

**Facultad 1**

**Portal Web Empresarial para marca cubana NAWE**

Trabajo de diploma para optar por el título de   
Ingeniero en Ciencias Informáticas

**Autor:** Dagoberto A Carralero

**Tutor:** Michel Pedrera Suen

La Habana, junio de 2024

Año 65 de la Revolución

**DECLARACIÓN DE AUTORÍA**

El autor del trabajo de diploma con título “***Portal WEB para marca cubana NAWE***” concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los <día> días del mes de junio del año 2024.

|  |
| --- |
| **Dagoberto A Carralero** |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Firma del Autor |
| **Michel Pedrera Suen** |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Firma del Tutor |

**DATOS DE CONTACTO**

Tutor: Ing. Michel Pedrera Suen. Graduado de Ingeniería en Ciencias Informáticas en el año 2020. Instructor del Departamento de Informática de la Facultad 1 de la UCI en la disciplina Técnicas de programación. Coautor del artículo "Context of the quality for Cloud Computing. Diagnosis and analysis for its acquirement in Cuban entities". Autor de la ponencia "Mecanismo de prevención de intrusos para Nova Servidores" (UCIENCIA 2023)

**AGRADECIMIENTOS**

A mi madre, que siempre ha querido que termine las cosas que comienzo. A mi familia, que me ha dado todo el apoyo. A la Universidad por darme esta magnífica oportunidad de estudio. Y a las profesoras Ailín y Madelin, cuya atención y preocupación durante todos estos años nunca olvidaré.

**RESUMEN**

Este trabajo de investigación se centra en el desarrollo de un portal web para la empresa NAWE, con el objetivo de mejorar su visibilidad en línea y facilitar a los clientes el acceso a información detallada sobre los productos y servicios ofrecidos. Se abordan los fundamentos teóricos y metodológicos que sustentan el proyecto, incluyendo el análisis de sistemas homólogos y la selección de las herramientas y tecnologías. Se valida la solución, con diversas estrategias de pruebas aplicadas. Los resultados de estas pruebas demuestran el cumplimiento de los estándares y requisitos establecidos, garantizando la calidad y fiabilidad del producto final antes de su entrega al cliente.

PALABRAS CLAVE

Visibilidad en línea. Portal web.

ABSTRACT

This research work focuses on the development of a web portal for the company NAWE, with the aim of improving its online visibility and facilitating access for clients to detailed information about the products and services offered. The theoretical and methodological foundations that support the project are addressed, including the analysis of similar systems and the selection of tools and technologies. The solution is validated through various testing strategies applied. The results of these tests demonstrate compliance with established standards and requirements, ensuring the quality and reliability of the final product before delivery to the client.

KEYWORDS

Online Visibility. Web Portal.

TABLA DE CONTENIDOS

[INTRODUCCIÓN 1](#_Toc170467891)

[CAPÍTULO I: Fundamentos y referentes teórico-metodológicos sobre el objeto de estudio 6](#_Toc170467892)

[I.1 Conceptos fundamentales 6](#_Toc170467893)

[I.2 Sistemas Homólogos 7](#_Toc170467894)

[I.3 Metodología de desarrollo, AUP-UCI, Escenario 4 8](#_Toc170467895)

[I.3 Herramientas y tecnologías 9](#_Toc170467896)

[Conclusiones del capítulo 16](#_Toc170467897)

[CAPÍTULO II: DISEÑO e Implementacion de LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO 17](#_Toc170467898)

[II.1 Propuesta de solución 17](#_Toc170467899)

[II.2 Requisitos funcionales 17](#_Toc170467900)

[II.3 Requisitos no funcionales 20](#_Toc170467901)

[II.4 Historias de usuario 20](#_Toc170467902)

[II.5 Estilo Arquitectónico 25](#_Toc170467903)

[II.6 Diagrama de Clases 26](#_Toc170467904)

[II.7 Patrones de diseño 27](#_Toc170467905)

[II.8 Modelo de datos 29](#_Toc170467906)

[II.9 Implementación 31](#_Toc170467907)

[Conclusiones del capítulo 33](#_Toc170467908)

[CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA 35](#_Toc170467909)

[III.1 Pruebas de sistema 35](#_Toc170467910)

[Pruebas de rendimiento, carga y stress. 37](#_Toc170467911)

[III.2 Pruebas de aceptación 38](#_Toc170467912)

[Conclusiones del capítulo 39](#_Toc170467913)

[CONCLUSIONES FINALES 41](#_Toc170467914)

[RECOMENDACIONES 42](#_Toc170467915)

[REFERENCIAS BIBLIOGRÁFICAS 43](#_Toc170467916)

[ANEXOS 45](#_Toc170467917)

ÍNDICE DE TABLAS

[Tabla 1. Comparativa de Homólogos. Elaborada por el autor. 8](#_Toc170467884)

[Tabla 2. Metodología de desarrollo. Tomado de Veitia (2022) 8](#_Toc170467885)

[Tabla 3. Requisitos Funcionales. Elaborada por el autor 18](#_Toc170467886)

[Tabla 4. Requisitos No Funcionales. Elaborada por el Autor. 20](#_Toc170467887)

[Tabla 5. Historia de usuario: Crear Servicio. Elaborada por el autor. 21](#_Toc170467888)

[Tabla 6. Historia de Usuario: Vista publica de Servicio. Elaborada por el autor. 23](#_Toc170467889)

[Tabla 7 Historia de Usuario: Realizar Orden de servicio. Elaborada por el autor. 24](#_Toc170467890)

ÍNDICE DE FIGURAS

[Figure 1. Modelo Vista Controlador. Tomado de SOFTWARE ENGINEERING Ninth Edition Ian Somerville 25](#_Toc170467871)

[Figure 2. Diagrama de clases. Vista pública de artículo. Elaborada por el autor. 26](#_Toc170467872)

[Figure 3. Ruta Articulo. Muestra del código fuente de la solución. Patrón experto. Tomada por el autor. 28](#_Toc170467873)

[Figure 4. Ruta de Servicio de Empaquetado con bolsas de Papel. Patrón Bajo Acoplamiento. Tomada por el autor. 29](#_Toc170467874)

[Figure 5. Modelo de datos. Elaborada por el autor. 30](#_Toc170467875)

[Figure 6. Ejemplo de normal Pasca Case en el trabajo de los componentes React. Elaborada por el autor. 31](#_Toc170467876)

[Figure 7. Diagrama de componentes. Caso de uso Vista pública de articulo. Elaborado por el autor. 32](#_Toc170467877)

[Figure 8. Diagrama de despliegue. Elaborado por el autor 32](#_Toc170467878)

[Figure 9. Prueba de extremo a extremo. Caso de uso de Vista publica de Articulo. Elaborada por el autor. 36](#_Toc170467879)

[Figure 10. Prueba de rendimiento carga y stress. Elaborada por el autor. 37](#_Toc170467880)

[Figure 11. Pruebas de Aceptación. Elaborada por el autor. 38](#_Toc170467881)

[Figure 12 Caso de prueba Realizar Orden. Elaborada por el autor. 38](#_Toc170467882)

[Figure 13. Escenarios de Prueba. Elaborada por el autor. 39](#_Toc170467883)

**OPINIÓN DEL(OS) TUTOR(ES)**

<Contenido de la opinión de los tutores>

**AVAL DEL CLIENTE**

Mi nombre es Clara Valdés Suárez, ejerzo como Trabajadora por Cuenta Propia desde el 2023, con el proyecto NAWE, una marca que comercializa productos de diseño original, y personaliza artículos para otros creadores y organizaciones. Combina medios tradicionales y contemporáneos de diseño, estampado y confección manual. Con el presente comunicado avalo el trabajo de tesis de Dagoberto A Carralero que busca aplicar sus conocimientos adquiridos durante sus años de estudios universitarios con consecuencias prácticas en el desarrollo de una Plataforma Web como como canal para hacer más accesible la cartera de productos y servicios que presenta el negocio.

# INTRODUCCIÓN

*“Es innegable que tanto las Tecnologías de la Información y las Comunicaciones (TIC), como el uso del Internet, han pasado a formar parte de prácticamente todos los entornos en la vida cotidiana.” (Rodríguez, Chapis, 2019)*

El uso de sitios web para promover productos y empresas en era digital se ha vuelto muy frecuente. Estudios previos han demostrado que la calidad de un sitio web afecta significativamente la satisfacción del cliente y su percepción de la marca. También la calidad del sitio influye en la intención de compra del cliente y su intención de volver a comprar. El sitio web ofrece una función para que los consumidores y las PYME (Pequeñas y Medianas Empresas) realicen transacciones cómodamente.

*“Un sitio web atractivo y fácil de usar creará conciencia de marca en la mente del consumidor y creará confianza en los consumidores, lo que en última instancia influirá en su decisión de comprar productos o servicios a través del sitio web” (Nurhadi, 2023)*

La presencia en línea de un negocio, sin importar la industria, puede tener un impacto masivo en su éxito, hoy día una gran parte de los clientes se acercará al sitio web de la empresa antes de comprar un producto o servicio. Una de las principales razones para que una organización tenga su sitio web es incrementar su credibilidad, no tener un sitio web las personas pueden cuestionarse la legitimidad del negocio. Tener un sitio web es una oportunidad de hacer una buena primera impresión y darle confort a las personas de que se trata de un negocio real. Estableciendo claramente quien es la organización, que representa y cuáles son sus valores aumentan las probabilidades de que los clientes conozcan la marca y compren. Una vez se tiene el sitio Web y se optimiza su SEO, aumentan las posibilidades de aparecer entre los primeros resultados de los motores de búsqueda, lo cual trae consigo un drástico aumento en la cantidad de clientes.

Un sitio web ayuda a ahorrar el tiempo en la atención al cliente, teniendo toda la información importante de forma fácil de encontrar y consumir, muchos de los clientes podrán encontrarla por sí mismos, en vez de tener que contactar directamente.

*“El sitio web es también un espacio para comunicar actualizaciones y anuncios a los clientes, eventos, publicaciones, entre otros. El sitio, brinda también una poderosa oportunidad para hacer Marketing Digital, captando los clientes interesados, a los cuales se les pueden enviar comunicados de marketing, ofertas especiales, entre otros.” (Kasey Kaplan, 2020)*

**NAWE** es una pequeña empresa cubana que oferta artículos promocionales personalizados para Emprendimientos, Eventos y Proyectos de Creación Independiente. Es una marca que comercializa productos de diseño original y confección. A su vez cuenta con una plataforma de Servicios y Artículos Promocionales Personalizados para que emprendimientos, entidades y creadores independientes puedan llevar a cabo diferentes proyectos de bajo y alto volumen de producción.

Se especializan en la Serigrafía y Sublimación, logrando diferentes resultados visuales, dependiendo de la necesidad del cliente. Estas técnicas son una gran ventaja por su versatilidad, resistencia al paso del tiempo y la relativa disminución en los costes en comparación con otras formas de impresión. Cuentan con la capacidad de importación y producción necesaria para crear merchandising de calidad, prendas y uniformes, etc. Lo mismo para eventos, festivales y encargos de grandes cantidades, manejan productos y materiales dentro de las gamas económica y premium.

Actualmente esta empresa no cuenta con un catálogo en línea que permita promocionar sus productos y servicios, lo que afecta a su visibilidad, el crecimiento de las ventas y clientes potenciales. Tampoco tiene su información de contacto online, lo cual dificulta la comunicación con sus clientes. Los clientes a menudo presentan las cuantiosas preguntas sobre las ofertas, que deben ser explicadas con gran nivel de detalle. Estas preguntas en su inmensa mayoría son recurrentes, y el pequeño equipo debe dar respuesta a ellas mediante llamadas telefónicas o conversaciones mediante aplicaciones de mensajería móvil como WhatsApp, lo cual les resta valioso tiempo que quisieran dedicar a la producción, y agilizar al máximo el proceso de venta.

Atendiendo a la situación problemática antes expuesta se plantea como **problema de investigación**: ¿Cómo contribuir a la visibilidad de los productos y servicios de NAWE en internet, y agilizar el proceso de venta? Donde **el objeto de estudio** de la presente investigación va orientado al proceso de visibilidad empresarial en internet.

Para solucionar la situación anteriormente expuesta se propone como **objetivo general** desarrollar un portal web para la empresa NAWE que contribuya con su visibilidad de productos y servicios en internet y agilice y facilite el proceso de venta.

Para guiar el cumplimiento del objetivo planteado, se formulan las siguientes preguntas científicas:

1. ¿Cuáles son los referentes teóricos fundamentales que sustentan la investigación relacionados con la visibilidad de los productos y servicios de la empresa NAWE?

2. ¿Qué propuesta de solución se define para mejorar la visibilidad de los productos y servicios de la empresa NAWE?

3. ¿Qué componentes son necesarios para implementar el funcionamiento del sitio web de la empresa NAWE?

4. ¿Cómo validar el funcionamiento del sitio web de la empresa NAWE?

Con el propósito de darle cumplimiento al objetivo general y al problema anteriormente planteado se trazaron las siguientes tareas de la investigación:

1. Estudio de los referentes teóricos fundamentales que sustentan la investigación relacionados con la visibilidad de los productos y servicios de la empresa.

2. Análisis y diseño del portal web.

3. Implementación de las funcionalidades del portal web.

4. Validación de las funcionalidades del portal web.

Los **métodos científicos** utilizados para desarrollar la investigación fueron:

**Métodos teóricos:**

**Histórico-Lógico**: se emplea para estudiar la evolución histórica y tendencias actuales de la empresa, para la construcción de los referentes teóricos que sustentan la investigación, el estudio de soluciones similares, así como de las metodologías de desarrollo, las tecnologías, herramientas, frameworks (marcos de trabajo), lenguajes de programación y de modelado utilizados.

**Análisis-Síntesis:** empleado para el análisis, evaluación y selección de las técnicas a emplear en el desarrollo del sitio. Así como para sintetizar la información que se obtuvo mediante la entrevista con el cliente de manera que pudiera ser usada en el desarrollo de este, además, en la identificación de los elementos del marco teórico de la investigación.

**Modelación:** para realizar una representación del proceso estudiado que sirva de guía en el desarrollo del portal, y mediante este, identificar las características y relaciones fundamentales que den cumplimiento a los requisitos funcionales de la solución propuesta.

El presente documento está compuesto por tres capítulos en los que se relacionan todo lo referente a la investigación. A continuación, se muestra la descripción de los capítulos:

**Capítulo 1.** Fundamentos y referentes teórico-metodológicos del sitio web para NAWE: se realiza un estudio del arte sobre los aspectos teóricos que sustentan el desarrollo de la solución propuesta. Se describe el proceso de desarrollo de software, así como las tendencias, técnicas, metodología y tecnologías usadas en la propuesta.

**Capítulo 2.** Diseño e implementación de la solución propuesta del sitio web para NAWE: se identifican y describen los conceptos asociados al dominio del problema y los procesos relacionados con el negocio teniendo en cuenta la metodología seleccionada en el capítulo anterior. Se definen cuáles son los requerimientos funcionales, no funcionales y el modelo de datos. Se procede con la implementación.

**Capítulo 3.** Validación de la solución propuesta del sitio web para NAWE: en este capítulo se define la estrategia de pruebas a aplicar para lograr una correcta validación del producto final, así como los resultados de la misma.

El presente trabajo contiene además Conclusiones, Recomendaciones, Referencias Bibliográficas y Anexos.

# CAPÍTULO I: Fundamentos y referentes teórico-metodológicos sobre el objeto de estudio

El presente capítulo tiene como objetivo abordar los diferentes elementos que brindan la base teórica y conceptual para el desarrollo de la investigación. Valorar de forma crítica las tendencias y tecnologías actuales, así como los antecedentes asociados al objeto de estudio. Se realiza un análisis de las características de los diversos portales web empresariales a nivel nacional e internacional. De este modo, se podrá realizar una correcta interpretación de la situación problemática y del problema a resolver.

## I.1 Conceptos fundamentales

Para obtener una mejor comprensión del dominio del problema, a continuación, son descritos conceptos asociados al mismo.

**Visibilidad en internet:**

*“Técnicamente, la visibilidad en internet consiste en la predisposición que tiene un sitio web para ser rastreado en su totalidad por un buscador e indizado dentro de la categoría que corresponda por su temática y con las palabras clave que utiliza el público objetivo en cada situación concreta.” (Arévalo, 2018)*

**Portal Web**

*“El término portal tiene como significado “puerta grande” el cual hace referencia a su función el cual es el punto de partida de un usuario que desea entrar y realizar búsquedas en la web de donde se obtiene información importante. Un portal debe tener necesariamente un tema y este debe estar referido a algún tema de interés de un grupo importante, ya que un portal presenta la información de una forma fácil e integrada. Una característica principal de un portal es que se encarga de centralizar enlaces de una forma fácil y organizada que facilite la navegación dentro de un tema específico.” (CERRON, 2017)*

## I.2 Sistemas Homólogos

[**Rushordertees.com**](https://www.rushordertees.com/)

Este portal web de la empresa Rush Order Tees brinda servicios y productos personalizables con una amplia variedad. En él se puede apreciar a nivel navegación el logo de la empresa, información de contacto, enlaces a las diversas secciones del sitio. Presenta una arquitectura de información donde se encuentran servicios y productos. Para determinados artículos, cuenta con la funcionalidad de editor, donde el mismo cliente puede personalizar el artículo que desea, ejemplo: seleccionar el texto, color, tipografía, y posición en el T-Shirt a personalizar. Una vez el cliente a personalizado, puede proceder a realizar la orden. En todo momento está presente la identidad de la empresa a nivel del logo, y el método de contacto para con el cliente.

[**Bolsas-de-papel.es**](https://bolsas-de-papel.es/)

Bolsas de papel es una tienda online de Bolsas de Papel personalizadas y embalaje decorativo. Presenta una arquitectura de artículos, bolsas de Papel, personalizables y no personalizables entre otros. No presenta funciones de personalización a nivel de la experiencia de usuario en el portal. A su vez, en todo momento deja presente la identidad corporativa y las opciones de contacto.

[**Spreadshirt.es**](https://www.spreadshirt.es/)

Spreadshirt brinda servicios de personalización de diversos artículos textiles, como camisetas, sudaderas, gorros entre otros. Presenta una arquitectura de dos tipos de artículos los personalizables por el cliente con texto propio, y otros artículos ya personalizados por terceros. Presenta funcionalidad de edición, donde el cliente una vez seleccionado el artículo que desea personalizar, tiene opciones para escoger el texto, tipografía, color, posición entre otros, en la pieza que se va a personalizar. Una vez realizado este paso, el cliente puede proceder a realizar su orden. En todo momento está presente la información de contacto.

**Tabla comparativa de los sistemas homólogos**

Tabla . Comparativa de Homólogos. Elaborada por el autor.

|  |  |  |  |
| --- | --- | --- | --- |
| Cualidades | Rushordertees | Bolsas de papel | Spreadshirt |
| Información Contacto | Si | Si | Si |
| Servicios | Si | Si | Si |
| Productos | Si | Si | Si |
| Función de Editor a nivel de producto | Si | No | Si |

En todos los casos estudiados, se encontraron satisfactorias las cualidades encontradas. Dan respuesta a la problemática de cada empresa en su contexto. Estos portales no son de código abierto, ni están adaptados al entorno y el contexto cubano y de la empresa NAWE en específico, por lo que no serán utilizados más allá que como referencias de cualidades deseadas en la realización del nuevo portal.

## I.3 Metodología de desarrollo, AUP-UCI, Escenario 4

La metodología de desarrollo de software AUP versión UCI La Universidad de las Ciencias Informáticas (UCI) desarrolló una versión de la metodología de desarrollo de software AUP (Proceso Ágil Unificado), con el fin de crear una metodología que se adapte al ciclo de vida definido por la actividad productiva de la universidad. Esta versión decide mantener para el ciclo de vida de los proyectos la fase de Inicio, pero modificando el objetivo de la misma y seunifican las restantes fases de la metodología de desarrollo de software AUP en una sola, nombrada Ejecución y agregándose también una nueva fase llamada Cierre. A continuación, se muestra una tabla con las fases de la metodología AUP-UCI:

Tabla . Metodología de desarrollo. Tomado de Veitia (2022)

|  |  |
| --- | --- |
| Fases Variación AUP-UCI | Objetivos de las fases (Variación AUP-UCI) |
| Inicio | Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto. |
| Ejecución | En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto. |
| Cierre | En esta fase se analizan tanto los resultados del proyecto como la ejecución y se realizan las actividades formales de cierre del proyecto. |

El escenario 4, se aplica a proyectos que hayan realizado una evaluación exhaustiva del negocio a informatizar, resultando en una definición precisa del mismo. El cliente mantendrá una colaboración constante con el equipo de desarrollo para acordar los pormenores de los requisitos, facilitando así su implementación, prueba y validación. Esta metodología se recomienda para proyectos de alcance moderado, dado que las Historias de Usuario (HU) no deben contener una cantidad excesiva de información. Todas las disciplinas previamente establecidas (desde el Modelado de negocio hasta las Pruebas de Aceptación) se ejecutan en la Fase de Ejecución, razón por la cual en esta etapa se llevan a cabo Iteraciones que producen resultados incrementales.

## I.3 Herramientas y tecnologías

**Visual Paradigm v17**

Visual Paradigm es una herramienta CASE que contribuye al desarrollo de sistemas de software fiables, mediante un enfoque orientado a objetos. Soporta el ciclo completo de desarrollo de software y permite su documentación en diferentes formatos, empleando UML como lenguaje de modelado.

**Figma v124**

Es una herramienta que permite el diseño de interfaces para aplicaciones web, móviles entre otras. Permite diseñar las diferentes páginas que compondrán el portal con gran fidelidad, así como todos los subcomponentes que la conforman. Brindando también un soporte para el trabajo con el cliente, donde este puede ver de manera rápida y sencilla los prototipos de las páginas, de forma que él pueda desde esta temprana fase ir pidiendo ajustes y modificaciones.

**Lenguajes de programación**

**TypeScript v5.1.6**

Es un lenguaje de programación de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript que añade tipos estáticos opcionales al lenguaje. TypeScript está diseñado para el desarrollo de aplicaciones grandes y transcompila a JavaScript. Esto significa que el código TypeScript se convierte en código JavaScript que puede ser ejecutado en cualquier entorno que soporte JavaScript, como navegadores web y servidores.

**JavaScript v2022**

JavaScript es un lenguaje de programación de alto nivel, multiparadigma, estandarizado por ECMA International a través de la especificación ECMAScript (ECMA-262). Utilizado tanto en el lado del cliente como en el servidor, JavaScript es dinámico, de tipado débil, y soporta funciones de primera clase y herencia basada en prototipos. Es esencial para el desarrollo web moderno, con un ecosistema rico de bibliotecas y frameworks como React. Además, soporta programación asíncrona y se actualiza regularmente con nuevas características a través de ECMAScript. En este caso se seleccionó la versión especificada por ECMA en el año 2022.

**HTML 5**

HTML5 (Hypertext Markup Language versión 5) es la última iteración principal del lenguaje de marcado estándar utilizado para estructurar y presentar contenido en la World Wide Web. Desarrollado y mantenido por el World Wide Web Consortium (W3C), HTML5 introduce nuevos elementos, atributos y APIs que mejoran la funcionalidad web, incluyendo un soporte multimedia mejorado, estructuras de contenido semántico y compatibilidad multiplataforma. Esta versión tiene como objetivo proporcionar una experiencia web más robusta, eficiente y accesible en diversos dispositivos y plataformas, manteniendo la compatibilidad con versiones anteriores de HTML.

**CSS 3**

CSS3 (Cascading Style Sheets versión 3) es la última especificación estándar del lenguaje de hojas de estilo utilizado para describir la presentación de documentos estructurados en HTML y XML. Desarrollado por el World Wide Web Consortium (W3C), CSS3 introduce capacidades avanzadas de diseño y efectos visuales, incluyendo animaciones, transiciones, diseños flexibles y responsivos, así como una mayor personalización tipográfica. Esta versión mejora significativamente el control sobre la apariencia y el diseño de las páginas web, permitiendo una separación más eficiente entre el contenido y su presentación visual, y facilitando la creación de interfaces de usuario más sofisticadas y adaptables a diferentes dispositivos y tamaños de pantalla.

**Tailwind CSS v3**

Tailwind CSS es un framework de CSS de código abierto, orientado a la utilidad, que permite a los desarrolladores construir interfaces de usuario personalizadas de manera rápida y eficiente. A diferencia de los frameworks tradicionales, Tailwind CSS proporciona una serie de clases utilitarias de bajo nivel que se pueden combinar para crear diseños complejos directamente en el marcado HTML, sin necesidad de escribir CSS personalizado. Este enfoque modular y altamente configurable facilita la creación de estilos consistentes y reutilizables, mejorando la productividad y manteniendo el código limpio y mantenible. Tailwind CSS es ampliamente adoptado en la industria del desarrollo web por su flexibilidad y capacidad para acelerar el proceso de implementación.

**React v18**

React es una biblioteca de JavaScript de código abierto desarrollada y mantenida por Facebook (ahora Meta) y una comunidad de desarrolladores individuales y empresas. Diseñada para crear interfaces de usuario interactivas y eficientes, React utiliza un paradigma de programación declarativo y basado en componentes. Su característica distintiva es el uso de un "Virtual DOM" (Document Object Model Virtual) que optimiza las actualizaciones de la interfaz de usuario, mejorando significativamente el rendimiento de las aplicaciones web. Con su enfoque en la reutilización de componentes y el flujo de datos unidireccional, React ha revolucionado el desarrollo de interfaces de usuario front-end, convirtiéndose en una de las bibliotecas más populares y ampliamente adoptadas en la industria del desarrollo web moderno.

**Remix v2**

Remix es un framework de desarrollo web de código abierto basado en React, diseñado para crear aplicaciones web modernas y eficientes. Desarrollado por los creadores de React Router, Remix se enfoca en la optimización del rendimiento y la experiencia del usuario mediante el aprovechamiento de las API web nativas y la implementación de patrones de carga y actualización de datos eficientes. Este framework facilita la creación de aplicaciones web robustas y escalables, ofreciendo características como renderizado del lado del servidor (SSR), carga de datos en paralelo, manejo de errores integrado y una arquitectura basada en rutas. Remix promueve la creación de interfaces de usuario dinámicas y responsivas, al tiempo que simplifica el proceso de desarrollo y mejora la mantenibilidad del código.

**Cloudflare Workers y Pages**

Cloudflare Workers y Cloudflare Pages son servicios de computación en la nube ofrecidos por Cloudflare, Inc., una empresa líder en seguridad y rendimiento web. Cloudflare Workers es una plataforma de computación serverless que permite ejecutar código JavaScript en el borde de la red de Cloudflare, cerca de los usuarios finales. Cloudflare Pages extiende esta funcionalidad, integrando Workers con un sistema de alojamiento y despliegue continuo para sitios web y aplicaciones web dinámicas. Posibilitando la creación de aplicaciones full-stack que combinan contenido estático con funcionalidad dinámica del lado del servidor. A la vez que abstrae todos los detalles del manejo de la infraestructura que ejecuta la aplicación del desarrollador. En este caso al ser servicios brindados por Cloudflare no tienen una versión específica, se encuentran en constante cambio y evolución, mientras preservan compatibilidad con las aplicaciones desplegadas en ellos. Se utilizaron para hostiar el portal web.

**Ubuntu Server v.20.04**

Ubuntu Server 20.04 LTS es una distribución de Linux basada en Debian, diseñada específicamente para servidores. Desarrollada y mantenida por Canonical Ltd., esta versión de Ubuntu ofrece estabilidad, seguridad y rendimiento optimizados para entornos de producción. Ubuntu Server 20.04 incluye soporte a largo plazo (LTS) con actualizaciones de seguridad y mantenimiento garantizadas durante cinco años. Entre sus características destacadas se encuentran la compatibilidad con las últimas tecnologías de contenedores y virtualización, herramientas de gestión de redes avanzadas, y un ecosistema robusto de software y servicios que facilitan la implementación y administración de servidores en entornos empresariales y en la nube. En este caso se utilizó para hospedar el CMS (descrito a continuación).

**PocketBase v0.22**

PocketBase es un sistema de gestión de contenidos (CMS) y backend de código abierto, diseñado para ofrecer una solución rápida y eficiente para el desarrollo de aplicaciones web y móviles. Desarrollado en Go, PocketBase combina una base de datos SQLite integrada con una API RESTful autogenerada y un panel de administración basado en web. Este CMS se destaca por su enfoque en la simplicidad y la facilidad de uso, permitiendo a los desarrolladores crear y gestionar bases de datos, autenticación de usuarios y almacenamiento de archivos con mínima configuración. PocketBase es altamente portable, se distribuye como un único archivo ejecutable, y ofrece características avanzadas como migraciones de base de datos, hooks en tiempo real y un sistema de plugins extensible, lo que lo convierte en una opción atractiva para proyectos de diversos tamaños y complejidades.

**Caddy v2**

Caddy es un servidor web de código abierto, listo para producción, que simplifica el proceso de configurar y gestionar sitios web y aplicaciones web seguras. Desarrollado en Go, Caddy se destaca por su aprovisionamiento automático de HTTPS, soporte para HTTP/2 y HTTP/3, y una configuración fácil de usar. Sus principales características incluyen la obtención y renovación automática de certificados SSL/TLS, la capacidad de actuar como proxy inverso y balanceador de carga, una arquitectura modular extensible a través de plugins, una configuración sencilla basada en el formato Caddyfile, un alto rendimiento gracias a su arquitectura dirigida por eventos, y la aplicación de mejores prácticas de seguridad como soporte para las últimas versiones de TLS. La combinación de facilidad de uso, características de seguridad y rendimiento hacen de Caddy una opción popular para desarrolladores y administradores de sistemas que buscan desplegar y gestionar aplicaciones web seguras de manera rápida. En este caso se usó para actuar como proxy reverso del backend PocketBase, para así manejar el aseguramiento del tráfico mediante certificados TLS que el mismo autogestiona.

**Git v2.42**

Git es un sistema de control de versiones distribuido, de código abierto, diseñado para manejar proyectos de cualquier tamaño con velocidad y eficiencia. Desarrollado por Linus Torvalds en 2005, Git permite a múltiples desarrolladores colaborar en proyectos de software, manteniendo un historial completo de cambios y facilitando la creación de ramas para el desarrollo paralelo. Con su arquitectura distribuida, Git ofrece operaciones rápidas, integridad de datos y soporte para flujos de trabajo no lineales, convirtiéndolo en la herramienta estándar de facto para el control de versiones en la industria del desarrollo de software.

**GitHub**

GitHub es una plataforma de desarrollo colaborativo y control de versiones basada en Git, que permite a los desarrolladores alojar, revisar y gestionar proyectos de software. Propiedad de Microsoft, GitHub ofrece herramientas avanzadas para la integración continua, despliegue continuo (CI/CD), y colaboración en equipo, facilitando la gestión de código fuente y la documentación. GitHub se ha convertido en un estándar de facto en la industria del software, promoviendo prácticas de desarrollo ágil y open source.

**Visual Studio Code v1.89**

Visual Studio Code, comúnmente conocido como VS Code, es un editor de código fuente gratuito y de código abierto desarrollado por Microsoft para Windows, Linux, macOS y navegadores web. Lanzado inicialmente en 2015, VS Code está escrito en TypeScript, JavaScript, HTML y CSS, y se ha convertido en uno de los entornos de desarrollo más populares entre los desarrolladores. Ofrece características avanzadas como soporte para depuración, resaltado de sintaxis, autocompletado inteligente, snippets, refactorización de código y control de versiones integrado con Git. VS Code es altamente personalizable a través de temas, atajos de teclado y una amplia gama de extensiones que amplían su funcionalidad. Su flexibilidad, rendimiento y ecosistema robusto lo han posicionado como una herramienta líder en la industria del desarrollo de software

**Loadtest v8**

Una herramienta de código abierto, que permite simular escenarios realistas de carga y estrés para probar el sistema web.

**Cypress v13**

Cypress es una herramienta de código abierto para realizar pruebas de tipo End to End, entre otras.

## Conclusiones del capítulo

En este capítulo se han abordado los elementos teóricos que dan sustento a la propuesta de solución del problema planteado, en tal sentido se concluye que el estudio de los conceptos fundamentales permitió tener una comprensión del objeto de estudio. El análisis de los sistemas homólogos permitió identificar elementos que sirvieron para facilitar una mejor implementación del producto. El análisis de las herramientas permitió escoger las herramientas idóneas para realizar el proyecto, seleccionando un servicio para hostiar el sitio, así como administrar su contenido, así como el lenguaje de programación y los framework a emplear.

# CAPÍTULO II: DISEÑO e Implementacion de LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO

El presente capítulo aborda los principales aspectos relacionados con las características de la propuesta de solución. Se identifican los requisitos funcionales y no funcionales con los que debe cumplir la solución propuesta, así como estilo arquitectónico y los patrones de diseño para lograr buenas prácticas en el diseño y posterior implementación del portal. Igualmente se muestran los principales artefactos de ingeniería de software propuestos por la metodología utilizada.

## II.1 Propuesta de solución

La solución que se propone es la de desarrollar un portal que permita a los usuarios, específicamente clientes conocer acerca de los productos y servicios de la empresa, con gran profundidad de detalle. Permita que estos clientes puedan encontrar estos recursos con mayor facilidad en internet, beneficiando así la visibilidad de la empresa en línea. Contará también con una sección de contacto y enlace a los perfiles en las redes sociales. Se priorizará mantener un lenguaje visual acorde a la marca, una arquitectura de la información sencilla, fácil de entender y navegar para el usuario, así como garantizar el rendimiento para lograr brindar una buena experiencia de usuario. A su vez el portal deberá contar con un sistema de administración del contenido, donde el personal de la empresa pueda administrar de manera sencilla el contenido del portal.

## II.2 Requisitos funcionales

*“Los requisitos de un sistema son las descripciones de lo que el sistema debería hacer: los servicios que proporciona y las limitaciones de su funcionamiento. Estos requisitos reflejan las necesidades de los clientes de un sistema que sirva para un determinado propósito, como controlar un dispositivo, realizar un pedido o encontrar información. El proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se denomina ingeniería de requisitos (RE)”* *(Somerville, 2011).*

Tabla . Requisitos Funcionales. Elaborada por el autor

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nombre** | **Prioridad** | **Descripción** |
|  |  |  |  |
| 1 | Crear usuario administrador | Alta | Permite crear un nuevo administrador de contenido en el sistema |
| 2 | Actualizar administrador | Baja | Actualiza la contraseña y o correo del usuario |
| 3 | Listar Administradores | Baja | Muestra la lista de los administradores |
| 4 | Eliminar Administrador | Baja |  |
| 5 | Autenticar Administrador | Alta | Elimina el usuario del sistema |
|  |  |  |  |
| 6 | Crear Servicio | Alta | Permite crear un nuevo servicio en el sistema |
| 7 | Modificar Servicio | Media | Permite actualizar la información del servicio |
| 8 | Listar Servicios | Media | Permite ver la lista de todos los servicios |
| 9 | Eliminar Servicio | Media | Permite eliminar el servicio del sistema |
|  |  |  |  |
| 10 | Crear Artículo | Alta | Permite crear un nuevo producto en el sistema |
| 11 | Actualizar Artículo | Alta | Permite modificar los detalles del artículo |
| 12 | Listar Artículos | Alta | Permite ver la lista de todos los artículos |
| 13 | Eliminar Articulo | Media | Permite eliminar el artículo del sistema |
|  |  |  |  |
| 14 | Mostrar Información de contacto | Alta | Permite visualizar la información de contacto |
| 15 | Actualizar Información de contacto | Alta | Permite modificar la información de contacto. |
|  |  |  |  |
| 16 | Vista pública de Información de contacto | Alta | Permite al visitante acceder a la información de contacto del negocio. |
|  |  |  |  |
| 17 | Vista pública del listado de servicios | Alta | Permite al visitante ver el catálogo de servicios |
| 18 | Vista pública del servicio | Alta | Permite al visitante ver con detalle toda la información referente al servicio |
| 19 | Ordenar el servicio | Alta | Permite al visitante realizar una orden del servicio |
|  |  |  |  |
| 20 | Vista pública del listado de artículos | Alta | Permite al visitante visualizar el catálogo de articulos |
| 21 | Vista pública del artículo | Alta | Permite al visitante ver con detalle toda la información referente al artículo |
| 22 | Realizar orden del artículo | Alta | Permite al visitante realizar una orden del artículo |

## II.3 Requisitos no funcionales

*“Estos son restricciones sobre los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, restricciones en el proceso de desarrollo y restricciones impuestas por estándares. Los requisitos no funcionales a menudo se aplican al sistema en su conjunto, en lugar de a características o servicios individuales del sistema. Requisitos no funcionales, como rendimiento, seguridad o disponibilidad, suelen especificar o restringir características del sistema en su conjunto”. (Somerville, 2011)*

Tabla . Requisitos No Funcionales. Elaborada por el Autor.

|  |  |  |
| --- | --- | --- |
| **Tipo** | **No** | **Requisito** |
|  |  |  |
| Eficiencia | 1 | El portal debe poder soportar al menos 120 personas usándolo de manera concurrente. |
|  | 2 | El portal debe poder usarse en dispositivos con al menos 1Gb de RAM. |
|  | 3 | El tiempo de respuesta medio del portal debe ser menor que 2 segundos |
|  |  |  |
| Seguridad | 4 | Debe garantizar la seguridad del acceso de sus usuarios, usando el protocolo HTTPS. |

## II.4 Historias de usuario

*“Una historia de usuario es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final. Su propósito es articular cómo proporcionará una función de software valor al cliente” (REHKOPF).*

En correspondencia con la selección del escenario número cuatro de la metodología empleada se procede a modelar el portal con historias de usuario, donde se define una por cada requisito funcional. Se muestran a continuación las HU “Crear Servicio”, “Ver vista pública del servicio” y “Ordenar servicio”.

Tabla . Historia de usuario: Crear Servicio. Elaborada por el autor.

|  |  |
| --- | --- |
| **Número:** 6 | **Nombre del requisito:** Crear Servicio |
| **Programador:** Dagoberto A Carralero | **Iteración Asignada:** 1 |
| **Prioridad:** Alta | **Tiempo Estimado:** 5h |
| **Riesgo de Desarrollo:** Alta | **Tiempo Real:** 3h |
| **Descripción:** Los administradores podrán crear servicios en el portal, para ello deben llenar los siguientes campos:  **Información de SEO:**   * Meta Título (obligatorio, campo de texto) * Meta Descripción (campo de texto) * Imagen Social (campo de imagen)   **Información del servicio:**   * Título (obligatorio, campo de texto) * Descripción (campo de texto con formatos) * Slug (obligatorio, campo de texto) * Imagen Principal Desktop (obligatorio, campo de imagen) * Imagen Principal Móvil (obligatorio, campo de imagen) * Texto Alternativo de la imagen principal (campo de texto) * Artículos (campo de selección múltiple con opción de articulo)     **Observaciones:**  Si el usuario introduce la información dejando campos obligatorios vacíos, el portal emite un mensaje indicándole que los campos obligatorios deben llenarse. | |
| **Prototipo de interfaz** | |

Tabla . Historia de Usuario: Vista publica de Servicio. Elaborada por el autor.

|  |  |
| --- | --- |
| **Número:** 18 | **Nombre del requisito:** Ver vista pública del servicio |
| **Programador:** Dagoberto A Carralero | **Iteración Asignada:** 1 |
| **Prioridad:** Alta | **Tiempo Estimado:** 5h |
| **Riesgo de Desarrollo:** Alta | **Tiempo Real:** 3h |
| **Descripción:** Los visitantes del portal podrán visualizar toda la información del servicio. | |
| **Prototipo de interfaz** | |

Tabla Historia de Usuario: Realizar Orden de servicio. Elaborada por el autor.

|  |  |
| --- | --- |
| **Número:** 19 | **Nombre del requisito:** Realizar orden del servicio |
| **Programador:** Dagoberto A Carralero | **Iteración Asignada:** 1 |
| **Prioridad:** Alta | **Tiempo Estimado:** 5h |
| **Riesgo de Desarrollo:** Alta | **Tiempo Real:** 5h |
| **Descripción:** Los visitantes del portal podrán realizar una orden del servicio directamente al WhatsApp del equipo de atención al cliente, ya con todos los detalles de la orden definidos. | |
| **Prototipo de interfaz** | |

## II.5 Estilo Arquitectónico

*“El diseño arquitectónico se ocupa de comprender cómo debe organizarse un sistema y diseñar la estructura general de ese sistema. En el modelo del proceso de desarrollo de software, el diseño arquitectónico es la primera etapa en el proceso de diseño de software. Es el vínculo crítico entre el diseño y la ingeniería de requisitos, ya que identifica los principales componentes estructurales de un sistema y las relaciones entre ellos. El resultado del proceso de diseño arquitectónico es un modelo arquitectónico que describe cómo se organiza el sistema como un conjunto de componentes que se comunican.” (Somerville, 2011)*

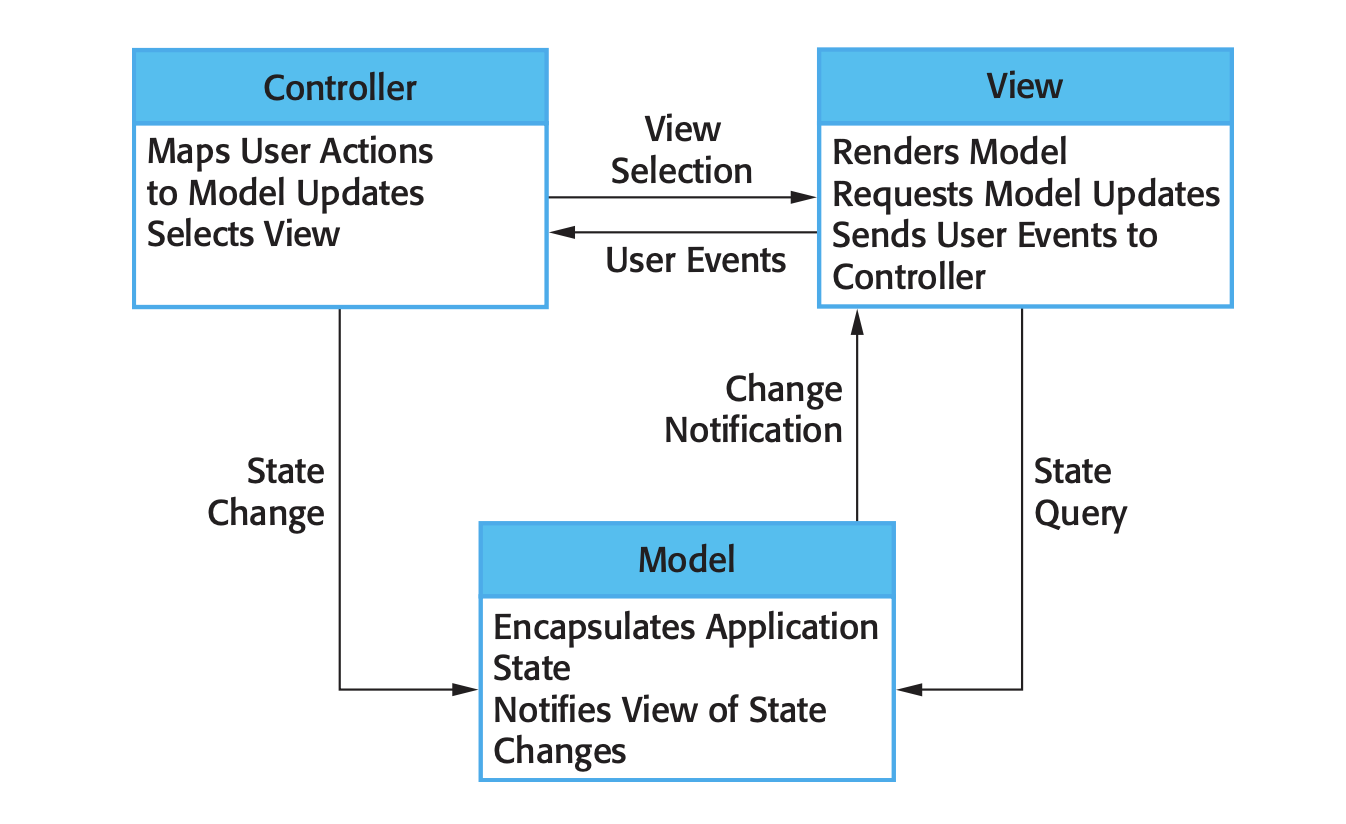


Figure . Modelo Vista Controlador. Tomado de SOFTWARE ENGINEERING Ninth Edition Ian Somerville, 2011

**El Framework Remix**

Remix es un web framework moderno, basado en React y especializado en aplicaciones de renderizado del lado del servidor, o Server Side Render en inglés, se acopla al patrón Modelo Vista Controlador, puede describirse su arquitectura de la siguiente forma:

**Vista**: Remix se centra en la interfaz de usuario y la capa de presentación, permitiendo a los desarrolladores definir la estructura y el diseño de las páginas web. Unifica la vista y el controlador en un solo código base, la ruta.

**Controlador**: Remix maneja las interacciones del usuario, la obtención de datos y la actualización de la vista en respuesta a esas interacciones. Proporciona una funcionalidad similar a un controlador para cada ruta.

**Modelo**: Si bien Remix no prescribe una capa de modelo específica, deja el modelado de datos, las bases de datos y la lógica empresarial a cargo del desarrollador para implementarlos según sea necesario. El modelo es responsabilidad del desarrollador de la aplicación que usa Remix. En este caso específico el modelo reside en el sistema de administración de PocketBase, que brinda una API REST para acceder al mismo y construir las vistas.

## II.6 Diagrama de Clases

Un diagrama de clases del diseño con estereotipos web tiene el mismo objetivo o propósito que un diagrama de clases tradicional, con la particularidad de que se emplea para el modelado de aplicaciones web. A continuación, se representa los diagramas de clases de diseño con estereotipos web realizado para el caso de uso Vista pública Artículo.

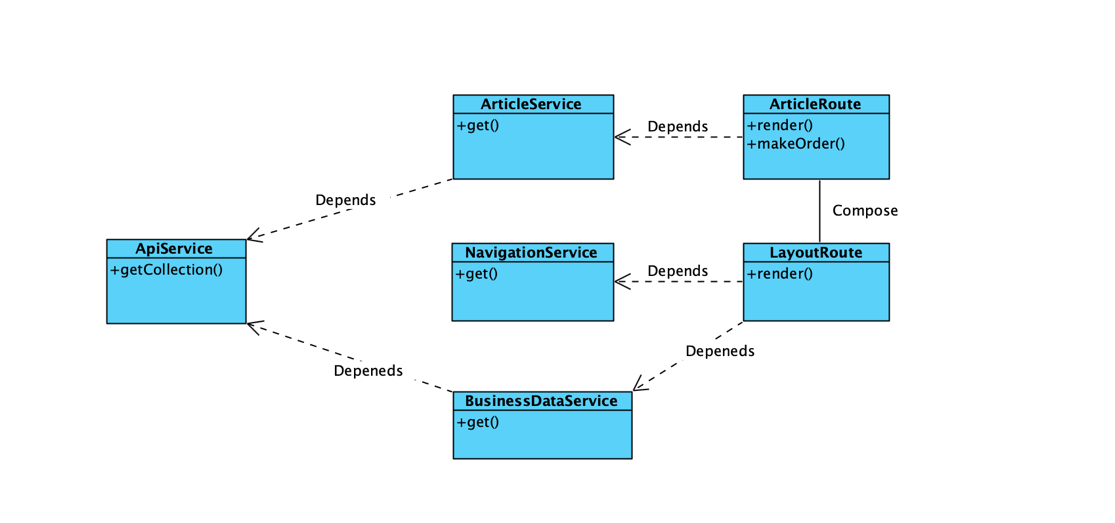


Figure . Diagrama de clases. Vista pública de artículo. Elaborada por el autor.

|  |  |  |
| --- | --- | --- |
| **No** | **Nombre de la Clase** | **Descripción** |
| 1 | APIService | Este servicio es el encargado de realizar las llamadas al API de contenido. |
| 2 | NavigationService | Este servicio maneja la información de la navegación principal del sitio. |
| 3 | ArticleService | Este servicio se encarga de, en colaboración con el APIService recuperar la información del artículo. |
| 4 | BusinessDataService | Este servicio maneja, en colaboración con el APIService, la información del negocio, como los datos de contacto |
| 5 | LayoutRoute | Esta Ruta maneja la presentación del layout general del sitio, incluye elementos navegacionales y la información de contacto. |
| 6 | ArticleRoute | Esta Ruta Maneja la presentación del artículo, así como la realización de la orden. |

## II.7 Patrones de diseño

Los patrones GRASP (por sus siglas en inglés, General Responsibility Assignment Software Patterns) describen los principios fundamentales de la asignación de responsabilidades a objetos. El nombre de se eligió́ para indicar la importancia de captar estos principios, si se quiere diseñar indicadamente el software orientado a objetos (Larman, 2003).

Los patrones GRASP utilizados en la solución propuesta fueron:

**Experto:** Este patrón define que la responsabilidad debe asignársele a la clase que tiene la información para concretar la tarea especifica. (Larman, 2003). Ejemplo de ello es la responsabilidad de Construir la Pagina de Artículo le es entregada a la Ruta producto, al contener esta toda la información necesaria para realizar la tarea.

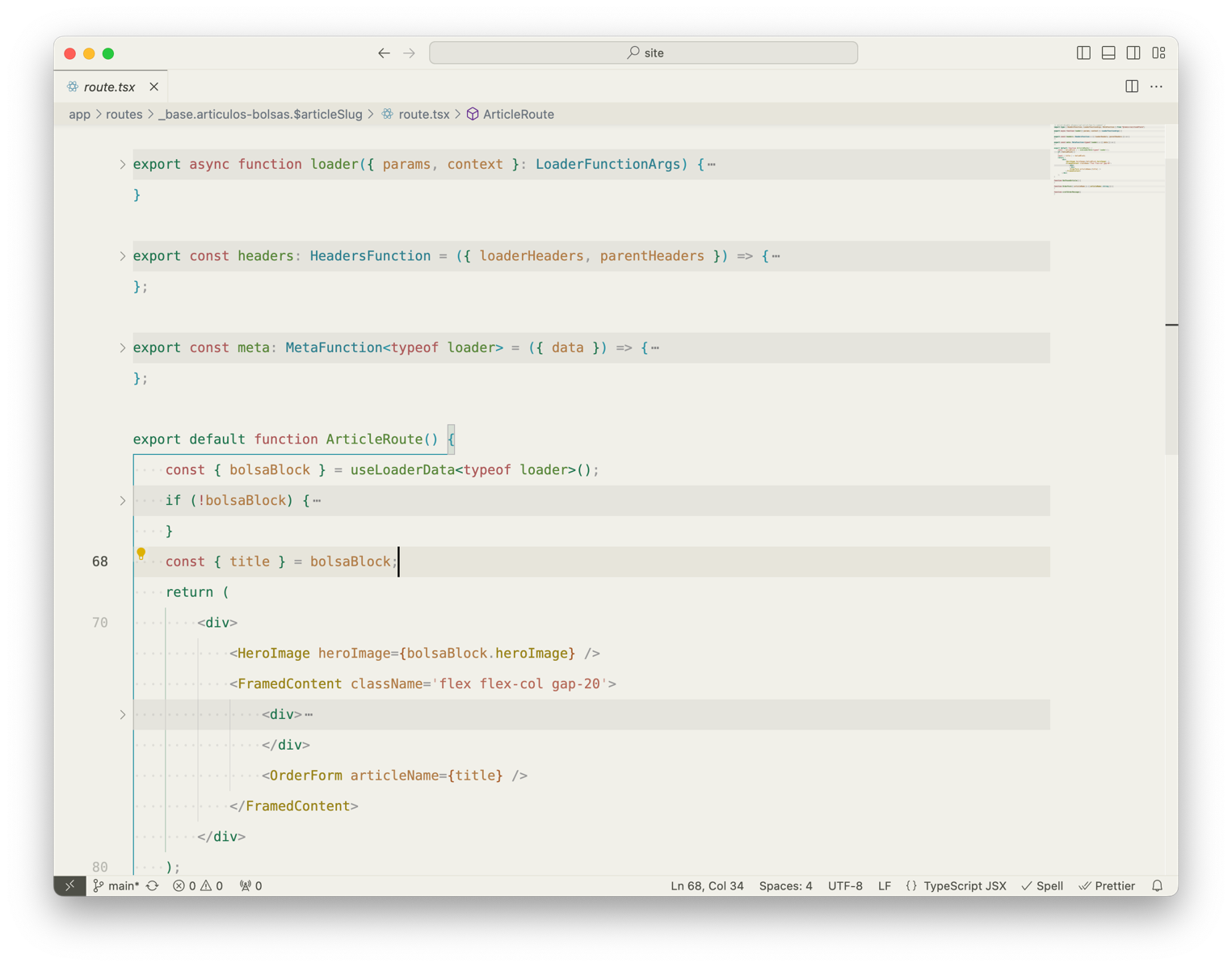


Figure . Ruta Articulo. Muestra del código fuente de la solución. Patrón experto. Tomada por el autor.

**Bajo Acoplamiento:** Define que se debe poder reutilizar las funcionalidades de las distintas clases, con un nivel de dependencia mínima. Ejemplo de este patrón en la presente solución es el hecho de que las diversas páginas pueden cambiar los URL que las invocan sin que esto implique cambio en el código que las construye, y se pueden realizar cambios en la implementación de una sección como la vista del servicio sin tener que cambiar las entidades de las que depende.

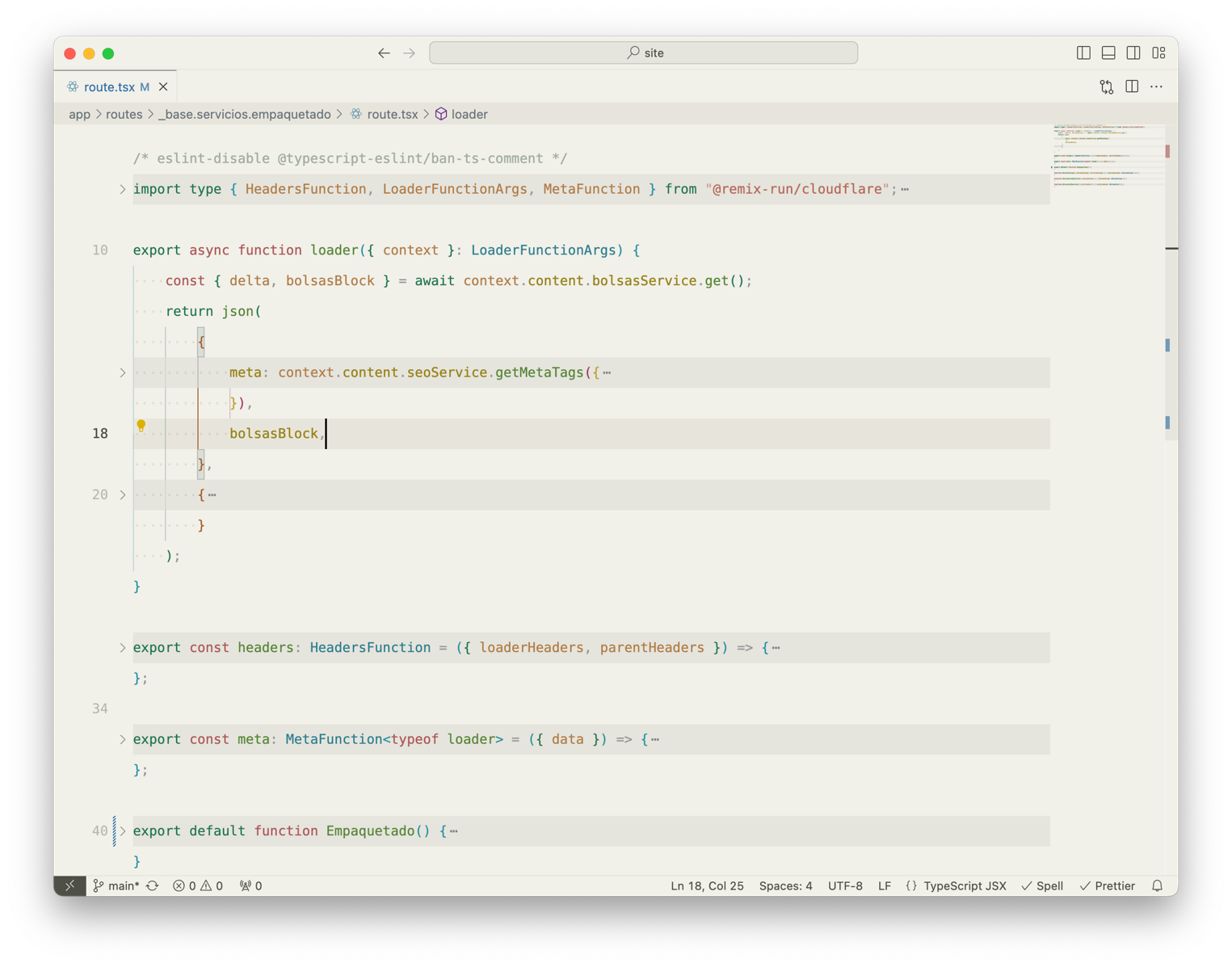


Figure . Ruta de Servicio de Empaquetado con bolsas de Papel. Patrón Bajo Acoplamiento. Tomada por el autor.

## II.8 Modelo de datos

Para construir las diversas funcionalidades extendió el modelo de datos base de sistema de contenido PocketBase, que utiliza una base de datos relacional, específicamente SQLite. El modelado y análisis de los modelos fueron realizados en la herramienta Visual Paradigm. A continuación, el esquema resultante:

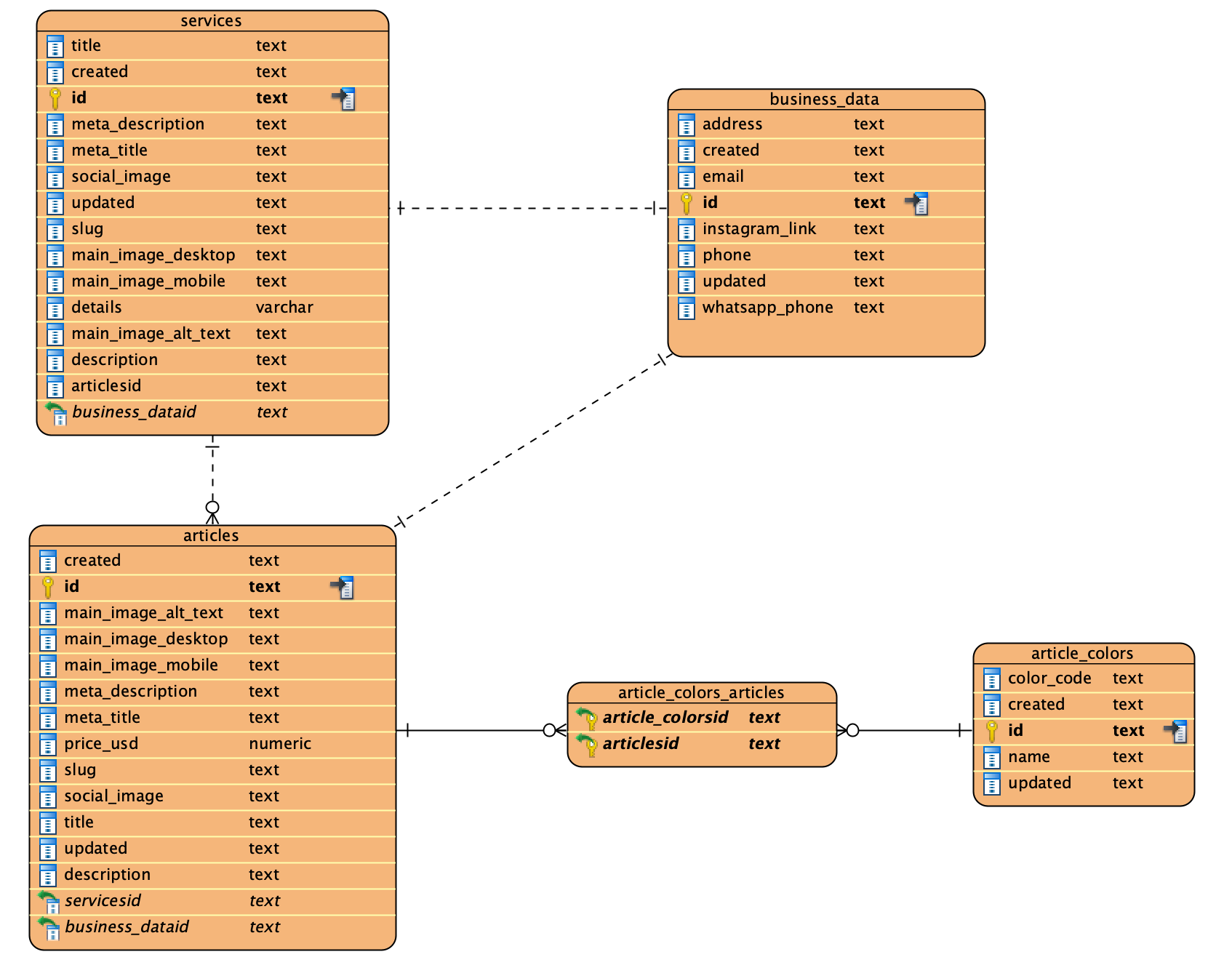


Figure . Modelo de datos. Elaborada por el autor.

**Descripción del modelo de la base de datos**

La tabla business\_data contiene la información de contacto del negocio. La tabla services la información de los servicios. La tabla articles contiene la información de los artículos. La table article\_colors contiene la información de las opciones de colores de los artículos, presenta relación de muchos a muchos con los artículos. Los servicios presentan relación de uno a muchos con los artículos. Y estos dos a su vez presentan relación de uno a uno con business\_data.

## II.9 Implementación

**Estándar de codificación**

En React, los nombres de componentes, deben empezar por mayúsculas, y atenerse a la convención de Pascal Case.

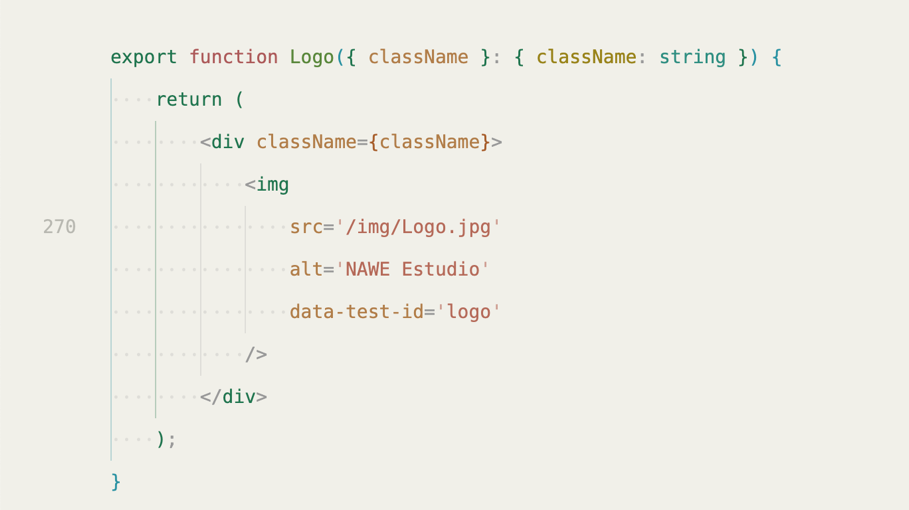


Figure . Ejemplo de normal Pasca Case en el trabajo de los componentes React. Elaborada por el autor.

**Diagrama de componentes**

Un diagrama de componentes descompone el sistema real en desarrollo en varios niveles altos de funcionalidad. Cada componente es responsable de un objetivo claro dentro del sistema completo y solo interactúa con otros elementos esenciales cuando es necesario. (Visual Paradigm)

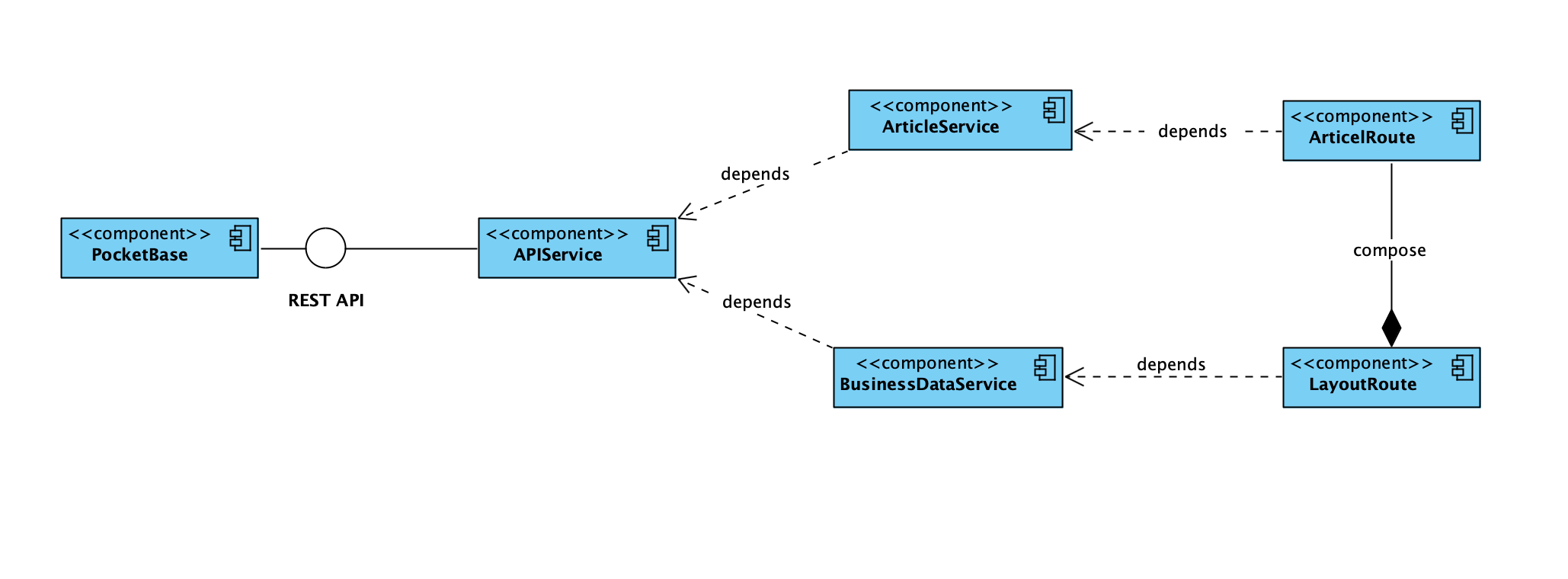


Figure . Diagrama de componentes. Caso de uso Vista pública de articulo. Elaborado por el autor.

**Diagrama de Despliegue**

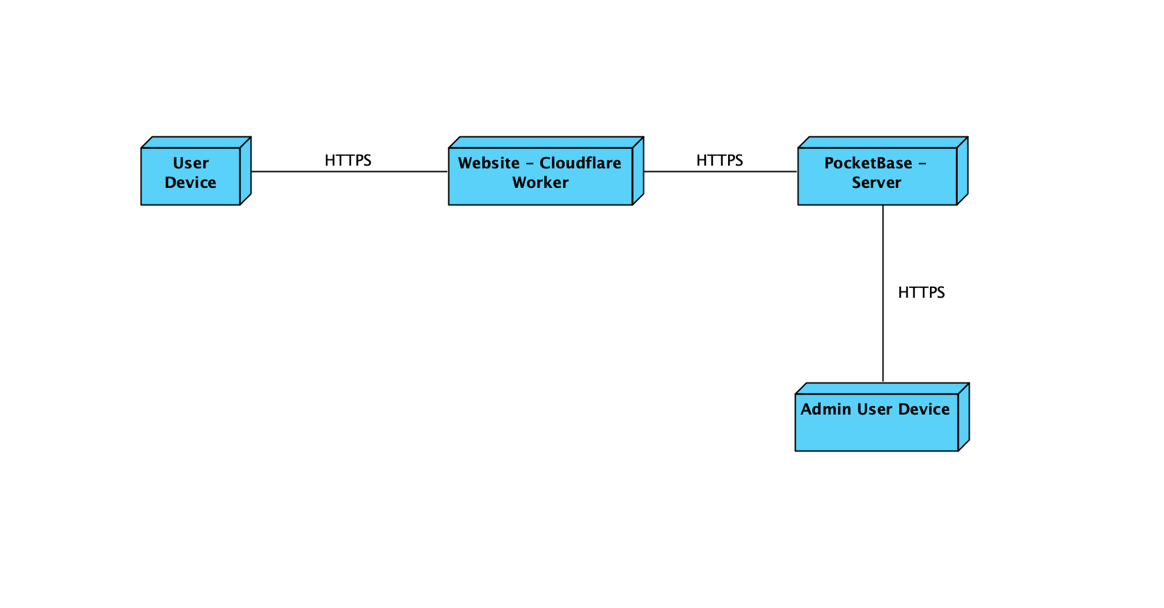
****

Figure 8. Diagrama de despliegue. Elaborado por el autor

Se tienen dos nodos que conforman el portal, PocketBase server, el sistema de contenido o CMS por sus siglas en inglés y la aplicación web que ante las peticiones de los usuarios se comunica con el backend vía HTTPS para obtener los datos de la vista en cuestión. También aparece reflejado el usuario administrativo que gestiona el contenido en el nodo PocketBase.

La aplicación web se desplego en Cloudflare Pages, servicio descrito con anterioridad en el Capítulo 1, este servicio es manejado por Cloudflare, y escala ante el tráfico de forma automática.

El nodo PocketBase se desplegó en usando Ubuntu Server 20.04, en un VPS de 1vCpu y 1Gb de RAM, en la práctica dada la eficiencia de Go y la manera en la que se programó, el consumo de RAM del servicio no sobrepasa los 20Mb, solo aumenta en casos de carga intensa como los que se simularon en la realización de esta investigación. Adicionalmente se configuro Caddy como proxy reverso, para manejar el tráfico público, y manejar el encriptado TLS que permite entablar comunicación segura entre el navegador cliente y el servidor, dado que el mismo maneja la concesión y auto renovación de los certificados requeridos.

## Conclusiones del capítulo

Tras llevar a cabo el análisis de la propuesta de solución y haber generado los artefactos que contempla la metodología AUP UCI en el escenario 4, es posible extraer las siguientes conclusiones:

La elaboración de una propuesta de solución ha permitido dar cumplimiento al objetivo planteado en la investigación. La extracción de requisitos funcionales y no funcionales ha posibilitado la creación de un portal web para la empresa NAWE acorde a sus necesidades específicas. La utilización de patrones de diseño ha permitido desarrollar una solución robusta, en consonancia con los estándares internacionales.

En definitiva, el proceso metodológico seguido, enmarcado en la metodología AUP UCI, ha resultado ser una herramienta eficaz para alcanzar los fines propuestos, dando lugar a una propuesta de solución sólida y adaptada a los requerimientos de la organización. La generación de los artefactos pertinentes ha sido clave para lograr un resultado satisfactorio, que cumple con las expectativas planteadas en la investigación.

# CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

En este capítulo se desarrolla evaluación de la calidad y fiabilidad de los resultados en el desarrollo del portal, se realiza trazando una estrategia de pruebas que aplica las disciplinas de pruebas internas y de aceptación definidas por la metodología que rige el desarrollo de la solución propuesta. Este proceso tiene como objetivo verificar y revelar la calidad del producto antes de su entrega al cliente, asegurando así el cumplimiento de los estándares y requisitos establecidos.

## III.1 Pruebas de sistema

*“Las pruebas de sistema durante el desarrollo implican integrar componentes para crear una versión del sistema y luego probar el sistema integrado. Las pruebas de sistema verifican que los componentes sean compatibles, interactúen correctamente y transfieran los datos adecuados en el momento apropiado a través de sus interfaces.”* *(Somerville, 2011).*

**Pruebas de extremo a extremo**

La prueba de extremo a extremo (end-to-end) es una metodología utilizada en el ciclo de vida de desarrollo de software (SDLC) que prueba la funcionalidad de una aplicación desde el inicio hasta el final en circunstancias de la vida real.El objetivo de las pruebas de extremo a extremo es simular cómo se vería un escenario de usuario real.

Se desarrolló una suite de este tipo de pruebas, usando la herramienta Cypres, que de manera automatiza controla el navegador y sigue las instrucciones de acciones que haría un usuario real. Logrando simular con gran nivel de fidelidad los escenarios reales en los que los usuarios del portal interactuaran con el mismo. Y la prueba actúa sobre todas las capas, desde la interfaz, los servicios que la soportan, así como el correcto funcionamiento del sistema de contenido. No se encontraron errores.

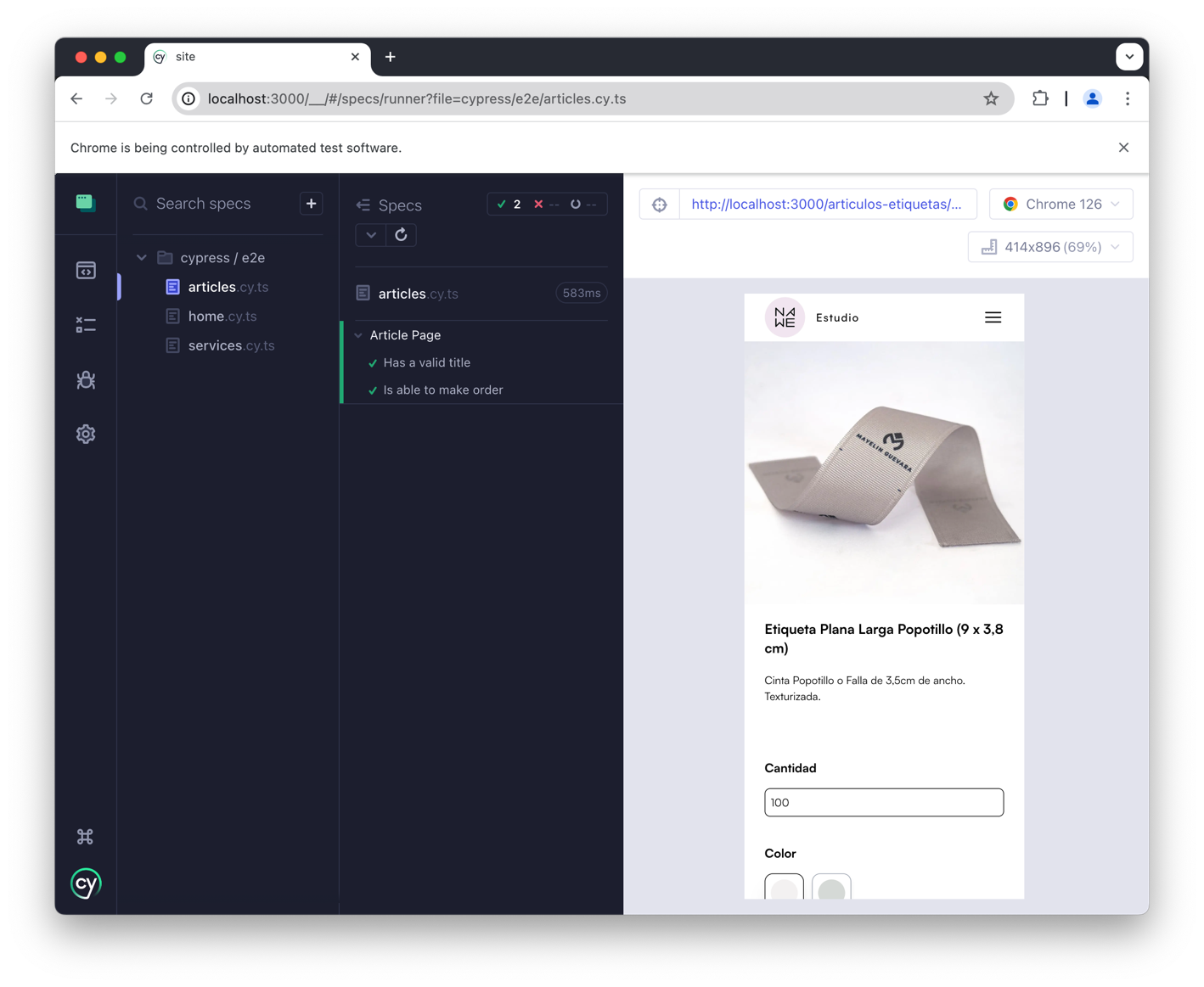


Figure . Prueba de extremo a extremo. Caso de uso de Vista publica de Articulo. Elaborada por el autor.

## Pruebas de rendimiento, carga y stress.

Estas pruebas permiten acertar el rendimiento real que presenta el sistema, y su capacidad de manejar la carga. Usando la herramienta loadles, se simulo un escenario en el que 120 personas intentan acceder al portal de forma concurrente. Se obtuvieron resultados satisfactorios resumidos a continuación:

|  |  |
| --- | --- |
| Nivel de concurrencia | 120 |
| Número de peticiones | 1000 |
| Tiempo de respuesta para el 50% de las peticiones | 575 ms |
| Tiempo de respuesta para el 100% de las peticiones | 984 ms |
| Peticiones por segundo efectivas | 202 |

Figure . Prueba de rendimiento carga y stress. Elaborada por el autor.

Se comprobó que el portal cumple con los requisitos de rendimiento, es por lo menos, entre 2 y 5 veces más rápido que el tiempo máximo de respuesta que fue levantado como requisito.

## III.2 Pruebas de aceptación

Se le realizó a la presente propuesta de solución el método de caja negra mediante la técnica de partición equivalente. Se tomó como base todos los requisitos funcionales, generándose un total de 4 Diseños de Casos de Prueba (DCP). A continuación, se muestran los resultados obtenidos una vez aplicadas las pruebas:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. Iteración | NC | Funcionalidad | Interfaz | Correspondencia | Resueltas |
| 1 | 6 | 2 | 4 | 0 | 6 |
| 2 | 2 | 0 | 2 | 0 | 2 |
| 3 | 0 | 0 | 0 | 0 | 0 |
| Total | 8 | 2 | 6 | 0 | 8 |

Figure . Pruebas de Aceptación. Elaborada por el autor.

Las causas de las no conformidades fueron:

* Interfaz: Algunos elementos no se correspondían con el diseño gráfico pautado.
* Funcionalidad: Algunos elementos interactivos de formularios de órdenes no procesaban de forma adecuada la información y esta no llegaba a la orden final.

|  |  |  |
| --- | --- | --- |
| CASO DE PRUEBA DE ACEPTACIÓN | | |
| Historia de Usuario | 19 Realizar orden del servicio | |
| Descripción | El portal debe permitir al usuario realizar una orden del servicio | |
| Condiciones de ejecución: | Al navegar a la página del servicio, el usuario verá un formulario que le permitirá realizar una orden. | |
| Escenarios de prueba: | Flujo del escenario: | Resultados esperados: |
| EP1 | Insertar Campos correctos | Se procede a realizar la orden |
| EP2 | Dejar campos obligatorios vacíos | Se notifica que los campos deben ser rellenados |

Figure Caso de prueba Realizar Orden. Elaborada por el autor.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Descripción | Superficie | Cantidad Total | Locación1 | Tamaño | colores | Respuesta del Portal | Resultado de la Prueba |
| 1 | Insertar campos correctos | V  T-Shirt | V  20 | V Frente | V 20\*10 | V  2 | Permite realizar la orden | Exitosa, permite realizar la orden |
| 2 | Dejar campos vacíos | I  nulo | V 20 | I nulo | I nulo | I nulo | Notifica al usuario que debe llenar los campos | Exitosa, no procede con el flujo de la orden, mas, notifica al usuario que debe llenar los campos. |

Figure . Escenarios de Prueba. Elaborada por el autor.

## Conclusiones del capítulo

Del desarrollo del presente capítulo se desprenden las siguientes conclusiones de notable relevancia:

La ejecución de las pruebas de caja funcionales, de sistema y aceptación, ha permitido constatar el correcto funcionamiento del código de la aplicación, validar el cumplimiento de los requisitos establecidos, así como determinar la aceptación por parte del cliente de manera fehaciente.

En suma, el trabajo desarrollado a lo largo de este capítulo ha permitido obtener producto de calidad y alineado plenamente con las expectativas y necesidades del cliente.

# CONCLUSIONES FINALES

• El marco teórico de la investigación, mediante el estudio y el análisis de los principales referentes teóricos en los que se sustenta la investigación, dio una idea de la estructura y los componentes de los sistemas homólogos, que fueron tomados como referentes en la construcción de la solución.

• La definición de las herramientas, metodologías y tecnologías propiciaron la implementación exitosa del sistema, cumpliendo de esta forma con el objetivo general.

• Los artefactos de Modelado de Negocio, Requisitos y Análisis y Diseño posibilitaron en

gran medida un correcto entendimiento y creación del diseño de la propuesta de

solución que contenga los requerimientos estéticos que desea el cliente.

• Las pruebas funcionales permitieron validar el correcto funcionamiento del

sistema desarrollado, logrando así los objetivos propuestos en la investigación.

# RECOMENDACIONES

Se plantean las siguientes recomendaciones:

* Adicionar una función de analítica, para poder conocer la tendencia y origines de las visitas de los usuarios al portal.
* Adicionar una sección de blog donde crear artículos específicamente dirigidos al público meta de la empresa, con contenido relevante y de alta calidad.

# REFERENCIAS BIBLIOGRÁFICAS

The Power of Website and social media for Strengthening Brand Image, E-WoM, and Purchase Decision Mochamad Nurhadi, Tatik Suryani, Abu Amar Fauzi February 2023 Journal of Economics Business and Accountancy Ventura. Recuperado de Research Gate 29 febrero 2024 <https://www.researchgate.net/publication/369801822_The_Power_of_Website_and_Social_Media_for_Strengthening_Brand_Image_E-WoM_and_Purchase_Decision>

Vladimir Alfonso Rodríguez, Edelmis Chapis Cabrera.IMPORTANCIA DE LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES, EL INTERNET Y LAS REDES SOCIALES EN EL MEJORAMIENTO Y DESARROLLO DE LAS EMPRESAS. Universidad de Cienfuegos. Recuperado marzo 1 2024 de <https://www.eumed.net/rev/ce/2019/1/tecnologias-informacion-empresas.html>

Kasey Kaplan. Revista Forbes Recuperado marzo 1 2024 en <https://www.forbes.com/sites/theyec/2020/02/03/why-every-business-needs-a-website/?sh=1e05947c6e75>

D. Sharma, R. Shukla, A. K. Giri and S. Kumar, "A Brief Review on Search Engine Optimization," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2019, pp. 687-692, doi: 10.1109/CONFLUENCE.2019.8776976. Recuperado en IEEE Xplore marzo 2 2024 <https://ieeexplore.ieee.org/abstract/document/8776976>

Arévalo, G. A. y Álvarez, M. I. (2019). Visibilidad en línea: análisis de los hoteles de Tunja (Colombia). Turismo y Sociedad, xxv, pp. 253-271. doi: <https://doi.org/10.18601/01207555.n25.12>

CERRON, 2017. IMPLEMENTACIÓN DE UN PORTAL WEB MEDIANTE LA METODOLOGÍA RUP PARA OPTIMIZAR LOS PROCESOS DE PRESTACIÓN DE SERVICIOS DE LA EMPRESA PROGRAMADORES WEB PERÚ S.A.C. UNIVERSIDAD DE CIENCIAS Y HUMANIDADES de Perú. <https://repositorio.uch.edu.pe/bitstream/handle/20.500.12872/93/CD-TISI-018-2017.pdf?sequence=1&isAllowed=y>

GARCERANT, I. Modelo de Dominio. Tecnología y Synergix. Recuperado de <https://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>

REHKOPF. Historias de usuario con ejemplos y plantilla. Recuperado de <https://www.atlassian.com/es/agile/project-management/user-stories>

Craig Larman. 2003. Applying UML and Patterns. Recuperado de <https://books.google.com.cu/books?redir_esc=y&id=r8i-4En_aa4C&q=grasp#v=onepage&q&f=false>

Visual Paradigm. Recuperado de <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-component-diagram/>

Sistema de gestión administrativa de proyectos de Investigación, Desarrollo e Innovación asociados a programas. Gilberto Veitia Urbay, Rixon Riverol Arévalo. 2022.Universidad de las Ciencias Informáticas.

# ANEXOS