Universidad de La Habana
Facultad de Matemática y Computación

Título de la tesis

Autor:

Nombre del autor

Tutores:

Nombre del primer tutor
Nombre del segundo tutor

Trabajo de Diploma

presentado en opción al título de
Licenciado en (Matemática o Ciencia de la Computación)

Fecha

github.com/username/repo

Dedicación

Agradecimientos

Agradecimientos

Opinión del tutor

Opiniones de los tutores

Resumen

Resumen en español

Abstract

Resumen en inglés

Índice general

Índice de figuras

Ejemplos de código

Introducción

Desde la prehistoria, la relación entre la humanidad y las plantas ha sido fundamental para la supervivencia y el desarrollo cultural de las civilizaciones. Las plantas han servido como fuente de alimento, medicina y recursos básicos, un conocimiento que se ha transmitido a través de generaciones, incorporándose en muchas culturas, donde su uso se extiende desde el tratamiento de enfermedades hasta prácticas espirituales y rituales.

El surgimiento de la agricultura durante el período neolítico revolucionó la historia, transformando el modo de vida y la supervivencia humana [11]. Gracias a la acumulación de conocimientos previos, las sociedades comenzaron a cultivar plantas con fines específicos, tanto culinarios como medicinales. Esta práctica sentó las bases del conocimiento en antiguas civilizaciones como la griega y la romana, donde las plantas no solo formaban parte de la medicina, sino de la mitología y la literatura.

En el contexto iberoamericano, las plantas medicinales han adquirido un valor estratégico tanto cultural como económico. La herencia de tradiciones ancestrales ha contribuido al auge del «consumo verde» a nivel mundial, revitalizando el interés por los remedios naturales y reconociendo así el potencial terapéutico de la naturaleza [20].

La medicina natural y tradicional ha sido, desde hace siglos, un elemento clave en la cultura y la identidad del pueblo cubano. Ante cualquier dolencia, es común encontrar quien recomiende un remedio casero, cultive su propio huerto medicinal o domine el arte de preparar cocimientos con propiedades terapéuticas. Estas prácticas, profundamente arraigadas, tienen sus raíces en la mezcla de tradiciones europeas, africanas y asiáticas, que confluían en el archipiélago desde la época colonial.

Con el paso de los años, esta riqueza cultural se integró en el sistema de salud

cubano, ganando un respaldo institucional a partir de eventos clave. La Organización Mundial de la Salud (OMS), al finalizar la Conferencia Internacional sobre Atención Primaria, celebrada en 1978, emitió su conocida Declaración de Alma Ata, la que entre diversas propuestas, realizó un llamado para incorporar las medicinas alternativas y terapias tradicionales con eficacia científicamente demostrada, a los Sistemas Nacionales de Salud [20]. Sin embargo, fue durante el "Período Especial", tras la caída del campo socialista en 1991, cuando el uso organizado de las plantas medicinales adquirió una nueva dimensión en Cuba. Frente a la escasez de medicamentos en el país, se desarrolló un Programa de Plantas Medicinales, estableciendo bases científicas para su producción y utilización. Este programa no solo respondió a una necesidad urgente, sino que también consolidó una tradición médica que fusionaba raíces populares y científicas [23].

El conocimiento acumulado a lo largo de la historia no habría sido posible sin la labor de destacados investigadores, como Juan Tomás Roig y Mesa, quien en el prólogo de su libro *"Plantas medicinales, aromáticas o venenosas de Cuba"* [19], publicado en 1945, detalla su propósito de documentar y sistematizar el uso de especies vegetales con aplicaciones medicinales, culturales y económicas en Cuba.

Según Roig, su obra pretende ofrecer información *"lo más completa y exacta que sea posible acerca de nuestras plantas medicinales o venenosas"* y, además, servir como fuente de consulta para estudiantes y científicos en disciplinas como botánica, farmacia y medicina, estimulando el estudio metódico de la flora médica del país. Además, subraya el potencial impacto económico de estas plantas al señalar que su estudio podría conducir *"a la creación de una industria farmacéutica, que podría proporcionar trabajo a muchos obreros en el campo, y empleo a numerosas personas en los laboratorios y oficinas comerciales"*. Esta visión demuestra el compromiso del autor no solo con la preservación del conocimiento, sino también con el desarrollo económico y social basado en los recursos naturales.

El esfuerzo de Roig por incluir aspectos como los nombres científicos y vulgares, descripciones botánicas y aplicaciones medicinales demuestra su interés por hacer el conocimiento accesible tanto para científicos como para el público en general. Su obra trasciende como un legado fundamental en la sistematización del uso de plantas medicinales en Cuba, contribuyendo al conocimiento científico y práctico, e inspirando iniciativas dedicadas a la conservación y estudio de la biodiversidad cubana.

Los esfuerzos históricos de preservación y sistematización del conocimiento sobre la flora cubana culminaron en la fundación del Jardín Botánico Nacional de Cuba el 24 de marzo de 1968. Esta institución emblemática forma parte de la Universidad de La Habana y combina la conservación de la flora con la educación ambiental. Se extiende por aproximadamente 600 hectáreas y alberga más de 4,000 especies vegetales, convirtiéndose en uno de los jardines botánicos más grandes y completos del mundo [8].

El Jardín Botánico Nacional tiene como misión principal promover el conocimiento sobre la flora cubana y tropical, enfatizando la importancia de la conservación ambiental. Su enfoque educativo busca involucrar a la población en general, ofreciendo un espacio donde se combinan actividades recreativas con la enseñanza sobre el medio ambiente, perpetuando los ideales de preservación, educación y aprovechamiento sostenible de los recursos naturales [14].

En este contexto, la integración de soluciones tecnológicas que permitan gestionar y analizar la información sobre plantas medicinales cubanas se vuelve una necesidad estratégica. La sistematización digital del conocimiento no solo facilitaría el acceso a datos esenciales para investigadores y estudiantes, sino que también podría impulsar nuevas líneas de investigación en biotecnología, farmacología y sostenibilidad ambiental.

Bajo esta visión, el Jardín Botánico Nacional busca desarrollar una solución computacional que integre y gestione la información contenida en la obra de Juan Tomás Roig Mesa. Esta herramienta facilitará el acceso y organización de datos sobre las plantas medicinales cubanas, contribuyendo al reconocimiento de su importancia cultural y científica. Además, permitirá a los investigadores y especialistas disponer de una base estructurada para profundizar en el estudio y aplicación de estas plantas en áreas de interés económico y social, fortaleciendo el rol del Jardín Botánico como un centro de referencia en el ámbito de la medicina natural y la biodiversidad.

La situación descrita permite definir el siguiente **problema científico**: el diseño e implementación de una solución computacional que permita extraer la información que ofrece la obra de Juan Tomás Roig y Mesa, en concreto: *"Plantas medicinales, aromáticas o venenosas de Cuba"*; y posteriormente facilitar el acceso y manipulación de la información científica presente en la misma.

A partir del problema planteado, se enuncia la siguiente **hipótesis**: la implementación de un sistema computacional para la gestión de la información científica basado en la obra de Juan Tomás Roig y Mesa mencionada anteriormente, bajo la concepción del desarrollo web y que utilice técnicas de Procesamiento de Lenguaje Natural para la manipulación de la información, permitirá extraer el conocimiento científico de la obra de Roig y resultará en un sistema de gestión de la información que brinde facilidades en cuanto al acceso y manipulación de los datos.

El **objetivo general** de este trabajo de diploma es implementar técnicas de Procesamiento de Lenguaje Natural para la obtención y estructuración de la información contenida en la obra *"Plantas medicinales, aromáticas o venenosas de Cuba"* de Juan Tomás Roig y Mesa, y su posterior manejo e integración como base inicial de conocimiento en un sitio web nacional para la gestión de la información sobre plantas medicinales cubanas, administrado por el Jardín Botánico Nacional de Cuba.

Para alcanzar el cumplimiento del objetivo general, se pueden definir un conjunto de objetivos específicos:

1. Profundizar en los elementos teórico-conceptuales y prácticos vinculados al procesamiento del lenguaje natural y el uso de bases de datos relacionales, que permitan la fundamentación teórico-metodológica de la propuesta.

2. Diseñar los modelos de datos y procesos que respondan a los requerimientos informacionales en función de los intereses de los especialistas de Botánica, Farmacia, Medicina, Agronomía y Veterinaria.

3. Diseñar, implementar y evaluar un prototipo de línea de trabajo que permita la digitalización, estructuración y almacenamiento de la información contenida en la obra de Juan Tomás Roig y Mesa respecto a plantas medicinales cubanas.

4. Diseñar, implementar y evaluar un prototipo de solución computacional que permita la gestión y organización de los datos, además de enriquecer la visualización de los resultados.

A continuación se expone la estructura del documento, que consta de otros tres capítulos en los que se detallan las bases, el diseño y la implementación de la solución adoptada.

Capítulo 1 - *"Marco teórico-conceptual"*: Aborda las bases teóricas que fueron objeto de estudio para fundamentar los métodos utilizados durante el diseño y puesta en práctica de la solución computacional al problema presentado.

Capítulo 2 - *"Concepción y diseño de la solución"*: Expone y caracteriza las elecciones en cada parte del proceso de concepción y diseño de la solución computacional, desde el punto de vista teórico.

Capítulo 3 -*Ïmplementación y experimentación"*: Detalla los aspectos técnicos de la solución práctica, y se evalúan los resultados.

Posteriormente se presenta un apartado con las conclusiones del trabajo realizado, así como la sección de recomendaciones, donde se proponen ideas que pueden ser objetivo de investigación para extender la funcionalidad del software, y dotarlo de un mayor valor de uso.

Para finalizar, se incluyen las referencias bibliográficas que respaldan la base científica de la solución propuesta, así como los anexos.

Capítulo 1

Marco teórico-conceptual

La creciente necesidad de preservar y sistematizar el conocimiento sobre la flora cubana requiere del uso de tecnologías computacionales para facilitar su organización, acceso y análisis. La digitalización de datos no solo mejora la estructuración de la información, sino también su consulta y utilización en la investigación científica y la práctica médica. En este contexto, el Procesamiento de Lenguaje Natural (NLP, por sus siglas en inglés) y las bases de datos se presentan como herramientas clave para gestionar de manera eficiente y dinámica este conocimiento. El presente capítulo explora los fundamentos teóricos y metodológicos que respaldan el desarrollo de una solución computacional destinada a integrar y gestionar dicha información.

1.1. Plantas: una visión general desde la medicina tradicional

Las plantas son la base de la vida en la Tierra. Son los principales productores de oxígeno y alimento para la mayoría de los ecosistemas, jugando un papel fundamental en el equilibrio de nuestro planeta. Además, desde tiempos antiguos, las plantas han sido mucho más que alimento: han representado una fuente inagotable de remedios naturales, esenciales para la salud y el bienestar humano.

La flora de Cuba es un tesoro de la naturaleza, rica en diversidad y con alto nivel de endemismo. La ubicación geográfica de la isla ha generado una evolución

única de las plantas, creando un gran número de especies que no se encuentran en ninguna otra parte del mundo. La isla ha actuado como un laboratorio natural, donde la flora ha podido desarrollarse a lo largo de millones de años, dando lugar a una gran variedad de formas y colores. Algunas de estas plantas están adaptadas a las condiciones específicas de cada región de la isla, desde las zonas costeras hasta las montañas. La flora cubana es un ejemplo fascinante de la capacidad de la naturaleza para generar vida en entornos únicos e invita a la exploración y la conservación de este patrimonio natural [22].

Las plantas medicinales han sido, y continúan siendo, una fuente invaluable de compuestos químicos con aplicaciones diversas en la salud humana. Desde aliviar síntomas menores hasta tratar enfermedades complejas, sus usos abarcan un amplio espectro terapéutico, incluyendo analgésicos, antiinflamatorios, antibióticos y tratamientos para afecciones cardiovasculares, digestivas y respiratorias.

Según el Dr. Francisco J. Morón Rodríguez en su artículo *"Necesidad de investigaciones sobre plantas medicinales"* [25], el botánico norteamericano James A. Duke estima que menos del 1 % de las más de 90 mil especies de plantas de bosques de América Latina han sido investigadas químicamente. Además, el autor expresa:

> *"Las cifras del Dr. Duke, nos hacen reflexionar en que apenas conocemos las potencialidades terapéuticas de las plantas medicinales, el clásico símil del iceberg, para expresar la relación entre lo que conocemos o vemos que es mucho menor que lo oculto o desconocido, resulta insuficiente, porque esos témpanos de hielo flotando a la deriva muestran aproximadamente un cuarto de su masa total."*

El autor, también subraya el prólogo del libro *"Plantas medicinales, aromáticas o venenosas de Cuba"* [19] de Juan Tomás Roig y Mesa, donde hace un llamado a la comunidad científica a comprobar, mediante investigaciones multidisciplinarias, los efectos de las plantas medicinales tradicionales.

Si bien la investigación científica continúa explorando y validando sus propiedades, la tradición ancestral en el uso de plantas medicinales ofrece un rico acervo de conocimiento para el desarrollo de nuevos fármacos y terapias.

La obra *"Plantas medicinales, aromáticas o venenosas de Cuba"* [19] de Juan Tomás Roig y Mesa representa un hito fundamental en el estudio de la flora medicinal cubana. Su exhaustiva compilación de información de las plantas, junto con descripciones botánicas detalladas y usos tradicionales, constituye una base inestimable para investigaciones posteriores. La obra de Roig no solo documentó un vasto conocimiento popular sobre las plantas medicinales cubanas, sino que también sentó las bases para la investigación científica rigurosa en este campo, dejando un legado invaluable para la fitoterapia y la conservación del patrimonio botánico de la isla.

Como parte del prólogo a la primera edición de la obra, Roig hace un llamado a los hombres de ciencia para que emprendan el estudio metódico de la flora médica y toxicológica cubana. Además, resalta la utilidad de algunas secciones pensando en una posible cultivación a escala comercial para la exportación.

A pesar de los avances científicos logrados en más de 60 años desde la primera publicación de la obra de Roig, la afirmación del Dr. en Ciencias Biológicas Víctor R. Fuentes Fiallo – *"el viejo sueño del doctor Juan Tomás Roig sigue siendo eso: un sueño"* – [9] pone de manifiesto que la ambiciosa visión que tenía Roig, aún no se ha materializado plenamente.

1.1.1. Nomenclatura y clasificación

Todas las especies de seres vivos conocidas por la humanidad se nombran según un sistema científico que regula la nomenclatura biológica. Este sistema estandarizado, establecido por organismos internacionales, busca asegurar que cada especie tenga un nombre único y universalmente aceptado, lo que facilita su identificación y clasificación dentro de la comunidad científica. En el caso de las plantas, la nomenclatura científica está regulada por el Código Internacional de Nomenclatura para Algas, Hongos y Plantas [17]. Cada nombre científico debe estar en latín y consta de tres partes fundamentales: un nombre genérico que identifica el *género*, un epíteto específico que distingue a la *especie* dentro del género y el nombre del autor o *autores* que describe oficialmente la especie.

Además de las tres categorías anteriores, algunos nombres científicos pueden incluir otras categorías para definir subgrupos dentro de una especie. Cuando una especie tiene diferencias geográficas, morfológicas o ecológicas significativas pero aún

pertenece a la misma especie, se clasifica en *subespecies*. La *variedad* es una categoría que agrupa individuos con variaciones de carácter local que pueden aparecer dentro de una misma población. La *forma* es una categoría taxonómica que representa una modificación ocasional de la especie, asociada o no a la distribución geográfica. La *familia* es un rango de clasificación taxonómica, que constituye un conjuntos de géneros entre los que se reconocen varios caracteres comunes importantes, y una *subfamilia* es una subdivisión dentro de una familia. [32]

Las plantas a menudo reciben diferentes nombres vulgares según la región y la cultura, reflejo de observaciones locales sobre sus propiedades, apariencia o historia. Esta diversidad de nombres, transmitidos oralmente, enriquece el conocimiento tradicional, pero puede complicar su identificación científica.

1.2. Procesamiento del Lenguaje Natural: fundamentos y aplicaciones

La complejidad y diversidad del lenguaje humano nos diferencia del resto de las especies. Nuestra capacidad de comunicarnos a través del lenguaje ha sido fundamental para el desarrollo de la civilización, permitiendo la transmisión de conocimiento cultural, científico y tecnológico.

El Procesamiento del Lenguaje Natural (NLP, por sus siglas en inglés) es un campo de la ciencia de la computación que busca dotar a las computadoras de la capacidad de entender, interpretar y generar lenguaje humano. Esto es importante porque nos permite que las computadoras puedan comunicarse con nosotros de forma natural, aprender de la inmensa cantidad de información escrita en nuestro idioma, y profundizar nuestra comprensión científica de cómo funciona el lenguaje [26].

Como parte del desarrollo de esta rama de la ciencia de la computación, y anterior al auge de los últimos años de los randes modelos de lenguaje (LLM, por sus siglas en inglés), se identificaron tareas comunes que buscan permitir a las computadoras entender, interpretar y generar lenguaje humano. Siguiendo la convención establecida en el libro *"Artificial intelligence: A modern approach"* de Russell y Norvig [26], se conservará la terminología original en inglés para describir cada una de ellas.

- **Speech recognition**: El reconocimiento de voz consiste en convertir el habla humana en texto escrito. Los sistemas modernos tienen una tasa de error bastante baja (entre un 3 % y 5 %), comparable a la de un transcriptor humano.
- **Text-to-speech**: Es el proceso inverso al reconocimiento de voz: transformar texto escrito en habla. El objetivo es que la voz generada suene natural, con pausas y énfasis apropiados. Se está avanzando en la creación de voces con diferentes acentos e incluso imitando voces de celebridades.
- **Machine translation**: La traducción automática implica la traducción de texto de un idioma a otro. Los sistemas de traducción aprenden a partir de grandes conjuntos de textos en dos idiomas (corpus bilingües) y se enfocan en traducir no solo palabras individuales, sino también el significado y la estructura gramatical de las oraciones.
- **Information extraction**: Consiste en extraer información específica de un texto. Por ejemplo, se puede utilizar para resumir textos, extraer direcciones de páginas web o información meteorológica de informes o datos de tablas. La dificultad de la tarea depende de la estructura del texto; un texto bien estructurado es más fácil de procesar que un texto no estructurado.
- **Information retrieval**: La recuperación de información se enfoca en encontrar documentos relevantes para una consulta dada. Los motores de búsqueda de internet son un ejemplo claro de sistemas que realizan esta tarea a gran escala. El objetivo es devolver los documentos más pertinentes a la búsqueda del usuario.
- **Question answering**: A diferencia de la recuperación de información, esta tarea busca responder preguntas específicas en lugar de simplemente mostrar una lista de documentos. Los sistemas modernos utilizan técnicas complejas para comprender el significado de la pregunta y encontrar la respuesta correcta en una base de datos de información o en Internet.

En años recientes, los LLM han revolucionado el campo del NLP al demostrar capacidades sobresalientes en tareas como la respuesta a preguntas, la traducción automática y la generación de texto.

El artículo *"Large Language Models on Wikipedia-Style Survey Generation: an Evaluation in NLP Concepts"* [10] analiza el impacto significativo de estos modelos, destacando su capacidad para generar texto coherente y contextualmente relevante, traducir idiomas y responder preguntas complejas, superando con creces a sistemas anteriores. Si bien este avance ha impulsado aplicaciones más sofisticadas y accesibles, también ha planteado nuevos desafíos relacionados con la eficiencia computacional, el sesgo en los datos y la ética de su uso.

1.2.1. Information Extraction (IE, por sus siglas en inglés)

En la era digital actual, nos enfrentamos a una inmensa cantidad de datos: 2.5 quintillones de bytes diariamente. Esta explosión de información, proveniente de fuentes tan diversas como las redes sociales y la literatura científica, ha hecho de la IE un campo crucial dentro del NLP. Se centra en la automatización del proceso de identificar y extraer información estructurada a partir de texto no estructurado o semiestructurado. Este proceso transforma datos complejos en formatos analíticamente útiles, facilitando la búsqueda, visualización y el aprovechamiento de la inmensa cantidad de conocimiento latente en el texto, con implicaciones significativas en diversas áreas como la inteligencia de negocios [21].

Desde sus inicios en la década de 1950, la IE ha evolucionado gracias a iniciativas como las Conferencias de Comprensión de Mensajes, logrando sistemas capaces de extraer información con precisión razonable, aunque con margen de mejora en cuanto a la complejidad del lenguaje y la inferencia [12]. Con el tiempo, se han desarrollado una serie de técnicas fundamentales que son esenciales en el campo de la IE [16]:

- **Named Entity Recognition**: Esta técnica consiste en identificar entidades (nombres de personas, organizaciones, lugares, fechas) en un texto. Se puede hacer usando reglas predefinidas, métodos estadísticos que analizan la probabilidad de que una palabra sea una entidad, o modelos de aprendizaje profundo que aprenden de grandes cantidades de texto.
- **Relation Extraction**: Aquí se busca identificar las conexiones entre las entidades nombradas. Se pueden usar reglas, modelos de aprendizaje automático que aprenden de ejemplos, o modelos que utilizan grandes bases de datos como

fuente de entrenamiento. Las redes neuronales también se aplican para clasificar las relaciones entre entidades.

- **Event Extraction**: Esta técnica consiste en identificar eventos que ocurren en un texto, como accidentes o reuniones, y los elementos involucrados (participantes, lugar, tiempo). Esto se puede lograr usando plantillas predefinidas, modelos de aprendizaje automático que aprenden a identificar eventos a partir de ejemplos, o redes neuronales que analizan la estructura del texto para comprender el evento.
- **Correference Resolution**: Se trata de identificar cuando diferentes palabras o frases en un texto se refieren a la misma entidad. Se usan reglas, modelos de aprendizaje automático que analizan características del lenguaje, o redes neuronales que aprenden a seguir las referencias a través del texto.
- **Template Filling**: Esta técnica consiste en extraer información específica de un texto para completar una plantilla predefinida. Se puede lograr utilizando reglas, modelos de aprendizaje automático que clasifiquen la información, o una combinación de ambos.
- **Open Information Extraction**: Esta técnica busca extraer información de una manera más flexible, sin necesidad de definir de antemano las relaciones que se buscan. Se basa en identificar patrones en el texto o mediante modelos estadísticos y de aprendizaje profundo para encontrar relaciones entre las palabras.

1.2.1.1. Template Filling

Anteriormente se ha mencionado el *Template Filling* como una técnica de extracción de información que utiliza una plantilla predefinida para estructurar la información extraída de un texto. Esta plantilla actúa como un molde, con espacios o 'slots' que deben ser rellenados con información específica extraída del texto.

En el libro *"Encyclopedia of Systems Biology"* [24], se aborda el tema de *"Template Filling, Text Mining"*, donde se resaltan y definen las dos componentes fundamentales que nombran esta técnica: la *plantilla* (template) y las *reglas de llenado* (fill rules).

Una plantilla es un esquema abstracto que se define en un dominio de interés, el que a su vez, determina la información genérica a extraer y el formato de la salida. Las reglas de llenado, por su parte, describen el proceso de extracción de la información, actuando como guía para completar la plantilla.

El diseño de una plantilla para extraer información depende del dominio de interés y la naturaleza de la tarea. En dependencia del tamaño y la complejidad del conjunto de datos a analizar, dos tipos de plantillas son las utilizadas comúnmente: las *plantillas planas* y las *plantillas orientadas a objetos*. La estructura de las plantillas planas consiste en una serie de espacios (que constituyen los atributos), cada uno con cero, una, o más de una posibilidad de llenado, que pueden completarse con texto, números, o símbolos de un conjunto definido. Las plantillas orientadas a objetos son estructuras de datos que representan información compleja mediante la organización de la misma en subplantillas u "objetos". A diferencia de las plantillas planas, que simplemente listan atributos, las plantillas orientadas a objetos permiten modelar escenarios más complejos y relaciones entre datos distribuidos en diferentes atributos o subplantillas, facilitando la gestión de información con interdependencias.

En el diseño de plantillas para la extracción de información, se identifican tres puntos que, según lo expuesto en *"Template Filling, Text Mining"*, son necesarios para definir la sintaxis y la semántica de la plantilla, así como para el proceso de llenado de la misma:

- La **definición de la plantilla** establece la estructura y el formato para la extracción de datos, especificando las entidades, atributos y su representación. Se centra en la creación de un esquema claro y consistente que guía el proceso, minimizando la ambigüedad y asegurando la uniformidad en la recolección de información. Un diseño preciso de la plantilla es fundamental para la eficiencia y la calidad del proceso de extracción.

- Las **reglas de interpretación** son instrucciones precisas que mapean la información contenida en los documentos fuente a los campos definidos en la plantilla. Estas reglas, que pueden basarse en patrones, ubicación, contexto o combinaciones de éstos, son cruciales para automatizar y estandarizar la extracción, minimizando la intervención humana y maximizando la precisión del proceso.

- La **documentación de casos** ("case law") consiste en un registro de ejemplos concretos de documentos procesados, incluyendo la información extraída y la resolución de cualquier ambigüedad o conflicto encontrado. Este registro funciona como una base de conocimiento para el perfeccionamiento de las reglas de interpretación, el entrenamiento de sistemas de aprendizaje automático y la evaluación del rendimiento general del proceso de extracción de información.

Existen distintas técnicas para abordar las tareas relacionadas con el *Template Filling*. Si bien los *métodos basados en reglas* ofrecen control y transparencia, su escalabilidad y adaptación a nuevos datos son limitadas. Por otro lado, los *enfoques de aprendizaje automático*, como los modelos de lenguaje, ofrecen mayor flexibilidad y capacidad de generalización, aunque a costa de una menor interpretabilidad. Finalmente, los *métodos híbridos* combinan las fortalezas de ambas aproximaciones, aprovechando las reglas para gestionar casos específicos y el aprendizaje automático para manejar la variabilidad y la generalización, logrando un sistema más robusto y adaptable para diversas tareas.

1.2.2. Information Retrieval (IR)

Durante muchísimos años, la humanidad ha organizado la información para su posterior recuperación y uso: los antiguos romanos y griegos registraban información en rollos de papiro, algunos de los cuales tenían etiquetas adjuntas que contenían un breve resumen para ahorrar tiempo al buscarlos. Los índices o tablas de contenido aparecieron por primera vez en los rollos griegos [30].

El primer representante de repositorios digitales de documentos para búsqueda fue el Sistema SMART de Cornell, desarrollado en la década de 1960 [27]. Los primeros sistemas de RI fueron utilizados principalmente por bibliotecarios especializados, quienes preparaban un conjunto de consultas o solicitudes de búsqueda, las enviaban al sistema todas juntas y luego esperaban a que se procesaran para recibir los resultados. Este enfoque no permitía ajustes inmediatos ni respuestas instantáneas, algo que hoy es común en cualquier buscador moderno.

El nacimiento de la World Wide Web [31] en 1989 y las computadoras modernas marcaron un cambio permanente en los conceptos de almacenamiento, acceso y bús-

queda de colecciones de documentos, haciéndolos accesibles al público en general e indexándolos para una recuperación precisa y de gran cobertura.

Este avance en la gestión de información sentó las bases para lo que hoy conocemos como IR, un campo clave en la búsqueda y acceso eficiente a datos. Según la definición planteada en el libro *"Introduction to Information Retrieval"* [18], *"la* ***recuperación de información*** *consiste en encontrar material (generalmente documentos) de naturaleza no estructurada (usualmente texto) que satisfaga una necesidad de información dentro de grandes colecciones (generalmente almacenadas en computadoras)".*

Sin embargo, la IR puede abarcar otros tipos de datos y problemas informáticos más allá de lo especificado en la definición antes mencionada. Con el tiempo, la recuperación de información ha evolucionado hasta convertirse en la forma dominante de acceder a la información, superando incluso a la búsqueda en bases de datos tradicionales, donde era necesario proporcionar identificadores específicos.

Esta disciplina no solo se limita a datos estructurados, como en las bases de datos relacionales, sino que también abarca datos no estructurados, como los textos, que aunque no siempre tienen una estructura evidente, presentan organización subyacente como títulos, párrafos y notas al pie. Además, también puede abarcar otros tipos de datos y problemas informáticos más allá de lo especificado en la definición central mencionada, como la búsqueda en datos semi-estructurados. Un ejemplo de esto es cuando se busca un documento que contenga ciertas palabras clave en su título y cuerpo. Además de la búsqueda, la recuperación de información incluye el apoyo al usuario en la navegación y filtrado de colecciones de documentos, así como el procesamiento de los resultados obtenidos. Esto puede incluir tareas como la agrupación de documentos basados en su contenido o la clasificación automática según categorías predeterminadas [18].

Para implementar estos procesos IR, se han desarrollado los Sistemas de Recuperación de Información (IRS, por sus siglas en inglés), que son conjuntos de herramientas y procesos diseñados para almacenar, organizar, recuperar y presentar información de manera eficiente en respuesta a las consultas del usuario. Los IRS están orientados a facilitar el acceso a grandes volúmenes de datos, tanto estructurados como no estructurados, como documentos, imágenes, videos y otros tipos de contenido digital. Su función principal es ayudar a los usuarios a encontrar información relevante dentro de un conjunto de datos, basándose en consultas o búsquedas [4].

Las estrategias de recuperación de información asignan una medida de similitud entre una consulta y un conjunto de documentos. Estas estrategias se basan en la idea de que cuán frecuentes aparecen los mismos términos en ambos.

Sin embargo, para lidiar con las ambigüedades inherentes al lenguaje, como la posibilidad de que un mismo concepto sea expresado con diferentes términos, algunas estrategias implementan medidas adicionales. Asimismo, un término puede tener múltiples significados dependiendo de su contexto, lo que requiere técnicas especializadas para garantizar que se interpretan correctamente los conceptos.

Desde una perspectiva teórica, un modelo de recuperación de información puede definirse formalmente [1] como un cuádruple $(D, Q, F, R(g_i, d_j))$, donde:

1. **D** representa el conjunto de vistas lógicas (o representaciones) de los documentos en la colección.

2. **Q** es el conjunto de vistas lógicas que representan las necesidades de información del usuario, conocidas como consultas.

3. **F** constituye el marco conceptual para modelar las representaciones de documentos, consultas y sus relaciones.

4. $\mathbf{R}(g_i, d_j)$ es una función de ranking que asocia un número real a cada consulta $q_i \in Q$ y a cada documento $d_j \in D$. Esta función establece un orden entre los documentos respecto a una consulta específica, facilitando la identificación de los más relevantes.

En este contexto, las estrategias de recuperación operan como algoritmos que procesan una consulta Q y un conjunto de documentos D_1, D_2, ..., D_n, definiendo una función de ranking. Grossman, por ejemplo, utiliza el Coeficiente de Similitud $SC(Q, D_i)$ como medida para determinar la relevancia de cada documento D_i, donde $1 \leq i \leq n$ [13].

Existen diversas estrategias de recuperación, y la elección del modelo adecuado depende de las características del sistema y los requisitos específicos de la consulta. A continuación, se presentan algunos de estos enfoques [2].

El **Modelo Booleano** está basado en la teoría de conjuntos y el álgebra de Boole. Las consultas se formulan mediante expresiones booleanas, utilizando conectores

lógicos como *not*, *and* y *or*, las cuales tienen una semántica precisa y pueden representarse en forma normal disyuntiva. Sin embargo, presenta limitaciones importantes. Al basarse en un criterio binario de relevancia, carece de una escala de gradación que permita medir la relevancia de manera más precisa, lo que puede resultar en la recuperación de muy pocos o demasiados documentos. Además, aunque las expresiones booleanas son formalmente claras, a menudo resulta difícil y poco intuitivo para los usuarios formular consultas complejas que reflejen sus necesidades de información.

A pesar de sus limitaciones, el modelo Booleano sigue siendo relevante en ciertos contextos debido a su simplicidad, especialmente para usuarios nuevos en el campo de la recuperación de información o en sistemas que no requieren un alto nivel de complejidad.

El **Modelo de Espacio Vectorial** (VSM, por sus siglas en inglés) propone una mejora sobre el modelo booleano al permitir coincidencias parciales mediante el uso de pesos no binarios asignados a los términos tanto en las consultas como en los documentos. Estos pesos permiten calcular un grado de similitud entre cada documento almacenado en el sistema y la consulta del usuario. Al ordenar los documentos recuperados en función de este grado de similitud, el modelo logra resultados más precisos y relevantes, ajustándose mejor a las necesidades de información del usuario.

En el VSM, tanto los documentos d_j como las consultas q se representan como vectores t-dimensionales, donde cada dimensión corresponde a un término índice en el sistema. Cada componente del vector de un documento $\boldsymbol{d_j}$ y de una consulta $\boldsymbol{q}$ está ponderada por los valores $w_{i,j}$ y $w_{i,q}$, respectivamente, con $w_{i,j} \geq 0$ y $w_{i,q} \geq 0$. La similitud entre un documento y una consulta se calcula mediante el coseno del ángulo entre sus vectores, utilizando la fórmula:

$$\text{sim}(d_j, q) = \frac{\sum_{i=1}^{t} w_{i,j} \cdot w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \cdot \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}$$

Aquí, $|\boldsymbol{d_j}|$ y $|\boldsymbol{q}|$ representan las normas de los vectores, donde el factor $|\boldsymbol{q}|$ no afecta el ordenamiento de los documentos, ya que es constante para todos ellos, mientras que $|\boldsymbol{d_j}|$ proporciona una normalización en el espacio de documentos. El valor de similitud $\text{sim}(d_j, q)$ oscila entre 0 y 1, indicando el grado de correlación entre el documento y la consulta. Este enfoque no predice directamente si un documento es relevante, sino

que clasifica los documentos en función de su grado de similitud con la consulta. De este modo, un documento puede ser recuperado incluso si solo coincide parcialmente con la consulta. Además, es posible establecer un umbral de similitud para filtrar únicamente los documentos cuya similitud exceda dicho valor. Para determinar estos valores de similitud, es necesario definir cómo se calculan los pesos de los términos índice.

Para calcular estos pesos, se emplea el factor de frecuencia de término (TF), que mide la frecuencia con la que un término k_i aparece en un documento d_j, reflejando cuán representativo es dicho término para el contenido del documento. A esto se le añade la frecuencia inversa de documento (IDF), que ajusta la relevancia de un término en función de cuán común o raro es en la colección completa. La frecuencia inversa de documento es especialmente útil para reducir la importancia de los términos que aparecen con demasiada frecuencia, ya que no contribuyen significativamente a la diferenciación de los documentos. La fórmula para calcular el peso TF-IDF es:

$$\text{TF-IDF}(k_i, d_j) = \frac{f_{i,j}}{\max_i(f_{i,j})} \times \log\left(\frac{N}{n_i}\right),$$

donde $f_{i,j}$ es la frecuencia bruta del término k_i en el documento d_j; $\max_i(f_{i,j})$ es la frecuencia máxima de cualquier término en el documento d_j; N es el número total de documentos en la colección y n_i es el número de documentos en los que aparece el término k_i.

Gracias a la combinación de los factores TF y IDF, este modelo mejora la precisión en la clasificación de documentos. Sin embargo, el VSM presenta una limitación teórica importante: asume que los términos dentro de un documento son independientes entre sí, lo que puede no reflejar la realidad en documentos donde los términos están relacionados o dependen unos de otros. A pesar de esta suposición, el modelo sigue siendo uno de los enfoques más utilizados en sistemas de búsqueda debido a su simplicidad y efectividad.

El **Modelo de Indexación Semántica Latente** (LSI, por sus siglas en inglés) es una variante del VSM que aborda problemas como la sinonimia y la polisemia, que afectan los modelos clásicos basados en términos índice. LSI utiliza una técnica matemática llamada Descomposición en Valores Singulares (SVD), que permite representar los datos de una manera más compacta y significativa. Mediante SVD, el

modelo transforma la matriz de términos y documentos en un espacio de menor dimensión, capturando las relaciones semánticas subyacentes entre términos, en lugar de simplemente emparejar palabras exactas. Este proceso ayuda a eliminar dimensiones que no aportan valor relevante, permitiendo una mejor comprensión y recuperación de la información.

El resultado es un modelo más eficiente que captura solo las características más relevantes del texto, facilitando el análisis y la recuperación de información pertinente.

1.3. Bases de Datos: de los sistemas tradicionales a las soluciones modernas

A lo largo de la historia, la necesidad de almacenar y organizar datos ha sido fundamental para el avance de la civilización, facilitando la transmisión de conocimientos y el desarrollo de la ciencia y la tecnología. Desde sus inicios, los seres humanos han utilizado sistemas de almacenamiento, como bibliotecas y librerías, que resguardaban grandes cantidades de información en libros y documentos.

La transición de los sistemas tradicionales de almacenamiento a soluciones digitales surgió como respuesta al crecimiento exponencial de la información. Con la llegada de la era digital, se presentó la necesidad no solo de almacenar grandes volúmenes de datos, sino de gestionarlos de manera eficiente, escalable y accesible.

El concepto de base de datos tiene sus raíces mucho antes de la llegada de las primeras computadoras electrónicas. Vannevar Bush, con su visión sobre la organización de la información, propuso estructuras para almacenar datos de manera más flexible, sin depender de configuraciones específicas de hardware, lo que influyó en el diseño de sistemas de almacenamiento y procesamiento de datos más avanzados [29].

Esta necesidad de un control más eficiente y escalable sobre los datos dio origen a la idea de separar la gestión de los datos de la lógica de las aplicaciones, lo que resultó en la creación de los Sistemas de Gestión de Bases de Datos (SGBD). Estos sistemas actúan como intermediarios entre las aplicaciones y los datos, proporcionando herramientas esenciales para definir, crear, consultar, actualizar y administrar la información.

En este contexto, Allen Taylor ofrece su propia definición de base de datos:

> *"Una base de datos es una colección autodescriptiva de registros integrados. Por autodescriptiva, me refiero a que contiene una descripción de su propia estructura como parte de los datos que almacena. Cuando digo que los registros en una base de datos son integrados, me refiero a que existen relaciones entre ellos que los vinculan, formando un sistema lógico y cohesivo"* [6].

La propuesta de Allen Taylor se amplió y consolidó sobre la base teórica desarrollada por Edgar Codd. En 1970, Codd introdujo el modelo relacional de bases de datos en su artículo titulado *"A Relational Model of Data for Large Shared Data Banks"* [7], en el que presentó una nueva teoría sobre la organización y gestión de los datos.

Las bases de datos relacionales se basan en un modelo matemático formal que organiza los datos y las relaciones entre ellos a través de tablas, donde cada tabla representa una relación entre los diferentes elementos almacenados. Cada tabla contiene filas (tuplas) y columnas (atributos), donde cada columna representa un atributo específico y cada fila almacena los valores correspondientes a esos atributos para una entidad particular [5]. Este modelo se define a través de un esquema que establece la estructura de las tablas, los atributos y las relaciones entre ellas. En relación con esto, Codd introduce el concepto de consistencia, que se refiere al estado que alcanza una base de datos cuando satisface un conjunto de restricciones, conocidas como restricciones de integridad, que son utilizadas para garantizar la estabilidad y fiabilidad de los datos en la base de datos durante la ejecución de operaciones que los modifican.

Los SGBD relacionales se caracterizan por el procesamiento transaccional de los datos, es decir, por un conjunto de operaciones que modifican el estado de la base de datos. Estos sistemas están construidos sobre los principios ACID, que garantizan la consistencia y fiabilidad en la gestión de las transacciones:

- **Atomicity (A)**: Las transacciones son indivisibles; o se completan en su totalidad o no se realizan en absoluto.
- **Consistency (C)**: Después de cada transacción, la base de datos pasa de un estado consistente a otro.

- **Isolation (I)**: Las transacciones no deben interferir entre sí, es decir, el estado intermedio de una transacción no es visible por el resto de transacciones.
- **Durability (D)**: Una vez completada una transacción, sus cambios son permanentes, incluso ante fallos del sistema.

A pesar de la robustez que ofrece el modelo relacional, su aplicación universal ha comenzado a mostrar limitaciones significativas. Entre los problemas más comunes se encuentran: el alto costo de las lecturas, ya que las consultas que involucran operaciones de unión (JOIN) entre tablas pueden ser costosas en términos de tiempo de ejecución y recursos de cómputo; la sobrecarga de transacciones, que pueden afectar el rendimiento si no se requieren para garantizar la integridad de los datos; la dificultad para escalar horizontalmente, ya que los SGBD relacionales no están diseñados para distribuir datos eficientemente entre varios servidores; y la ineficiencia al representar algunos dominios complejos, como los modelos orientados a objetos o las redes sociales, que no se ajustan bien al modelo relacional [3].

Con el objetivo de superar las limitaciones del modelo relacional, surgieron las bases de datos NoSQL, que adoptan un enfoque diferente a la gestión de datos. En lugar de basarse en las garantías de consistencia estrictas que ofrece ACID, los SGBD NoSQL priorizan la disponibilidad del sistema, fundamentándose en los principios BASE [28]

- **Basically Available (BA)**: Garantiza que el sistema esté disponible para consultas y operaciones de escritura en todo momento, permitiendo la accesibilidad simultánea por parte de los usuarios sin necesidad de esperar a que otros finalicen sus transacciones para actualizar los registros, incluso si no todas las réplicas están al día.
- **Soft state (S)**: Hace referencia a la noción de que los datos pueden tener estados transitorios o temporales que pueden cambiar con el tiempo, incluso sin nuevas entradas. Describe el estado de transición del registro cuando varias aplicaciones lo actualizan en simultáneo. El valor del registro se finaliza solo después de que se hayan completado todas las transacciones.

- **Eventual consistency (E)**: Asegura que, aunque no haya consistencia inmediata, todas las réplicas del sistema alcanzarán eventualmente un estado consistente.

Estos principios se alinean con las restricciones establecidas por el Teorema CAP [15], cuya conjetura fue enunciada por Eric Brewer. Este Teorema establece que un sistema distribuido no puede ofrecer simultáneamente las tres garantías clave de consistencia, disponibilidad y tolerancia a particiones. Los SGBD NoSQL, al priorizar la disponibilidad y la tolerancia a particiones sobre la consistencia, ofrecen una solución robusta para aplicaciones que requieren escalabilidad y rendimiento en la gestión de grandes volúmenes de datos.

Estos sistemas se diferencian principalmente por el modelo de datos empleado para el almacenamiento, lo que les permite adaptarse a diversas necesidades según el caso de uso. En general, los SGBD NoSQL prescinden de un esquema fijo y rígido, característica propia de los sistemas relacionales, y optan por estructuras más dinámicas que facilitan la representación de datos heterogéneos. Esta flexibilidad no solo simplifica la integración de nuevos datos, sino que también permite realizar cambios en la estructura sin interrumpir el funcionamiento del sistema.

Además, la capacidad de escalar horizontalmente es una de las principales ventajas de los SGBD NoSQL. Esto significa que, a medida que aumentan los volúmenes de datos o la carga de trabajo, es posible distribuirla entre múltiples nodos o servidores, lo que garantiza un rendimiento eficiente incluso en entornos con altos niveles de concurrencia. Esta propiedad resulta particularmente beneficiosa en sistemas donde la alta disponibilidad y la tolerancia a fallos son esenciales para garantizar la continuidad del servicio.

Aunque existen diferentes enfoques y paradigmas dentro del ecosistema NoSQL, todos comparten la característica de estar diseñados para resolver problemas específicos que los sistemas relacionales tradicionales encuentran difíciles de abordar.

La elección entre un modelo relacional y uno NoSQL depende de las necesidades específicas del sistema y las características de los datos a manejar. Mientras que los SGBD relacionales son ideales para aplicaciones que requieren integridad, consistencia y transacciones complejas, los sistemas NoSQL resultan más adecuados para mane-

jar grandes volúmenes de datos distribuidos, con alta disponibilidad y escalabilidad, especialmente en entornos con datos semi-estructurados o no estructurados.

Capítulo 2

Concepción y diseño de la solución

En este capítulo se describe la concepción de la solución propuesta para el desarrollo del sistema destinado a la gestión de la información científica sobre las plantas medicinales cubanas, con base de conociemiento inicial en el libro de Tomás Roig. El capítulo está estructurado en tres secciones principales.

En primer lugar, se presenta el contexto en que se desarrolla la solución, explicando las motivaciones y necesidades que llevaron a su concepción. Posteriormente, se exponen los requerimientos que sirven como base de una modelación del problema, detallando cómo se definió y estructuró la problemática a resolver. Finalmente, se describe el diseño de la solución, dividida en dos subproblemas específicos, abordando el enfoque adoptado para dar respuesta a cada uno de ellos. Este análisis establece las bases conceptuales y técnicas necesarias para la posterior implementación y experimentación del sistema.

2.1. Contexto del problema

La iniciativa para desarrollar el presente sistema surge a partir de un diagnóstico conjunto realizado entre la Universidad de La Habana y el Jardín Botánico Nacional de Cuba, en el contexto de un acuerdo de colaboración científica. Este acuerdo tiene como objetivo principal la creación de soluciones que apoyen la preservación, organización y difusión del conocimiento botánico en el país, un campo que reviste

gran importancia tanto para la investigación científica, como para la educación y el conociemiento general de las personas.

Durante este diagnóstico, se identificó que uno de los recursos más valiosos en el ámbito de la botánica cubana, el libro *"Plantas medicinales, aromáticas o venenosas de Cuba"* de Tomás Roig y Mesa, enfrentaba múltiples desafíos relacionados con su accesibilidad y aprovechamiento. Este libro, publicado originalmente en 1945, constituye una obra de referencia fundamental que recopila una vasta cantidad de información científica sobre la flora medicinal de Cuba, incluyendo descripciones botánicas, usos terapéuticos y distribución geográfica de las plantas documentadas. Sin embargo, a pesar de su relevancia, el acceso a esta información sigue siendo limitado debido a varios factores:

- **Formato físico predominantemente tradicional**: Aunque existen versiones digitales del libro, estas no cuentan con un diseño modular que facilite su consulta o análisis de la información. Esto reduce significativamente su usabilidad en contextos modernos donde predominan las herramientas tecnológicas.
- **Pérdida potencial del conocimiento**: El envejecimiento de los ejemplares físicos y la falta de iniciativas de conservación digital de alta calidad ponen en riesgo la preservación de este importante recurso.
- **Falta de integración en sistemas modernos de información**: Los datos contenidos en el libro no están organizados de manera que puedan ser utilizados en aplicaciones automatizadas, análisis de datos o sistemas de consulta avanzada.

A partir de esta realidad, el Jardín Botánico Nacional planteó la necesidad de desarrollar un sistema que no solo permitiera la digitalización de esta información, sino que también la estructurara en un formato accesible y flexible, capaz de responder a las demandas de diferentes tipos de usuarios. Este sistema debía estar alineado con el interés institucional de promover la conservación del patrimonio científico y natural de Cuba, a la vez que facilitara su divulgación a nivel nacional.

La Universidad de La Habana, como institución de referencia en la formación de profesionales en ciencias y tecnología, asumió el reto de apoyar esta iniciativa

mediante el desarrollo de una solución tecnológica que integre técnicas de recuperación de información y bases de datos científicas. Este proyecto, en particular, representa un esfuerzo no solo por preservar los recursos botánicos, sino también por sentar las bases para la creación de sistemas similares que puedan aplicarse a otros ámbitos del conocimiento.

2.2. Análisis de requerimientos

Luego de un análisis exhaustivo de los objetivos del sistema y las necesidades identificadas, se han definido los siguientes requerimientos funcionales. Estos buscan garantizar una fiel representación de la información, así como lograr una interacción fluida y efectiva, satisfaciendo las expectativas de los distintos tipos de usuarios a los que está destinado el producto final.

- Presentar de forma estructurada y comprensible toda la información contenida en las monografías del libro de Tomás Roig. Esto incluye las diferentes secciones, como nombres científicos, hábitat, propiedades medicinales y composición química. La representación visual debe facilitar el acceso y la interpretación de los datos, con un diseño que priorice la claridad.
- Proveer un mecanismo avanzado de búsqueda, que permita consultar la información de las plantas almacenadas en el sistema, no solo mediante el nombre de las mismas, sino mediante el contexto que ofrece su monografía.
- Incluir un módulo administrativo que permita a un usuario administrador gestionar la información almacenada. Esto incluye la creación de nuevas monografías de plantas, la edición de información existente para corregir o actualizar datos y la eliminación de registros que ya no sean relevantes o que presenten inconsistencias.
- Ofrecer la visualización de otras secciones relevantes del libro, que enriquezcan el acceso a la información.

Para comprender mejor las interacciones de los usuarios y las funcionalidades del sistema, se ha elaborado un diagrama de casos de uso. Este diagrama representa de

manera gráfica las principales acciones que los diferentes tipos de usuarios pueden realizar dentro del sistema. Su objetivo principal es proporcionar una visión clara y estructurada de los requisitos funcionales, destacando los roles de los usuarios y sus respectivos casos de uso. Además, facilita la identificación de los límites del sistema, asegurando que las interacciones previstas cubran todas las necesidades y expectativas planteadas durante la modelación del problema.

En la figura 2.1, se presenta el diagrama de casos de uso correspondiente a la modelación propuesta del problema.

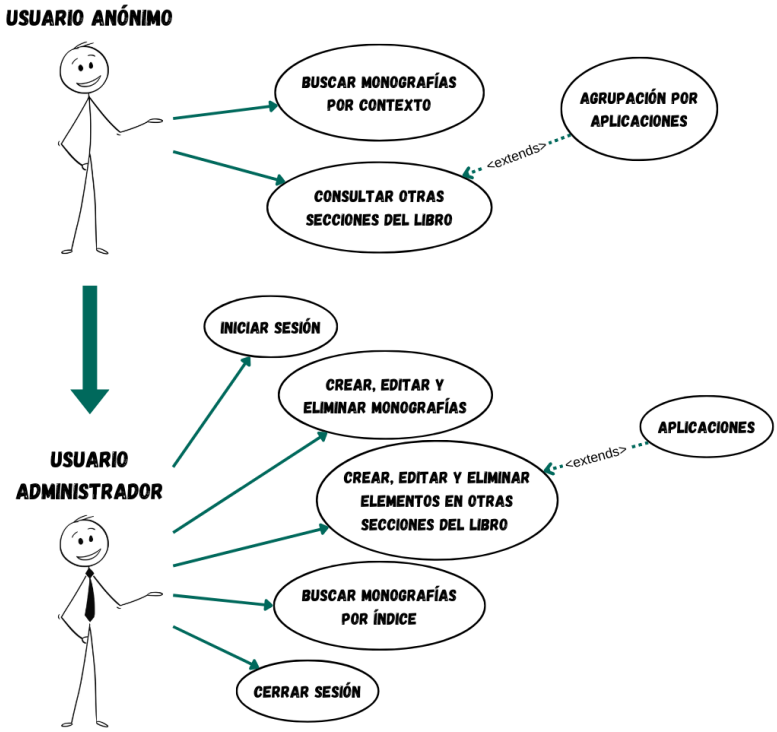

Figura 2.1: Diagrama de casos de uso

2.3. Diseño de la solución

Es posible identificar dos subproblemas principales dentro del contexto del problema anteriormente descrito mediante requerimientos. Estos subproblemas están in-

terrelacionados y son fundamentales para garantizar que el sistema cumpla con los objetivos establecidos:

- **Problema de la extracción de la información**: Este subproblema se refiere al proceso de extraer, estructurar y almacenar de manera eficiente la información contenida en el libro de Tomás Roig, de forma que estos datos puedan ser consumidos por un software computacional.
- **Problema del sistema de gestión y visualización de la información**: Una vez extraída y estructurada la información, surge el desafío de diseñar e implementar un sistema que permita gestionar y visualizar eficientemente los datos.

Esta sección tiene como objetivo presentar las estrategias y propuestas de diseño abstracto desarrolladas para abordar los subproblemas identificados anteriormente. Estos subproblemas, requieren soluciones específicas que garanticen tanto la fidelidad de los datos extraídos como su presentación efectiva a los usuarios finales.

En esta sección, se describirán los enfoques conceptuales diseñados para resolver cada uno de los subproblemas, teniendo en cuenta los requerimientos funcionales previamente establecidos. Se analizarán las características principales de cada solución, incluyendo sus componentes clave y cómo estos se integran para formar un sistema coherente y eficiente. Este análisis establecerá las bases para la implementación detallada del sistema.

2.3.1. Extracción de la información

Una solución al problema planteado debe partir de un análisis exhaustivo del corpus sobre el cual se realizará la extracción de información. En este sentido, tras examinar detalladamente el libro *"Plantas medicinales, aromáticas o venenosas de Cuba"*, se identificaron una serie de observaciones preliminares. Estas observaciones constituyen la base para tomar decisiones informadas respecto a las estrategias de extracción de información que sean adecuadas y acordes con el estado del arte.

Las observaciones identificadas son las siguientes:

1. La obra está dividida en dos tomos, por lo que es deseable que la solución adoptada sea lo más general posible, permitiendo su aplicación efectiva en ambos tomos.

2. Ambos tomos se encuentran en el formato: *Portable Document Format* (PDF)

3. Aunque pertenecen a la misma editorial, existen diferencias significativas en cuanto a la maquetación y el diseño digital entre ambos tomos.

4. Las monografías constituyen la sección fundamental del libro.

5. El tomo 1 contiene las monografías de las plantas cuyos nombres inician con las letras de la 'A' a la 'K', mientras que el tomo 2 abarca aquellas cuyas iniciales están entre la 'L' y la 'Z'.

6. En las monografías se pueden identificar secciones principales que contienen cierta información sobre las plantas. Estas secciones mantienen un orden fijo, aunque no siempre están presentes todas en cada monografía. Las secciones identificadas son:

    - Nombre con que se conoce la plantas.
    - Nombre científico.
    - Sinónimos.
    - Otros nombres vulgares asociados.
    - Hábitat y distribución geográfica.
    - Descripción botánica.
    - Composición química.
    - Partes empleadas.
    - Propiedades medicinales.
    - Aplicaciones.
    - Cultivo.
    - Referencias bibliográficas.

7. Algunas monografías incluyen imágenes de baja calidad de las plantas, acompañadas de un pie de foto que identifica el nombre de la especie.

8. El formato de presentación del texto no es uniforme, lo que responde a decisiones editoriales y de diseño. Por ejemplo, las monografías están dispuestas en una sola columna para facilitar la lectura continua, mientras que otras secciones como la dedicada a la agrupación de plantas según sus aplicaciones están organizadas en tres columnas para optimizar el uso del espacio al listar múltiples nombres.

Estas características del corpus son consideradas para garantizar que la solución propuesta sea capaz de abordar los retos específicos que plantea la extracción de información en un contexto tan heterogéneo.

Es factible realizar la extracción del texto contenido en los documentos en formato *PDF* mediante el uso de lenguajes de programación modernos, apoyándose en bibliotecas especializadas para la manipulación y el procesamiento de este tipo de archivos. No obstante, debe considerarse que el texto presenta características de maquetación no uniformes a lo largo de la obra, lo que podría requerir un manejo cuidadoso de las estructuras y formatos para asegurar una extracción precisa y completa.

En la sección 1.2 se describe el problema general de la IE, identificándola como una de las tareas más comunes y relevantes en el ámbito del Procesamiento del NLP. La IE se enfoca en identificar, estructurar y representar conocimiento relevante a partir de textos no estructurados o semiestructurados.

Dadas las características del corpus objeto de estudio y su estructura textual, la técnica seleccionada para llevar a cabo el proceso de extracción de información es la denominada *template filling* abordada en la sección 1.2.1.1. Esta técnica permite extraer información específica mediante la identificación de patrones predefinidos y su mapeo en plantillas estructuradas. La elección de esta metodología responde a varios factores:

1. La necesidad de obtener los datos en un formato estructurado que facilite su posterior uso en sistemas computacionales.

2. La importancia de preservar las palabras y expresiones originales del autor para garantizar una representación precisa del contenido de la obra.

3. La adecuación de esta técnica para procesar textos con una organización semiuniforme, como las monografías presentes en la obra, permitiendo capturar información clave como nombres científicos, descripciones botánicas, propiedades y aplicaciones.

Para mayor conveniencia, se optará por realizar la extracción de información del libro de manera segmentada, abordando cada sección de forma independiente. Este enfoque permite aplicar el proceso de *template filling* a cada sección por separado, lo cual simplifica significativamente los algoritmos necesarios para la extracción.

2.3.1.1. Template Filling en monografías

Para la extracción de información de las monografías, se optará por un enfoque basado en reglas, considerando las características semiestructuradas de los datos presentes en esta sección del libro. El diseño de la plantilla requerirá abordar tres aspectos fundamentales: la definición de la estructura de la plantilla, la especificación de las reglas de interpretación y la documentación de casos. Sin embargo, este último punto no será desarrollado, dado que su utilidad sería limitada en ausencia de un enfoque basado en aprendizaje automático.

En una etapa inicial, es posible extraer la información correspondiente a cada monografía de manera básica. Esto implica definir una plantilla inicial que incluya los nombres de todas las plantas mencionadas en el libro, asociando a cada nombre el texto plano que representa la información respectiva. La regla de interpretación empleada en este paso se encargará de identificar el inicio de cada monografía, utilizando como criterio el nombre de la planta, que se encuentra destacado con un tamaño de fuente significativo en el texto.

A partir de esta extracción inicial, se procederá a estructurar la información de cada monografía basándose en su contenido. En la Figura 2.2, se presenta la definición de la plantilla utilizada para este propósito, en la que se incluye el nombre de cada atributo, junto con el tipo de dato del mismo.

Cada atributo de la plantilla corresponde a una sección identificable dentro del contenido de una monografía. Por ejemplo, `Sc` representa el nombre científico, `Sy` los

PLANTILLA DE MONOGRAFÍA

ATRIBUTO	TIPO
Sc	subplantilla
Sy	string[]
Vul	string[]
Hab	string
Des	string
Cmp	string
Use	string
Pro	string
App	string
Cul	string
Bib	string[]

PLANTILLA DE NOMBRE CIENTÍFICO

ATRIBUTO	TIPO
genus	string
species	string
authors	string
var	string
subsp	string
f	string
family	string
subfamily	string

Figura 2.2: Plantillas de monografía y nombre científico

sinónimos, `Vul` los nombres vulgares asociados, `Hab` el hábitat y distribución geográfica, `Des` la descripción botánica, `Cmp` la composición química, `Use` las partes empleadas, `Pro` las propiedades medicinales, `App` las aplicaciones, `Cul` el cultivo y `Bib` las referencias bibliográficas.

Como se observa en la Figura 2.2, este diseño emplea una plantilla híbrida que combina características de plantillas planas y plantillas orientadas a objetos, permitiendo que los atributos puedan almacenar tanto datos primitivos como subplantillas, lo que facilita una representación más estructurada y flexible de la información.

Definamos entonces las reglas de interpretación para la plantilla de las monografías:

1. Las secciones dentro de una monografía siempre aparecen en el mismo orden, y no necesariamente aparecen todas en una monografía.

2. El nombre científico siempre aparece en la primera línea inmediatamente después del título de la monografía.

3. Los sinónimos están precedidos por la cadena de texto `"SINÓNIMOS:"` y se encuentran separados entre sí por comas (`,`).

4. Los otros nombres vulgares están precedidos por la cadena de texto `"OTROS NOMBRES VULGARES:"`. Los nombres correspondientes a un mismo territorio están separados por comas (`,`), mientras que los nombres entre territorios están separados por punto y coma (`;`).

5. El texto correspondiente al hábitat y distribución está precedido por la cadena de texto `"HÁBITAT Y DISTRIBUCIÓN:"`.

6. El texto correspondiente a la descripción botánica está precedido por la cadena de texto `"DESCRIPCIÓN BOTÁNICA:"`.

7. El texto correspondiente a la composición química está precedido por la cadena de texto `"COMPOSICIÓN:"`.

8. El texto correspondiente a las partes empleadas está precedido por la cadena de texto `"PARTES EMPLEADAS:"`.

9. El texto correspondiente a las propiedades de la planta está precedido por la cadena de texto `"PROPIEDADES:"`.

10. El texto correspondiente a las aplicaciones está precedido por la cadena de texto `"APLICACIONES:"`.

11. El texto correspondiente al cultivo de la planta está precedido por la cadena de texto `"CULTIVO:"`.

12. El texto correspondiente a las referencias bibliográficas está precedido por la cadena de texto `"BIBLIOGRAFÍA"`, y cada bibliografía termina en el año correspondiente a la misma.

En cuanto a los nombres científicos, se pueden definir las siguientes reglas en función de la plantilla:

1. Las partes que componen un nombre científico siempre siguen un orden específico, aunque no necesariamente todas deben estar presentes.

2. Las primeras dos palabras corresponden al género y la especie, respectivamente.
3. La autoridad de la planta siempre aparece inmediatamente después del género y la especie.
4. La variedad siempre está precedida por la cadena de texto `"var."`.
5. La subespecie siempre está precedida por la cadena de texto `"subsp."`.
6. La forma siempre está precedida por la cadena de texto `"f."`.
7. La familia siempre está precedida por la cadena de texto `"Fam."`.
8. La subfamilia siempre está precedida por la cadena de texto `"Subfam."`.

Con la definición de estas plantillas y reglas de interpretación, se logra un diseño adecuado para la extracción de la información de las monografías, asegurando que los datos sean identificados y estructurados correctamente de acuerdo con su formato original.

2.3.1.2. Template Filling en agrupación de plantas por aplicaciones

La sección del libro que agrupa las plantas según sus aplicaciones se presenta con un diseño de tres columnas, optimizado para maximizar el uso del espacio disponible. En consecuencia, será necesario realizar una lectura ordenada que permita obtener el texto plano de toda la sección, a fin de aplicar la técnica de *template filling*.

De manera análoga a lo realizado en la sección de las monografías, se empleará un enfoque basado en reglas para abordar esta sección del libro. Inicialmente, es posible identificar todas las aplicaciones mencionadas, bajo la regla de que los nombres de las aplicaciones aparecen completamente en mayúsculas y finalizan con el caracter de dos puntos (`:`), mientras que los nombres de las plantas contienen caracteres en minúsculas. Esto nos lleva a tener una plantilla que tiene como atributos los nombres de todas las aplicaciones.

Al analizar la sección, se observa que algunas aplicaciones incluyen referencias a otras ya mencionadas, debido a que estas representan sinónimos o significados equivalentes. En estos casos, resulta útil que cada aplicación cuente con una lista de

sinónimos que agrupe dichas referencias. Por lo tanto, se definirá una subplantilla como tipo de cada atributo en la plantilla anterior, adoptando la estructura ilustrada en la Figura 2.3.

PLANTILLA DE LAS APLICACIONES

ATRIBUTO	TIPO
ABORTIVOS	subplantilla
ABSORVENTES	subplantilla
AFRODISÍACOS	subplantilla
...	subplantilla
VULNERARIOS	subplantilla

PLANTILLA DE UNA APLICACIÓN

ATRIBUTO	TIPO
plants	string[]
sys	string[]

Figura 2.3: Plantillas de aplicaciones

Es posible observar que para cada aplicación tendremos una subplantilla que almacena dos listas, una con los nombres de todas las plantas que tienen esa aplicación, y otra con los sinónimos identificados en el texto.

En base a lo anterior definimos las reglas de llenado:

1. Las plantas asociadas a cada aplicación comienzan con el nombre de esta última, seguido por el caracter de dos puntos (`:`).

2. En general, los nombres de las plantas se identifican por los saltos de línea, salvo en algunos casos excepcionales.

3. Los casos excepcionales incluyen palabras que no caben en una misma línea, las cuales se indican mediante un guion (`-`), señalando que el nombre continúa en la siguiente línea.

4. Algunas aplicaciones presentan, justo a continuación de su nombre, el nombre de otra aplicación entre paréntesis, lo que indica que heredan las plantas asociadas a esta última.

5. Las referencias a otras aplicaciones se identifican por la cadena de caracteres `"Véase"`, seguida del nombre de la aplicación correspondiente.

2.3.2. Sistema de gestión y visualización

El diseño de un sistema web eficiente y escalable requiere la separación clara entre el frontend y el backend, siguiendo los principios de modularidad y responsabilidad única. El frontend se ocupa de la presentación y experiencia del usuario, mientras que el backend se encarga de procesar la lógica del negocio, gestionar la persistencia de datos y exponer interfaces (APIs) para interactuar con el sistema. Este enfoque asegura que cada componente pueda desarrollarse, mantenerse y escalarse de manera independiente.

En el caso del backend, se empleará una arquitectura por capas que permita una separación clara de responsabilidades. Esta arquitectura incluirá las siguientes capas principales:

1. **Capa de datos**: Responsable de modelar las entidades del dominio.

2. **Capa de acceso a datos**: Proveerá métodos especializados para interactuar con la base de datos, aislando las operaciones sobre la misma de la lógica de negocio.

3. **Capa de servicios**: Implementará la lógica del negocio y las reglas del sistema, ofreciendo funcionalidades como servicios que serán consumidos posteriormente.

Estas capas estarán integradas en un proyecto principal de backend, cuya función será consumir los servicios para exponer una serie de APIs que sirvan como puente entre los frontends y la lógica del sistema. Este diseño por capas garantiza mantenibilidad, escalabilidad y reutilización del código.

Debido a los requerimientos del sistema, se necesitarán dos módulos principales: un módulo de administración y un módulo orientado al cliente final de internet. Cada

uno de estos módulos contará con un proyecto independiente de frontend para cubrir las necesidades específicas de cada tipo de usuario. Este enfoque asegura que ambos módulos sean autónomos en su desarrollo y despliegue, al tiempo que se preserva la consistencia general del sistema.

Para optimizar el uso de recursos y evitar redundancias, las capas de datos, acceso a datos y servicios del backend serán compartidas entre los módulos. Esto significa que, aunque se desarrollen dos proyectos de web API (uno para el módulo de administración y otro para el cliente final), ambos consumirán las mismas capas internas de lógica y acceso a datos. De esta manera, se centraliza la lógica común en un solo lugar, promoviendo la reutilización del código, simplificando el mantenimiento y facilitando la escalabilidad del sistema ante posibles cambios o nuevas funcionalidades en el futuro. En la Figura 2.4 se ilustra la arquitectura explicada del sistema.

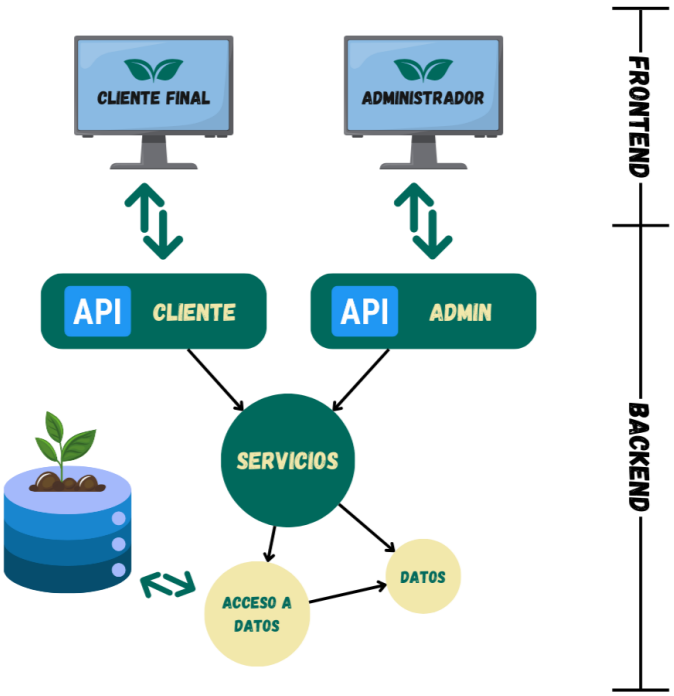

Figura 2.4: Arquitectura del sistema

2.3.2.1. El modelo de datos

El modelo de datos proporciona la estructura necesaria para almacenar, gestionar y acceder de manera eficiente a la información. Basado en las entidades fundamentales del dominio, el modelo representa los elementos clave que se gestionarán a lo largo de su ciclo de vida, estableciendo relaciones entre ellos. Es esencial que el diseño de estas relaciones esté bien estructurado, alineado con los requerimientos funcionales y las características del sistema, para garantizar la alta integridad y disponibilidad de los datos, así como para facilitar las consultas y optimizar el rendimiento del sistema.

Los principales componentes del modelo de datos son:

1. **Entidades del dominio**: Son los objetos principales con los que interactuarán los usuarios y el sistema. En nuestro caso, estas entidades incluyen elementos como plantas, términos, aplicaciones y usuarios. Cada entidad está representada por una tabla en la base de datos, y sus atributos corresponden a las columnas de dichas tablas.

2. **Relaciones entre entidades**: Las entidades no operan de manera aislada, sino que están interconectadas entre sí. Las relaciones entre las entidades pueden ser de uno a uno, uno a muchos o muchos a muchos, y deben estar modeladas adecuadamente en el esquema de la base de datos. Estas relaciones permiten que los datos sean accesibles y actualizados de manera coherente en todo el sistema.

3. **Normalización y optimización**: Para garantizar la eficiencia en el acceso y la integridad de los datos, se llevará a cabo un proceso de normalización. La normalización busca reducir la redundancia de los datos y mejorar la consistencia de las relaciones. Esto significa que cada tipo de información debe estar en su lugar adecuado, sin duplicar datos innecesarios. Además, se implementarán índices y claves foráneas para optimizar las consultas y mantener la integridad referencial.

4. **Persistencia de datos**: El modelo de datos se implementará utilizando herramientas que permitan almacenar, consultar y actualizar la información de manera eficiente.

El modelo de datos será la base sobre la cual se construirán los servicios y las APIs del sistema. Cada operación o servicio del backend interactuará con la base de datos a través de las capas de acceso a datos y servicios, garantizando que las transacciones sean consistentes y eficaces.

Aunque el carácter variable de las secciones presentes en las monografías podría sugerir el uso de una base de datos NoSQL, se optará por utilizar una base de datos relacional (SQL). Esta decisión de emplear SQL está motivada por el enfoque adoptado en el diseño, donde se utiliza un modelo basado en plantillas estructuradas para representar la información de las monografías.

El modelo de relaciones se basará en un enfoque de muchos a muchos entre las entidades Planta y Término (a través de PlantTerm), así como entre Planta y Aplicación (a través de PlantApp). Estas relaciones intermedias permitirán flexibilidad en el sistema, permitiendo que cada planta esté asociada a múltiples términos y aplicaciones y viceversa, y de esta forma mantener la integridad referencial y evitar problemas de redundancia.

En el caso de la entidad Usuario, no se encontrará interrelacionada directamente con las demás entidades, siendo completamente autónoma, lo que simplifica su manejo y permite que los usuarios interactúen con el sistema sin necesidad de relaciones directas con las demás entidades del dominio.

Para facilitar la comprensión de la estructura de la base de datos, el diagrama de la Figura 2.5 ilustra el modelo Entidad-Relación-Extendido (MERX) del sistema, que refleja las entidades principales y sus relaciones clave.

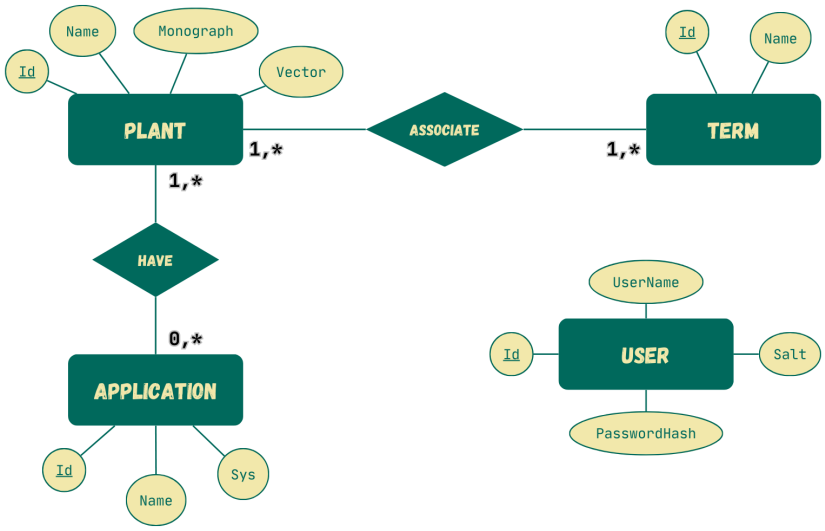

Figura 2.5: MERX

Capítulo 3

Implementación y experimentación

En el presente capítulo se presentan los elementos prácticos que han sido empleados para materializar la solución conceptualizada en el diseño previo. El objetivo principal es detallar cómo se implementaron las diversas tecnologías, metodologías y herramientas seleccionadas para construir un sistema funcional que cumpla con los requisitos establecidos y los objetivos trazados.

Se expondrá en primer lugar la identidad del sitio web, destacando las decisiones relacionadas con su diseño visual y los elementos que refuerzan su alineación con los valores de marca del Jardín Botánico Nacional de Cuba. Posteriormente, se describirán de manera estructurada los procesos técnicos y las decisiones tomadas durante la implementación, haciendo énfasis en aspectos clave como la selección de tecnologías, la organización del código y las estrategias empleadas para integrar y probar los componentes del sistema.

Además, se discutirán los retos encontrados durante esta etapa y las soluciones adoptadas para superarlos, asegurando que el producto final sea técnicamente robusto y alineado con las necesidades del proyecto.

Se explicarán por separado las soluciones propuestas para las dos problemáticas abordadas en el capítulo anterior. Este enfoque se adopta porque cada problemática puede considerarse un problema independiente, lo que facilita una comprensión más clara y detallada de la solución final.

Adicionalmente, se incluyen las primeras pruebas experimentales realizadas con la solución implementada, con el fin de evaluar su desempeño y validar que los re-

sultados obtenidos cumplen con las expectativas definidas en las fases iniciales. Estos experimentos no solo verifican el cumplimiento funcional, sino que también permiten identificar posibles áreas de mejora o ajuste, garantizando que el sistema sea escalable y adaptable a futuros requerimientos.

3.1. Identidad del sitio

El sistema desarrollado como resultado de este trabajo ha sido nombrado **BotaniQ**, un nombre que refleja de manera directa su propósito y esencia. La elección de este nombre surge de la combinación de dos elementos clave: la palabra *"botánica"*, que alude al estudio de las plantas, y la letra *"Q"*, que hace referencia a *"query"* (significa consulta en inglés), resaltando su función principal como una herramienta para la consulta y gestión de información sobre plantas medicinales. Este nombre busca transmitir simplicidad, profesionalismo y un enfoque claro en la temática del proyecto, a la vez que facilita su identificación y asociación con su objetivo principal. En la Figura 3.1 se muestra el logotipo de BotaniQ.

Figura 3.1: Logotipo de BotaniQ

Las interfaces de BotaniQ están diseñadas para reflejar y reforzar un valor de marca que pueda asociarse directamente con el Jardín Botánico Nacional de Cuba. Para lograr este propósito, se ha adoptado una paleta cromática basada en los colores primario y secundario presentes en el logotipo de dicha institución, asegurando así una identidad visual coherente y representativa. La paleta de colores se muestra en la Figura 3.2.

Para garantizar que el diseño del sitio web BotaniQ transmita una identidad visual acorde a su propósito, se seleccionaron fuentes tipográficas que equilibran profesiona-

Figura 3.2: Paleta de colores

lismo, claridad y frescura, alineadas con la temática botánica y científica del proyecto. Estos estilos son accesibles libre de costo desde el sitio: Google Fonts. Una muestra de estos estilos se puede apreciar en la Figura 3.3.

- Estilo primario: `Montserrat Alternates`
- Estilo secundario: `Quicksand`
- Estilo complementario: `Sniglet`

Figura 3.3: Fuentes tipográficas

3.2. Solución al problema de Extracción de información

Para implementar esta solución, se seleccionó **Python**[1] como lenguaje de programación debido a su versatilidad, facilidad de uso y la sólida comunidad que lo respalda, ofreciendo una amplia variedad de bibliotecas para diversas tareas. Python es un lenguaje de programación interpretado, de alto nivel y multiparadigma, que permite trabajar con estilos como la programación orientada a objetos, funcional y procedimental. Su diseño enfatiza la legibilidad del código, lo que facilita el desarrollo y mantenimiento de proyectos.

En este contexto, se eligieron las bibliotecas `PyMuPDF`[2] y `pdfplumber`[3] para abordar la lectura de los documentos en formato *PDF*. La biblioteca `PyMuPDF` fue escogida principalmente por su extensa documentación y su eficacia en la extracción de texto de manera uniforme, lo que resulta fundamental para garantizar una base inicial consistente de los datos extraídos. Por otro lado, `pdfplumber` se seleccionó por las ventajas que ofrece en términos de manejo del diseño del documento, particularmente su capacidad para identificar y encuadrar bloques de texto.

La información extraída será almacenada en un archivo en formato *JSON*. Este formato es ampliamente utilizado en la actualidad debido a su capacidad para representar datos de manera estructurada y su gran adaptabilidad en los sistemas computacionales modernos. *JSON* es un formato ligero y de fácil lectura tanto para humanos como para máquinas, lo que lo convierte en una opción ideal para la interoperabilidad entre diferentes sistemas y plataformas, especialmente en aplicaciones web y servicios API.

Tal como se expuso en el capítulo anterior, se adoptará un enfoque basado en *template filling* para estructurar la información extraída. Las plantillas se implementarán como diccionarios de Python, una estructura de datos que permite almacenar información en pares clave-valor de forma eficiente. Los diccionarios son fundamentales para garantizar una representación coherente y ordenada de los datos, facilitando su posterior transformación al formato *JSON*.

[1] Visitar en `https://www.python.org`
[2] Visitar en `https://pypi.org/project/PyMuPDF/`
[3] Visitar en `https://pypi.org/project/pdfplumber/`

Para la implementación del flujo de llenado de plantillas, se adoptó un estilo de programación imperativa. El llenado de las plantillas se realizó de manera jerárquica, progresando desde los niveles más generales hacia los más específicos. En otras palabras, primero se completaron los atributos simples de la plantilla principal, y posteriormente se procedió al llenado de los atributos que corresponden a subplantillas.

Este enfoque permite encapsular las reglas de llenado de las subplantillas en algoritmos independientes, lo que no solo mejora la modularidad del código, sino que también facilita la comprensión y el mantenimiento del flujo de trabajo. Al trabajar con subplantillas de manera autónoma, se asegura que cada componente de la plantilla sea manejado de forma eficiente y aislada, reduciendo la complejidad del sistema general y permitiendo futuros ajustes o ampliaciones de manera más sencilla.

Durante el desarrollo del algoritmo para la extracción de las monografías, surgieron ciertos problemas que requirieron ser resueltos sobre la marcha. Estas dificultades se debieron, en algunos casos, a excepciones en las reglas de llenado previamente definidas, ya sea por errores tipográficos en el texto original o por inconsistencias durante el proceso de extracción del contenido del libro. Entre los problemas identificados se encuentran los siguientes:

- Secciones en las que no se detectó la palabra clave que determina su inicio fueron insertadas erróneamente como una continuación de la sección previamente identificada.
- Duplicación de nombres de monografías idénticos, lo que generaba conflictos al intentar utilizarlos como claves en los atributos de la plantilla.
- Inclusión de pies de página de las imágenes dentro del texto extraído.
- Inclusión de números de página entre el texto.
- Fragmentación de palabras en el texto debido al uso de guiones (`-`) cuando estas no cabían en la línea del texto original, lo que afectaba la integridad del texto plano extraído.

La solución a estos problemas no fue particularmente compleja de identificar. Algunos casos, debido a su naturaleza limitada y a la falta de un patrón recurrente en

el texto, fueron resueltos de forma directa y específica mediante soluciones personalizadas adaptadas a cada caso puntual.

Al finalizar el algoritmo de extracción, cada atributo que almacena una cadena de texto queda representado en un formato plano. Esto significa que no se preservan las separaciones de párrafos, siendo el contenido una simple secuencia de oraciones concatenadas. Además, en ocasiones donde el texto original contiene listas de elementos, estas se presentan como elementos continuos separados únicamente por espacios. Este tipo de formato puede dificultar la lectura y la interpretación del contenido, lo que hace necesario estructurarlo de manera más clara para mejorar su comprensión. Incluso una medida sencilla, como la inclusión de saltos de línea (`\n`), puede contribuir significativamente a mejorar la legibilidad.

Para abordar este problema, se aprovecha el poder de los modelos de lenguaje para interpretar y generar texto de manera coherente. En este caso, se utilizó el modelo `gemini-1.5-flash` de **Gemini**[4], desarrollado por Google, cuya elección se fundamenta en la disponibilidad de una API gratuita y su integración sencilla con lenguajes como Python para tareas automatizadas. Este modelo ha demostrado excelentes resultados en procesos de procesamiento y generación de texto. Para garantizar que el texto resultante cumpla con los requisitos esperados, se aplicaron técnicas de ingeniería de prompts diseñadas para guiar al modelo hacia la generación de resultados precisos y adecuados.

El resultado final es una plantilla completa, organizada y bien estructurada, con datos listos para ser leídos e interpretados de manera eficiente. Esta transformación mejora la presentación visual del contenido y optimiza su utilidad en contextos prácticos.

A continuación, se procedió a extraer la información correspondiente a la agrupación de plantas según sus aplicaciones. Similar al caso anterior, se presentaron algunos inconvenientes debido a inconsistencias en las reglas definidas durante el diseño de la solución o a limitaciones inherentes a la biblioteca utilizada para la extracción de texto. Entre los problemas identificados se encuentran:

- En ciertos casos puntuales, nombres de plantas que continúan en la siguiente línea no fueron detectados correctamente. Esto ocurrió porque, al no estar sepa-

[4]Visitar a través de Google AI Studio en `https://aistudio.google.com/`

rados por un guion (-), no se considera un corte de palabra, y la palabra en la línea siguiente comienza con mayúscula, lo que impide la asociación automática.

- En un caso específico, un nombre de aplicación no fue identificado en su posición correspondiente, lo que resultó en que su contenido fuera erróneamente incluido dentro de la aplicación anterior.

Dado que estos problemas son casos excepcionales y no recurrentes, se abordaron mediante correcciones directas en el código implementado.

El resultado final es una plantilla estructurada y completamente llena, representada en formato *JSON*, que está lista para ser utilizada en otros entornos computacionales.

3.3. Solución al problema del Sistema de gestión y visualización: BotaniQ

En el desarrollo de BotaniQ, se emplearon tecnologías modernas y ampliamente utilizadas en la industria para garantizar un sistema robusto, eficiente y escalable. Estas herramientas fueron seleccionadas cuidadosamente para abordar los requerimientos específicos del proyecto, permitiendo una implementación organizada y una experiencia de usuario óptima.

Para el frontend, se utilizó **React** [5] en combinación con **TypeScript**[6]. React es una biblioteca de JavaScript enfocada en la creación de interfaces de usuario dinámicas y reactivas, estructuradas en componentes modulares que facilitan el desarrollo y la reutilización de elementos visuales. TypeScript, al ser un superconjunto tipado de JavaScript, aporta seguridad al proceso de desarrollo, permitiendo detectar errores antes de la ejecución y promoviendo un código más estructurado, lo cual es esencial en un proyecto de esta magnitud. Esta combinación mejora la experiencia del usuario final con una interfaz intuitiva, y facilita el mantenimiento y la escalabilidad del sistema.

En el backend, se optó por **ASP.NET Core**[7] como marco de desarrollo, una

[5] Visitar en `https://react.dev`
[6] Visitar en `https://www.typescriptlang.org`
[7] Visitar en `https://dotnet.microsoft.com/en-us/apps/aspnet`

plataforma de código abierto diseñada para construir aplicaciones modernas de alto rendimiento. Este framework permite desarrollar servicios web estructurados que manejan eficientemente la lógica del negocio y el acceso a datos. La implementación del backend se organizó en capas previamente definidas: la capa de datos, la capa de acceso a datos y la capa de servicios, cada una desarrollada como una biblioteca dinámica (DLL) utilizando **C#**[8].

La información se gestiona mediante una base de datos **PostgreSQL**[9], un sistema de gestión de bases de datos relacional reconocido por su rendimiento y capacidad para manejar datos estructurados. A pesar de su naturaleza relacional, PostgreSQL ofrece características similares a las bases de datos NoSQL, como la capacidad de almacenar datos en formato JSON. Esta flexibilidad nos permite almacenar las monografías de plantas de manera eficiente, asegurando que, a pesar de que algunas secciones puedan estar vacías, todas las monografías sigan el mismo esquema gracias a la técnica de template filling.El backend es responsable de procesar las solicitudes, interactuar con PostgreSQL a través de la capa de acceso a datos y exponer los datos mediante servicios web que pueden ser consumidos por el frontend.

Estas tecnologías trabajan de forma integrada para garantizar un flujo de datos coherente y confiable entre el cliente y el servidor, asegurando que los usuarios de BotaniQ puedan acceder a la información de manera rápida y eficiente. Estas herramientas modernas y probadas en entornos de producción aseguran que el sistema sea robusto, mantenible y esté preparado para futuros requerimientos.

Esta estructura tecnológica se ilustra en la Figura 3.4, proporcionando una visión clara de cómo se integran los diferentes componentes del sistema.

En ambos proyectos de frontend de BotaniQ, el desarrollo se realizó siguiendo el diseño de identidad del sitio, que se basa en ofrecer una experiencia visual coherente y profesional que refleje la marca asociada al Jardín Botánico Nacional de Cuba. A lo largo del proceso de implementación, se prestó especial atención a la adaptabilidad de la interfaz, asegurando que el sitio fuera completamente responsivo y brindara una experiencia de usuario óptima en diversos dispositivos, desde computadoras de escritorio hasta teléfonos móviles y tabletas.

[8]Visitar en `https://dotnet.microsoft.com/en-us/languages/csharp`
[9]Visitar en `https://www.postgresql.org`

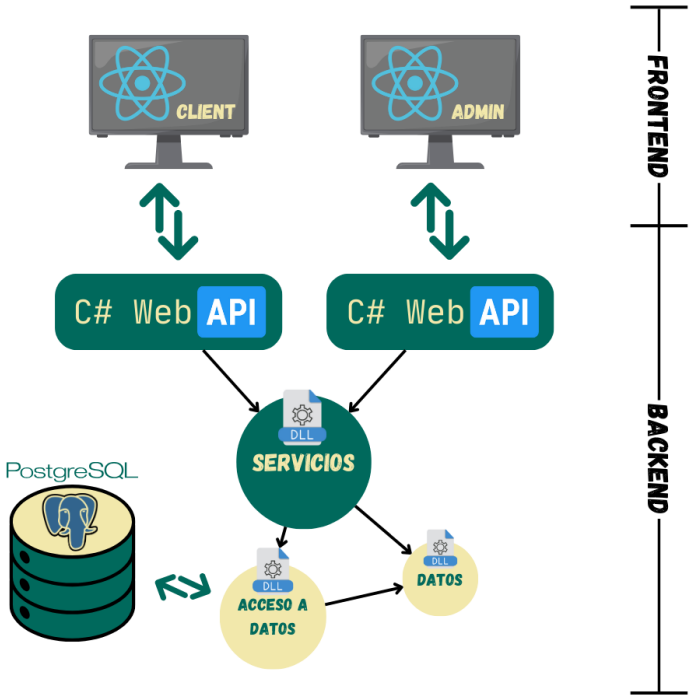

Figura 3.4: Estructura tecnológica del sistema

Para lograr esta adaptabilidad, se utilizó **Tailwind CSS**[10], un framework de CSS altamente flexible y eficiente que permite una personalización precisa y una creación de interfaces de usuario rápida y efectiva. Con Tailwind, fue posible aplicar un enfoque de diseño móvil primero, garantizando que la interfaz respondiera de manera eficiente a las diferentes resoluciones de pantalla sin perder la coherencia en su estructura visual. Además, el uso de clases utilitarias de Tailwind facilitó el diseño modular, lo que mejoró la mantenibilidad y escalabilidad del sistema.

Adicionalmente, se implementó **Flowbite**[11], una biblioteca de componentes basada en Tailwind CSS, para acelerar el desarrollo y asegurar una interfaz de usuario consistente. Flowbite proporcionó una serie de componentes preconstruidos y personalizables, como botones, formularios y modales, lo que permitió integrar elementos

[10] Visitar en `https://tailwindcss.com`
[11] Visitar en `https://flowbite.com`

interactivos con facilidad y sin necesidad de construir cada componente desde cero. Gracias a la combinación de Tailwind y Flowbite, se logró una interfaz visualmente atractiva, funcional y alineada con los principios de diseño del sitio.

Por otro lado, en el desarrollo del backend de BotaniQ, se implementaron las APIs necesarias para exponer los servicios que permiten la interacción entre el frontend y la base de datos. Estas APIs fueron diseñadas cuidadosamente para garantizar que solo se expusieran los servicios relevantes a cada uno de los dos proyectos de Web API, lo que contribuye a una arquitectura más organizada y segura, restringiendo el acceso a datos innecesarios.

Para la interacción con la base de datos, se utilizó Entity Framework Core[12] como Object-Relational Mapper (ORM, por sus siglas en inglés), bajo un enfoque *"Code First"*, lo que permitió definir las entidades de la base de datos directamente en el código. Este enfoque simplificó el proceso de creación y mantenimiento del esquema de la base de datos, ya que las clases de C# fueron las que definieron la estructura de las tablas y sus relaciones. Gracias a este ORM, se facilitó la gestión de las entidades y la realización de consultas, permitiendo que el acceso a los datos fuera más eficiente y seguro.

En cuanto a la base de datos, se realizó una población inicial de la misma utilizando la información extraída del libro de plantas medicinales. Esta operación se ejecuta la primera vez que se inicia la Web API del administrador, permitiendo que los datos extraídos de manera estructurada sean cargados en la base de datos de manera automática. Esto asegura que la base de datos esté correctamente inicializada y preparada para su uso desde el inicio.

Una característica fundamental del sistema es la capacidad de realizar búsquedas por contexto. Este mecanismo permitirá que el usuario pueda obtener resultados más relevantes y precisos basados en el contexto de su consulta, mejorando la experiencia de búsqueda y garantizando que los datos obtenidos sean los más apropiados para el usuario en cada situación. Este aspecto será detallado con más profundidad a continuación, pues constituye una de las funcionalidades clave incluidas en BotaniQ.

[12]Visitar en `https://www.nuget.org/packages/EntityFramework`

3.3.1. La búsqueda por contexto

HABLAR DE ESTO

3.4. Experimentación

Conclusiones

Conclusiones

Recomendaciones

Recomendaciones

Bibliografía

[1] Ricardo Baeza-Yates. «A Formal Characterization of IR Models». En: *Modern Information Retrieval*. 1999. Cap. 2.4, pág. 23 (vid. pág. 16).

[2] Ricardo Baeza-Yates. «Modern Information Retrieval». En: 1999, págs. 24-46 (vid. pág. 16).

[3] Erick Camacho. *NoSQL la evolución de las bases de datos*. Disponible en: https://sg.com.mx/revista/28/nosql-evolucion-bases-datos. Accedido el 22 de diciembre de 2024. URL: https://sg.com.mx/revista/28/nosql-evolucion-bases-datos (vid. pág. 21).

[4] Stefano Ceri. En: *Web Information Retrieval*. Springer, 2013. Cap. An Introduction to Information Retrieval, págs. 3-11. DOI: 10.1007/978-3-642-39314-3_1 (vid. pág. 15).

[5] «History of Databases». En: *The Relational Model*. Ed. por P. S. Chen y D. G. Lockwood. Springer, 2016, págs. 278-281. DOI: 10.1007/978-3-319-33138-6_20. URL: https://doi.org/10.1007/978-3-319-33138-6_20 (vid. pág. 20).

[6] E. F. Codd. «A Relational Model of Data for Large Shared Data Banks». En: *IBM Research Laboratory* (1970). Artículo disponible en: https://www.seas.upenn.edu/ zives/03f/cis550/codd.pdf. URL: https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf (vid. pág. 20).

[7] E. F. Codd. «A Relational Model of Data for Large Shared Data Banks». En: *IBM Research Laboratory* (1970). Artículo disponible en: https://www.seas.upenn.edu/ zives/03f/cis550/codd.pdf. URL: https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf (vid. pág. 20).

[8] EcuRed. *Jardín Botánico Nacional de Cuba*. Disponible en: *https://www.ecured.cu/*. Accedido el 9 de diciembre de 2024. URL: https://www.ecured.cu/ (vid. pág. 3).

[9] Víctor R. Fuentes Fiallo. «La Flora Medicinal de Cuba: un sueño de Roig no alcanzado». es. En: *Revista Cubana de Plantas Medicinales* 14 (sep. de 2009). Accedido el 20 de diciembre de 2024. ISSN: 1028-4796. URL: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1028-47962009000300001&nrm=iso (vid. pág. 8).

[10] Fan Gao. «Large Language Models on Wikipedia-Style Survey Generation: an Evaluation in NLP Concepts». En: *ArXiv* abs/2308.10410 (2023). Disponible en: *https://api.semanticscholar.org/CorpusID:261049765*. URL: https://api.semanticscholar.org/CorpusID:261049765 (vid. pág. 11).

[11] Cristina Crespo Garay. *¿Cuál fue el origen de la agricultura?* Disponible en: *https://www.nationalgeographic.es/historia/2022/01/cual-fue-el-origen-de-la-agricultura*. Accedido el 3 de diciembre de 2024. 2022 (vid. pág. 1).

[12] Ralph Grishman. «Information Extraction: Techniques and Challenges». En: *International Summer School on Information Extraction*. Disponible en: *https://api.semanticsch* 1997. URL: https://api.semanticscholar.org/CorpusID:17479975 (vid. pág. 11).

[13] David A. Grossman. *Information Retrieval: Algorithms and Heuristics*. 2nd. Disponible en: *https://doi.org/10.1007/978-1-4020-3005-5*. Citado en la página 9. 2004. ISBN: 978-1-4020-3004-8. DOI: 10.1007/978-1-4020-3005-5 (vid. pág. 16).

[14] Redacción Cadena Habana. *Jardín Botánico Nacional*. Disponible en: *https://www.cadenahaban botanico-nacional-03082022/*. Accedido el 9 de diciembre de 2024. 2022. URL: https://www.cadenahabana.icrt.cu/jardin-botanico-nacional-03082022/ (vid. pág. 3).

[15] IBM. *Teorema CAP*. Disponible en: https://ibm.com/mx-es/topics/cap-theorem. Accedido el 23 de diciembre de 2024. URL: https://ibm.com/mx-es/topics/cap-theorem (vid. pág. 22).

[16] *Information Extraction in NLP*. Disponible en: https://www.geeksforgeeks.org/information-extraction-in-nlp. Accedido el 13 de diciembre de 2024. Jun. de 2024. URL: https://www.geeksforgeeks.org/information-extraction-in-nlp (vid. pág. 11).

[17] «International Code of Nomenclature for algae, fungi and plants (Melbourne Code) adopted by the Eighteenth International Botanical Congress Melbourne, Australia, July 2011». En: vol. 154. Dic. de 2012, págs. 1-240. ISBN: 978-3-87429-425-6 (vid. pág. 8).

[18] Christopher D. Manning. «Introduction to Information Retrieval». En: Cambridge University Press, 2008, págs. 1-3. ISBN: 978-0-511-41405-3 (vid. pág. 15).

[19] Juan Tomás Roig y Mesa. *Plantas medicinales, aromáticas o venenosas de Cuba*. Vol. I-II. La Habana, Cuba: Editorial Científico-Técnica, 1945 (vid. págs. 2, 7, 8).

[20] Rafael Angel Ocampo. «Situación actual del comercio de plantas medicinales en América Latina». En: *Boletín Latinoamericano y del Caribe de Plantas Medicinales y Aromáticas* (2002). Artículo disponible en: *http://www.redalyc.org/articulo.oa?id=8561* URL: http://www.redalyc.org/articulo.oa?id=85610403 (vid. págs. 1, 2).

[21] Gursev Pirge. *The Complete Guide to Information Extraction from Texts with Spark NLP and Python*. Accedido el 13 de diciembre de 2024. Mayo de 2023 (vid. pág. 11).

[22] Tamás Pôcs. «BIOGEOGRAPHY OF THE CUBAN BRYOPHYTE FLORA». En: *TAXON* 37.3 (1988). Disponible en: *https://onlinelibrary.wiley.com/doi/abs/10.2307/1221* pages 615-621. DOI: https://doi.org/10.2307/1221103. URL: https://onlinelibrary.wiley.com/doi/abs/10.2307/1221103 (vid. pág. 7).

[23] Pedro López Puig. «Integración de la medicina natural y tradicional cubana en el sistema de salud». En: *Revista Cubana de Salud Pública* (2019). Artículo disponible en: *https://revsaludpublica.sld.cu/index.php/spu/article/view/1168/1240*, págs. 3-5. URL: https://revsaludpublica.sld.cu/index.php/spu/article/view/1168/1240 (vid. pág. 2).

[24] Kalpana Raja. «Template Filling, Text Mining». En: *Encyclopedia of Systems Biology*. Ed. por Werner Dubitzky et al. Springer New York, 2013, págs. 2150-2154. ISBN: 978-1-4419-9863-7. DOI: 10.1007/978-1-4419-9863-7_173. URL: https://doi.org/10.1007/978-1-4419-9863-7_173 (vid. pág. 12).

[25] Dr. Francisco J. Morón Rodríguez. «Necesidad de investigaciones sobre plantas medicinales». En: *Revista Cubana de Plantas Medicinales* (dic. de 2007). Accedido el 20 de diciembre de 2024. ISSN: 1028-4796. URL: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1028-47962007000400001&nrm=iso (vid. pág. 7).

[26] Stuart J Russell. «Artificial intelligence: A modern approach». En: Pearson, 2020, págs. 849-851 (vid. pág. 9).

[27] G. Salton. «The Automatic Analysis of Document Collections: An Experiment in Bibliographic Description». En: *Communications of the ACM* 8.6 (1965). Disponible en: *https://dl.acm.org/doi/10.1145/364955.364990*, pages 391-398. DOI: 10.1145/364955.364990 (vid. pág. 14).

[28] Amazon Web Services. *The Difference Between ACID and BASE Database*. Disponible en: https://aws.amazon.com/es/compare/the-difference-between-acid-and-base-database/. Accedido el 23 de diciembre de 2024. URL: https://aws.amazon.com/es/compare/the-difference-between-acid-and-base-database/ (vid. pág. 21).

[29] Allen G. Taylor. *Database Development For Dummies®*. Se consultó el Capítulo 1. Indianapolis, Indiana: Wiley Publishing, Inc., 2001. ISBN: 0-7645-0752-4. URL: https://books.google.com/books/about/Database_Development_For_Dummies.html?id=p58OYKP6Pg4C (vid. pág. 19).

[30] Wikipedia. *Rollo (manuscrito)*. Accedido el 23 de diciembre de 2024. URL: https://es.wikipedia.org/wiki/Rollo_(manuscrito) (vid. pág. 14).

[31] Wikipedia. *World Wide Web*. Accedido el 24 de diciembre de 2024. URL: https://en.wikipedia.org/wiki/World_Wide_Web/ (vid. pág. 14).

[32] C. Romero Zarco. *Principios de Botánica Sistemática*. Accedido el 10 de enero de 2025. 2017. URL: https://personal.us.es/zarco/PIM-Botanica/Temas/PIM_t1/T1_2A_Taxonomia.html (vid. pág. 9).