



UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E INFORMÁTICOS

Tema:

IMPLEMENTACIÓN DE PLATAFORMA WEB MEDIANTE LA APLICACIÓN DE LA METODOLOGÍA OBJECT ORIENTED HYPERMEDIA DESIGN METHOD (OOHDM), PARA APOYO A MICRO, PEQUEÑAS Y MEDIANAS EMPRESAS (MIPYMES) AFECTADAS EN SUS VENTAS ANTE LA EMERGENCIA SANITARIA EN LA PROVINCIA DE NAPO

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos

ÁREA: Software

LÍNEA DE INVESTIGACIÓN: Desarrollo de software

AUTOR: Luis Alfredo Guachamin Yumbo

TUTOR: Ing. PhD. Félix Oscar Fernández Peña, MSc.

Ambato - Ecuador

marzo – 2022

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Titulación con el tema: IMPLEMENTACIÓN DE PLATAFORMA WEB MEDIANTE LA APLICACIÓN DE LA METODOLOGÍA OBJECT ORIENTED HYPERMEDIA DESIGN METHOD (OOHDM), PARA APOYO A MICRO, PEQUEÑAS Y MEDIANAS EMPRESAS (MIPYMES) AFECTADAS EN SUS VENTAS ANTE LA EMERGENCIA SANITARIA EN LA PROVINCIA DE NAPO, desarrollado bajo la modalidad Proyecto de Investigación por el señor Luis Alfredo Guachamin Yumbo, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, marzo 2022.

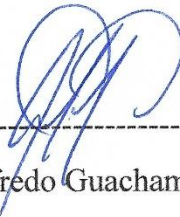
Ing. PhD. Félix Oscar Fernández Peña, MSc.

TUTOR

AUTORÍA

El presente Proyecto de Investigación titulado: IMPLEMENTACIÓN DE PLATAFORMA WEB MEDIANTE LA APLICACIÓN DE LA METODOLOGÍA OBJECT ORIENTED HYPERMEDIA DESIGN METHOD (OOHDM), PARA APOYO A MICRO, PEQUEÑAS Y MEDIANAS EMPRESAS (MIPYMES) AFECTADAS EN SUS VENTAS ANTE LA EMERGENCIA SANITARIA EN LA PROVINCIA DE NAPO es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, marzo 2022.



Luis Alfredo Guachamin Yumbo

C.C. 1500955495

AUTOR

APROBACIÓN TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Luis Alfredo Guachamin Yumbo, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado IMPLEMENTACIÓN DE PLATAFORMA WEB MEDIANTE LA APLICACIÓN DE LA METODOLOGÍA OBJECT ORIENTED HYPERMEDIA DESIGN METHOD (OOHDM), PARA APOYO A MICRO, PEQUEÑAS Y MEDIANAS EMPRESAS (MIPYMES) AFECTADAS EN SUS VENTAS ANTE LA EMERGENCIA SANITARIA EN LA PROVINCIA DE NAPO, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidenta del Tribunal.

Ambato, marzo 2022.

Ing. Pilar Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL

Ing. Franklin Mayorga
PROFESOR CALIFICADOR

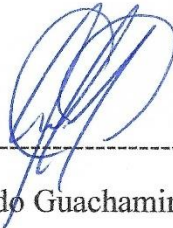
Ing. Rubén Nogales
PROFESOR CALIFICADOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, marzo 2022.



Luis Alfredo Guachamin Yumbo

C.C. 1500955495

AUTOR

DEDICATORIA

Dedico con todo mi corazón el presente proyecto a mis padres Sonia Yumbo y Luis Guachamin por ser los pilares fundamentales en mi formación, en especial a mi madre por su apoyo incondicional, esfuerzo y sacrificio por verme crecer y cumplir mis sueños pese a los momentos difíciles.

A mis hermanas Keyko y Evelyn por su apoyo constante en cada etapa de mi vida.

A mi sobrina Sofía por la alegría que ha traído a mi vida.

A mi abuelo Francisco que está en el cielo y siempre ha sido mi ejemplo de superación.

Luis Alfredo Guachamin Yumbo

AGRADECIMIENTO

A Dios, por ser mi guía y darme la fortaleza necesaria para seguir adelante, por la salud y todas las bendiciones a lo largo de mi vida.

Agradezco a mi madre por todo el esfuerzo realizado para que yo llegara a cumplir mis sueños, por sus consejos, por sus enseñanzas, por su paciencia y amor, por ser mi ejemplo a seguir para no rendirme y por enseñarme que puedo conseguir todo lo que me proponga. A mis hermanas y mi familia por todo su apoyo brindado en cada momento.

A mis amigos Leonel, Jonathan, Anthony, Liseth, Lisbeth y Adriana que con su amistad, compañerismo y apoyo moral aportaron para que mis ganas de seguir adelante no desvanezcan, agradezco infinitamente a mi amiga Johanna por su amistad, por su apoyo, por las risas, por los momentos compartidos, por ser mi fuente de inspiración y por permitirme aprender más de la vida.

A la Universidad Técnica de Ambato, en especial a la Facultad de Ingeniería en Sistemas, Electrónica e Industrial y a sus docentes por abrirme las puertas a un mar de conocimientos y experiencias que me forjaron profesionalmente.

A mi tutor el Ing. Félix Fernández, por su paciencia y por haberme guiado con su conocimiento en la elaboración de este proyecto.

A todas las personas, compañeros y amigos que formaron parte de esta etapa y aportaron con su granito de arena para que yo llegara al punto en el que me encuentro.

A todos, muchas gracias.

Luis Alfredo Guachamin Yumbo

ÍNDICE DE CONTENIDOS

APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
APROBACIÓN TRIBUNAL DE GRADO	iv
DERECHOS DE AUTOR	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
RESUMEN EJECUTIVO	xix
ABSTRACT	xx
CAPÍTULO I	1
1. MARCO TEÓRICO	1
1.1. Tema de investigación.....	1
1.2. Antecedentes investigativos.....	1
1.2.1. Contextualización del problema.....	1
1.2.2. Delimitación.....	3
1.2.3. Justificación.....	3
1.2.4. Fundamentación teórica.....	4
1.3. Objetivos.....	17
1.3.1. General.....	17
1.3.2. Específicos.....	17
CAPÍTULO II	18
2. METODOLOGÍA	18
2.1. Materiales.....	18
2.2. Métodos.....	20
2.2.1. Modalidad de la investigación.....	20
2.2.2. Población y muestra.....	21
2.2.3. Recolección de información.....	21
2.2.4. Procesamiento y análisis de datos.....	29
CAPITULO III	31
3. RESULTADOS Y DISCUSIÓN	31
3.1. Análisis y discusión de resultados.....	31
3.1.1. Object Oriented Hypermedia Design Method (OOHDM).....	31

3.1.2.	Integración de OOHDM con UML	39
3.2.	Desarrollo de la propuesta	41
3.2.1.	Primera etapa – Recopilación de requisitos	41
3.2.2.	Segunda etapa - Diseño conceptual.....	62
3.2.3.	Tercera etapa - Diseño navegacional	66
3.2.4.	Cuarta etapa - Diseño de interfaces abstractas.....	72
3.2.5.	Quinta etapa - Implementación	85
3.2.6.	Pruebas	112
3.2.7.	Despliegue de la plataforma web	122
CAPÍTULO IV	124
CONCLUSIONES Y RECOMENDACIONES	124
4.1.	Conclusiones	124
4.2.	Recomendaciones	125
Bibliografía	126
Anexos	131
Anexo 1. Cuestionario de la evaluación heurística con los principios de Nielsen para evaluar la usabilidad de la plataforma web.	131

ÍNDICE DE TABLAS

Tabla 2.1 Resultados de encuesta - Pregunta 1	21
Tabla 2.2 Resultados de encuesta - Pregunta 2	22
Tabla 2.3 Resultados de encuesta - Pregunta 3	23
Tabla 2.4 Resultados de encuesta - Pregunta 4	24
Tabla 2.5 Resultados de encuesta - Pregunta 5	25
Tabla 2.6 Resultados de encuesta - Pregunta 6	26
Tabla 2.7 Resultados de encuesta - Pregunta 7	27
Tabla 2.8 Resultados de encuesta - Pregunta 8	28
Tabla 3.1 Resumen de la metodología OOHDm	33
Tabla 3.2 Actores	43
Tabla 3.3 Caso de uso - Iniciar sesión	46
Tabla 3.4 Caso de uso - Crear cuenta.....	47
Tabla 3.5 Caso de uso - Consultar catálogo de empresas	48
Tabla 3.6 Caso de uso - Filtrar empresas	49
Tabla 3.7 Caso de uso - Consultar perfil de una empresa.....	50
Tabla 3.8 Caso de uso - Registrar empresa.....	51

Tabla 3.9 Caso de uso - Gestionar datos de la empresa	52
Tabla 3.10 Caso de uso - Gestionar datos de la cuenta	53
Tabla 3.11 Caso de uso - Comentar perfil de empresa	54
Tabla 3.12 Caso de uso – Reaccionar sobre el perfil de una empresa	55
Tabla 3.13 Caso de uso - Gestionar usuarios	56
Tabla 3.14 Caso de uso - Gestionar información de las tablas	57
Tabla 3.15 Caso de uso - Gestionar comentarios	58
Tabla 3.16 Caso de uso - Registrar nuevo administrador	59
Tabla 3.17 Caso de uso - Gestionar cuenta administrativa	60
Tabla 3.18 Caso de uso - Gestionar empresas.....	61
Tabla 3.19 Rutas de la API-REST	97

ÍNCIDE DE FIGURAS

Figura 1.1 Esquema básico del servicio web	7
Figura 1.2 Patrón de diseño MVC	10
Figura 2.1 Resultados de encuesta - Pregunta 1	22
Figura 2.2 Resultados de encuesta - Pregunta 2.....	23
Figura 2.3 Resultados de encuesta - Pregunta 3.....	24
Figura 2.4 Resultados de encuesta - Pregunta 4.....	25
Figura 2.5 Resultados de encuesta - Pregunta 5.....	26
Figura 2.6 Resultados de encuesta - Pregunta 6.....	27
Figura 2.7 Resultados de encuesta - Pregunta 7.....	28
Figura 2.8 Resultados de encuesta - Pregunta 8.....	29
Figura 3.1 Fases en el diseño e implementación de aplicaciones web.....	32
Figura 3.2 Relaciones de generalización entre actores	44
Figura 3.3 Diagrama de casos de uso.....	45
Figura 3.4 Modelo conceptual de la plataforma web.....	62
Figura 3.5 Diagrama de clases – Usuarios	63
Figura 3.6 Diagrama de clases - Gestión de MiPymes	64

Figura 3.7 Diagrama de clases – MiPymes.....	65
Figura 3.8 Esquema de clases navegacionales – Visitante de sitio.....	67
Figura 3.9 Esquema de clases navegacionales - Usuario autenticado	67
Figura 3.10 Esquema de clases navegacionales – Administrador.....	68
Figura 3.11 Contexto navegacional - Visitante de sitio	69
Figura 3.12 Contexto navegacional - Usuario autenticado	70
Figura 3.13 Contexto navegacional – Administrador	71
Figura 3.14 Diagrama de navegación final	72
Figura 3.15 ADV – Barra de navegación.....	73
Figura 3.16 ADV – Pie de página	73
Figura 3.17 ADV – Inicio	73
Figura 3.18 ADV - Acerca de	74
Figura 3.19 ADV - Inicio de sesión	74
Figura 3.20 ADV – Registro	75
Figura 3.21 ADV - Perfil de MiPyme.....	75
Figura 3.22 ADV – Barra de navegación de usuario autenticado.....	76
Figura 3.23 ADV - Perfil de usuario autenticado	76

Figura 3.24 ADV – Actividad de usuario autenticado.....	76
Figura 3.25 ADV - Editar perfil de usuario autenticado.....	77
Figura 3.26 ADV - Cambiar contraseña	77
Figura 3.27 ADV - Eliminar cuenta de usuario autenticado.....	78
Figura 3.28 ADV - Registro de empresa.....	78
Figura 3.29 ADV - Gestionar empresa	79
Figura 3.30 ADV - Editar datos de la empresa	79
Figura 3.31 ADV - Redes y canales de venta de la empresa	80
Figura 3.32 ADV - Galería de imágenes de la empresa.....	80
Figura 3.33 ADV - Horario de la empresa.....	81
Figura 3.34 ADV - Actividad de la empresa.....	81
Figura 3.35 ADV – Catálogo de guías	81
Figura 3.36 ADV – Guía.....	82
Figura 3.37 ADV - Barra de navegación del administrador	82
Figura 3.38 ADV - Menú lateral del administrador	82
Figura 3.39 ADV - Gestionar usuarios	83
Figura 3.40 ADV - Gestionar MiPymes	83

Figura 3.41 ADV - Gestionar cuenta administrativa	84
Figura 3.42 ADV - Gestionar administradores	84
Figura 3.43 ADV - Gestionar información de las tablas.....	85
Figura 3.44 ADV – Dashboard	85
Figura 3.45 Arquitectura de la aplicación	86
Figura 3.46 Modelo físico de la base de datos	86
Figura 3.47 API REST - Configuración de JWT	87
Figura 3.48 API REST - Registro de usuario.....	88
Figura 3.49 API REST - Inicio de sesión.....	88
Figura 3.50 API REST - Devolución del token	89
Figura 3.51 API REST - Obtener información del usuario.....	89
Figura 3.52 API REST - Actualización de datos del usuario.....	90
Figura 3.53 API REST - Cambio de contraseña	91
Figura 3.54 API REST - Cerrar sesión.....	91
Figura 3.55 API REST - Modelo Usuario.....	92
Figura 3.56 API REST - Modelo Mipyme.....	98
Figura 3.57 API REST - Lista de empresas	99

Figura 3.58 API REST - Registrar empresa.....	100
Figura 3.59 Configuración de las rutas de los usuarios	101
Figura 3.60 Configuración de las rutas de los recursos	102
Figura 3.61 Modelos	103
Figura 3.62 Controladores.....	103
Figura 3.63 Configuración de la ruta de acceso a la API.....	104
Figura 3.64 Configuración de HTTP Interceptor	104
Figura 3.65 Rutas protegidas.....	105
Figura 3.66 Consumo de servicios - Registro de usuario.....	105
Figura 3.67 Consumo de servicios - Inicio de sesión.....	106
Figura 3.68 Subscribe - Inicio de sesión	106
Figura 3.69 Subscribe - Registro de usuario	107
Figura 3.70 Subscribe - Registro de empresa.....	108
Figura 3.71 Subscribe - Lista de empresas.....	109
Figura 3.72 Rutas de navegación del frontend.....	110
Figura 3.73 Rutas de navegación del usuario	110
Figura 3.74 Rutas de navegación del administrador	111

Figura 3.75 Componentes del frontend.....	111
Figura 3.76 Núcleo del frontend	112
Figura 3.77 Estructura de diseño de los usuarios.....	112
Figura 3.78 Servicios	112
Figura 3.79 Prueba - Pantalla de "Registro"	113
Figura 3.80 Prueba - Pantalla de "Inicio de sesión"	113
Figura 3.81 Prueba - Pantalla de "Catálogo"	114
Figura 3.82 Prueba - Pantalla de "Perfil de empresa"	114
Figura 3.83 Prueba - Pantalla de "Registro de una empresa"	115
Figura 3.84 Prueba - Pantalla de "Gestión de la información de una empresa"	115
Figura 3.85 Prueba - Pantalla de "Perfil de usuario"	116
Figura 3.86 Prueba - Añadir comentario.....	116
Figura 3.87 Prueba - Reacción sobre el perfil de una empresa.....	117
Figura 3.88 Prueba - Pantalla de "Gestión de usuarios"	117
Figura 3.89 Prueba - Pantalla de "Gestión de tablas"	118
Figura 3.90 Prueba - Pantalla de "Gestión de empresas"	118
Figura 3.91 Prueba - Pantalla de "Gestión de administradores"	119

Figura 3.92 Prueba - Pantalla de "Gestión de cuenta administrativa"	119
Figura 3.93 Prueba - Pantalla de "Catálogo de guías"	119
Figura 3.94 Prueba - Pantalla de "Visualización de una guía"	120
Figura 3.95 Prueba - Pantalla de "Acerca de"	120
Figura 3.96 Resultado - Prueba de estrés 1 a "promocionatuempresa.online"	121
Figura 3.97 Resultado - Prueba de estrés 2 a "promocionatuempresa.online"	121
Figura 3.98 Resultados de la evaluación heurística	122
Figura 3.99 Configuración del archivo ".env" para producción	122
Figura 3.100 Configuración del archivo ".htaccess" para el backend	123
Figura 3.101 Configuración del archivo ".htaccess" para el frontend	123
Figura 3.102 Plataforma implementada vista desde el navegador web	123

RESUMEN EJECUTIVO

El confinamiento obligatorio en el país generado por la pandemia global a causa del Covid-19 intensificó el uso de la tecnología en diferentes ámbitos como la educación, el campo laboral y comercio electrónico, la necesidad por continuar con las actividades de la vida cotidiana a través del uso de estos medios ha generado dependencia de los mismos para poder subsistir, es el caso de las micro, pequeñas y medianas empresas que pese que la mayoría de ellas cuentan con conectividad a internet se han visto afectadas en sus ventas al no contar con los recursos tecnológicos suficientes para poder mantenerse activos en el mercado en el que se desenvuelven y hacer llegar sus productos o servicios a sus clientes.

El presente proyecto se apoyó en las fases de la metodología Object Oriented Hypermedia Design para su documentación y diseño mediante el uso del lenguaje unificado de modelado (UML), además se basó en el aprovechamiento de herramientas tecnológicas gratuitas que permitieron acelerar el proceso de desarrollo de la plataforma web para ayudar a la promoción de los productos y servicios de las empresas ubicadas en la provincia de Napo, además de ofrecer un espacio de aprendizaje sobre los inicios para la migración electrónica con herramientas de uso común que están al alcance de las empresas y los consumidores.

La arquitectura de la plataforma web se basó en el esquema cliente-servidor donde la aplicación del lado del cliente se desarrolló en el marco de trabajo angular, mientras que la aplicación del lado del servidor en el marco de trabajo laravel por la facilidad que ofrecen al simplificar la elaboración de las aplicaciones, como resultado final la plataforma fue implementada después de haber superado las pruebas de estrés, funcionamiento y usabilidad en un hosting compartido en internet a disponibilidad del público en general.

Palabras clave: Object oriented hypermedia design, angular, laravel, lenguaje unificado de modelado, servicio web, comercio electrónico.

ABSTRACT

The mandatory confinement in the country generated by the global pandemic due to Covid-19 intensified the use of technology in different areas such as education, the work environment and electronic commerce, the need to the activities of daily life through the use of these means has generated dependence on them in order to survive, this is the case of micro, small and medium-sized companies that, despite the fact that most of them have internet connectivity, have been affected in their sales by not have sufficient technological resources to be able to remain active in the market in which they operate and to deliver their products or services to their customers.

This project relied on the phases of the Object Oriented Hypermedia Design methodology for its documentation and design through the use of the unified modeling language (UML), in addition it was based on the use of free technological tools that allowed to accelerate the development process of the web platform to help promote the products and services of companies located in the province of Napo, in addition to offering a learning space on the beginnings of electronic migration with commonly used tools that are available to companies and the consumers.

The architecture of the web platform was based on the client-server scheme where the client-side application was developed in the angular framework, while the server-side application was developed in the laravel framework for the ease they offer to the simplify the development of the applications, as a final result the platform was implemented after having passed the stress, performance and usability tests in a shared hosting on the internet available to the general public.

Keywords: Object-oriented hypermedia design, angular, laravel, unified modeling language, web service, e-commerce.

CAPÍTULO I

1. MARCO TEÓRICO

1.1. Tema de investigación

IMPLEMENTACIÓN DE PLATAFORMA WEB MEDIANTE LA APLICACIÓN DE LA METODOLOGÍA OBJECT ORIENTED HYPERMEDIA DESIGN METHOD (OOHDM), PARA APOYO A MICRO, PEQUEÑAS Y MEDIANAS EMPRESAS (MIPYMES) AFECTADAS EN SUS VENTAS ANTE LA EMERGENCIA SANITARIA EN LA PROVINCIA DE NAPO.

1.2. Antecedentes investigativos

1.2.1. Contextualización del problema

En América latina y el Caribe desde hace siete años se ha visto lentitud en el crecimiento económico con incrementos en la tasa de pobreza extrema y desempleo en los años 2002 a 2019 como se muestra en el estudio realizado por la Comisión Económica de América Latina y el Caribe (CEPAL) [1], a partir de ello la proyección de los efectos económicos negativos provocados por la pandemia a corto plazo muestra un aumento de desempleo y reducción de salarios e ingresos, mientras que a mediano y largo plazo se estima el cierre de empresas y menor crecimiento económico [1].

En la región con datos del año 2018 las empresas presentaban un alto grado de conectividad a Internet, pero muy pocas aprovechaban la plataforma para el despliegue de canales de venta digital a pesar de poseer sitios web propios [2], ante esta situación con el paso del tiempo la necesidad de mantener la economía activa y generar ingresos ha orillado a las micro, pequeñas y medianas empresas a aprovechar las oportunidades de la digitalización de la producción mediante la inversión de canales tecnológicos para la implementación de plataformas de soporte de negocios, de ventas y

aplicaciones móviles [3] frente al auge del crecimiento del desarrollo tecnológico y comercio electrónico en los países latinoamericanos.

En Ecuador el impulso del uso de las tecnologías de la información y la comunicación (TIC) e internet han contribuido a la innovación y la promoción de la economía digital para el crecimiento económico y generación de empleo [4]. Actualmente el empleo de sistemas tecnológicos favorece a la creación de emprendimientos que aportan al cambio de la matriz productiva y mejoran la calidad en la prestación de servicios en diferentes áreas. La implementación del comercio electrónico por parte de las empresas a generado un impacto importante en el incremento de ventas, aumento de oportunidades de negocio, producción y oportunidades de empleo [5]. A pesar de la emergencia sanitaria en el país la intensificación del uso de estos medios puede ofrecer a las micro, pequeñas y medianas empresas (MiPymes) una ayuda para mantener su economía activa.

En la provincia de Napo las MiPymes son un factor sustancial en la generación de empleo y la promoción del consumo de productos propios de la región oriental del país. Alrededor del 90% son microempresas y su participación en ventas es apenas del 0,1% con respecto a las demás provincias [6], un valor extremadamente bajo que antecede al problema actual por el que atraviesan estas entidades en ingresos económicos derivado de las medidas de confinamiento obligatorio generadas por la pandemia, la causa radica en que aún no se han adaptado al uso de medios digitales por la escasez de recursos tecnológicos lo cual hace difícil afrontar la situación.

En las MiPymes y pequeños negocios con ausencia de medios tecnológicos para el despliegue de canales digitales de venta se ven afectadas económicamente por el estado de emergencia sanitaria y denotan en problemas como:

- Acumulación de deudas y posible cierre del negocio.
- Recorte de personal que aumenta más la tasa de desempleo en el país.
- Disminución de la producción por elevación de costes de importación de materia prima.
- Reducción de ventas en el mercado en el que se desenvuelven.

1.2.2. Delimitación

Línea de investigación: Desarrollo de software.

Sublínea de investigación: Aplicaciones web.

Delimitación espacial: El proyecto está delimitado para el territorio comprendido por la provincia de Napo y destinado a las MiPymes del sector.

Delimitación temporal: La investigación se realizará durante el periodo académico octubre 2020 – febrero 2021.

1.2.3. Justificación

La importancia de tener una herramienta web que permite cumplir con objetivos de mejora en promoción y ventas, se basa en la aplicación correcta de una metodología que abarque los procesos de diseño, desarrollo e implementación de aplicaciones multimedia como es el caso de Object Oriented Hypermedia Design Method (OOHDM). Metodología que se investigó en la realización de la plataforma web de ayuda a las MiPymes que aún no han dado el salto tecnológico necesario para ser conocidas en el mercado en el que se desenvuelven.

Con la implementación de la plataforma web gratuita usable y funcional para agrupar, clasificar y guiar a las MiPymes al comercio electrónico, se podrá ofrecer una alternativa para reactivar la economía en diferentes sectores de la provincia y salvaguardar la integridad física de los empleados y consumidores. Los beneficiarios de la implementación de la plataforma web son:

- MiPymes de la provincia de Napo quienes podrán darse a conocer en internet y obtener una guía sobre la migración al comercio electrónico para mantener su ingreso económico, darle continuidad a su negocio y preservar empleos.

- Los consumidores quienes podrán adquirir productos de primera necesidad de los establecimientos cercanos y de esta manera cumplir con las medidas de distanciamiento social.
- La provincia de Napo, con la reactivación de la economía poder afrontar la crisis generada por la pandemia y aportar al movimiento económico del país.

1.2.4. Fundamentación teórica

A través de la investigación bibliográfica realizada en los repositorios de las universidades del Ecuador se han encontrado trabajos relacionados que son de utilidad en el presente trabajo de investigación:

Según Sandra Isabel Maji Llinin en su tesis “El comercio electrónico y el crecimiento económico de las empresas del sector textil del cantón Ambato” [7], trabajo realizado como tesis de la Universidad Técnica de Ambato en el año 2020, donde la finalidad es determinar si el comercio electrónico influye en el crecimiento económico en la industria del sector textil, concluye que las empresas que hacen uso de nuevas tecnologías se les facilita la expansión de su negocio en el mercado en el que se desenvuelven, permite llegar a satisfacer las necesidades de sus clientes, optimizar el tiempo, recursos y la calidad de la producción para así aumentar sus ingresos, lo que implica que el comercio electrónico influye en su crecimiento económico [7].

Según Jimmy Rolando Molina Ríos, Mariuxi Paola Zea Ordóñez, María José Contento Segarra y Fabricio Gustavo García Zerda en su publicación “Comparación de metodologías en aplicaciones web” [8], trabajo realizado como artículo de investigación en el Grupo de Investigación de la carrera de Ingeniería en Sistemas de la Universidad Técnica de Machala publicada en la Revista 3C Tecnología en el año 2018, donde se destaca las características de la metodología OOHDM frente a las variantes que existen en el área de metodologías para aplicaciones web, lo que determinó que las metodologías que contemplan la mayoría de los requerimientos de las aplicaciones web son OOHDM y UWE, sin embargo, la metodología OOHDM es la más utilizada por la facilidad de adaptarse en el desarrollo de cualquier proyecto al

ser un marco de trabajo que agiliza el proceso de construcción en cada etapa del ciclo de vida [8].

Según Alex Napoleón Espinoza Sanchez en su tesis “Desarrollo de una plataforma online para promoción de servicios personales y profesionales” [9], trabajo realizado como tesis de la Pontificia Universidad Católica del Ecuador en el año 2019 donde uno de sus puntos de investigación se basa en la aplicación de las etapas de la metodología OOHDM para el desarrollo de la plataforma online, determinó que la documentación del proceso de desarrollo e implementación de su plataforma fue simplificada gracias a la aplicación de cada etapa que la metodología OOHDM proporciona [9].

Los términos utilizados en el proyecto de investigación en los cuales se basa para su elaboración son:

Ingeniería de software

Es una disciplina de la ingeniería que se enfoca en el proceso de la producción de software y que abarca actividades como la especificación, desarrollo, validación y evolución del mismo [10]. Estas actividades son comunes en los procesos de software cuyo desempeño se explica:

- Especificación del software, define el software a producir y las restricciones de operación
- Desarrollo del software, se realiza el diseño y programación del software
- Validación del software, se verifica el funcionamiento del software y que cumpla con los requerimientos del cliente
- Evolución del software, se modifica el software para que refleje nuevos requerimientos del cliente y del mercado [10].

La ingeniería es la encargada de obtener buenos resultados en términos de calidad del software a partir de la aplicación de herramientas, métodos y teorías que apoyan a la producción con un tiempo y presupuesto determinado [10].

Aplicaciones web

También llamadas “webapps”, son una de varias categorías distintas de software centrado en redes que asocia una extensa gama de aplicaciones. Las webapps son un conjunto de archivos de hipertexto vinculados que presentan información con uso de texto y gráficas limitadas. Sin embargo, desde que surgió Web 2.0, las webapps han evolucionado hacia ambientes de cómputo sofisticados que no sólo proveen características aisladas, funciones de cómputo y contenido para el usuario final, sino que también están integradas con bases de datos corporativas y aplicaciones de negocios [11].

Naturaleza única de las webapps. Las webapps brindan capacidad de cómputo junto con contenido de información y presentan los siguientes atributos [11]:

- Uso intensivo de redes, una webapp se aloja en la red y permite el acceso a los clientes de una intranet o comunidades mundiales para atender sus necesidades;
- Concurrencia, puede soportar el acceso de un gran número de usuarios al mismo tiempo;
- Carga impredecible, la cantidad de usuarios varía de un día a otro;
- Rendimiento, se mide por el tiempo de espera que toma la carga, procesamiento y despliegue de la webapp de lado del cliente y del servidor;
- Disponibilidad, demanda de acceso a la webapp las 24 horas de los 365 días del año, aunque en ocasiones pueden no estar en línea por cuestiones de mantenimiento;
- Orientadas a los datos, uso de hipermedios para presentar contenido de texto, gráficos, audio y video. Además, las webapps comúnmente acceden a información que reside en bases de datos;
- Contenido sensible, la calidad de una webapp se mide por la calidad de su contenido;
- Evolución continua, el contenido y funcionamiento de la webapp evoluciona en etapas cortas;
- Inmediatez, plazos cortos para el desarrollo y lanzamiento en el mercado;

- Seguridad, el objetivo es proteger contenido sensible, transmitir de manera segura los datos, autenticar y restringir acciones a los usuarios;
- Estética, es la percepción de los usuarios de la apariencia de la webapp, un buen diseño de interfaz tiene que ver con el éxito para su comercialización [11].

Arquitectura cliente - servidor

Las aplicaciones web se basan en la arquitectura básica cliente – servidor el cual es un modelo de aplicación distribuida que se divide en los proveedores de servicios (servidor) y los demandantes (cliente), este esquema por lo general se usa en internet, intranet y extranet corporativas en el caso de servidores web [12].

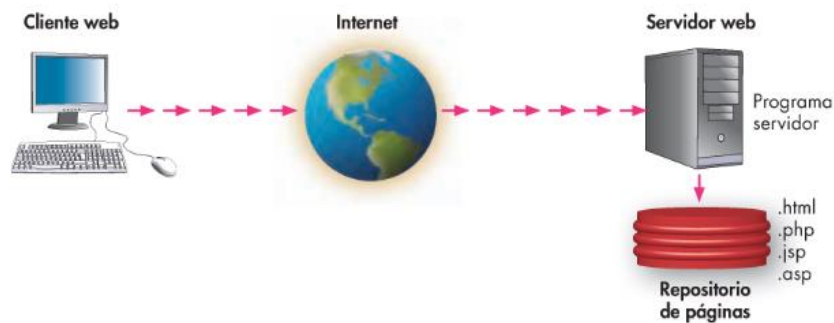


Figura 1.1 Esquema básico del servicio web
Fuente: Aplicaciones Web, 2013 [12]

Tipo de aplicaciones web

- Estáticas, la información se limita a los usuarios sin poder interactuar con ella [13].
- Dinámicas, establece una comunicación activa entre el cliente y la aplicación mediante la actualización del contenido en base a las acciones del usuario [13].
- Comercio electrónico, se enfocan en la venta de productos y la inclusión de métodos de pago [14].
- Portales, permiten el acceso a contenido clasificado por categorías que incluyen chats, foros, correo electrónico, buscadores, entre otros [14].
- Sistemas gestores de contenido, un CMS (Content Management Systems) es un software que facilita la gestión de sitios web de grandes dimensiones en

Internet o una Intranet corporativa, el contenido es actualizado continuamente de forma dinámica [15].

Desarrollo de aplicaciones web

El desafío del desarrollo de aplicaciones surgió con la creación del internet y la posibilidad de acceder a la información desde cualquier dispositivo, los avances tecnológicos permitieron crear aplicaciones rápidas, ligeras y robustas para sacar provecho a la red. Ante la presencia de estas nuevas tecnologías, en la actualidad el reto es escoger correctamente herramientas que faciliten el desarrollo, aplicación de arquitecturas y el uso de lenguajes de programación [13].

Metodologías para la Ingeniería Web

La ingeniería web se vincula con metodologías orientadas al desarrollo y evolución de aplicaciones web con un alto grado de eficiencia y calidad, estas metodologías implican técnicas que abarcan todos los procesos para el desarrollo del software en su mayoría [16].

La Ingeniería Web Dirigida por Modelos (MDWE) es el empleo de la arquitectura dirigida por modelos al desarrollo de aplicaciones web que resulta útil debido a la evolución continua de las tecnologías y plataformas web [16].

Existen propuestas de modelado como:

- Metodologías orientadas al diseño navegacional, cuyo objetivo es crear aplicaciones hipermedia en sistemas estáticos basadas en el modelo relacional clásico. Algunos ejemplos son OOHDM, WebML, ADM, AutoWeb y RMM;
- El grupo que se basa en la idea de ampliar los métodos de desarrollo orientados a aplicaciones dinámicas e introducir la semántica de la hipermedia como característica propia de este nuevo tipo de sistemas de software. En este grupo se encuentra los métodos UWE, WSDM, EORM, OOW y OO-Method [16].

Framework para aplicaciones web

Es un conjunto de archivos y directorios que facilitan la creación de aplicaciones [17] y se utiliza en diversos ámbitos del desarrollo de sistemas [18]. Un marco de trabajo concentra funcionalidades ya desarrolladas y probadas, implementadas en un determinado lenguaje de programación [17].

Un marco de trabajo web tiene como objetivo acelerar el desarrollo de sistemas web, con reutilización de código, uso de patrones de diseño y componentes que faciliten implementar funcionalidades [17].

Ventajas de utilizar un marco de trabajo.

Un marco de trabajo ofrece ventajas importantes en el desarrollo de aplicaciones como [17]:

- **Uso de patrones de diseño**, el patrón de diseño más utilizado en los marcos de trabajo es el Modelo-Vista-Controlador (MVC) y organiza la aplicación en tres modelos separados:
 - **Modelo**: Representa los datos de la aplicación y las reglas de negocio.
 - **Vista**: Representa la información de entrada y salida presentada al usuario.
 - **Controlador**: Procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema.

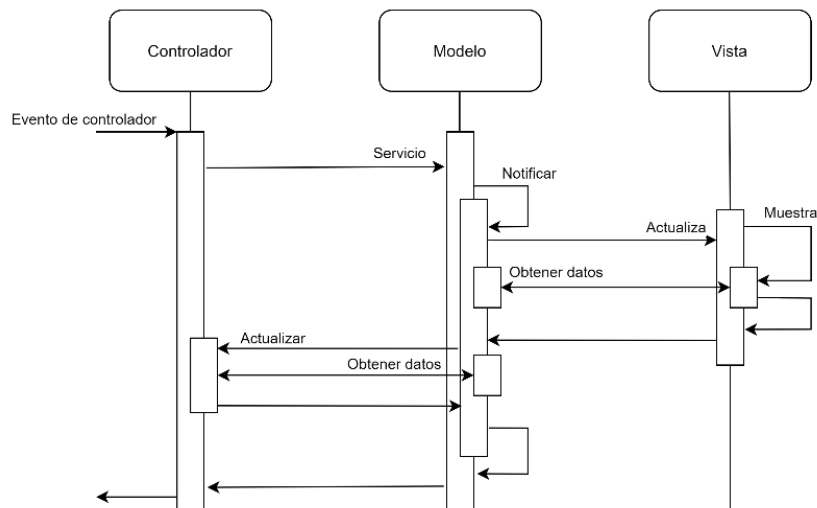


Figura 1.2 Patrón de diseño MVC
Fuente: ¿Qué es un Framework Web? [18]

- **Estructura predefinida de la aplicación**, un marco de trabajo proporciona una estructural global de la aplicación por lo que mantiene los archivos perfectamente organizados.
- **Código altamente testado**, garantiza el buen funcionamiento debido a que el código es probado antes de ser lanzado.
- **Soporte de la comunidad**, detrás de cada marco de trabajo existe una gran comunidad de usuarios que ayudan a la creación de nuevas funcionalidades.
- **Facilita el trabajo en equipo**.

Desventajas de utilizar un framework

El uso de un marco de trabajo puede traer ciertas limitaciones como [17]:

- **Tiempo de aprendizaje**, aprender cómo funciona un marco de trabajo puede llevar cierto tiempo.
- **Exceso de líneas de código**, en ocasiones aplicaciones desarrolladas en un marco de trabajo puede incluir demasiadas líneas de código, más de las que son necesarias, según expertos.
- **Limitaciones**, existen aspectos de un marco de trabajo que no pueden ser modificados, es mejor buscar uno que se adapte a las necesidades.

Lenguaje UML

Fue desarrollado en 1990 por la compañía Rational Software Corporation, pero fue en enero de 1997 que Object Management Group (OMG) un consorcio dedicado a la creación y cuidado de estándares de tecnologías orientadas a objetos publicó la versión 1.0 del Lenguaje Unificado de Modelado (UML) diseñado para modelar sistemas de software, desde entonces ha evolucionado y publicado nuevas versiones que incluyen nuevas mejoras [19].

UML está diseñado para detallar, representar, alterar, construir y documentar un sistema. Permite una representación de artefactos, acciones, actores, lógicas de negocio y esquemas de bases de datos [19].

Base de datos

Las bases de datos surgieron con la finalidad de computarizar el manejo de archivadores manuales, por lo que se definen como una colección de datos almacenados de forma organizada mediante una estructura de datos y están diseñadas para acceder, gestionar y actualizar la información de una empresa u otro tipo de organización [20].

Características de las bases de datos [21]:

- **Independencia de datos**, cualquier aplicación puede hacer uso de los datos.
- **Reducción de la redundancia**, evita inconsistencia de datos mediante la reducción de la duplicidad de datos para aprovechar el espacio de almacenamiento.
- **Seguridad**, es el resguardo de la base de datos ante usuarios no autorizados.

MySQL

Es un sistema gestor de base de datos dedicado a la administración de bases de datos relacionales, es conocido por ser rápido, sólido y flexible. Es óptimo para crear bases

de datos con acceso a aplicaciones web, sistemas de transacción en línea o alguna solución profesional que involucre almacenar datos [22].

Ventajas de MySQL con respecto a otros gestores de bases de datos [22]:

- Licencia publica, permite utilizar el programa, consultar y modificar su código fuente.
- Facilidad de integración, está desarrollada en C y C++ por lo que facilita la integración de aplicaciones que usan los mismos lenguajes.
- Descarga gratuita.
- Sistema cliente/servidor, permite trabajar como servidor multiusuario, el servidor crea subprocesos para administrar las solicitudes del cliente.
- Lenguaje SQL, utiliza el lenguaje de consulta estandarizado para bases de datos relacionales.
- Portable, tiene presencia en diferentes plataformas con distribuciones como; Microsoft Windows, Linux, Mac X y UNIX.

Servicio web

El consorcio WWW (W3C) define a un servicio web como un conjunto de procedimientos que a través de una red permiten la interacción entre diferentes equipos de cómputo, los servicios están definidos en formatos que son admitidos e interpretados por la mayoría de las aplicaciones que interactúan con sus funciones [23].

API-REST

Un interfaz de programación de aplicaciones (API) es una agrupación de funciones y procedimientos que se ejecutan con el fin de interactuar con otras aplicaciones. Para que cualquier cliente pueda acceder a ella se requiere el uso de protocolos estándar para el intercambio de datos con el servidor [24].

Roy Fielding propuso a Representational State Transfer (REST) como un enfoque arquitectónico utilizado para el desarrollo de servicios web y para construir sistemas

distribuidos basados en hipermedia. Es independiente de cualquier protocolo y no está precisamente emparentado a HTTP. Sin embargo, las implementaciones REST más habituales emplean HTTP como protocolo de aplicación, la ventaja de REST sobre HTTP es que cualquier aplicación cliente puede consumir los servicios sin importar el lenguaje de programación que utilicen [24].

Métodos de petición HTTP [24]:

- **GET**, recupera la respuesta del recurso solicitado.
- **POST**, crea un nuevo recurso.
- **PUT**, crea o modifica un recurso específico.
- **PATCH**, modifica un recurso de forma parcial.
- **DELETE**, elimina un recurso específico.

JWT

JSON Web Token es un formato compacto de representación de reclamos que se transfieren entre dos partes, destinado a entornos con limitaciones de espacio, como encabezados de autorización HTTP y parámetros de consulta URI. JWT representa las peticiones que se transmiten como un objeto JSON que se codifican como JSON Web Signature o como texto sin formato de una estructura JSON Web Encryption, lo que asegura que estén firmados digitalmente o que su integridad está protegida con un código de autenticación de mensaje MAC y/o cifrado. JWT siempre se representa mediante la serialización JWS Compact o la serialización JWE Compact [25].

JSON

JavaScript Object Notation – Notación de Objetos de JavaScript, es un formato ligero ideal para el intercambio de datos. Es fácil de leer y escribir para las personas y fácil de interpretar y generar para las máquinas. JSON es un formato de texto que es independiente del lenguaje, pero emplea convenciones que son conocidos por los programadores de la familia de lenguajes C, Java, JavaScript, Python, Perl, entre otros [26].

ReactiveX

Es una librería utilizada para la creación de programas asíncronos y basados en eventos a través del manejo de secuencias observables. Provee una colección de operadores para poder filtrar, seleccionar, transformar, combinar y componer observables, lo que permite el manejo de errores asíncronos, abstraer problemas de subprocesos, sincronización y concurrencia de bajo nivel [27].

Evaluación heurística

Es un método de inspección de usabilidad realizado por expertos que analizan y revisan si cada elemento de la interfaz de usuario concuerda con principios de usabilidad válidos [28].

Para desarrollar una evaluación heurística es necesario conocer las tareas a realizar y el perfil de los usuarios que van a utilizar el sistema. Mediante la elaboración de un cuestionario que persigue los objetivos de la evaluación, se establecen criterios de acuerdo al contexto de trabajo y se cuantifican en forma de severidad y prioridad, además, es necesario definir una escala de frecuencia para ajustar los valores obtenidos en la evaluación [28].

El cuestionario se debe aplicar a tres o cinco expertos quienes evalúan y validan cada área que contemplan los principios heurísticos, normalmente se usan los propuestos por Jakob Nielsen y Rolf Molich [28].

Heurísticas de Jakob Nielsen y Rolf Molich

El ingeniero en interfaces de usuario y ciencias de la computación, Jakob Nielsen, propone diez principios básicos que todo sitio web debería cumplir [28]:

- 1. Visibilidad del estado del sistema**, el sistema debe comunicar a los usuarios el estado del sistema, mediante una retroalimentación adecuada en un tiempo razonable.

2. **Relación entre el sistema y el usuario**, el sistema debe hablar el mismo lenguaje del usuario, usar términos conocidos y legibles.
3. **Control y libertad para el usuario**, el usuario debe contar con opciones para deshacer y rehacer cambios en el sistema.
4. **Consistencia y estándares**, el sistema debe asegurarse que el usuario siga las normas y convenciones sobre las que esta implementada.
5. **Prevención de errores**, es importante proporcionar asistencia y mantener informado al usuario de lo que sucede en el sistema para prevenir que caiga en un error.
6. **Reconocer antes que recordar**, el sistema debe hacer visible las acciones para reducir la información que el usuario debe recordar.
7. **Flexibilidad y eficiencia de uso**, se debe asegurar que la capacidad del sistema sea aprovechada por usuarios expertos y que sea útil para usuarios básicos al proporcionar métodos abreviados o atajos de teclado.
8. **Diálogos estéticos y diseño minimalista**, el sistema debe evitar contener información innecesaria porque reduce la visibilidad de la información relevante.
9. **Reconocimiento, diagnóstico y recuperación de errores**, los mensajes de error deben indicar el problema de forma clara y sugerir soluciones constructivas.
10. **Ayuda y documentación**, el caso de que el usuario requiera ayuda el sistema debe proporcionar documentación fácil de localizar, que muestre brevemente su funcionalidad.

Directorio de empresas en línea

Es un portal web que agrupa y lista a empresas por categorías según la actividad que desempeñan, también es conocido como una herramienta de marketing. La mayoría de estos portales ofrecen información básica de las empresas y los servicios que brindan, y de esta manera ayudarlas con el posicionamiento en internet, conocer a su competencia y contactar con clientes potenciales [29].

Canales de venta en línea

Resultan ser muy económicos y ofrecen grandes ventajas con respecto a los canales de venta “offline”, permiten llegar fácilmente al consumidor con información de los productos o servicios que ofrece un negocio. Algunos ejemplos de canales de venta en línea son las tiendas virtuales, redes sociales, buscadores como canal, Marketplace y programas de afiliados [30].

MiPymes

El termino MiPymes se usa para mencionar a las micro, pequeñas y medianas empresas cuya índole se determina por su orden jurídico o no jurídico, volumen de ventas, capital social, número de empleados y su nivel de activos [31]. En Ecuador el 99,52% de las empresas son MiPymes [6].

Clasificación por su tamaño [6]:

- Micro empresa, sus ventas anuales son inferiores o iguales a \$100.000, con personal de 1 a 9.
- Pequeña empresa, sus ventas anuales van de \$100.001 a \$1'000.000, con personal de 10 a 49.
- Mediana empresa, sus ventas anuales van de \$1'000.001,00 a \$5'000.000, con personal de 50 a 199.

1.3. Objetivos

1.3.1. General

Implementar una Plataforma Web mediante la aplicación de la metodología OOHDM en el desarrollo, enfocado en las Micro, Pequeñas y Medianas Empresas (MiPymes) afectadas en sus ventas ante la emergencia sanitaria en la Provincia de Napo.

1.3.2. Específicos

- Estudiar las características del estado actual de las MiPymes de la Provincia de Napo que permita la identificación de los requerimientos del sistema.
- Aplicar la metodología OOHDM con UML en el desarrollo de la Plataforma Web.
- Desplegar la Plataforma Web en un servicio de alojamiento web superada las pruebas.

CAPÍTULO II

2. METODOLOGÍA

2.1. Materiales

El material para la recolección de información se apoya en una encuesta dirigida a los propietarios de negocios y empresas con el objetivo de estudiar las características de la situación actual de las MiPymes e identificar los requerimientos del sistema.

Universidad Técnica de Ambato

Facultad de Ingeniería en Sistemas, Electrónica e Industrial

Esta encuesta está dirigida a las MiPymes (Micro, Pequeña y Mediana empresa) y pequeños negocios de la provincia de Napo con la finalidad recolectar información sobre su situación actual y a la vez datos que permitan identificar los requerimientos para implementar una plataforma web gratuita que agrupe a las empresas del sector para promocionar sus servicios y guiarlas en la migración al comercio electrónico.

PREGUNTA 1. ¿Está familiarizado con alguno de estos medios para promocionar sus servicios? Facebook Marketplace, Google Mi Negocio, Mercado Libre, Sitio web, WhatsApp Business, OLX.

- Si
- No

PREGUNTA 2. Si ha usado algún medio digital para gestionar sus ventas, ¿cómo ha sido su experiencia?

- Bueno
- Malo

- Regular
- No he usado ninguno

PREGUNTA 3. Seleccione los medios que le resultaría interesante aprender a usar para promocionar su empresa o negocio. (seleccione uno o más)

- Facebook Marketplace
- Google Mi Negocio
- Mercado Libre
- Sitio web
- WhatsApp Business
- OLX

PREGUNTA 4. ¿Cree usted que orientar a las MiPymes al uso de medios digitales ayudara a mejorar las ventas?

- Si
- No
- Tal vez

PREGUNTA 5. ¿Cree usted que se podrá llegar fácilmente al consumidor con la integración de nuevas tecnologías?

- Si
- No
- Tal vez

PREGUNTA 6. ¿Está usted de acuerdo que la agrupación de empresas y pequeños negocios en una plataforma web facilitara a los consumidores encontrar información útil sobre los servicios que ofrecen?

- Si
- No

- Tal vez

PREGUNTA 7. ¿Estaría usted de acuerdo en que se utilice el nombre y la ubicación de su empresa o negocio en una plataforma web gratuita para promocionar sus servicios?

- Si
- No
- Tal vez

PREGUNTA 8. ¿Cree usted que la idea de promocionar a las empresas y negocios de la provincia ayudara al movimiento económico local?

- Si
- No
- Tal vez

2.2. Métodos

2.2.1. Modalidad de la investigación

- **Investigación de campo:** El investigador realizó la recolección de información sobre la situación actual de las MiPymes mediante encuestas directamente de sus propietarios o encargados de forma que la información fuese verídica para el análisis de datos.
- **Investigación bibliográfica:** El proyecto se basó en la revisión del material existente relacionado al tema de investigación tomando como fuente artículos científicos, libros, documentos técnicos, tesis realizadas en universidades, revistas y estatutos legales.
- **Investigación de modalidad especial:** El desarrollo de la plataforma web para apoyo a las MiPymes responden a factores socioeconómicos en la provincia por lo cual la investigación también es de modalidad especial.

2.2.2. Población y muestra

- **Población:** Para el presente trabajo de investigación la población fue definida por las MiPymes de la provincia de Napo con un total de 7062 de acuerdo a los datos del 2019 presentados por el Instituto Nacional de Estadísticas y Censos (INEC) [6].
- **Muestra:** Debido a la naturaleza del problema y a la gran cantidad de MiPymes registradas en la zona, se aplicó el método de muestreo no probabilístico basado en el criterio del investigador en la selección del público objetivo considerado dentro del propósito del estudio, esto con la intención de evitar resultados sesgados, por lo tanto, se tomó un grupo de 20 empresas de distintos sectores de la provincia de Napo como muestra.

2.2.3. Recolección de información

Resultados obtenidos una vez aplicada la encuesta:

Pregunta 1. ¿Está familiarizado con alguno de estos medios para promocionar sus servicios? Facebook Marketplace, Google Mi Negocio, Mercado Libre, Sitio web, WhatsApp Business, OLX.

Opción	Respuestas	Porcentaje
Si	9	45%
No	11	55%
Total	20	100%

*Tabla 2.1 Resultados de encuesta - Pregunta 1
Fuente: Investigador*

20 respuestas

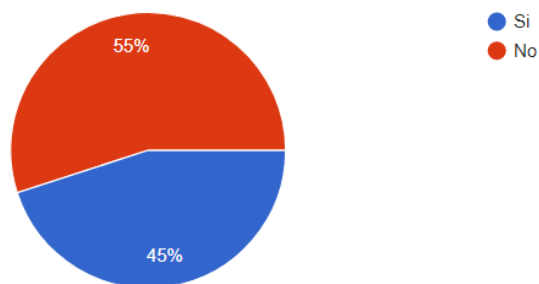


Figura 2.1 Resultados de encuesta - Pregunta 1
Fuente: Investigador

Análisis: El 55% de las personas encuestadas mencionan que no están familiarizadas con ninguna herramienta de la lista, mientras que el 45% manifiesta que sí.

Interpretación de resultados: En lo que respecta a la promoción de servicios más de la mitad de los encuestados no están familiarizados con los medios expuestos, lo que demuestra que la mayoría aun no da la transición tecnológica para impulsar su negocio con el uso de las herramientas gratuitas más conocidas.

Pregunta 2. Si ha usado algún medio digital para gestionar sus ventas, ¿cómo ha sido su experiencia?

Opción	Respuestas	Porcentaje
Bueno	3	15%
Malo	1	5%
Regular	5	25%
No he usado ninguno	11	55%
Total	20	100%

Tabla 2.2 Resultados de encuesta - Pregunta 2
Fuente: Investigador

20 respuestas

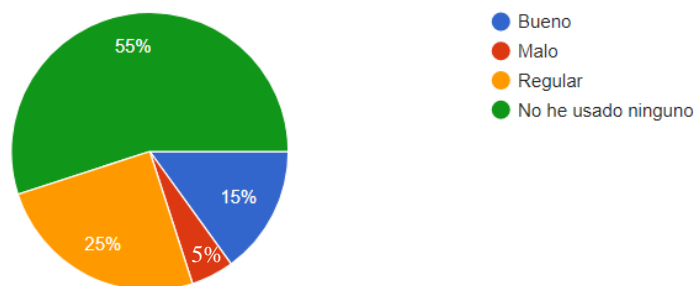


Figura 2.2 Resultados de encuesta - Pregunta 2
Fuente: Investigador

Análisis: El 55% de las personas encuestadas mencionan que no han usado algún medio digital para gestionar sus ventas, mientras quienes sí han tenido experiencia de uso manifiestan su grado de satisfacción que corresponde al 25% regular, 15% bueno y 5% malo.

Interpretación de resultados: Se mantiene el porcentaje de quienes no han usado ninguna herramienta con respecto a la pregunta anterior y muy pocas han tenido una buena experiencia de uso, lo que demuestra que estamos en presencia de usuarios con escaso conocimiento sobre el uso de aplicaciones y refleja la importancia del diseño de interfaces apropiadas.

Pregunta 3. Seleccione los medios que le resultaría interesante aprender a usar para promocionar su empresa o negocio. (seleccione uno o más)

Opción	Respuestas / 20	Porcentaje / 100%
Facebook Marketplace	19	95%
Google Mi Negocio	8	40%
Mercado Libre	8	40%
Sitio web	6	30%
WhatsApp Business	15	75%
OLX	2	10%

Tabla 2.3 Resultados de encuesta - Pregunta 3
Fuente: Investigador

20 respuestas

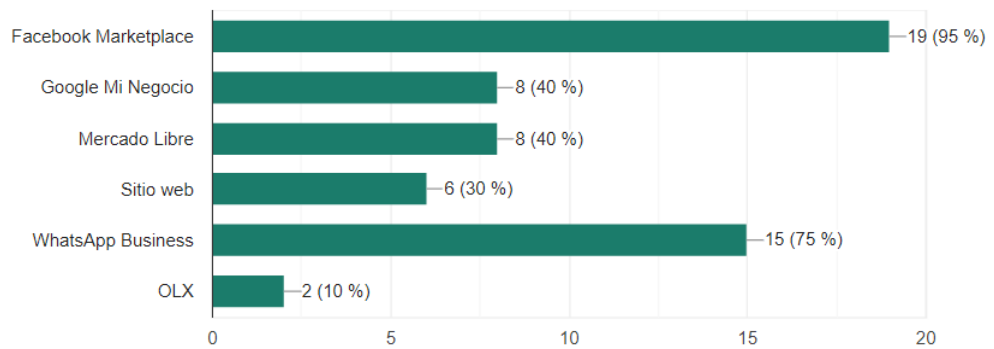


Figura 2.3 Resultados de encuesta - Pregunta 3
Fuente: Investigador

Análisis: Al 95% de las personas encuestadas les interesa aprender el uso de Facebook Marketplace para promocionar su empresa, al 75% WhatsApp Business, al 40% Google Mi Negocio y Mercado Libre, al 30% Sitio Web y al 10% OLX.

Interpretación de resultados: Es importante conocer los medios que son de interés para los encuestados, la mayoría se centra en Facebook Marketplace y WhatsApp Business, aunque también se presta atención a otros medios como Google Mi Negocio y Mercado Libre, esto determina el contenido a incluir en las guías de aprendizaje.

Pregunta 4. ¿Cree usted que orientar a las MiPymes al uso de medios digitales ayudara a mejorar las ventas?

Opción	Respuestas	Porcentaje
Si	17	85%
No	0	0%
Tal vez	3	15%
Total	20	100%

Tabla 2.4 Resultados de encuesta - Pregunta 4
Fuente: Investigador

20 respuestas

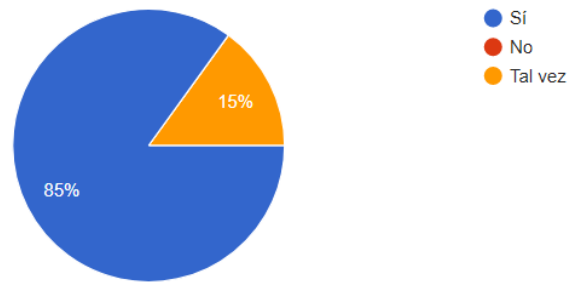


Figura 2.4 Resultados de encuesta - Pregunta 4
Fuente: Investigador

Análisis: El 85% de las personas encuestadas afirman que con la orientación del uso de medios digitales podrían mejorar sus ventas, mientras que el 15% menciona que tal vez podría ayudar y el 0% que no.

Interpretación de resultados: La mayoría está de acuerdo que recibir una orientación de uso de medios digitales puede ayudar a mejorar sus ventas, lo que demuestra la necesidad de establecer instrucciones de uso apropiados.

Pregunta 5. ¿Cree usted que se podrá llegar fácilmente al consumidor con la integración de nuevas tecnologías?

Opción	Respuestas	Porcentaje
Si	18	90%
No	0	0%
Tal vez	2	10%
Total	20	100%

Tabla 2.5 Resultados de encuesta - Pregunta 5
Fuente: Investigador

20 respuestas

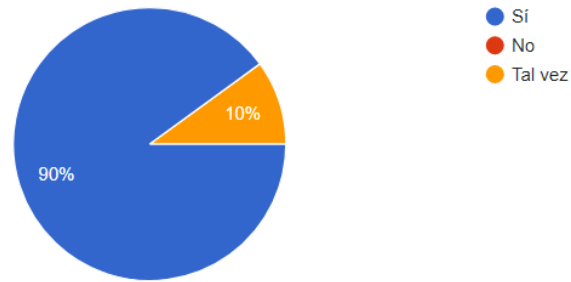


Figura 2.5 Resultados de encuesta - Pregunta 5
Fuente: Investigador

Análisis: El 90% de los encuestados menciona que se podrá llegar fácilmente al consumidor con la integración de nuevas tecnologías, mientras que el 10% manifiesta que tal vez sería posible y el 0% que no.

Interpretación de resultados: Demuestra que la inclusión de nuevas tecnologías puede despertar el interés del consumidor sobre la empresa y los servicios que ofrece, como lo indican la mayoría de los encuestados.

Pregunta 6. ¿Está usted de acuerdo que la agrupación de empresas y pequeños negocios en una plataforma web facilitara a los consumidores encontrar información útil sobre los servicios que ofrecen?

Opción	Respuestas	Porcentaje
Si	18	90%
No	0	0%
Tal vez	2	10%
Total	20	100%

Tabla 2.6 Resultados de encuesta - Pregunta 6
Fuente: Investigador

20 respuestas

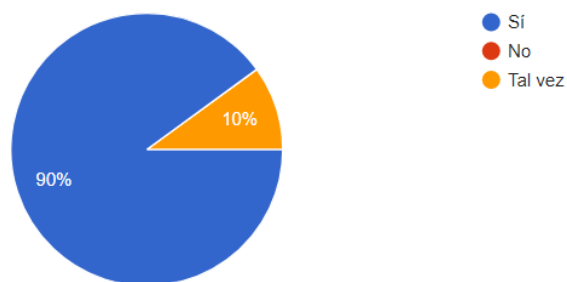


Figura 2.6 Resultados de encuesta - Pregunta 6
Fuente: Investigador

Análisis: El 90% de los encuestados menciona que la agrupación de las empresas y negocios en la Plataforma Web facilitará encontrar información útil a los consumidores, mientras que el 10% menciona que tal vez será útil y el 0% que no.

Interpretación de resultados: La mayoría está de acuerdo que agrupar a las empresas y pequeños negocios ayudará a difundir información útil para ofertar sus servicios a los consumidores, lo que determina la viabilidad de la creación de la plataforma web.

Pregunta 7. ¿Estaría usted de acuerdo en que se utilice el nombre y la ubicación de su empresa o negocio en una plataforma web gratuita para promocionar sus servicios?

Opción	Respuestas	Porcentaje
Si	17	85%
No	0	0%
Tal vez	3	15%
Total	20	100%

Tabla 2.7 Resultados de encuesta - Pregunta 7
Fuente: Investigador

20 respuestas

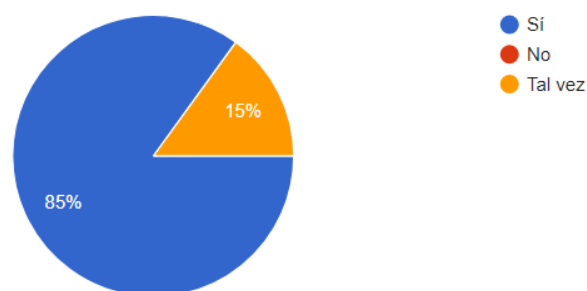


Figura 2.7 Resultados de encuesta - Pregunta 7
Fuente: Investigador

Análisis: El 85% de las personas encuestadas están de acuerdo en compartir información de su empresa o negocio en la Plataforma Web gratuita para promocionar sus servicios, mientras que el 15% menciona que tal vez sería posible y el 0% que no.

Interpretación de resultados: La mayoría está de acuerdo que se use el nombre y ubicación de la empresa o negocio para difundir información de sus productos y servicios en internet de manera gratuita.

Pregunta 8. ¿Cree usted que la idea de promocionar a las empresas y negocios de la provincia ayudará al movimiento económico local?

Opción	Respuestas	Porcentaje
Si	16	80%
No	0	0%
Tal vez	4	20%
Total	20	100%

Tabla 2.8 Resultados de encuesta - Pregunta 8
Fuente: Investigador

20 respuestas

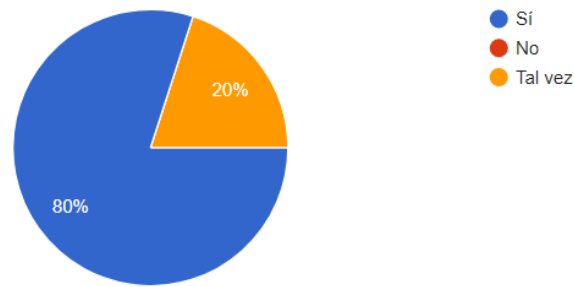


Figura 2.8 Resultados de encuesta - Pregunta 8
Fuente: Investigador

Análisis: El 80% de los encuestados afirman que promocionar a las empresas y negocios ayudará al movimiento económico local, mientras que el 20% manifiestan que tal vez sería posible y el 0% que no.

Interpretación de resultados: El sector económico de la Provincia de Napo se ha visto afectado ante la presente crisis sanitaria, por ello es importante conocer la opinión de los propietarios de las empresas y negocios del sector, quienes en su mayoría creen que la promoción de su establecimiento también ayudará al movimiento económico local. Gracias a ello se puede demostrar la viabilidad y factibilidad del presente trabajo de investigación.

2.2.4. Procesamiento y análisis de datos

Con los resultados obtenidos después de aplicar la encuesta a los propietarios de negocios y empresas se pudo determinar que:

- La mayoría de los encuestados indican que no están familiarizados con los medios más conocidos para promocionar sus servicios y gestionar sus ventas, por lo tanto, la oferta de establecer un espacio de guías sobre creación de canales de venta mediante el uso de herramientas gratuitas resulta ser de gran interés, donde Facebook Marketplace y WhatsApp Business destacan como los medios que más desean aprender.

- Con las guías de uso los propietarios conseguirán crear sus propios canales de venta por lo que podrían incrementar el alcance de su negocio en el mercado en el que se desenvuelven y mejorar sus ventas.
- Los encuestados están de acuerdo que compartir su información en la plataforma web facilitará la difusión de información acerca de su negocio para que los consumidores puedan conocer sus productos, generar ingresos y ayudar al movimiento económico local.

CAPITULO III

3. RESULTADOS Y DISCUSIÓN

3.1. Análisis y discusión de resultados

3.1.1. Object Oriented Hypermedia Design Method (OOHDM)

A partir de la información recabada sobre OOHDM dentro de las metodologías para la ingeniería web existentes y como parte del estudio propuesto en el presente proyecto, se realizó una investigación bibliográfica más detallada para la comprensión de su estructura y facilitar así su aplicación en el desarrollo de la plataforma web.

OOHDM es un modelo enfocado en el desarrollo de aplicaciones hipermedia, propuesto por Daniel Schwabe y Gustavo Rossi con el objetivo de simplificar y facilitar el diseño de las aplicaciones hipermedia, en la actualidad es ampliamente aceptada para el diseño de diferentes tipos de aplicaciones como: sitios web, sistemas de información, tiendas interactivas, presentaciones multimedia, entre otros [32].

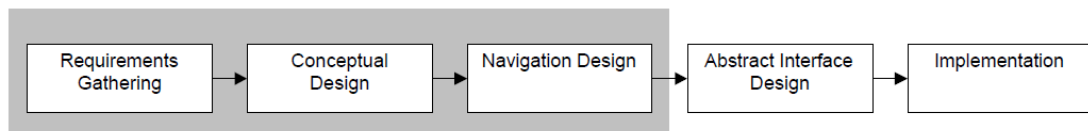
Ante la propuesta creciente de modelos de ciclo de vida involucrados en el desarrollo de sistemas como el modelo en espiral y cascada, OOHDM surgió como un proceso de construcción basado en hipermedia que no difiere en algunos aspectos de las utilizadas para desarrollar aplicaciones convencionales, está inspirado en la metodología de Diseño de Aplicaciones de Hipertexto (HDM) y mejora el proceso de concepción orientado a objetos [33], OOHDM está centrado en la estructura relacionada con el modelo de dominio cuyo propósito es proveer un marco de trabajo enfocado en la aplicación final, la navegación y el comportamiento funcional [32].

Bases de la metodología OOHDM [32]:

- Los objetos de navegación son vistas, en base de datos son objetos conceptuales.

- Uso de estrategias adecuadas para organizar el espacio de navegación con la introducción de contextos de navegación.
- Separa los problemas de la interfaz y de la navegación.
- La identificación explícita de decisiones de diseño solo deben tomarse en el momento de la implementación.

En un inicio la metodología estaba compuesta por 4 etapas como el diseño conceptual, diseño navegacional, diseño de interfaz abstracta e implementación, con el tiempo se le agregaría la recopilación de requisitos como primera fase del ciclo de vida [34].



*Figura 3.1 Fases en el diseño e implementación de aplicaciones web
Fuente: Modeling Interactions and Navigation in Web Applications [35]*

Fases de la metodología OOHDM

Las etapas de la metodología se combinan dentro del desarrollo incremental, iterativo y basado en prototipos, durante cada actividad a excepción de la última, se construyen un conjunto de modelos orientados a objetos que describen los requerimientos de diseño a partir de las iteraciones anteriores [36].

Etapa	Productos	Mecanismos	Preocupaciones de diseño
Recopilación de requisitos	Casos de uso, actores, escenarios	Identificación de roles y tareas, especificación de escenarios, especificación de diagramas de interacción de usuario	Identificación del dominio de la aplicación

Diseño conceptual	Clases, subsistemas, relaciones, atributos	Clasificación, composición, generalización y especialización	Modelado de la semántica del dominio de la aplicación
Diseño navegacional	Nodos, enlaces, estructuras de acceso, contextos de navegación, transformaciones de navegación	Mapeo entre objetos conceptuales y de navegación	Tiene en cuenta el perfil y la tarea del usuario. Énfasis en aspectos cognitivos.
Diseño de interfaz abstracta	Objetos abstractos de interfaz, respuesta a eventos externos, transformaciones de interfaz	Mapeo entre navegación y objetos perceptibles	Modelado de objetos perceptibles, implementando metáforas elegidas. Describir la interfaz para objetos de navegación.
Implementación	Aplicación de ejecución	Los proporcionados por el entorno de destino	Rendimiento, integridad

Tabla 3.1 Resumen de la metodología OOHDM
Fuente: Object Oriented Hypermedia Design Method [32], [35], [36]

- **Recopilación de requisitos**

El análisis y obtención de requerimientos es una fase importante en el desarrollo de todo software, la recopilación de información de los usuarios y las partes involucradas de forma adecuada hacen que los diseños sean exitosos, dentro de la metodología OOHDM la recopilación de requisitos se divide en las siguientes etapas [35]:

- **Identificación de roles y tareas**

Es esta etapa el experto interactúa con el dominio para identificar los roles, usuarios y tareas que son parte del sistema. Esta interacción se logra mediante el análisis de documentos y entrevistas a los usuarios [35].

- **Especificación de escenario**

En esta etapa cada usuario especifica de forma textual o verbal los escenarios que describen sus tareas identificadas en la etapa anterior. Los escenarios son descripciones narrativas de como la aplicación debe ser usada [35].

- **Especificación de caso de uso**

Un caso de uso detalla el proceder de un sistema mediante la descripción de un conjunto de secuencias de acciones que se ejecutan para un actor. Son empleados para capturar el comportamiento esperado del sistema en desarrollo y proveen un medio para que los desarrolladores, usuarios finales y los expertos del dominio tengan una comprensión común del sistema [37].

- **Especificación de diagrama de interacción de usuario**

Los diagramas de interacción de usuario (UID) representan la interacción entre el usuario y la aplicación, descrita textualmente en los casos de uso. En esta etapa se define un UID por cada caso de uso, por lo que este diagrama solo describe el intercambio de información entre el usuario y la aplicación sin considerar aspectos específicos de la interfaz, a diferencia de los diagramas de interacción de UML que consideran el intercambio de mensajes entre objetos del sistema y son más apropiados para la fase de diseño [35].

- **Validación de caso de uso**

En esta etapa, se validan los casos de uso y los UID a través de la interacción del experto con cada usuario al que le corresponde el rol y la tarea, con el fin de comprobar la aceptación del mismo [35].

• **Diseño conceptual**

Durante esta fase, se construye el esquema conceptual de la aplicación para representar los objetos, relaciones y colaboraciones existentes en el dominio, para esta representación se aplica los principios de modelado orientado a objetos, similar a la notación UML [32]. En OOHDM el esquema conceptual se elabora en base a clases, relaciones y subsistemas con el propósito de capturar el dominio semántico de la aplicación [36].

El diagrama de clases consiste en un conjunto de clases con atributos, conectadas entre sí. Los objetos son instancias de clases, por lo tanto, cuando las clases están vinculadas representan abstractamente las relaciones objeto a objeto. Las clases serán usadas más adelante en el diseño navegacional para establecer nodos, y las relaciones para establecer enlaces [32].

Desde el punto de vista de diseño hipermedia, las relaciones expresan un aspecto significativo del dominio de la aplicación y no deben ser ocultados de los atributos de la clase. En el diseño del modelo conceptual la metodología proporciona tres construcciones de abstracción: agregación, generalización y subsistemas. La agregación se aplica para describir clases complejas como agregación de clases más simples, la generalización se usa para establecer jerarquías de clases mediante la herencia y los subsistemas abstraen modelos de dominio completos [32].

El comportamiento del modelo conceptual es el reflejo de las características deseadas de la aplicación final por lo que el diseño navegacional y de interfaces dependerán de como se ha estructurado este esquema [32].

- **Diseño navegacional**

El diseño navegacional se considera como una fase crítica en el diseño de aplicaciones hipermedia porque organiza la información presentada en el modelo conceptual para construir la estructura de navegación de una aplicación web. El modelo de navegación se construye como una vista subjetiva sobre el modelo conceptual y permite elaborar diferentes modelos de acuerdo al perfil de los usuarios [32].

El diseño navegacional se expresa en dos esquemas, el esquema de clases de navegación y el esquema de contextos de navegación. El esquema de clases de navegación se define por objetos navegables de una aplicación hipermedia, para plasmar una vista del dominio, el esquema se compone por nodos, enlaces y estructuras de acceso, los cuales se organizan dentro de contextos de navegación [32].

- **Nodos**

Son los contenedores básicos de información en aplicaciones hipermedia, se definen como vistas orientadas a objetos de las clases conceptuales. Un nodo permite combinar atributos de clases diferentes y puede contener atributos de un solo tipo y enlaces anclados [38].

- **Enlaces**

Reflejan las relaciones que los usuarios finales van a explorar, se definen como una vista de las relaciones del esquema conceptual y representan el flujo de la navegación. Una clase de enlace está determinado por la especificación de atributos y comportamientos, estos atributos expresan las

propiedades del propio enlace y son útiles para definir enlaces múltiples. Un enlace puede comportarse como un nodo intermedio entre el origen y el destino durante la navegación [38].

- **Estructuras de acceso**

Funcionan como índices para guiar a los usuarios finales a localizar la información deseada, algunos ejemplos son los menús, índices y vistas guiadas. Estas estructuras se modelan como clases que contienen selectores, objetos de destino y predicados. Los selectores representan los atributos de los objetos de destino ordenados de acuerdo a la estructura de datos definida y los predicados expresan los objetos que son accesibles en términos de sus propiedades [38].

- **Contexto navegacional**

Organiza el espacio de navegación en un orden particular con el fin de evitar información redundante y ayudar a los usuarios en la navegación [32]. Un contexto navegacional es un conjunto de nodos, enlaces, clases de contexto y contextos navegacionales anidados, donde, las clases de contexto determinan la información que será mostrada y los anclajes que estarán disponibles cuando se acceda a un objeto en particular [38].

- **Transformaciones de navegación**

Para completar la estructura de navegación se define las transformaciones de navegación que ocurren al atravesar los enlaces, y así especificar si un nodo origen-destino o ambos permanecerán activos o si todos los nodos destino de un enlace se activarán simultáneamente. En OOHDM se utilizan cartas de navegación para representar una navegación dinámica mediante el anidamiento estructural y de comportamiento [36].

- **Diseño de interfaces abstractas**

Una vez definida la estructura de navegación de la aplicación, ésta debe ser perceptible para el usuario mediante las interfaces, para ello esta fase define un modelo de interfaz abstracta a partir de la determinación de los objetos de la interfaz que percibirá el usuario. En particular la forma en que se verán los objetos, que objetos de la interfaz activarán la navegación, la sincronización de los objetos de la interfaz multimedia y que transformaciones de la interfaz se llevarán a cabo para no afectar al estado de la navegación [32].

Esta fase consiste en la construcción de diferentes interfaces sobre el modelo de navegación, estructurado en la fase anterior, esto permite comprender de mejor manera la estructura general de la aplicación. En OOHDM se utiliza las vistas abstractas de datos para describir una aplicación hipermedia [38].

- **Vistas abstractas de datos (ADV)**

Este enfoque de diseño se emplea para describir las interfaces de usuario de la aplicación, son abstractos porque solo representan el estado y la interfaz, mas no la implementación, la interfaz se ejerce a través de mensajes (eventos externos generados por el usuario). Los ADV en general son usados para mostrar la estructura de la interfaz y sus atributos, la relación de la interfaz con los objetos de navegación y el comportamiento de la interfaz ante eventos externos [38].

- **Diagramas de configuración**

Son útiles para representar la comunicación entre los objetos proporcionados y requeridos, este diagrama muestra la estructura de anidamiento de los ADV compuestos, lo que es adecuado para definir la forma en que el usuario interactuara con la aplicación hipermedia [38].

- **Implementación**

En esta última fase de la metodología el experto implementa el diseño a partir de todos los modelos que se han construido. La implementación de la aplicación requiere llevar a cabo la programación en el lenguaje que el desarrollador elija, por ejemplo, se puede realizar la interfaz con HTML y definir su apariencia con CSS, mientras que el comportamiento funcional en lenguajes de programación como PHP, JavaScript o C#, para finalmente obtener una aplicación de ejecución que pueda publicarse en un entorno donde el usuario final tendrá acceso [32], [36].

3.1.2. Integración de OOHDM con UML

Diferentes fases de la metodología OOHDM requieren la elaboración de diagramas, esquemas y modelos para representar el proceso de la construcción de la interfaz de usuario, UML es el lenguaje de modelado más utilizado en la actualidad para documentar este tipo de procesos y va de acuerdo a lo necesario para documentar lo requerido en ciertas fases de la metodología.

UML se implementa en fases de la metodología OOHDM como:

- Recopilación de requisitos, para establecer los casos de uso y los actores.
- Diseño conceptual, para la elaboración del esquema conceptual con sus clases, relaciones y atributos.
- Diseño navegacional, para la elaboración de los esquemas de clases navegacionales y los contextos de navegación, con sus nodos y enlaces.
- Diseño de interfaces abstractas, para representar los objetos abstractos de la interfaz y sus atributos.

A continuación, se define la notación requerida para la integración de UML con OOHDM:

Clases: se representa gráficamente como un rectángulo, que regularmente incluye nombre, atributos y operaciones, su función es describir un conjunto de objetos que tienen mismos atributos [37].

Interfaz: se utiliza para representar todo o una parte del comportamiento de una clase o componente [37].

Nodo: representa a los elementos físicos que existen en tiempo de ejecución y sirven para simbolizar un recurso computacional [37].

Relaciones

En UML existen cuatro tipos de relaciones [37]:

- **Dependencia:** se representa como una línea discontinua, es la relación semántica existente entre dos elementos.
- **Asociación:** se representa como una línea continua, es la relación estructural que existe entre clases que describe un conjunto de enlaces.
- **Generalización:** se representa como una línea continua con punta de flecha vacía, es la relación existente entre dos elementos, donde el primer elemento comparte una parte de las características del otro.
- **Realización:** se representa como una línea discontinua con punta de flecha vacía, es la relación semántica existente entre clasificadores.

Diagrama de clases: es una agrupación de clases, interfaces y colaboraciones relacionados entre sí, que abarcan la vista de diseño estático de un sistema [37].

Diagrama de objetos: es una agrupación de objetos y sus relaciones, sirven para representar instancias estáticas de los elementos existentes en los diagramas de clases [37].

Casos de uso: describen una agrupación de acciones que se ejecutan en un sistema por parte de diferentes actores con el fin de estructurar el comportamiento de un modelo [37].

Diagramas de casos de uso: presenta un conjunto de casos de uso y actores con sus relaciones para cubrir la vista de casos de uso estática de un sistema [37].

Elementos estructurales: son modelos UML que representan conceptos o cosas materiales, están compuestos de clases, interfaces, colaboraciones, casos de uso, componentes y nodos [37].

3.2. Desarrollo de la propuesta

3.2.1. Primera etapa – Recopilación de requisitos

Análisis de requerimientos del sistema

Por la similitud de la propuesta con directorios empresariales en línea y la existencia numerosa en la web, se requirió hacer una observación de los servicios ofertantes, a partir de ello se encontró características que comúnmente prometen, como la clasificación y promoción de los establecimientos, ayuda en el posicionamiento en internet, contacto con clientes potenciales y la disponibilidad del medio para conocer a la competencia. Muy pocas son gratuitas ya que la mayoría cobra una comisión al estar indexadas en motores de búsqueda [29].

También están las páginas de registro de empresas a cargo del gobierno, donde se reúne y clasifica con fines estadísticos, además de acceder a beneficios que brinda el estado, sin embargo, la información no es pública para los consumidores [39]. Otra variedad de sitio se encarga de brindar información pública del ranking de empresas por países como único fin [40].

Como resultado de la observación se decidió optar por agrupar las funcionalidades comunes de los sitios observados y proponer guías sobre migración al comercio

electrónico a través del uso de herramientas gratuitas como algo nuevo en sistemas de este tipo. La recolección de información con la encuesta ha sido útil para determinar el enfoque del desarrollo basado en software libre, esto permite responder a la necesidad actual de las MiPymes de la provincia de Napo que se han visto afectadas en sus ventas en su punto físico.

Requerimientos funcionales

La implementación de la plataforma web para apoyo a las MiPymes requiere estar disponible en internet y está enfocado en usuarios básicos y avanzados para consulta, gestión de información y administración del sitio. De los cuales se puede identificar los siguientes requerimientos funcionales:

- Registrar usuario
- Autenticar usuario
- Consultar catálogo de empresas
- Filtrar catálogo de empresas
- Consultar información individual de empresas
- Comentar el perfil de las empresas
- Gestionar comentarios de las empresas
- Valorar el perfil de las empresas
- Registrar empresa
- Gestionar información de la empresa
- Gestionar datos de la cuenta de usuario
- Consultar guías de aprendizaje
- Gestionar información de las tablas de la base de datos
- Gestionar usuarios del sistema
- Gestionar cuentas administrativas

Requerimientos no funcionales

Dentro de los requerimientos no funcionales del desarrollo de la plataforma web se considera lo siguiente:

- Seguridad del sitio
- Diseño adaptativo para garantizar el acceso desde diferentes dispositivos
- Interfaz amigable con el usuario

Actores

La plataforma web requiere la participación de distintos tipos de usuarios para la realización de tareas distintas, el administrador es el encargado de gestionar toda la información entrante al sistema, un usuario autenticado se encarga de gestionar toda la información referente a su cuenta y a su empresa registrada, mientras que un visitante de sitio es aquel que no requiere autenticación y se le permite el acceso a las vistas públicas del sitio que concierne al catálogo e información de las empresas.

Identificado a los actores principales del sistema se detalla las actividades que realizan:

Actor	Actividad
Administrador	<ul style="list-style-type: none"> • Gestionar usuarios, empresas, comentarios, información de las tablas de la base de datos y cuentas administrativas. • Registrar nuevos administradores.
Usuario autenticado	<ul style="list-style-type: none"> • Registrar una empresa • Gestionar la información del usuario y de la empresa. • Visualizar, comentar y valorar sobre el perfil de una empresa. • Acceder a guías de aprendizaje.
Visitante de sitio	<ul style="list-style-type: none"> • Visualizar el catálogo de empresas. • Filtrar empresas por nombre, categorías y ubicación. • Interactuar con el mapa del sitio.

*Tabla 3.2 Actores
Fuente: Investigador*

Relaciones de generalización entre actores

Los actores identificados están clasificados de acuerdo a su rol y autenticación en el sistema.

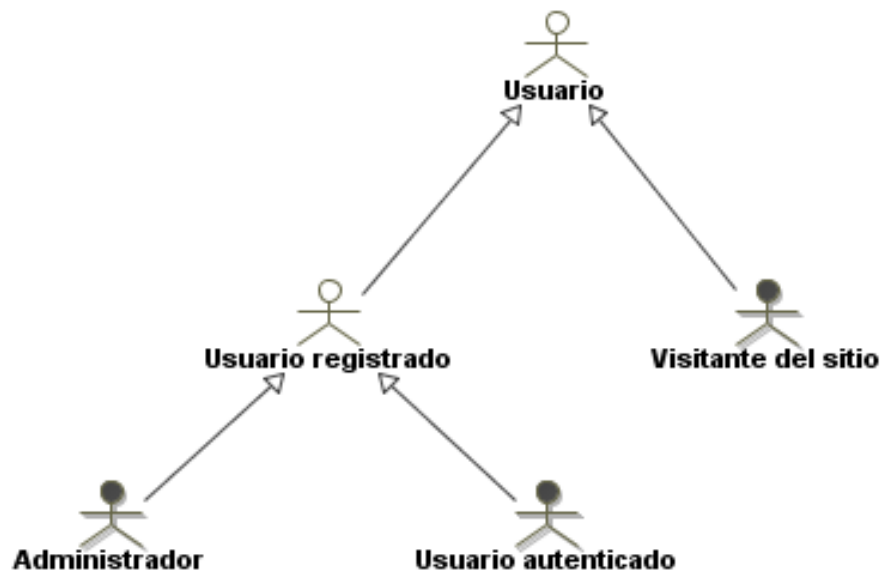


Figura 3.2 Relaciones de generalización entre actores
Fuente: Investigador

Diagrama de casos de uso

El diagrama de casos de uso muestra a los actores y sus tareas dentro de la aplicación, cada actor está vinculado con uno o varios casos de uso, los cuales describen una tarea en específico que les corresponde.

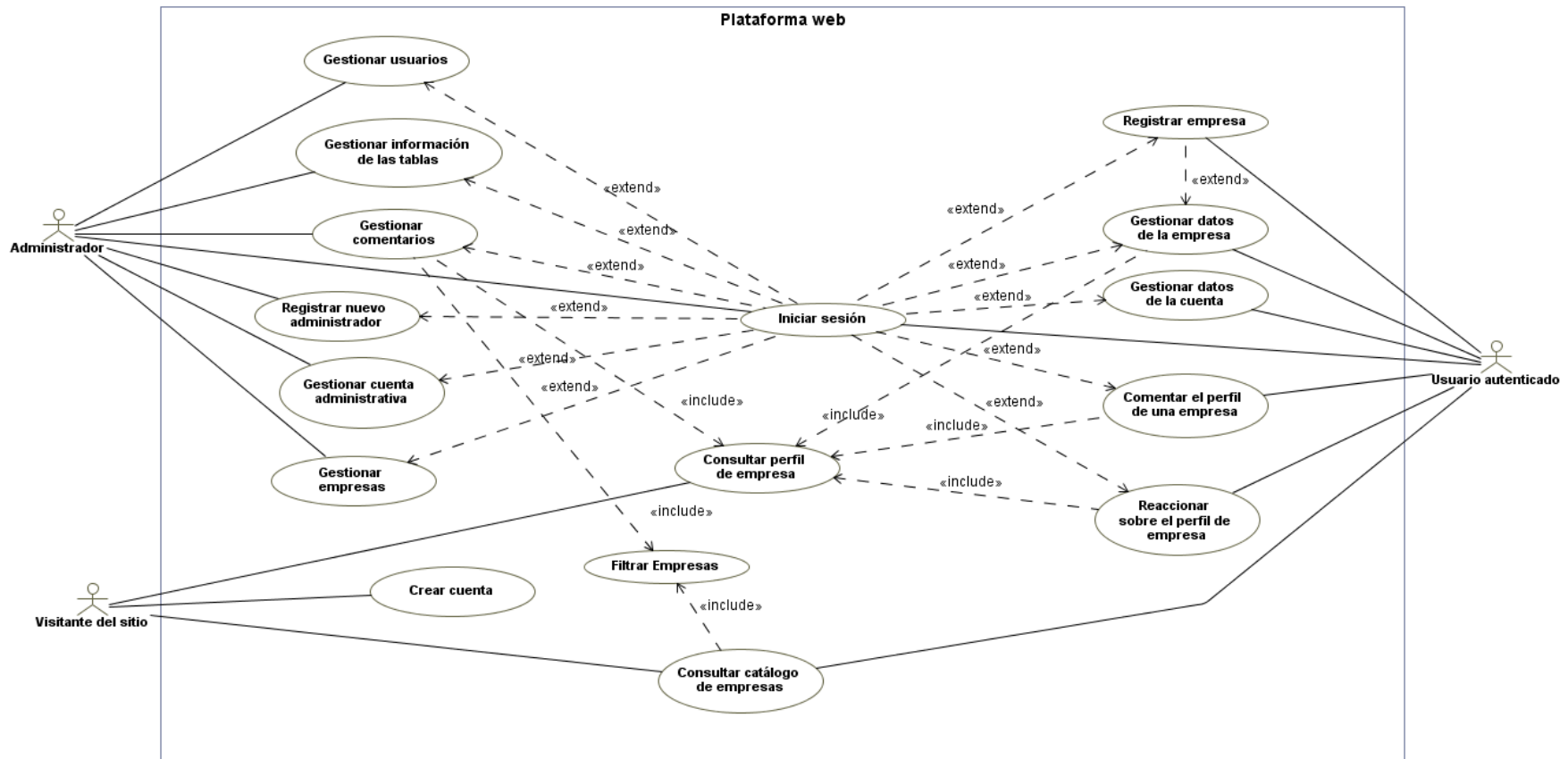


Figura 3.3 Diagrama de casos de uso
Fuente: Investigador

Documentación de casos de uso

Caso de uso	Iniciar sesión			CU1	
Actor(es)	Usuario autenticado, Administrador				
Precondición	El actor debe estar registrado				
Postcondición	El actor queda con una sesión activa para realizar actividades especiales en el sistema				
Autor	Luis Guachamin	Fecha	1/12/2020	Versión	1.0
Actualizado por		Fecha			
Propósito	Permitir el ingreso al sistema a un usuario y administrador				
Resumen	El actor proporciona sus credenciales en el formulario, los cuales se validan y se le permite el acceso.				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1	Ingresa email y contraseña				
2	Presiona en iniciar sesión		Valida los datos y redirecciona		
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		
2.1			Muestra un mensaje de error en caso de que los datos sean incorrectos o no se han llenado todos los campos		

Tabla 3.3 Caso de uso - Iniciar sesión
Fuente: Investigador

Caso de uso	Crear cuenta			CU2	
Actor(es)	Visitante de sitio				
Precondición	El actor no está registrado en el sistema				
Postcondición	El actor queda registrado y habilitado para iniciar sesión				
Autor	Luis Guachamin	Fecha	1/12/2020	Versión	1.0
Actualizado por		Fecha			
Propósito	Crear una cuenta de usuario nueva				
Resumen	El visitante de sitio llena el formulario con datos que son validados para la creación de una cuenta de usuario				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1	Ingresa el email, teléfono, contraseña y confirmación de contraseña		Valida que los datos ingresados sean correctos		
2	Presiona en registrar		Registra los datos y redirecciona		
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		
1.1			Muestra mensajes en caso de que los datos estén incorrectos		
2.1			Avisa si el email ya ha sido registrado en otra cuenta o no se han llenado todos los campos.		

*Tabla 3.4 Caso de uso - Crear cuenta
Fuente: Investigador*

Caso de uso	Consultar catálogo de empresas			CU3	
Actor(es)	Visitante de sitio, Usuario autenticado				
Precondición	El sistema debe cargar la lista de empresas registradas				
Postcondición	El actor podrá seleccionar una empresa del catálogo				
Autor	Luis Guachamin	Fecha	1/12/2020	Versión	1.0
Actualizado por		Fecha			
Propósito	Permitir a los actores consultar el listado de empresas registradas en el sistema				
Resumen	Los actores acceden a la lista de empresas almacenadas en la base de datos representadas en tarjetas y paginadas				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1	Accede al catálogo		Muestra todas las empresas visibles de la base de datos		
2			Lista todas las empresas en forma de tarjetas y con paginación		
3	Visualiza el catálogo				
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		

*Tabla 3.5 Caso de uso - Consultar catálogo de empresas
Fuente: Investigador*

Caso de uso	Filtrar empresas			CU4	
Actor(es)	Visitante de sitio, Usuario autenticado, Administrador				
Precondición	El sistema debe cargar la lista de empresas				
Postcondición	El actor podrá visualizar una nueva lista de empresas				
Autor	Luis Guachamin	Fecha	1/12/2020	Versión	1.0
Actualizado por		Fecha			
Propósito	Mostrar una nueva lista de empresas de acuerdo a los filtros o parámetros seleccionados				
Resumen	El actor tiene la posibilidad de seleccionar filtros para realizar consultas parametrizadas y obtener una búsqueda más a fin a sus requerimientos.				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1	Seleccionar uno o más filtros como: por nombre, por categorías y por ubicación.				
2	Presiona en filtrar		Muestra una lista nueva de empresas con los parámetros seleccionados		
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		
2.1			Muestra una lista vacía en caso de no encontrar empresas que coincidan con los filtros seleccionados		

*Tabla 3.6 Caso de uso - Filtrar empresas
Fuente: Investigador*

Caso de uso	Consultar perfil de una empresa		CU5		
Actor(es)	Visitante de sitio, Usuario autenticado, Administrador				
Precondición	El actor debió consultar las empresas				
Postcondición	El actor podrá visualizar la información pública de la empresa seleccionada				
Autor	Luis Guachamin	Fecha	1/12/2020	Versión	1.0
Actualizado por		Fecha			
Propósito	Permitir a los actores visualizar la información de una empresa				
Resumen	Los actores seleccionan una empresa de la lista y el sistema muestra el contenido				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1	Selecciona una empresa del listado		Carga la información de la empresa		
2			Muestra una nueva página con la información completa de la empresa		
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		

*Tabla 3.7 Caso de uso - Consultar perfil de una empresa
Fuente: Investigador*

Caso de uso	Registrar empresa			CU6	
Actor(es)	Usuario autenticado				
Precondición	El actor debe iniciar sesión				
Postcondición	Al usuario se le habilitará nuevas funciones para gestionar los datos de su empresa				
Autor	Luis Guachamin	Fecha	1/12/2020	Versión	1.0
Actualizado por		Fecha			
Propósito	Registrar una empresa en el sistema				
Resumen	El usuario ingresa los datos de la empresa y se almacena en la base de datos para su visualización en el catálogo				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1	Ingresar datos de la empresa como: información general, dirección, redes sociales, canales de venta, galería de imágenes y horario de atención.		Valida que los datos sean correctos		
2	Presiona en registrar		Registra los datos, habilita nuevas funciones y redirecciona		
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		
1.1			Muestra mensaje de error en caso de ingresar datos incorrectos		

*Tabla 3.8 Caso de uso - Registrar empresa
Fuente: Investigador*

Caso de uso	Gestionar datos de la empresa		CU7		
Actor(es)	Usuario autenticado				
Precondición	Iniciar sesión Tener una empresa registrada				
Postcondición	Empresa modificada para mostrar u ocultar del catálogo				
Autor	Luis Guachamin	Fecha	1/12/2020	Versión	1.0
Actualizado por		Fecha			
Propósito	Permitir al usuario actualizar o eliminar datos de la empresa				
Resumen	El usuario ingresa nuevos datos en los campos reemplazando los anteriores y el sistema actualiza los cambios				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1	Actualiza o elimina datos de la empresa				
2	Presiona en el botón para confirmar la acción		Valida los datos ingresados y los actualiza		
3			Muestra mensaje de los cambios realizados		
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		
2.1			Muestra mensaje de error en caso de que los datos sean incorrectos		
3.1			Muestra mensaje de error en caso de que la sesión del usuario haya expirado o no tiene permisos suficientes		

*Tabla 3.9 Caso de uso - Gestionar datos de la empresa
Fuente: Investigador*

Caso de uso	Gestionar datos de la cuenta			CU8	
Actor(es)	Usuario autenticado				
Precondición	Iniciar sesión				
Postcondición	Usuario modificado o eliminado				
Autor	Luis Guachamin	Fecha	1/12/2020	Versión	1.0
Actualizado por		Fecha			
Propósito	Permitir al usuario modificar o eliminar su información				
Resumen	El usuario cambia o elimina información de su cuenta y el sistema registra los cambios				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1	El usuario ingresa o elimina datos de su cuenta				
2	Presiona en el botón para confirmar la acción		Valida y registra los cambios realizados		
3			Muestra mensaje de los cambios realizados		
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		
2.1			Muestra mensaje de error en caso de que los datos sean incorrectos		
3.1			Muestra mensaje de error en caso de que la sesión del usuario haya expirado o no tiene permisos suficientes		

*Tabla 3.10 Caso de uso - Gestionar datos de la cuenta
Fuente: Investigador*

Caso de uso	Comentar el perfil de una empresa		CU9		
Actor(es)	Usuario autenticado				
Precondición	Iniciar sesión Consultar el perfil de una empresa				
Postcondición	El comentario quedará registrado y será visible en el perfil de la empresa				
Autor	Luis Guachamin	Fecha	1/12/2020	Versión	1.0
Actualizado por		Fecha			
Propósito	Permitir al usuario publicar un comentario en el perfil de una empresa				
Resumen	El usuario comenta sobre el perfil de la empresa seleccionada y el sistema lo registra				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1	Ingresa el comentario en el campo de texto				
2	Presiona en publicar		Valida y registra el comentario		
3			Muestra el contenido publicado		
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		
2.1			Muestra mensaje de error en caso de que los datos estén incorrectos		
3.1	No ha iniciado sesión		Muestra aviso en caso de que el usuario no se ha autenticado		

*Tabla 3.11 Caso de uso - Comentar perfil de empresa
Fuente: Investigador*

Caso de uso	Reaccionar sobre el perfil de una empresa		CU10		
Actor(es)	Usuario autenticado				
Precondición	Iniciar sesión Consultar el perfil de una empresa				
Postcondición	El contador de me gusta del perfil la empresa se actualiza				
Autor	Luis Guachamin	Fecha	1/12/2020	Versión	1.0
Actualizado por		Fecha			
Propósito	Permitir al usuario indicar interés sobre el perfil de una empresa				
Resumen	El usuario presiona en el botón me gusta, el sistema lo registra y actualiza los cambios				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1	Presiona en me gusta		Registra la acción y actualiza el contador		
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		
1.1	No ha iniciado sesión		Muestra aviso en caso de que el usuario no se ha autenticado		

*Tabla 3.12 Caso de uso – Reaccionar sobre el perfil de una empresa
Fuente: Investigador*

Caso de uso	Gestionar usuarios			CU11	
Actor(es)	Administrador				
Precondición	Iniciar sesión				
Postcondición	Usuario nuevo, modificado o eliminado				
Autor	Luis Guachamin	Fecha	1/12/2020	Versión	1.0
Actualizado por		Fecha			
Propósito	Permitir al administrador gestionar a los usuarios registrados				
Resumen	El administrador crea, actualiza o elimina un usuario normal del sistema				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1	Crea, actualiza o elimina un usuario				
2	Confirma la acción		Valida y registra los cambios		
			Muestra mensaje de los cambios realizados		
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		
2.1			Muestra mensaje de error en caso de ingresar datos incorrectos		
2.2			Muestra mensaje de error en caso de que la sesión del usuario haya expirado o no tiene permisos suficientes		

*Tabla 3.13 Caso de uso - Gestionar usuarios
Fuente: Investigador*

Caso de uso	Gestionar información de las tablas			CU12	
Actor(es)	Administrador				
Precondición	Iniciar sesión				
Postcondición	Contenido de las tablas insertadas, modificadas o eliminadas				
Autor	Luis Guachamin	Fecha	1/12/2020	Versión	1.0
Actualizado por		Fecha			
Propósito	Permitir al administrador gestionar el CRUD del contenido de las tablas				
Resumen	El administrador consulta el contenido de las tablas permitidas para insertar, modificar y eliminar filas para mantener el sistema actualizado				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1	Selecciona la tabla a consultar como: provincia, cantón, red social, canal de venta, categoría y subcategoría		Muestra el contenido de la tabla		
2	Inserta, modifica, elimina filas		Registra los cambios realizados		
3			Muestra mensaje de la acción realizada con éxito		
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		
3.1			Muestra mensaje de error en caso de que la sesión del usuario haya expirado o no tiene permisos suficientes		

Tabla 3.14 Caso de uso - Gestionar información de las tablas

Fuente: Investigador

Caso de uso	Gestionar comentarios			CU13	
Actor(es)	Administrador				
Precondición	Iniciar sesión Filtrar y seleccionar empresa				
Postcondición	Comentario eliminado del perfil de una empresa				
Autor	Luis Guachamin	Fecha	1/12/2020	Versión	1.0
Actualizado por		Fecha			
Propósito	Permitir al administrador consultar y eliminar comentarios no apropiados del perfil de una empresa				
Resumen	El administrador selecciona un comentario, lo elimina y el sistema lo registra				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1			Carga los comentarios del perfil de la empresa		
2	Selecciona un comentario				
3	Visualiza y elimina el comentario		Registra los cambios realizados		
4			Muestra mensaje de la acción realizada con éxito		
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		
4.1			Muestra mensaje de error en caso de que la sesión del usuario haya expirado o no tiene permisos suficientes		

*Tabla 3.15 Caso de uso - Gestionar comentarios
Fuente: Investigador*

Caso de uso	Registrar nuevo administrador		CU14		
Actor(es)	Administrador				
Precondición	Iniciar sesión				
Postcondición	Nuevo administrador registrado y habilitado en el sistema				
Autor	Luis Guachamin	Fecha	1/12/2020	Versión	1.0
Actualizado por		Fecha			
Propósito	Permitir al administrador crear un nuevo usuario administrador				
Resumen	El actor ingresa los datos del nuevo administrador y el sistema lo registra				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1	Ingresa los datos del nuevo administrador como: email, teléfono móvil y contraseña		Valida los datos ingresados		
2	Presiona en registrar		Registra los datos		
3			Muestra mensaje de la acción realizada con éxito		
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		
1.1			Muestra mensaje de error si los datos son incorrectos		

*Tabla 3.16 Caso de uso - Registrar nuevo administrador
Fuente: Investigador*

Caso de uso	Gestionar cuenta administrativa		CU15		
Actor(es)	Administrador				
Precondición	Iniciar sesión				
Postcondición	Cuenta de administrador actualizada				
Autor	Luis Guachamin	Fecha	1/12/2020	Versión	1.0
Actualizado por		Fecha			
Propósito	Permitir al administrador modificar los datos de su cuenta				
Resumen	El administrador ingresa nuevos datos y el sistema los actualiza reemplazando los anteriores				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1	Ingresa nuevos datos como: email, contraseña, móvil		Valida los datos ingresados		
2	Presiona en guardar cambios		Registra los cambios realizados		
			Muestra mensaje de la acción realizada con éxito		
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		
1.1			Muestra mensaje de error si los datos son incorrectos		
3.1			Muestra mensaje de error si la acción no se realizó con éxito		

*Tabla 3.17 Caso de uso - Gestionar cuenta administrativa
Fuente: Investigador*

Caso de uso	Gestionar empresas			CU16	
Actor(es)	Administración				
Precondición	Iniciar sesión				
Postcondición	Empresa modificada o eliminada				
Autor	Luis Guachamin	Fecha	1/12/2020	Versión	1.0
Actualizado por		Fecha			
Propósito	Permitir al administrador modificar o eliminar información de la empresa				
Resumen	El administrador actualizar datos (visualización y verificación) o elimina y el sistema registra los cambios				
Curso/Flujo Normal (Básico)					
Paso	Actor		Sistema		
1	El administrador ingresa o elimina datos de la empresa				
2	Presiona en el botón para confirmar la acción		Valida y registra los cambios realizados		
3			Muestra mensaje de los cambios realizados		
Curso/Flujo Alterno (Excepcional)					
Paso	Actor		Sistema		
2.1			Muestra mensaje de error en caso de que los datos sean incorrectos		

*Tabla 3.18 Caso de uso - Gestionar empresas
Fuente: Investigador*

3.2.2. Segunda etapa - Diseño conceptual

El modelo conceptual está conformado por clases, atributos, operaciones y relaciones correspondientes a las entidades identificadas en la aplicación.

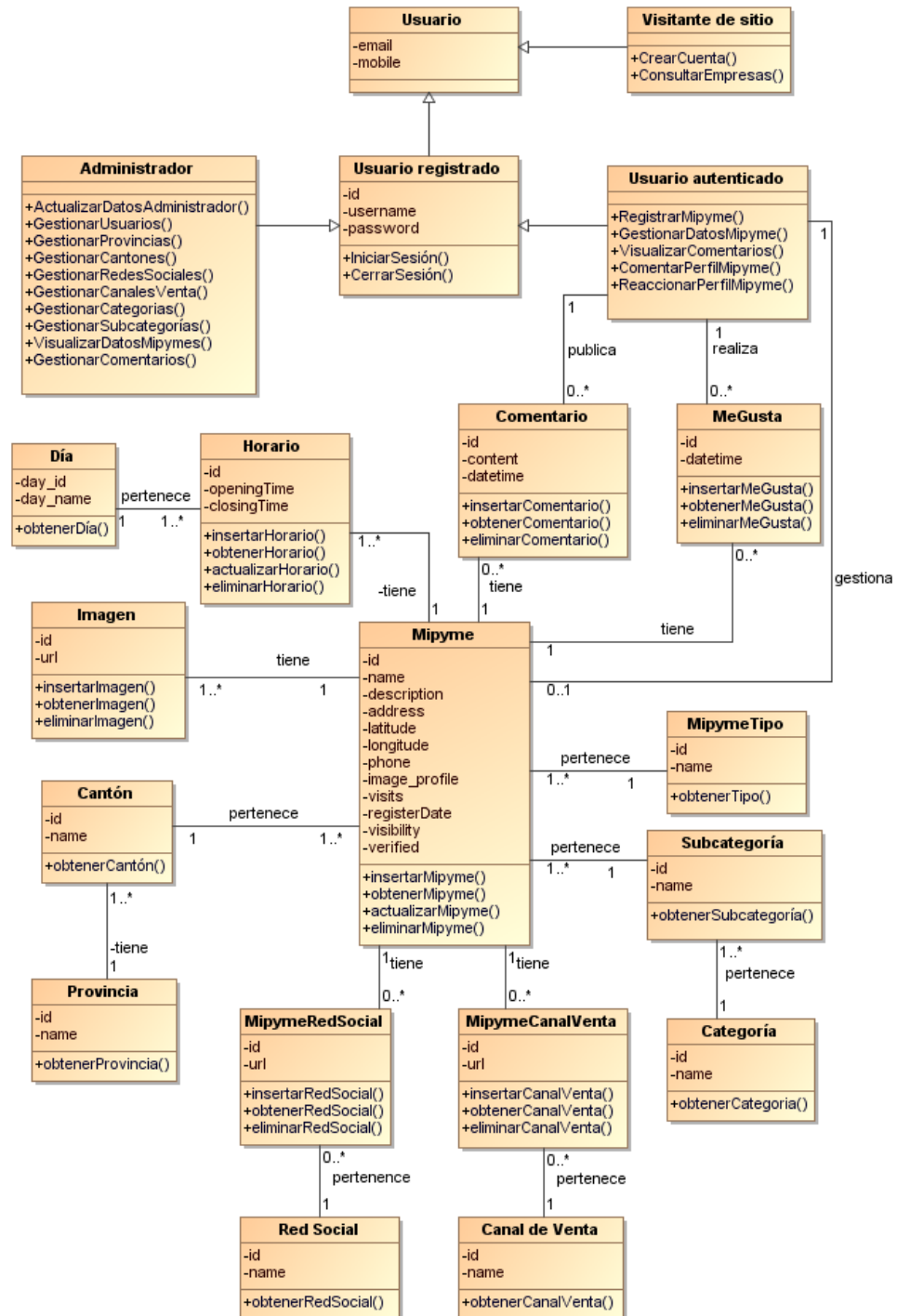


Figura 3.4 Modelo conceptual de la plataforma web
Fuente: Investigador

Las clases representan a los objetos del sistema, es importante detallar su participación dentro del esquema:

Diagrama de clases – Usuarios

La clase usuario representa a la clase general del cual las clases visitante de sitio y usuario autenticado heredan sus atributos. La clase administrador y usuario normal heredan los atributos y operaciones del usuario autenticado. La clase administrador es la representación del usuario que se hará cargo de la administración de la aplicación. La clase usuario normal representa al usuario que registra o no una empresa y realiza actividades habilitadas.

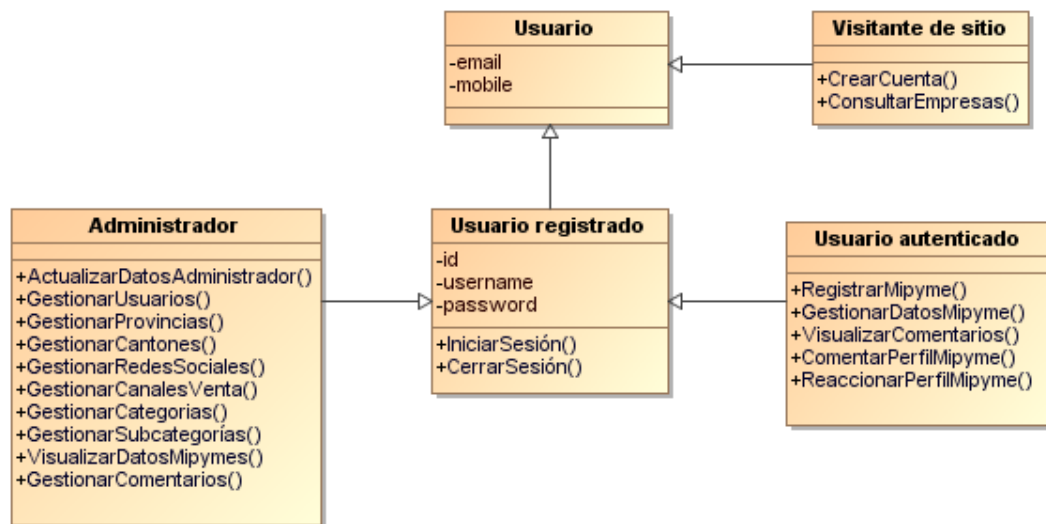


Figura 3.5 Diagrama de clases – Usuarios
Fuente: Investigador

Diagrama de clases – Gestión de MiPymes

El diagrama representa la relación de un usuario normal y la empresa, donde el usuario puede ser o no el propietario para gestionar la información, además realiza acciones como comentar y votar sobre el perfil de una empresa registrada.

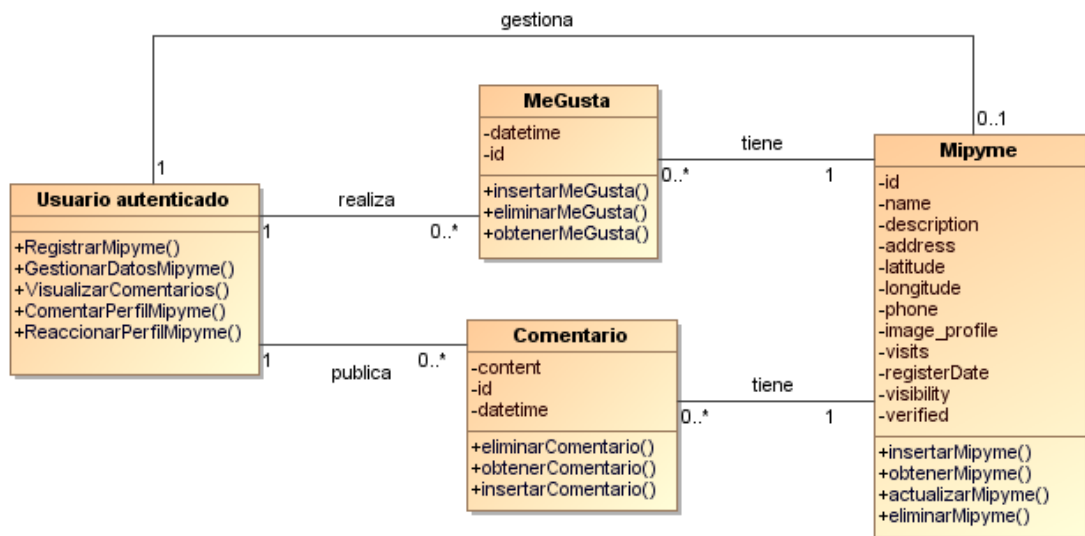


Figura 3.6 Diagrama de clases - Gestión de MiPymes
Fuente: Investigador

Diagrama de clases – MiPymes

El diagrama muestra la estructura de una empresa dentro del contexto de la aplicación, ésta cuenta con vinculaciones a diferentes clases como: la clases horario que define los días y horas de atención al cliente, la clase imagen que almacena imágenes, las clase MipymeTipo para clasificar a la empresa por su tipo, la clase MipymeSubcategoría cataloga a la empresa por una subcategoría en específico que proviene de una categoría más grande de la clase categoría, las clases canal de venta y red social están vinculadas a la empresa a través de clases intermedias (MipymeCanalVenta y MipymeRedSocial) para especificar los medios en los que se encuentra internet, por último la relación existente con la clase cantón cataloga a la empresa por cantones dentro de una provincia.

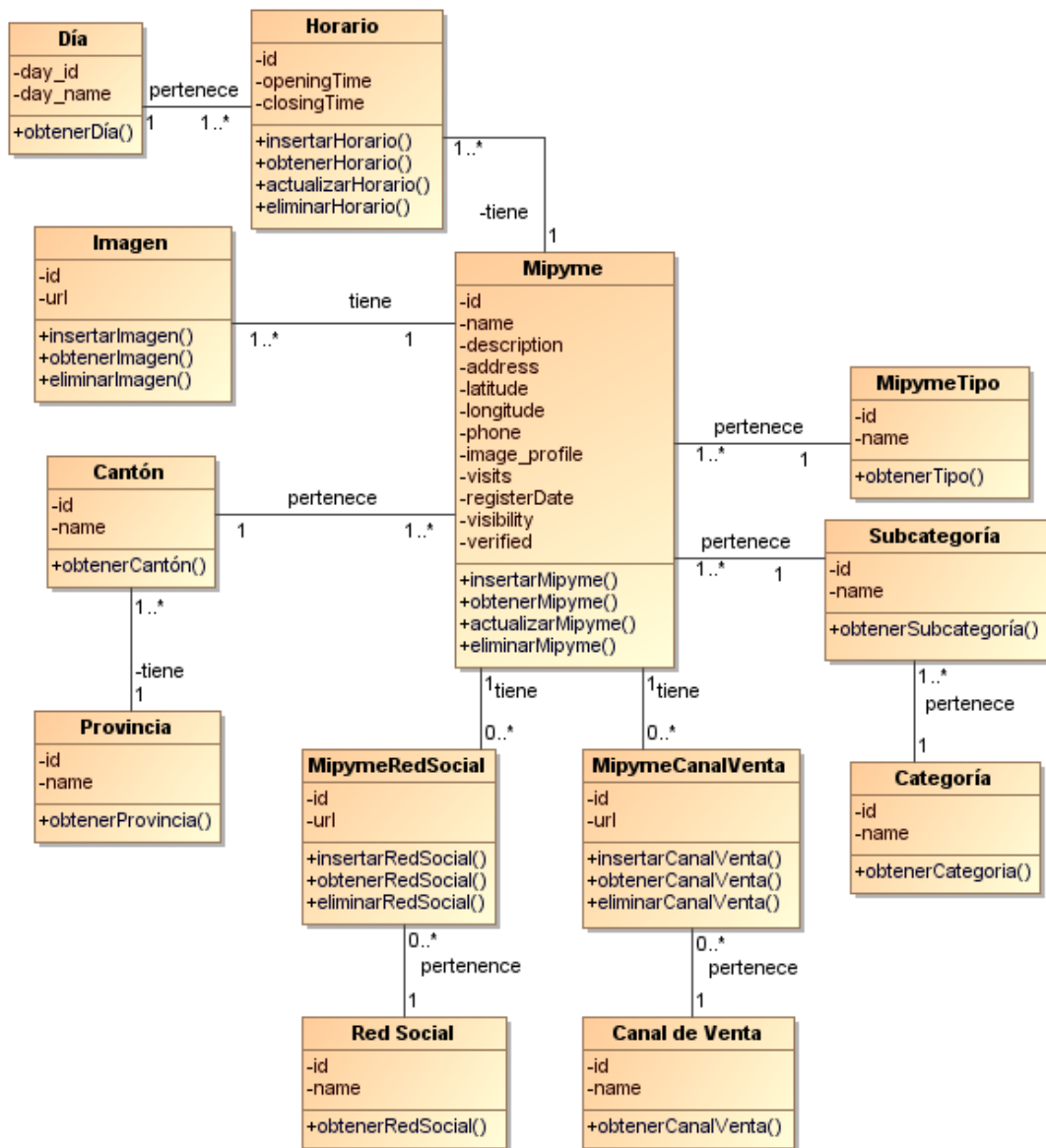


Figura 3.7 Diagrama de clases – MiPymes
Fuente: Investigador

3.2.3. Tercera etapa - Diseño navegacional

Las clases del modelo conceptual se convierten en objetos de navegación y los escenarios en contextos de navegación. Los objetos de navegación identificados para el diseño de clases y contextos navegacionales son:

- Administrador
- Usuario autenticado
- Visitante de sitio

Antes de diseñar los contextos de navegación, se define los esquemas de clases de navegación los cuales son:

- Esquema de clases navegacionales de un visitante de sitio
- Esquema de clases navegacionales de un usuario autenticado
- Esquema de clases navegacionales de un administrador

Esquema de clases navegacionales de un visitante de sitio

El esquema muestra los nodos identificados en la navegación del visitante del sitio, de igual manera los enlaces correspondientes al flujo permitido para la navegación dentro la plataforma, los nodos requeridos están representados por el menú principal, inicio, catalogo MiPymes, MiPyme, acerca de, iniciar sesión y registrarse como se muestra en la Figura 3.8.

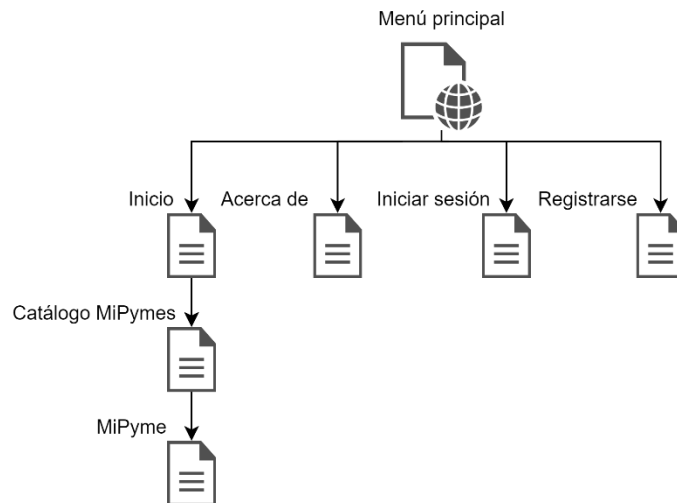


Figura 3.8 Esquema de clases navegacionales – Visitante de sitio

Fuente: Investigador

Esquema de clases navegacionales de un usuario autenticado

El esquema muestra los nodos identificados en la navegación del usuario autenticado, de igual manera los enlaces correspondientes al flujo permitido para la navegación dentro la plataforma, los nodos requeridos están representados por iniciar sesión, menú del usuario autenticado, inicio, catálogo MiPymes, MiPyme, acerca de, catálogo de guías, guía, registrar MiPyme y cuenta del usuario autenticado como se muestra en la Figura 3.9.

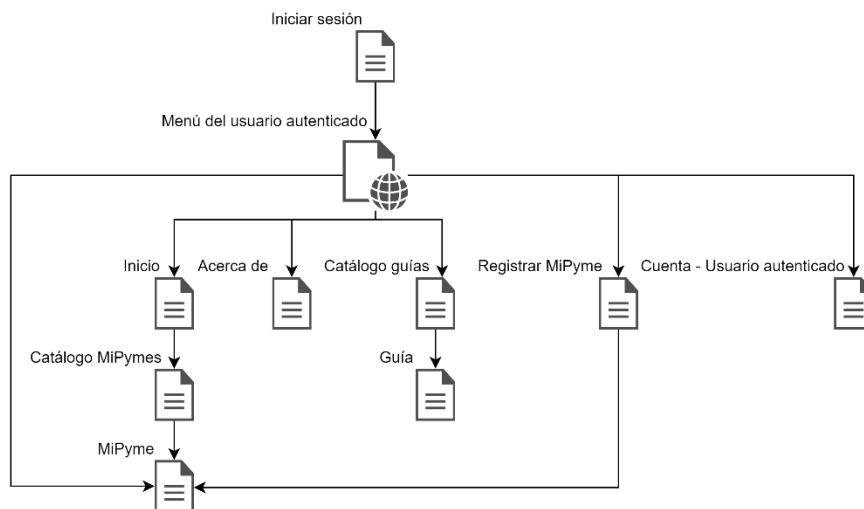


Figura 3.9 Esquema de clases navegacionales - Usuario autenticado

Fuente: Investigador

Esquema de clases navegacionales de un administrador

El esquema muestra los nodos identificados en la navegación del administrador, de igual manera los enlaces correspondientes al flujo permitido para la navegación dentro la plataforma, los nodos requeridos están representados por iniciar sesión, menú del administrador, provincias y cantones, MiPymes, dashboard, categorías, redes sociales, canales de venta, administradores, usuarios y cuenta del administrador como se muestra en la Figura 3.10.

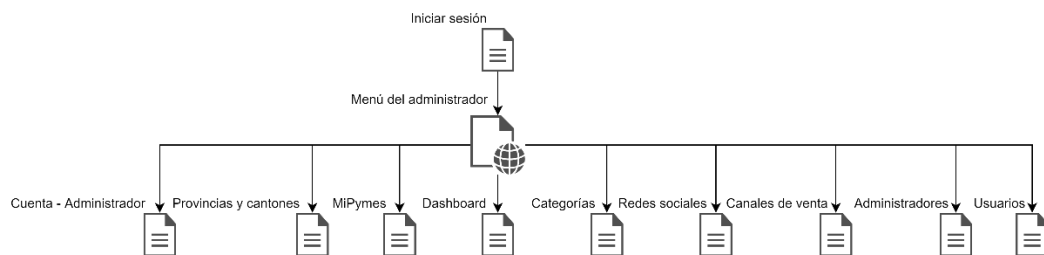


Figura 3.10 Esquema de clases navegacionales – Administrador

Fuente: Investigador

Los contextos de navegación son:

- Contexto navegacional de un visitante de sitio
- Contexto navegacional de un usuario autenticado
- Contexto navegacional de un administrador

Contexto navegacional de un visitante de sitio

El diagrama muestra el flujo de la navegación de un visitante de sitio en la aplicación, la navegación está restringida por la información disponible al público en general. Por lo tanto, el menú principal contiene enlaces a distintos nodos públicos como:

- Inicio, contiene el nodo del catálogo con la información de las empresas y a su vez enlaza a sus perfiles.
- Acerca de, enlaza al nodo de la descripción del sitio.

- Inicio de sesión, enlaza al nodo del formulario de ingreso de las credenciales del usuario registrado.
- Registro, enlaza al nodo del formulario de registro para la creación de una cuenta de usuario nueva.

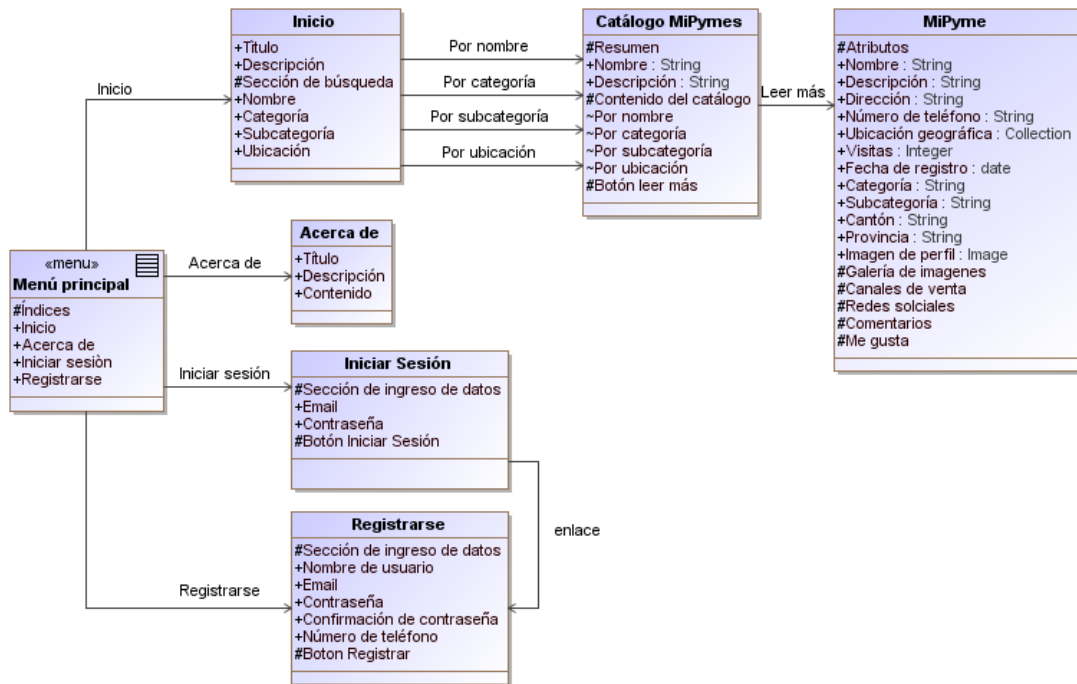


Figura 3.11 Contexto navegacional - Visitante de sitio
Fuente: Investigador

Contexto navegacional de un usuario autenticado

El diagrama muestra el flujo de la navegación de un usuario autenticado, al ser un usuario con una sesión activa dispone de nuevos enlaces con destino hacia nuevas áreas del sistema, estas nuevas funcionalidades se representan dentro del nodo menú de usuario autenticado el cual enlaza a otros nodos como:

- Cerrar sesión, enlaza al menú principal del sitio después de realizar el proceso para terminar con la sesión del usuario.
- Registrar MiPyme, enlaza al nodo con el formulario de registro de una empresa, esta función está habilitada siempre y cuando el usuario no tenga un registro existente.

- Gestionar MiPyme, enlaza al nodo que dispone de formularios para gestionar la información de la empresa, está disponible siempre y cuando el usuario haya registrado una empresa.
- Cuenta, enlaza al nodo que contiene el perfil del usuario, dentro de este el usuario puede modificar sus datos.
- Guías, enlaza al nodo que contiene las guías de aprendizaje, disponible para todos los usuarios autenticados.

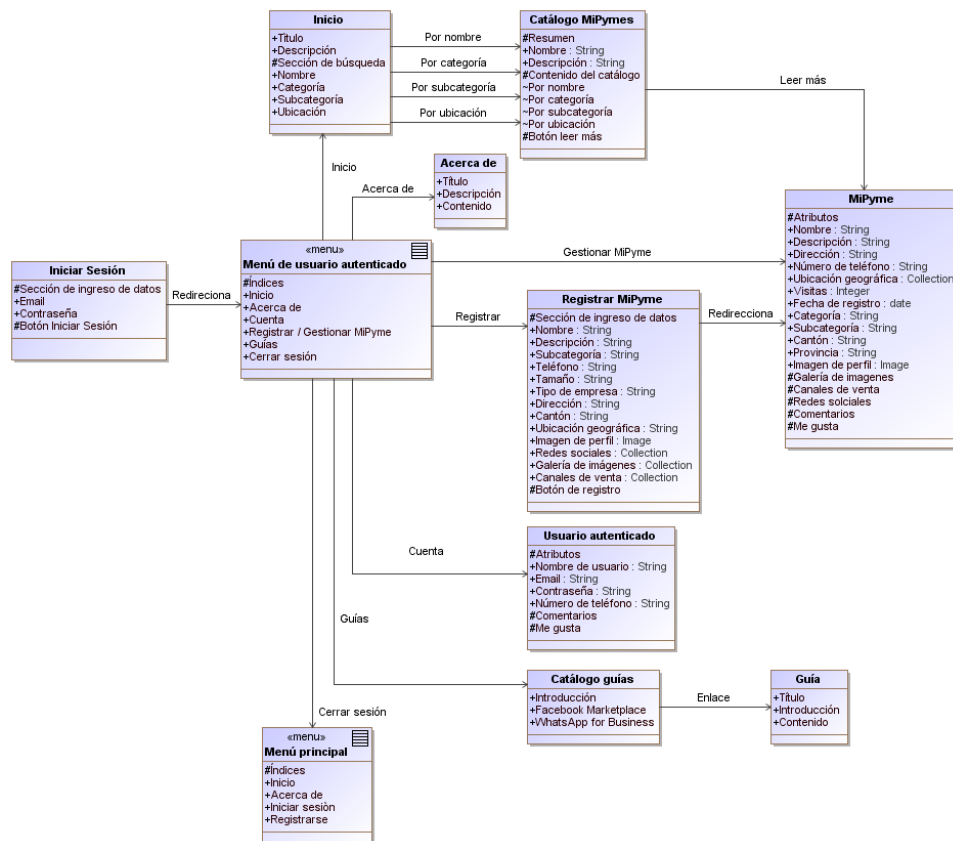


Figura 3.12 Contexto navegacional - Usuario autenticado
Fuente: Investigador

Contexto navegacional de un administrador

El diagrama muestra el flujo de la navegación de un administrador, al ser un usuario con funcionalidades especiales cuenta con enlaces a nodos que corresponden a la administración de la información de la plataforma. El menú del administrador tiene enlaces que derivan en distintos nodos como:

- Usuarios, enlaza al nodo de la administración de los usuarios del sistema.
- MiPymes, enlaza al nodo para gestionar las empresas registradas.
- Cuenta, enlaza al nodo que contiene funciones para modificar datos de la cuenta del administrador.
- Provincias y cantones, enlaza al nodo para gestionar la información de las provincias y cantones, con el fin de delimitar geográficamente la cobertura de la plataforma.
- Dashboard, enlaza al nodo que contiene tarjetas, gráficos y estadísticas que resumen los datos de la plataforma.
- Categorías, enlaza al nodo para gestionar la información de las categorías y subcategorías que clasifican a las empresas registradas.
- Redes sociales, enlaza al nodo para gestionar la información de las redes sociales existentes en internet.
- Canales de venta, enlaza al nodo para gestionar la información de los medios en los cuales una empresa ofrece sus productos.
- Administradores, enlaza al nodo que contiene a los administradores del sistema y permite gestionar su información.

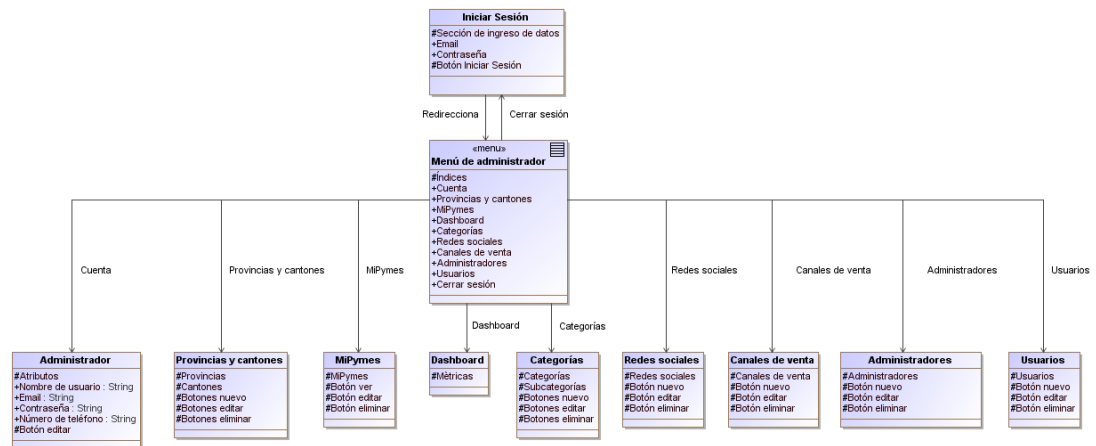


Figura 3.13 Contexto navegacional – Administrador
Fuente: Investigador

Diagrama de navegación final

Una vez definido el contexto navegacional para cada uno de los involucrados de la plataforma, se diseñó el diagrama de navegación final para proporcionar una vista

general del flujo de navegación de toda la aplicación. El diagrama parte desde el menú principal y se divide en diferentes ramas dependiendo del usuario que interactúa en el sitio, a diferencia de los contextos navegacionales anteriores, el diagrama muestra de forma resumida los enlaces y los nodos, esto permite representar la delimitación del espacio de navegación.

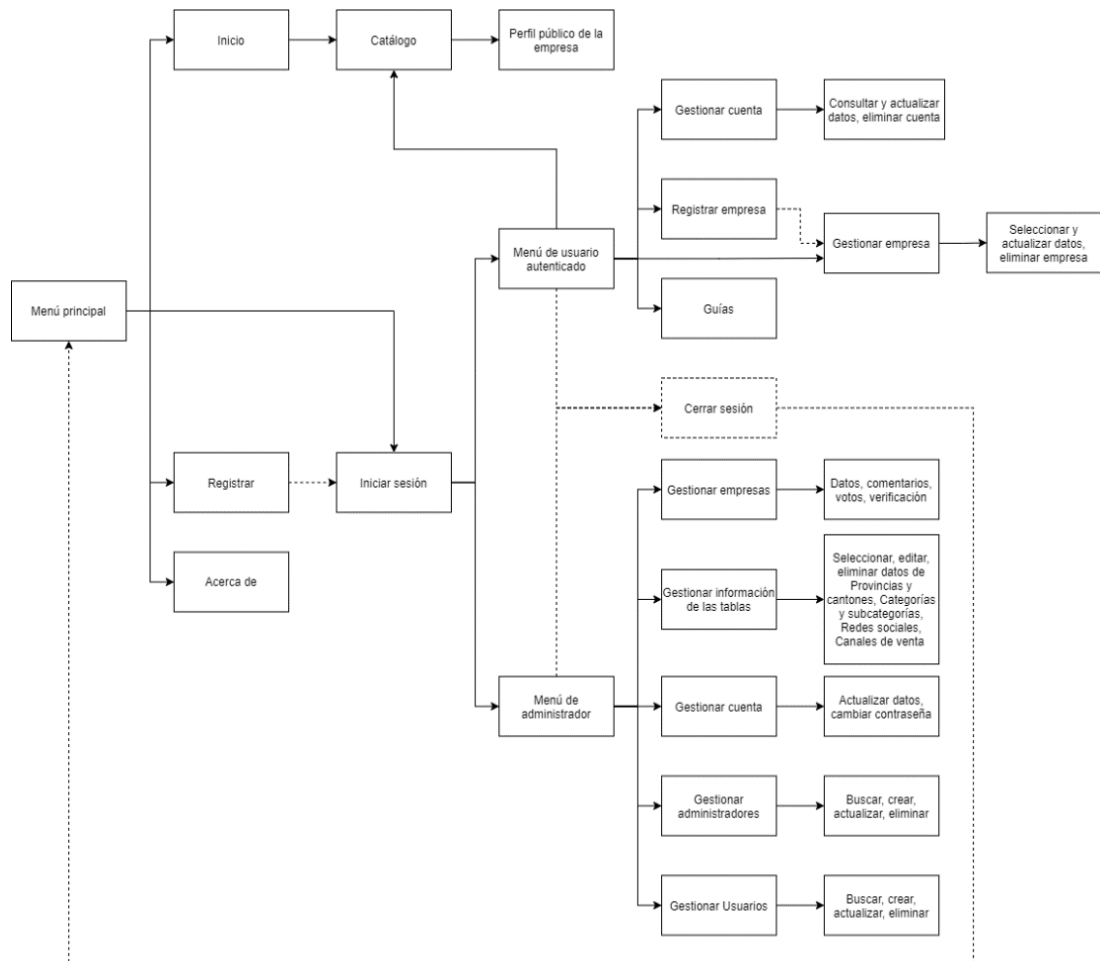


Figura 3.14 Diagrama de navegación final
Fuente: Investigador

3.2.4. Cuarta etapa - Diseño de interfaces abstractas

Para el diseño de las interfaces abstractas se aplicará el modelo de vistas abstractas de datos (ADV) como propone la metodología OOADM, ya que se enfoca en la representación del estado y la interfaz, cada vista cuenta con estructuras y atributos que representan a la información que se muestra en las interfaces y como éstas serán percibidas, para finalmente comprender la estructura general de la aplicación.

La vista de la “barra de navegación” representa al elemento que se ubica en el encabezado de la página, contiene los enlaces públicos para los visitantes del sitio como se muestra en la Figura 3.15.



Figura 3.15 ADV – Barra de navegación
Fuente: Investigador

La vista “pie de página” representa al elemento que se ubica en la parte inferior de la página y contiene la marca registrada como se muestra en la Figura 3.16.

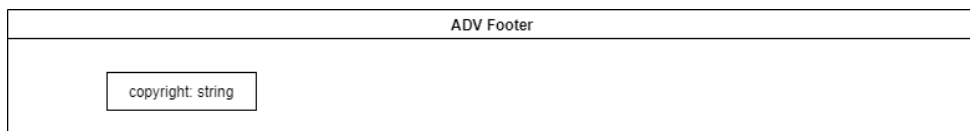


Figura 3.16 ADV – Pie de página
Fuente: Investigador

La vista “inicio” representa a la página principal que se muestra al cargar la plataforma, esta vista contiene un formulario de búsqueda y el catálogo de las empresas registradas como se muestra en la Figura 3.17.

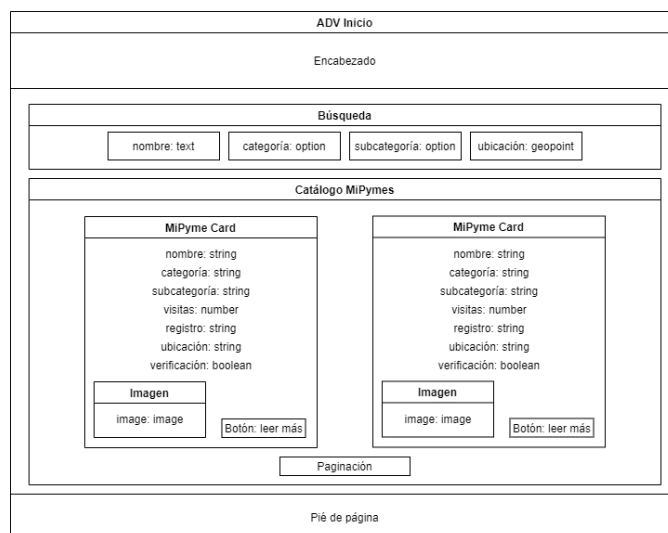


Figura 3.17 ADV – Inicio
Fuente: Investigador

La vista “acerca de” representa a la página que contiene la información, descripción y beneficios del uso de la plataforma como se muestra en la Figura 3.18.

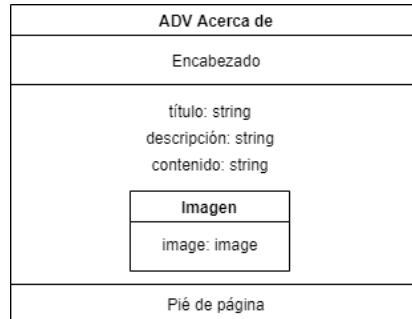


Figura 3.18 ADV - Acerca de
Fuente: Investigador

La vista “inicio de sesión” representa a la página que contiene el formulario para el ingreso de las credenciales de un usuario registrado como se muestra en la Figura 3.19.

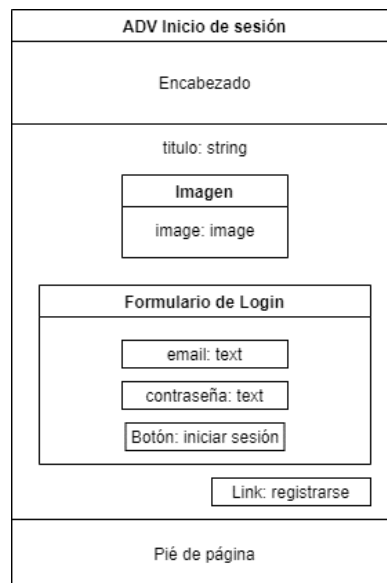


Figura 3.19 ADV - Inicio de sesión
Fuente: Investigador

La vista “registro” representa a la página que contiene el formulario de ingreso de datos personales para el registro de nuevos usuarios como indica la Figura 3.20.

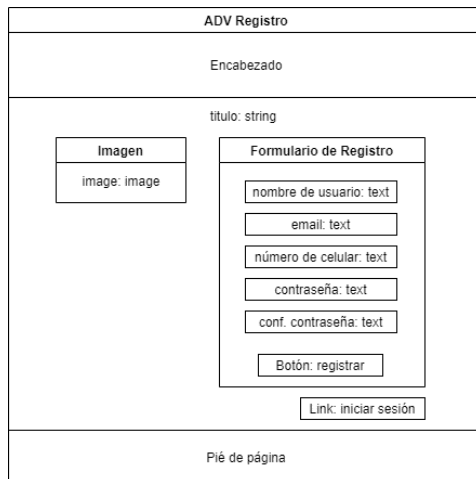


Figura 3.20 ADV – Registro
Fuente: Investigador

La vista “perfil de mipyme” representa la página que contiene la información pública de una empresa, dentro de la vista se incluye el formulario de comentario y botones para realizar acciones sociales como se muestra en la Figura 3.21.

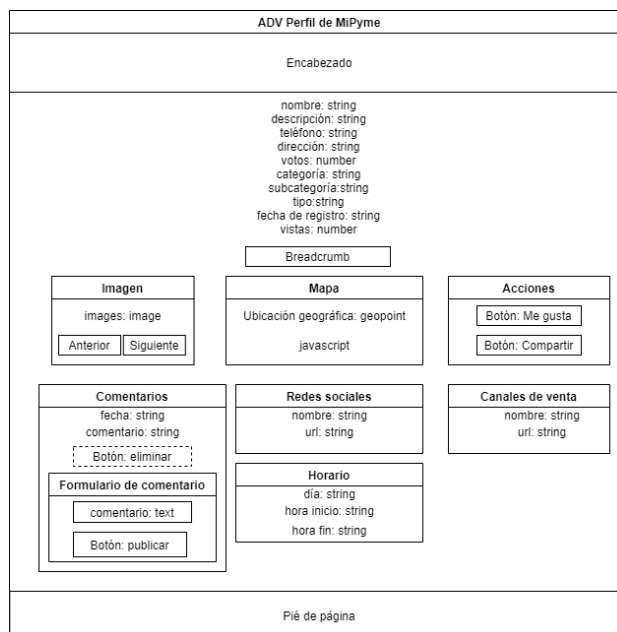


Figura 3.21 ADV - Perfil de MiPyme
Fuente: Investigador

La vista “barra de navegación de usuario autenticado” representa al elemento que se ubica en el encabezado de la página y contiene enlaces a otras vistas que son únicamente para usuarios con sesiones activas en la plataforma como muestra la Figura 3.22.

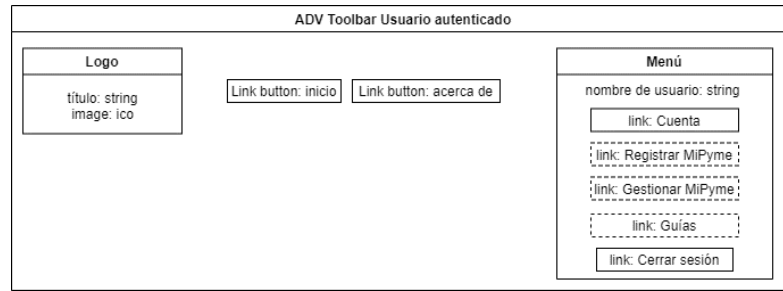


Figura 3.22 ADV – Barra de navegación de usuario autenticado
Fuente: Investigador

La vista “perfil de usuario autenticado” representa a la página que contiene la información completa de un usuario como muestra la Figura 3.23.

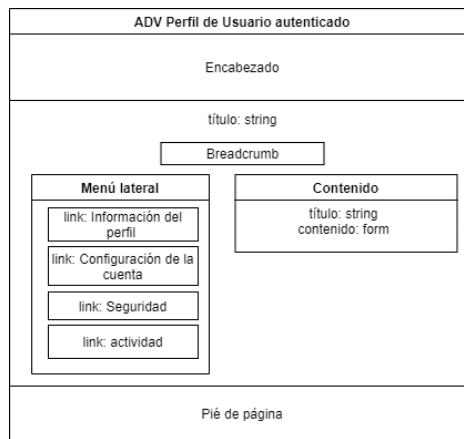


Figura 3.23 ADV - Perfil de usuario autenticado
Fuente: Investigador

La vista “actividad de usuario autenticado” representa a la página que contiene la lista de comentarios y reacciones que ha realizado un usuario en distintos perfiles de empresas como muestra la Figura 3.24.

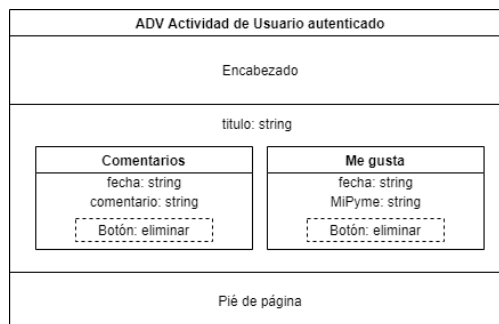


Figura 3.24 ADV – Actividad de usuario autenticado
Fuente: Investigador

La vista “editar perfil de usuario autenticado” representa la página que contiene el formulario para editar la información de un usuario como indica la Figura 3.25.

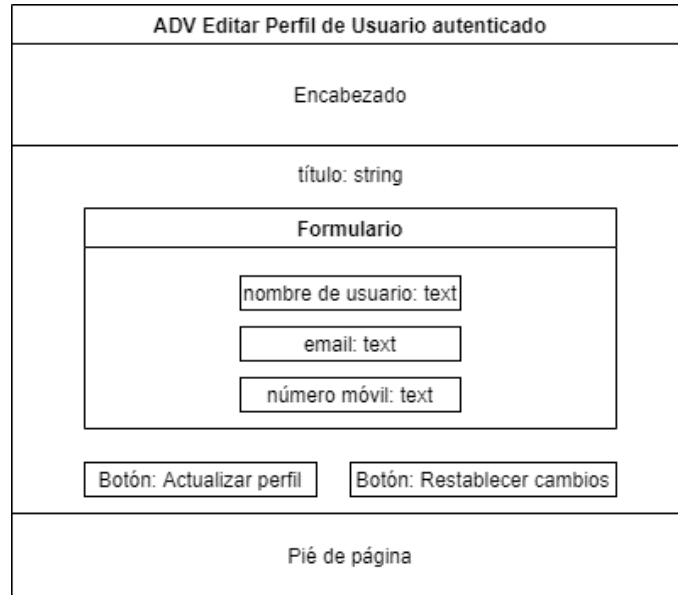


Figura 3.25 ADV - Editar perfil de usuario autenticado
Fuente: Investigador

La vista “cambiar contraseña” representa a la página que contiene el formulario para establecer una nueva contraseña de un usuario como indica la Figura 3.26.

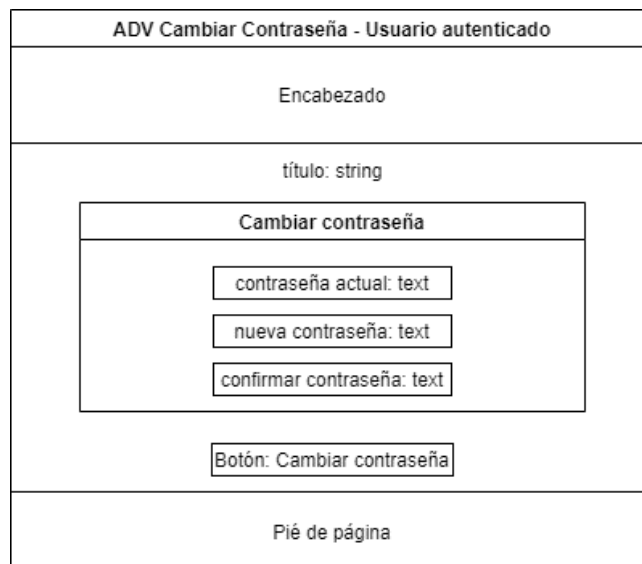


Figura 3.26 ADV - Cambiar contraseña
Fuente: Investigador

La vista “eliminar cuenta de usuario autenticado” representa la página que contiene el formulario para la eliminación de una cuenta de usuario como muestra la Figura 3.27.

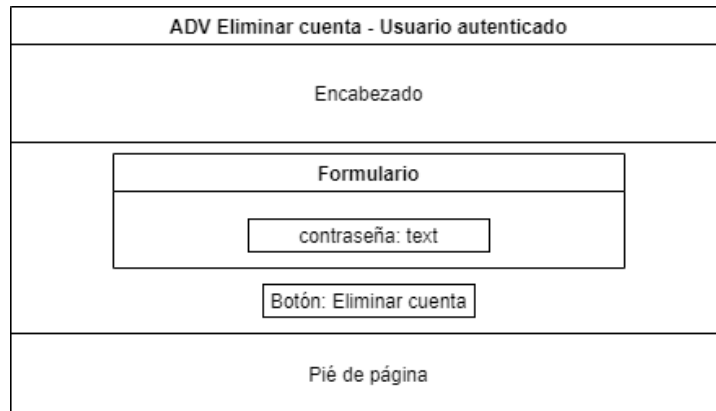


Figura 3.27 ADV - Eliminar cuenta de usuario autenticado
Fuente: Investigador

La vista “registro de empresa” representa a la página que contiene el formulario para el ingreso de los datos de una empresa como se muestra en la Figura 3.28.

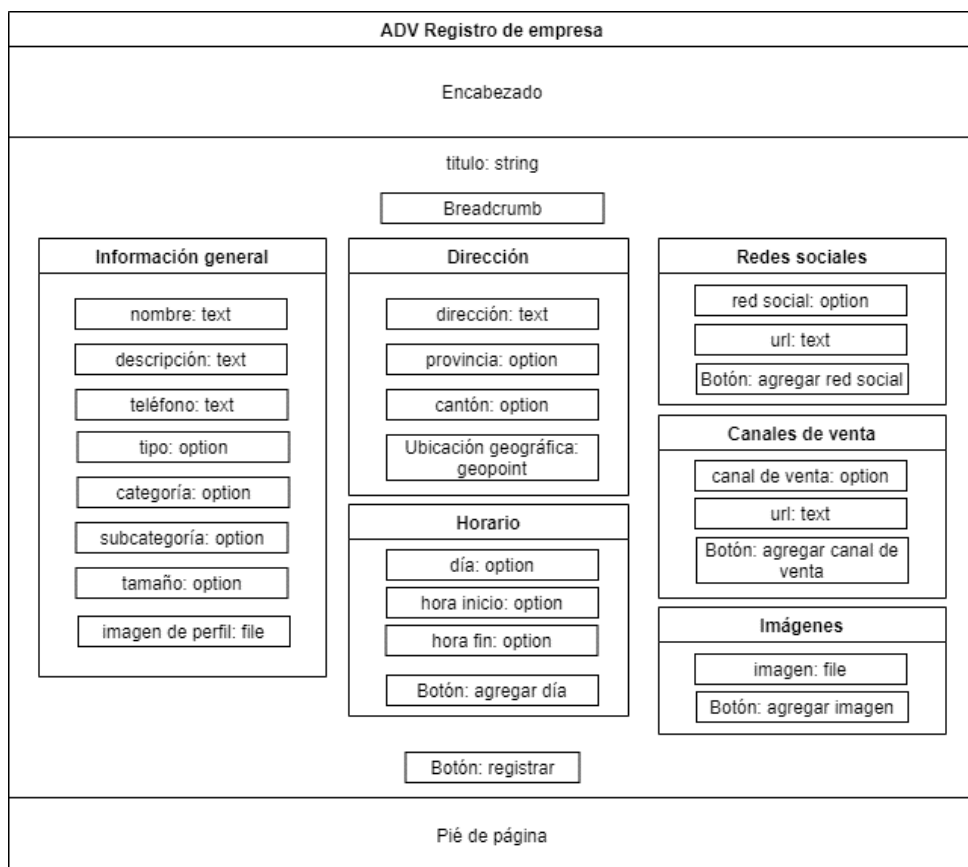


Figura 3.28 ADV - Registro de empresa
Fuente: Investigador

La vista “gestionar empresa” representa a la página que contiene la información completa de una empresa registrada como muestra la Figura 3.29.

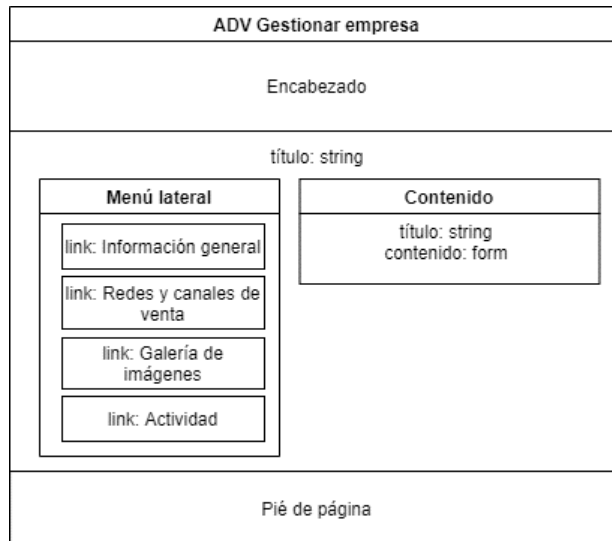


Figura 3.29 ADV - Gestionar empresa
Fuente: Investigador

La vista “editar datos de empresa” representa a la página que contiene el formulario para editar la información general de una empresa como muestra la Figura 3.30.

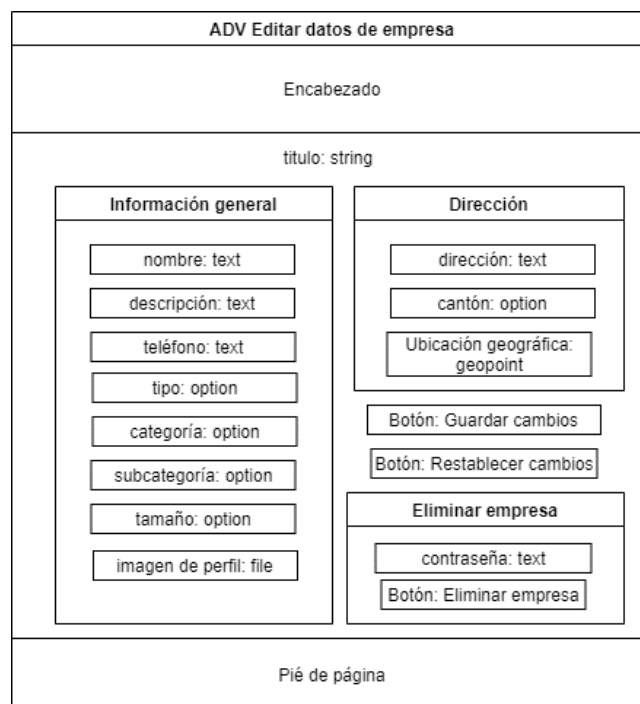


Figura 3.30 ADV - Editar datos de la empresa
Fuente: Investigador

La vista “redes y canales de venta de la empresa” representa a la página que contiene el formulario para editar las redes sociales y canales de venta de la empresa como indica la Figura 3.31.

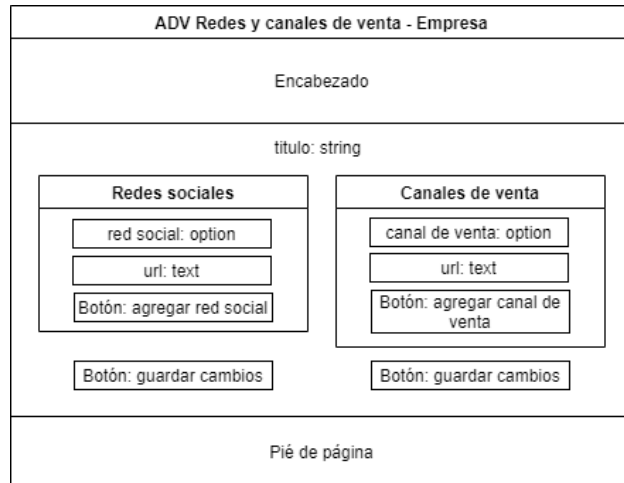


Figura 3.31 ADV - Redes y canales de venta de la empresa
Fuente: Investigador

La vista “galería de imágenes” representa a la página que contiene el formulario para agregar imágenes a la galería como muestra la Figura 3.32.

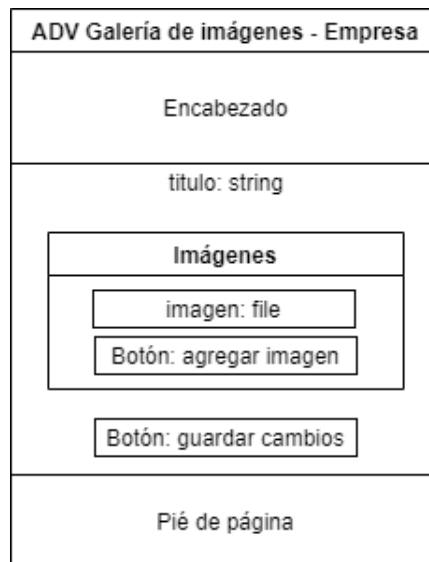


Figura 3.32 ADV - Galería de imágenes de la empresa
Fuente: Investigador

La vista “horario de la empresa” representa a la página que contiene el formulario para agregar o eliminar el horario de la empresa como muestra la Figura 3.33.

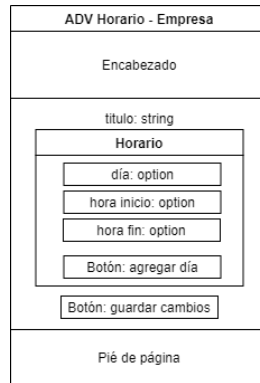


Figura 3.33 ADV - Horario de la empresa
Fuente: Investigador

La vista “actividad de empresa” representa a la página que contiene la lista de comentarios y reacciones realizadas al perfil de la empresa como se muestra en la Figura 3.34.

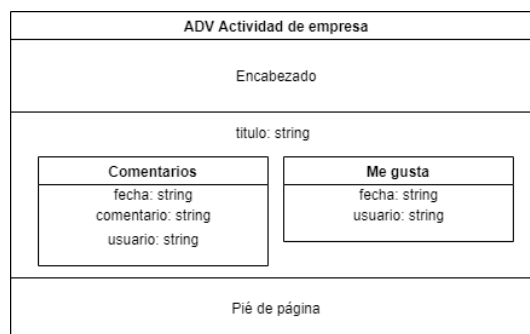


Figura 3.34 ADV - Actividad de la empresa
Fuente: Investigador

La vista “catálogo de guías” representa a la página que contiene la lista de guías disponibles en el sistema como muestra la Figura 3.35.

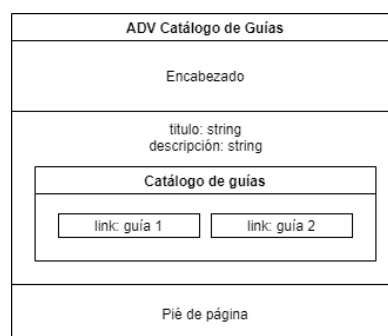


Figura 3.35 ADV – Catálogo de guías
Fuente: Investigador

La vista “guía” representa a la página que contiene la información y contenido de una guía de uso como muestra la Figura 3.36.

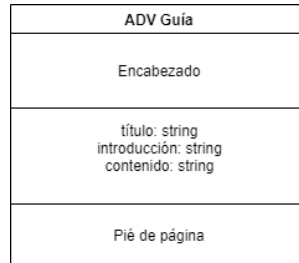


Figura 3.36 ADV – Guía
Fuente: Investigador

La vista “barra de navegación del administrador” muestra al elemento que se posiciona en la parte superior de la página para la navegación del administrador de la plataforma como muestra la Figura 3.37.

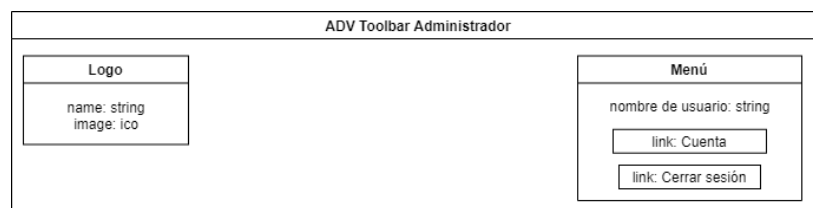


Figura 3.37 ADV - Barra de navegación del administrador
Fuente: Investigador

La vista “menú lateral del administrador” representa al elemento que se posiciona en la parte izquierda de la página para la navegación del administrador en la plataforma como muestra la Figura 3.38.

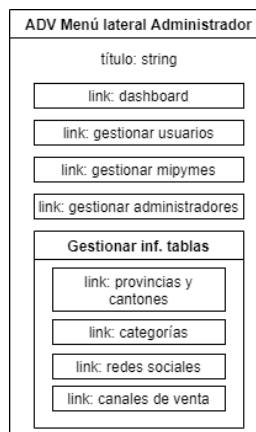


Figura 3.38 ADV - Menú lateral del administrador
Fuente: Investigador

La vista “gestionar usuarios” representa a la página que contiene la lista de usuarios registrados y formularios de gestión de datos como muestra la Figura 3.39.

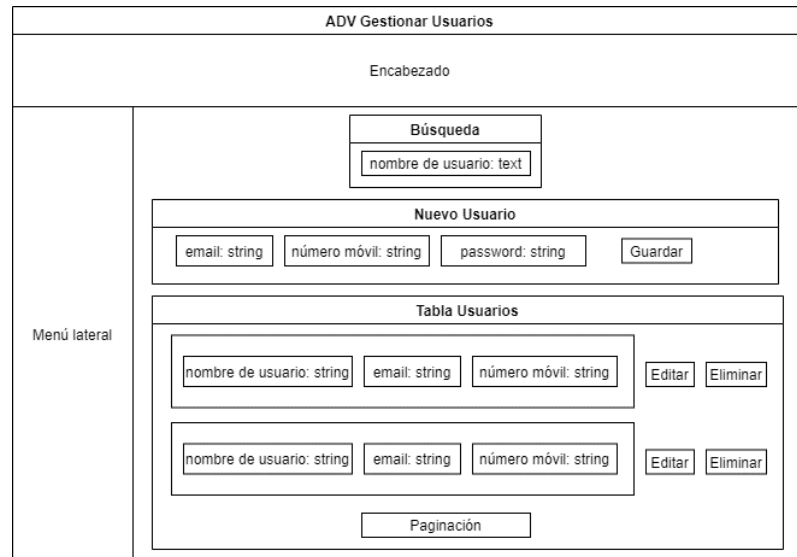


Figura 3.39 ADV - Gestionar usuarios
Fuente: Investigador

La vista “gestionar MiPymes” representa a la página que contiene la lista de empresas registradas y formularios de actualización de datos como muestra la Figura 3.40.

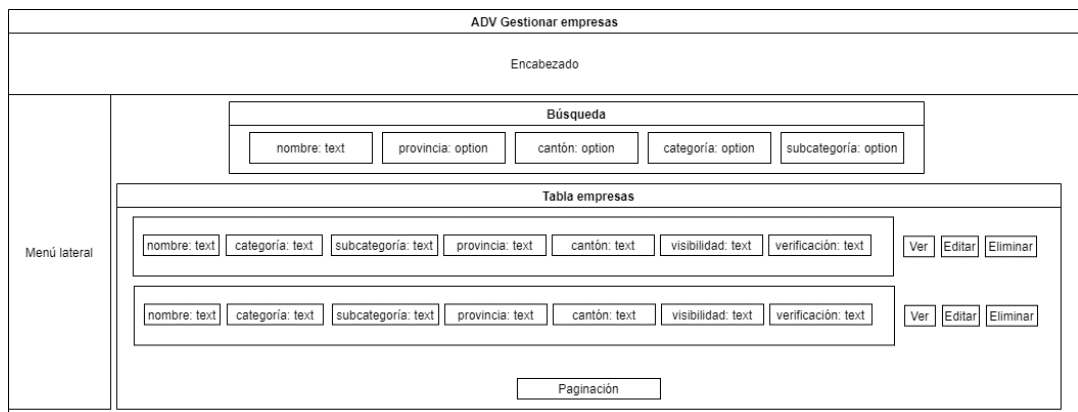


Figura 3.40 ADV - Gestionar MiPymes
Fuente: Investigador

La vista “gestionar cuenta administrativa” representa a la página que contiene formularios de actualización de la información y la contraseña del administrador como muestra la Figura 3.41.

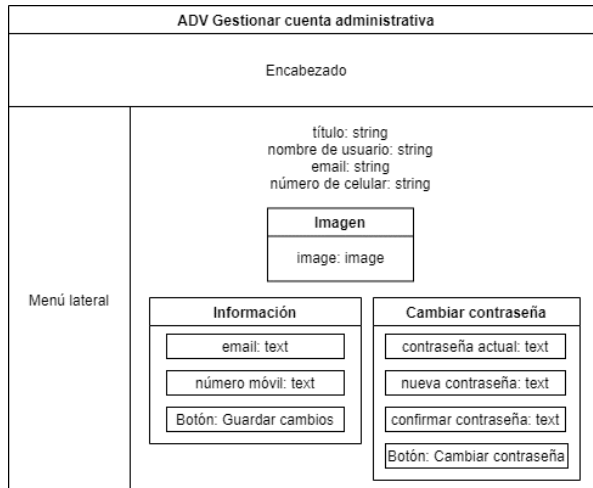


Figura 3.41 ADV - Gestionar cuenta administrativa
Fuente: Investigador

La vista “gestionar administradores” representa a la página que contiene los formularios de creación, actualización y eliminación de datos de los administradores registrados en la plataforma como muestra la Figura 3.42.

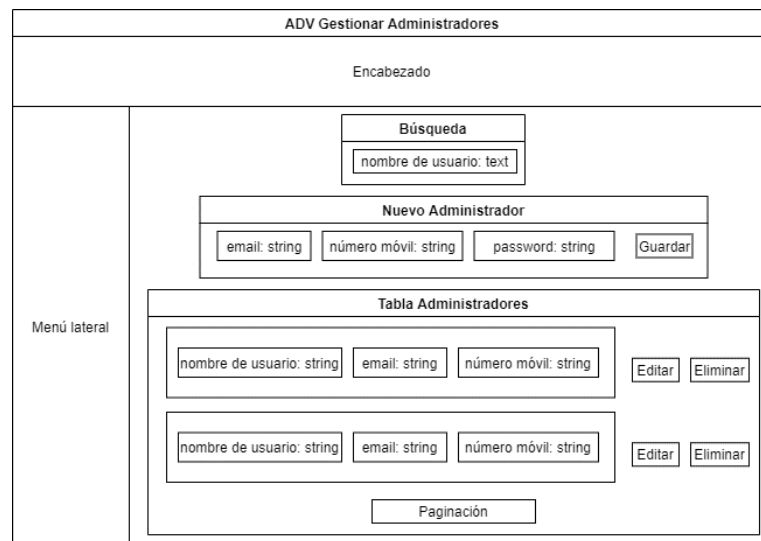


Figura 3.42 ADV - Gestionar administradores
Fuente: Investigador

La vista “gestionar información de las tablas” representa a las páginas que contienen formularios de registro, actualización y eliminación de datos sobre las tablas provincias y cantones, categorías y subcategorías, redes sociales y canales de venta como muestra la Figura 3.43.

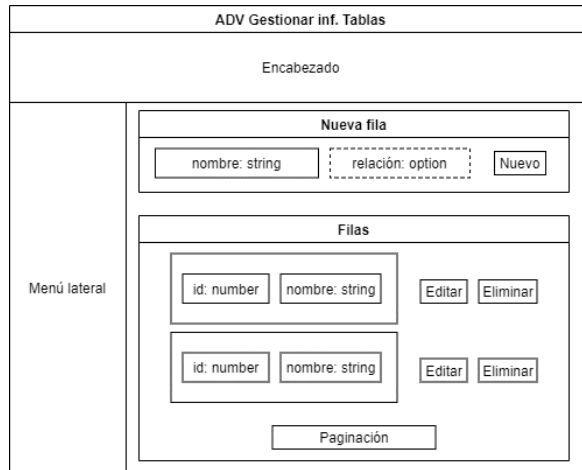


Figura 3.43 ADV - Gestionar información de las tablas
Fuente: Investigador

La vista “dashboard” representa a la página que contiene métricas estadísticas de los datos de la plataforma como muestra la Figura 3.44.

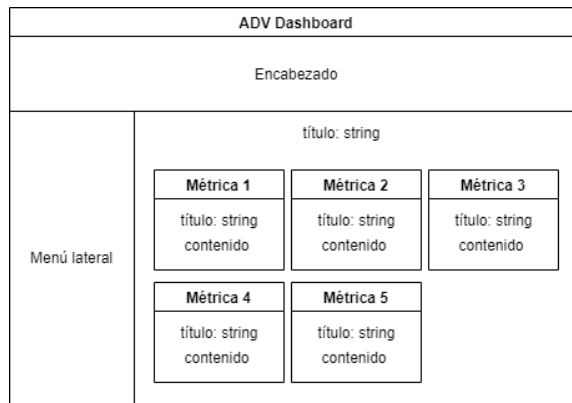


Figura 3.44 ADV – Dashboard
Fuente: Investigador

3.2.5. Quinta etapa - Implementación

Arquitectura de la aplicación

Para establecer el funcionamiento del sistema en conjunto, la arquitectura utilizada se basó en el modelo cliente-servidor, donde la aplicación del frontend del lado del cliente hace las peticiones a través de métodos HTTP a la aplicación del backend, que provee una API REST la cual recibe las peticiones, las procesa y envía una respuesta al cliente en formato JSON para su interpretación.

Puesto que se consideró el uso de software libre para el desarrollo de la plataforma, la aplicación del frontend se desarrolló en el marco de trabajo Angular en lenguaje typescript, mientras que el backend está desarrollada en el marco de trabajo Laravel en lenguaje php.

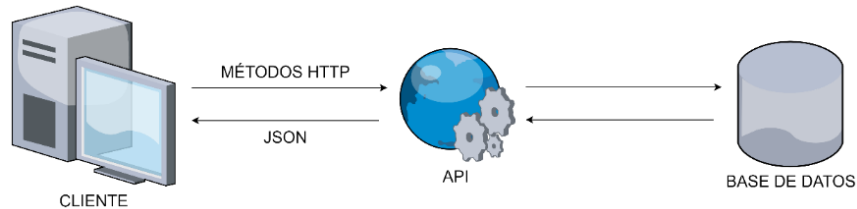


Figura 3.45 Arquitectura de la aplicación
Fuente: Investigador

Modelo físico de la base de datos

El modelo físico se elaboró a partir de las especificaciones presentadas en el modelo conceptual, el modelo contiene únicamente las tablas que se crean en la base de datos y sus relaciones.

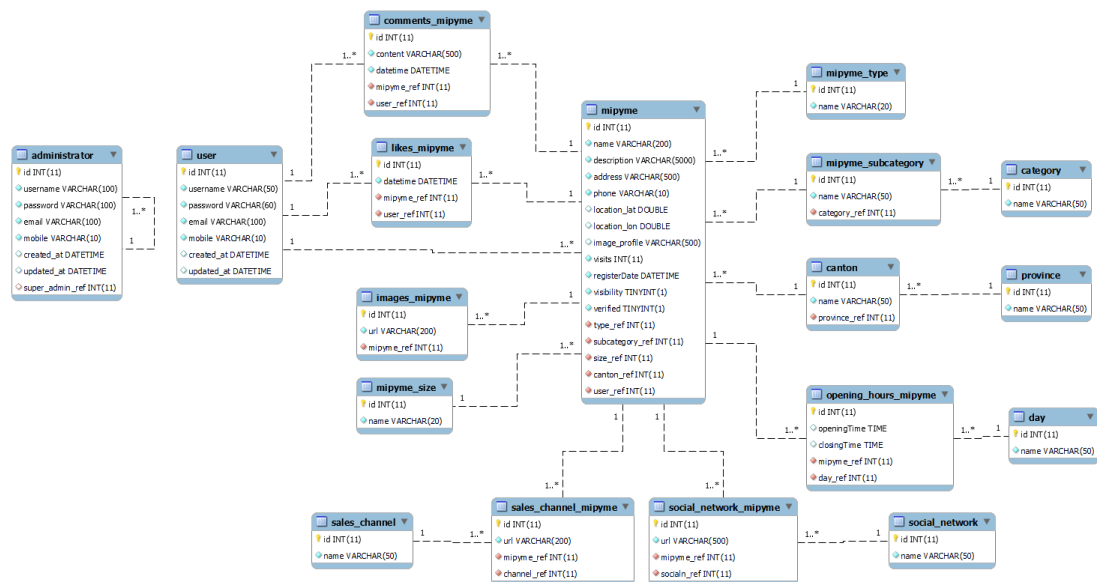


Figura 3.46 Modelo físico de la base de datos
Fuente: Investigador

Control de sesión de usuario con JWT para la API-REST

Para establecer un nivel de seguridad en el acceso a los recursos del backend se utilizó el estándar JWT (JSON Web Tokens), JWT permite crear tokens de acceso para la autenticación de usuarios, este token está firmado por la clave secreta del servidor. El proveedor de servicios de JWT se instala en laravel con el nombre “tymon/jwt-auth”, por lo cual es necesario especificar el uso de la librería dentro del archivo “config/app.php” como muestra la Figura 3.47.

```
'providers' => [
    .
    .
    /** JWT */
    Tymon\JWTAuth\Providers\LaravelServiceProvider::class,
]
'aliases' => [
    .
    .
    /** JWT */
    'JWTAuth' => Tymon\JWTAuth\Facades\JWTAuth::class,
    'JWTFactory' => Tymon\JWTAuth\Facades\JWTFactory::class,
],
```

*Figura 3.47 API REST - Configuración de JWT
Fuente: Investigador*

El proceso de autenticación de usuarios empieza desde el registro, inicio de sesión, actualización de datos y cierre de sesión, en cada aspecto se validan los datos, la sesión de usuario y se devuelve una respuesta en función de la petición.

El registro permite a los visitantes del sitio crear una cuenta de usuario en el sistema, para este proceso el método “registrar” valida los datos requeridos, comprueba que no existan registros duplicados y finalmente guarda los datos en la base de datos como se muestra en la Figura 3.48.

```

public function register(Request $request)
{
    $validator = Validator::make($request->all(), [
        'username' => 'required|string|between:2,100',
        'email' => 'required|string|email|max:100',
        'password' => 'required|string|between:8,30',
        'mobile' => 'required|string|min:10|max:10',
    ]);
    if ($validator->fails()) {
        return response()->json($validator->errors()->toJson(), 400);
    }
    if (User::where('email', '=', $request->email)->exists()) {
        return response()->json(['error' => 'Conflict'], 409);
    }
    $user = new User();
    $user->username = $request->username;
    $user->email = $request->email;
    $user->password = Hash::make($request->password);
    $user->mobile = $request->mobile;
    $user->created_at = date('Y-m-d H:i:s');
    $user->save();
    return response()->json([
        'message' => 'User successfully registered',
        'user' => $user
    ], 201);
}

```

*Figura 3.48 API REST - Registro de usuario
Fuente: Investigador*

El inicio de sesión permite validar las credenciales de un usuario para obtener acceso a los recursos privados con el token generado, como muestra la Figura 3.49.

```

public function login(Request $request)
{
    $validator = Validator::make($request->all(), [
        'email' => 'required|email',
        'password' => 'required|string',
    ]);
    if ($validator->fails()) {
        return response()->json($validator->errors(), 422);
    }
    if (!User::where('email', '=', $request->email)->exists()) {
        return response()->json(['error' => 'Email does not exist'], 404);
    }
    if (!$token = JWTAuth::attempt($validator->validated())) {
        return response()->json(['error' => 'Unauthorized'], 401);
    }
    return $this->createNewToken($token);
}

```

*Figura 3.49 API REST - Inicio de sesión
Fuente: Investigador*

Al validar las credenciales de un usuario se devuelve en formato JSON información del token y del usuario con el método “createNewToken” como se muestra en la Figura 3.50.

```
protected function createNewToken($token)
{
    return response()->json([
        'access_token' => $token,
        'token_type' => 'bearer',
        'expires_in' => JWTAuth::factory()->getTTL() * 60,
        'user' => JWTAuth::user()
    ]);
}
```

*Figura 3.50 API REST - Devolución del token
Fuente: Investigador*

El método “userProfile” devuelve la información del usuario autenticado con la función “JwtAuth:user()” que se encarga de validar el token y obtener los registros de la base de datos como muestra la Figura 3.51.

```
public function userProfile()
{
    return response()->json(JwtAuth::user());
}
```

*Figura 3.51 API REST - Obtener información del usuario
Fuente: Investigador*

El método “updateUserProfile” permite a un usuario autenticado actualizar información de su cuenta, el método genera excepciones en caso de que el token no se encuentre, no es válido o expiró como muestra la Figura 3.52.

```

public function updateUserProfile(Request $request)
{
    try {
        if (!$user = JWTAuth::parseToken()->authenticate()) {
            return response()->json(['User not found'], 404);
        }
        $validator = Validator::make($request->all(), [
            'email' => 'string|email|max:100',
            'mobile' => 'string|min:10',
        ]);
        if ($validator->fails()) {
            return response()->json($validator->errors()->toJson(), 400);
        }
        User::find($user->id)->update(
            [
                'email' => $request->email,
                'mobile' => $request->mobile,
                'updated_at' => date('Y-m-d H:i:s')
            ]
        );
        return response()->json(['message' => 'User successfully updated'],
            201);
    } catch (Exception $e) {
        if ($e instanceof \Tymon\JWTAuth\Exceptions\TokenInvalidException) {
            return response()->json(['status' => 'Token is Invalid']);
        } else if ($e instanceof
            \Tymon\JWTAuth\Exceptions\TokenExpiredException) {
            return response()->json(['status' => 'Token is Expired']);
        }
        $errorCode = $e->errorInfo[1];
        if ($errorCode == 1062) {
            return response()->json(['status' => 'Duplicate entry.'], 409);
        }
    }
}

```

*Figura 3.52 API REST - Actualización de datos del usuario
Fuente: Investigador*

El método “changePassword” permite a los usuarios autenticados cambiar la contraseña de acceso a sus cuentas, para ello se requiere la confirmación de las credenciales anteriores y las nuevas para la validación y registrar los cambios como muestra la Figura 3.53.

```

public function changePassword(Request $request)
{
    try {
        if (!$user = JWTAuth::parseToken()->authenticate()) {
            return response()->json(['User not found'], 404);
        }
        $validator = Validator::make($request->all(), [
            'email' => 'required|string|email',
            'password' => 'required|string',
            'newPassword' => 'required|string|between:8,30',
            'newPasswordConfirm' => 'required|string',
        ]);
        $auth = Validator::make($request->all(), [
            'email' => 'required|string|email',
            'password' => 'required|string',
        ]);
        if ($validator->fails()) {
            return response()->json($validator->errors()->toJson(), 400);
        }
        if (!$JWTAuth::attempt($auth->validated())) {
            return response()->json(['error' => 'Unauthorized'], 403);
        }
        User::find($user->id)->update(
            [
                'password' => Hash::make($request->newPassword),
                'updated_at' => date('Y-m-d H:i:s')
            ]
        );
        return response()->json(['message' => 'Password successfully
        changed'],201);
    } catch (Exception $e) {
        if ($e instanceof \Tymon\JWTAuth\Exceptions\TokenInvalidException) {
            return response()->json(['status' => 'Token is Invalid']);
        } else if ($e instanceof
            \Tymon\JWTAuth\Exceptions\TokenExpiredException) {
            return response()->json(['status' => 'Token is Expired']);
        }
    }
}

```

*Figura 3.53 API REST - Cambio de contraseña
Fuente: Investigador*

El método “logout” invalida el token de un usuario con sesión activa, la función “JWTAuth::invalidate()” no aplica para tokens no validos o expirados, este proceso se muestra en la Figura 3.54.

```

public function logout()
{
    JWTAuth::invalidate(JWTAuth::getToken());
    return response()->json(['message' => 'User successfully signed out'],
    200);
}

```

*Figura 3.54 API REST - Cerrar sesión
Fuente: Investigador*

El modelo del “usuario” implementa la interfaz JWTSubject para hacer uso de la definición de sus métodos, especificar el identificador del token y los reclamos personalizados como muestra la Figura 3.55.

```
class User extends Authenticatable implements JWTSubject
{
    use Notifiable;
    protected $table = 'user';
    public $timestamps = false;
    protected $primaryKey = 'id';
    protected $fillable = [
        'username', 'email', 'password', 'mobile', 'created_at', 'updated_at'
    ];
    protected $hidden = [
        'password',
    ];
    public function getJWTIdentifier()
    {
        return $this->getKey();
    }
    public function getJWTCustomClaims()
    {
        return [
            'role' => 'user',
        ];
    }
}
```

Figura 3.55 API REST - Modelo Usuario

Fuente: Investigador

Documentación de la API-REST implementada

Una API-REST provee un conjunto de recursos que se acceden mediante solicitudes por medio de puntos finales o “endpoints” como se conoce comúnmente, estos puntos finales establecen un canal de comunicación en el extremo de la API e interactúa con el cliente a través de una URL para resolver una función del recurso solicitado.

La API-REST desarrollada para la plataforma cuenta con diferentes puntos finales para acceder a sus recursos como se detalla en la tabla 3.27 a continuación.

Método	Ruta	Función	Descripción
Punto final: admin			
post	api/admin/changePassword	changePassword	Cambiar contraseña
post	api/admin/login	login	Iniciar sesión
post	api/admin/logout	logout	Cerrar sesión
get	api/admin/profile	profile	Obtener información del administrador
post	api/admin/updateProfile	updateProfile	Actualizar la información del administrador
Punto final: administrators			
get	api/administrators	index	Lista de administradores
post	api/administrators	store	Registrar administrador
put	api/administrators/changePassword/{id}	changePassword	Cambiar la contraseña de un administrador
get	api/administrators/{administrato r}	show	Obtener un administrador por id
delete	api/administrators/{administrato r}	destroy	Eliminar un administrador
put	api/administrators/{administrato r}	update	Actualizar la información del administrador
Punto final: canton			
post	api/canton	store	Registrar cantón
get	api/canton	index	Obtener lista de cantones
delete	api/canton/{canton}	destroy	Eliminar un cantón
put	api/canton/{canton}	update	Actualizar información del cantón
get	api/canton/{canton}	show	Obtener un cantón por id
Punto final: category			
post	api/category	store	Registrar categoría
get	api/category	index	Obtener lista de categorías
delete	api/category/{category}	destroy	Elimina una categoría

put	api/category/{category}	update	Actualizar información de la categoría
get	api/category/{category}	show	Obtener una categoría por id
Punto final: comment			
post	api/comment	store	Registrar comentario
get	api/comment	index	Obtener lista de comentarios
delete	api/comment/{comment}	destroy	Eliminar comentario
get	api/comment/{comment}	show	Obtener comentario por id
get	api/getCommentsByMipyme/{id}	getCommentsByMipyme	Obtener comentarios de una empresa
get	api/getCommentsByUser/{id}	getCommentsByUser	Obtener comentarios de un usuario
Punto final: day			
get	api/day	index	Obtener lista de días
Punto final: image			
post	api/image	store	Registrar imagen
get	api/image	index	Obtener lista de imágenes
delete	api/image/{image}	destroy	Eliminar imagen
Punto final: like			
post	api/like	store	Registrar reacción
delete	api/like/{like}	destroy	Eliminar reacción
get	api/likesByMipyme/{id}	likesByMipyme	Obtener reacciones de una empresa
get	api/likesByUser/{id}	likesByUser	Obtener reacciones de un usuario
get	api/countLikesByMipyme/{id}	countLikesByMipyme	Obtener número de reacciones por id de empresa
Punto final: mipyme			
post	api/mipyme	store	Registrar empresa

get	api/mipyme	index	Obtener lista de empresas
get	api/mipyme/{ mipyme }	show	Obtener empresa por id
delete	api/deleteMipyme/{ id }	deleteMipyme	Eliminar empresa con rol de usuario
delete	api/deleteMipymeAdmin/{ id }	deleteMipymeAdmin	Eliminar empresa con rol de administrador
get	api/existMipymeByUser/{ id }	existMipymeByUser	Obtener existencia de registro de empresa por parte de un usuario
get	api/filterMipymes	filterMipymes	Obtener lista de empresas mediante parámetros
get	api/getMipymeByUser/{ id }	getMipymeByUser	Obtener empresa por id de usuario
put	api/updateVisits/{ id }	updateVisits	Actualizar contador de vistas de una empresa
post	api/updateMipyme	updateMipyme	Actualizar información de una empresa
put	api/verifyMipyme/{ id }	verifyMipyme	Actualizar la autenticidad de una empresa
put	api/visibilityMipyme/{ id }	visibilityMipyme	Actualizar la visibilidad de una empresa en el catálogo
Punto final: openinghours			
post	api/openinghours	store	Registrar horario de una empresa
delete	api/openinghours/{ openinghour }	destroy	Eliminar horario de una empresa
Punto final: provincia			
post	api/provincia	store	Registrar una provincia
get	api/provincia	index	Obtener lista de provincias
delete	api/provincia/{ provincia }	destroy	Eliminar una provincia

put	api/provincia/{provincia}	update	Actualizar información de una provincia
get	api/provincia/{provincia}	show	Obtener provincia por id
Punto final: salechannel			
post	api/salechannel	store	Registrar un canal de venta
get	api/salechannel	index	Obtener lista de canales de venta
delete	api/salechannel/{salechannel}	destroy	Eliminar un canal de venta
put	api/salechannel/{salechannel}	update	Actualizar información del canal de venta
get	api/salechannel/{salechannel}	show	Obtener un canal de venta por id
Punto final: salechannelmipyme			
post	api/salechannelmipyme	store	Registrar un canal de venta para una empresa
delete	api/salechannelmipyme/{salechannelmipyme}	destroy	Eliminar un canal de venta de una empresa
Punto final: size			
get	api/size	index	Obtener tamaño de empresas
Punto final: socialnetwork			
post	api/socialnetwork	store	Registrar una red social
get	api/socialnetwork	index	Obtener una red social
delete	api/socialnetwork/{socialnetwork}	destroy	Eliminar una red social
put	api/socialnetwork/{socialnetwork}	update	Actualizar información de una red social
get	api/socialnetwork/{socialnetwork}	show	Obtener una red social
Punto final: socialnetworkmipyme			
post	api/socialnetworkmipyme	store	Registrar una red social para una empresa
delete	api/socialnetworkmipyme/{socialnetworkmipyme}	destroy	Eliminar una red social para una empresa
Punto final: subcategory			

post	api/subcategory	store	Registrar una subcategoría
get	api/subcategory	index	Obtener lista de subcategorías
delete	api/subcategory/{subcategory}	destroy	Eliminar una subcategoría
put	api/subcategory/{subcategory}	update	Actualizar información de una subcategoría
get	api/subcategory/{subcategory}	show	Obtener una subcategoría
Punto final: type			
get	api/type	index	Obtener lista de los tipos de empresas
Punto final: user			
post	api/user/login	login	Iniciar sesión
post	api/user/logout	logout	Cerrar sesión
get	api/user/profile	userProfile	Obtener información del usuario
post	api/user/register	register	Crear una cuenta de usuario
post	api/user/updateProfile	updateUserProfile	Actualizar información del usuario
post	api/user/changePassword	changePassword	Cambiar contraseña del usuario
post	api/user/deleteAccount	deleteUserAccount	Eliminar la cuenta del usuario
Punto final: users			
post	api/users	store	Registrar un usuario
get	api/users	index	Obtener lista de usuarios
delete	api/users/{user}	destroy	Eliminar un usuario
put	api/users/{user}	update	Actualizar la información de un usuario
get	api/users/{user}	show	Obtener la información de un usuario por id
put	api/users/changePassword/{id}	changePassword	Cambiar la contraseña

*Tabla 3.19 Rutas de la API-REST
Fuente: Investigador*

Laravel separa los datos, la lógica de negocio y la presentación de la aplicación de acuerdo al patrón de diseño MVC, la aplicación desarrollada se centró en la creación de modelos y controladores para el funcionamiento del backend y así delegar la presentación al frontend.

El modelo “Mipyme” se estructuró como muestra la Figura 3.56, donde se declara el nombre de la tabla, sus atributos y su clave primaria para facilitar el manejo de la información.

```
class Mipyme extends Model
{
    protected $table = 'mipyme';
    protected $primaryKey = 'id';
    public $timestamps = false;
    protected $fillable = [
        'name',
        'description',
        'address',
        'phone',
        'location_lat',
        'location_lon',
        'image_profile',
        'visits',
        'registerDate',
        'visibility',
        'verified',
        'type_ref',
        'subcategory_ref',
        'size_ref',
        'canton_ref',
        'user_ref',
    ];
}
```

*Figura 3.56 API REST - Modelo Mipyme
Fuente: Investigador*

El método “index” del controlador “MipymeController” devuelve la lista de empresas registradas en el sistema, este método es público por lo cual puede ser accedido por cualquier usuario como muestra la Figura 3.57.

```

public function index()
{
    $mipymes = DB::table('mipyme')
        ->join('mipyme_type', 'mipyme.type_ref', '=', 'mipyme_type.id')
        ->join('mipyme_subcategory', 'mipyme.subcategory_ref', '=',
            'mipyme_subcategory.id')
        ->join('category', 'category.id', '=',
            'mipyme_subcategory.category_ref')
        ->join('mipyme_size', 'mipyme.size_ref', '=', 'mipyme_size.id')
        ->join('canton', 'mipyme.canton_ref', '=', 'canton.id')
        ->join('province', 'canton.province_ref', '=', 'province.id')
        ->join('user', 'mipyme.user_ref', '=', 'user.id')
        ->select(
            'mipyme.*',
            'mipyme_type.name as type',
            'mipyme_subcategory.name as subcategory',
            'mipyme_size.name as size',
            'canton.name as canton',
            'province.name as province',
            'category.id as category_ref',
            'category.name as category',
            'user.username as user'
        )->get();
    return response()->json(['data' => $mipymes], 200);
}

```

Figura 3.57 API REST - Lista de empresas

Fuente: Investigador

El método “store” se encarga de recibir los datos de una empresa, validarlos, comprobar existencia de registros previos, almacenar la imagen de perfil en la carpeta pública y registrar los datos en la base de datos como muestra la Figura 3.58.

```

public function store(Request $request)
{
    $user = JwtAuth::user();
    if (!$user) {
        return response()->json(['error' => 'Unauthorized'], 401);
    }
    $exist = DB::table('mipyme')->select('*')
        ->where('mipyme.user_ref', $user->id)->count();
    if ($exist > 0) {
        return response()->json(['errors' => array(['message' => 'Recurso
            duplicado'])], 409);
    }
    $validator = Validator::make($request->all(), [
        'name' => 'required',
        'description' => 'required',
        'address' => 'required',
        'phone' => 'required',
        'location_lat' => 'required',
        'location_lon' => 'required',
        'image_profile' => 'required|mimes:jpeg,png,jpg|max:1024',
        'visits' => 'required',
        'visibility' => 'required',
        'verified' => 'required',
        'type_ref' => 'required',
        'subcategory_ref' => 'required',
        'size_ref' => 'required',
        'canton_ref' => 'required',
    ]);
    if ($validator->fails()) {
        return response()->json($validator->errors()->toJson(), 400);
    }
    $file = $request->file('image_profile');
    $imageName = time() . '.' . $file->extension();
    $file->move(public_path('images/' . $user->username . '/'), $imageName);
    $path = 'images/' . $user->username . '/' . $imageName;
    $mipyme = new Mipyme([
        'name' => $request['name'],
        'description' => $request['description'],
        'address' => $request['address'],
        'phone' => $request['phone'],
        'location_lat' => $request['location_lat'],
        'location_lon' => $request['location_lon'],
        'image_profile' => $path,
        'visits' => 0,
        'registerDate' => date('Y-m-d H:i:s'),
        'visibility' => 1,
        'verified' => 0,
        'type_ref' => $request['type_ref'],
        'subcategory_ref' => $request['subcategory_ref'],
        'size_ref' => $request['size_ref'],
        'canton_ref' => $request['canton_ref'],
        'user_ref' => $user->id,
    ]);
    $mipyme->save();
    return response()->json(['message' => 'Mipyme successfully registered',
        'data' => $mipyme], 201);
}

```

Figura 3.58 API REST - Registrar empresa
Fuente: Investigador

Una vez definidos todos los modelos y controladores de la aplicación, se especificó dentro del archivo “routes/api.php” las rutas de acceso por medio de URL a cada uno de los métodos dentro de sus respectivos controladores, para los usuarios de acuerdo a su rol las rutas se definieron como muestra la Figura 3.59.

```
Route::group(['middleware' => 'api','prefix' => 'user'], function () {
    Route::post('login', [JwtAuthController::class, 'login']);
    Route::post('register', [JwtAuthController::class, 'register']);
    Route::post('logout', [JwtAuthController::class, 'logout']);
    Route::get('profile', [JwtAuthController::class, 'userProfile']);
    Route::post('updateProfile', [JwtAuthController::class,
'updateUserProfile']);
    Route::post('deleteAccount', [JwtAuthController::class,
'deleteUserAccount']);
    Route::post('changePassword', [JwtAuthController::class, 'changePassword']);
});
Route::group(['middleware' => 'api','prefix' => 'admin'], function () {
    Route::post('login', [AdminJwtAuthController::class, 'login']);
    Route::post('register', [AdminJwtAuthController::class, 'register']);
    Route::post('logout', [AdminJwtAuthController::class, 'logout']);
    Route::get('profile', [AdminJwtAuthController::class, 'userProfile']);
    Route::post('updateProfile', [AdminJwtAuthController::class,
'updateAdminProfile']);
    Route::post('changePassword', [AdminJwtAuthController::class,
'changePassword']);
});
```

*Figura 3.59 Configuración de las rutas de los usuarios
Fuente: Investigador*

Las rutas para el acceso a los recursos de la aplicación se definieron de acuerdo al tipo de petición (POST, GET, PUT, DELETE) como muestra la Figura 3.60. La definición “resources” en laravel crea una referencia múltiple hacia los métodos de ciertos recursos como “index, store, show, put, destroy” en el caso de que estos existan en el controlador.

```

Route::resources([
    'administrators' => AdministratorController::class,
    'users' => UserController::class,
    'type' => TypeController::class,
    'canton' => CantonController::class,
    'provincia' => ProvinciaController::class,
    'category' => CategoryController::class,
    'subcategory' => SubcategoryController::class,
    'day' => DayController::class,
    'salechannel' => SaleChannelController::class,
    'salechannelmipyme' => SaleChannelMipymeController::class,
    'socialnetwork' => SocialNetworkController::class,
    'socialnetworkmipyme' => SocialNetworkMipymeController::class,
    'size' => SizeController::class,
    'comment' => CommentController::class,
    'like' => LikeController::class,
    'mipyme' => MipymeController::class,
    'image' => ImageController::class,
    'openinghours' => OpeningHoursController::class,
]);
Route::get('getCommentsByUser/{id}', [CommentController::class,
'getCommentsByUser']);
Route::get('getCommentsByMipyme/{id}', [CommentController::class,
'getCommentsByMipyme']);
Route::get('likesByUser/{id}', [LikeController::class, 'likesByUser']);
Route::get('likesByMipyme/{id}', [LikeController::class, 'likesByMipyme']);
Route::get('countLikesByMipyme/{id}', [LikeController::class,
'countLikesByMipyme']);
Route::get('getMipymeByUser/{id}', [MipymeController::class,
'getMipymeByUser']);
Route::get('existMipymeByUser/{id}', [MipymeController::class,
'existMipymeByUser']);
Route::put('verifyMipyme/{id}', [MipymeController::class, 'verifyMipyme']);
Route::put('visibilityMipyme/{id}', [MipymeController::class,
'visibilityMipyme']);
Route::put('updateVisits/{id}', [MipymeController::class, 'updateVisits']);
Route::put('users/changePassword/{id}', [UserController::class,
'changePassword']);
Route::put('administrators/changePassword/{id}',
[AdministratorController::class, 'changePassword']);
Route::post('filterMipymes', [MipymeController::class, 'filterMipymes']);
Route::post('updateMipyme', [MipymeController::class, 'updateMipyme']);
Route::post('deleteMipyme/{id}', [MipymeController::class, 'deleteMipyme']);
Route::delete('deleteMipymeAdmin/{id}', [MipymeController::class,
'deleteMipymeAdmin']);

```

*Figura 3.60 Configuración de las rutas de los recursos
Fuente: Investigador*

Todos los modelos utilizados en la aplicación finalmente se estructuraron como muestra la Figura 3.61.

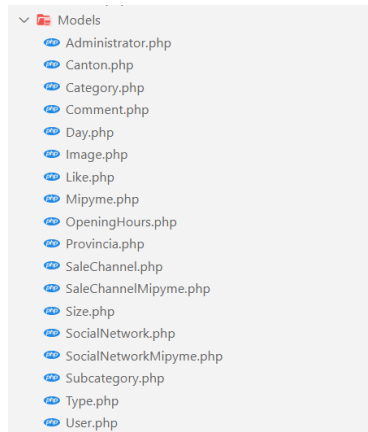


Figura 3.61 Modelos
Fuente: Investigador

Todos los controladores utilizados en la aplicación finalmente se estructuraron como muestra la Figura 3.62.

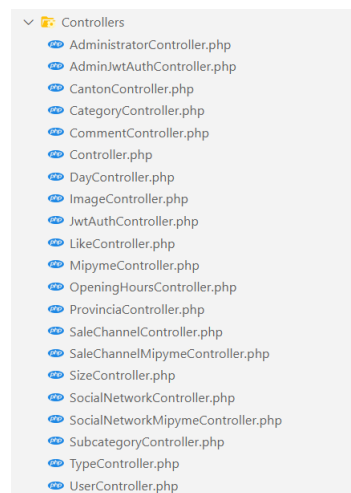


Figura 3.62 Controladores
Fuente: Investigador

Desarrollo del frontend

El frontend está vinculado a la aplicación del backend para acceder a los recursos y mostrar los datos a los usuarios, para ello se especificó la ruta de acceso a la API, esta configuración se realizó mediante el uso de variables de entorno en angular como muestra la Figura 3.63.

```

export class Constants {
  private static URL: string = 'https://promocionatuempresa.online/api/';
  public static get SERVER_URL_API(): string {
    return `${this.URL}api`;
  }
  public static get SERVER_URL(): string {
    return `${this.URL}`;
  }
}

```

*Figura 3.63 Configuración de la ruta de acceso a la API
Fuente: Investigador*

Las peticiones HTTP en ciertos casos requieren de autorización, por lo tanto, mediante un interceptor de peticiones se inspecciona y modifica las solicitudes en caso de existir algún tipo de autenticación, en este caso el token. El interceptor implementa la interfaz “HttpInterceptor” y tiene la función de incluir el token en el encabezado de las peticiones, el servidor se encarga de validarlo y devolver una respuesta en caso de ser válido. Además, es importante el uso del interceptor para el manejo de errores, para permitir a las clases de las que proviene la petición capturar los errores que se puedan generar en la transacción, la configuración del interceptor para la aplicación del frontend se muestra en la Figura 3.64.

```

intercept(
  req: HttpRequest<any>,
  next: HttpHandler
): Observable<HttpEvent<any>> {
  const token = this.authService.getAccessToken();
  if (token) {
    const cloned = req.clone({
      headers: req.headers.set('Authorization', 'Bearer ' + token),});
    return next.handle(cloned).pipe(
      catchError((error) => {return throwError(error);}));
  } else {
    return next.handle(req).pipe(
      catchError((error) => {return throwError(error);}));
  }
}

```

*Figura 3.64 Configuración de HTTP Interceptor
Fuente: Investigador*

En angular los “guards” son interfaces que se ejecutan antes de acceder a una ruta para determinar si se puede cargar o no bajo condiciones que se establezcan en la navegación de los usuarios en función de sus roles. Para la protección de las rutas se

implementó la interfaz “canActivate” a través de su método que devuelve valores de verdadero o falso para autorizar o no el acceso como se muestra en la Figura 3.65.

```
canActivate() {
  const role = this.userAuthService.getUserRole();
  if (role && role.role == 'user') {
    if (moment() < moment(this.userAuthService.getExpiration())) {
      return true;
    } else {
      this.removeCurrentUser();
      this.router.navigate(['/home/auth/login']);
      return false;
    }
  }
  this.router.navigate(['/home/auth/login']);
  return false;
}
```

*Figura 3.65 Rutas protegidas
Fuente: Investigador*

La Figura 3.66 muestra el consumo del servicio de registro de un nuevo usuario, el método requiere los datos del usuario y envía la petición a través URL.

```
register(user: User): Observable<any> {
  return this.httpClient.post(`${this.API_URL}/register`, user);
}
```

*Figura 3.66 Consumo de servicios - Registro de usuario
Fuente: Investigador*

La librería “rxjs” de reactivex permite ejecutar eventos de manera asíncrona, para el consumo de servicios la respuesta del servidor lo convierte a un observable como origen de la información a los cuales se accede mediante una suscripción. La Figura 3.67 muestra el método para el inicio de sesión que convierte la respuesta del servidor en un observable y a su vez manipula la colección de datos para guardar los datos del token y del usuario en el almacenamiento local.

```

login(user: User): Observable<any> {
    return this.httpClient
        .post<JwtResponseI>(`${this.API_URL}/login`, user)
        .pipe(
            tap((res: JwtResponseI) => {
                if (res) {
                    this.saveAccessToken(res.access_token, res.expires_in);
                    this.currentUsername.next(res.user.username);
                }
            })
        );
}

```

*Figura 3.67 Consumo de servicios - Inicio de sesión
Fuente: Investigador*

El método “onLogin” que se muestra en la Figura 3.68 utiliza la función de suscripción al observable de reactivex para la manipulación de datos en caso de devolver una respuesta valida o emitir un aviso de error en caso de que la petición no se haya completado correctamente.

```

onLogin(): void {
    this.spinnerService.show();
    this.resetErrors();
    this.user.email = this.fgLogin.get('email').value;
    this.user.password = this.fgLogin.get('password').value;
    this.userAuthService.login(this.user).subscribe(
        (res) => {
            this.hasMipyme();
        },
        (error) => {
            if (error.status == 422) {
                this.isIncomplete = true;
            } else if (error.status == 404) {
                this.emailNotRegistered = true;
            } else if (error.status == 401) {
                this.isIncorrectPassword = true;
            }
            this.spinnerService.hide();
        }
    );
}

```

*Figura 3.68 Subscribe - Inicio de sesión
Fuente: Investigador*

El método “registerUser” realiza una suscripción al observable del servicio para registrar un usuario como se muestra en la Figura 3.69.

```
registerUser() {
  var element = <HTMLInputElement>document.getElementById('register');
  element.disabled = true;
  if (this.fgRegister.valid) {
    if (
      this.fgRegister.get('password').value ===
      this.fgRegister.get('confirmPassword').value
    ) {
      this.spinnerService.show();
      this.resetErrors();
      this.user.email = this.fgRegister.get('email').value;
      this.user.mobile = this.fgRegister.get('mobile').value;
      this.user.password = this.fgRegister.get('password').value;
      this.user.username = this.fgRegister.get('email').value.toString()
        .split('@', 1)[0];
      this.userAuthService.register(this.user).subscribe(
        (result) => {
          this.userAuthService.login(this.user).subscribe((r) => {
            this.spinnerService.hide();
            this.router.navigateByUrl('/home');
          });
        },
        (error) => {
          if (error.status == 400) {
            } else if (error.status == 409) {
              this.existingAccount = true;
            }
            this.spinnerService.hide();
          }
        });
    } else {
      this.noPasswordMatch = true;
    }
  }
  element.disabled = false;
}
```

*Figura 3.69 Subscribe - Registro de usuario
Fuente: Investigador*

El método “registerMipyme” realiza una suscripción al observable del servicio para registrar una empresa, a su vez muestra un aviso y redirecciona según la respuesta obtenida como se muestra la Figura 3.70.

```

registerMipyme() {
    var btnRegistrar = <HTMLInputElement>document.getElementById('register');
    btnRegistrar.disabled = true;
    if (this.registerForm.valid) {
        this.spinnerService.show();
        this.mipyme = this.getRegisterFormValues();
        this.mipymeService.postMipyme(this.mipyme).subscribe(
            (result) => {
                this.postSocialNetworks(result.data);
                this.postSaleChannels(result.data);
                this.postOpeningHours(result.data);
                this.postImages(result.data);
                this.mipymeService.saveHasMipyme(true);
                this.spinnerService.hide();
                this.notifyService.showSuccess('Empresa registrada exitosamente.','');
                this.router.navigateByUrl('/home/user/manage-business');
            },
            (error) => {
                this.notifyService.showError(
                    'Ha ocurrido un error. Por favor intente más tarde.','');
                this.spinnerService.hide();
            }
        );
    }
    btnRegistrar.disabled = false;
}

```

*Figura 3.70 Subscribe - Registro de empresa
Fuente: Investigador*

La Figura 3.71 muestra al método “getMipymes”, el cual realiza una suscripción al observable del servicio para obtener la lista de empresas registradas, la respuesta obtenida es almacenada en un arreglo de objetos de tipo “Mipyme”. Esta lista representa al catálogo de las empresas que serán visibles para el usuario.


```

public getMipymes() {
    this.spinnerService.show();
    this.mipymesServices.getMipymes().subscribe(
        (resultado) => {
            this.mipymes = resultado.data as Array<Mipyme>;
            this.onlyVisibleMipymes();
            if (this.mipymes.length <= 0) {
                this.noResults = true;
            } else {
                this.noResults = false;
                this.mipymesFilter = this.mipymes;
            }
            this.spinnerService.hide();
        },
        (error) => {
            this.noResults = true;
            this.spinnerService.hide();
        }
    );
    return false;
}

```

*Figura 3.71 Subscribe - Lista de empresas
Fuente: Investigador*

La especificación de las rutas es un aspecto importante para definir la navegación de los usuarios en la aplicación del frontend, la navegación general se divide en 3 secciones, la ruta por defecto “home” que carga las interfaces para los usuarios sin rol de administrador y redirecciona a la página principal y a las rutas secundarias ligadas al rol del usuario, la ruta “admin” carga las interfaces para los usuarios con rol de administrador y redirecciona a las rutas secundarias ligadas al dicho rol, mientras que las rutas “404” y “**” redireccionan a la vista de error 404 en el caso de no encontrar un recurso solicitado como se muestra en la Figura 3.72.

```

const routes: Routes = [
  {
    path: '',
    children: [
      { path: '', redirectTo: 'home', pathMatch: 'full'},
      { path: 'home', component: DefaultComponent, children: DEFAULT_ROUTES},
      { path: 'admin', component: AdminLayoutComponent, children: ADMIN_ROUTES,
      },
      { path: '404', component: NotFoundComponent },
      { path: '**', redirectTo: '404' },
    ],
  },
];

```

*Figura 3.72 Rutas de navegación del frontend
Fuente: Investigador*

La Figura 3.73 muestra las rutas definidas para la navegación del usuario, las rutas “user” y “guides” y sus rutas secundarias están protegidas con el guard “AuthGuard”.

```

export const DEFAULT_ROUTES: Routes = [
  { path: '', component: MipymesListComponent },
  { path: 'about-us', component: AboutUsComponent },
  { path: 'detalle/:id', component: MipymeDetalleComponent },
  {
    path: 'auth',
    loadChildren: () =>
      import('src/app/components/auth/auth.module').then((m) => m.AuthModule),
  },
  {
    path: 'user',
    loadChildren: () =>
      import('src/app/components/user/user.module').then((m) => m.UserModule),
    canActivate: [AuthGuard],
  },
  {
    path: 'guides',
    loadChildren: () =>
      import('src/app/components/guides/guides.module').then((m) => m.GuidesModule
    ),canActivate: [AuthGuard],
  },
  { path: '404', component: NotFoundComponent },
  { path: '**', redirectTo: '404' },
];

```

*Figura 3.73 Rutas de navegación del usuario
Fuente: Investigador*

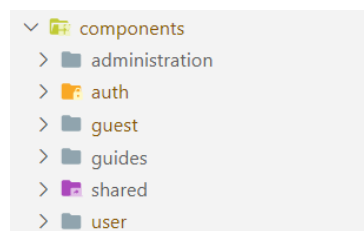
La Figura 3.74 muestra las rutas definidas para la navegación del administrador, la ruta “dashboard” y sus rutas secundarias están protegidas con el guard “AdminGuard”.

```
export const ADMIN_ROUTES: Routes = [
  { path: '', redirectTo: 'admin-login', pathMatch: 'full' },
  {
    path: 'dashboard',
    loadChildren: () =>
      import('src/app/components/administration/admin.module').then(
        (m) => m.AdminModule ),
    canActivate: [AdminGuard],
  },
  { path: 'admin-login', component: AdminLoginComponent },
  { path: '404', component: NotFoundComponent },
  { path: '**', redirectTo: '404' },
];
```

*Figura 3.74 Rutas de navegación del administrador
Fuente: Investigador*

La estructura de la aplicación del frontend está dividida por módulos, cada módulo es independiente de otro, este paradigma de programación se aplicó con la finalidad de hacer el programa más legible y manejable al momento del desarrollo y posteriormente ampliación de la funcionalidad.

Las páginas en angular están representadas por componentes, cada componente está agrupado de acuerdo a su función dentro de la aplicación como se muestra en la Figura 3.75.



*Figura 3.75 Componentes del frontend
Fuente: Investigador*

El módulo “core” representa a los elementos y servicios que se utilizan a nivel de toda la aplicación como muestra la Figura 3.76.

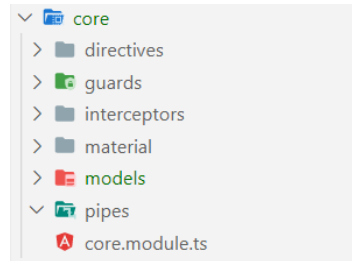


Figura 3.76 Núcleo del frontend
Fuente: Investigador

Los módulos “admin-layout” y “default” definen la estructura de diseño para los distintos usuarios y sus rutas respectivamente como muestra la Figura 3.77.

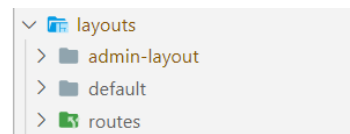


Figura 3.77 Estructura de diseño de los usuarios
Fuente: Investigador

La Figura 3.78 muestra la agrupación de los servicios de acuerdo a su función y finalidad, esto permite separar la lógica al momento de acceder a los recursos de la API.

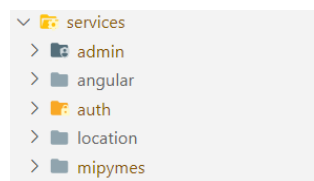


Figura 3.78 Servicios
Fuente: Investigador

3.2.6. Pruebas

Pruebas de funcionalidad

- **Crear cuenta de usuario.** La página de registro contiene un formulario que permite el ingreso de los datos requeridos para la creación de la cuenta como se

muestra en la Figura 3.79, cuando el registro es completado satisfactoriamente el usuario es redireccionado a la pantalla principal con funciones habilitadas para usuarios autenticados.

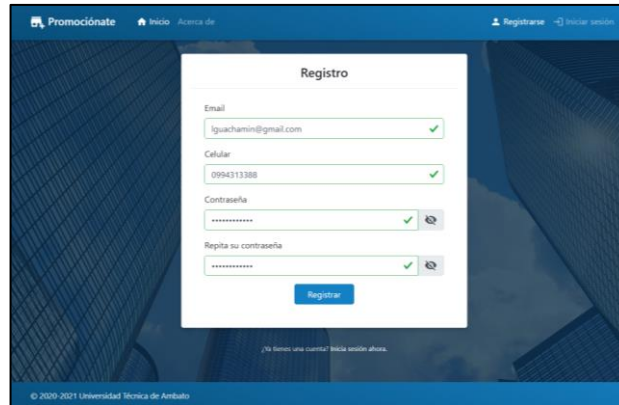


Figura 3.79 Prueba - Pantalla de "Registro"
Fuente: Investigador

- **Inicio de sesión.** Los usuarios registrados en la plataforma llenan el formulario con sus credenciales en la pantalla de inicio de sesión, si los datos son correctos se les permite el ingreso caso contrario se muestra un aviso debajo del campo que no sea válido, si el usuario no tiene una cuenta se le da la opción para crear una como se muestra en la Figura 3.80. Para usuarios con sesiones activas el ingreso es de forma automática a través del token almacenado en el navegador.

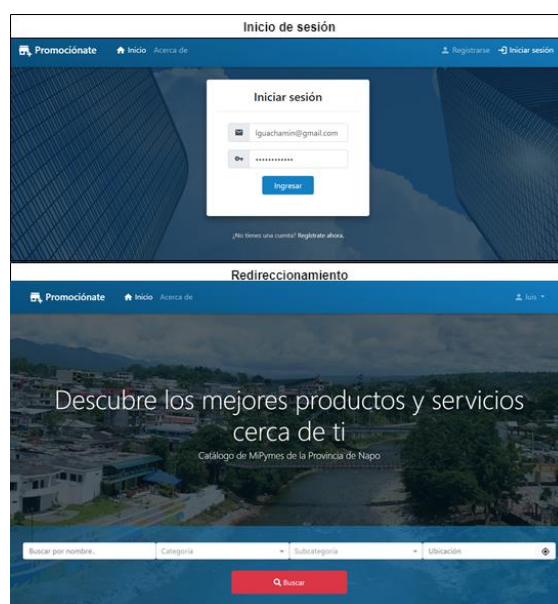


Figura 3.80 Prueba - Pantalla de "Inicio de sesión"
Fuente: Investigador

- **Consultar y filtrar el catálogo.** La pantalla principal de la plataforma contiene el catálogo completo de las empresas registradas, los usuarios pueden modificar los parámetros que se ajusten a sus preferencias y filtrar el catálogo.

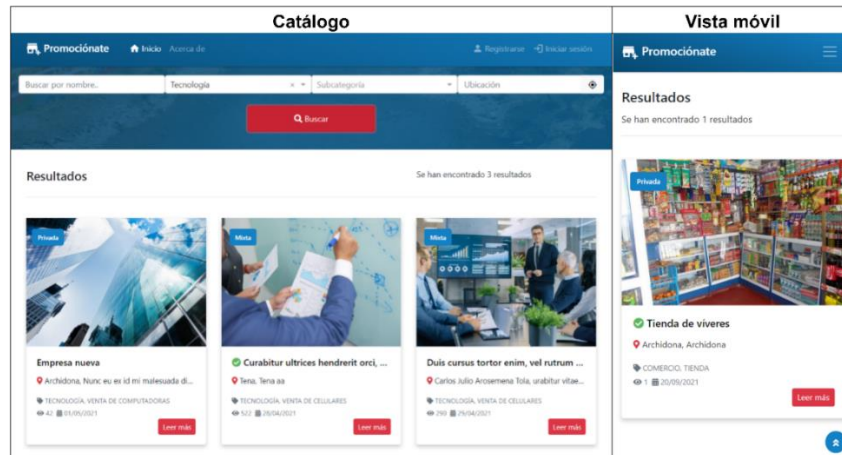


Figura 3.81 Prueba - Pantalla de "Catálogo"
Fuente: Investigador

- **Consultar el perfil de una empresa.** Para obtener más información de una empresa el usuario puede acceder al detalle completo al visitar su perfil, en él va a encontrar la información distribuida en la interfaz de tal manera que pueda conocer más sobre sus productos o servicios.

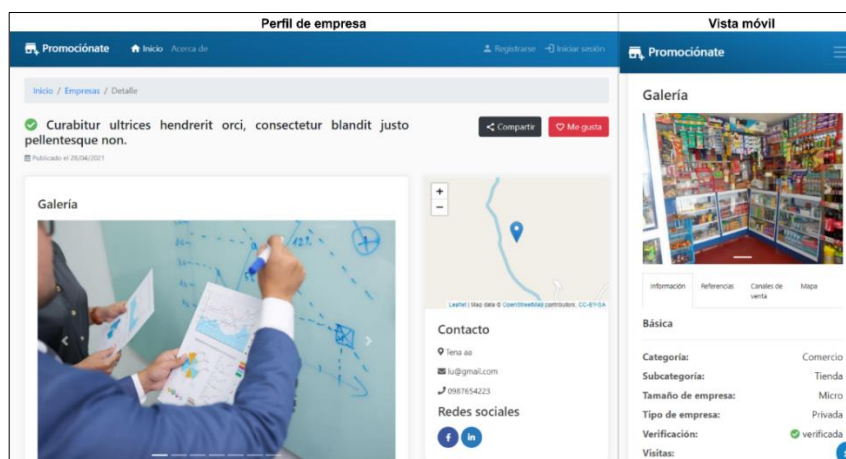


Figura 3.82 Prueba - Pantalla de "Perfil de empresa"
Fuente: Investigador

- **Registrar una empresa.** Un usuario autenticado tiene la posibilidad de registrar una empresa, misma que será visible en el catálogo cuando el registro se haya completado exitosamente, el formulario es visible únicamente cuando el usuario

no tenga un perfil de empresa registrado con anterioridad. El proceso de registro se evidencia en la Figura 3.83.

Figura 3.83 Prueba - Pantalla de "Registro de una empresa"
Fuente: Investigador

- **Gestionar la información de la empresa.** La información de una empresa puede ser modificada por el usuario, las actualizaciones como la información general, redes sociales, canales de venta, galería de imágenes, horario de atención y la actividad se reflejan instantáneamente para visualización del público.

Figura 3.84 Prueba - Pantalla de "Gestión de la información de una empresa"
Fuente: Investigador

- **Gestionar la cuenta de usuario.** Los usuarios registrados tienen la posibilidad de gestionar su información y configuración de su cuenta desde la pantalla de “perfil de usuario”.

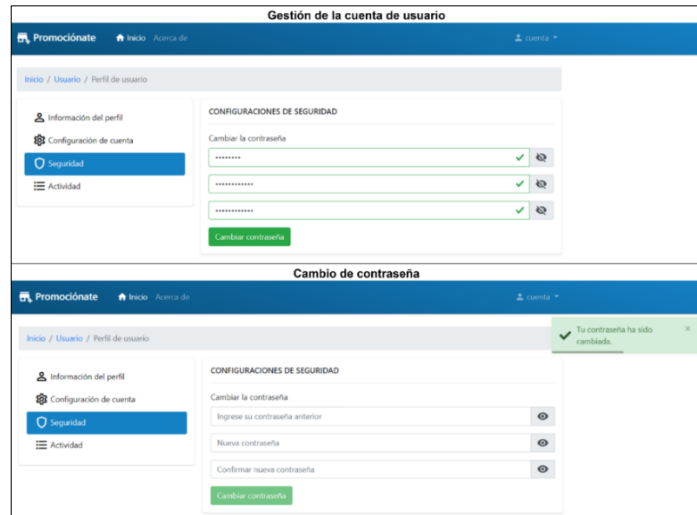


Figura 3.85 Prueba - Pantalla de "Perfil de usuario"
Fuente: Investigador

- **Añadir comentario.** El formulario de la Figura 3.86 permite a los usuarios autenticados realizar comentarios sobre el perfil de una empresa.

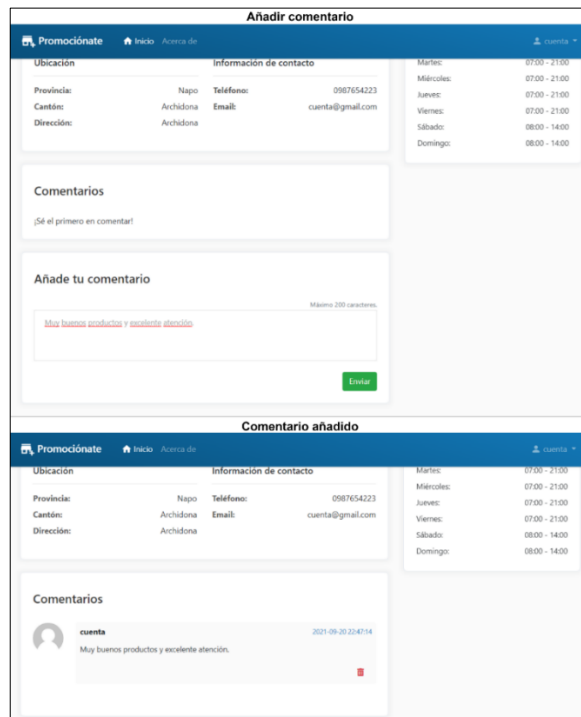


Figura 3.86 Prueba - Añadir comentario
Fuente: Investigador

- **Reaccionar sobre el perfil de una empresa.** El botón “Me gusta” permite al usuario autenticado reaccionar sobre perfil de una empresa como se aprecia en la Figura 3.87.



Figura 3.87 Prueba - Reacción sobre el perfil de una empresa
Fuente: Investigador

- **Gestionar usuarios.** La sección “Usuarios” del panel administrativo permite al administrador gestionar la información básica de los usuarios registrados como se muestra en la Figura 3.88.

The screenshot displays the 'Administrar Usuarios' page. It includes a search bar for user names and a table with the following columns: #, Usuario, Email, Teléfono, Fecha de registro, Última actualización, and Acciones. The table lists 8 users with their respective details.

#	Usuario	Email	Teléfono	Fecha de registro	Última actualización	Acciones
4	userancie	pablo@gmail.com	0800607872		2021-05-23 21:45:33	[Edit] [Delete]
5	luis	luj@gmail.com	0987654321	2021-02-01 07:41:20	2021-04-15 14:38:47	[Edit] [Delete]
7	cuenta	cuenta@gmail.com	0987654329	2021-02-25 14:25:41	2021-08-20 22:44:37	[Edit] [Delete]
9	user	user@gmail.com	0987654324	2021-02-26 16:09:16		[Edit] [Delete]
10	user2	user2@gmail.com	0987654365	2021-02-28 16:14:49	2021-03-29 03:04:45	[Edit] [Delete]
11	cuenta1	cuenta1@gmail.com	0987654567	2021-05-21 14:08:27		[Edit] [Delete]
16	cuenta2	cuenta2@gmail.com	097654564	2021-05-21 13:38:49	2021-05-22 16:56:32	[Edit] [Delete]

Figura 3.88 Prueba - Pantalla de "Gestión de usuarios"
Fuente: Investigador

- **Gestionar tablas.** El o los administradores de la plataforma gestionan el contenido de las tablas principales del sistema las cuales sirven para las consultas de los usuarios. El proceso de la gestión de esta información se puede observar en la Figura 3.89.

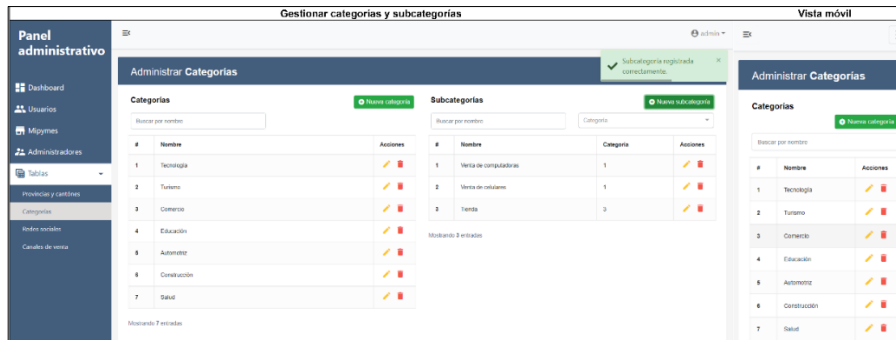


Figura 3.89 Prueba - Pantalla de "Gestión de tablas"
Fuente: Investigador

- Gestionar empresas.** El o los administradores de la plataforma gestionan la información de las empresas registradas con el fin de controlar el contenido que se comparte al público, además pueden ocultar o presentar una empresa en el catálogo y verificar que la información compartida sea verídica como se muestra en la Figura 3.90.

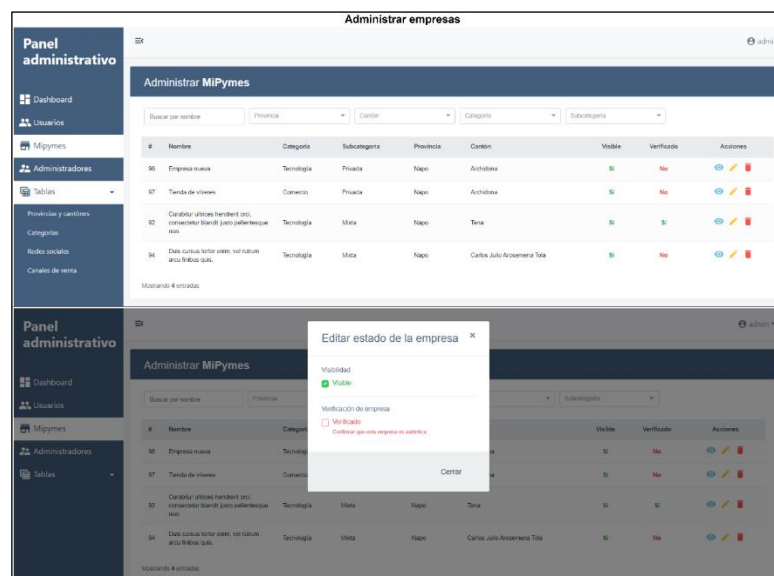


Figura 3.90 Prueba - Pantalla de "Gestión de empresas"
Fuente: Investigador

Gestionar administradores. Un administrador puede agregar más usuarios con ese rol, sin embargo, estos nuevos usuarios no tienen habilitada la función para agregar más administradores, su información se gestiona desde el panel de administración como se muestra en la Figura 3.91.

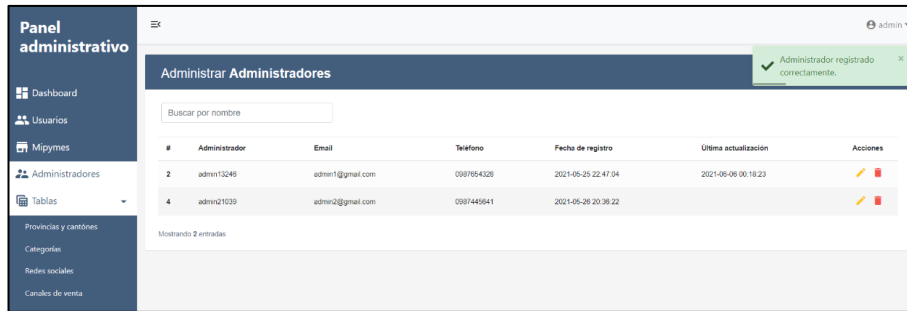


Figura 3.91 Prueba - Pantalla de "Gestión de administradores"
Fuente: Investigador

Gestionar cuenta administrativa. Un administrador puede gestionar su propia información relacionada a sus credenciales de acceso al panel administrativo como se muestra en la Figura 3.92.

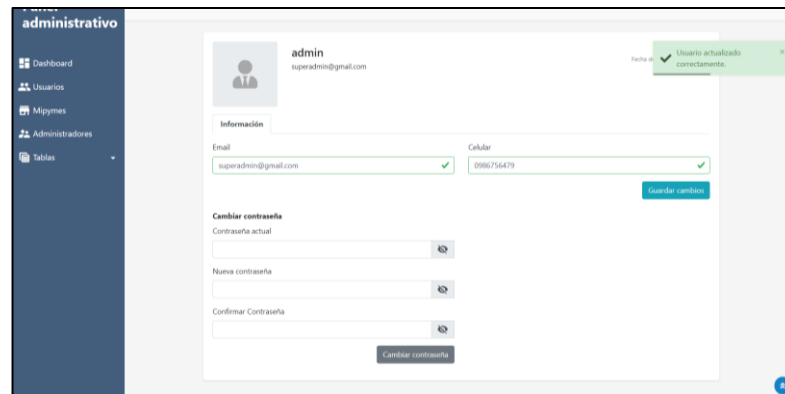


Figura 3.92 Prueba - Pantalla de "Gestión de cuenta administrativa"
Fuente: Investigador

Guías. Los usuarios autenticados tienen la posibilidad de consultar el catálogo de guías disponibles en la plataforma como se muestran en las Figura 3.93 y Figura 3.94.

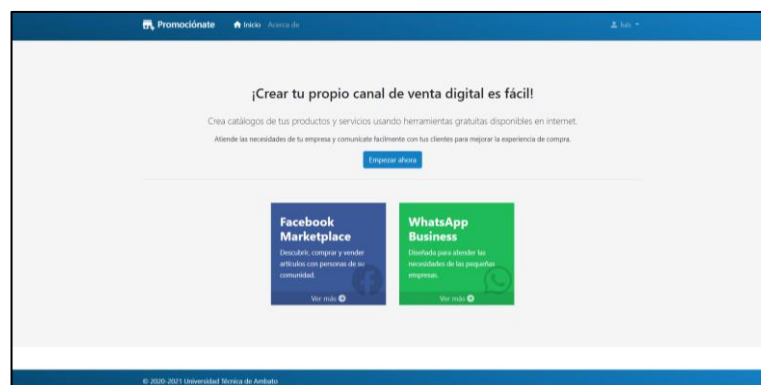


Figura 3.93 Prueba - Pantalla de "Catálogo de guías"
Fuente: Investigador



Figura 3.94 Prueba - Pantalla de "Visualización de una guía"
Fuente: Investigador

Acerca de. Esta pantalla muestra la información sobre la plataforma y los datos de contacto evidenciada en la Figura 3.95.



Figura 3.95 Prueba - Pantalla de "Acerca de"
Fuente: Investigador

Pruebas de estrés

Las pruebas de estrés se realizaron con la herramienta de evaluación benchmarking del servidor apache (ab), el procedimiento se destinó a la aplicación implementada en el servicio de alojamiento (Hostinger) con y sin certificado SSL instalado para evaluar el tiempo de respuesta con un número de solicitudes de 100 y concurrencia de 10 para la primera prueba como se muestra en la Figura 3.96.

<pre> Server Software: LiteSpeed Server Hostname: promocionatuempresa.online Server Port: 80 Document Path: / Document Length: 707 bytes Concurrency Level: 10 Time taken for tests: 15.589 seconds Complete requests: 100 Failed requests: 0 Non-2xx responses: 100 Total transferred: 90900 bytes HTML transferred: 70700 bytes Requests per second: 6.41 [#/sec] (mean) Time per request: 155.855 [ms] (mean) Time per request: 155.886 [ms] (mean, across all concurrent requests) Transfer rate: 5.69 [kbytes/sec] received Connection Times (ms) min mean[+/-sd] median max Connect: 118 169 24.1 154 278 Processing: 281 1204 243.9 1320 1666 Waiting: 234 915 349.8 921 1666 Total: 414 1434 248.0 1479 1818 Percentage of the requests served within a certain time (ms) 50% 1479 66% 1519 75% 1560 80% 1599 90% 1630 95% 1677 98% 1749 99% 1818 100% 1818 (longest request) </pre>	<pre> Server Software: LiteSpeed Server Hostname: promocionatuempresa.online Server Port: 443 SSL/TLS Protocol: TLSv1.2,ECN,PSK,RSA,AE128,GCM,SHA256,4096,128 Server Temp Key: X25519 253 bits TLS Server Name: promocionatuempresa.online Document Path: / Document Length: 1971 bytes Concurrency Level: 10 Time taken for tests: 19.430 seconds Complete requests: 100 Failed requests: 0 Total transferred: 222200 bytes HTML transferred: 197100 bytes Requests per second: 5.15 [#/sec] (mean) Time per request: 374.035 [ms] (mean) Time per request: 374.104 [ms] (mean, across all concurrent requests) Transfer rate: 11.17 [kbytes/sec] received Connection Times (ms) min mean[+/-sd] median max Connect: 411 1105 401.2 1095 1910 Processing: 123 670 375.9 655 1867 Waiting: 121 392 272.0 290 1270 Total: 1040 1775 325.6 1750 2823 Percentage of the requests served within a certain time (ms) 50% 1758 66% 1884 75% 1980 80% 2017 90% 2222 95% 2377 98% 2678 99% 2823 100% 2823 (longest request) </pre>
--	--

Figura 3.96 Resultado - Prueba de estrés 1 a "promocionatuempresa.online"
Fuente: Investigador

En esta primera prueba se pudo observar que el tiempo de respuesta para completar la evaluación en ambos casos tiene una pequeña variación de 3,8 segundos aproximadamente, mientras que las solicitudes se completaron al 100 % sin ningún problema y dentro de un tiempo de respuesta esperado. A continuación, la segunda prueba se realizó con un número de 1000 solicitudes y un nivel de concurrencia de 10 bajo las mismas condiciones de la prueba anterior como se muestra en la Figura 3.97.

<pre> Server Software: LiteSpeed Server Hostname: promocionatuempresa.online Server Port: 80 Document Path: / Document Length: 707 bytes Concurrency Level: 10 Time taken for tests: 188.053 seconds Complete requests: 1000 Failed requests: 0 Non-2xx responses: 1000 Total transferred: 909000 bytes HTML transferred: 707000 bytes Requests per second: 5.32 [#/sec] (mean) Time per request: 188.053 [ms] (mean) Time per request: 188.053 [ms] (mean, across all concurrent requests) Transfer rate: 4.72 [kbytes/sec] received Connection Times (ms) min mean[+/-sd] median max Connect: 110 187 72.8 171 1265 Processing: 122 1684 414.2 1618 4384 Waiting: 122 1097 519.5 1079 4384 Total: 282 1870 437.6 1800 4589 Percentage of the requests served within a certain time (ms) 50% 1800 66% 1978 75% 2052 80% 2094 90% 2280 95% 2527 98% 3122 99% 3561 100% 4589 (longest request) </pre>	<pre> Server Software: LiteSpeed Server Hostname: promocionatuempresa.online Server Port: 443 SSL/TLS Protocol: TLSv1.2,ECN,PSK,RSA,AE128,GCM,SHA256,4096,128 Server Temp Key: X25519 253 bits TLS Server Name: promocionatuempresa.online Document Path: / Document Length: 1971 bytes Concurrency Level: 10 Time taken for tests: 175.588 seconds Complete requests: 1000 Failed requests: 0 Total transferred: 2222000 bytes HTML transferred: 1971000 bytes Requests per second: 5.70 [#/sec] (mean) Time per request: 300.000 [ms] (mean) Time per request: 300.000 [ms] (mean, across all concurrent requests) Transfer rate: 12.36 [kbytes/sec] received Connection Times (ms) min mean[+/-sd] median max Connect: 384 1073 338.6 1046 2252 Processing: 110 662 318.3 651 1687 Waiting: 110 424 241.4 400 1541 Total: 598 1735 310.9 1721 2876 Percentage of the requests served within a certain time (ms) 50% 1721 66% 1872 75% 1980 80% 2037 90% 2163 95% 2237 98% 2306 99% 2405 100% 2876 (longest request) </pre>
--	---

Figura 3.97 Resultado - Prueba de estrés 2 a "promocionatuempresa.online"
Fuente: Investigador

Los resultados de la segunda prueba muestran que se completaron el 100% de las solicitudes enviadas con un total de cero errores en ambas situaciones, al igual que la prueba anterior existe una leve variación de tiempo. En cuanto a los resultados generales de las pruebas se puede observar que la aplicación implementada responde acorde a lo esperado con tiempos óptimos para su correcto desempeño.

Evaluación heurística con los principios de Nielsen

Con el objetivo de evaluar la usabilidad del sistema, el cuestionario se modeló basado en los diez principios heurísticos de Nielsen con el método de calificación detallado en el Anexo 1. El cuestionario se aplicó a 3 expertos en el área y se obtuvo los siguientes resultados:

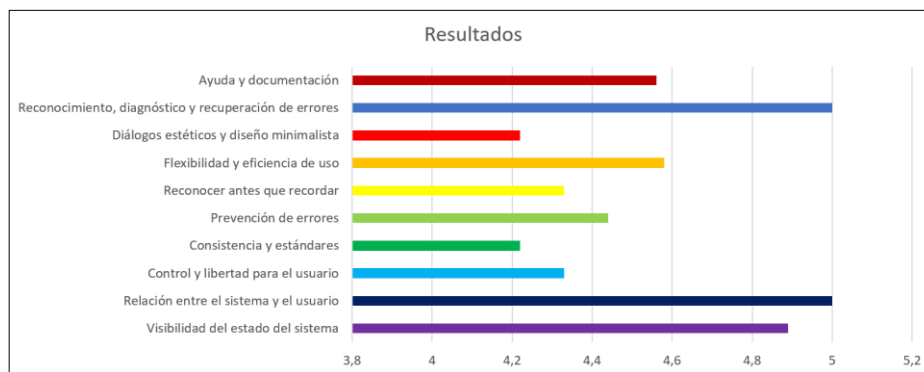


Figura 3.98 Resultados de la evaluación heurística
Fuente: Investigador

Los resultados indican que el promedio de la puntuación otorgada por los evaluadores para todas las preguntas de cada principio es de 4,56/5 lo que corresponde a un porcentaje de 91,14% de usabilidad y demuestra que el sistema cumple con la mayoría de los estándares de usabilidad web basados en los principios heurísticos de Nielsen.

3.2.7. Despliegue de la plataforma web

Tanto el backend y el frontend se subieron al servicio de alojamiento Hostinger el cual ofrece precios económicos para nuevos usuarios. Para publicar la aplicación de laravel es necesario configurar las variables de entorno en modo producción y especificar la ruta de la aplicación con el nombre de dominio comprado como se muestra en la Figura 3.99.

```
APP_NAME=Laravel
APP_ENV=production
APP_KEY=*****
APP_DEBUG=false
APP_URL=https://promocionatuempresa.online/api
```

Figura 3.99 Configuración del archivo ".env" para producción
Fuente: Investigador

Por seguridad se deben ocultar las carpetas de la aplicación del servidor mediante el archivo “.htaccess” como se muestra en la Figura 3.100.

```
<IfModule mod_alias.c>
  RedirectMatch 301 laravel/.*/error/?
</IfModule>
```

Figura 3.100 Configuración del archivo ".htaccess" para el backend
Fuente: Investigador

La aplicación de angular se exporta con el comando “ng build --prod” lo que genera una carpeta con las librerías optimizadas de la aplicación final. Además, se configuró el archivo “.htaccess” para redireccionar el tráfico al protocolo https y para configurar las rutas de navegación de angular como se muestra la Figura 3.101.

```
<IfModule mod_rewrite.c>
  RewriteEngine On
  # Redirection to HTTPS:
  RewriteCond %{HTTPS} !on
  RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
  # Redirection of requests to index.html
  RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -s [OR]
  RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -f [OR]
  RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -d
  RewriteRule ^.*$ - [NC,L]
  # Redirect all non-file routes to index.html
  RewriteRule ^(?!.*\.).*$ index.html [NC,L]
</IfModule>
```

Figura 3.101 Configuración del archivo ".htaccess" para el frontend
Fuente: Investigador

La aplicación puede ser accedida desde cualquier navegador que soporte HTML y JavaScript a través del dominio “https://promocionatuempresa.online/”, la Figura 3.102 muestra el acceso a la aplicación a través del navegador Google Chrome.



Figura 3.102 Plataforma implementada vista desde el navegador web
Fuente: Investigador

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

Los objetivos planteados para la ejecución del proyecto fueron cumplidos de manera satisfactoria, lo cual permite concluir que:

- El desarrollo e implementación de la plataforma para ayudar a las MiPymes afectadas en sus ventas se ejecutó con éxito al aplicar la metodología OOHDM conforme a las características que implica la construcción de una aplicación web de este tipo.
- Conocer la opinión de los propietarios de las empresas en el estudio del estado actual de las MiPymes de la Provincia de Napo mediante la recolección de información permitió identificar la viabilidad y factibilidad del proyecto, además se pudo identificar los requerimientos para el desarrollo del sistema.
- La metodología OOHDM proporciona un modelado de objetos muy fuerte y su uso en conjunto con UML facilitó la documentación de cada fase del desarrollo de la plataforma lo cual agilizó el proceso de construcción de las interfaces de usuario y el funcionamiento en general.
- El despliegue de la plataforma web en internet ayudará a los propietarios de negocios a promocionar su establecimiento, productos y servicios de forma fácil, directa y sin costo dentro de un entorno seguro. Este proceso se completó con éxito al superar las pruebas correspondientes al funcionamiento, carga de trabajo y usabilidad del sistema.

4.2. Recomendaciones

- Se recomienda el uso de una herramienta completa en modelación UML para facilitar la construcción de los diagramas que la metodología OOHDm sugiere.
- Se recomienda tomar como referencia este proyecto para el desarrollo de sistemas de comercio electrónico orientado a grupos afectados por la crisis sanitaria, apoyados en software libre para su construcción.
- Se recomienda la aplicación de OOHDm en sistemas más complejos ya que puede ser adaptada y modificada de acuerdo a las características que el sistema requiere.

Bibliografía

- [1] CEPAL y A. Bárcena, “Coyuntura, escenarios y proyecciones hacia 2030 ante la presente crisis de Covid-19”, p. 66, 2020.
- [2] M. Agudelo *et al.*, “La Digitalización En América Latina Frente Al Covid-19”, *Corporación Andin. Fomento, 2020. Nac. Unidas, 2020, 2020*, [En línea]. Disponible en: <https://scioteca.caf.com/handle/123456789/1541>.
- [3] J. Romano, *La gran transición: retos y oportunidades del cambio tecnológico exponencial*, 1a ed. 2018.
- [4] Ministerio de Telecomunicaciones y de la Sociedad de la Información, “Las TIC como motor de la economía digital”. <https://www.telecomunicaciones.gob.ec/las-tic-como-motor-de-la-economia-digital/> (consultado jun. 02, 2020).
- [5] P. E. Tello Pérez y L. F. Pineda González, “Análisis del comercio electrónico en Ecuador”, QUITO/UIDE/2018, 2018.
- [6] INEC, “Directorio de Empresas y Establecimientos 2019”, 2020. [En línea]. Disponible en: https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Economicas/DirectorioEmpresas/Directorio_Empresas_2019/Principales_Resultados_DIEE_2019.pdf.
- [7] S. I. Maji Llinin, “El comercio electrónico y el crecimiento económico de las empresas del sector textil del cantón Ambato”, 2020.
- [8] J. Rolando, M. Ríos, M. Paola, y Z. Ordóñez, “Comparación de metodologías en aplicaciones web”, *3C Technol.*, vol. 7, p. 19, 2018, [En línea]. Disponible en: <https://www.3ciencias.com/wp-content/uploads/2018/03/art1.pdf>.
- [9] A. N. Espinoza Sánchez, “Desarrollo de una plataforma online para promoción

de servicios personales y profesionales”, Pontificia Universidad Católica del Ecuador, 2019.

- [10] I. Sommerville, *Ingeniería de Software*, 9a ed. México, D. F: Pearson Education, 2011.
- [11] R. S. Pressman, *Ingeniería del software Un Enfoque Práctico*, 7a ed. 2010.
- [12] R. V. Lerma-Blasco, *Aplicaciones Web*. Aravaca - Madrid: McGraw-Hill España, 2013.
- [13] J. Zofío Jiménez, *Aplicaciones web*. Macmillan Iberia, S.A, 2013.
- [14] J. Ferrer Martínez, *Implantación de aplicaciones web en entornos internet, intranet y extranet*. Madrid: RA-MA Editorial, 2015.
- [15] P. Lara Navarra y J. Á. Martínez Usero, *La accesibilidad de los contenidos web*. Barcelona: Editorial UOC, 2016.
- [16] Á. E. Santamarina, “Ingeniería web dirigida por modelos”, 2009.
- [17] acensTechnologies, “Framework para el desarrollo ágil de aplicaciones”, 2014. Consultado: nov. 14, 2020. [En línea]. Disponible en: <https://www.acens.com/wp-content/images/2014/03/frameworks-white-paper-acens-.pdf>.
- [18] J. J. Gutiérrez, “¿Qué es un framework web?” Consultado: nov. 14, 2020. [En línea]. Disponible en: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
- [19] J. Casas Roma y J. Conesa i Caralt, *Diseño conceptual de bases de datos en UML*. Barcelona: Editorial UOC, 2014.

- [20] M. Marqués, *Bases de datos*. Castelló de la Plana: D - Universitat Jaume I. Servei de Comunicació i Publicacions, 2009.
- [21] E. Pulido Romero, Ó. Escobar Domínguez, y J. Á. Núñez Pérez, *Base de datos*. México: Grupo Editorial Patria, 2019.
- [22] D. Pérez y Á. Cobo, *Tecnologías para el desarrollo de aplicaciones web*. Madrid: Ediciones Díaz de Santos, 2007.
- [23] W3C, “Web Services Glossary”. <https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice> (consultado nov. 10, 2020).
- [24] Microsoft, “API design guidance - Best practices for cloud applications”. <https://docs.microsoft.com/en-us/azure/architecture/best-practices/api-design> (consultado nov. 10, 2020).
- [25] RFC 7519, “JSON Web Token (JWT)”, 2015. <https://tools.ietf.org/html/rfc7519> (consultado nov. 13, 2020).
- [26] “The JSON Data Interchange Syntax COPYRIGHT PROTECTED DOCUMENT”, 2017. Consultado: nov. 13, 2020. [En línea]. Disponible en: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>.
- [27] “ReactiveX - Intro”. <http://reactivex.io/intro.html> (consultado nov. 13, 2020).
- [28] P. E. Fernández Casado, *Usabilidad web: teoría y uso*. Bogotá - Colombia: Ediciones de la U, 2019.
- [29] J. Guerrero, “Directorio de empresas, ¿Qué es y para que sirve? | Noticias de Financiación Alternativa, Crowdfunding, Inversores, Emprendedores e Fintech”, 2016. <https://crowdemprende.com/directorio-empresas-sirve/> (consultado nov. 11, 2020).

- [30] Adtuo, “5 canales de venta online para tu negocio | Adtuo”. https://adtuo.com/blog/5-canales-de-venta-online-para-tu-negocio/#Canales_de_venta_online (consultado nov. 11, 2020).
- [31] Servicio Ecuatoriano de Normalización, “Mipymes y Organizaciones de Economía Popular y Solidaria son una pieza clave para la economía del país”. <https://www.normalizacion.gob.ec/mipymes-y-organizaciones-de-economia-popular-y-solidaria-son-una-pieza-clave-para-la-economia-del-pais/> (consultado nov. 09, 2020).
- [32] Schwabe y Rossi, “An Object Oriented Approach to Web-Based Application Design”, 1998.
- [33] P. Lopistéguy, B. Losada, y P. Dagorret, “Metodologías de Concepción para Aplicaciones Hipermedia: Análisis crítico”, p. 15, 2001.
- [34] R. Soliz y F. Morales, “OOHDM (MÉTODO DE DISEÑO HIPERMEDIA OBJETO ORIENTADO) & NORMATIVA ISO 9126”, 2014.
- [35] N. Güell, D. Schwabe, y P. Vilain, “Modeling interactions and navigation in web applications”, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1921, pp. 115–127, 2000, doi: 10.1007/3-540-45394-6_11.
- [36] D. Schwabe, G. Rossi, y S. D. J. Barbosa, “Systematic Hypermedia Application Design with OOHDM”, 1996. Consultado: dic. 14, 2020. [En línea]. Disponible en: <http://www.lids.puc-rio.br/~pp>.
- [37] J. Rumbaugh y I. Jacobson, *El lenguaje unificado de modelado: guía del usuario*, 2a ed. Madrid: Pearson Educación, 2006.
- [38] D. Schwabe, G. Rossi, y S. D. J. Barbosa, “Abstraction, Composition and Lay-Out Definition Mechanisms in OOHDM”, 1995, Consultado: dic. 14, 2020. [En

línea]. Disponible en:
<https://www.cs.uic.edu/~ifc/mmwsproc/schwabe/MainPage.html#schw94>.

[39] “Emisión de Registro Único de Mipymes | Ecuador - Guía Oficial de Trámites y Servicios”. <https://www.gob.ec/mpceip/tramites/emision-registro-unico-mipymes> (consultado nov. 20, 2020).

[40] “Ranking Empresarial del Ecuador | Ekosnegocios”. <https://www.ekosnegocios.com/ranking-empresarial> (consultado nov. 20, 2020).

Anexos

Anexo 1. Cuestionario de la evaluación heurística con los principios de Nielsen para evaluar la usabilidad de la plataforma web.

Cuestionario	Puntuación (0-5)	Recomendaciones
1. Visibilidad del estado del sistema		
¿Conoce el usuario su posición actual dentro del recorrido de la navegación?		
¿El usuario tiene la retroalimentación adecuada al interactuar con formularios, botones o enlaces?		
¿El diseño de la interfaz de usuario y los iconos es consistente en todo el sistema?		
2. Relación entre el sistema y el usuario		
¿Se reconocen fácilmente los elementos de la interfaz de usuario y las indicaciones de interacción?		
¿El contenido está escrito en un idioma familiar para el usuario (sin jerga ni términos demasiado técnicos)?		
3. Control y libertad para el usuario		
¿El usuario puede salir de todos los estados, como las ventanas emergentes? ¿El estado de salida es coherente y claro?		
¿El usuario puede interactuar con todo en la página sin iniciar sesión o ser un usuario registrado?		
¿Tiene el usuario control sobre su información personal?		

¿El usuario puede reproducir / pausar animaciones (como carruseles automáticos)?		
4. Consistencia y estándares		
¿Existe una estructura y convenciones coherentes en nuestras páginas que reducen la carga cognitiva?		
¿Existe un diseño coherente para todos los elementos de la interfaz de usuario en la página?		
¿Nuestros elementos de la interfaz de usuario siguen los estándares y los usos e implementaciones esperados?		
5. Prevención de errores		
¿Existen limitaciones visuales y útiles que eviten que el usuario cometa errores?		
¿Se presenta al usuario un formato indulgente para la información?		
¿Los mensajes de error muestran un lenguaje gramaticalmente correcto al usuario?		
6. Reconocer antes que recordar		
¿Se le presenta al usuario una opción para recuperar una nueva contraseña cuando la necesite?		
¿Se le presenta al usuario una lista de contenido visualizado recientemente?		
¿Se hace uso del tamaño de letra, realce de fuente, subrayado, color, sombreado o tipografía especial para mostrar la cantidad relativa o importancia de los diferentes ítems en pantalla?		
7. Flexibilidad y eficiencia de uso		

¿Se le presentan al usuario atajos para finalizar los objetivos?		
¿Puede el usuario adaptar acciones frecuentes?		
¿Se le presenta al usuario información ambiental para acciones rápidas?		
¿Permite que los usuarios nuevos se familiaricen fácilmente con el uso del sistema?		
8. Diálogos estéticos y diseño minimalista		
¿El diseño de la interfaz de usuario es simple y fácil de entender?		
¿Está abrumado el usuario con demasiada información y acciones primarias que debe realizar?		
¿Tiene claro el usuario qué significan todos los íconos y por qué están incluidos en el diseño?		
9. Reconocimiento, diagnóstico y recuperación de errores		
¿El usuario ve un mensaje de error descriptivo cuando introduce datos incorrectos?		
¿Se le proporciona al usuario retroalimentación comprensible para los humanos (sin términos técnicos o jerga)?		
¿Se le presentan al usuario mensajes de error educados que no culpan al usuario por el error?		
10. Ayuda y documentación		
¿Se le presenta al usuario información clara sobre herramientas o información adicional cuando la necesita?		
¿Tiene el usuario acceso a información relevante que pueda necesitar al utilizar el sitio web?		

¿Se le presentan al usuario canales de comunicación para hacer preguntas o resolver dudas?		
--	--	--