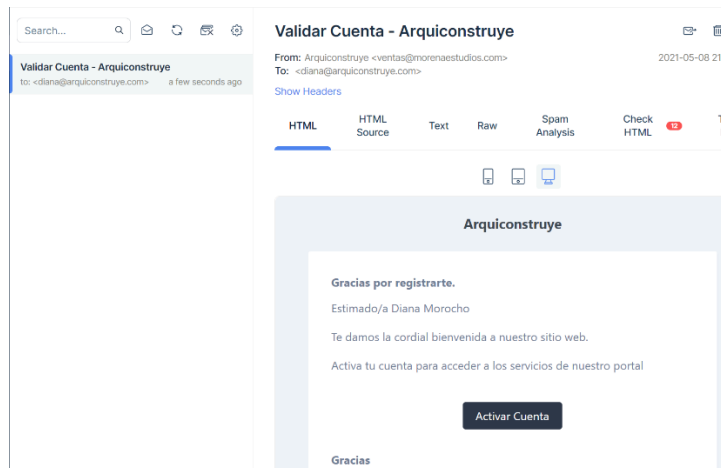


Figura 32. Correo electrónico de validación de cuenta.
Elaborado por: Diana Morocho

- **LOGIN Y LOGOUT:** Los usuarios inician y cierran sesión en el sistema.

The image shows a login form titled 'Iniciar Sesión'. It has two input fields: 'Correo electrónico' with the value 'diana@arquiconstruye.com' and 'Contraseña' with masked characters. Below the password field is a link that says '¿Olvidaste tu contraseña?'. At the bottom is a teal button labeled 'Ingresar'. Below the button is a link that says '¿No tienes credenciales de acceso? [Crear cuenta](#)'.

Figura 33. Formulario – Login
Elaborado por: Diana Morocho

- **RECUPERACIÓN DE CONTRASEÑA:** Esta función es parte del módulo de autenticación que permite recuperar los datos de acceso al sistema, se realiza mediante el envío de un correo electrónico, para reestablecer la contraseña.

The image shows a password reset form titled 'Reestablecer contraseña'. It has one input field for 'Correo electrónico' with the value 'diana@arquiconstruye.com'. Below the field is a teal button labeled 'Resetear contraseña'.

Figura 34. Formulario – Reestablecer contraseña
Elaborado por: Diana Morocho



Figura 35. Correo electrónico de reseteo de contraseña.
Elaborado por: Diana Morocho

http://localhost:4200/change-password?token=5%7C7S1MIMQ5256FJOoYm1qjYOsZWH5PldPIIgaofh&id_user=2

The image shows a screenshot of a web form titled "Cambiar Contraseña". The form has two input fields: "Nueva Contraseña" and "Confirmación de contraseña", both with masked characters (dots) and a toggle icon to show/hide the password. Below the fields, there is a teal button labeled "Cambiar contraseña" and a link labeled "Inicio". A note below the first field states: "La contraseña debe tener entre 8 y 16 caracteres y, al menos: un dígito, una minúscula, una mayúscula."

Figura 36. Formulario – Resetear Contraseña
Elaborado por: Diana Morocho

ITERACIÓN PE – 3. INSUMOS DE PREFACTIBILIDAD.

El estudio de prefactibilidad en los proyectos de construcción requiere de varios insumos para su cálculo, por ello se crean las interfaces para gestionar las entidades Área de vivienda, Acabado de Construcción, Ambiente. A continuación, se muestra las funcionalidades para realizar las acciones de creación, edición y borrado de las entidades.

The screenshot shows a mobile application interface with a teal header containing a menu icon and the text 'Wetch'. Below the header, there is a search icon and a '+' button. The main content is a table titled 'Áreas de vivienda' with the following data:

No. vivienda	Área de vivienda	% FCP ?	% FDT ?	Acciones
1	Normal	0.35	0.40	[Edit] [Delete]
2	Lujo	0.55	0.90	[Edit] [Delete]
3	Mínima	5.15	6.10	[Edit] [Delete]
4	Grande	4.16	7.40	[Edit] [Delete]

Figura 37. Lista de áreas de vivienda – Vista Mobile
 Elaborado por: Diana Morocho

The screenshot shows a desktop web application interface. The browser address bar shows 'http://localhost:4200/prefactibilidad/tipo-area-vivienda'. The page has a teal header with 'Wetch' and navigation links 'Mis Proyectos' and 'Mi perfil'. A sidebar on the left lists menu items: Prefactibilidad, Área de Vivienda, Ambientes, Acabados, Presupuesto, Rubros, and Costo Indirecto. The main content area is titled 'Áreas de vivienda' and contains a table with the same data as Figure 37, plus 'Importar' and '+ Nuevo' buttons.

No.	Área de vivienda	% FCP ?	% FDT ?	Acciones
1	Normal	0.35	0.40	[Edit] [Delete]
2	Lujo	0.55	0.90	[Edit] [Delete]
3	Mínima	5.15	6.10	[Edit] [Delete]
4	Grande	4.16	7.40	[Edit] [Delete]

Figura 38. Lista de áreas de vivienda – Vista Desktop
 Elaborado por: Diana Morocho

The screenshot shows a mobile application form titled 'Nuevo Área de Vivienda'. It includes the following fields:

- Nombre (text input)
- Factor Circulación de Paredes % x m² (text input)
- Factor de Dirección Técnica % x m² (text input)

At the bottom, there are two buttons: 'Cerrar' and 'Guardar'.

Figura 39. Formulario – Área de vivienda – Vista Mobile
 Elaborado por: Diana Morocho

The screenshot shows a desktop web application form titled 'Actualizar Área de Vivienda'. It includes the following fields:

- Nombre (text input, value: Lujo)
- Factor Circulación de Paredes (text input, value: 0.55 % x m²)
- Factor de Dirección Técnica (text input, value: 0.90 % x m²)

At the bottom, there are two buttons: 'Cerrar' and 'Guardar'.

Figura 40. Formulario – Área de vivienda – Vista Desktop
 Elaborado por: Diana Morocho

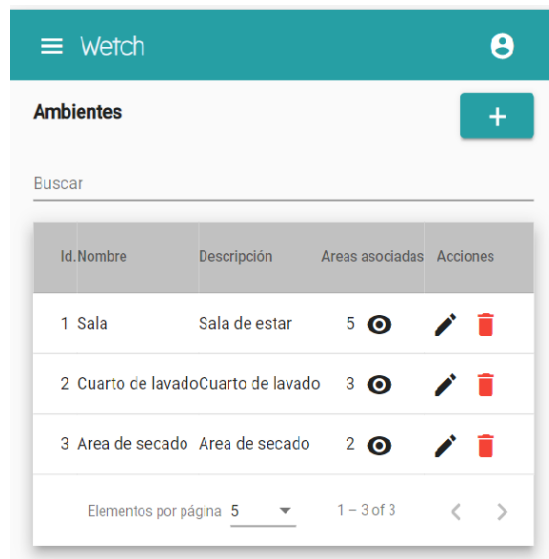


Figura 41. Lista de ambientes – Vista Mobile.
 Elaborado por: Diana Morocho

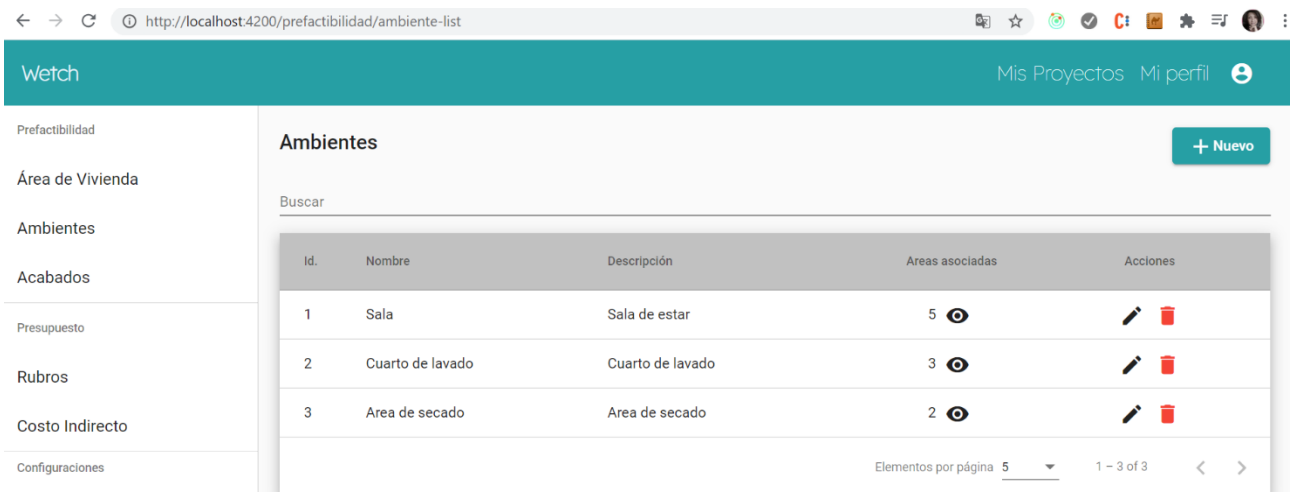


Figura 42. Lista de ambientes – Vista Desktop.
 Elaborado por: Diana Morocho

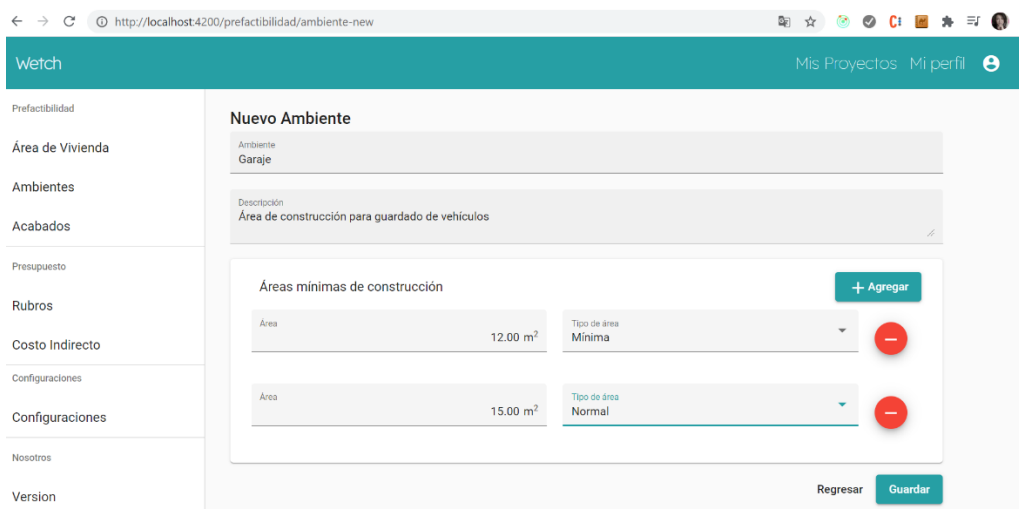


Figura 43. Formulario – Ambiente – Vista Desktop
 Elaborado por: Diana Morocho

Wetch

Nuevo Ambiente

Ambiente
Garaje

Descripción
Área de construcción para el almacenamiento de vehículos

Áreas mínimas de construcción +

Área 12.00 m²

Tipo de área Normal -

Área 14.00 m²

Tipo de área Mínima -

Regresar Guardar

© Wetch 2021. Todos los derechos reservados

Figura 44. Formulario – Ambiente – Vista Mobile
Elaborado por: Diana Morocho

ITERACIÓN PE – 4. PROYECTOS

Para mejorar la organización del espacio de trabajo, los estudios de prefactibilidad son asociados a la construcción de un proyecto, por lo tanto, se genera la interfaz para agregar proyectos.

Wetch Mis Proyectos Mi perfil

Prefactibilidad

Área de Vivienda

Ambientes

Acabados

Presupuesto

Rubros

Costo Indirecto

Configuraciones

Configuraciones

Nosotros

Versión

Mis Proyectos

+ Nuevo

Los Colmenares :
Última actualización: 5/8/21, 5:11 PM
Estudios de Prefactibilidad: 5
Cálculo de Presupuesto: 5
Agregar estudios

Proyecto Casa Fácil :
Última actualización: 5/8/21, 5:11 PM
Estudios de Prefactibilidad: 0
Cálculo de Presupuesto: 0
Agregar estudios

Asociación civil Loja :
Última actualización: 5/8/21, 5:13 PM
Estudios de Prefactibilidad: 0
Cálculo de Presupuesto: 0
Agregar estudios

ÉXITO Actualizado con éxito OK

Figura 45. Lista de proyectos – Vista Desktop.
Elaborado por: Diana Morocho

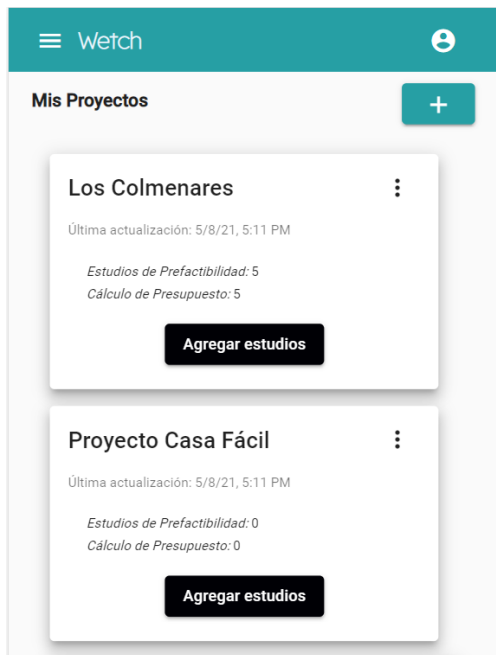


Figura 46. Lista de proyectos – Vista Mobile.
Elaborado por: Diana Morocho



Figura 47. Formulario – Proyecto – Vista Mobile
Elaborado por: Diana Morocho

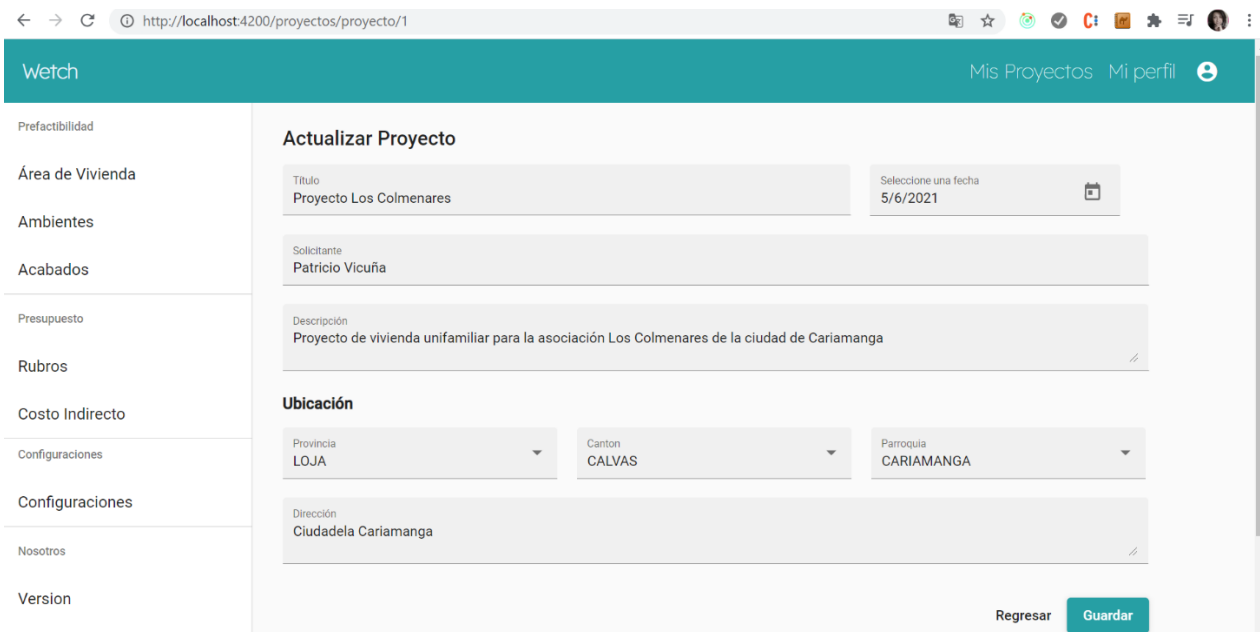


Figura 48. Formulario – Proyecto – Vista Desktop
Elaborado por: Diana Morocho

ITERACIÓN PE – 5. PREFACTIBILIDAD

Los estudios de prefactibilidad requieren incluir los insumos generados en la iteración 3. Aquí se incluye las áreas de vivienda, ambientes, tipos de acabado y ubicación.

The screenshot shows the 'Estudios de Prefactibilidad' (Feasibility Studies) section of the Wetch web application. The interface includes a sidebar with navigation options: Proyectos, Mis Proyectos, Prefactibilidad, Área de Vivienda, Ambientes, Acabado de construcción, and Atención a clientes. The main content area features a search bar and a table with the following data:

Id.	Fecha	Área de Construcción	Costo de Construcción	Inversión Total	Acciones
1	1/06/2021	78.72m ²	\$2,391.27	\$47,701.60	[Iconos de acciones]

Below the table, there is a pagination control showing 'Elementos por página 5' and '1 - 1 of 1'.

Figura 49. Lista de estudios de prefactibilidad – Vista Desktop.
 Elaborado por: Diana Morocho

The screenshot shows the 'Nuevo Estudio de Prefactibilidad' (New Feasibility Study) form in the Wetch web application. The form is titled 'Nuevo Estudio de Prefactibilidad' and includes a sidebar with the same navigation options as Figure 49. The main content area displays the following information:

Área registrada: 2.160,00 m² Área de retiros: 120,00 m² Área COS: 2.040,00 m² % COS: 94,44 %

Área de terreno: Regular Irregular

Frente	40.00 m	Fondo	54.00 m
Retiro frontal	2.00 m	Retiro posterior	1.00 m
Retiro Lateral Izquierdo	0.00 m	Retiro Lateral Derecho	0.00 m

At the bottom right, there are 'Regresar' and 'Continuar' buttons.

Figura 50. Estudio de prefactibilidad, área de construcción - Vista Desktop
 Elaborado por: Diana Morocho

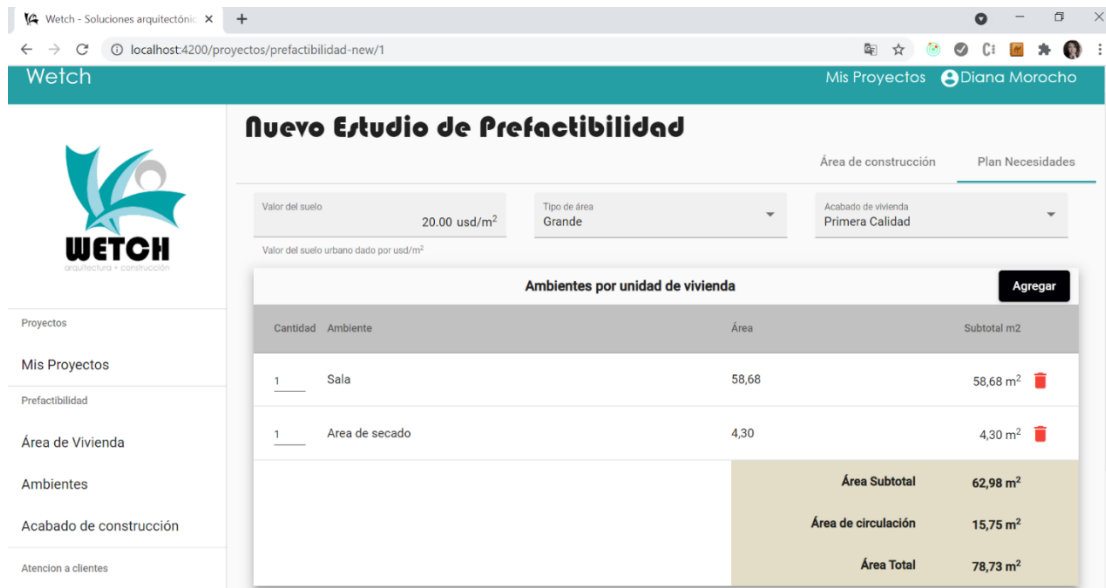


Figura 51. Estudio de prefactibilidad, plan de necesidades – Vista Desktop
 Elaborado por: Diana Morocho

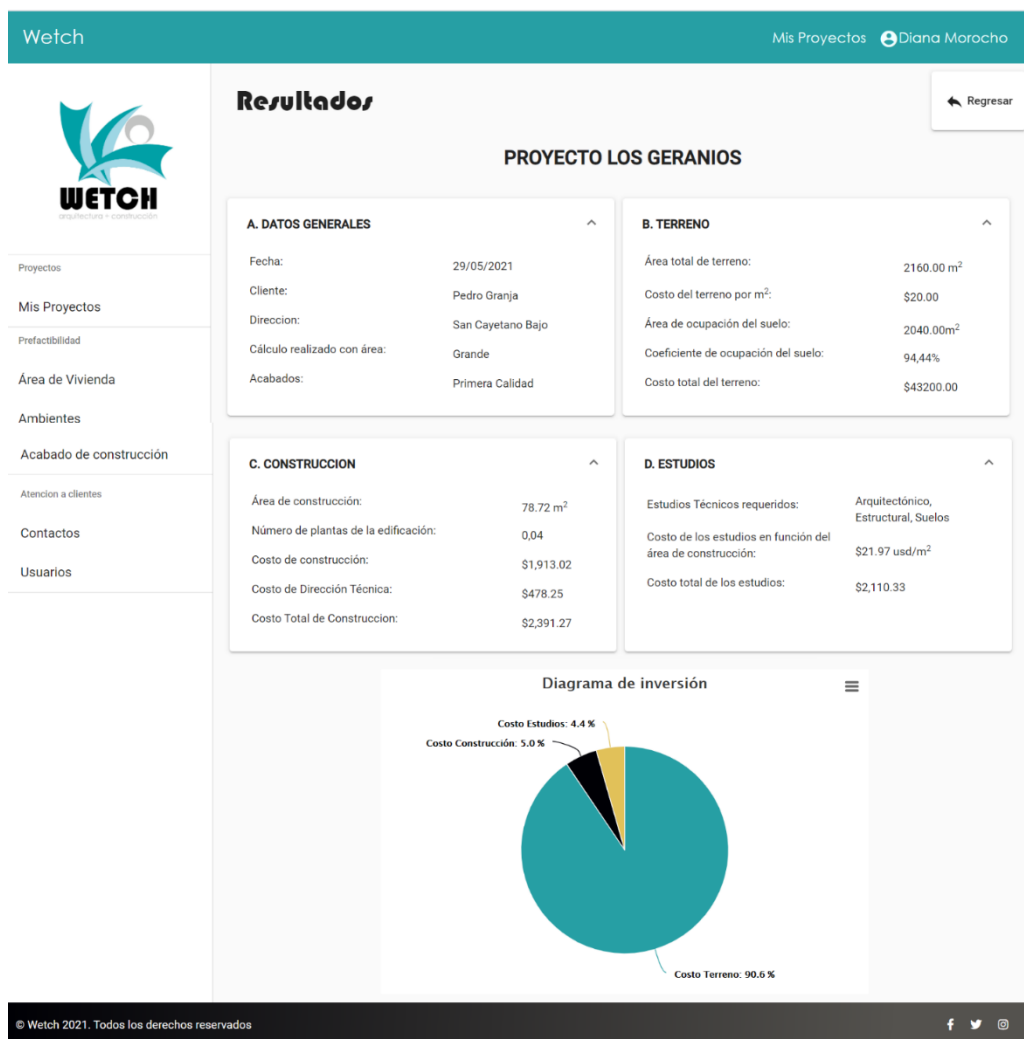


Figura 52. Resultados del estudio de prefactibilidad – Vista Desktop
 Elaborado por: Diana Morocho

ITERACIÓN PE – 6. SETTINGS

En esta iteración, se realizaron varias configuraciones del sitio, como las vistas por tipos de usuario y sus respectivos perfiles. Adicionalmente, se cuenta con un formulario de contactos en el exterior de la página para recibir las comunicaciones de los usuarios que deseen dejar sus comentarios o realizar solicitudes particulares.

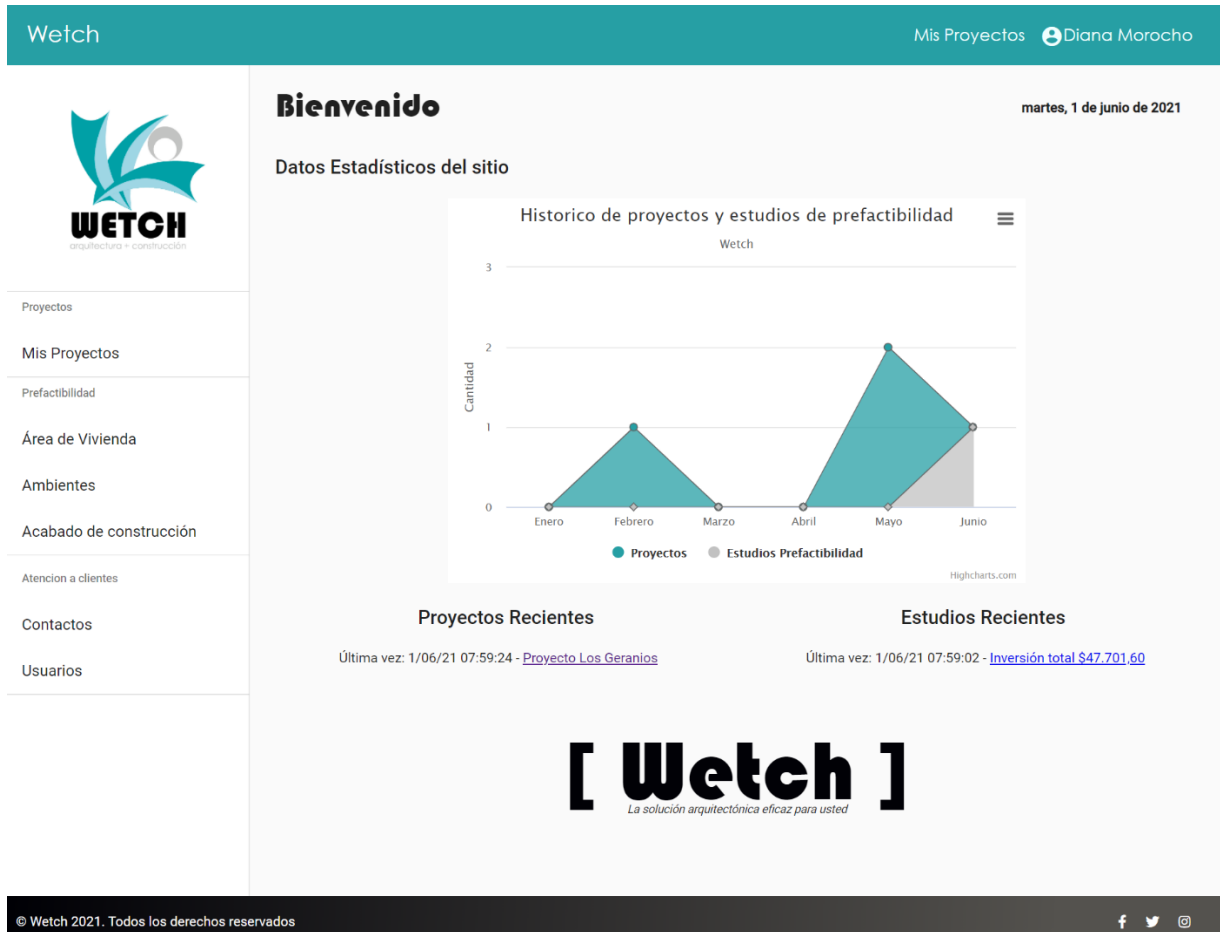


Figura 53. Vista dashboard del usuario Administrador – Vista Desktop
Elaborado por: Diana Morocho



Figura 54. Vista dashboard del usuario Planificador – Vista Desktop
Elaborado por: Diana Morocho

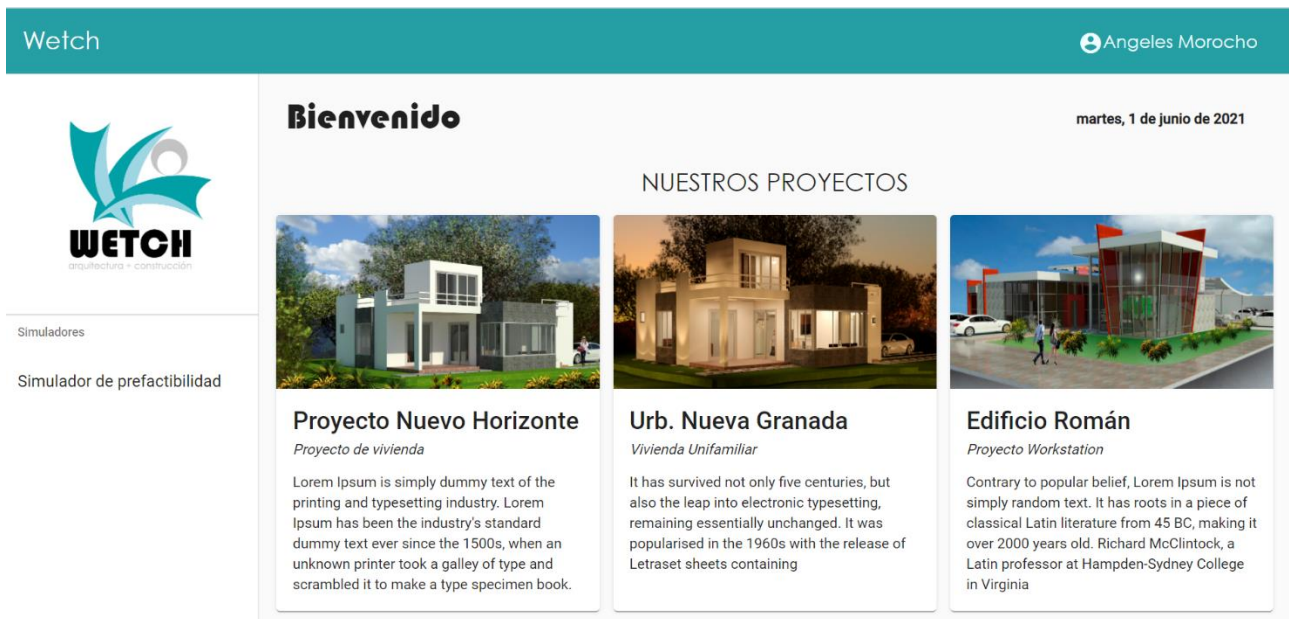


Figura 55. Vista dashboard del usuario cliente – Vista Desktop
 Elaborado por: Diana Morocho

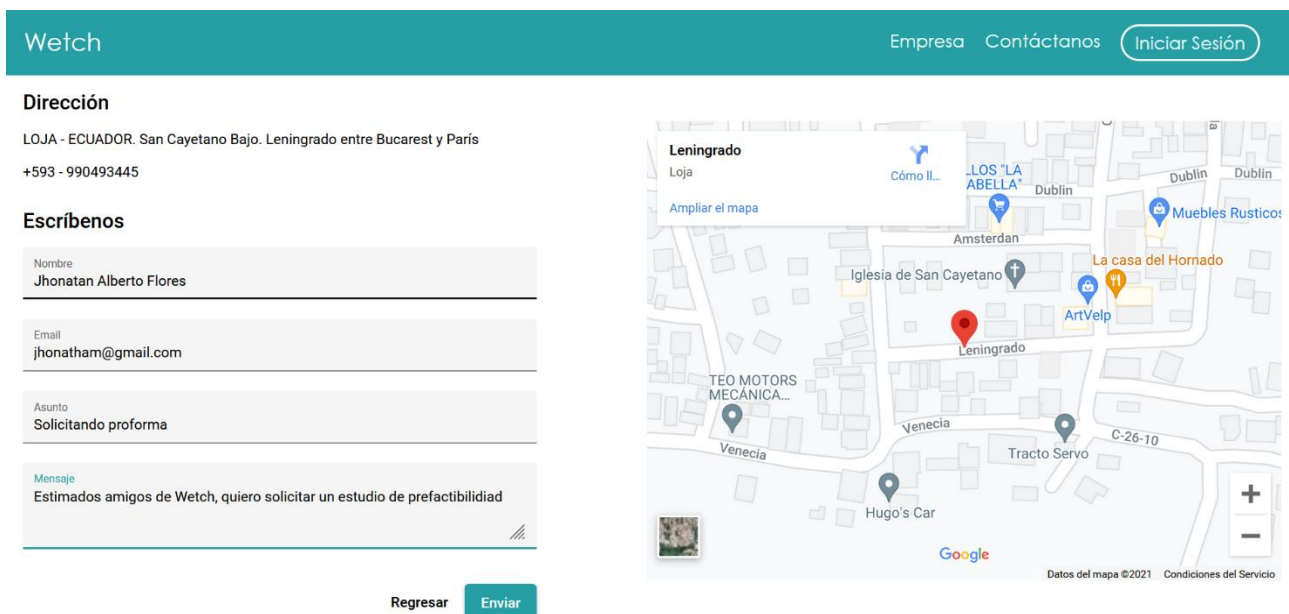


Figura 56. Formulario de contacto para clientes – Vista Desktop
 Elaborado por: Diana Morocho

Wetch Mis Proyectos Diana Morocho

Comunicaciones recibidas

Detalle del mensaje

Nombre: Jhonatan Alberto Flores Email: jhonatham@gmail.com

Asunto: Solicitando proforma

Mensaje: Estimados amigos de Wetch, quiero solicitar un estudio de prefactibilidad

Mensajes recibidos

No.	Fecha	Correo electrónico	Asunto	Acciones
1	1/06/2021 08:30:29	jhonatham@gmail.com	Solicitando proforma	

Figura 57. Formulario de comunicaciones recibidas - Vista Desktop
Elaborado por: Diana Morocho

Wetch Mis Proyectos Diana Morocho

Usuario

Usuario: ANgel Songor Email: dianamoroch@calasanzloja.edu.ec

Administrador Planificador Cliente

Actualizar Usuario

Lista de usuarios

Buscar

Id.	Nombre	Email	Administrador	Planificador	Cliente	Activo	Acciones
2	Wilson Tapia	wilson@wetch.com	No	Si	No	Si	
3	Angeles Morocho	angeles@wetch.com	No	No	Si	Si	
4	Alexandra	alexandra@wetch.com	No	No	Si	Si	
5	ANgel Songor	dianamoroch@calasanzloja.edu.ec	No	No	Si	No	
6	dasadasd	dianamor8@gmail.com	No	No	Si	Si	

Items per page: 5 0 of 0

© Wetch 2021. Todos los derechos reservados

Figura 58. Administración de usuarios – Vista Desktop
Elaborado por: Diana Morocho

CAPÍTULO V. VALIDACIÓN

1. Tests

De acuerdo al contexto y naturaleza de la aplicación Wetch, considero importante aplicar el test de funcionalidad a modo de pruebas unitarias y prueba de rendimiento ejecutadas por el programador. Además, con la finalidad de mejorar la experiencia de usuario se realizó una validación de aceptación ejecutado por el cliente.

1.1. Pruebas unitarias

Este tipo de pruebas, permite determinar que el software cumple con las funcionalidades para las que se ha requerido, de manera prioritaria, se seleccionó a los Servicios de autenticación para realizar las pruebas. En la siguiente imagen, visualizará dos pruebas unitarias de tipo test-double implementadas en el desarrollo, una para realizar el login en el back-end y otra para verificar la existencia de un usuario.

```
TS auth.service.specs M • TS auth.service.ts M
src > app > auth > services > TS auth.service.specs > describe('AuthService') callback
1 import { HttpXsrfTokenExtractor } from '@angular/common/http';
2 import { Router } from '@angular/router';
3 import { of } from 'rxjs';
4 import { MessageService } from 'src/app/utilitarios/messages/services/message.service';
5 import { AuthService } from './auth.service';
6
7 describe('AuthService', () => {
8   let httpClientSpy: { get: jasmine.Spy, post: jasmine.Spy, delete: jasmine.Spy, put: jasmine.Spy };
9   let authService: AuthService;
10  let messageService: MessageService = new MessageService(null);
11  let router: Router;
12  let tokenService: HttpXsrfTokenExtractor;
13
14  beforeEach(() => {
15    httpClientSpy = jasmine.createSpyObj('HttpClient', ['get', 'post', 'delete', 'put']);
16    authService = new AuthService(httpClientSpy as any, messageService, router, tokenService);
17  });
18
19  it('[AUTH SERVICE] - LOGIN TEST', () => {
20    const expectedUser = {
21      access_token: "12|1crXQa1hRftoDJBp0QhsEYU2uUcMHca3WgMfHqtW",
22      token_type: "Bearer",
23      user: {
24        id: 1,
25        name: "Diana Morocho",
26        email: "mor8diana@wetch.com",
27        email_verified_at: "2021-05-29T10:41:29.000000Z",
28        profile: {
29          id: 1,
30          nombres: "Diana Morocho",
31          apellidos: null,
32          telefono: null,
33          direccion: null
34        },
35        permissions: [],
36        roles: [...
177 ]
178 }
179 }
180 httpClientSpy.post.and.returnValue(of(expectedUser));
181 authService.login('mor8diana@wetch.com','Mdiana1105').subscribe(
182   (user) => expect(user).toEqual(expectedUser, 'expected user')
183 );
184 expect(httpClientSpy.post.calls.count()).toBe(1, 'one call');
185 });
```

Figura 59. Prueba unitaria – Login test
Elaborado por: Diana Morocho

```
187 it('[AUTH SERVICE] - VERIFY USER', () => {
188   const expectedUser: any= {
189     user: {
190       id: 1,
191       name: "Diana Morocho",
192       email: "mor8diana@wetch.com",
193       email_verified_at: "2021-05-29T10:41:29.000000Z",
194       created_at: "2021-05-29T15:41:13.000000Z",
195       updated_at: "2021-05-29T15:41:13.000000Z",
196       empresa_id: 1,
197       active: 1
198     }
199   };
200
201   httpClientSpy.get.and.returnValue(of(expectedUser));
202   authService.userExist('mor8diana@gmail.com')
203     .subscribe(
204     (user) => expect(user).toEqual(expectedUser, 'expected user'),
205     );
206   expect(httpClientSpy.get.calls.count()).toBe(1, 'one call');
207 });
208 });
```

Figura 60. Prueba unitaria – Verify User
Elaborado por: Diana Morocho

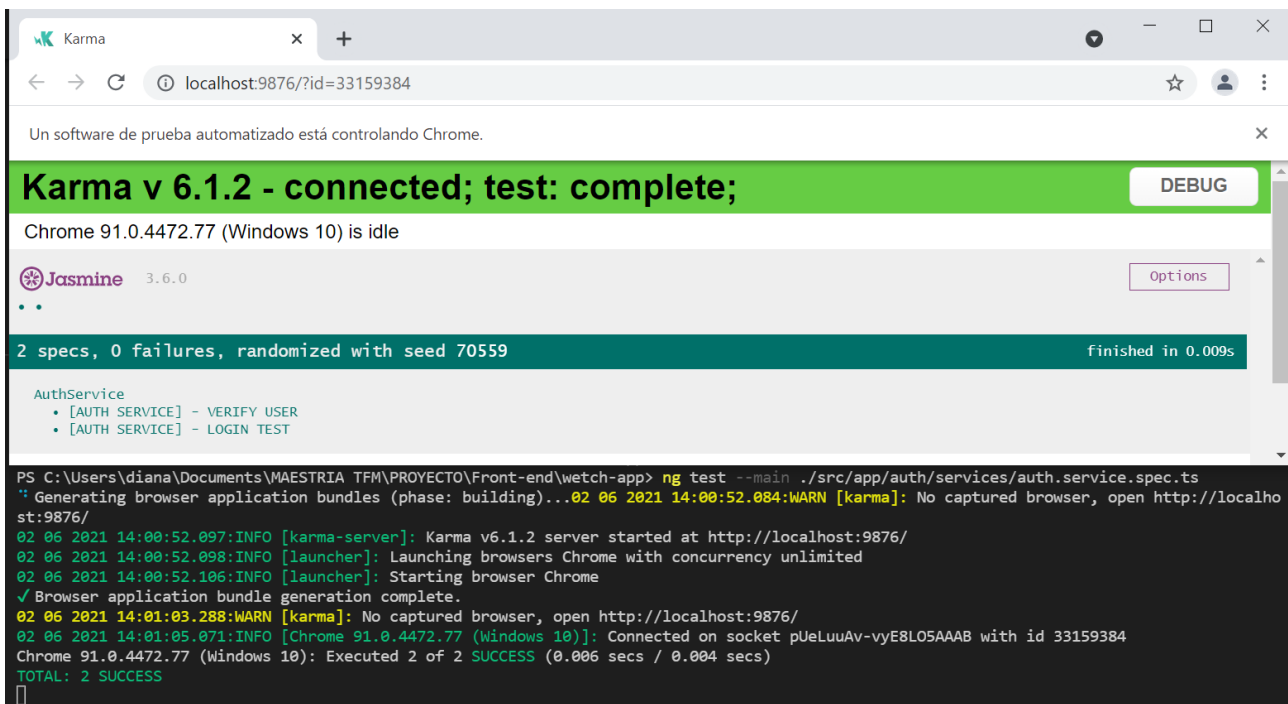


Figura 61. Ejecución exitosa de pruebas unitarias
Elaborado por: Diana Morocho

1.2. Prueba de rendimiento

Para realizar la toma de medidas del rendimiento, se procede a ejecutar siete test en la plataforma PageSpeed Insights, los que permitirán definir el tiempo de rendimiento con mayor aproximación, así como de proveer las sugerencias necesarias para su optimización. En la siguiente tabla se resumen los datos obtenidos en ordenador.

*Los registros 1 y 7 se han descartado para realizar la métrica óptima entre las 5 pruebas.

Nro	Fecha	Scoring	Primer renderizado con contenido	Índice de velocidad	Largest Contentfull Paint	Tiempo hasta que está interactiva	Tiempo total de bloqueo	Cambios de diseño acumulados
1	02/06/2021 14:55	45	3.1s	3.9s	16.4s	8.4s	680ms	0.001
2	02/06/2021 14:57	48	2.6s	3.1s	13.2s	10.9s	600ms	0.001
3	02/06/2021 15:00	52	2.2s	2.9s	16.1s	8.3s	600ms	0.001
4	02/06/2021 15:03	51	2.2s	5.4s	12.2s	8.0s	500ms	0.002
5	02/06/2021 15:07	50	2.5s	2.9s	13.2s	10.8s	570ms	0.001
6	02/06/2021 15:11	55	2.2s	3.3s	15.9s	8.0s	470ms	0.001
7	02/06/2021 15:13	56	2.2s	3.7s	16.4s	6.7s	610ms	0.001
Media		51.2	2.34s	3.52s	14.12s	9.2s	548ms	0,001

Tabla 22. Métricas obtenidas de la plataforma PageSpeeds Insights
Elaborado por: Diana Morocho

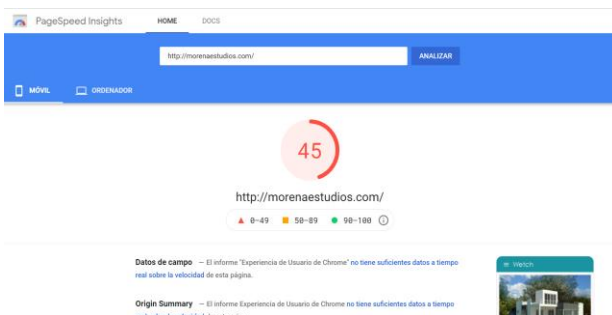


Figura 62. Test de rendimiento 1 de 7.

Fuente: <https://developers.google.com/speed/pagespeed/insights/>
Elaborado por: Diana Morocho

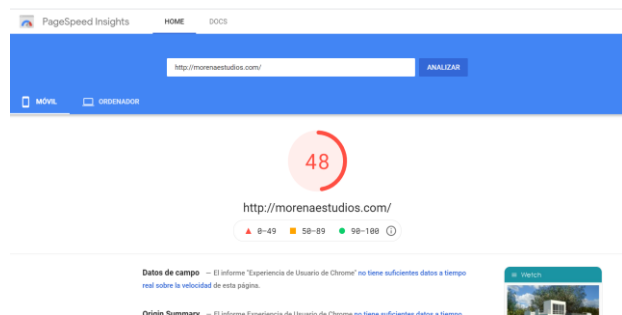


Figura 63. Test de rendimiento 2 de 7.

Fuente: <https://developers.google.com/speed/pagespeed/insights/>
Elaborado por: Diana Morocho

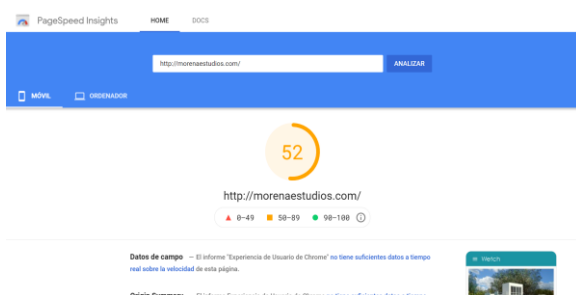


Figura 64. Test de rendimiento 3 de 7.

Fuente: <https://developers.google.com/speed/pagespeed/insights/>
Elaborado por: Diana Morocho

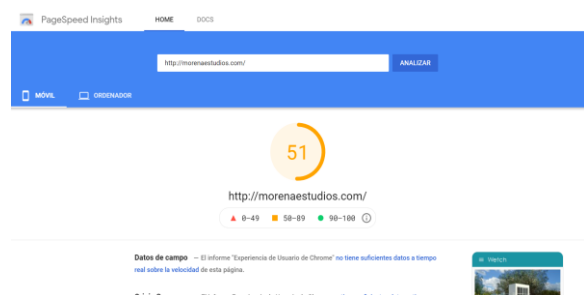


Figura 65. Test de rendimiento 4 de 7.

Fuente: <https://developers.google.com/speed/pagespeed/insights/>
Elaborado por: Diana Morocho

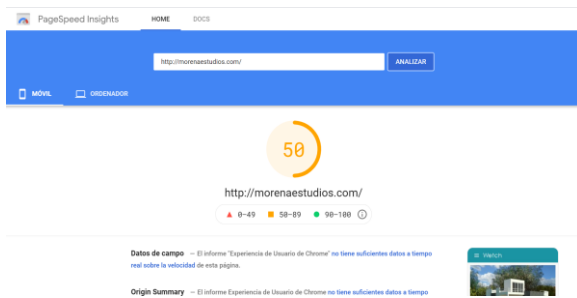


Figura 66. Test de rendimiento 3 de 7.

Fuente: <https://developers.google.com/speed/pagespeed/insights/>
Elaborado por: Diana Morocho

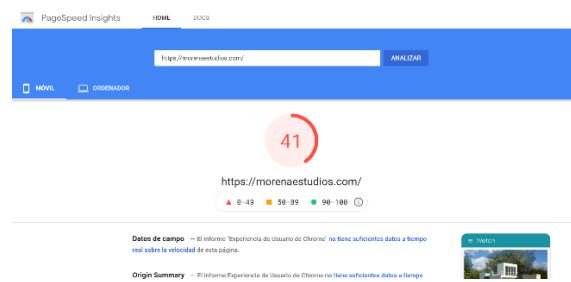


Figura 67. Test de rendimiento 4 de 7.

Fuente: <https://developers.google.com/speed/pagespeed/insights/>
Elaborado por: Diana Morocho

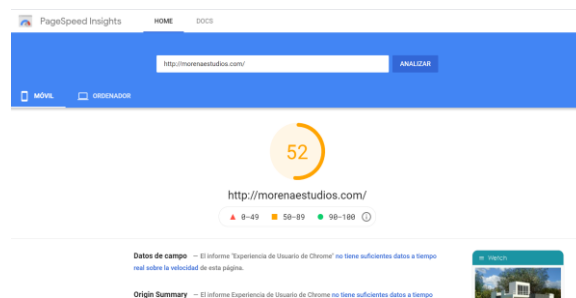


Figura 68. Test de rendimiento 7 de 7.

Fuente: <https://developers.google.com/speed/pagespeed/insights/>
Elaborado por: Diana Morocho

Análisis de la media de las métricas de rendimiento en los resultados obtenidos

- Scoring – 51,2 Para obtener la puntuación de velocidad, se compara la mediana de cada una de estas métricas (FCP - First Contentful Paint o DCL - DOMContentLoaded) con todas las páginas incluidas en el informe CrUX (Experiencia de Usuario de Chrome). Una vez hecho esto, en función de la posición de cada métrica en el total de páginas incluidas, se asignan a una de estas categorías: Rápida, Lenta y Normal. El puntaje obtenido, corresponde a la categoría Normal, lo que implica que la mediana de la métrica se encuentra en el tercio intermedio de todas las cargas de páginas.
- Primer renderizado con contenido – 2,34s: First Contentful Paint (FCP), es una de las seis métricas rastreadas en la sección Rendimiento del informe de Lighthouse. Determina el primer renderizado con contenido e indica el momento en el que se renderiza el primer texto o la primera imagen. Para calcular la métrica, los sitios que se desempeñan en el percentil noventa y nueve procesan FCP en aproximadamente 1,5 segundos, por lo tanto, se asignan tres categorías Rápido (0 a 2 segundos), Moderada (2 a 4 segundos), Lento (Más de 4 segundos) y para éste resultado, el primer renderizado se realiza en modo Moderada.
- Índice de velocidad – 3,52s: En inglés, Speed Index, el índice de velocidad indica la rapidez con la que se puede ver el contenido de una página. El puntaje de índice de velocidad es una comparación del índice de velocidad de su página y los índices de velocidad de sitios web reales, basados en datos del Archivo HTTP.

Para categorizar sus resultados usa Verde (0-4,3 segundos), Naranja (4,4-5,8 segundos), Rojo (Más de 5,8). Por lo tanto, en la ejecución del test, el índice de la velocidad es verde.

- Largest Contentfull Paint - 14,12s: La métrica Largest Contentful Paint indica el tiempo que se tarda en dibujar el texto o la imagen de mayor tamaño. LCP mide cuándo el elemento de contenido más grande en la ventana gráfica se representa en la pantalla. Esto se aproxima cuando el contenido principal de la página es visible para los usuarios. La clasificación en el código de colores es: Verde (0-2 segundos, rápido), Naranja (2-4 segundos, moderado), Rojo (Más de 4 segundos, lento). En la media del test, se obtiene un puntaje al margen de rojo.
- Tiempo hasta que está interactiva - 9,2s: Time to Interactive (TTI), Mide cuánto tiempo tarda una página en ser completamente interactiva. Una página se considera completamente interactiva cuando:
 - La página muestra contenido útil, que se mide con la primera pintura satisfactoria,
 - Los controladores de eventos están registrados para la mayoría de los elementos de página visibles, y
 - La página responde a las interacciones del usuario en 50 milisegundos.

Para determinar el puntaje, se categoriza con código de colores, Verde (0-5,2 segundos), Naranja (5,3-7,3 segundos) y Rojo (Más de 7,3) para determinar a la velocidad entre rápido, moderado y lento. La media del TTI en el sitio analizado es lento. Sin embargo, se puede optimizar considerando que el tiempo de carga es sobre el total de la SPA.

- Tiempo total de bloqueo - 4ms: Total Blocking Time. Suma de los periodos, en milisegundos, entre FCP y Time to Interactive cuando la duración de la tarea excede los 50 ms. TBT mide la cantidad total de tiempo que una página está bloqueada para que no responda a la entrada del usuario, como clics del mouse, toques de pantalla o pulsaciones del teclado. La suma se calcula agregando la parte de bloqueo de todas las tareas largas entre First Contentful Paint y Time to Interactive. Cualquier tarea que se ejecute durante más de 50 ms es una tarea larga. Para éste caso, contar con 4ms es una métrica aceptable.
- Cambios de diseño acumulados - 0,001: Cumulative Layout Shift (CLS) Los cambios de diseño acumulados miden el movimiento de los elementos visibles dentro de la ventana gráfica. CLS mide la suma total de todas las puntuaciones de los cambios de diseño individuales para cada cambio de diseño inesperado que ocurre durante toda la vida útil de la página. Se produce un cambio de diseño cada vez que un elemento visible cambia su posición de un cuadro a otro. En el caso de la media de los test, para proporcionar una buena experiencia de usuario, los sitios deben esforzarse por tener un puntaje CLS de menos de 0.1, por lo tanto, ésta métrica es óptima.

Luego de analizar cada una de las métricas, se decidió optimizar la mayoría de los parámetros, de manera prioritaria, disminuir la carga de imágenes utilizadas en el sitio web. Para ello se utilizó conversor de imágenes a formato webp. Además, se limpió el Sass no utilizado, obteniendo el siguiente puntaje.

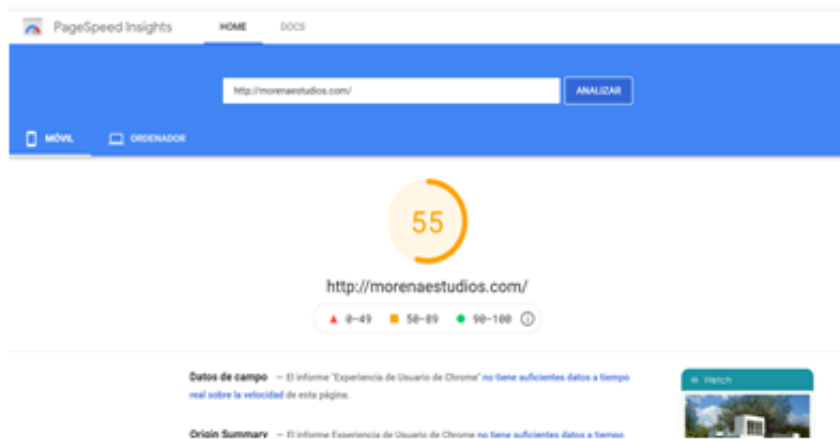


Figura 69. Puntaje obtenido luego de la optimización
Elaborado por: Diana Morocho

1.3. Prueba de validación de usuario

Adicionalmente, XP al ser una metodología de desarrollo iterativo e incremental, progresivamente se presentaron los avances de cada iteración, con la finalidad de que el usuario valide cada requerimiento incluido en el software.

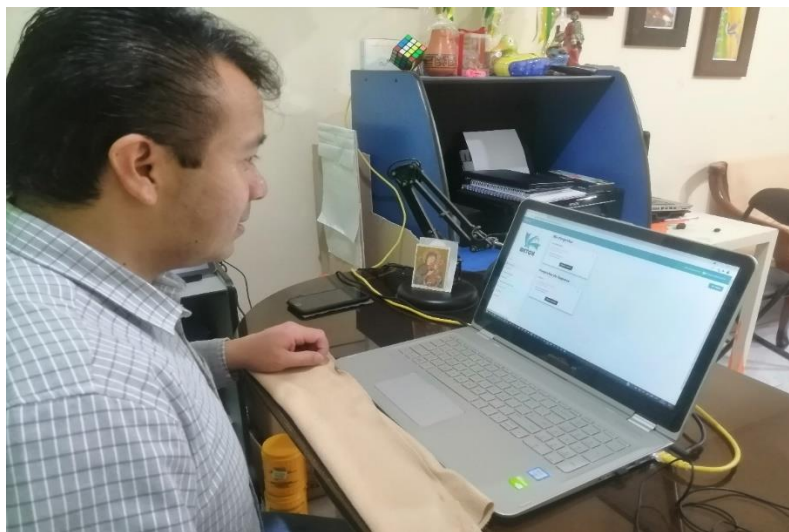


Figura 70. Validación de usuario. Gerente Wetch.
Elaborado por: Diana Morocho

2. Requisitos de implantación

2.1. Requisitos de Back-end

SOFTWARE

Los requerimientos mínimos del servidor para ejecutar Laravel son:

- PHP \geq 7.2.5
- BCMath PHP Extension
- Ctype PHP Extension
- Fileinfo PHP extension
- JSON PHP Extension
- Mbstring PHP Extension
- OpenSSL PHP Extension
- PDO PHP Extension
- Tokenizer PHP Extension
- XML PHP Extension

Para el servidor de base de datos, de preferencia se requiere de un equipo de las siguientes características:

- Ubuntu 21.04 – Arquitectura x86_64
- Ubuntu 20.04 LTS - Arquitectura x86_64
- Ubuntu 18.04 LTS - Arquitectura x86_32, x86_64
- Ubuntu 16.04 LTS - Arquitectura x86_32, x86_64
- Microsoft Windows 2019 Server - Arquitectura x86_64

HARDWARE

El servidor de base de datos MySQL, requiere como mínimo un equipo con:

- 4 CPU Cores o más
- 8 GB RAM o más
- RAID10 o RAID 0+1 disk setup
- Linux x86 32-bit – 1.3 GB – Almacenamiento mínimo en disco
- Mac OS X – 1.2 GB – Almacenamiento mínimo en disco
- Windows x86 64-bit – 800 GB – Almacenamiento mínimo en disco

2.2. Requisitos de Front-end

SOFTWARE

Los requerimientos de servidor para ejecutar Angular son:

- Node v15.6.0
- Npm 7.7.6
- Apache
- SSL

Para los requerimientos de cliente, los navegadores: Google Chrome, Mozilla Firefox, Safari, Edge.

HARDWARE

Node, tiene como requerimientos de hardware lo siguientes:

- CPU(s): x86-64 (Intel preferible); 4 cores mínimo (16 o más recomendado).
- RAM: 8GB mínimo, 16GB o más recomendado.
- Disk: 300 GB o más de almacenamiento usable para la instalación en un servidor, que incluye:
 - 30 GB mínimo para sistema operativo
 - 35 GB para estructura
 - 700 a 1700 MB de almacenamiento para usuarios finales.

3. Instrucciones de implantación

3.1. Implantación del Back-end

El código fuente del Back-end se encuentra disponible en el siguiente repositorio público de Github: <https://github.com/dianamor8/wetch-backendv2.git>, del cual se puede realizar la clonación del código.

El proceso de implantación del back-end, inicia con la puesta en marcha de la base de datos, para ello, se puede hacer uso de las herramientas que proporciona su proveedor de hosting.

- Crear una base de datos en Mysql

Bases de datos actuales

Buscar	Ir		
Base de datos	Tamaño	Usuarios con privilegio	Acciones
morendio_db	12.35 MB	morendio_db	Cambiar el nombre Borrar
morendio_tfm-backend	2.53 MB	morendio_wetch	Cambiar el nombre Borrar

Figura 71. Creación de base de datos.
Elaborado por: Diana Morocho

- Para facilitar el modelado de la estructura de la aplicación, se puede tomar un backup de la aplicación en desarrollo e importarlo mediante la herramienta phpMyAdmin.

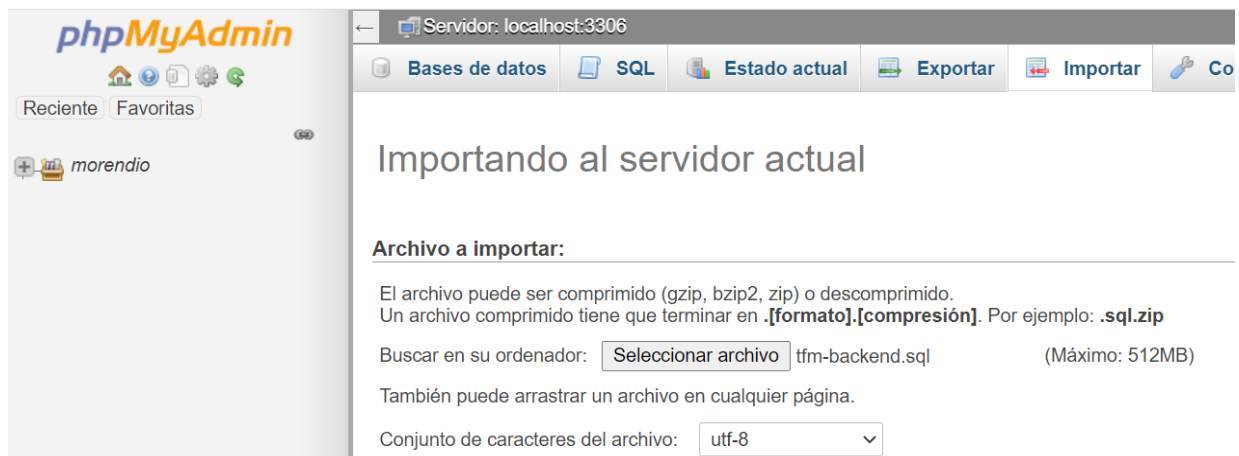


Figura 72. Importación de la estructura de la db
Elaborado por: Diana Morocho

- Dentro del espacio de almacenamiento en el servidor, es importante crear un subdominio para aislar la aplicación de otras disponibles.

Subdominios.Dominio de raíz	Directorio raíz	redirección	Acciones
wetch-api.morenaestudios.com	/wetch-api.morenaestudios.com	not redirected	Borrar Administrar redireccionamiento

Figura 73. Creación de subdominio
Elaborado por: Diana Morocho

- Para acceder de forma remota y subir los archivos al servidor, se debe crear un cliente FTP

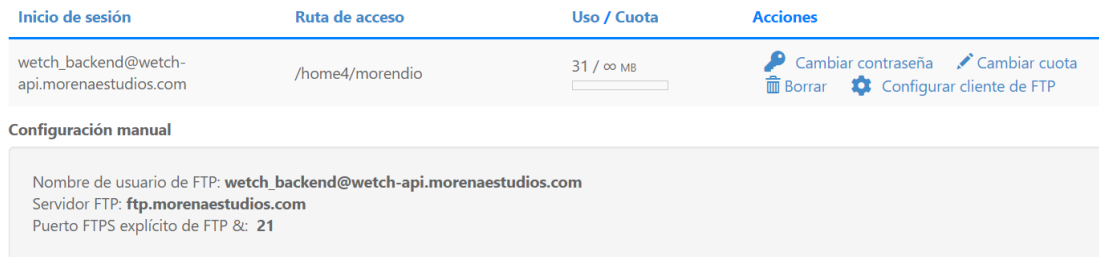


Figura 74. Cliente FTP
 Elaborado por: Diana Morocho

- Mediante el cliente FTP Filezilla se realiza la conexión con el servidor y se transfieren al subdominio todos los archivos de Laravel.

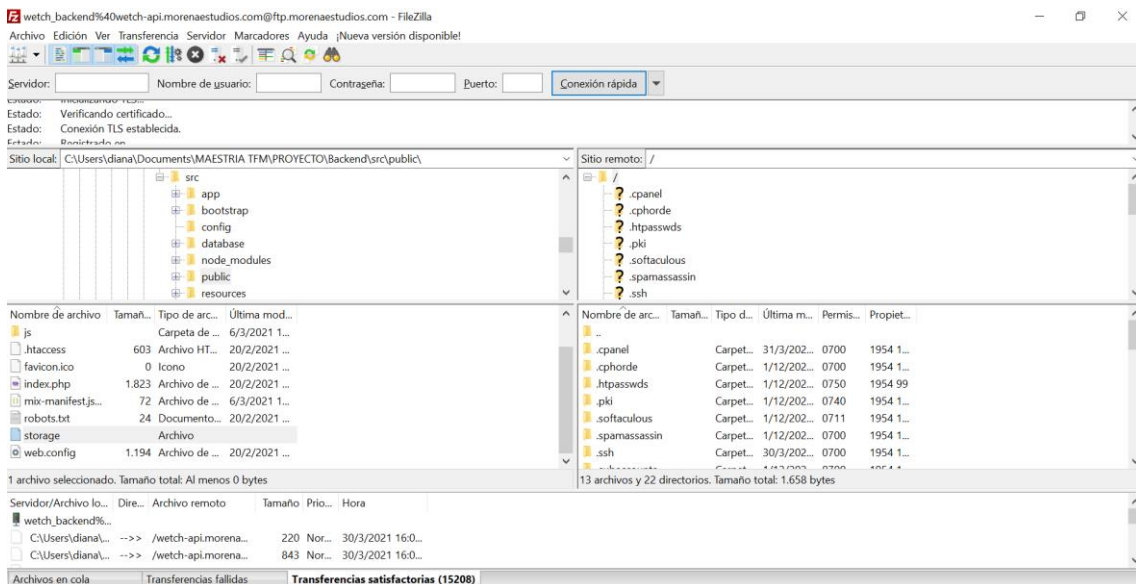


Figura 75. Cliente FTP
 Elaborado por: Diana Morocho

- Adicionalmente sobre el directorio storage, se debe modificar los permisos de acceso.

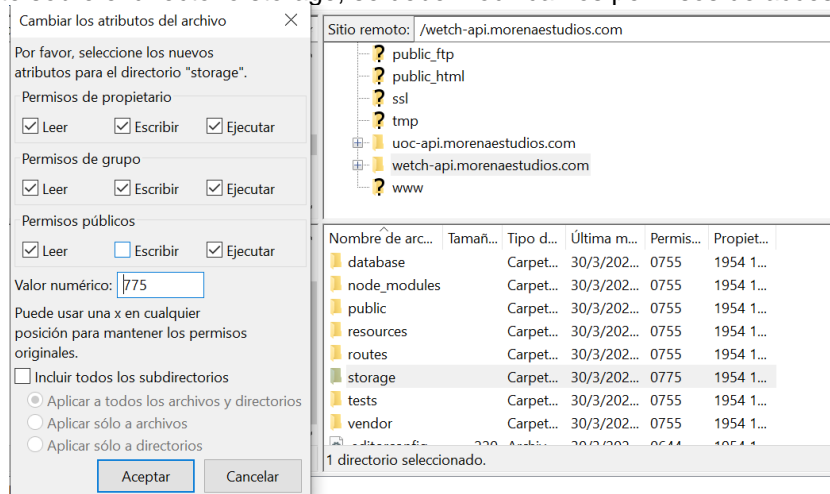
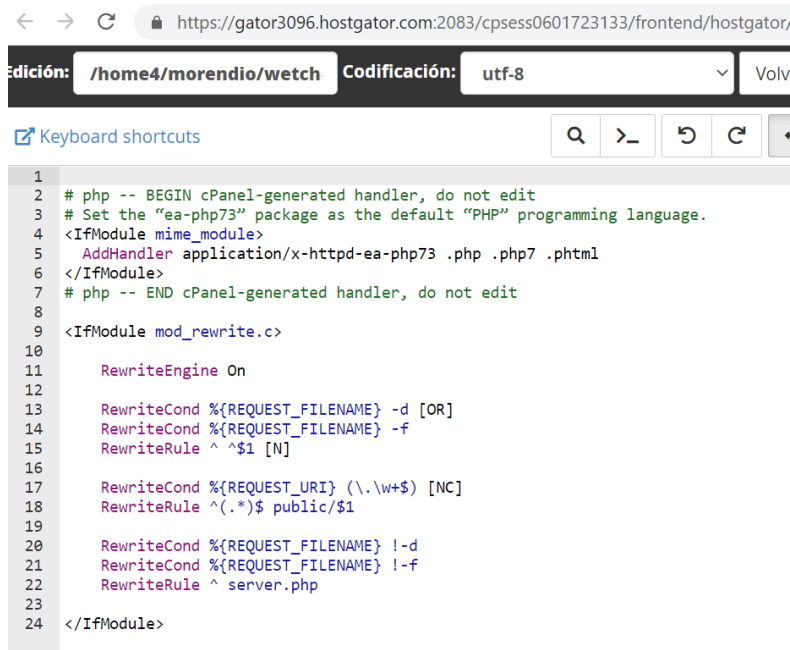


Figura 76. Permiso al directorio storage
 Elaborado por: Diana Morocho

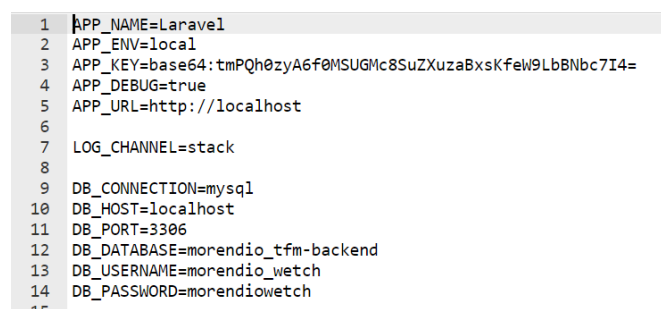
- Adicionalmente configurar un archivo .htaccess en el directorio raíz y /public



```
1 # php -- BEGIN cPanel-generated handler, do not edit
2 # Set the "ea-php73" package as the default "PHP" programming language.
3 <IfModule mime_module>
4   AddHandler application/x-httpd-ea-php73 .php .php7 .phtml
5 </IfModule>
6 # php -- END cPanel-generated handler, do not edit
7
8 <IfModule mod_rewrite.c>
9   RewriteEngine On
10
11   RewriteCond %{REQUEST_FILENAME} -d [OR]
12   RewriteCond %{REQUEST_FILENAME} -f
13   RewriteRule ^ ^$1 [N]
14
15   RewriteCond %{REQUEST_URI} (\.\w+$) [NC]
16   RewriteRule ^(\.*)$ public/$1
17
18   RewriteCond %{REQUEST_FILENAME} !-d
19   RewriteCond %{REQUEST_FILENAME} !-f
20   RewriteRule ^ server.php
21
22 </IfModule>
23
24
```

Figura 77. Configuración del archivo .htaccess
Elaborado por: Diana Morocho

- Configurar el archivo .env de acuerdo al nombre y usuario de la base de datos



```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:tmPQh0zyA6f0MSUGMc8SuZxzaBxsKfeW9LbBNbc7I4=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8
9 DB_CONNECTION=mysql
10 DB_HOST=localhost
11 DB_PORT=3306
12 DB_DATABASE=morendio_tfm-backend
13 DB_USERNAME=morendio_wetch
14 DB_PASSWORD=morendiowetch
15
```

Figura 78. Configuración del archivo .env
Elaborado por: Diana Morocho

- Se configura mediante SSH la ruta de acceso al storage desde la carpeta pública. Para ello, se utiliza el aplicativo Putty, donde se agregan las claves de acceso desde la configuración del servidor.

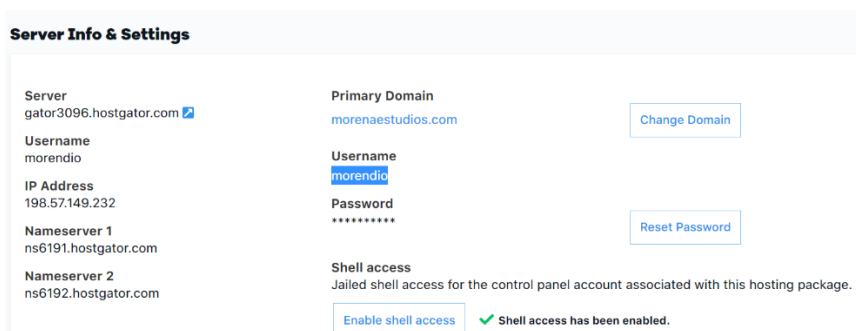
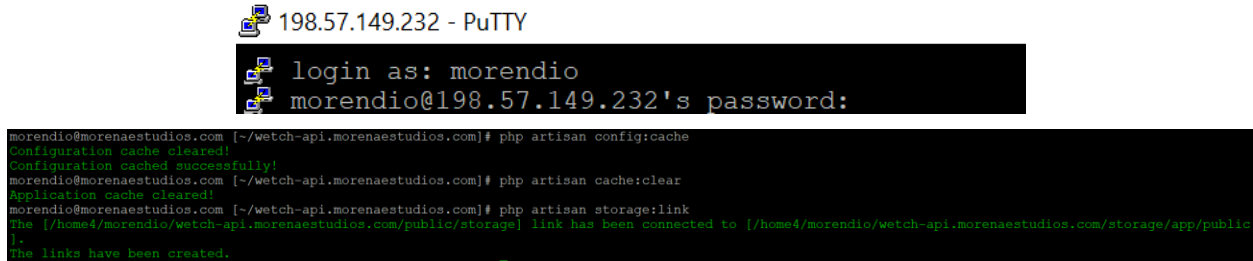


Figura 79. Habilitación de claves SSH
Elaborado por: Diana Morocho

- Con las claves SSH de acceso proporcionadas por el servidor y mediante la aplicación PuTTY, acceder a la ruta de alojamiento de los archivos de Laravel, generar una limpieza de la caché y se genera una nueva ruta pública del storage



```
198.57.149.232 - PuTTY
login as: morendio
morendio@198.57.149.232's password:

morendio@morenaestudios.com [~/wetch-api.morenaestudios.com]# php artisan config:cache
Configuration cache cleared!
Configuration cached successfully!
morendio@morenaestudios.com [~/wetch-api.morenaestudios.com]# php artisan cache:clear
Application cache cleared!
morendio@morenaestudios.com [~/wetch-api.morenaestudios.com]# php artisan storage:link
The [/home4/morendio/wetch-api.morenaestudios.com/public/storage] link has been connected to [/home4/morendio/wetch-api.morenaestudios.com/storage/app/public].
The links have been created.
```

Figura 80. Acceso mediante terminal
Elaborado por: Diana Morocho

- Finalmente se verifica el acceso a la aplicación

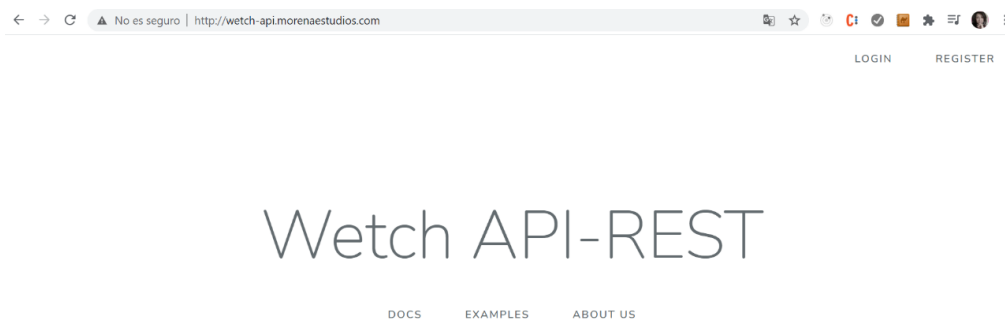
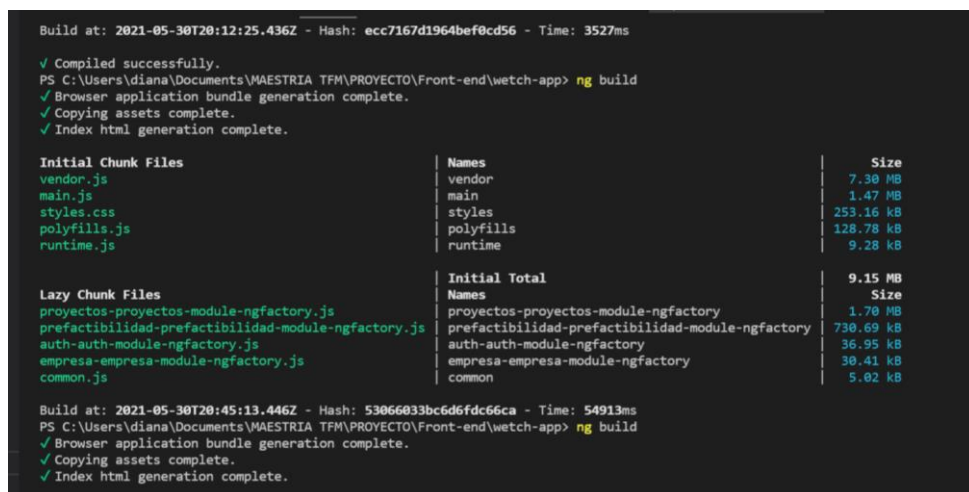


Figura 81. Api rest en servidor
Elaborado por: Diana Morocho

3.2. Implantación del Front-End

- Para la implantación del Front-End, previamente se debe compilar el proyecto, utilizando el comando `ng build --prod`. El asistente de angular cli, generará los archivos dentro de la carpeta `dist` que serán los que se copien en el servidor de producción



```
Build at: 2021-05-30T20:12:25.436Z - Hash: ecc7167d1964bef0cd56 - Time: 3527ms
✓ Compiled successfully.
PS C:\Users\diana\Documents\MAESTRIA TFM\PROYECTO\Front-end\wetch-app> ng build
✓ Browser application bundle generation complete.
✓ Copying assets complete.
✓ Index html generation complete.

Initial Chunk Files | Names | Size
---|---|---
vendor.js | vendor | 7.30 MB
main.js | main | 1.47 MB
styles.css | styles | 253.16 kB
polyfills.js | polyfills | 128.78 kB
runtime.js | runtime | 9.28 kB

Initial Total | 9.15 MB
---|---|---
Names | Size
---|---|---
projectos-proyectos-module-ngfactory.js | proyectos-proyectos-module-ngfactory | 1.70 MB
prefactibilidad-prefactibilidad-module-ngfactory.js | prefactibilidad-prefactibilidad-module-ngfactory | 730.69 kB
auth-auth-module-ngfactory.js | auth-auth-module-ngfactory | 36.95 kB
empresa-empresa-module-ngfactory.js | empresa-empresa-module-ngfactory | 30.41 kB
common.js | common | 5.02 kB

Build at: 2021-05-30T20:45:13.446Z - Hash: 53066033bc6d6fd6c6ca - Time: 54913ms
PS C:\Users\diana\Documents\MAESTRIA TFM\PROYECTO\Front-end\wetch-app> ng build
✓ Browser application bundle generation complete.
✓ Copying assets complete.
✓ Index html generation complete.
```

Figura 82. Construcción de la SPA versión compilada para producción.
Elaborado por: Diana Morocho

- A continuación, se sube el contenido de la carpeta dist al directorio público del servidor.

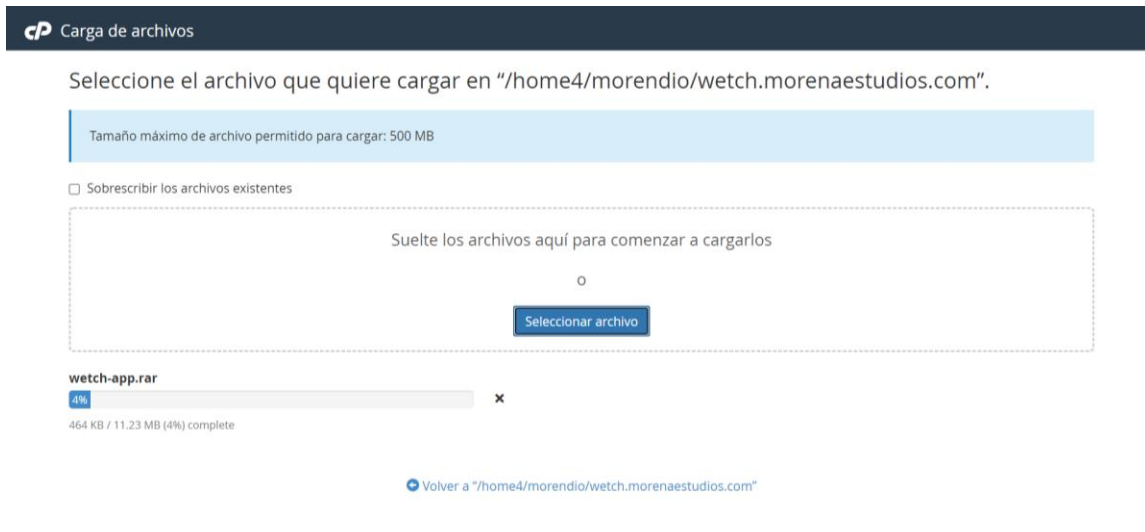


Figura 83. Subida de archivos al servidor.
Elaborado por: Diana Morocho

- De acuerdo a la documentación oficial, se debe agregar un archivo .htaccess para redireccionar la navegación de las páginas no encontradas al archivo index.html

```
sobrescritas.  
14 RewriteEngine On  
15 # If an existing asset or directory is requested go to it as it  
16 RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -f [OR]  
17 RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -d  
18 RewriteRule ^ - [L]  
19 # If the requested resource doesn't exist, use index.html  
20 RewriteRule ^ /index.html  
21
```

Figura 84. Configuración de acceso .htaccess
Elaborado por: Diana Morocho



Figura 85. Acceso a la aplicación en producción
Elaborado por: Diana Morocho

CAPÍTULO VI. DISCUSIÓN

1. Proyección a futuro

Una característica importante del desarrollo de la aplicación Wetch, es la escalabilidad, ya que su arquitectura está diseñada para adaptarse a las circunstancias cambiantes del entorno de la construcción. Actualmente, el sistema cuenta con un módulo para la creación de proyectos, que bien es usado para una de las fases iniciales en la construcción de proyectos arquitectónicos, como es el cálculo de prefactibilidad, de ahí que, para su expansión, se puede escalar a la siguiente fase de un proyecto arquitectónico con módulos para realizar cálculos de presupuestos con materiales de construcción, insumos, maquinaria u otros que permitirán obtener rápidamente el presupuesto de una construcción.

Es importante considerar el crecimiento de la empresa, es así que, es posible expandir los cálculos para que el sistema pueda ampliar sus funcionalidades en otros ámbitos de la construcción como edificios y conjuntos habitacionales, donde cada uno presenta características y regulaciones diferentes.

El software Wetch, también puede ser adaptado para otras empresas que dediquen su actividad profesional a la gestión de proyectos arquitectónicos.

Finalmente, la aplicación podría ampliarse para acompañar a los altos estándares de construcción y optimización de recursos como proyectos de vanguardia en el ámbito de optimización energética.

2. Conclusiones

El desarrollo de la SPA Wetch permitió implementar una aplicación web que automatiza la fase de cálculo de prefactibilidad de los proyectos de construcción de vivienda y a la vez en cada fase del desarrollo de software, cumplir con todos los objetivos formulados en el presente proyecto.

Es así que, para determinar los requerimientos de software de la aplicación web, se realizó un contacto directo mediante observación con los usuarios, permitiendo identificar sus intereses y analizar sus perfiles de usuario para aplicarlos como parte de la Experiencia de Usuario en la navegabilidad del software, así como fue de gran utilidad para determinar sus necesidades y poder ser automatizadas. El desarrollo de esta actividad facilitó el cumplimiento de la fase de planeación de la metodología de software XP, donde se determinó historias de usuario, velocidad del proyecto, roles, requerimientos funcionales, requerimientos no funcionales y definición del plan de iteraciones, cuya información detallada se encuentra en el apartado del Capítulo II. Análisis de este documento.

A continuación, se aplicó la fase de Diseño, en este punto, se tomaron como insumos los artefactos obtenidos de la fase de análisis. Aquí se desarrollaron modelos conceptuales, el diseño de la arquitectura de la aplicación tanto para el back-end como el front-end y la selección de toda la plataforma de herramientas a utilizar para el desarrollo. Es importante mencionar que en este punto de la construcción del software se pudo aplicar los nuevos aprendizajes obtenidos del master como: aplicar los principios de la experiencia de usuario, la selección adecuada de herramientas del boilerplate para el desarrollo de la SPA y la prototipación de interfaces de usuario aplicando los conceptos del diseño web Mobile-First. Toda esta información puede ser ampliada, dando lectura al apartado Capítulo III. Diseño de este documento.

En el planteamiento de los objetivos de la investigación se propuso implementar los módulos para el cálculo de prefactibilidad en proyectos de construcción de vivienda bajo la arquitectura Single Page Application, aquí se puso en acción las buenas prácticas de construcción de software y los conocimientos obtenidos de las asignaturas impartidas en el master como la configuración de las herramientas del back-end para la construcción de una API-REST propia y la generación de interfaces dinámicas en el front-end para gestionar los datos ingresados por el usuario. Se codificaron los módulos definidos en la fase de diseño y se validaron los requerimientos de la fase del análisis de software a modo de iteraciones como lo propone la metodología XP. Para ampliar esta información, acceder al Capítulo 4. Implementación de este documento.

Finalmente, previo a la implantación, se realizaron varias pruebas de funcionalidad al sistema y su validación con el usuario final, permitiendo mejorar los detalles de la implementación, de tal manera que se pudo establecer el estado favorable de aceptabilidad, usabilidad, efectividad, utilidad del sistema frente a los usuarios.

Anexo 1. Entregables del proyecto

- Archivo: PAC_FINAL_MorochoPuchaicela_DianaAlexandra.zip
Contenido: Archivo comprimido de toda la documentación, que incluye la memoria, código fuente, presentación y video.
 - Carpeta: PAC_FINAL_mem_MorochoPuchaicela_DianaAlexandra
Contenido:
 - Archivo: PAC_FINAL_mem_MorochoPuchaicela_DianaAlexandra.pdf
Contiene: Memoria completa del trabajo de fin de master en formato pdf
 - Archivo: PAC_FINAL_manual_MorochoPuchaicela_DianaAlexandra.pdf
Contiene: Archivo con instrucciones para usuario en formato .pdf
 - Carpeta: PAC_FINAL_prj_MorochoPuchaicela_DianaAlexandra
Contenido:
 - Archivo: Backend.zip
Contiene: Código fuente del back-end en archivo comprimido .zip
 - Archivo: Frontend.zip
Contiene: Código fuente del front-end en archivo comprimido .zip
 - Archivo: Database.zip
Contiene: Archivo de estructura de base de datos en formato .sql
 - Carpeta: PAC_FINAL_vid_MorochoPuchaicela_DianaAlexandra
Contenido:
 - Archivo: Presentacion.mp4
Contiene: Archivo de video de la presentación del proyecto.
 - Archivo: PresentacionVideo.pdf
Contiene: Archivo de presentación utilizada en el video.
 - Carpeta: PAC_FINAL_prs_MorochoPuchaicela_DianaAlexandra
Contenido:
 - Archivo: Presentacion.exe
Contiene: Archivo de presentación de la aplicación.

Anexo 2. Librerías utilizadas

Las siguientes librerías fueron utilizadas para el desarrollo del front-end.

- **FONTAWESOME.**

Librería de íconos en diversos formatos para ser utilizados en la web. La librería está disponible en: <https://fontawesome.com/>

- **HIGHCHARTS.**

Librería javascript para la creación de gráficos estadísticos. Disponible en: <https://www.highcharts.com/>

- **NGRX.**

Librería de implementación del patrón Redux en Angular. Disponible en: <https://ngrx.io/>

- **ANGULAR MATERIAL.**

Librería de diseño de componentes para Angular basado en Material Design. Disponible en: <https://material.angular.io/>

- **ANGULAR FLEX LAYOUT.**

Módulo de componentes Layout Api usando FlexBox Css + mediaQuery. Disponible en: <https://github.com/angular/flex-layout>

- **NGRX STORE DEV TOOLS**

Entorno gráfico para la gestión del estado basado en Redux para Google Chrome. Disponible en: <https://ngrx.io/guide/store-devtools>

- **NGRX EFFECTS**

Librería de funciones para realizar acciones asociadas a las Actions del patrón Redux. Disponible en: <https://ngrx.io/guide/effects>

- **LOCAL STORAGE NGRX**

Librería de almacenamiento automático de estado de la aplicación basada en Redux en local storage. Disponible en: <https://www.npmjs.com/package/ngrx-store-localstorage>

Librerías para el desarrollo del Backend

- **LARAVEL UI.**

Paquete de autenticación para Laravel. Instalación disponible vía composer mediante el comando: `composer require laravel/ui`

- **LARAVEL SANCTUM**

Paquete de autenticación por cookies para SPA y Laravel. Instalación disponible vía composer mediante el comando: `composer require laravel/sanctum`

- **LARAVEL SPATIE**

Paquete para a gestión de roles y permisos en Laravel. Instalación disponible vía composer mediante el comando: `composer require spatie/laravel-permission`

- **SWIFTMAILER**

Paquete para el envío de correos electrónicos mediante peticiones http. Instalación disponible vía composer mediante el comando: `composer require wilddbit/swiftmailer-postmark`

Anexo 3. Ciclo de vida NgRx

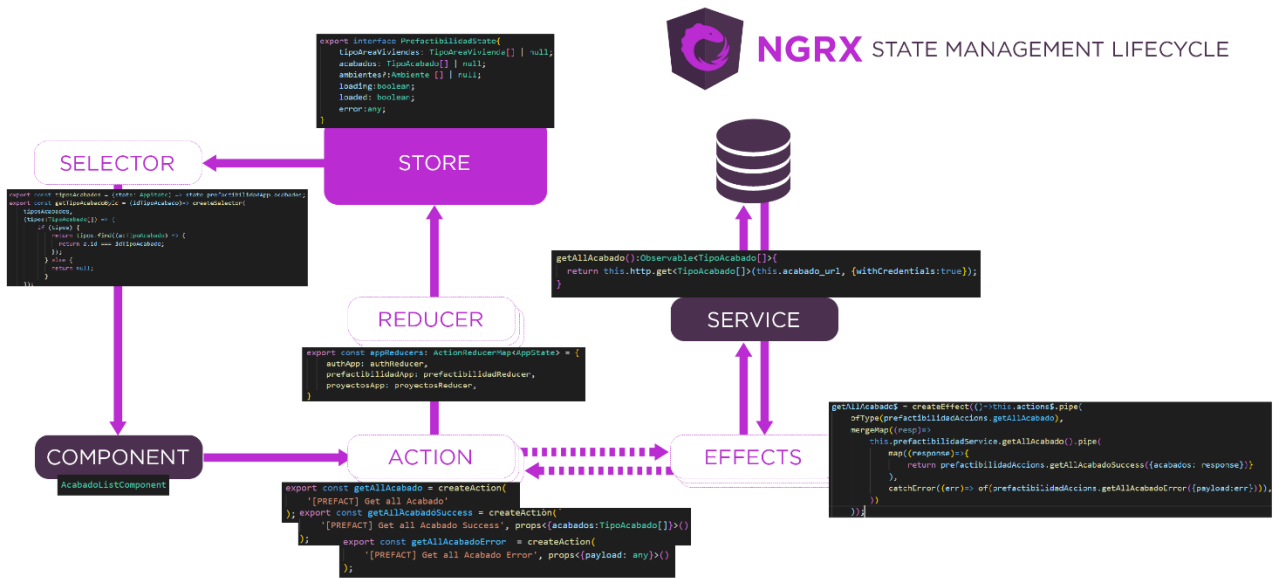


Figura 86. Ciclo de vida NgRx
 Elaborado por: Diana Morocho

Anexo 4. Libro de estilo

Dado que la aplicación está enfocada para ser utilizada en la empresa de arquitectura y construcción Wetch, se han utilizado los colores corporativos y tipos de letra ya definidos para el desarrollo del software.

- Logotipo y sus variaciones



Figura 87. Logotipo de la empresa Wetch.
Elaborado por: Diana Morocho

- Paleta de colores.

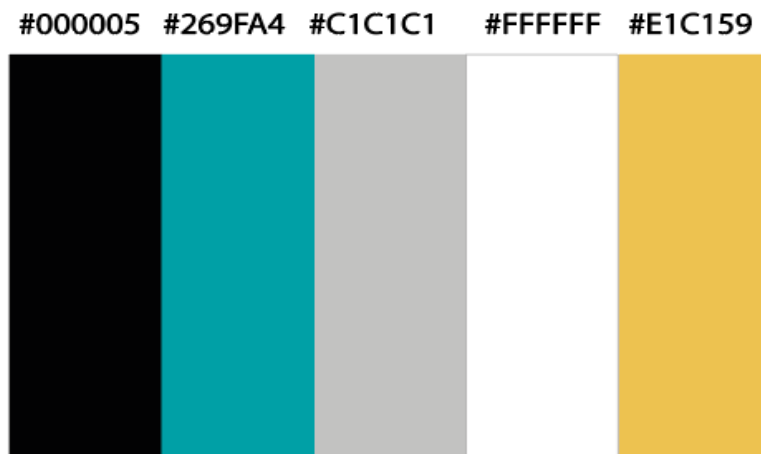


Figura 88. Paleta de colores corporativa
Elaborado por: Diana Morocho

- Tipografía.

Tipografías usadas:

WETCH ← **baubaus 93 regular**
arquitectura + construcción ← CENTURY GOTHIC REGULAR

Figura 89. Tipografía corporativa
Elaborado por: Diana Morocho

- Otros elementos gráficos.



Figura 90. Imagen de pie de página
Elaborado por: Diana Morocho

Anexo 5. Glosario

Artefacto. Es un producto tangible resultante del proceso de desarrollo de software. (Cardozzo, 2016)

Rubro. Tareas preliminares o equipo que se contrata para su uso y montaje. (Jorge, 2021)

Costo. Los costos básicos de una obra están conformados por: materiales, mano de obra, equipos y herramientas, gastos generales e impuestos. (Domingo, 2021)

Prefactibilidad. Análisis preliminar de una construcción para determinar si es viable y convertirla en un proyecto arquitectónico. (Botero, 2014)

Acabado. Se conoce como acabados, revestimientos o recubrimientos a todos aquellos materiales que se colocan sobre una superficie de obra negra. (Botero, 2014)

Área de construcción. Parte edificada que corresponde a la suma de las superficies de los pisos. (Cardozzo, 2016)

Anexo 6. Bibliografía

- Angular. (12 de Marzo de 2021). *Introduction to Angular Concepts*. Obtenido de <https://angular.io/guide/architecture>
- Angular Material. (18 de Marzo de 2021). Obtenido de <https://material.angular.io/>
- Botero, A. (2014). *Factibilidad para la construcción de un edificio en un lote o vivienda unifamiliar*. Obtenido de https://repository.eafit.edu.co/bitstream/handle/10784/5136/Alejandro_BoteroMariaca_JohnnyAndres_BedoyaMunoz_2014.pdf?sequence=2&isAllowed=y
- Cardozzo, D. R. (2016). *Desarrollo de software. Requisitos, estimaciones y análisis*. IT Campus Academy.
- Castán, Y. (2014). *Introducción al método científico y sus etapas*. Obtenido de <http://www.haykhuyay.com/A1/Generic/ECO1/U1U2/metodoCientifico.pdf>
- Ceria, S. (2012). *Ingeniería de software I. Casos de uso, un método práctico para explorar requerimientos*.
- Composer. (17 de Marzo de 2021). *Composer Doc*. Obtenido de <https://getcomposer.org/>
- Docker. (17 de Marzo de 2021). *Docker Blog*. Obtenido de <https://www.docker.com/blog/>
- Domingo, C. (2021). *Arquitectura y construcciones sostenibles*. California.
- Echeverry, L., & Delgado, L. (13 de Marzo de 2021). *Caso práctico de la metodología ágil XP al desarrollo de software*. Obtenido de <http://repositorio.utp.edu.co/dspace/handle/11059/794>
- Fontawesome. (18 de Marzo de 2021). *Fontawesome*. Obtenido de <https://fontawesome.com/how-to-use/on-the-web/referencing-icons/basic-use>
- Google Analytics. (12 de Marzo de 2021). *Single Page Application Meashure*. Obtenido de <https://developers.google.com/analytics/devguides/collection/analyticsjs/single-page-applications?hl=es-419>
- Informe Nacional del Ecuador. (2015). *Tercera Conferencia de las Naciones Unidas sobre la vivienda y el desarrollo urbano sostenible. Habitat III*. Obtenido de https://www.habitatyvivienda.gob.ec/wp-content/uploads/downloads/2017/05/Informe-Pais-Ecuador-Enero-2016_vf.pdf
- Jaramillo, W. (06 de Junio de 2018). *Software Obras*. Obtenido de <https://softwareobras.blogspot.com/2018/06/curso-virtual-presupuestos-de-obras.html>
- Jorge, V. (2021). *Ejecución de obra*. Argentina: Universidad de Buenos Aires.
- Laravel. (18 de Marzo de 2021). *Laravel Documentation*. Obtenido de <https://laravel.com/>
- Leteiler, p., & Penadéz, C. (2006). *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Valencia: Universidad Politécnica de Valencia.
- Lopez, M. G. (8 de Mayo de 2020). *Aplicación web para calcular el índice de calidad de vida, en función del interés de un usuario con fines migratorios*. Catalunya: UOC.
- Lucidchart. (11 de Marzo de 2021). *Que es un wireframe para un sitio web*. Obtenido de <https://www.lucidchart.com/pages/es/que-es-un-wireframe-para-un-sitio-web>

- Madhuri, J., Balkrishna, S., & Anushree, D. (2015). *Single Page Application using AngularJS*. Obtenido de <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.736.4771&rep=rep1&type=pdf>
- MDN Web Docs. (27 de Marzo de 2021). *Conceptos básicos de HTML*. Obtenido de https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/HTML_basics
- MDN Web Docs. (11 de Marzo de 2021). *Generalidades del protocolo HTTP*. Obtenido de <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>
- Ministerio de Desarrollo Urbano y Vivienda. (27 de Febrero de 2021). *Norma Ecuatoriana de la Construcción*. Obtenido de <https://www.habitatyvivienda.gob.ec/norma-ecuatoriana-de-la-construccion/>
- Muro, J. A. (27 de Marzo de 2021). *Deleoite*. Obtenido de <https://www2.deloitte.com/es/es/pages/technology/articles/que-es-orm.html>
- MySQL. (21 de Marzo de 2021). *MySQL Documentation*. Obtenido de <https://dev.mysql.com/doc/>
- NgRx. (27 de Marzo de 2021). *What is NgRx?* Obtenido de <https://ngrx.io/docs>
- NgRx Community. (14 de Marzo de 2021). *Reactive State for Angular*. Obtenido de <https://ngrx.io/>
- Organización de Naciones Unidas. (10 de Diciembre de 1948). *La Declaración Universal de Derechos Humanos*. Obtenido de <https://www.un.org/es/universal-declaration-human-rights/#:~:text=Toda%20persona%20tiene%20derecho%20a%20un%20nivel%20de%20vida%20adecuado,de%20desempleo%2C%20enfermedad%2C%20invalidez%2C>
- Php. (17 de Marzo de 2021). *Php Documentation*. Obtenido de <https://www.php.net/manual/es/intro-what-is.php>
- Quercusoft. (18 de Marzo de 2021). *Quercusoft Características*. Obtenido de <http://quercusoft.com/caracteristicas/>
- Rodríguez, J. S. (2011). *Matemáticas financieras y evaluación de proyectos*.
- Salazar, I. P. (2011). *Guía práctica para la identificación, formulación y evaluación de proyectos*.
- Sass. (21 de Marzo de 2021). *Sass Blog*. Obtenido de <https://sass-lang.com/>
- Typescript. (14 de Marzo de 2021). *Typescript Docs*. Obtenido de <https://www.typescriptlang.org/>
- Unger, R., & Chandler, C. (2009). *A project guide to UX Design: For user experience designers in the field or in the making*. New Riders.

Anexo 7. Vita



Diana Alexandra Morocho Puchaicela

Ingeniera en Sistemas graduada de la Universidad Nacional de Loja - Ecuador, periodo 2007 – 2012, especializada en análisis, diseño e implementación de software con lenguajes de programación Python, Php, Typescript.

Estudiante de master universitario en Desarrollo de Sitios y aplicaciones web por la Universidad Oberta de Catalunya.

Áreas de interés: ingeniería de requisitos, ingeniería de software, diseño de experiencia de usuario, desarrollo front-end.

Provincia de Loja, ciudad Loja, Ecuador 2021.