



POLITÉCNICA

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES  
UNIVERSIDAD POLITÉCNICA DE MADRID

José Gutiérrez Abascal, 2. 28006 Madrid  
Tel.: 91 336 3060  
info.industriales@upm.es

[www.industriales.upm.es](http://www.industriales.upm.es)



Sofía Hernangómez Flores

05 TRABAJO FIN DE GRADO

INDUSTRIALES

TRABAJO FIN DE GRADO

# SISTEMA DE CONOCIMIENTO PARA ROBÓTICA COGNITIVA SOCIAL

JUNIO 2021

**Sofía Hernangómez Flores**

DIRECTOR DEL TRABAJO FIN DE GRADO:

**Daniel Galán Vicente**



POLITÉCNICA



“Si una máquina se comporta  
en todos los aspectos como inteligente,  
entonces debe ser inteligente”.

Alan M. Turing. Padre de la Inteligencia Artificial  
Prueba de Turing (1950)

# AGRADECIMIENTOS

En primer lugar y como no podía ser de otra forma, me gustaría agradecerle a Daniel Galán Vicente la oportunidad que me ha brindado al permitirme formar parte de este proyecto, siendo mi apoyo y mi guía en la recta final de esta primera etapa de mis estudios universitarios, así como los conocimientos transmitidos durante el desarrollo de este Trabajo Fin de Grado.

Siguiendo en el ámbito académico, me veo en la necesidad de darle las gracias a todos aquellos docentes que me han motivado a continuar formándome durante los cinco años que he permanecido cursando el Grado en Ingeniería en Tecnologías Industriales. Gracias a vosotros he descubierto mundos totalmente desconocidos que me han fascinado, como la programación, la automática y la robótica.

Por otro lado, me gustaría agradecer a las personas que me sirvieron de inspiración para introducirme en el ámbito de la Ingeniería Industrial, Javier Padilla Moreno y Albert Espinosa i Puig.

En el ámbito personal, quiero agradecerle a mis padres y a mi hermana la paciencia, la compañía en los éxitos y en los fracasos y el apoyo moral y económico. Si no me he rendido en todos estos años ha sido gracias a vosotros.

A mis amigos de toda la vida, por estar siempre a mi lado y a mis compañeros y amigos etisianos. Son muchas las cosas que he aprendido gracias a vosotros:

He aprendido que la ingeniería no trata exclusivamente de adquirir conocimientos técnicos y aprobar exámenes.

He aprendido a ofrecer ayuda a quien la necesita.

He aprendido el significado de compañerismo, ya que en esta Escuela siempre hay alguien dispuesto a ayudarte.

Y, por encima de todo, he aprendido a confiar en mi capacidad para seguir adelante, sin importar las dificultades del camino.

Gracias a todos.

# RESUMEN

Se presenta el Trabajo Fin de Grado “Sistema de conocimiento para robótica cognitiva social”.

Como su propio nombre indica, para poder entender el alcance de este proyecto es necesario estar familiarizado con los términos “sistema de conocimiento” y “robótica cognitiva social”, los cuales se explican a continuación.

Si bien la mayoría de las empresas cuentan con robots industriales en sus fábricas de producción, los robots de servicio y, más concretamente, los robots destinados a aplicaciones sociales solo están al alcance de algunos privilegiados.

Esto se debe a que, a pesar de que los conocimientos en Inteligencia Artificial han aumentado considerablemente en los últimos años, aún queda mucho por hacer; siendo la representación del conocimiento uno de los principales problemas sin resolver.

Así, en un mundo en el que la tecnología avanza a gran velocidad, nace la robótica cognitiva, una rama de la robótica que trata de integrar el razonamiento, la percepción y la acción en robots de comportamiento inteligente.

Para lograr esa meta son necesarios los sistemas o bases de conocimiento. Pero ¿qué es una base de conocimiento? Fundamentalmente, se trata de bases de datos que permiten centralizar toda la información disponible sobre un tema concreto, estando diseñadas, generalmente, desde el punto de vista de la persona que va a realizar la consulta.

En este contexto, el objetivo principal del proyecto es presentar una posible solución al problema de estructuración, almacenamiento y recuperación de conocimiento en robots sociales.

En cuanto a la metodología seguida, en base a los antecedentes existentes y al estudio y análisis de los diferentes modos de representación del conocimiento, se ha decidido desarrollar un sistema formado por bases de datos de modelo no relacional.

Para ello, el alumno se ha servido de MongoDB como servidor de bases de datos y de Microsoft Visual Studio como entorno de desarrollo, llevando a cabo una programación en C++.

Como resultado, se ha obtenido un sistema capaz de realizar operaciones CRUD (Create, Read, Update, Delete) con la información que compone la base de conocimiento. Es decir, utilizando la interfaz gráfica desarrollada se puede interaccionar con el conocimiento mediante el acceso, creación, modificación o eliminación de bases de datos, colecciones, documentos y archivos multimedia.

Además, para diferenciarse de una base de datos común, se ha incluido la posibilidad de establecer relaciones entre términos: tanto entre dos documentos como entre un documento y un archivo multimedia.

Al tratarse del paso previo a la implantación, el alumno no ha entrado en contacto con ningún robot social durante el desarrollo del proyecto, haciéndose cargo única y exclusivamente del diseño y la programación de la base de conocimiento.

No obstante, con el objetivo de otorgar portabilidad al programa y facilitar la posterior integración en robots, el código está muy estructurado en funciones equivalentes a cada uno de los botones representados en la interfaz.

En el futuro se espera poder realizar pruebas en sistemas físicos reales, como el robot de asistencia personal POTATO desarrollado por el Grupo de Control Inteligente de la Universidad Politécnica de Madrid, al que estaba inicialmente dirigido este proyecto.

#### **PALABRAS CLAVE:**

Inteligencia Artificial, robot social, sistema de conocimiento, robótica cognitiva, base de datos.

#### **CÓDIGO UNESCO:**

330419. Ciencias tecnológicas. Tecnología de los ordenadores. Robótica.

#### **CÓDIGO DEL PROYECTO:**

<https://github.com/SHernangomez/Base-de-conocimiento.git>

# ABSTRACT

This paper describes the design of a knowledge base for the Engineering Degree Final Project “Knowledge system for social-cognitive robotics”.

To understand the scope of the project, it is necessary to be familiar with the terms “knowledge system” and “social-cognitive robotics”, the definitions of which are explained in the following paragraphs.

While most companies have industrial robots in their production plants, service robots and, more specifically, robots for social applications are only available to a privileged few. This is because, although Artificial Intelligence skills have grown considerably in recent years, there is still a long way to go, with knowledge representation being one of the main unsolved problems.

Thus, in a world in which technology advances at a great speed, a branch of robotics that tries to integrate reasoning, perception, and action in robots with intelligent behaviour was born. This is what we call cognitive robotics.

To make it possible, knowledge systems, commonly known as knowledge bases, are necessary. However, what is a knowledge system? In essence, they are databases that centralize all available information on a specific topic, generally designed from the point of view of the person who is going to make the query.

In this context, the aim of this work is to present a possible solution to the problem of structuring, storing, and retrieving knowledge for social robots.

Based on the study of the different existing ways to represent knowledge, the methodology used consists of developing a system made up of non relational databases.

To achieve this goal, the student has used the MongoDB database server and the Microsoft Visual Studio development environment to program a knowledge base using C++.

As a result, it has been obtained a system which is capable of performing CRUD (Create, Read, Update, Delete) operations on the database information using the developed graphical interface to interact with the knowledge by accessing, creating, modifying, or deleting databases, collections, documents, and multimedia files.

In addition, to make the difference between our system and a common database, it includes the possibility of establishing relations between two documents and between a document and a multimedia file.

It is important to emphasize that the current study is limited to the design and programming of the knowledge base, which is why the student has not come into contact with any social robot during the development of the project.

However, as a preliminary step for subsequent integration into robots, the program has been made portable developing a highly structured code made up of functions equivalent to the buttons represented on the interface.

Given these points, future studies should focus on testing the knowledge base on physical systems, such as POTATO, a personal assistance robot developed by researchers from the Intelligent Control Group of the Polytechnic University of Madrid.

#### KEYWORDS:

Artificial Intelligence, social robot, knowledge system, cognitive robotics, database.

#### PROGRAM CODE AT:

<https://github.com/SHernangomez/Base-de-conocimiento.git>



# ÍNDICE

<b>INTRODUCCIÓN.....</b>	<b>10</b>
1.1. MOTIVACIÓN .....	10
1.2. MARCO GENERAL DEL PROYECTO .....	11
1.3. ANTECEDENTES .....	12
1.4. OBJETIVOS .....	16
<b>ESTADO DEL ARTE .....</b>	<b>17</b>
2.1. ROBÓTICA .....	17
2.1.1. Breve historia de la robótica .....	17
2.1.2. Clasificación de los robots .....	18
2.1.3. Robótica cognitiva.....	19
2.2. INTELIGENCIA ARTIFICIAL.....	21
2.2.1. Origen .....	21
2.2.2. Definición .....	21
2.2.3. Aplicaciones.....	22
2.2.4. Inteligencia Artificial en robots .....	22
2.3. INGENIERÍA DEL CONOCIMIENTO .....	24
2.3.1. Definición .....	24
2.3.2. Tipos de conocimiento .....	24
2.3.3. Representación del conocimiento .....	25
<b>METODOLOGÍA .....</b>	<b>27</b>
3.1. HERRAMIENTAS SOFTWARE.....	27
3.1.1. Servidor de bases de datos.....	27
3.1.2. Sistema operativo .....	30
3.1.3. Lenguaje de programación.....	31
3.1.4. Entorno de desarrollo .....	31
3.2. PREPARACIÓN DEL ENTORNO DE TRABAJO .....	32
3.2.1. Instalación de Microsoft Visual Studio 2019 .....	32
3.2.2. Instalación API Formularios de Windows para C++ .....	33
3.2.3. Instalación MongoDB .....	34
3.2.4. Instalación del controlador de MongoDB para C++ .....	36
3.2.5. Preparación Microsoft Visual Studio 2019 para MongoDB.....	38

<b>RESULTADOS.....</b>	<b>40</b>
4.1. BASE DE CONOCIMIENTO .....	41
4.1.1. Bases de datos .....	42
4.1.2. Colecciones .....	43
4.1.3. Documentos .....	45
4.1.4. Archivos multimedia .....	56
4.2. INTERFAZ GRÁFICA .....	61
4.2.1. Bases de datos y colecciones .....	62
4.2.2. Documentos .....	65
4.2.3. Archivos multimedia .....	72
<b>CONCLUSIONES.....</b>	<b>76</b>
5.1. CONCLUSIONES DEL PROYECTO .....	76
5.1.1. Sistema de conocimiento .....	76
5.1.2. Interfaz gráfica .....	77
5.1.3. Portabilidad .....	78
5.2. LÍNEAS FUTURAS.....	79
5.2.1. Implementación.....	79
5.2.2. Filtrado de información .....	79
5.2.3. Sistema de aprendizaje .....	79
5.3. VALORACIÓN DE IMPACTOS Y DE ASPECTOS DE RESPONSABILIDAD LEGAL, ÉTICA Y PROFESIONAL.....	80
5.4. PLANIFICACIÓN TEMPORAL.....	81
5.4.1. Estructura de Descomposición del Trabajo (EDT) .....	81
5.4.2. Diagrama de Gantt .....	82
5.5. PRESUPUESTO .....	83
5.5.1. Materia prima .....	83
5.5.2. Mano de obra .....	83
5.5.3. Presupuesto total .....	83
<b>BIBLIOGRAFÍA.....</b>	<b>84</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>88</b>
<b>ÍNDICE DE TABLAS.....</b>	<b>90</b>
<b>ABREVIATURAS, UNIDADES Y ACRÓNIMOS .....</b>	<b>91</b>
<b>GLOSARIO .....</b>	<b>92</b>

## Capítulo 1

# INTRODUCCIÓN

### 1.1. MOTIVACIÓN

Desde la publicación en 1920 de la obra de teatro Rossum's Universal Robots (R.U.R.) por el escritor Karel Capek, los investigadores del mundo de la robótica han dedicado grandes esfuerzos para hacer posible el desarrollo de robots interactivos.

La Federación Internacional de Robótica (IFR) define al robot de servicio como "un robot que realiza tareas útiles para humanos o equipos, excluyendo las aplicaciones de automatización industrial". En la actualidad, su departamento de estadística tiene conocimiento de aproximadamente 890 empresas que producen este tipo de robots.

Los robots de servicio se pueden clasificar en distintos grupos, entre los que se encuentra el de los robots sociales, formado por aquellos robots que interactúan con los seres humanos de una manera socialmente aceptable. Entre las tareas que se les pretenden asignar en el futuro se encuentran las de utilizar los electrodomésticos del hogar, atender visitas y ayudar en el día a día a las personas mayores o con necesidades especiales.

Puesto que para hacer realidad la coexistencia entre humanos y robots en hogares se necesita un cierto grado de autonomía, no ha sido hasta finales de la primera década del siglo XXI cuando, gracias al desarrollo de la tecnología robótica, los robots domésticos sociales han comenzado a representar roles como el de asistente personal en la vida diaria de algunas personas. No obstante, según los expertos en la materia, este tipo de robots pronto se comercializarán y serán lanzados en masa a los mercados, acoplándose a la vida del hombre como lo hicieron los teléfonos móviles.

En este contexto, de la necesidad de los robots sociales de mantener conversaciones de alto nivel, nace la robótica cognitiva, una rama de la robótica que trata de solucionar el problema de las interacciones sociales robot-humano dotando al robot de una arquitectura de procesamiento que le permita aprender y razonar acerca de cómo comportarse.

Así, con el objetivo de acelerar la inserción de los robots en la vida cotidiana, la representación del conocimiento basada en ontologías ha ganado mucha importancia en los últimos años.

Una ontología es más que una taxonomía, se trata de un mecanismo para conceptualizar el conocimiento mediante la definición de conceptos pertenecientes a un dominio específico, pudiendo servir como base para la comunicación entre humanos y máquinas.

De esta forma, se puede concluir que compartir una ontología común equivale a compartir una visión común del mundo.

## 1.2. MARCO GENERAL DEL PROYECTO

Este Trabajo Fin de Grado (TFG) se enmarca en el panorama de los robots para aplicaciones sociales, cuya popularidad ha aumentado en los últimos años, llegando a ser necesarios en algunos aspectos de la vida cotidiana.

El Grupo de Control Inteligente de la Universidad Politécnica de Madrid, el cual forma parte del Centro de Automática y Robótica (CAR UPM-CSIC), desarrolla desde el año 2000, apoyándose en Trabajos Fin de Grado y Trabajos Fin de Máster, distintos robots para aplicaciones sociales, entre los que se encuentran URBANO, DORIS y POTATO.

Concretamente, este proyecto trata de modelar el conocimiento de POTATO, un robot social formado por una Tablet a la que se han ido añadiendo diferentes módulos y sensores a lo largo de los años.

Potato es capaz de comunicarse con el usuario de forma hablada, simulando al mismo tiempo inteligencia emocional mediante gestos, de manera que en el futuro podría llegar a convertirse en un robot de asistencia personal.

### 1.3. ANTECEDENTES

Uno de los primeros robots capaces de demostrar interacciones sociales con el ser humano es Kismet, fabricado a finales de la década de 1990 en el Instituto de Tecnología de Massachusetts (MIT). Puesto que, además de poseer la habilidad de realizar tareas y de entender el lenguaje humano, un robot social debe entender la empatía y ser capaz de hacer amigos, el sistema de software de Kismet se basa en modelos humanos de comportamiento inteligente, estando constituido por un sistema de percepción de alto nivel, un sistema de motivación, un sistema de comportamiento, un sistema de habilidades sociales y un sistema motor facial.

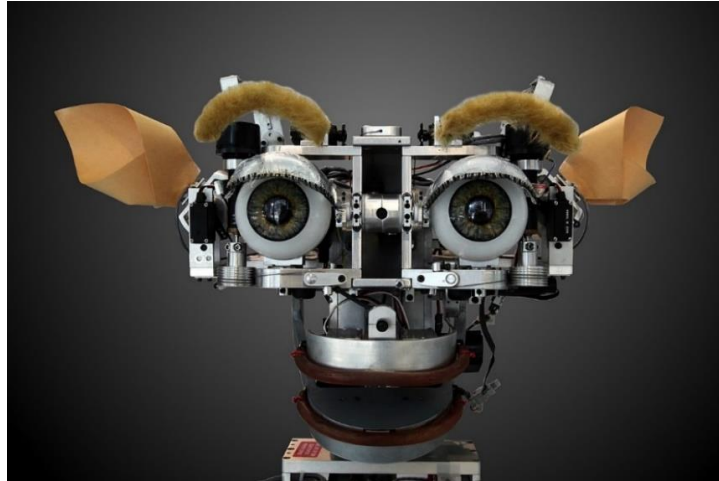


Figura 1: Kismet. MIT. 1998.

Fuente: Wikipedia

La Universidad de Valladolid tomó a Kismet como inspiración para la presentación en 2008 de Arisco, un robot social que cuenta con capacidades de interacción, motivación y aprendizaje. Su sistema de aprendizaje se basa en la realización de asociaciones estímulo-respuesta a través de una red neuronal que es capaz de procesar estímulos sonoros, visuales, infrarrojos y de habla.

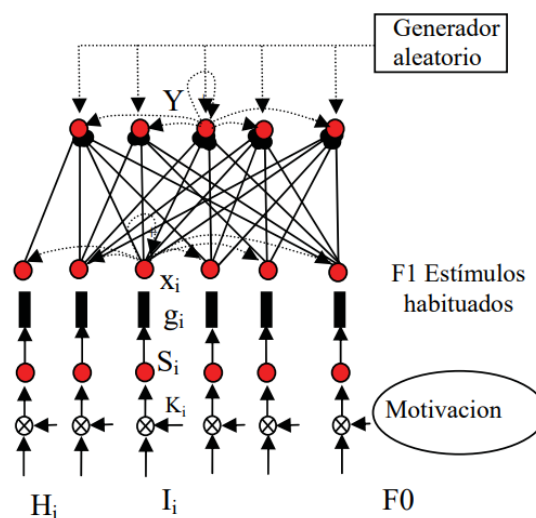


Figura 2: Arquitectura neuronal de Arisco. Universidad de Valladolid. 2008.

Fuente: Universidad de Valladolid

Unos años más tarde, los avances en el mundo de la robótica permitieron la presentación en 2014, por parte de la empresa japonesa SoftBank Group y la francesa Aldebaran Robotics, de Pepper, un robot humanoide capaz de comunicarse con el ser humano de forma natural, reconocer emociones e incluso entretener a la gente haciendo bromas.

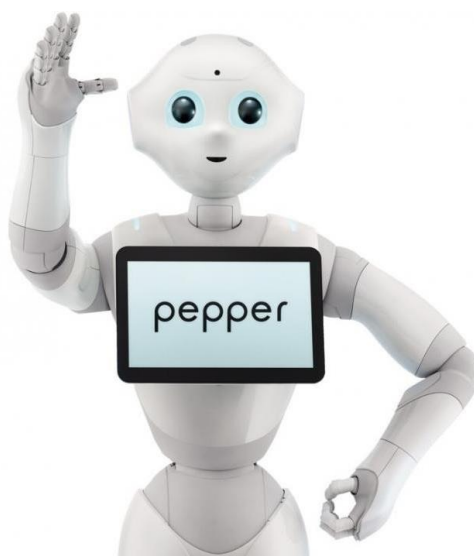


Figura 3: Pepper. SoftBank Robotics. 2014.

Fuente: PNGWing

En 2015, la Universidad de Osaka y Kyoto y el Advanced Telecommunications Research Institute International (ATR) presentaron a Erica, un robot dotado de una gran autonomía intelectual que le permite imitar el comportamiento humano a la hora de conversar, mostrar expresiones faciales y expresarse mediante lenguaje corporal. En 2020 se anunció que Erica protagonizaría una película de ciencia ficción, convirtiéndose de este modo en el primer robot del mundo en lograr algo así.



Figura 4: Erica. Hiroshi Ishiguro. 2015.

Fuente: Diario Página/12

En 2015, la compañía Hanson Robotics creó a Sophia, un robot humanoide que en 2017 adquirió la nacionalidad saudí, convirtiéndose así en la primera ciudadana robot del mundo. Con el objetivo de imitar el pensamiento humano, el creador del software de Sophia incorporó búsquedas de información para que el robot pueda responder correctamente a las preguntas que se le plantean y realizó una asociación entre el sistema de voz y las expresiones faciales de Sophia.



Figura 5: Sophia. Hanson Robotics. 2015.

Fuente: Revista Alto Nivel

En 2019, la empresa madrileña Grupo ADD lanza Tokyo the Robot, el primer robot social español. Entre sus múltiples funcionalidades destaca la de atender a personas mayores o con necesidades especiales a través de ejercicios cognitivos.



Figura 6: Tokyo the Robot. Grupo ADD. 2019.

Fuente: Grupo ADD

En cuando a las bases de conocimiento que se han desarrollado a lo largo de la historia de la informática, podemos destacar Yet Another Great Ontology (YAGO) y Tufts University Sciences Knowledgebase (TUSK), ambas de código abierto.

YAGO es una base de conocimiento de alta calidad desarrollada en el Instituto Max Planck de Ciencias de la Computación, en Alemania. Se trata de una ontología anclada en el tiempo y el espacio que extrae información de Wikipedia, WordNet y GeoNames con una precisión superior al 95%.

TUSK es una base de conocimiento desarrollada por la Universidad Tufts, en Estados Unidos. Se trata de un sistema dinámico de gestión del conocimiento que respalda la enseñanza en las facultades de ciencias de la salud.

En última instancia, vamos a analizar algunos de los proyectos desarrollados por el alumnado de la Universidad Politécnica de Madrid.

Por un lado, es importante señalar la tesis doctoral “Análisis y diseño de sistemas cognitivos para robots autónomos”, presentada por Ignacio J. Chang J. en 2009. En ella, propone el uso de una ontología informática diseñada en lenguaje OWL para representar el conocimiento relacionado con la toma de decisiones.

Por otro lado, como parte del proyecto URBANO del Grupo de Control Inteligente, cabe destacar el Trabajo Fin de Grado “Aprendizaje en robots sociales” y el Trabajo Fin de Máster “Adquisición de conocimiento de un robot social”, ambos desarrollados por Koro Irusta Gonzalo durante los años 2017 y 2019, respectivamente. En ellos, la alumna desarrolla una ontología en lenguaje OWL mediante el uso de la herramienta Protégé, incorporando además un sistema de aprendizaje automático y un sistema de filtrado de información.



### 1.4. OBJETIVOS

Este proyecto pretende abordar el problema de cómo estructurar, almacenar y recuperar el conocimiento adquirido por un robot social de manera eficiente, lo que en Inteligencia Artificial se conoce como representación del conocimiento.

Para ello, el alumno deberá llevar a cabo un estudio sobre los diferentes métodos que la robótica cognitiva propone para almacenar conocimiento en robots inteligentes y, posteriormente, seleccionar uno de ellos.

Así, el objetivo principal de este Trabajo Fin de Grado será la creación de un sistema de conocimiento que, en términos generales, permita al usuario:

- Acceder a la información contenida en la base de conocimiento.
- Añadir nuevos datos.
- Modificar la información existente.
- Eliminar datos concretos.
- Establecer relaciones entre términos.

Por otro lado, es importante destacar que se trata del paso previo a la implantación en robots, por lo que el alumno no entrará en contacto con ningún robot social, haciéndose cargo única y exclusivamente del diseño y la programación. De este modo, para lograr los objetivos propuestos, es necesario desarrollar una interfaz gráfica que permita al usuario interactuar con el contenido de la base de conocimiento.

Además, el alumno deberá tratar de facilitar la posterior implantación del programa en robots, desarrollando el programa de manera portable mediante la estructuración del código en funciones.

## Capítulo 2

# ESTADO DEL ARTE

## 2.1. ROBÓTICA

### 2.1.1. Breve historia de la robótica

Aunque el origen de los primeros mecanismos animados se encuentra en los relojes de agua ideados por Ctesibius en el año 270 a.C., la palabra robot fue usada por primera vez en el año 1921, en el estreno de la obra Rossum's Universal Robot, del escritor Karel Capek, en el teatro nacional de Praga.

Posteriormente, en 1942, el también escritor Isaac Asimov enunció las tres leyes de la robótica:

1. Un robot no puede perjudicar a un ser humano, ni con su inacción permitir que un ser humano sufra daño.
2. Un robot ha de obedecer las órdenes recibidas de un ser humano, excepto si tales órdenes entran en conflicto con la primera ley.
3. Un robot debe proteger su propia existencia mientras tal protección no entre en conflicto con la primera o segunda ley

En 1948, R. C. Goertz desarrolló, con el objetivo de operar con elementos radioactivos sin riesgo para el ser humano, el primer sistema de tele manipulación maestro-esclavo; convirtiéndolo, en 1954, en un sistema electrónico con servo control, que dio lugar al concepto de robot.

Unos años más tarde, en 1961, tras la fundación de la Consolidated Controls Corporation, George C. Devol y Joseph F. Engelberger sentaron las bases del robot industrial mediante la instalación de una Máquina de Transferencia Universal para fundición en la fábrica de General Motors, en Trenton. Este hecho motivó a otras empresas estadounidenses y japonesas a desarrollar máquinas similares contribuyendo, de este modo, a la popularización del término robot.

No obstante, la investigación en robótica en las universidades no comenzó hasta la década de los setenta, cuando Victor Scheinman desarrolló, en la Universidad de Standford, uno de los primeros brazos manipuladores. Además, gracias a las tecnologías de tele operación, en esta misma década, empezaron a fabricarse robots para realizar trabajos en el entorno espacial y el submarino.

En Europa, por su parte, Suecia tomó la iniciativa con la fundación en 1980 de la Federación Internacional de Robótica, con sede en Estocolmo y, posteriormente, Alemania se convirtió en el primer país en instalar una línea de soldadura formada exclusivamente por robots, gracias a la empresa KUKA.

En cuanto a los robots que imitan el aspecto de los seres vivos, en 1985 un profesor de la Universidad de Waseda creó a Wasubot, un robot humanoide capaz de tocar el piano y leer partituras; mientras que, en 1999, la compañía Sony desarrolló un robot mascota llamado Aibo, capaz de responder a estímulos de un ser humano.

Actualmente, la robótica se encuentra presente en la mayoría de los hogares gracias a la aspiradora autónoma Roomba, comercializada por la empresa iRobot en 2002.

En el futuro, por su parte, la investigación en robótica se centrará en mejorar la precisión y movilidad de los robots, así como en hacer posible las interacciones humano-robot, siendo la robótica social la base de este último propósito.

### 2.1.2. Clasificación de los robots

En la actualidad, podemos clasificar los robots utilizando diferentes métodos, siendo los más comunes la clasificación según su cronología y la clasificación según el área de aplicación.

#### 2.1.2.1. Clasificación según la cronología

La clasificación de robots atendiendo a la cronología se basa en generaciones, las cuales hacen referencia al momento tecnológico en el que aparece un robot.

Generación	Tipo	Características
Primera	Robots manipuladores	Repiten tareas programadas de manera secuencial, sin prestar atención a las posibles alteraciones de su entorno.
Segunda	Robots de aprendizaje	Pueden adquirir cierta información del entorno y adaptarse a las circunstancias, además de memorizar secuencias de movimientos a partir de los movimientos de un operador humano.
Tercera	Robots con control sensorizado	Son programados a través de lenguajes de programación que les otorgan la habilidad de actuar en consecuencia a la información obtenida del entorno a través de sus sensores.
Cuarta	Robots inteligentes	Están basados en técnicas de Inteligencia Artificial que les permiten adaptarse y aprender de su entorno utilizando diferentes métodos de análisis y obtención de datos.

Tabla 1: Clasificación de los robots atendiendo a su cronología.

#### 2.1.2.2. Clasificación según el área de aplicación

Aunque la mayor parte de los robots actuales se utilizan en aplicaciones industriales (ensamblado, soldadura, líneas de producción...), atendiendo al área de aplicación podemos distinguir dos tipos de robots: los robots industriales y los robots de servicio.

Un robot industrial manipula materias, piezas o herramientas siguiendo trayectorias variables; mientras que un robot de servicio lleva a cabo tareas o servicios útiles para el bienestar humano, excluyendo operaciones de manufactura, de forma total o parcialmente autónoma.

A continuación, se muestran las diferentes aplicaciones de los robots de servicio existentes en la actualidad.

Aplicación	Ejemplos
Robots personales y domésticos	Tareas domésticas: aspiradores.
	Entretenimiento: juguetes robotizados.
	Ayuda a discapacitados: sillas de ruedas robotizadas.
	Seguridad y vigilancia de la vivienda.
Robots de servicios profesionales	Exteriores: agricultura, minería, espacio.
	Limpieza profesional: suelos, depósitos, tuberías.
	Sistemas de inspección: alcantarillas, tuberías.
	Construcción y demolición.
	Sistemas logísticos: mensajería.
	Medicina: sistemas de diagnóstico y rehabilitación.
	Defensa, rescate y seguridad: vehículos aéreos no tripulados.
	Submarinos.
Robots para I + D	Robots de laboratorio.
	Micro robots.
	Nano robots.
	Materiales.

Tabla 2: Aplicaciones de los robots de servicio.

Fuente: Fundamentos de robótica. McGraw Hill. Segunda edición.

### 2.1.3. Robótica cognitiva

#### 2.1.3.1. Definición

La robótica cognitiva es la rama de la robótica que trata de crear máquinas inteligentes basándose en la facultad que poseen los seres vivos para procesar información mediante la integración de la percepción, el conocimiento adquirido y el razonamiento.

A diferencia de la robótica tradicional, que se centra en las tareas de control, la lectura de sensores y el poder de manipulación; la robótica cognitiva persigue el objetivo de dotar a los robots de comportamientos inteligentes que les permitan aprender a comportarse frente a objetivos complejos y tomar decisiones en entornos dinámicos y desconocidos.

#### 2.1.3.2. Punto de partida

La robótica cognitiva utiliza la cognición animal como inspiración o punto de partida para el desarrollo de algoritmos computacionales de procesamiento de información.

Es decir, se apoya en la psicología y la investigación en neurociencia para llevar a cabo análisis de aspectos que los robots puedan imitar, como son: la memoria, la representación del conocimiento, los mecanismos neuronales, la motivación, el razonamiento y el aprendizaje.

#### 2.1.3.3. Características

La robótica cognitiva estudia el procesamiento de la percepción, la anticipación, la planificación y el razonamiento.

De esta forma, un robot cognitivo debe poder solucionar conflictos entre objetivos y salir de situaciones de incertidumbre, para lo cual necesita perfeccionar su comportamiento mediante

el aprendizaje de su propia experiencia y tomar las decisiones necesarias para cumplir los objetivos propuestos.

Un robot de este tipo presenta las siguientes características: conocimiento, creencias, preferencias, metas, actitud informativa y motivacional y, por último, coordinación motora que permita el desplazamiento en el mundo real y la interacción segura con objetos.

En el futuro, se espera que los robots cognitivos sean capaces de interactuar con humanos en entornos dinámicos, mostrando un comportamiento similar al de los seres vivos que les permita interactuar en tiempo real con el mundo y aprender de él.

#### 2.1.3.4. Aplicaciones

Entre las aplicaciones de la robótica cognitiva, podemos destacar: robots sociales, asistentes personales, robots asistentes para personas con discapacidad, robots autónomos de exploración espacial y robots de rescate y emergencias.

Mención especial merece Watson, un sistema capaz de responder cuestiones preguntadas en lenguaje natural, desarrollado por la empresa IBM en 2011.

Para hacer esto posible, posee la capacidad de adecuar sus respuestas a partir de las interacciones que tiene con humanos y la interpretación de información desestructurada; respondiendo a las preguntas mediante el uso de una base de datos local.

Con el objetivo de demostrar sus capacidades, Watson participó en un concurso de televisión de tipo pregunta-respuesta llamado Jeopardy, en el que consiguió batir a sus oponentes sin estar conectado a Internet durante el juego.



Figura 7: Watson en el concurso Jeopardy.

Fuente: La Vanguardia

## 2.2. INTELIGENCIA ARTIFICIAL

### 2.2.1. Origen

En los años 40, científicos de varias disciplinas del Instituto Tecnológico de Massachusetts (MIT) formaron un grupo que dio lugar a la cibernética, una ciencia que estudia los mecanismos de comunicación y regulación automática de los seres vivos con el objetivo de reproducirlos en sistemas artificiales.

La cibernética puede considerarse la ciencia precursora de la Inteligencia Artificial (IA), pero no fue hasta 1950 cuando un artículo publicado por Alan Turing planteó la famosa pregunta “¿puede pensar una máquina?”, que dio lugar al Test de Turing, una prueba que pretendía determinar si una máquina era inteligente o no en función a su capacidad para suplantar a un ser humano mediante el procesamiento del lenguaje y el razonamiento automatizado.

Finalmente, en 1956, el término Inteligencia Artificial como ciencia fue acuñado formalmente por John McCarthy, de la universidad de Stanford, en una conferencia celebrada en el Dartmouth College.

### 2.2.2. Definición

A pesar de que la Inteligencia Artificial es un fenómeno del que todo el mundo ha oído hablar, es difícil comprender en qué consiste exactamente, pues no existe una definición rigurosa para esta ciencia.

Si nos remontamos a sus orígenes en Dartmouth, la IA nace de la creencia de la posibilidad de describir cualquier elemento de la inteligencia humana de manera tan precisa que pueda ser emulado por una máquina. Así, esta ciencia permitiría crear, mediante la combinación de algoritmos, sistemas dotados de las mismas capacidades cognitivas que los seres humanos.

En definitiva, la IA es una ciencia que pretende desarrollar sistemas que muestren comportamientos inteligentes en entornos complejos, siendo capaces de interpretar datos externos, aprender de ellos y utilizar el conocimiento adquirido en la toma de decisiones.

Normalmente, un sistema basado en Inteligencia Artificial utiliza tecnologías capaces de identificar patrones y tendencias y determinar correlaciones dentro de grandes conjuntos de datos, lo que les permite formular predicciones, hacer sugerencias y ejecutar transacciones de forma automática. Para ello, se rige por los cuatro principios fundamentales que se presentan a continuación: capacidad de autoaprendizaje, mediante procedimientos de prueba y error o procesos de búsqueda de problemas similares ya resueltos; capacidad de interacción con el usuario, mediante la comprensión de órdenes y solicitudes y la ejecución de respuestas sólidas y claras; capacidad de reacción en tiempo real y, en último lugar, autonomía, pudiendo realizar una tarea asignada sin que el usuario intervenga.

Finalmente, es importante aclarar que podemos enfocar la Inteligencia Artificial de dos formas distintas: por un lado, pretende comprender el funcionamiento del cerebro humano y sus funciones cognitivas mediante la descomposición de la inteligencia en sus procesos más elementales (conexionismo) y, por otro lado, trata de reproducir la inteligencia humana mediante el desarrollo de sistemas expertos (ingeniería del conocimiento).

### 2.2.3. Aplicaciones

Gracias a los avances tecnológicos de los últimos años, hoy en día los sistemas de Inteligencia Artificial llevan a cabo muchas de las tareas que antes estaban reservadas única y exclusivamente a los humanos.

Así, la IA ha pasado a estar presente en la vida diaria de las personas de tal forma que encontramos aplicaciones prácticas en el transporte, el hogar, las oficinas, los hospitales, los bancos, etc.

Entre los logros más recientes de la Inteligencia Artificial, destacan: el reconocimiento de imágenes, la detección y clasificación de objetos y la protección contra amenazas de seguridad cibernética.

En nuestro día a día utilizamos diferentes aplicaciones de IA, entre las que se encuentran: el reconocimiento de voz, presente en asistentes virtuales como Siri o Alexa; el procesamiento del lenguaje natural, presente en chatbots y traductores online; y el análisis de datos, presente en filtros de correo no deseado y sistemas de recomendación utilizados en plataformas digitales como Netflix.

Por otro lado, el desarrollo de la tecnología ha permitido la implementación de sistemas basados en IA en distintos campos, entre los que podemos destacar: la mejora en la precisión y rapidez de los diagnósticos en asistencia sanitaria, los sistemas de recomendación que utilizan empresas de comercio, los chatbots presentes en servicios de atención al cliente, el desarrollo de aplicaciones para asesoramiento financiero e incluso la implementación de sistemas de formación para directivos de empresas.

En el futuro, se estima que la Inteligencia Artificial llegue a hacer sugerencias y predicciones sobre distintos aspectos de la vida humana y, además, permita que las máquinas realicen tareas peligrosas para el ser humano.

### 2.2.4. Inteligencia Artificial en robots

Una de las principales aplicaciones de la IA en las empresas tiene como objetivo aumentar la productividad, mejorar la seguridad en el trabajo y ahorrar tiempo a los trabajadores, todo ello mediante la expansión de la funcionalidad de los robots, utilizando la salida de los algoritmos de Inteligencia Artificial como entrada a estas máquinas físicas.

Las líneas de investigación actuales en este campo tratan de optimizar el rendimiento de robots y procesos, permitir que los robots detecten y respondan a su entorno y hacer posible la creación de robots que operen como sistemas de información interactivos.

#### 2.2.4.1. Optimización del rendimiento de robots y procesos

El rendimiento de un robot se puede optimizar mediante el análisis en tiempo real de los datos de los sensores, lo cual nos permite determinar la confiabilidad del robot y predecir si es probable que requiera mantenimiento y cuándo.

En cuanto al rendimiento de procesos en los que intervienen robots, podemos utilizar la Inteligencia Artificial para analizar los datos de todas las máquinas que se encuentran interconectadas, optimizando de este modo todo el proceso.

#### 2.2.4.2. Detección y respuesta de robots a un entorno

Uno de los grandes retos de la robótica móvil es la adaptación a entornos impredecibles, posible solo en casos en los que el robot tiene la capacidad de lidiar con eventos inesperados.

Si un robot tradicional encuentra un obstáculo en su camino, se detiene para evitar la colisión, pero no es capaz de encontrar una ruta alternativa hacia su objetivo; mientras que un robot que utiliza técnicas de IA puede construir un mapa de su entorno en tiempo real y conocer su ubicación dentro del mismo.

#### 2.2.4.3. Robots de información interactivos

Un ejemplo de aplicación de Inteligencia Artificial para la creación de robots de información interactivos son los robots móviles utilizados en hoteles, tiendas o aeropuertos, que son capaces de responder preguntas y conducir a los clientes a una ubicación solicitada, entre otras funcionalidades.



## 2.3. INGENIERÍA DEL CONOCIMIENTO

### 2.3.1. Definición

La ingeniería del conocimiento es una rama de la Inteligencia Artificial, que surge a finales de los años 70, con el objetivo de establecer metodologías que hagan posible la creación de Sistemas Basados en Conocimiento y Sistemas Expertos. Es decir, se trata de una disciplina que pretende representar el conocimiento y razonamiento humanos dentro de un sistema artificial.

Para ello, es necesaria la extracción del conocimiento relativo a un dominio específico de expertos humanos y su codificación de manera que pueda ser procesado e interpretado por un sistema; además de la colaboración entre el experto que posee el conocimiento y el ingeniero encargado de informatizarlo.

Uno de los principales problemas de esta disciplina es el hecho de que el conocimiento puede evolucionar tanto durante la fase de desarrollo del sistema como después, lo cual implica la necesidad de una depuración continua de la base de conocimiento, siendo este un proceso que puede prolongarse durante toda la vida del sistema.

Por último, cabe destacar que no se trata de un proceso individual, pues la ingeniería del conocimiento se divide en tres fases principales: la adquisición del conocimiento, la representación del conocimiento y la manipulación y validación del conocimiento.



Figura 8: Procesos de la ingeniería del conocimiento.

### 2.3.2. Tipos de conocimiento

Aunque el conocimiento cambia constantemente y es difícil de caracterizar con precisión, podemos distinguir los siguientes tipos de conocimiento:

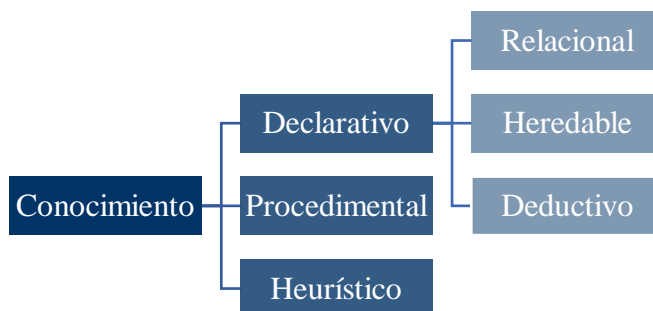


Figura 9: Tipos de conocimiento.

### 2.3.2.1. Conocimiento declarativo

El conocimiento declarativo tiene que ver con el “qué”. Se trata de un conocimiento de carácter consciente y explícito formado por sentencias sobre el mundo que nos rodea, hechos, conceptos e ideas que se pueden almacenar como proposiciones. Este tipo de conocimiento se divide, a su vez, en: conocimiento relacional, conocimiento heredable y conocimiento deductivo.

### 2.3.2.2. Conocimiento procedimental

El conocimiento procedimental tiene que ver con el “cómo”. Se trata de un conocimiento implícito e inconsciente que se adquiere gradualmente con la práctica y el aprendizaje de destrezas; es decir, indica qué pasos u operaciones seguir para completar una tarea determinada.

### 2.3.2.3. Conocimiento heurístico

El conocimiento heurístico tiene que ver con la resolución de problemas complejos. Se trata de un conocimiento obtenido generalmente por experimentación que da lugar a reglas sin justificación teórica.

## 2.3.3. Representación del conocimiento

La representación del conocimiento es un proceso de ingeniería del conocimiento que consiste en estructurar el conocimiento de tal forma que facilite la extracción de conclusiones a partir del mismo, posibilitando en todo momento un acceso rápido y preciso al contenido.

Una buena representación del conocimiento debe presentar las siguientes características: eficiencia, facilidad de actualización, cobertura amplia de la información para resolver ambigüedades y consistencia para evitar conocimiento redundante o conflictivo. Además, debe presentar de forma explícita los objetos y las relaciones importantes y ser comprensible por humanos, soportando modularidad y jerarquía de clases.

En cuanto a las metodologías de representación del conocimiento, podemos destacar: los esquemas lógicos, los esquemas basados en redes y la representación estructurada.

### 2.3.3.1. Esquemas lógicos

Este tipo de esquemas utiliza la lógica matemática como lenguaje de representación del conocimiento, pues permite codificar diferentes hechos y propiedades mediante formalismos, así como extraer conclusiones de los mismos.

Toda lógica que se utilice para representar el conocimiento debe constar de tres elementos: una sintaxis, que explique cómo construir los elementos; una semántica, que asocie los elementos del lenguaje de representación con los elementos reales del dominio y, finalmente, un conjunto de reglas de inferencia, que permitan deducir nuevo conocimiento a partir del ya existente.

Las expresiones representadas en lenguaje lógico se denominan fórmulas y están constituidas por términos, que pueden tomar la forma de variables, constantes o funciones.

Estos términos se pueden relacionar entre sí definiendo predicados, siendo la correspondencia entre la interpretación que hacemos de un predicado y el mundo real lo que nos dirá si el predicado es cierto o no.

Finalmente, los predicados pueden combinarse entre sí mediante conectivas lógicas representadas por símbolos, como son: la conjunción ( $\wedge$ ), la disyunción ( $\vee$ ), la implicación ( $\rightarrow$ ) y la negación ( $\neg$ ).

Uno de los formalismos lógicos más utilizados para representar el conocimiento es la lógica de primer orden, constituida por la lógica proposicional y la lógica de predicados, ya que permite razonar sobre objetos aislados y sobre conjuntos de objetos.

#### 2.3.3.2. Esquemas basados en redes

Los esquemas basados en redes utilizan estructuras de grafo formadas por nodos (conceptos) y arcos (relaciones), para representar el conocimiento.

Los ejemplos más conocidos de este tipo de esquemas son: las redes semánticas, los grafos conceptuales y las ontologías.

Las redes semánticas incluyen definiciones de conceptos y relaciones entre conceptos, facilitando, de este modo, las consultas y la inferencia de conocimiento.

Los grafos conceptuales, por su parte, representan el conocimiento mediante vértices, que pueden ser conceptos o relaciones conceptuales, de tal manera que los vértices de conceptos solo tienen arcos hacia vértices de relaciones conceptuales y viceversa. Así, cada grafo representa una única proposición, haciendo necesaria la utilización de muchos grafos para representar el conocimiento de un dominio.

Las ontologías son esquemas de representación de partes de una realidad compartida por un dominio que pueden ser interpretados por sistemas informáticos. Para ello, constan de los siguientes elementos: conceptos, que representan categorías; relaciones, que conectan conceptos; instancias, que representan objetos concretos y atributos, que representan las propiedades y valores asignados a conceptos o instancias.

El esquema de representación seleccionado para desarrollar la base de conocimiento de este proyecto es una ontología constituida por bases de datos no relacionales.

#### 2.3.3.3. Representación estructurada

Este tipo de representación es una ampliación de los esquemas basados en redes, ya que utiliza grafos en los que los vértices son estructuras complejas formadas por una colección de campos.

Dentro de este tipo de representación encontramos los guiones y los marcos.

Los guiones utilizan escenas, roles, objetos y lugares para describir situaciones típicas por medio de la representación secuencial de los eventos que suceden en ellas. De este modo, las escenas son los sucesos descritos, de manera que la realización de una escena permita el desarrollo de la siguiente; mientras que los roles son los personajes que intervienen en la escena.

Los marcos son estructuras de datos que permiten representar entidades estereotipadas, clases y atributos. Así, un marco contiene un identificador y campos con relaciones hacia otros marcos e información sobre el objeto; así como restricciones de integridad, que indican las condiciones necesarias para que un objeto encaje en el estereotipo definido por el marco.

## Capítulo 3

# METODOLOGÍA

### 3.1. HERRAMIENTAS SOFTWARE

Antes de comenzar a desarrollar el proyecto es necesario tomar una serie de decisiones:

En primer lugar, debemos identificar qué servidor de bases de datos es el más adecuado para lograr el objetivo propuesto, así como el sistema operativo que servirá de soporte para poder codificar la base de conocimiento.

En segundo lugar, es importante analizar y comparar los distintos lenguajes de programación que nos permitirían desarrollar el código de nuestra base de conocimiento en base al servidor de bases de datos seleccionado.

Por último, debemos seleccionar qué herramientas de programación vamos a utilizar en función del servidor de bases de datos, el lenguaje de programación y el sistema operativo seleccionados.

#### 3.1.1. Servidor de bases de datos

A la hora de seleccionar un servidor de bases de datos, lo primero que hay que tener en cuenta es el modelo que vamos a utilizar: relacional o no relacional.

Para realizar la comparación, en primera instancia vamos a exponer las características de cada modelo.

	Modelo de bases de datos relacional	Modelo de bases de datos no relacional
Lenguaje de programación	Lenguaje de consulta SQL	Depende de la base de datos
Estructura de los datos	Tablas con registros	No requiere estructuras fijas
Flexibilidad	Una vez definida la estructura de la base de datos, es complicado realizar modificaciones	Al permitir esquemas dinámicos, es muy sencillo realizar modificaciones en la base de datos
Espacio en memoria	Se desperdicia espacio en memoria al tener campos vacíos	Se desperdicia espacio en memoria al repetir información
Propiedades	Modelo ACID	Teorema CAP
Escalabilidad	Vertical	Horizontal

Tabla 3: Cuadro comparativo modelos de bases de datos relacional y no relacional.

Fuente: Elaboración propia

Una diferencia fundamental entre el modelo de bases de datos relacional y el no relacional es el lenguaje: las bases de datos relacionales utilizan un lenguaje estándar de consulta llamado

SQL, mientras que el modelo no relacional utiliza un lenguaje de programación u otro en función de la base de datos seleccionada y de las preferencias del programador.

Por otro lado, el uso del lenguaje SQL implica una preparación previa significativa, pues necesita predefinir esquemas muy claros para determinar la estructura de la información almacenada. Por el contrario, las bases de datos no relacionales permiten el uso de esquemas dinámicos que almacenan datos no estructurados, no siendo necesario llevar a cabo una planificación previa de la estructura de la base de datos.

Otra diferencia importante entre los modelos relacional y no relacional hace referencia a la forma de estructurar los datos almacenados: mientras que las bases de datos relacionales están basadas en tablas, las bases de datos NoSQL pueden utilizar pares clave-valor, documentos o grafos, entre otros.

Además, debido a su atomicidad, una base de datos relacional presenta una forma muy rígida de almacenar datos: si la información no encaja en las tablas, el programador debe redefinir el esquema de la base de datos. Esto da lugar a una escasa flexibilidad a la hora de realizar modificaciones.

Por el contrario, una base de datos no relacional presenta una gran flexibilidad, ya que cada documento puede tener una estructura propia y los cambios se ejecutan documento a documento.

Es decir, en una base de datos relacional al aumentar un campo a la tabla este campo es para todos los registros de la misma, mientras que en un no relacional existe libertad para agregar un campo a un objeto específico sin necesidad de modificar los demás.

A la hora de analizar las propiedades de ambos modelos, es necesario definir el término transacción. En informática, una transacción es una interacción con un conjunto de datos complejo. Se trata de varias instrucciones que se ejecutan en bloque, es decir, o se ejecutan todas con éxito o no se ejecuta ninguna.

Las bases de datos relacionales, al contrario que las no relacionales, permiten realizar transacciones al trabajar con el modelo ACID (Atomicity, Consistency, Isolation and Durability):

Atomicidad: garantía de que la transacción se va a ejecutar en su totalidad o no se va a ejecutar en absoluto.

Consistencia: garantía de que la transacción que se va a realizar no incumple las directrices de integridad ya planteadas de la base de datos.

Aislamiento: garantía de que las transacciones sean independientes, de manera que se puedan realizar varias operaciones a un mismo registro sin que se produzcan errores.

Durabilidad: garantía de que, una vez ejecutada una transacción, los cambios perdurarán en el tiempo.

Las bases de datos no relacionales se rigen por el Teorema de CAP (Consistency, Availability, Partition Tolerance), el cual asegura que únicamente se pueden garantizar simultáneamente dos de las siguientes propiedades: consistencia, disponibilidad y tolerancia al particionado.

**Consistencia:** garantiza que una lectura retornará la escritura más reciente. Es decir, la actualización o modificación que se realice debe reflejarse en todos los nodos de la base de datos.

**Disponibilidad:** una petición de lectura recibirá siempre una respuesta correcta en un periodo de tiempo razonable.

**Tolerancia al particionamiento:** el sistema debe funcionar incluso si alguna de sus partes no se encuentra accesible en ese momento.

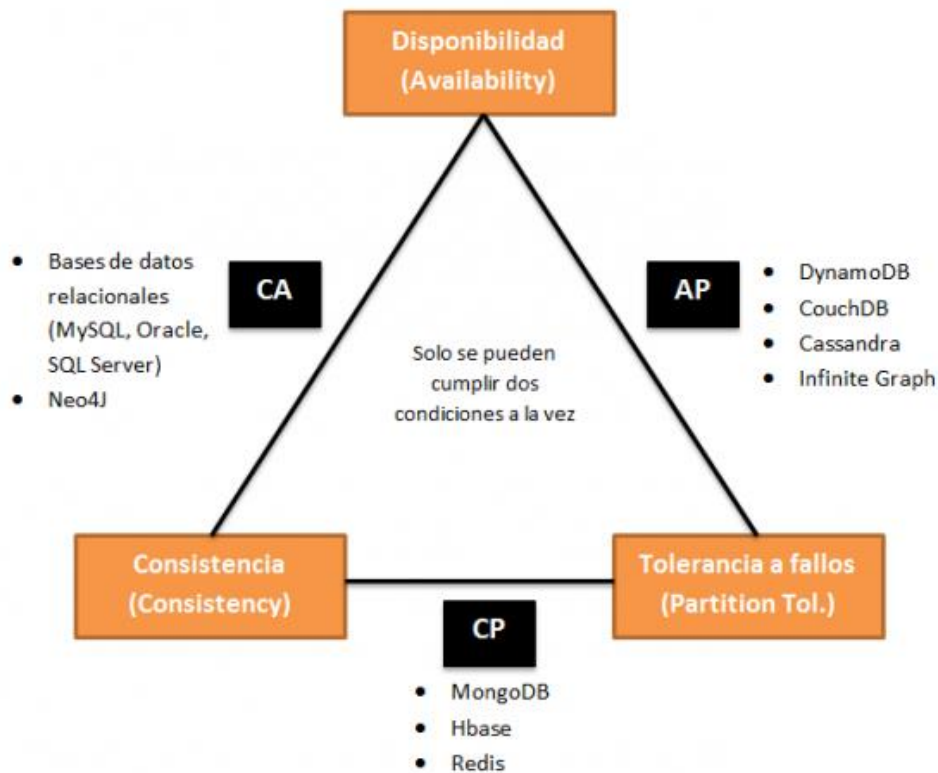


Figura 10: Clasificación de las bases de datos según el teorema CAP.

Fuente: Genbeta

Una última diferencia fundamental entre ambos modelos es su escalabilidad: las bases de datos relacionales son escalables verticalmente, mientras que las bases de datos NoSQL son escalables horizontalmente.

Con la escalabilidad vertical, el conjunto mejora al aumentar el hardware traspasando la información a un nuevo servidor más robusto, mientras que la escalabilidad horizontal implica aumentar el número de nodos para mejorar el conjunto de manera que si un nodo falla los otros seguirán trabajando.

Tras haber analizado las características principales de cada modelo, no se puede decir a ciencia cierta qué tipo de base de datos es mejor.

Las bases de datos relacionales aseguran la integridad de los datos, son muy eficaces cuando los datos almacenados están muy estructurados y utilizan un lenguaje de programación (SQL) ligero y declarativo, permitiendo llevar a cabo muchas tareas pesadas en la base de datos.

Las bases de datos no relacionales destacan por su escalabilidad, alto rendimiento y flexibilidad, resultando adecuadas cuando existe falta de uniformidad de los datos o no se dispone de un esquema predefinido.

De este modo, aunque las diferencias entre ambos modelos son minúsculas para programas de pequeña magnitud como el que se aborda en este proyecto, debido a la flexibilidad que presentan las bases de datos no relacionales y a la inexistencia de un esquema exacto de la información que se pretende almacenar en la base de conocimiento, se ha tomado la decisión utilizar un servidor de bases de datos no relacional.

En última instancia, es necesario analizar los diferentes tipos de bases de datos no relacionales.

Base de datos	Estructura	Aplicaciones	Ejemplos
clave-valor	Claves y valores indexados	Internet: carritos de compra en línea, sesiones de clientes	Dynamo DB (Amazon) Cassandra
documental	Documentos XML, JSON, BSON...	Sistemas de gestión de contenido, blogs	MongoDB Apache CouchDB
en grafo	Gráficos formados por nodos y bordes	Redes sociales, servicios de gestión en la nube	Neo4j OrientDB
columnar	Columnas de datos	Análisis de BigData, volumen de escritura pesado	Cassandra (Facebook) Big Table (Google)

Tabla 4: Cuadro comparativo tipos de bases de datos no relacionales.

Fuente: Elaboración propia

Con el objetivo de seleccionar qué tipo de base de datos no relacional es más adecuado para nuestro propósito, debemos tener en cuenta que no vamos a utilizar un volumen masivo de información (base de datos columnar) ni múltiples asociaciones de información (base de datos en grafo).

Así, entre las bases de datos clave-valor y las documentales, podemos observar que la estructura clave-valor carece de esquema, mientras que la estructura basada en documentos resulta muy intuitiva y fácil de escribir y leer para el desarrollador.

Por todo ello, el servidor finalmente seleccionado es MongoDB, una base de datos de uso gratuito que almacena información en documentos BSON, similares a JSON; permitiendo campos variados entre documentos y cambios en la estructura de datos.

### 3.1.2. Sistema operativo

Una vez seleccionado el servidor de bases de datos de MongoDB, comprobamos que cuenta con controladores destinados a programar en distintos lenguajes de programación utilizando cualquiera de los siguientes tres sistemas operativos: Windows, Linux y Mac Os.

El primer sistema operativo que descartamos es Mac Os, ya que el alumno no dispone de un ordenador de la gama Mac de Apple, necesario para utilizar este sistema operativo.

De este modo, basándonos en la comparativa entre Windows y Linux llevada a cabo por la empresa 1&1 IONOS [1], podemos ver que tanto Windows como Linux tienen ventajas e inconvenientes, no habiendo diferencias significativas a la hora de desarrollar un programa de la magnitud que requiere este TFG.

Así, debido a la familiaridad del alumno con Windows a la hora de programar y a la necesidad de conocimientos concretos y experiencia para poder utilizar y conocer a fondo Linux, el sistema operativo seleccionado para sacar adelante el proyecto es Windows.

### 3.1.3. Lenguaje de programación

Utilizando Windows como sistema operativo, el servidor de bases de datos de MongoDB dispone de controladores para lenguajes de programación de propósito general (C), lenguajes programación orientada a objetos (C++, Go, Java, PHP, Python) y lenguajes de programación multiparadigma (C#, Rust, Scala).

Aunque cualquiera de los lenguajes de programación mencionados anteriormente es adecuado para alcanzar los objetivos propuestos, puesto que el desarrollo de este Trabajo Fin de Grado supone la culminación de los estudios de Grado en Ingeniería en Tecnologías Industriales, se ha decidido aprovechar al máximo los conocimientos adquiridos durante el mismo, tales como la programación en C y en C++.

Así, el lenguaje de programación seleccionado es C++, por tratarse de un lenguaje de programación híbrido que nos permite programar en estilo procedimental (C), orientado a objetos y en ambos a la vez.

### 3.1.4. Entorno de desarrollo

El último elemento que necesitamos para completar los objetivos propuestos es un entorno de desarrollo que nos permita crear una interfaz gráfica para interactuar con la base de conocimiento.

Existen múltiples entornos de desarrollo que nos permiten trabajar en C++ utilizando Windows como sistema operativo, entre los que podemos destacar: Netbeans, Microsoft Visual Studio, Code::Blocks, CodeLite y Zinjal.

No obstante, el controlador de MongoDB para C++ destinado a Windows está disponible única y exclusivamente para Visual Studio.

De este modo, la base de conocimiento se codificará utilizando la última versión de Microsoft Visual Studio, la de 2019.



## 3.2. PREPARACIÓN DEL ENTORNO DE TRABAJO

### 3.2.1. Instalación de Microsoft Visual Studio 2019

Microsoft Visual Studio 2019 es el entorno de desarrollo seleccionado para codificar la base de conocimiento.

Su instalación se lleva a cabo desde la página oficial de Microsoft Visual Studio, seleccionando el apartado *Descargas* en el menú superior.

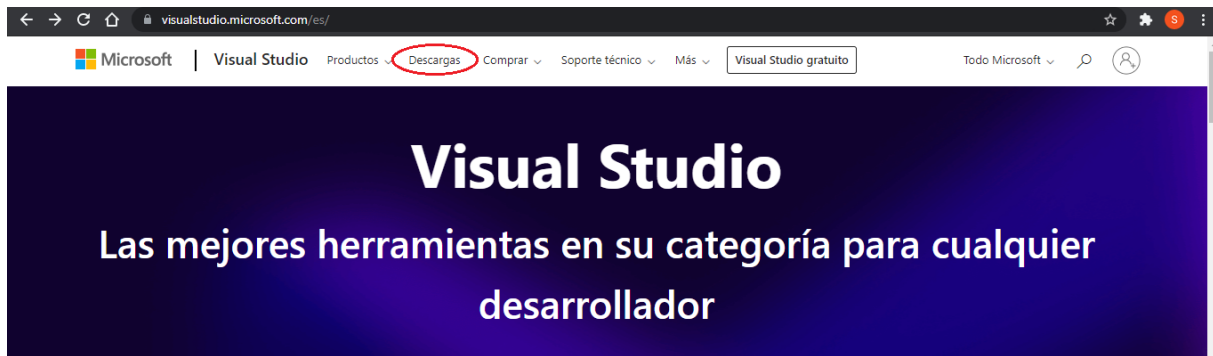


Figura 11: Instalación Microsoft Visual Studio 2019. Paso 1.

A continuación, se muestran los distintos paquetes correspondientes a la última versión disponible de Microsoft Visual Studio, en este caso, Visual Studio 2019. Para el alcance de este proyecto, basta con seleccionar la opción *Community* pulsando el botón *Descarga gratuita*.



Figura 12: Instalación Microsoft Visual Studio 2019. Paso 2.

En este momento comenzará la descarga del instalador de Microsoft Visual Studio 2019, un archivo de extensión .exe.

Una vez finalizada la descarga, al ejecutar dicho archivo se abre un cuadro de diálogo en el que seleccionaremos las cargas de trabajo que se desean instalar. En el caso concreto de este proyecto, las cargas de trabajo instaladas son aquellas necesarias para poder trabajar con el lenguaje de programación C++.

Un paso a tener en cuenta antes de comenzar la instalación de las cargas de trabajo es seleccionar la opción *Compatibilidad con C++/CLI* dentro de la carga de trabajo *Desarrollo*

para el escritorio con C++. Esto es necesario para poder instalar posteriormente la API que nos permitirá desarrollar una interfaz gráfica en C++ mediante Formularios de Windows.

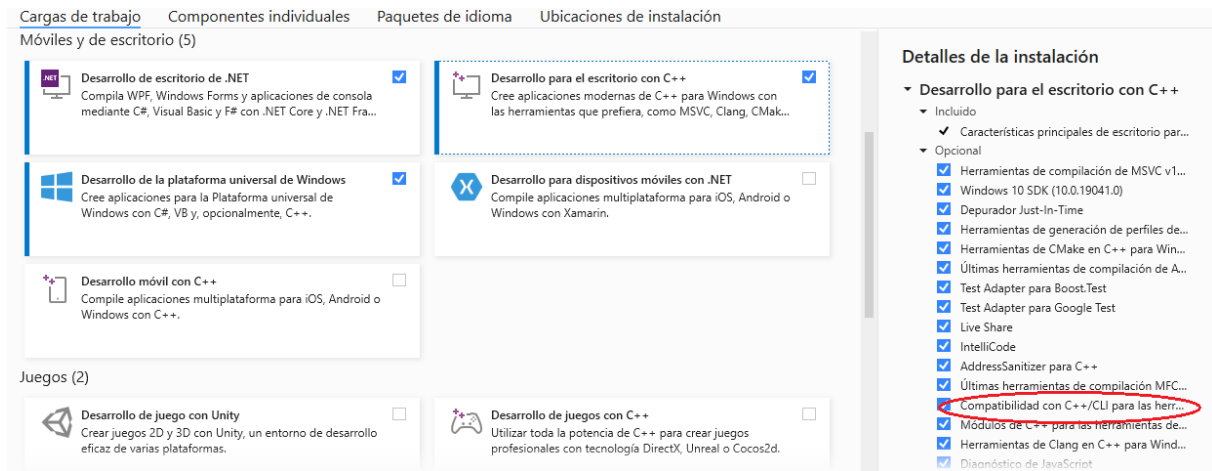


Figura 13: Instalación Microsoft Visual Studio 2019. Paso 3.

Finalmente, comienza la descarga e instalación de Microsoft Visual Studio, un proceso que tendrá una duración aproximada de una hora.

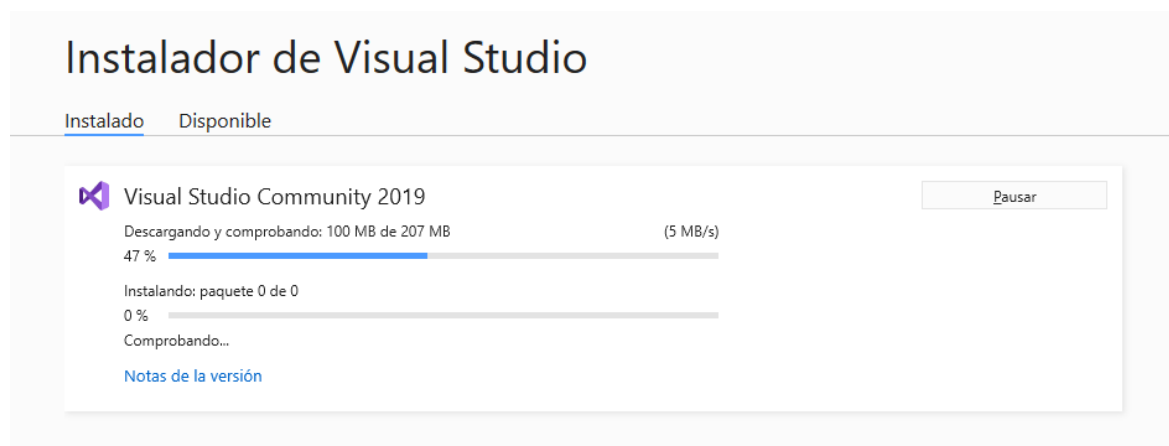


Figura 14: Instalación Microsoft Visual Studio 2019. Paso 4.

### 3.2.2. Instalación API Formularios de Windows para C++

Formularios de Windows para C++ es la API de Microsoft Visual Studio 2019 que nos permite crear una interfaz gráfica en lenguaje de programación C++.

Para su instalación, ejecutamos Microsoft Visual Studio 2019 y seleccionamos el apartado *Extensiones* en el menú superior y, dentro de este, la opción *Administrar extensiones*.

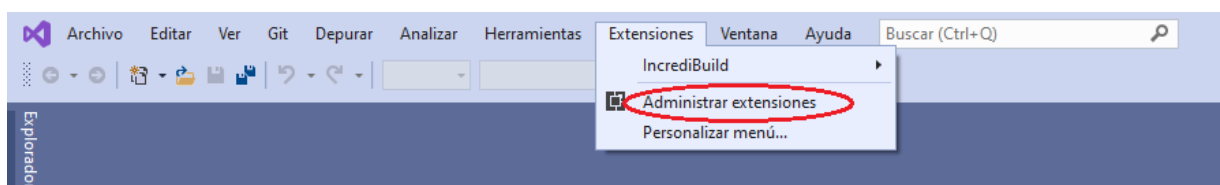


Figura 15: Instalación API Formularios de Windows para C++. Paso 1.

De este modo, el programa nos muestra un cuadro de diálogo en el que debemos introducir las palabras *C++ Windows Form* en el buscador. Entre los resultados obtenidos se debe seleccionar la API de nombre *C++/CLR Windows Forms for Visual Studio 2019*.

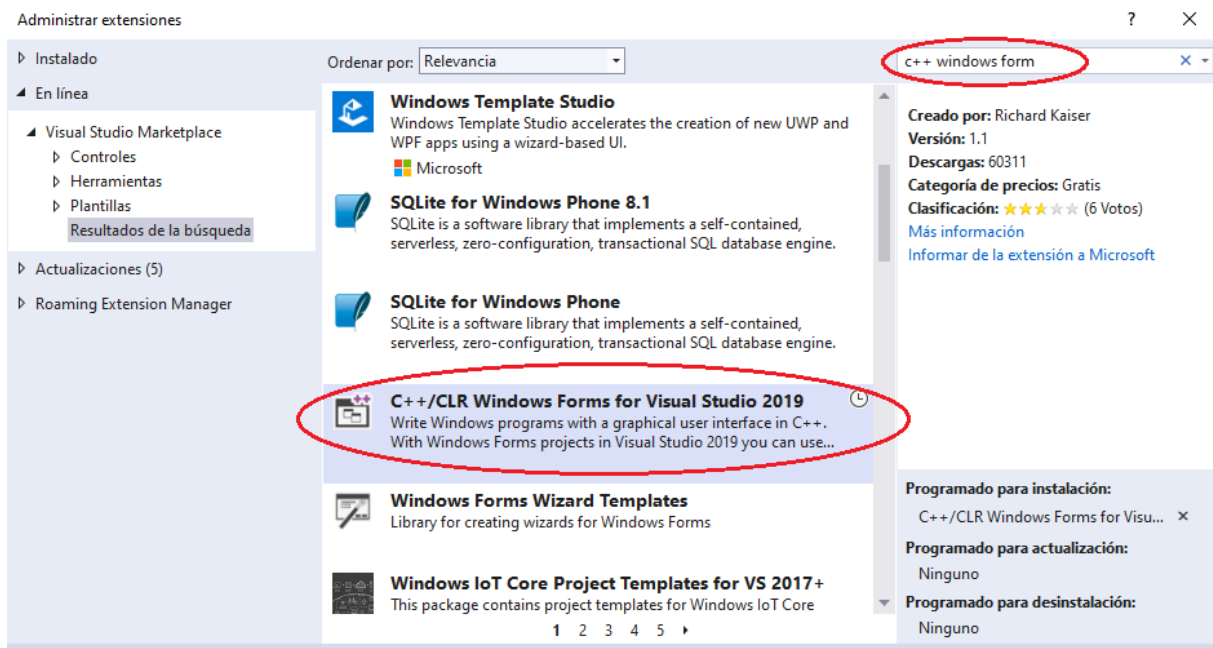


Figura 16: Instalación API Formularios de Windows para C++. Paso 2.

Para que comience la instalación es necesario cerrar la aplicación Microsoft Visual Studio 2019.

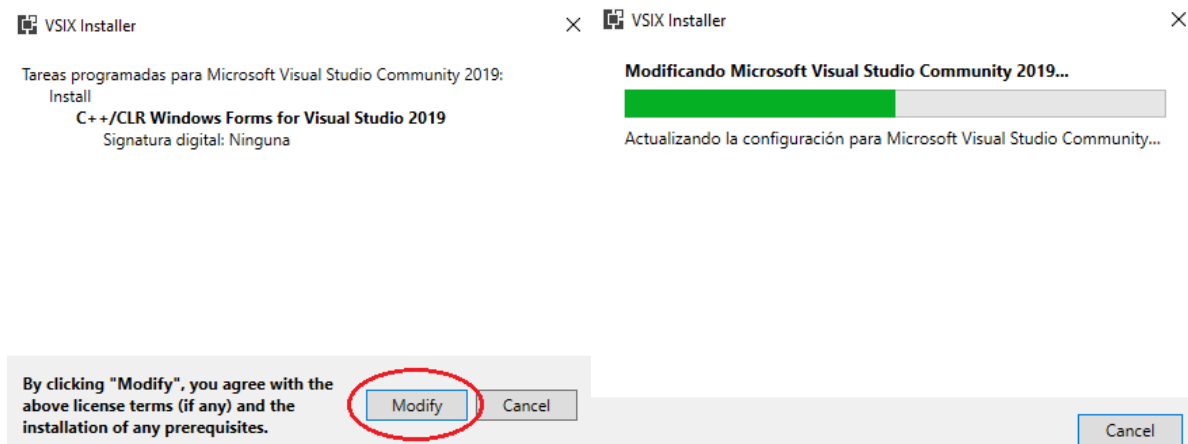


Figura 17: Instalación API Formularios de Windows para C++. Paso 3.

### 3.2.3. Instalación MongoDB

Mongo DB es el servidor de bases de datos no relacionales seleccionado para desarrollar la base de conocimiento de este proyecto.

Su instalación se lleva a cabo desde la página oficial de la empresa, seleccionando el apartado *Software* en el menú superior y, dentro de este, la opción *Community Server*.

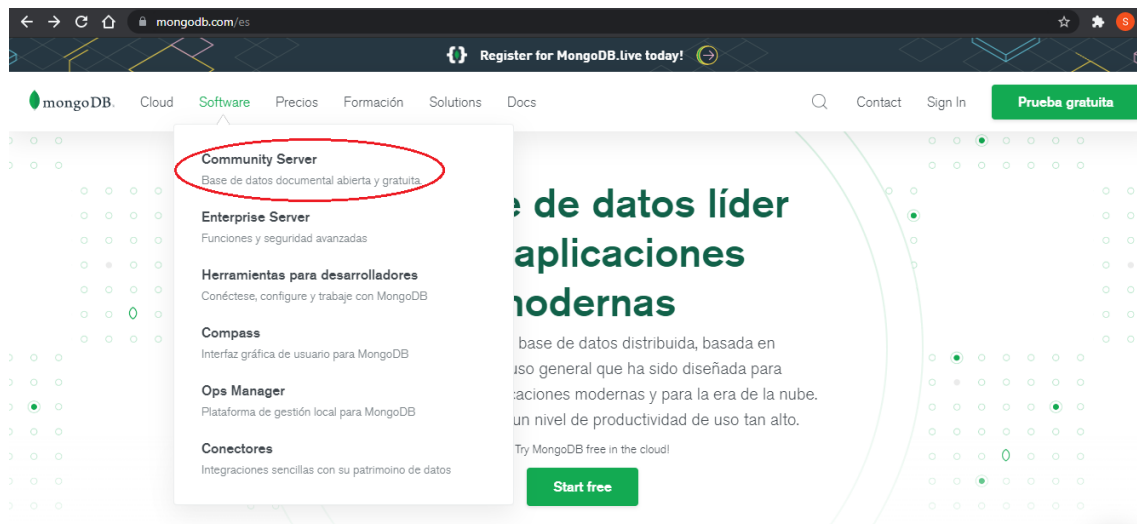


Figura 18: Instalación servidor MongoDB. Paso 1.

Esta selección redirige la página web al apartado descargas, donde se muestran las versiones del servidor disponibles para descargar.

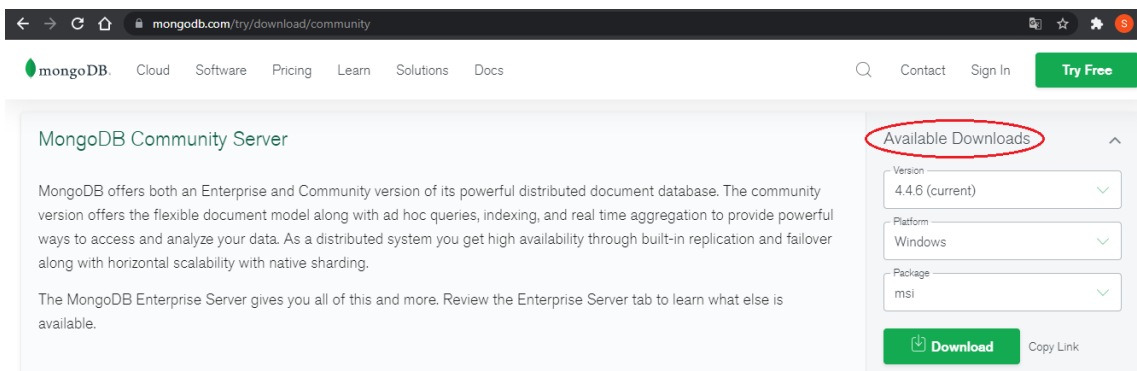


Figura 19: Instalación servidor MongoDB. Paso 2.

Al pulsar el botón *Download* comienza la descarga del instalador, un archivo de extensión .exe que ejecutaremos posteriormente.



Figura 20: Instalación servidor MongoDB. Paso 3.

En último lugar, siguiendo las instrucciones del instalador, llevamos a cabo la descarga e instalación del servidor MongoDB para bases de datos no relacionales.

### 3.2.4. Instalación del controlador de MongoDB para C++

El servidor de bases de datos no relacionales MongoDB dispone de un controlador con funciones codificadas en C++, las cuales permiten la interacción con la base de conocimiento.

La instalación de esta librería se lleva a cabo desde la página oficial de MongoDB seleccionando el apartado *Doc* en el menú superior y, dentro de este, la opción *Drivers*.

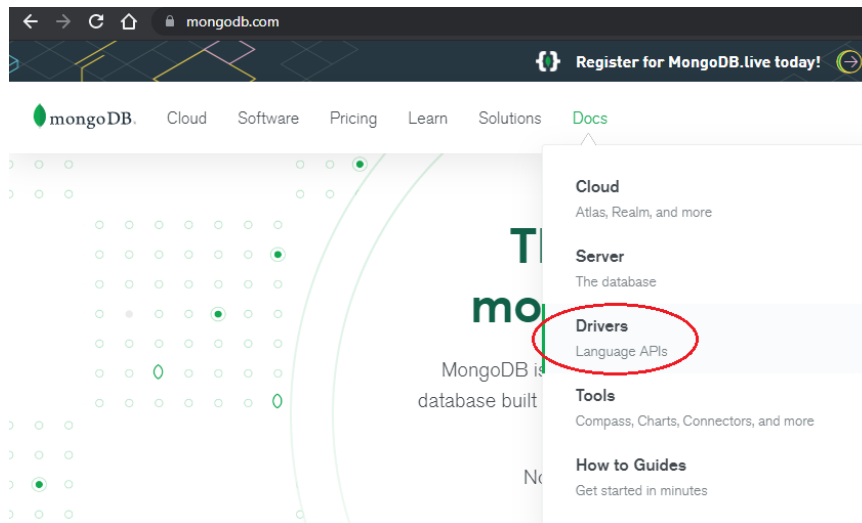


Figura 21: Instalación MongoDB Driver C++. Paso 1.

Esta selección nos muestra los controladores de MongoDB disponibles para distintos lenguajes de programación.

Se debe seleccionar el lenguaje de programación deseado, así como el sistema operativo en el que deseamos llevar a cabo la instalación. En el caso de este proyecto, seleccionaremos C++ como lenguaje de programación y Windows como sistema operativo.

## Start Developing with MongoDB

Connect your application to your database with one of our official libraries.

The following libraries are officially supported by MongoDB. They are actively maintained, support new MongoDB features, and receive bug fixes, performance enhancements, and security patches.

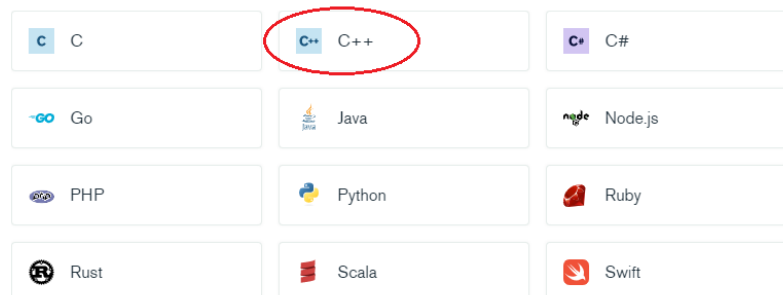


Figura 22: Instalación MongoDB Driver C++. Paso 2.

## Installing the mongocxx driver

- [Windows](#)
- [macOS](#)
- [Linux](#)
- [Advanced Configuration and Installation Options](#)

Figura 23: Instalación MongoDB Driver C++. Paso 3.

A continuación, para acceder al repositorio donde se encuentran los enlaces de descarga de las librerías de MongoDB debemos seleccionar la opción *C Driver releases*.

### Windows

Step 1: Install the latest version of the MongoDB C driver.

The mongocxx driver builds on top of the MongoDB C driver.

- For mongocxx-3.6.x, libmongoc 1.17.0 or later is required.
- For mongocxx-3.5.x, libmongoc 1.15.0 or later is required.
- For mongocxx-3.4.x, libmongoc 1.13.0 or later is required.
- For mongocxx-3.3.x, libmongoc 1.10.1 or later is required.
- For mongocxx-3.2.x, libmongoc 1.9.2 or later is required.
- For mongocxx-3.1.4+, libmongoc 1.7.0 or later is required.
- For mongocxx-3.1.[0-3], libmongoc 1.5.0 or later is required.
- For mongocxx-3.0.x, we recommend the last 1.4.x version of libmongoc

Unless you know that your package manager offers a [high enough version](#), you will need to download and build from the source code. Get a tarball from the [C Driver releases](#) page.

Figura 24: Instalación MongoDB Driver C++. Paso 4.

Una vez en el repositorio de github, descargamos la última versión del controlador deseado.

Finalmente, utilizaremos la herramienta CMake a través del Símbolo del Sistema (CMD) para llevar a cabo la instalación.

Las instrucciones necesarias para la correcta instalación de la librería mongoc.h son las siguientes:

```
C:\>cmake -G "Visual Studio 16 2019" -DCMAKE_INSTALL_PREFIX=C:\mongo-c-driver -DCMAKE_PREFIX_PATH=C:\mongo-c-driver
```

Figura 25: Instalación MongoDB Driver C++. Instrucción 1.

```
C:\>cmake --build . --config RelWithDebInfo
```

Figura 26: Instalación MongoDB Driver C++. Instrucción 2.

```
C:\>cmake --build . --config RelWithDebInfo --target install
```

Figura 27: Instalación MongoDB Driver C++. Instrucción 3.

### 3.2.5. Preparación Microsoft Visual Studio 2019 para MongoDB

Con el objetivo de utilizar la librería `mongoc.h` en nuestro proyecto de Microsoft Visual Studio 2019, llevamos a cabo algunas modificaciones en las propiedades del mismo.

Para ello, dentro del *Explorador de soluciones* de Visual Studio, hacemos click con el botón derecho del ratón en nuestro proyecto, un archivo de extensión `.sln`, y posteriormente, seleccionamos la opción propiedades.

En primer lugar, debemos añadir al apartado *Directorios de inclusión adicionales* la ruta donde se encuentran los archivos de extensión `.h` de la librería.

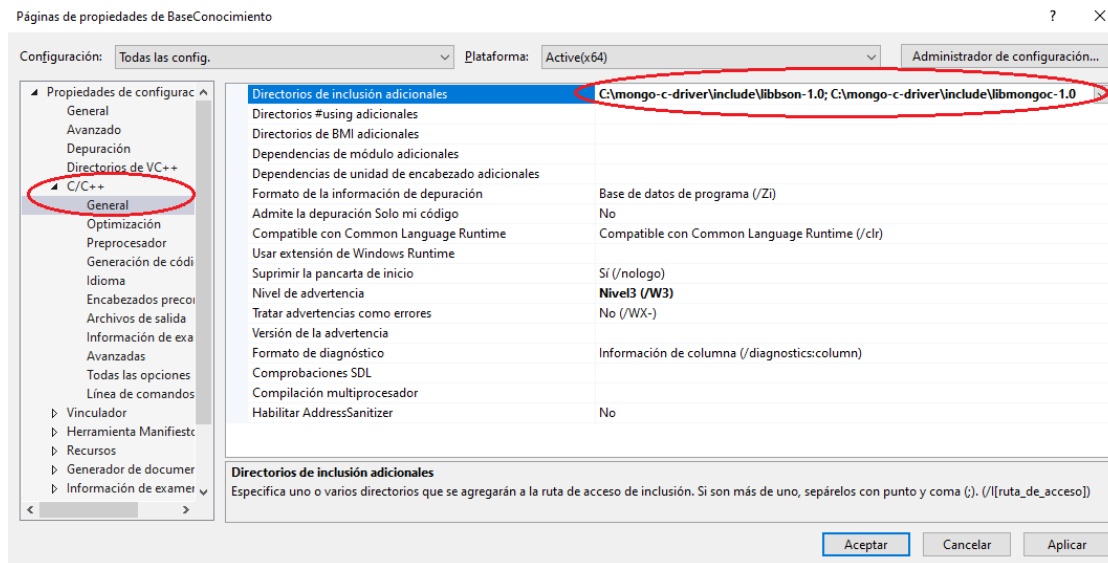


Figura 28: Preparación Visual Studio 2019. Paso 1.

A continuación, añadimos al apartado *Dependencias adicionales* los archivos de extensión `.lib` de la librería.

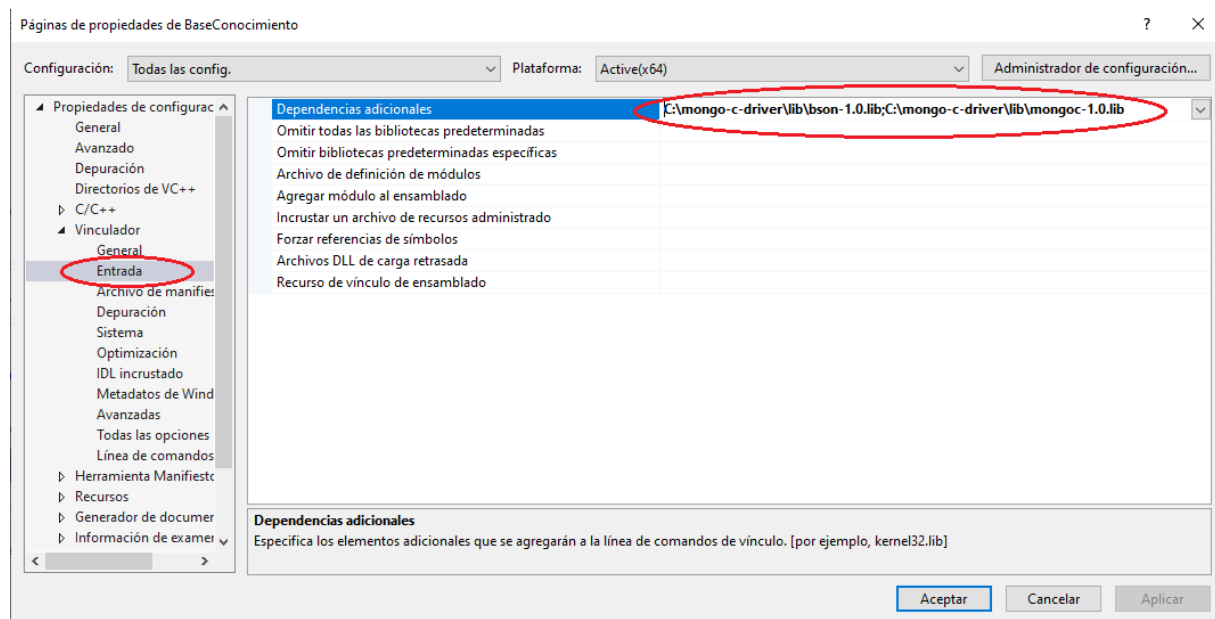


Figura 29: Preparación Visual Studio 2019. Paso 2.

Finalmente, añadimos al apartado *Entorno* la ruta donde se encuentran los archivos de extensión .dll de la librería.

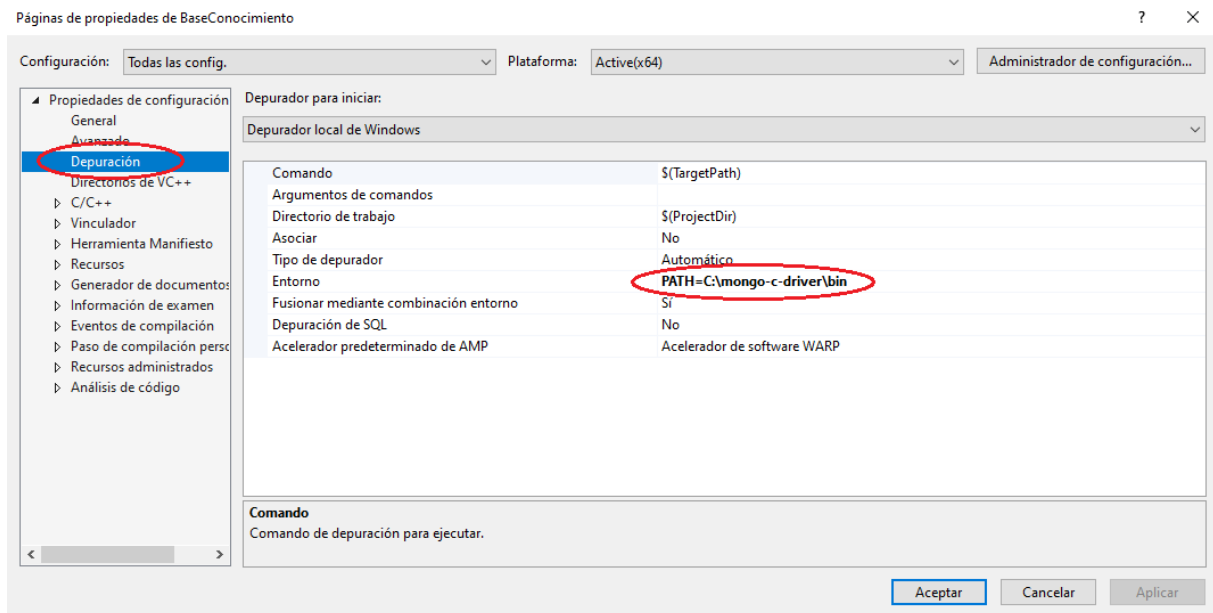


Figura 30: Preparación Visual Studio 2019. Paso 3.

Para poder desarrollar la base de conocimiento utilizando la librería `mongoc.h` solo hay que incluir la sentencia `#include <mongoc/mongoc.h>` en nuestro proyecto.



## Capítulo 4

# RESULTADOS

Como resultado de la realización de este proyecto, se ha obtenido un sistema de conocimiento con la siguiente estructura:

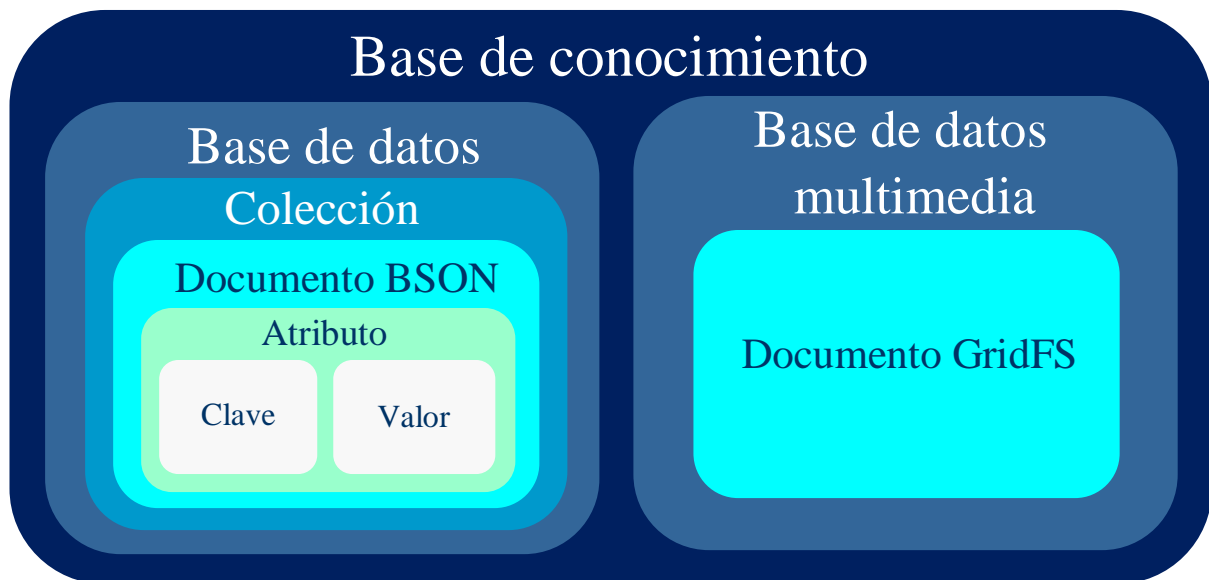


Figura 31: Estructura sistema de conocimiento.

Como podemos ver en la imagen, el sistema nos permite almacenar documentos de tipo BSON y de tipo GridFS.

Los documentos GridFS representan archivos multimedia, pudiendo ser archivos de imagen, vídeo, audio y texto; motivo por el cual se han creado cuatro bases de datos (una por cada tipo de archivo) destinadas a almacenarlos.

Así, los documentos BSON se incorporarán a la base de datos y colección seleccionadas por el usuario, mientras que los archivos multimedia se añadirán a la base de datos correspondiente en función del tipo de archivo seleccionado.

Una vez explicada la estructura del sistema, pasamos a analizar los resultados obtenidos tras la realización del proyecto, tanto desde el punto de vista funcional (base de conocimiento) como de diseño (interfaz gráfica).

## 4.1. BASE DE CONOCIMIENTO

En cuanto a funcionalidad se refiere, el desarrollo de este Trabajo Fin de Grado ha obtenido como resultado una serie de funciones, codificadas en lenguaje de programación C++.

En primer lugar, se ha desarrollado un conjunto de funciones relacionadas con MongoDB, destinadas a simplificar las funciones más complejas, facilitando, de este modo, el trabajo.

Dentro de este grupo, podemos destacar las siguientes funciones:

vector <string> ListaBasesDeDatosDisponibles( )

Parámetros	
Funcionalidad	Devuelve un vector con los nombres de todas las bases de datos pertenecientes a la base de conocimiento.

Tabla 5: Función ListaBasesDeDatosDisponibles( ). Parámetros y funcionalidad.

vector <string> ListaColeccionesDisponibles( )

Parámetros	string Base
Funcionalidad	Devuelve un vector con los nombres de todas las colecciones pertenecientes a la base de datos de nombre <i>Base</i> .

Tabla 6: Función ListaColeccionesDisponibles( ). Parámetros y funcionalidad.

vector <string> ListaDocumentosDisponibles( )

Parámetros	string Base, string Coleccion
Funcionalidad	Devuelve un vector con los nombres de todos los documentos pertenecientes a la colección de nombre <i>Coleccion</i> que se encuentra en la base de datos de nombre <i>Base</i> .

Tabla 7: Función ListaDocumentosDisponibles( ). Parámetros y funcionalidad.

vector <string> ListaArchivosMultimediaDisponibles( )

Parámetros	string Tipo (puede tomar los valores imagen, video, audio o texto)
Funcionalidad	Devuelve un vector con los nombres de todos los archivos multimedia de tipo <i>Tipo</i> pertenecientes a la base de conocimiento.

Tabla 8: Función ListaArchivosMultimediaDisponibles( ). Parámetros y funcionalidad.

vector <string> ListaAtributosDisponibles( )

Parámetros	string Base, string Coleccion, string Documento
Funcionalidad	Devuelve un vector con los nombres de todos los atributos presentes en el documento de nombre <i>Documento</i> perteneciente a la colección de nombre <i>Coleccion</i> que se encuentra en la base de datos de nombre <i>Base</i> .

Tabla 9: Función ListaAtributosDisponibles( ). Parámetros y funcionalidad.

En segundo lugar, se ha desarrollado un conjunto de funciones más complejas que podemos dividir en cuatro grandes grupos atendiendo al nivel en el que trabajan: funciones que trabajan con bases de datos, funciones que trabajan con colecciones, funciones que trabajan con documentos y, finalmente, funciones que trabajan con archivos multimedia.

#### 4.1.1. Bases de datos

En este apartado se describen las funciones desarrolladas mediante diagramas de flujo, según la operación que permiten realizar utilizando bases de datos.

##### 4.1.1.1. Añadir una base de datos a la base de conocimiento

Para que MongoDB reconozca una base de datos, esta debe contener al menos una colección. De este modo, para añadir una base de datos a la base de conocimiento, necesitaremos una base de datos y una colección.

`void CrearBaseDeDatos( )`

Parámetros	string Base, string Coleccion
------------	-------------------------------

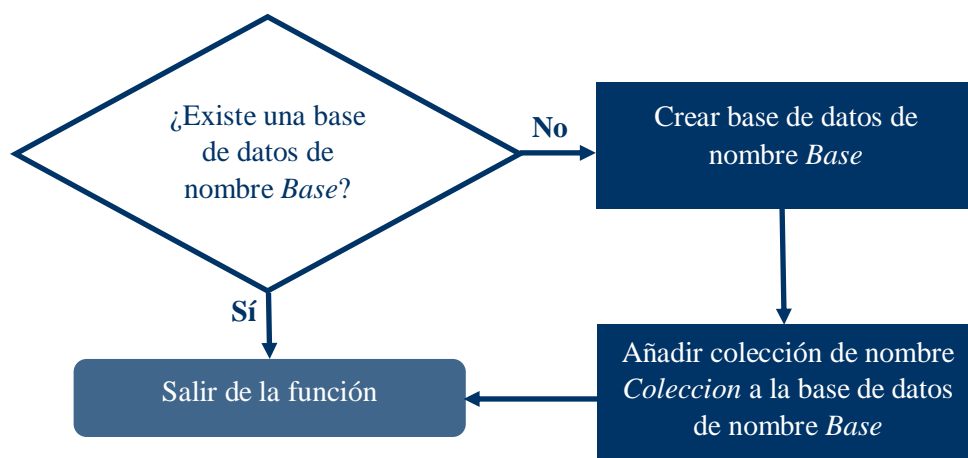


Figura 32: Función `CrearBaseDeDatos()`. Diagrama de flujo.

##### 4.1.1.2. Eliminar una base de datos de la base de conocimiento

`void EliminarBaseDeDatosExistente( )`

Parámetros	string Base
------------	-------------

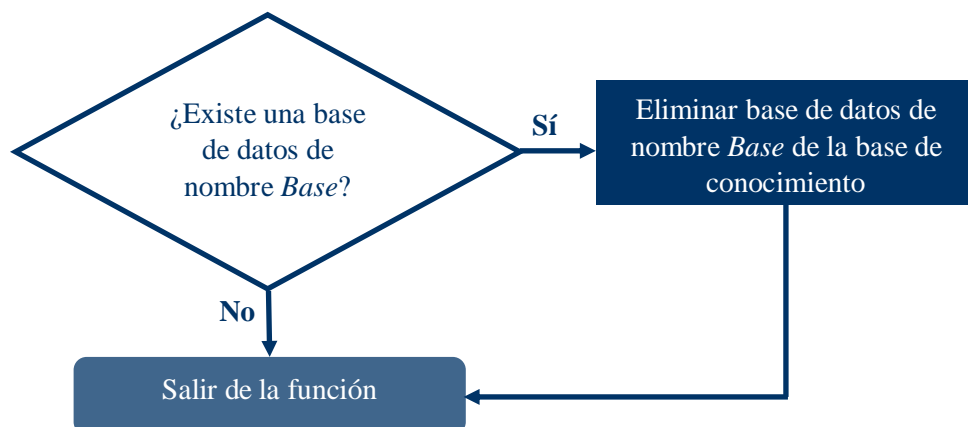


Figura 33: Función `EliminarBaseDeDatosExistente()`. Diagrama de flujo.

### 4.1.2. Colecciones

En este apartado se describen las funciones desarrolladas mediante diagramas de flujo, según la operación que permiten realizar utilizando colecciones.

#### 4.1.2.1. Añadir una colección a una base de datos

void CrearColeccion( )

Parámetros	string Base, string Coleccion
------------	-------------------------------

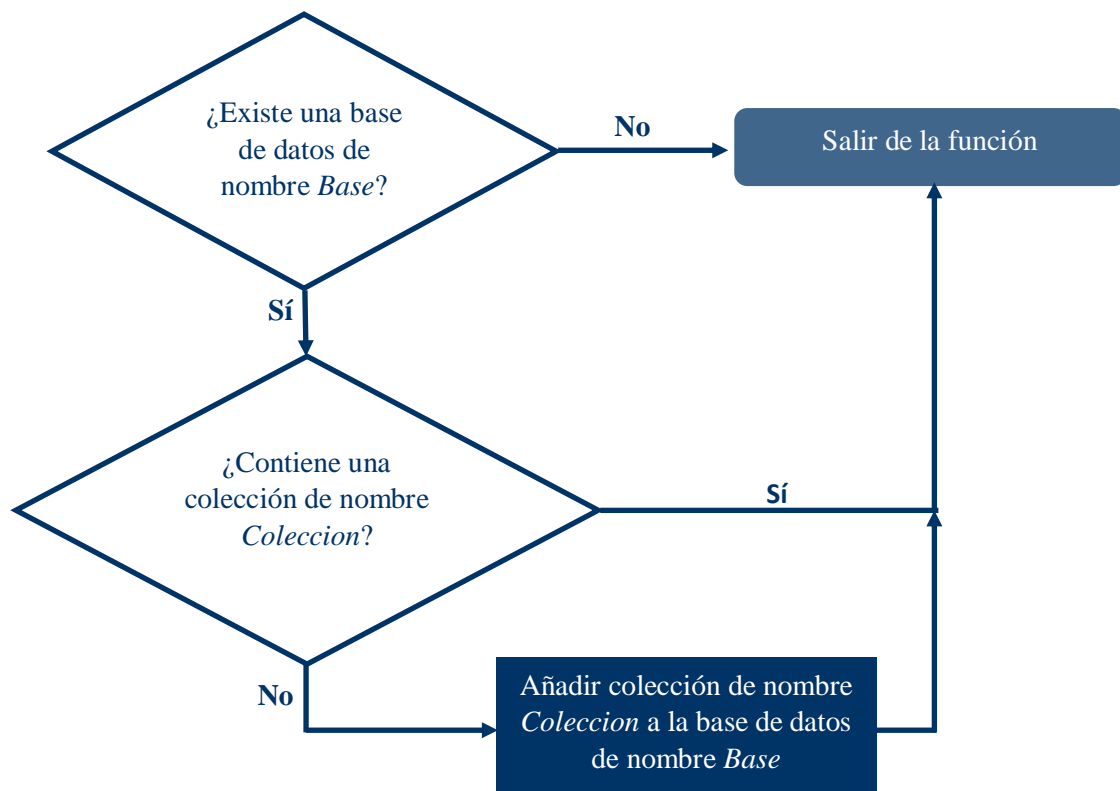


Figura 34: Función `CrearColeccion()`. Diagrama de flujo.

## 4.1.2.2. Eliminar una colección de una base de datos

```
void EliminarColeccionExistente( )
```

Parámetros	string Base, string Coleccion
------------	-------------------------------

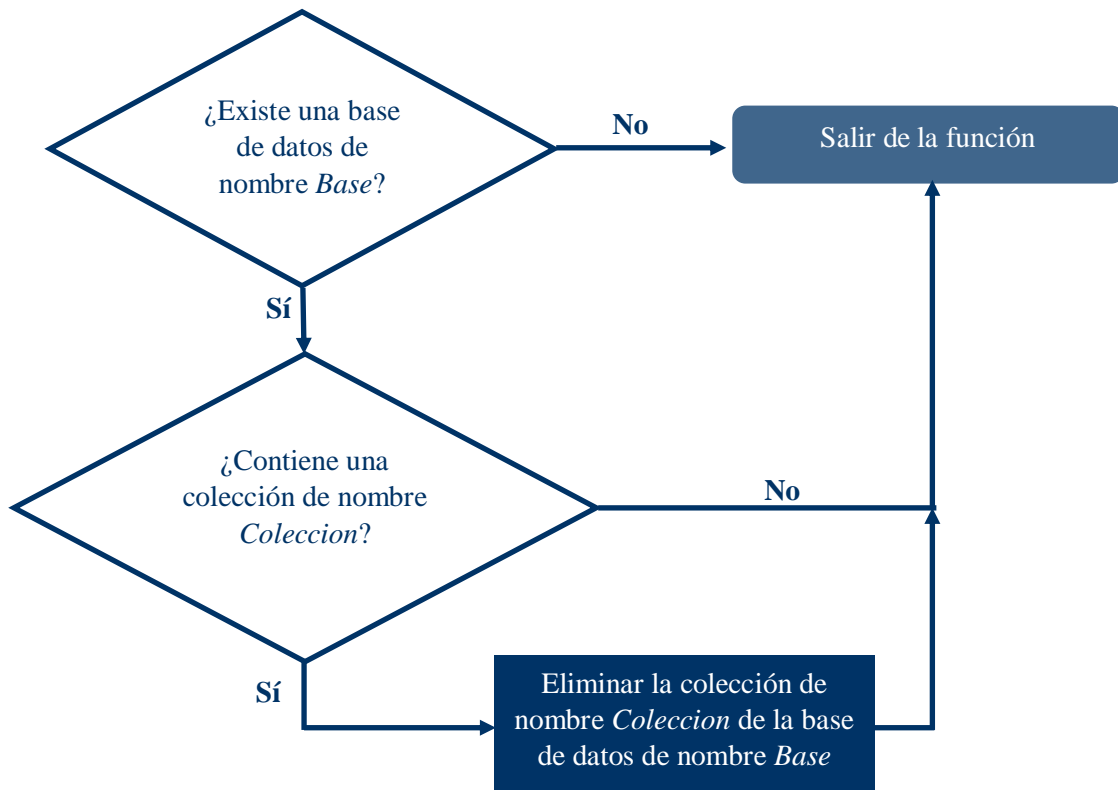


Figura 35: Función `EliminarColeccionExistente()`. Diagrama de flujo.

### 4.1.3. Documentos

En este apartado se describen las funciones desarrolladas mediante diagramas de flujo, según la operación que permiten realizar utilizando documentos BSON.

#### 4.1.3.1. Añadir un documento a una colección

void CrearDocumento( )

Parámetros	string Base, string Coleccion, string Documento, vector <string> atributos, string confianza
------------	--

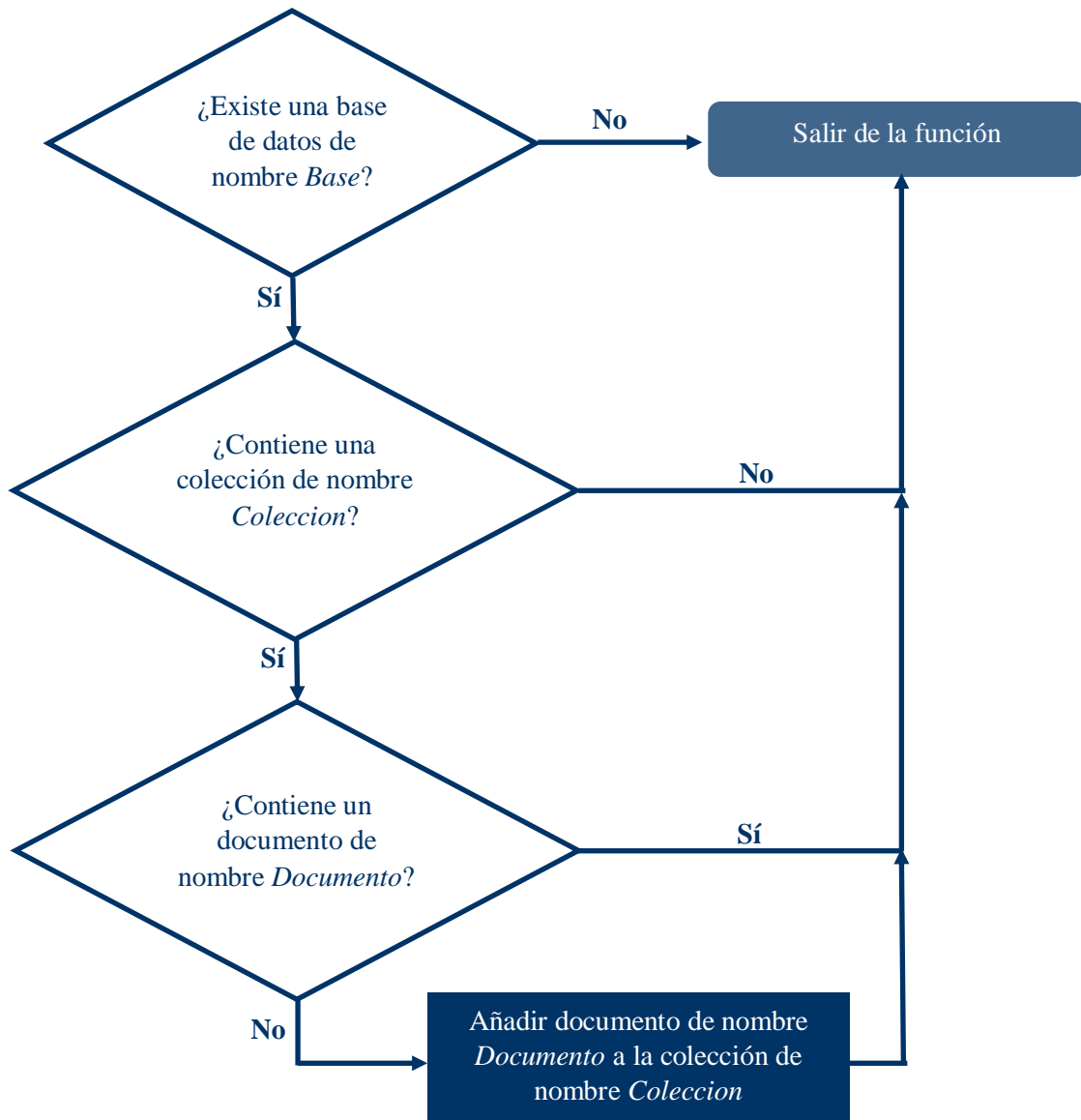


Figura 36: Función CrearDocumento( ). Diagrama de flujo.

## 4.1.3.2. Eliminar un documento de una colección

```
void EliminarTema( )
```

Parámetros	string Base, string Colección, string Documento
------------	---

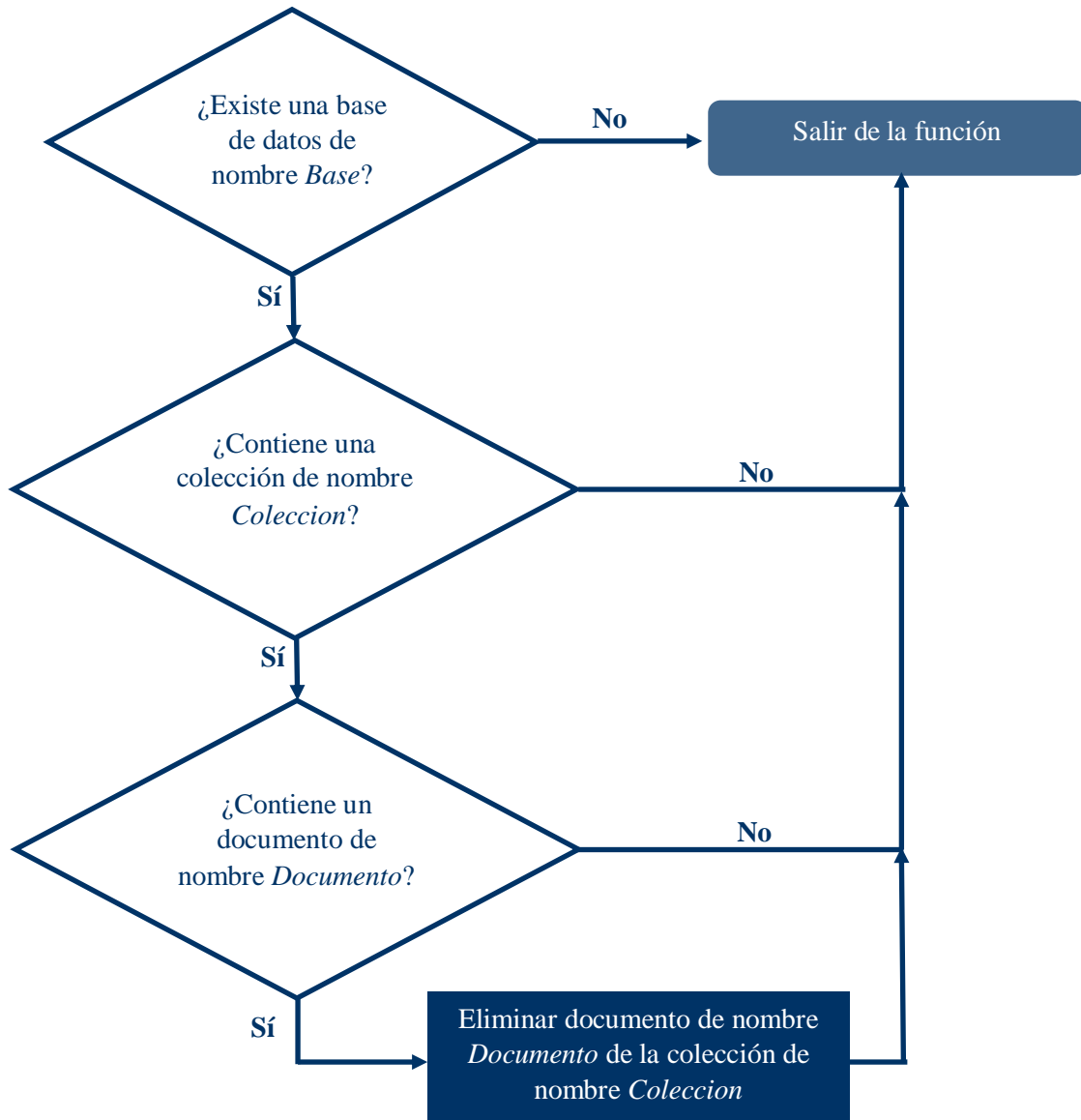


Figura 37: Función EliminarTema( ). Diagrama de flujo.

## 4.1.3.3. Consultar un documento

A la hora de consultar el contenido de uno o varios documentos, tenemos las siguientes opciones: consultar el contenido de todos los documentos pertenecientes a una colección, consultar el contenido de un documento introduciendo la cadena alfanumérica (ID) que lo identifica, consultar el contenido de un documento introduciendo su nombre y consultar el contenido de todos los documentos que contengan ciertos atributos.

vector <string> BuscarDocumentosColeccion( )

Parámetros	string Base, string Coleccion
------------	-------------------------------

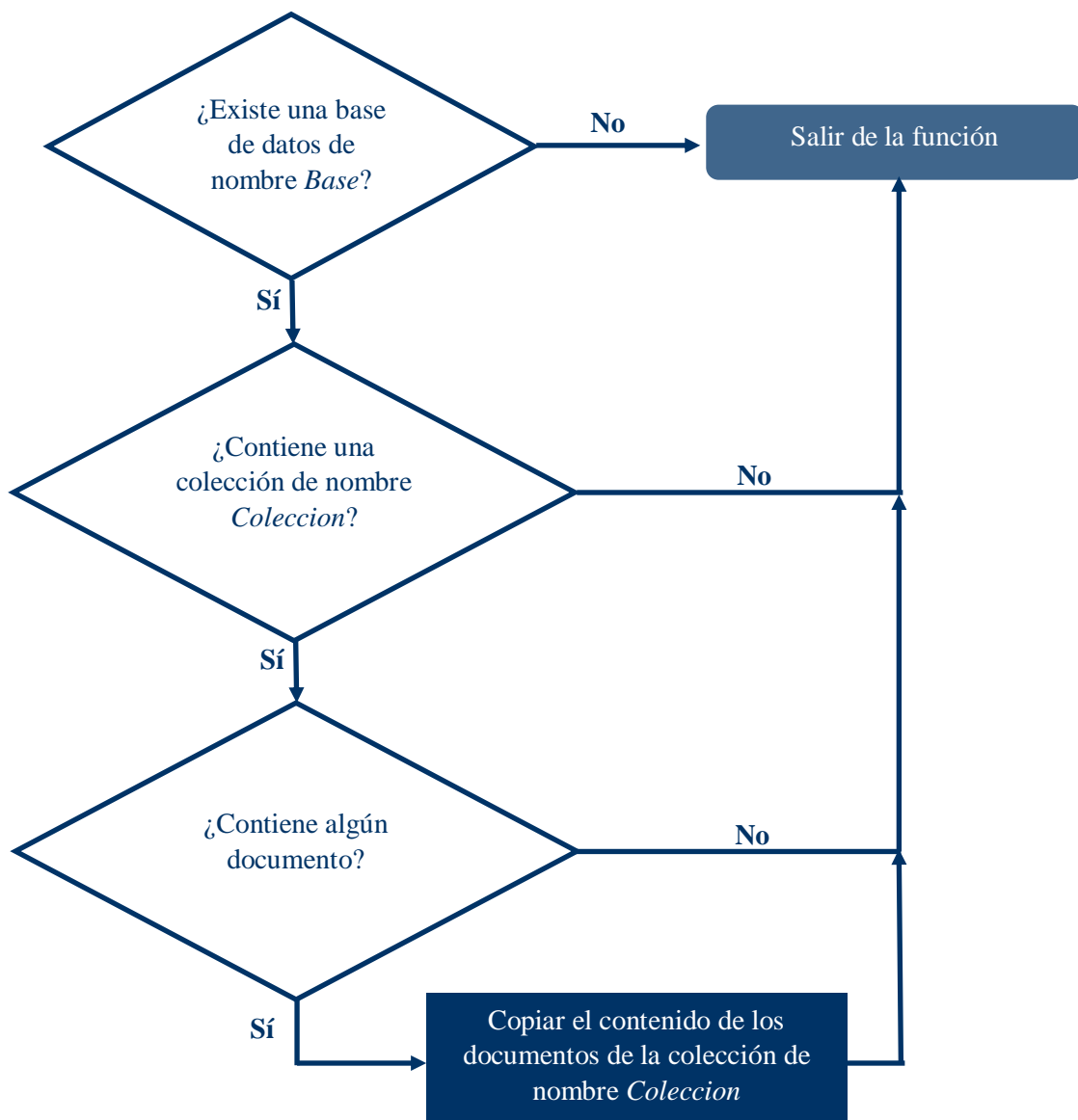


Figura 38: Función BuscarDocumentosColeccion( ). Diagrama de flujo.



vector <string> BuscarDocumentoPorID( )

Parámetros	string ID
------------	-----------

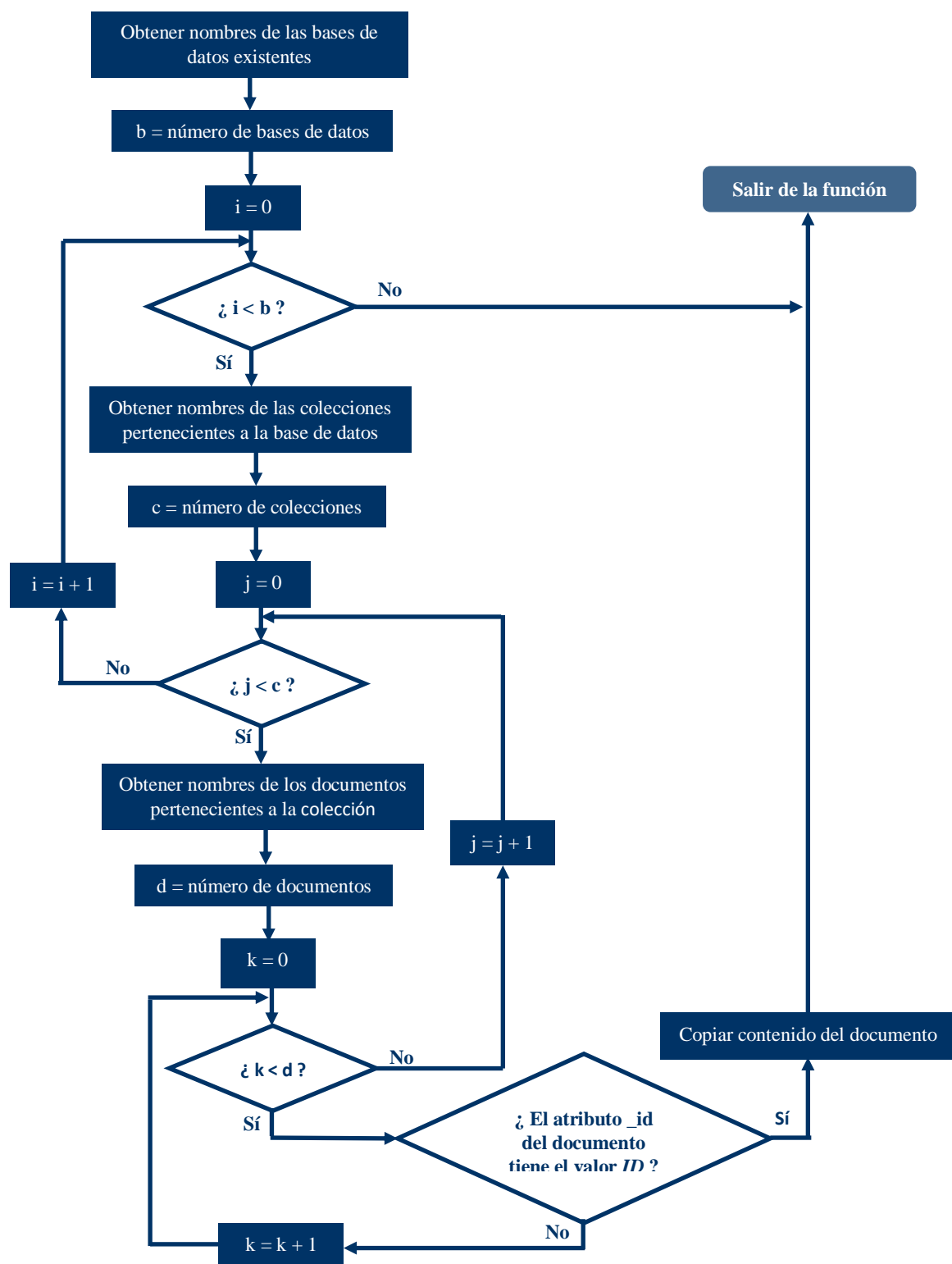


Figura 39: Función BuscarDocumentoPorID( ). Diagrama de flujo.

vector <string> BuscarDocumento( )

Parámetros string Base, string Coleccion, string Documento

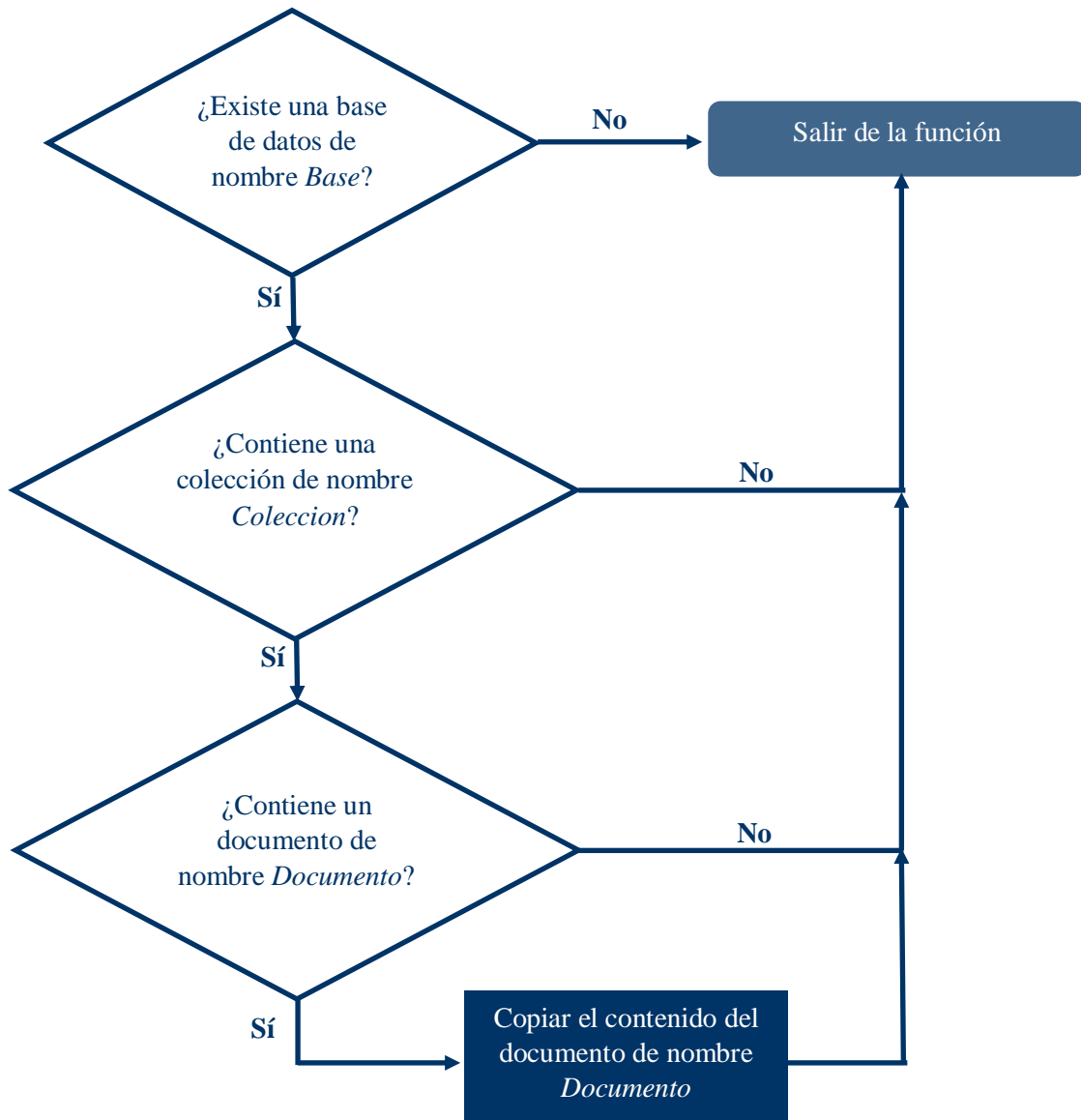


Figura 40: Función BuscarDocumento( ). Diagrama de flujo

Al utilizar esta función, también tenemos la opción de dejar en blanco los atributos *Base* y *Colección*, en cuyo caso el programa ejecutará un barrido por todas las bases de datos y colecciones existentes en la base de conocimiento hasta encontrar todos los documentos de nombre *Documento*.

vector <string> BuscarDocumentoPorAtributos( )

Parámetros vector <string> Atributos

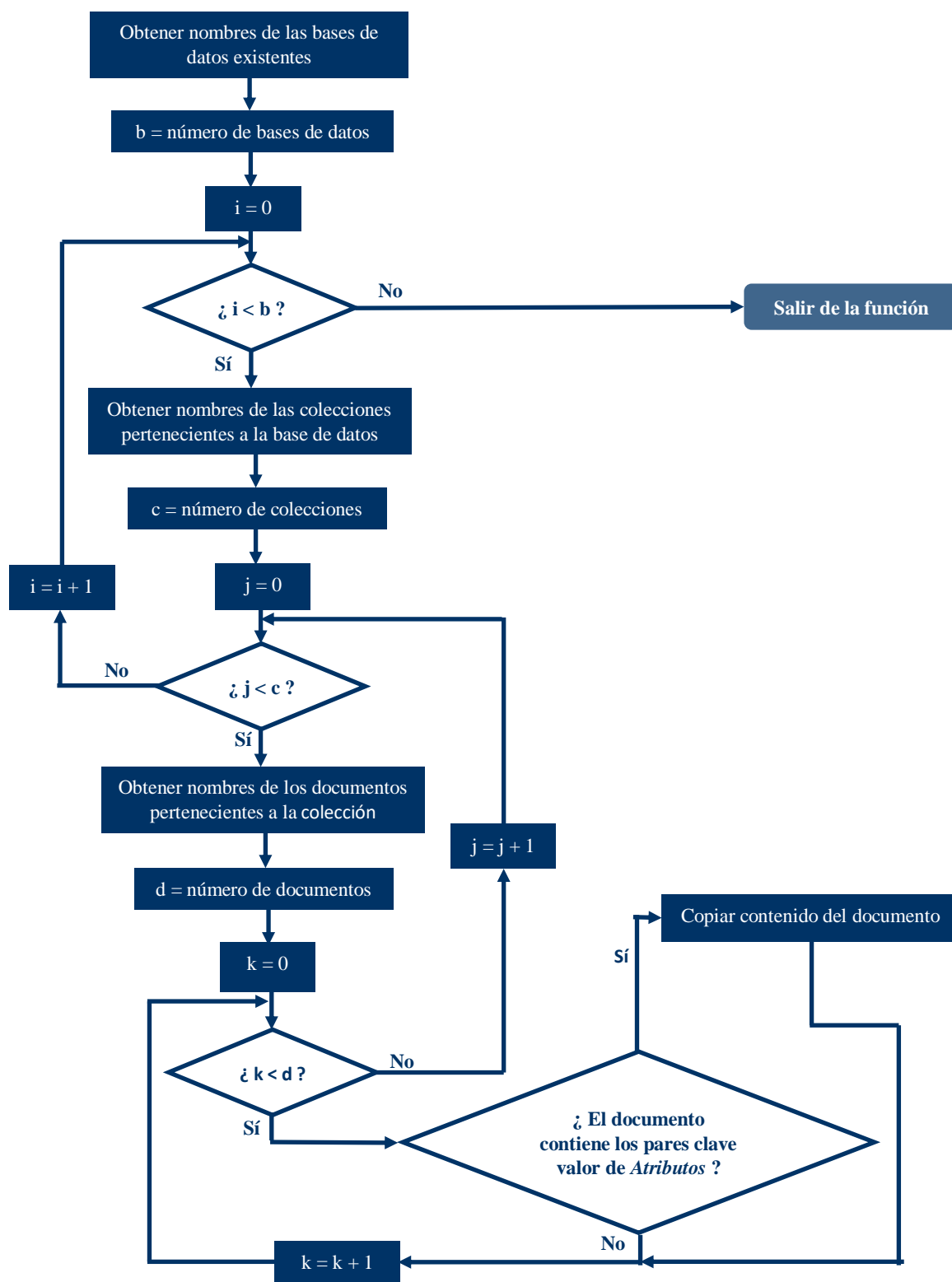


Figura 41: Función BuscarDocumentoPorAtributos( ). Diagrama de flujo.

## 4.1.3.4. Modificar un documento

En cuanto a la modificación de documentos, el sistema permite llevar a cabo las siguientes operaciones: añadir un atributo a un documento, modificar el valor de un atributo de un documento y eliminar un atributo de un documento.

bool AñadirAtributoADocumento( )

Parámetros	string Base, string Coleccion, string Documento, string Clave, string Valor, string confianza
------------	---

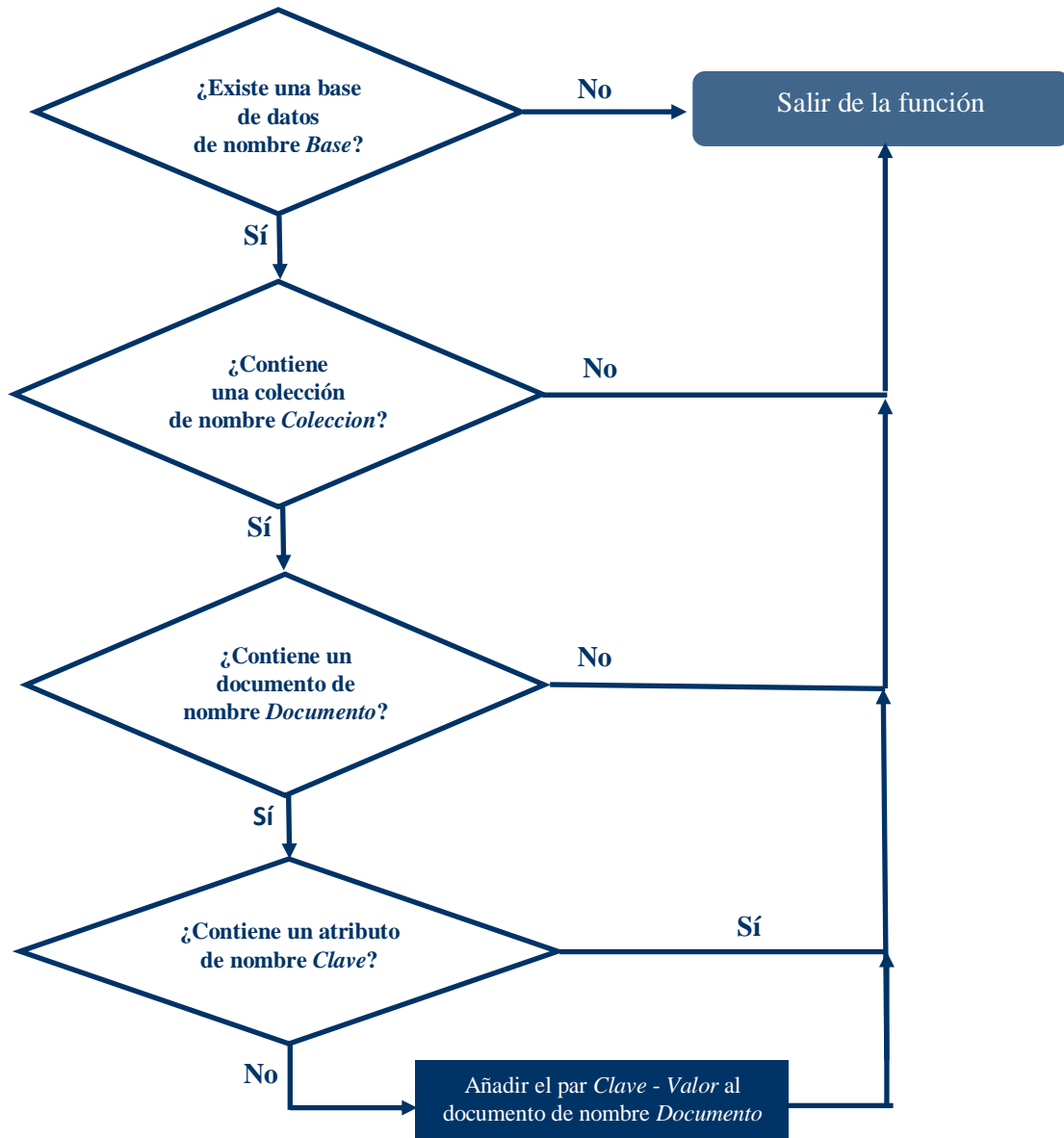


Figura 42: Función AñadirAtributoADocumento( ). Diagrama de flujo.

void ModificarAtributoDocumento()

Parámetros	string Base, string Coleccion, string Documento, string Clave, string Valor, string confianza
------------	---

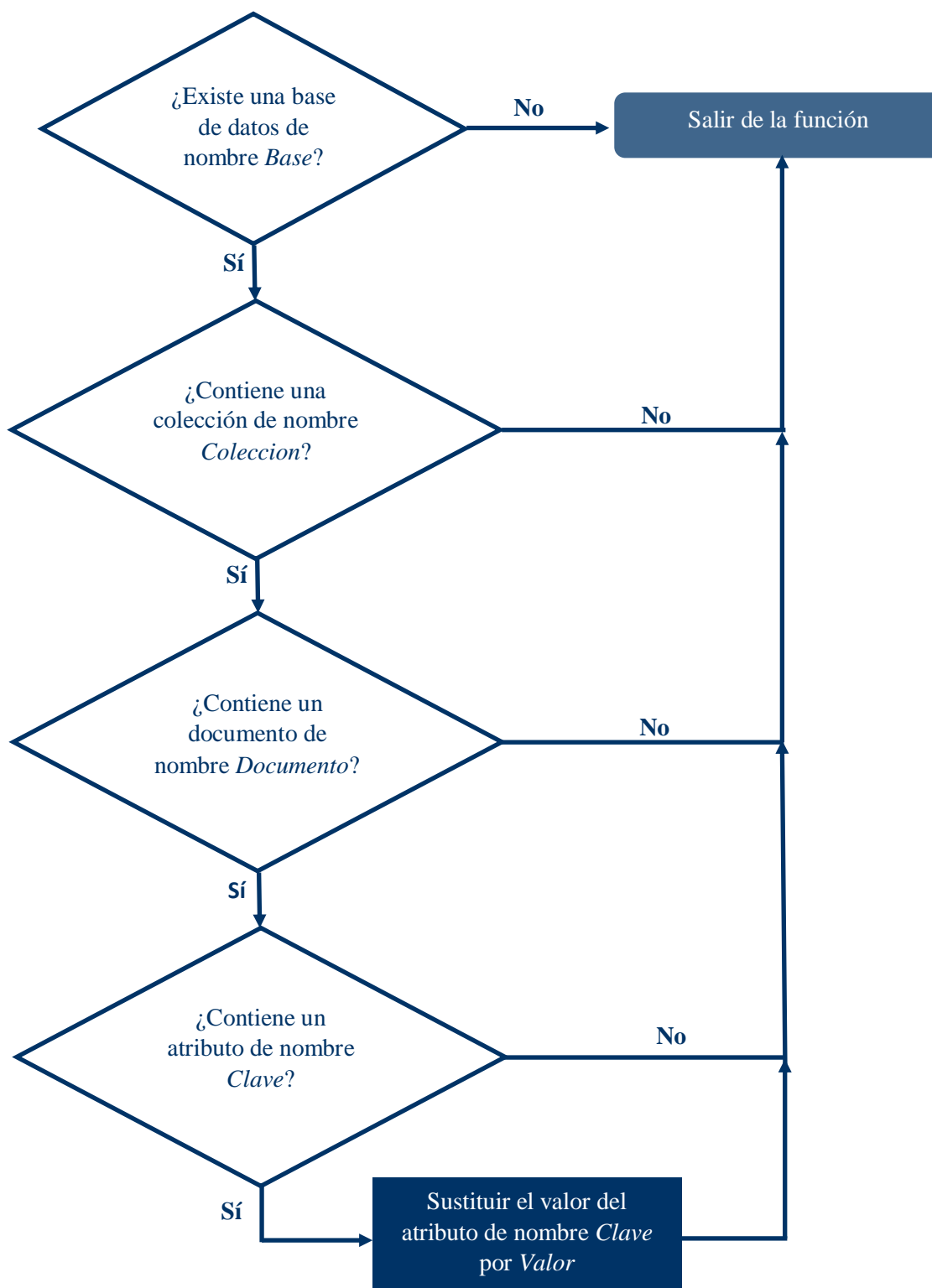


Figura 43: Función ModificarAtributoDocumento(). Diagrama de flujo.

void EliminarAtributoDocumento()

Parámetros	string Base, string Coleccion, string Documento, string Clave
------------	---

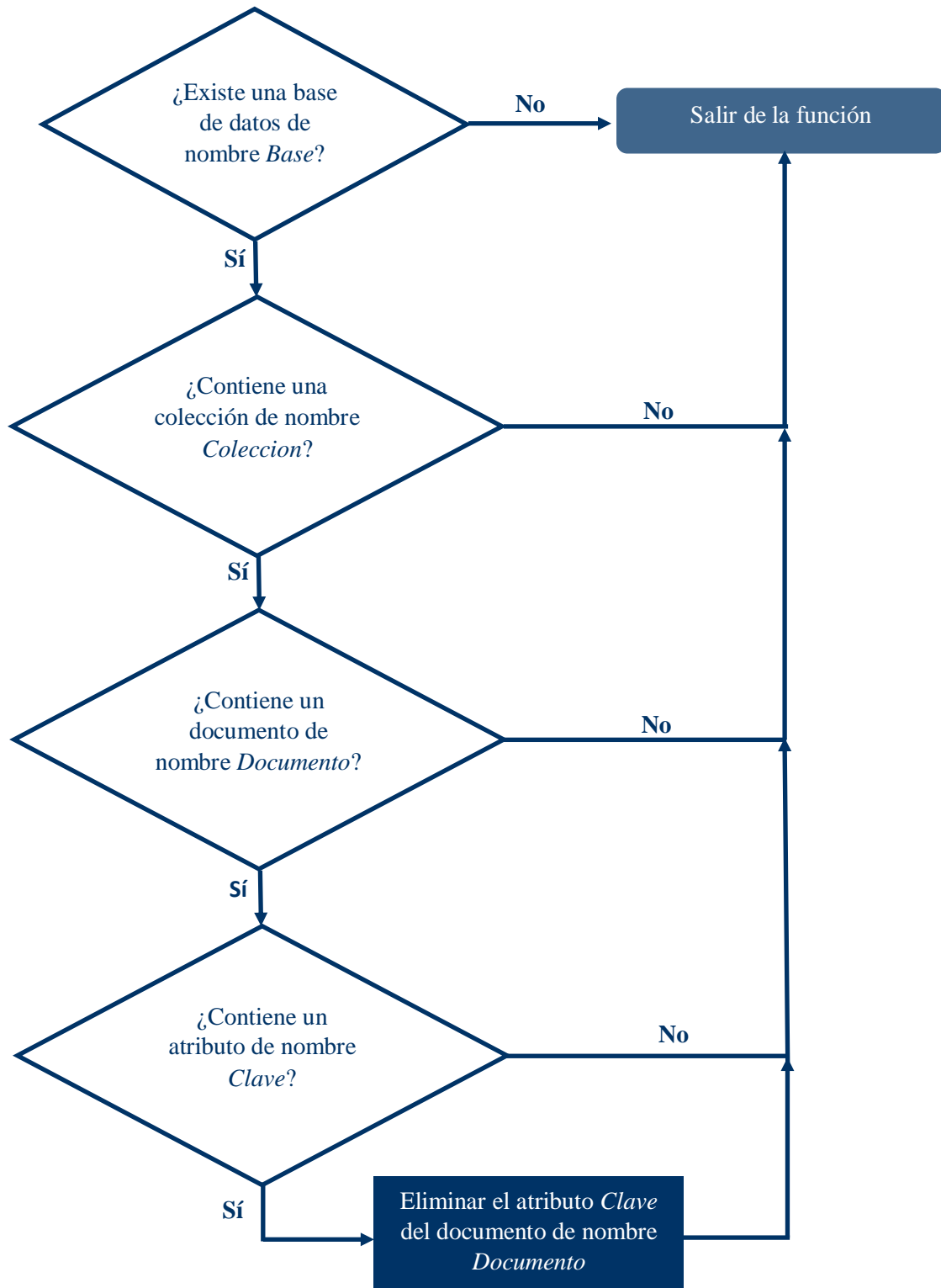


Figura 44: Función EliminarAtributoDocumento(). Diagrama de flujo.

## 4.1.3.4. Establecer una relación entre dos documentos

```
void EstablecerRelacion( )
```

Parámetros	string Base1, string Coleccion1, string Documento1, string Base2, string Coleccion2, string Documento2, string confianza
------------	--

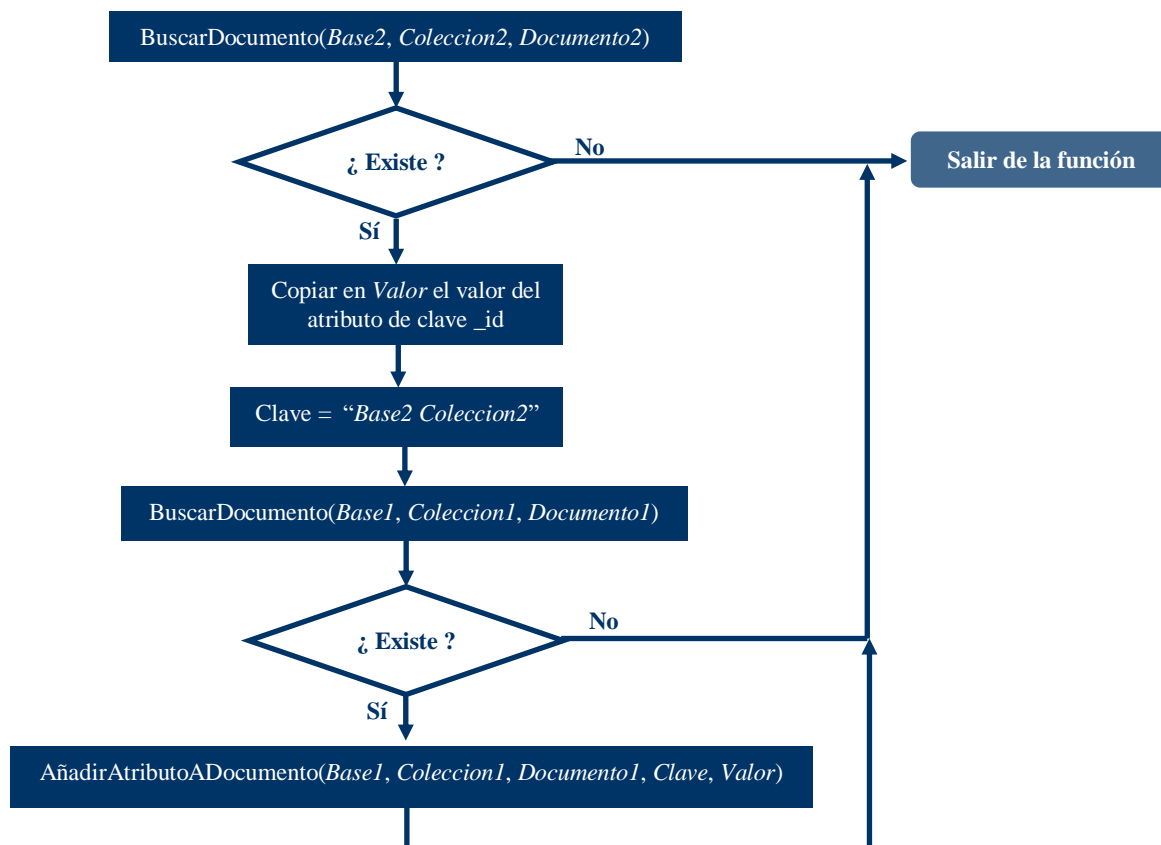


Figura 45: Función EstablecerRelacion( ). Diagrama de flujo.

#### 4.1.3.4. Completar un documento

El sistema puede considerar incompletos aquellos documentos que no contengan alguno de los atributos comunes a su colección, siendo estos aquellos atributos presentes en más del 75% de los documentos que componen la colección.

void PreguntarCuriosidad( )

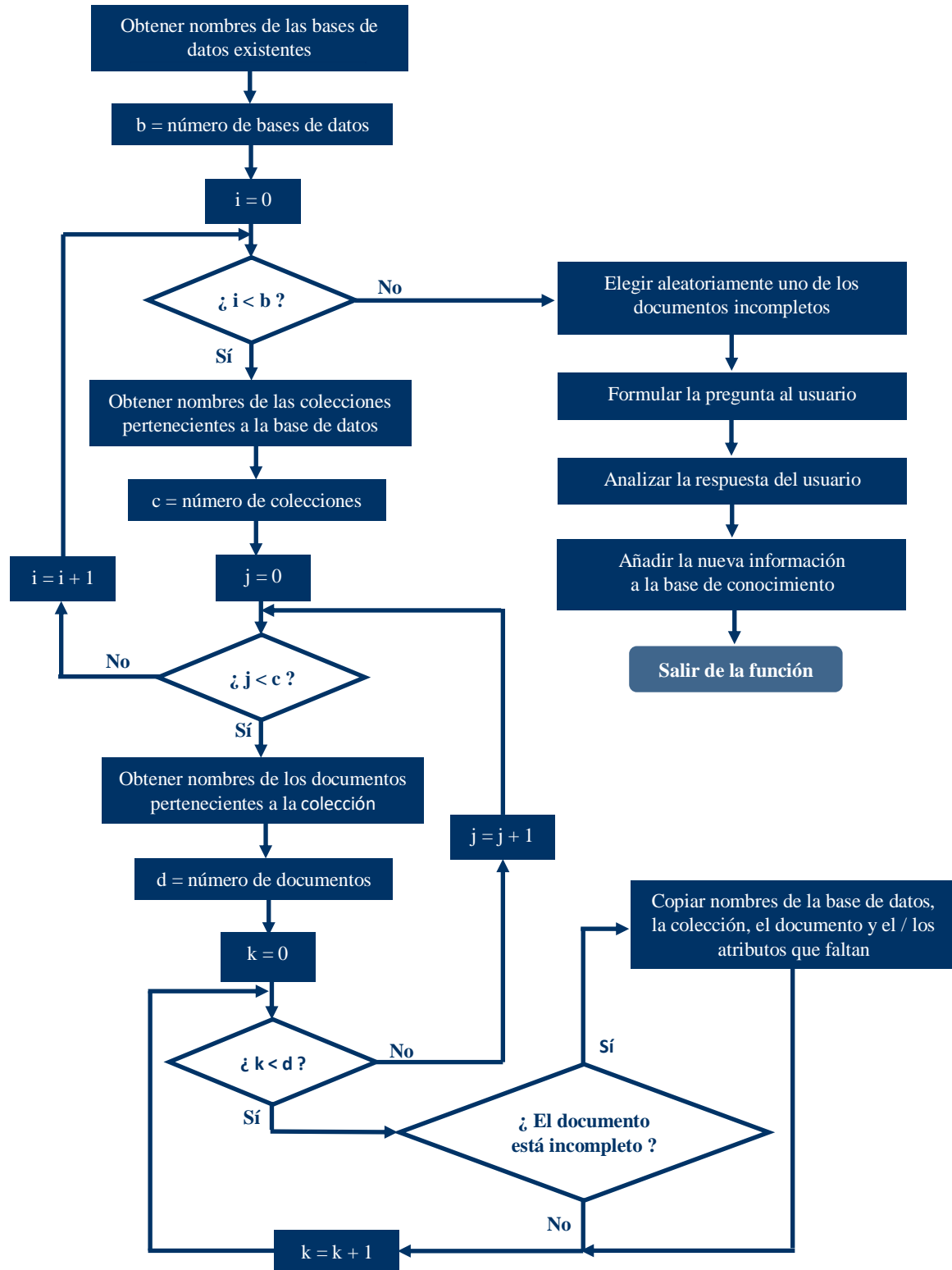


Figura 46: Función PreguntarCuriosidad( ). Diagrama de flujo.



#### 4.1.4. Archivos multimedia

En este apartado se describen las funciones desarrolladas mediante diagramas de flujo, según la operación que permiten realizar utilizando archivos multimedia.

##### 4.1.4.1. Añadir archivo multimedia a la base de conocimiento

void AñadirArchivoMultimedia( )

Parámetros	string Tipo, string Archivo, string Ruta
------------	--

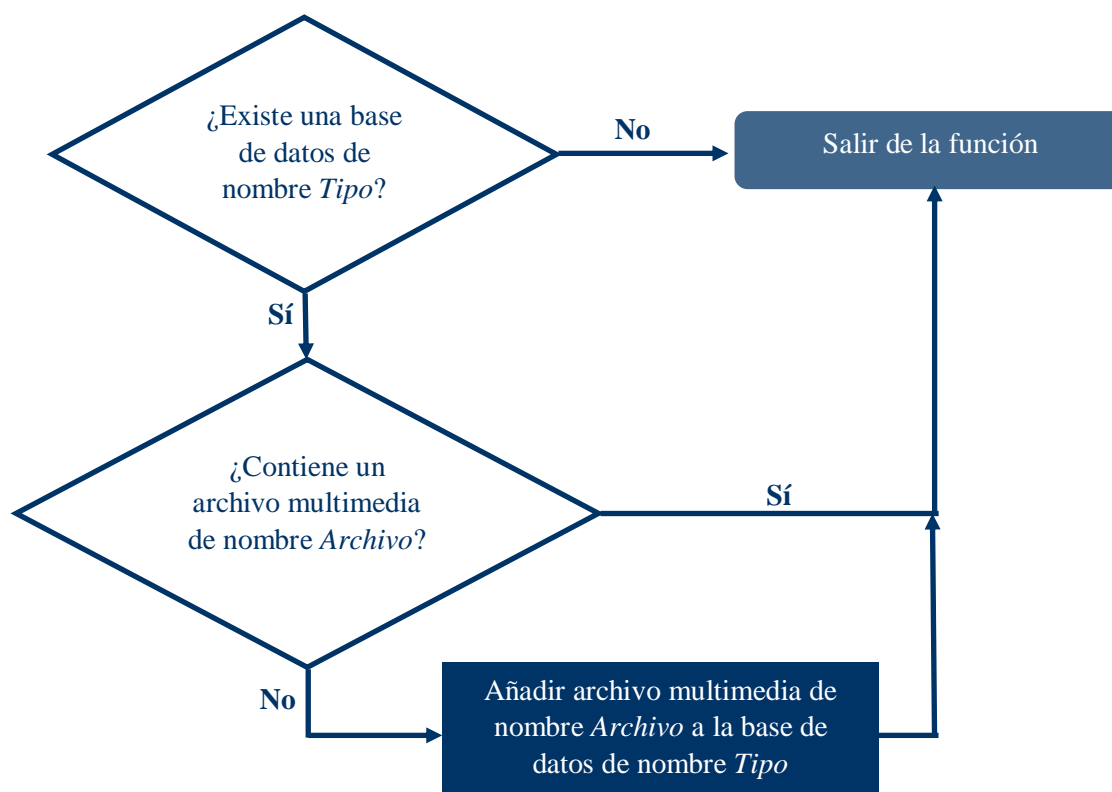


Figura 47: Función AñadirArchivoMultimedia( ). Diagrama de flujo.

## 4.1.4.2. Eliminar archivo multimedia de la base de conocimiento

```
void EliminarArchivoMultimedia( )
```

Parámetros	string Tipo, string Archivo
------------	-----------------------------

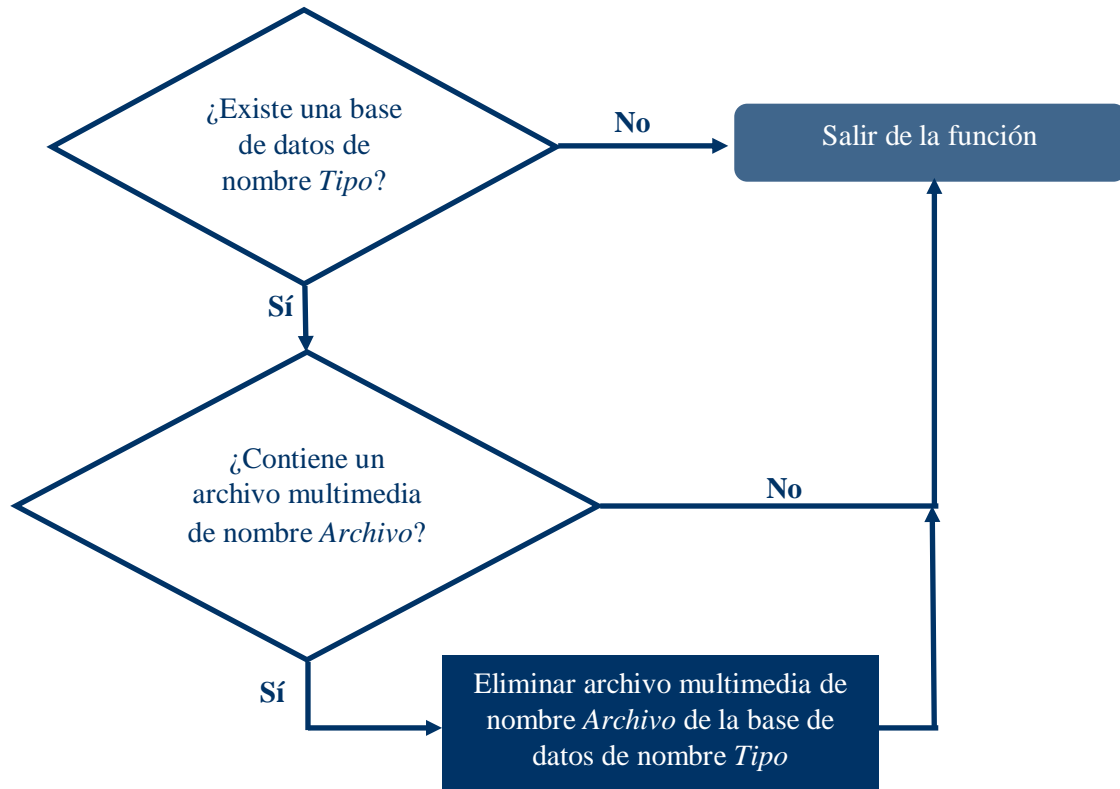


Figura 48: Función `EliminarArchivoMultimedia()`. Diagrama de flujo.

## 4.1.4.3. Consultar archivo multimedia

Podemos llevar a cabo la consulta de un archivo multimedia introduciendo la cadena que lo identifica (ID) o introduciendo su nombre.

```
void ConsultarArchivoMultimedia( )
```

Parámetros	string ID, string Ruta
------------	------------------------

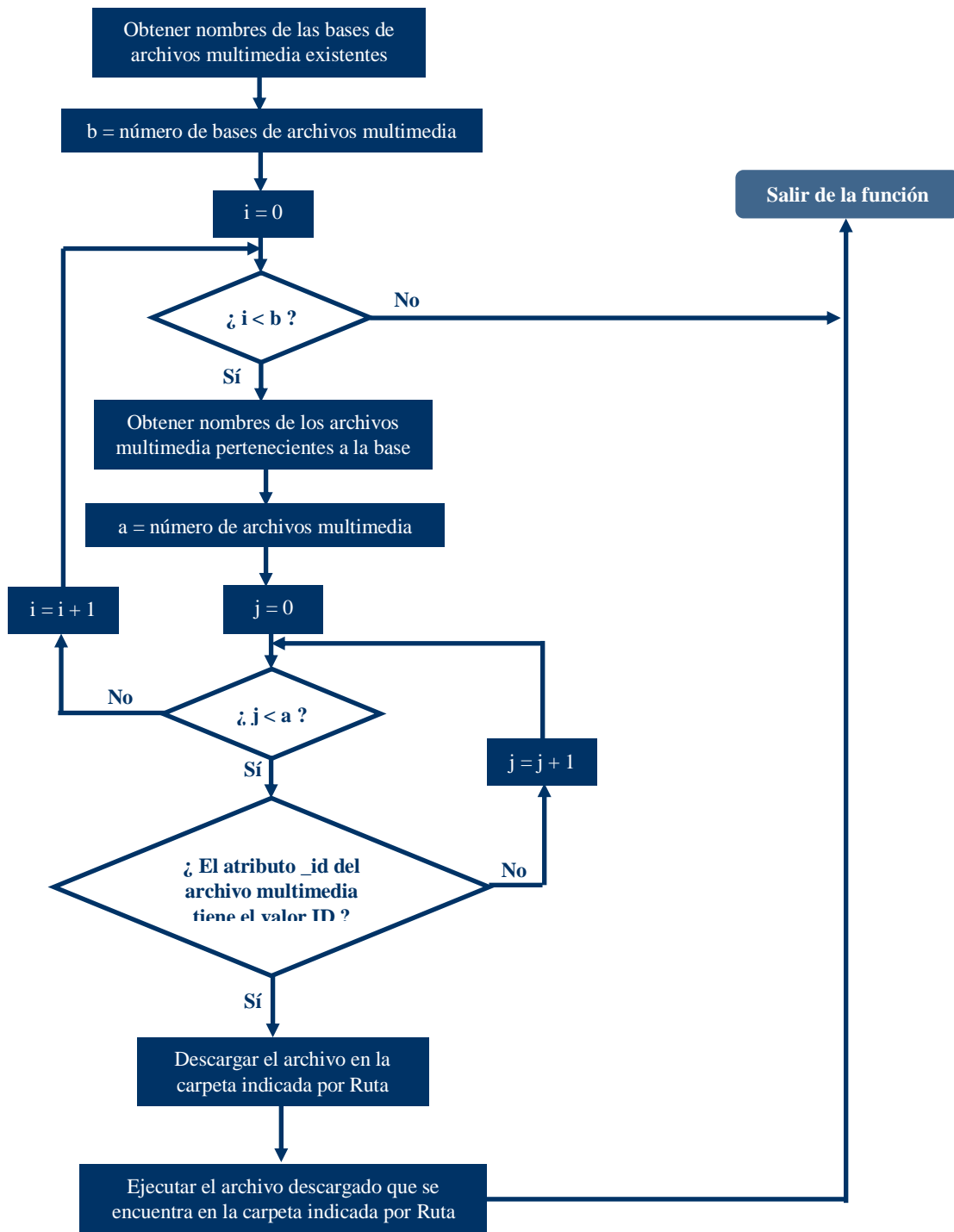


Figura 49: Función ConsultarArchivoMultimedia( ). Diagrama de flujo

```
void ConsultarArchivoMultimediaPorNombre( )
```

Parámetros	string Tipo, string Archivo, string Ruta
------------	--

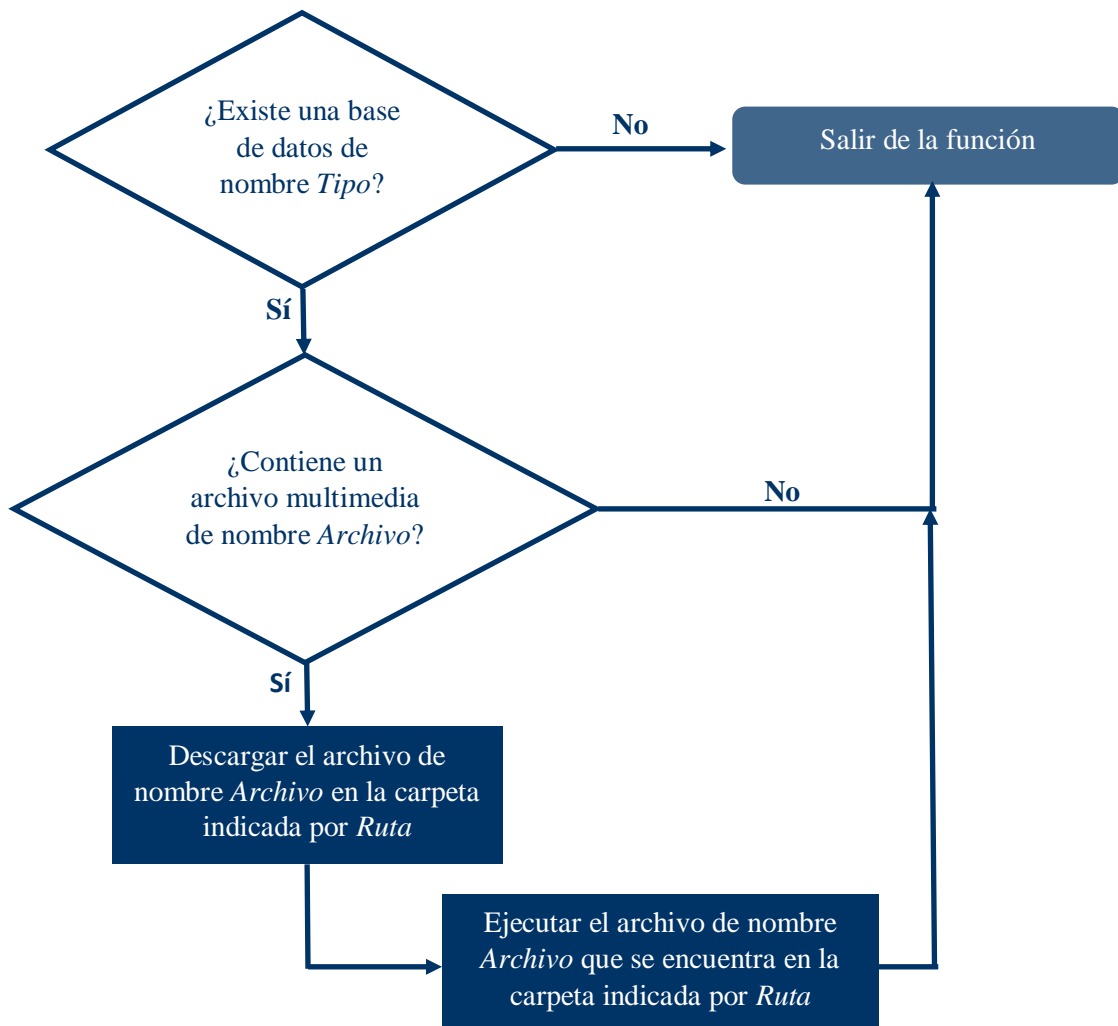


Figura 50: Función `ConsultarArchivoMultimediaPorNombre()`. Diagrama de flujo.

## 4.1.4.4. Establecer relación entre un documento y un archivo multimedia

```
void EstablecerRelacionMultimedia ( )
```

Parámetros	string Base, string Coleccion, string Documento, string Clave, string Tipo, string Archivo, string confianza
------------	--

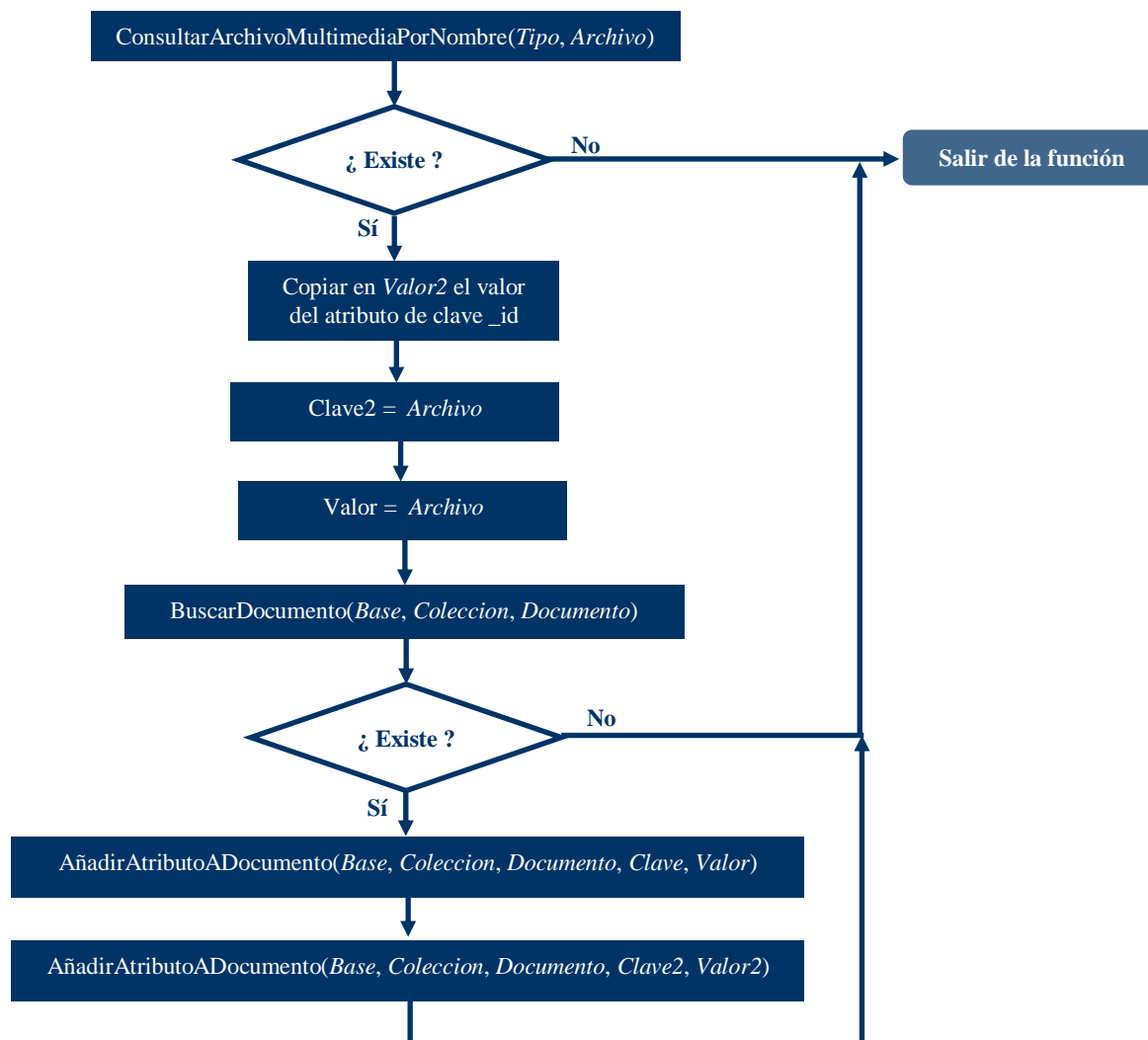


Figura 51: Función EstablecerRelaciónMultimedia( ). Diagrama de flujo.

## 4.2. INTERFAZ GRÁFICA

En cuanto a diseño se refiere, el desarrollo de este Trabajo Fin de Grado ha obtenido como resultado una interfaz gráfica codificada en lenguaje de programación C++.

El primer formulario de Windows que la aplicación muestra al ejecutar el proyecto de Microsoft Visual Studio *BaseConocimiento.sln* es el siguiente:

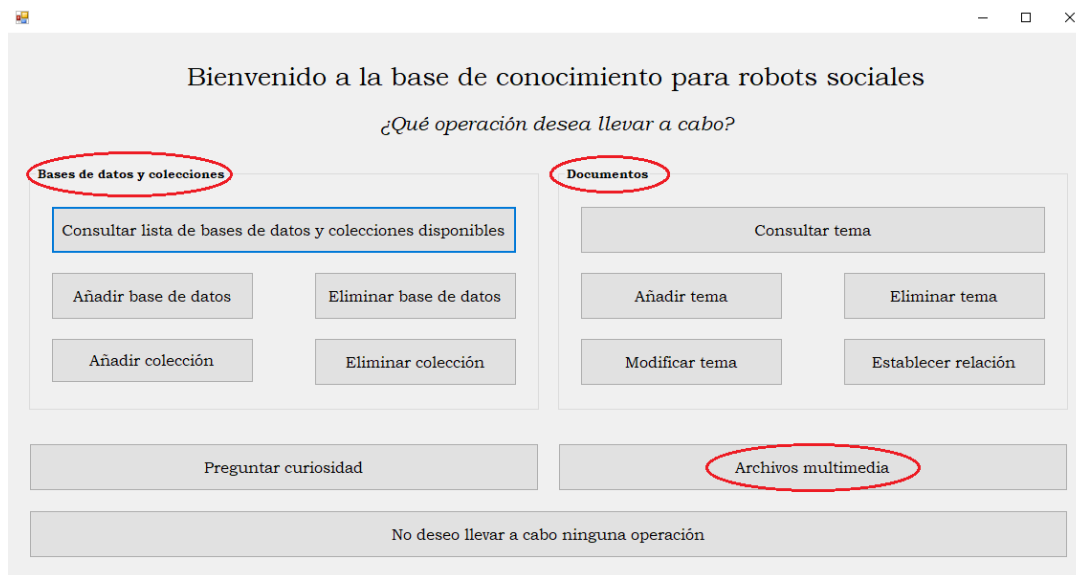


Figura 52: Panel de inicio.

Se trata de un panel formado única y exclusivamente por botones que nos permiten desplazarnos a nuevos paneles, a excepción del botón *No deseo llevar a cabo ninguna operación*, el cual sirve para abandonar la base de conocimiento.

Según el nivel del sistema en el que operan, podemos clasificar los botones en tres grupos: botones relacionados con las bases de datos y colecciones, botones relacionados con los documentos existentes y botones relacionados con archivos multimedia.

	Botón	Panel
Bases de datos y colecciones	Consultar lista de bases de datos y colecciones disponibles	Lista bases de datos y colecciones
	Añadir base de datos	Añadir base de datos
	Eliminar base de datos	Eliminar base de datos
	Añadir colección	Añadir colección
	Eliminar colección	Eliminar colección
Documentos	Consultar tema	Consultar documento
	Añadir tema	Añadir documento
	Eliminar tema	Eliminar documento
	Modificar tema	Modificar documento
	Establecer relación	Establecer relación entre documentos
Archivos multimedia	Archivos multimedia	Archivos multimedia

Tabla 10: Nombres paneles y botones de Panel de inicio.

### 4.2.1. Bases de datos y colecciones

En este apartado se describe el diseño de los paneles desarrollados, según la operación que permiten realizar utilizando bases de datos y colecciones.

#### 4.2.1.1. Panel lista bases de datos y colecciones

Está dividido en dos zonas claramente diferenciadas: a la izquierda encontramos una vista de árbol con los nombres de las bases de datos disponibles (nodos) y sus correspondientes colecciones (subnodos), además de un buscador y el botón *Actualizar*; mientras que, en el lado derecho, el panel muestra cuatro botones destinados a hacer operaciones básicas con bases de datos y colecciones.

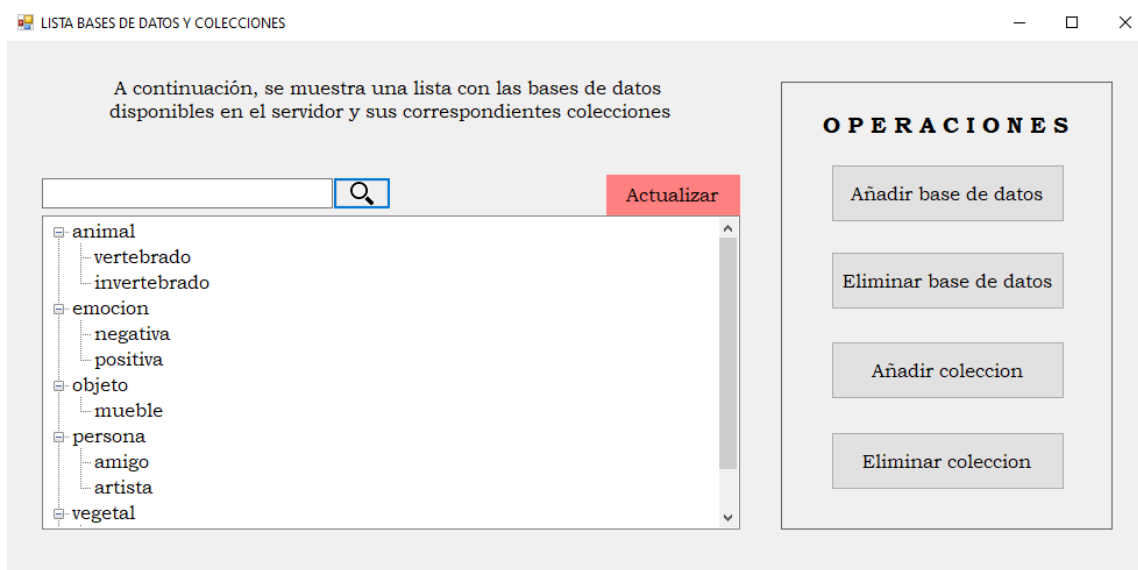


Figura 53: Panel lista bases de datos y colecciones.

Tanto al cargar el panel como al pulsar el botón *Actualizar*, el programa muestra en la vista de árbol los nombres de todas las bases de datos y colecciones existentes en la base de conocimiento.

En cuanto al buscador situado en la parte superior del panel, permite comprobar si una base de datos o una colección existe. Para ello, el usuario debe escribir el nombre de la consulta en el espacio en blanco y pulsar el botón representado por una lupa.

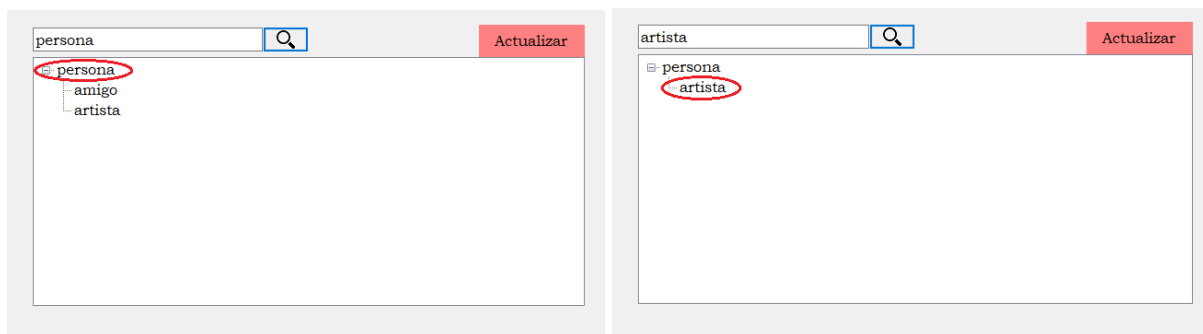


Figura 54: Panel lista de bases de datos y colecciones. Ejemplo.

Por otro lado, los botones del cuadro de operaciones, situado a la derecha del panel, dan acceso a los siguientes paneles.

Botón	Panel
Añadir base de datos	Añadir base de datos
Eliminar base de datos	Eliminar base de datos
Añadir colección	Añadir colección
Eliminar colección	Eliminar colección

Tabla 11: Panel lista bases de datos y colecciones. Botones.

#### 4.2.1.2. Panel añadir base de datos

Está formado por dos cuadros de texto, uno para introducir el nombre de la base de datos que se desea añadir a la base de conocimiento y otro para el nombre de la colección que MongoDB nos exige añadir con el objetivo de reconocer la base de datos como tal; además de un botón, que será el encargado de iniciar la operación.

Figura 55: Panel añadir base de datos.

Al pulsar el botón *Añadir base de datos*, el programa ejecuta la función *CrearBaseDeDatos()*, descrita en el apartado anterior.

#### 4.2.1.3. Panel eliminar base de datos

Está formado por un cuadro de texto en el que el usuario debe introducir el nombre de la base de datos que desea eliminar de la base de conocimiento, y un botón, que será el encargado de iniciar la operación.

Con el objetivo de evitar nombres incorrectos o erratas, el cuadro de texto dispone de un desplegable en el que se muestran los nombres de todas las bases de datos existentes en el servidor.

Esto otorga al usuario la posibilidad de seleccionar un ítem o, si así lo desea, escribir el nombre como lo haría en un cuadro de texto normal.



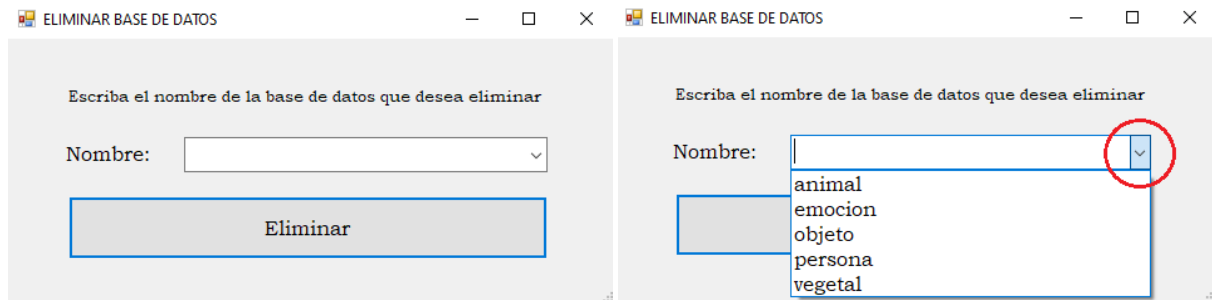


Figura 56: Panel eliminar base de datos.

Al pulsar el botón *Eliminar base de datos*, el programa ejecuta la función *EliminarBaseDeDatosExistente()*, descrita en el apartado anterior.

#### 4.2.1.4. Panel añadir colección

Está formado por dos cuadros de texto, uno para introducir el nombre de la colección que se desea añadir a la base de conocimiento y otro para el nombre de la base de datos a la que el usuario desea que pertenezca; además de un botón, que será el encargado de iniciar la operación.

Al igual que en el panel anterior, el cuadro de texto destinado al nombre de la base de datos incluye un desplegable en el que se muestran los nombres de todas las bases de datos existentes.

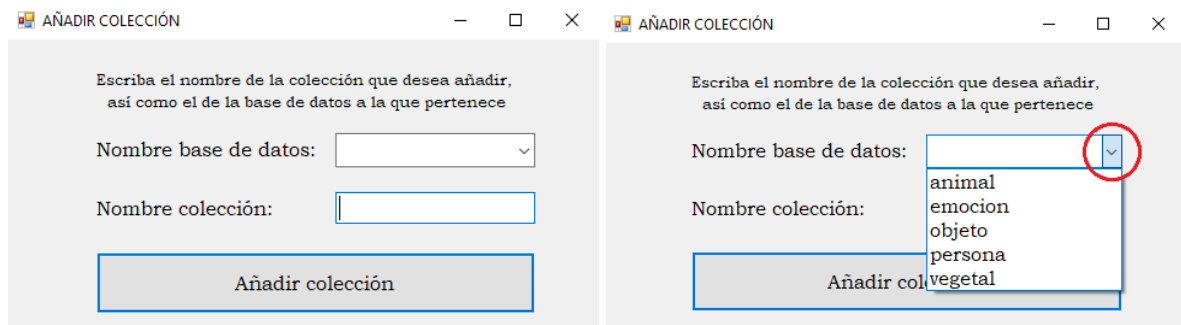


Figura 57: Panel añadir colección.

Al pulsar el botón *Añadir colección*, el programa ejecuta la función *CrearColeccion()*, descrita en el apartado anterior.

#### 4.2.1.5. Panel eliminar colección

Está formado por dos cuadros de texto, uno para introducir el nombre de la colección que se desea eliminar de la base de conocimiento y otro para el nombre de la base de datos a la que pertenece; además de un botón, que será el encargado de iniciar la operación.

En este caso, para evitar nombres erróneos, tanto el cuadro de texto destinado a la base de datos como el que está destinado a la colección incluyen desplegables.

El primero muestra los nombres de todas las bases de datos existentes en el sistema y, el segundo, una vez seleccionado un ítem del primero, muestra todas las colecciones pertenecientes a la base de datos elegida.

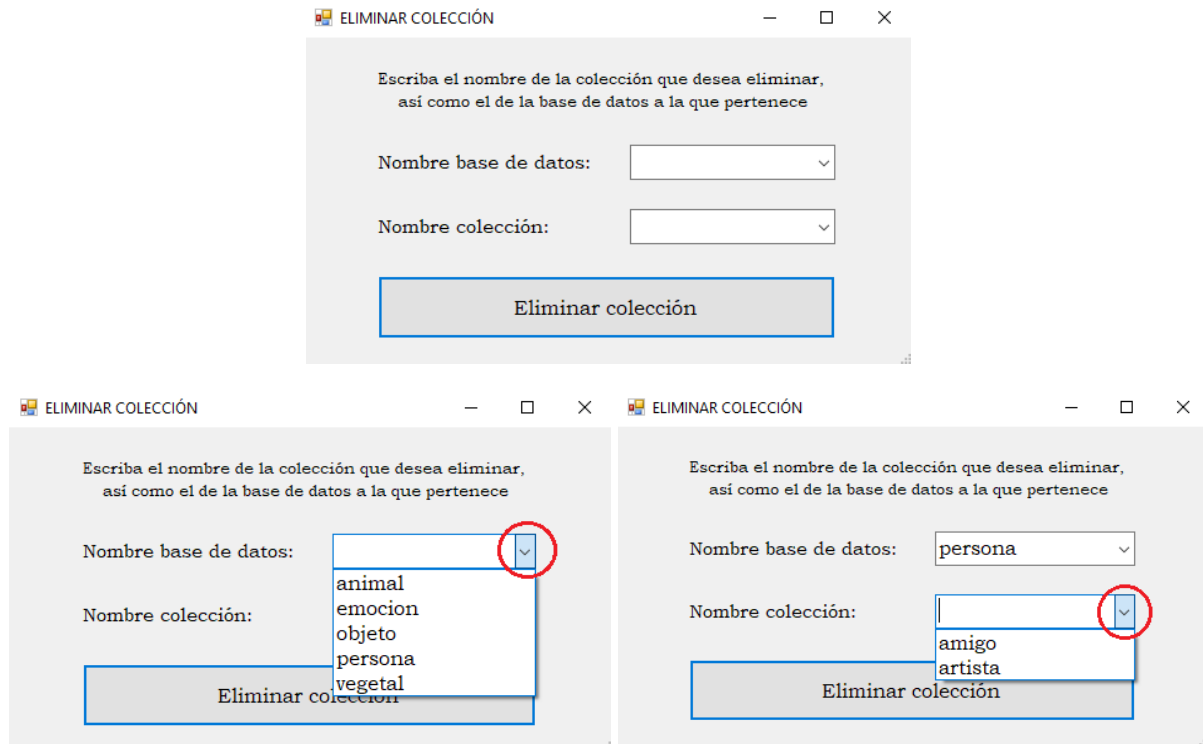


Figura 58: Panel eliminar colección.

Al pulsar el botón *Eliminar colección*, el programa ejecuta la función *EliminarColeccionExistente()*, descrita en el apartado anterior.

#### 4.2.2. Documentos

En este apartado se describe el diseño de los paneles desarrollados, según la operación que permiten realizar utilizando documentos BSON.

##### 4.2.2.1. Panel consultar documento

El panel está dividido, en primer lugar, en dos zonas claramente diferenciadas: la zona situada en el lado derecho del panel, formada por un cuadro de texto en el que se mostrarán los documentos resultantes de las búsquedas realizadas y la zona situada en el lado izquierdo del panel, destinada a introducir los datos necesarios para realizar consultas.

Como hemos visto en el apartado anterior, a la hora de consultar el contenido de uno o varios documentos, tenemos distintas opciones, motivo por el cual el lado izquierdo del panel está formado por los siguientes grupos: *Consultar por colección*, que permite al usuario visualizar el contenido de todos los documentos pertenecientes a una colección; *Consultar por ID*, para que el usuario pueda obtener el contenido de un documento introduciendo la cadena alfanumérica (ID) que lo identifica; *Consultar por nombre*, para consultar el contenido de un documento según nombre y, finalmente *Consultar por atributos*, que hace posible la opción de consultar el contenido de todos los documentos que contengan ciertos atributos.

The screenshot shows a web application window titled "CONSULTAR DOCUMENTO". It contains four search sections on the left and a large empty area on the right.

- Consultar por colección:** Includes dropdowns for "BD:" and "C:", and a search button.
- Consultar por ID:** Includes a single text input field and a search button.
- Consultar por nombre:** Includes dropdowns for "BD:" and "C:", a text input field, and a search button.
- Consultar por atributo:** Includes four pairs of text input fields labeled "Atributo 1:" through "Atributo 4:", and a "Consultar" button at the bottom.

Figura 59: Panel consultar documento.

Para consultar el contenido de todos los documentos que forman parte de una colección, disponemos de dos cuadros de texto: uno para introducir el nombre de la colección seleccionada y el otro para la base de datos a la que pertenece.

The screenshot shows the same "CONSULTAR DOCUMENTO" window, but now with search results displayed in the right-hand area. The "Consultar por colección" section has "BD:" set to "persona" and "C:" set to "artista".

The search results are displayed as a list of JSON objects:

```
{ "_id": { "$oid": "60cbacfab70000010005ab3" }
  "identificador": "Picasso"
  "base de datos": "persona"
  "coleccion": "artista"
  "apellido": "Picasso"
  "disciplina": "pintura" }

{ "_id": { "$oid": "60cbad14fb70000010005ab5" }
  "identificador": "Velazquez"
  "base de datos": "persona"
  "coleccion": "artista"
  "nombre": "Diego"
  "apellido": "Velazquez"
  "disciplina": "pintura" }

{ "_id": { "$oid": "60cbad3afb70000010005ab7" }
  "identificador": "Mozart"
  "base de datos": "persona"
  "coleccion": "artista"
  "nombre": "Wolfgang Amadeus"
  "apellido": "Mozart" }

{ "_id": { "$oid": "60cbad59fb70000010005ab9" }
  "identificador": "Beethoven"
  "base de datos": "persona"
  "coleccion": "artista"
  "nombre": "Ludwig van"
  "apellido": "Beethoven"
  "disciplina": "musica" }
```

Figura 60: Consultar documento por colección.

Al pulsar el botón representado por una lupa, el programa ejecuta la función *BuscarDocumentosColeccion( )*, descrita en el apartado anterior.

Otra opción es consultar el contenido de un documento introduciendo su nombre, para lo que disponemos de tres cuadros de texto. No obstante, los espacios reservados para los nombres de la colección y la base de datos a la que pertenece pueden dejarse en blanco, en cuyo caso el programa mostrará todos los documentos que coincidan con el nombre introducido.

The screenshot shows a window titled "CONSULTAR DOCUMENTO" with four search methods on the left and a JSON document on the right.

- Consultar por colección:** BD: [dropdown], C: [dropdown], [Search button]
- Consultar por ID:** [Text input], [Search button]
- Consultar por nombre:** BD: [person] dropdown, C: [artista] dropdown, [Picasso] text input, [Search button]
- Consultar por atributo:** Atributo 1: [ ] [ ], Atributo 2: [ ] [ ], Atributo 3: [ ] [ ], Atributo 4: [ ] [ ], [Consultar button]

The JSON document displayed on the right is:

```
{ "_id" : { "$oid" : "60cbacfab70000010005ab3" }
  "identificador" : "Picasso"
  "base de datos" : "persona"
  "coleccion" : "artista"
  "apellido" : "Picasso"
  "disciplina" : "pintura"
  "emocion positiva" : "60cbb105fb70000010005abb" }
```

Figura 61: Consultar documento por nombre.

Al pulsar el botón representado por una lupa, el programa ejecuta la función *BuscarDocumento( )*, descrita en el apartado anterior.

La opción consultar documento por ID es útil en casos en los que algún documento contiene una relación, como ocurre en la imagen anterior, en la que Picasso está relacionado con un documento perteneciente a la colección positiva de la base de datos emoción.

Para poder consultar ese documento, basta con introducir la cadena alfanumérica en el cuadro de texto reservado para el ID y pulsar el botón. Esto generará un nuevo panel en el que se mostrará el documento seleccionado, sin ocultar el panel actual, de manera que el usuario pueda ver a la vez tanto el documento base, en este caso Picasso, como el documento relativo a la relación.

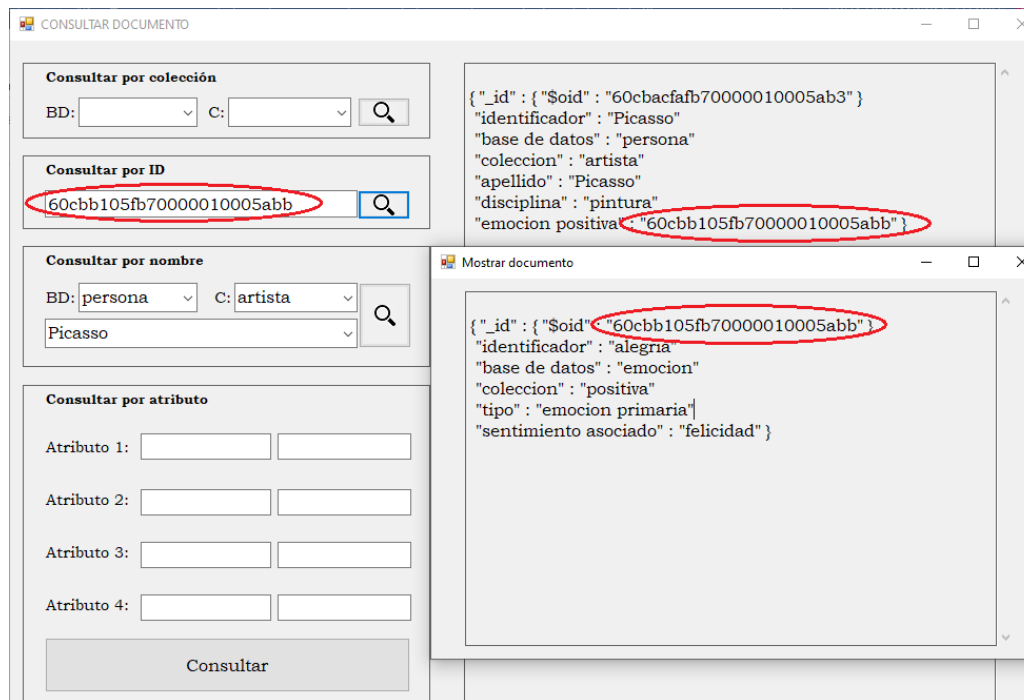


Figura 62: Consultar documento por ID.

Al pulsar el botón representado por una lupa, el programa ejecuta la función *BuscarDocumentoPorID()*, descrita en el apartado anterior.

Finalmente, podemos consultar los documentos que contienen determinados atributos de la siguiente forma:

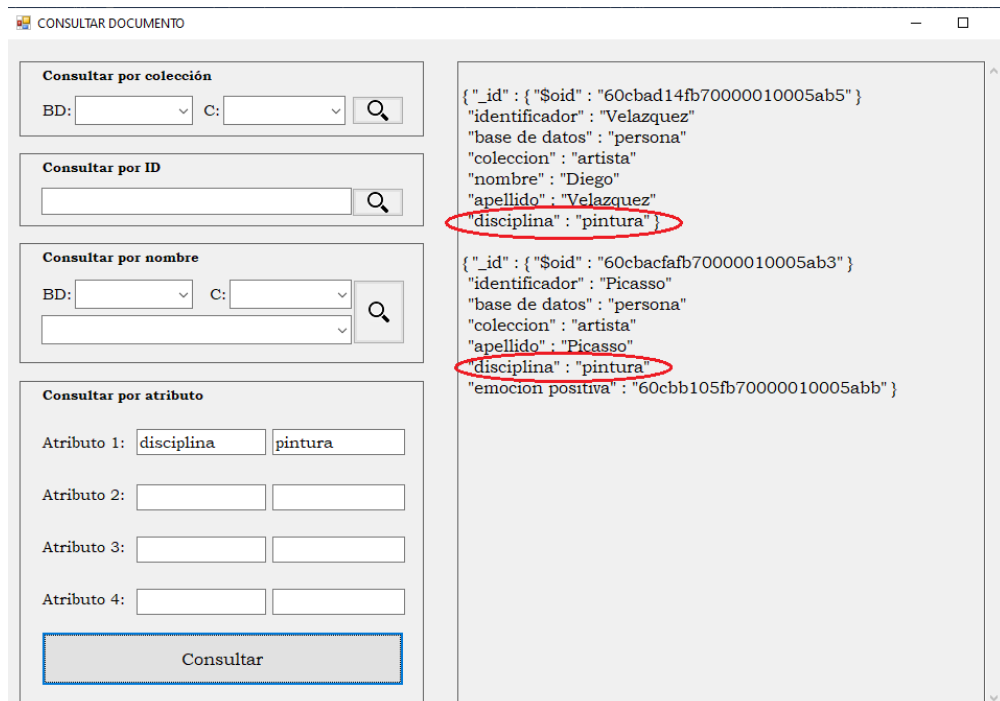


Figura 63: Consultar documento por atributos.

Al pulsar el botón representado por una lupa, el programa ejecuta la función *BuscarDocumentoPorAtributos()*, descrita en el apartado anterior.

#### 4.2.2.2. Panel añadir documento

Está dividido en dos zonas claramente diferenciadas: la zona situada en el lado izquierdo del panel, donde el usuario debe seleccionar la base de datos y la colección a la que desea añadir el documento; y la zona situada en el lado derecho del panel, destinada a introducir el nombre y los atributos del nuevo documento.

Figura 64: Panel añadir documento.

Al pulsar el botón *Añadir tema*, el programa ejecuta la función *CrearDocumento()*, descrita en el apartado anterior.

#### 4.2.2.3. Panel eliminar documento

Este panel, además de eliminar documentos, permite visualizarlos en el cuadro situado a la derecha, antes de llevar a cabo la operación.

Figura 65: Panel eliminar documento.

Al pulsar el botón *Eliminar tema*, el programa ejecuta la función *EliminarTema()*, descrita en el apartado anterior.

#### 4.2.2.4. Panel modificar documento

El panel está dividido, en primer lugar, en dos zonas claramente diferenciadas: la zona situada en el lado derecho del panel, formada por un cuadro de texto en el que se mostrarán los documentos modificados; y la zona situada en el lado izquierdo del panel, destinada a introducir los datos necesarios para realizar modificaciones.

El cuadro situado en la parte superior izquierda está destinado a introducir el nombre, la colección y la base de datos del documento que se desea modificar.

Por otro lado, como hemos visto en el apartado anterior, a la hora de modificar un documento, tenemos distintas opciones, motivo por el cual el lado izquierdo del panel está formado por los siguientes grupos: *Añadir atributo*, que permite al usuario añadir un par clave-valor a un documento; *Modificar atributo*, para que el usuario pueda modificar el valor de un atributo perteneciente a un documento y, finalmente *Eliminar atributo*, que hace posible la opción de eliminar un par clave-valor de un documento.

Figura 66: Panel modificar documento.

Al pulsar el botón *Añadir atributo*, el programa ejecuta la función *AñadirAtributoADocumento()*, el botón *Modificar atributo* llama a la función *ModificarAtributoDocumento()* y, finalmente, el botón *Eliminar atributo* utiliza la función *EliminarAtributoDocumento()*, todas ellas descritas en el apartado anterior.

#### 4.2.2.5. Panel establecer relación entre documentos

Está dividido en dos zonas: la de la izquierda para el documento base y la de la derecha para el documento relación; pudiendo consultar el contenido de ambos documentos antes y después de llevar a cabo la operación de relación.

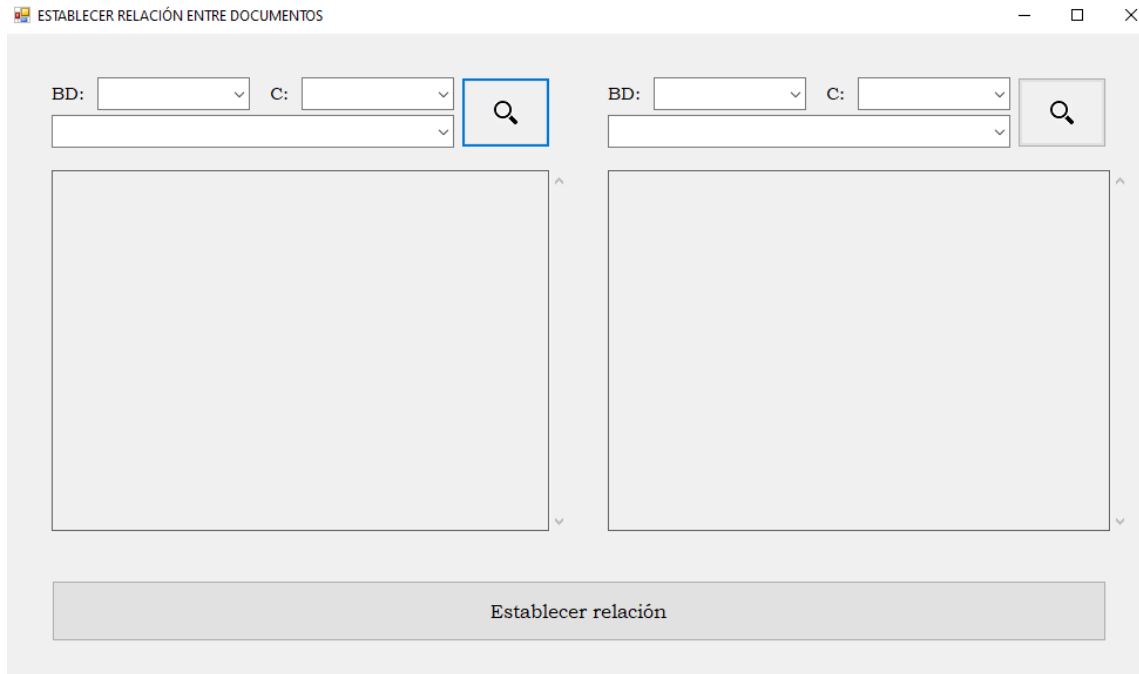


Figura 67: Panel establecer relación entre documentos.

Al pulsar el botón *Establecer relación*, el programa ejecuta la función *EstablecerRelacion()*, descrita en el apartado anterior.

A continuación, se muestra un ejemplo en el que se relacionan los documentos Picasso y alegría.

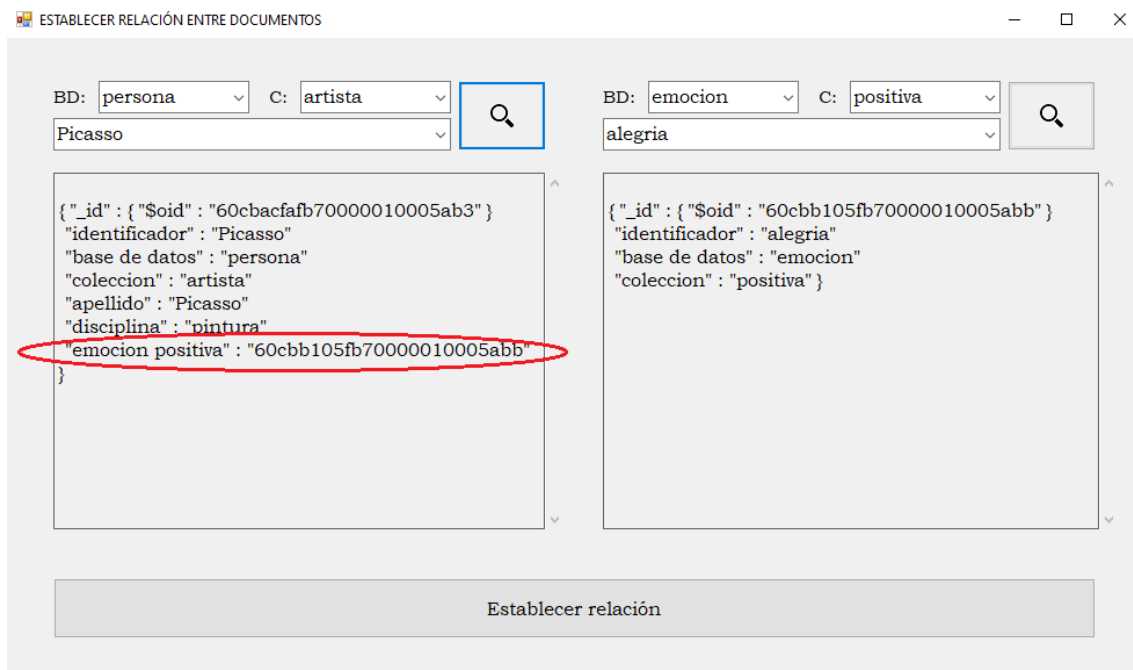


Figura 68: Panel establecer relación. Ejemplo.



4.2.2.6. Panel preguntar curiosidad

Al cargar el panel, el programa ejecuta la función *PreguntarCuriosidad()*, descrita en el apartado anterior; dando lugar a una distribución en la que encontramos la pregunta formulada por el sistema, un cuadro de texto para introducir la respuesta y, finalmente, un botón.

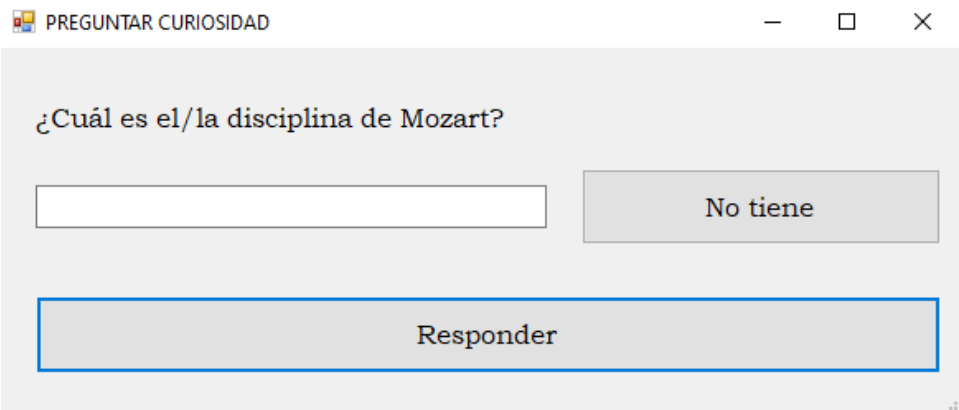


Figura 69: Panel preguntar curiosidad.

4.2.3. Archivos multimedia

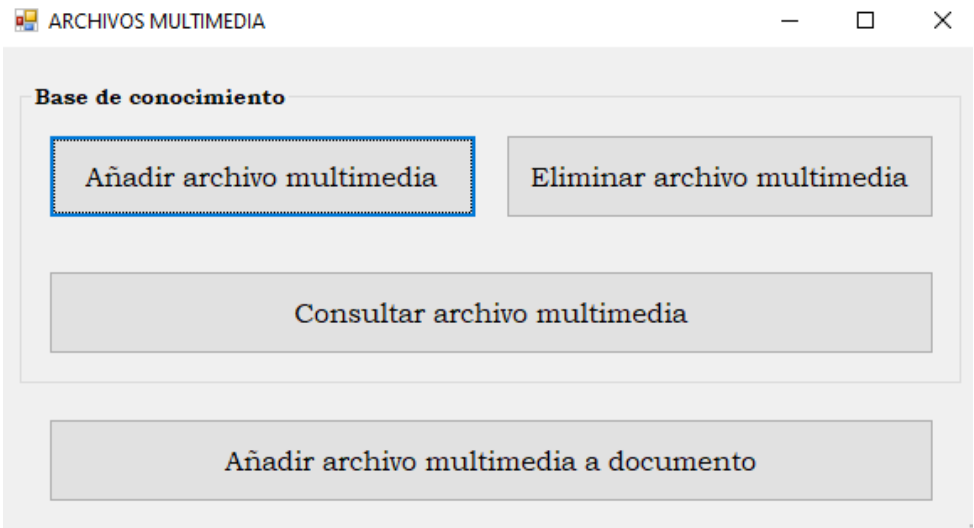


Figura 70: Panel archivos multimedia.

Se trata de un panel formado única y exclusivamente por botones, cada uno de los cuales nos permite desplazarnos a un nuevo panel en función de la operación que el usuario desee realizar.

	Botón	Panel
Base de conocimiento	Añadir archivo multimedia	Añadir archivo multimedia
	Eliminar archivo multimedia	Eliminar archivo multimedia
	Consultar archivo multimedia	Consultar archivo multimedia
Documentos	Añadir archivo multimedia a documento	Establecer relación entre documento y archivo multimedia

Tabla 12: Panel archivos multimedia. Botones.

A continuación, se describe el diseño de los paneles desarrollados, según la operación que permiten realizar utilizando archivos multimedia.

#### 4.2.3.1. Añadir archivo multimedia

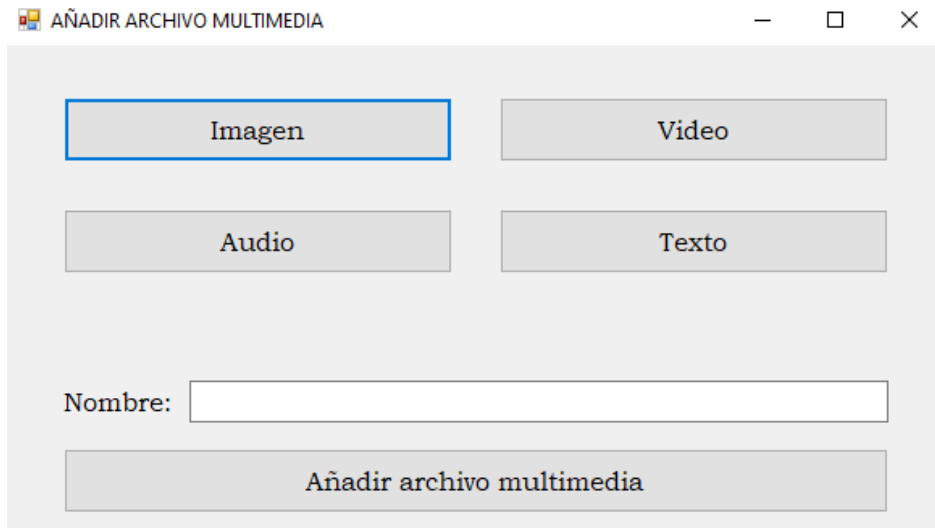


Figura 71: Panel añadir archivo multimedia.

Al pulsar uno de los cuatro botones superiores, el programa abre el explorador de archivos para que el usuario seleccione el archivo multimedia del tipo seleccionado que desea añadir a la base de conocimiento.

Finalmente, cuando se pulsa el botón *Añadir archivo multimedia*, el programa ejecuta la función *AñadirArchivoMultimedia()*, descrita en el apartado anterior.

#### 4.2.3.2. Eliminar archivo multimedia

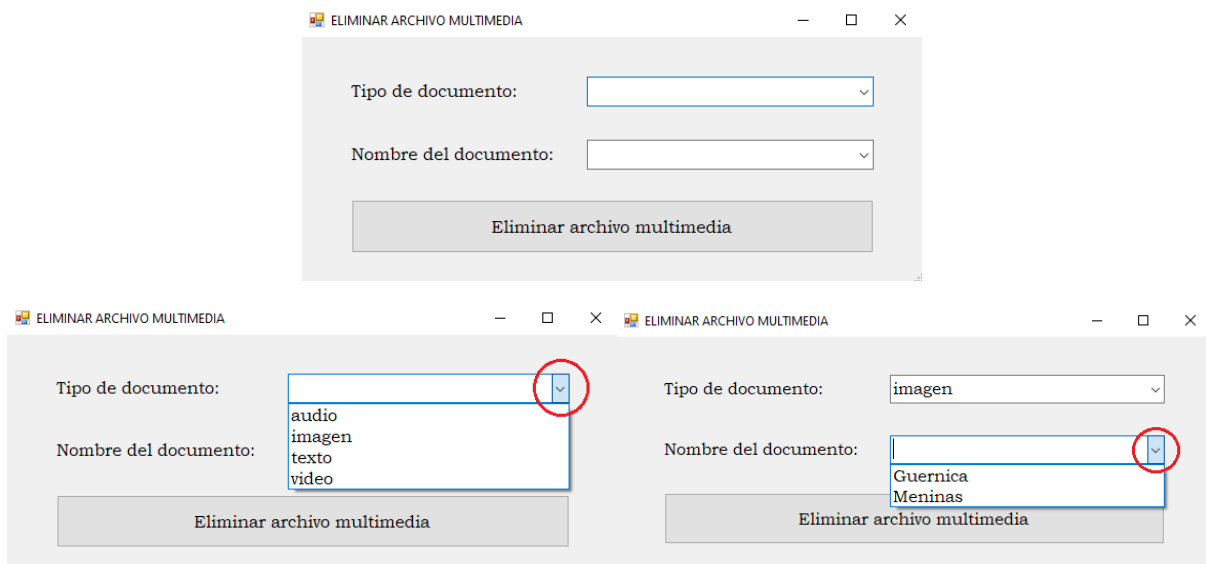


Figura 72: Panel eliminar archivo multimedia.

Al pulsar el botón *Eliminar archivo multimedia*, el programa ejecuta la función *EliminarArchivoMultimedia()*, descrita en el apartado anterior.

#### 4.2.3.3. Consultar archivo multimedia

Al cargar el panel *consultar archivo multimedia*, el programa abre el explorador de archivos e informa al usuario de la necesidad de seleccionar la carpeta en la que desea descargar el archivo multimedia.

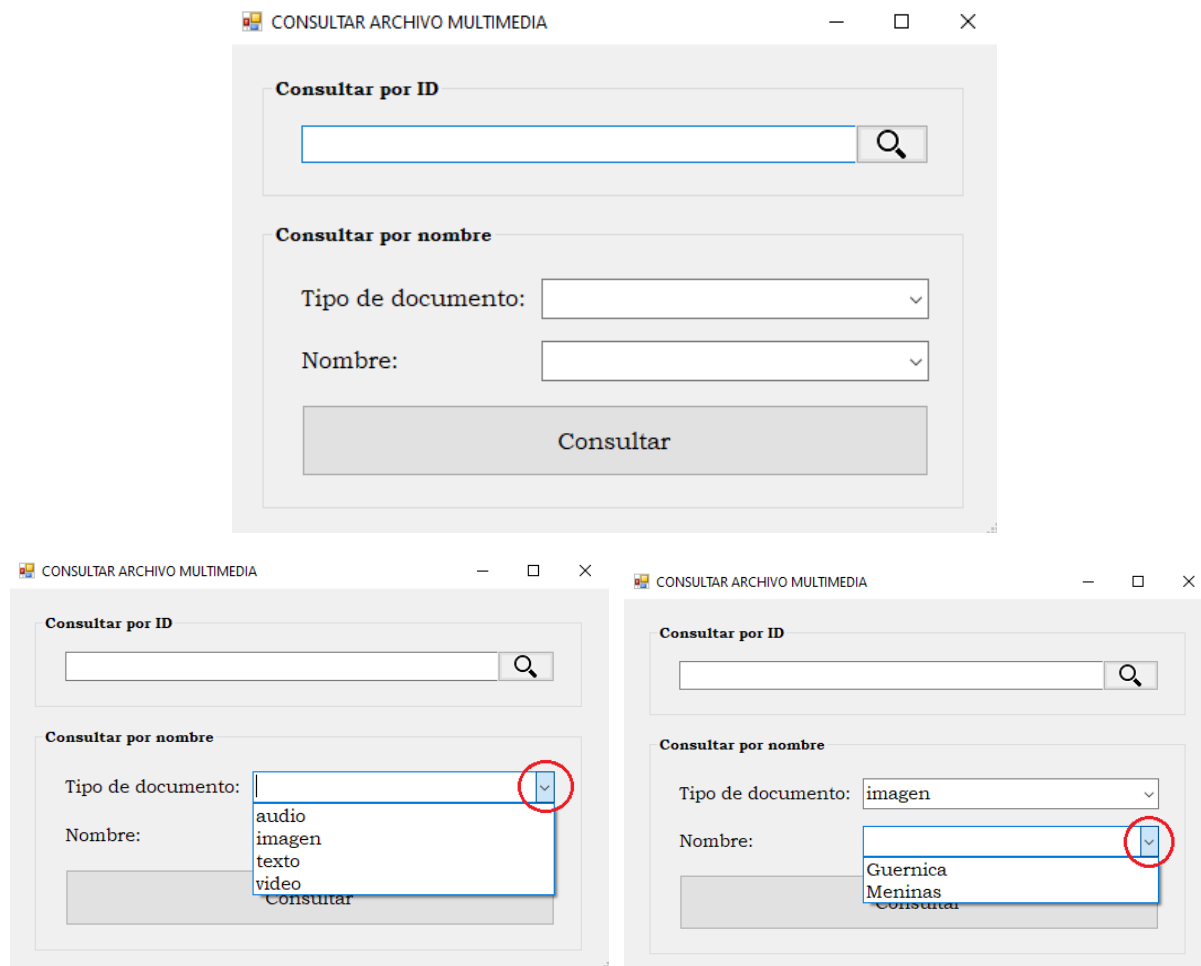


Figura 73: Panel consultar archivo multimedia.

Como se puede observar en la imagen, el panel está dividido en dos zonas, ofreciendo la posibilidad de consultar un archivo multimedia a partir de la cadena que lo identifica (ID) o a partir de su nombre.

Así, si el usuario elige la opción *Consultar por ID*, al pulsar el botón representado por una lupa, el programa ejecuta la función *ConsultarArchivoMultimedia()*, mientras que si el usuario decide por la opción *Consultar por nombre*, el programa ejecuta la función *ConsultarArchivoMultimediaPorNombre()*, ambas descritas en el apartado anterior.

#### 4.2.3.4. Establecer relación entre documento y archivo multimedia

Este panel está dividido en dos zonas: la de la izquierda corresponde al documento BSON y la de la derecha al archivo multimedia.

Los botones representados por lupas nos permiten consultar tanto el documento como el archivo multimedia antes de llevar a cabo la operación de relación.

En el caso del documento BSON, al pulsar el botón buscador aparecen sus atributos en el cuadro de texto reservado para ello en la parte inferior izquierda del panel; mientras que, en el caso del archivo multimedia, al pulsar el botón de consulta, el programa descarga el archivo en la carpeta previamente seleccionada y lo ejecuta.

**ESTABLECER RELACIÓN ENTRE DOCUMENTO Y ARCHIVO MULTIMEDIA**

**Documento:**

BD:  C:

**Atributo a añadir**

Nombre:

**Valor**

**Archivo multimedia:**

Tipo de documento:

Nombre documento:

Figura 74: Panel establecer relación entre documento y archivo multimedia

Al pulsar el botón *Establecer relación*, el programa ejecuta la función *EstablecerRelacionMultimedia()*, descrita en el apartado anterior.

Así, se añade dos atributos al documento BSON: el primero de ellos de clave la cadena introducida en el cuadro de texto “Nombre” y valor el texto introducido en el espacio reservado “Nombre documento”; y el segundo de clave el nombre del archivo multimedia y valor la cadena alfanumérica que identifica al archivo multimedia (ID).

En la siguiente imagen se muestra el resultado de relacionar el documento BSON y el archivo multimedia Guernica.

**ESTABLECER RELACIÓN ENTRE DOCUMENTO Y ARCHIVO MULTIMEDIA**

**Documento:**

BD:  C:

**Atributo a añadir**

Nombre:

**Valor**

**Archivo multimedia:**

Tipo de documento:

Nombre documento:

```
{ "_id" : { "$oid" : "60cbacfaf70000010005ab3" }
  "identificador" : "Picasso"
  "base de datos" : "persona"
  "coleccion" : "artista"
  "apellido" : "Picasso"
  "disciplina" : "pintura"
  "emocion positiva" : "60cbb105fb70000010005abb"
  "cuadro mas famoso" : "Guernica"
  "Guernica" : "60cbb18bfb70000010005abc" }
```

Figura 75: Panel establecer relación entre documento y archivo multimedia

## Capítulo 5

# CONCLUSIONES

### 5.1. CONCLUSIONES DEL PROYECTO

En este capítulo se procede a comprobar el cumplimiento de los objetivos propuestos en el capítulo de introducción.

Para ello, se deben analizar los resultados obtenidos prestando atención al sistema de conocimiento, la interfaz gráfica desarrollada y la portabilidad del programa.

#### 5.1.1. Sistema de conocimiento

Se ha hecho uso del servidor MongoDB para llevar a cabo el desarrollo de un sistema de conocimiento constituido por bases de datos no relacionales.

En la siguiente tabla se muestran los objetivos inicialmente establecidos junto a los resultados obtenidos tras la realización del Trabajo Fin de Grado.

	Objetivo sistema a desarrollar	Resultado sistema desarrollado
Acceso	Permitir el acceso a la información contenida en la base de conocimiento.	Permite la obtención del nombre y el contenido de las bases de datos, colecciones, documentos y archivos multimedia disponibles.
Adición	Permitir la adición de nuevos datos a la base de conocimiento.	Permite la adición de nuevas bases de datos, colecciones, documentos y archivos multimedia.
Modificación	Permitir al usuario modificar la información existente en la base de conocimiento.	Permite la modificación de documentos mediante la adición, modificación o eliminación de atributos.
Eliminación	Permitir al usuario eliminar datos concretos de la base de conocimiento.	Permite la eliminación de bases de datos, colecciones, documentos y archivos multimedia existentes.
Relación	Permitir al usuario establecer relaciones entre términos pertenecientes a la base de conocimiento.	Permite el establecimiento de relaciones entre dos documentos y entre un documento y un archivo multimedia.

Tabla 13: Objetivos y resultados.

Además de los resultados mostrados en la tabla anterior, se ha dotado a la base de conocimiento de una funcionalidad extra: preguntar curiosidades al usuario.

Si el sistema considera que uno de los documentos está incompleto (por ejemplo, si un documento de la base de datos persona no contiene el atributo nombre), puede consultar al usuario cuál es el valor correspondiente a dicho atributo para ese documento y añadirlo.

### 5.1.2. Interfaz gráfica

Para permitir la interacción entre el usuario y el sistema de conocimiento se ha desarrollado una interfaz gráfica en Microsoft Visual Studio 2019, constituida por los siguientes paneles:

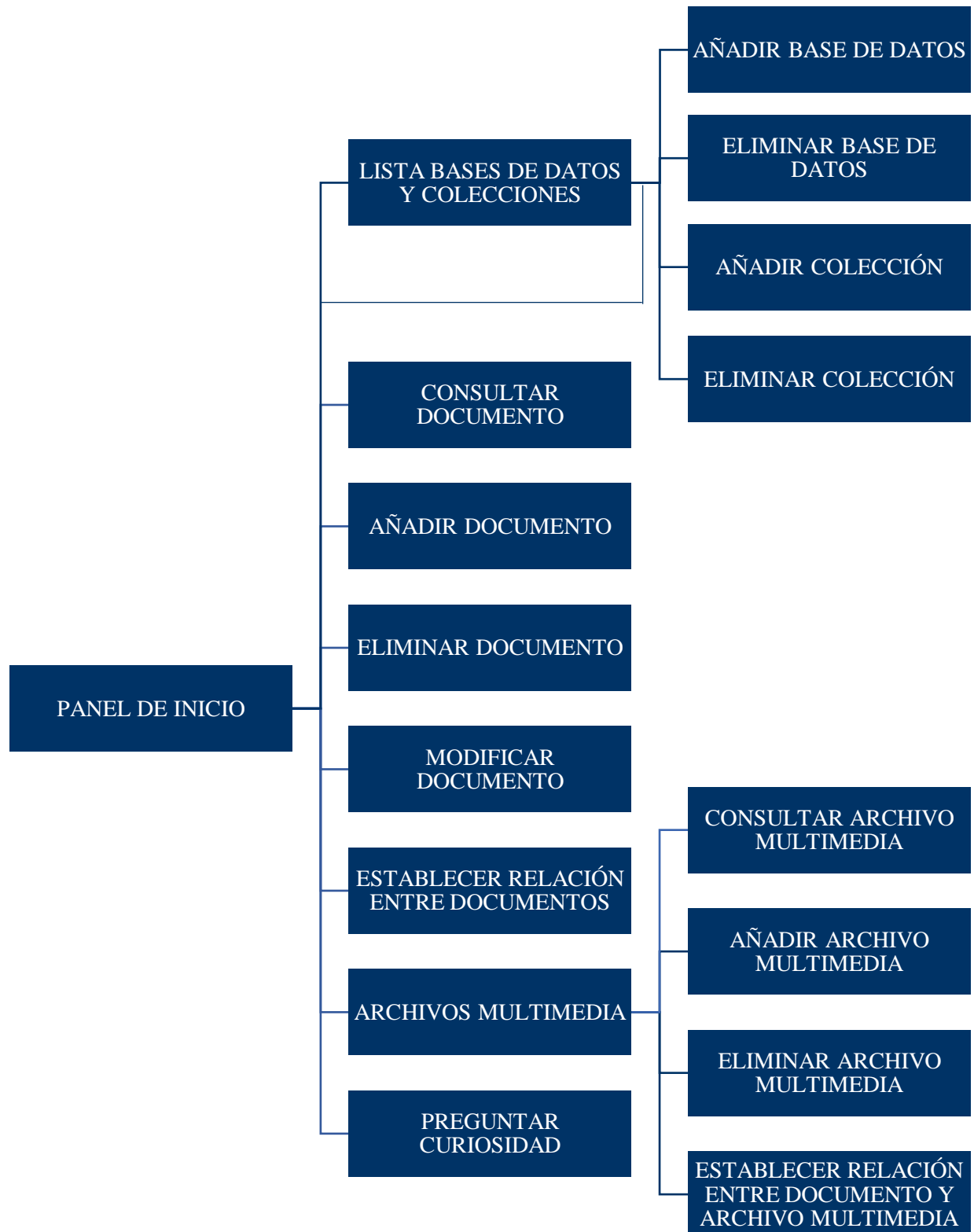


Figura 76: Paneles interfaz gráfica.

### 5.1.3. Portabilidad

Finalmente, para facilitar la posterior implantación del programa en robots sociales, se ha tratado en todo momento por separado la interfaz gráfica y la base de conocimiento, estructurando el código en funciones, contenidas en los siguientes ficheros de encabezamiento:

#### Botones\_BaseDeDatos.h

Función	Botón
CrearBaseDeDatos( )	Añadir base de datos
EliminarBaseDeDatosExistente( )	Eliminar base de datos

Tabla 14: Funciones y botones bases de datos.

#### Botones\_Coleccion.h

Función	Botón
CrearColeccion( )	Añadir colección
EliminarColeccionExistente( )	Eliminar colección

Tabla 15: Funciones y botones colecciones.

#### Botones\_Documento.h

Función	Botón
CrearDocumento( )	Añadir documento
EliminarTema( )	Eliminar documento
BuscarDocumentosColeccion( )	Consultar documento por colección
BuscarDocumentoPorID( )	Consultar documento por ID
BuscarDocumento( )	Consultar documento por nombre
BuscarDocumentoPorAtributos( )	Consultar documento por atributos
AñadirAtributoADocumento( )	Modificar documento añadiendo atributo
ModificarAtributoDocumento( )	Modificar documento modificando atributo
EliminarAtributoDocumento( )	Modificar documento eliminando atributo
EstablecerRelacion( )	Establecer relación entre dos documentos
PreguntarCuriosidad( )	Completar documentos

Tabla 16: Funciones y botones documentos.

#### Botones\_ArchivoMultimedia.h

Función	Botón
AñadirArchivoMultimedia( )	Añadir archivo multimedia
EliminarArchivoMultimedia( )	Eliminar archivo multimedia
ConsultarArchivoMultimedia( )	Consultar archivo multimedia por ID
ConsultarArchivoMultimediaPorNombre( )	Consultar archivo multimedia por ID
EstablecerRelacionMultimedia( )	Establecer relación entre un documento y un archivo multimedia

Tabla 17: Funciones y botones archivos multimedia.

## 5.2. LÍNEAS FUTURAS

En este apartado se exponen posibles líneas de investigación que pueden desarrollarse como continuación de este TFG.

### 5.2.1. Implementación

En primer lugar, debe llevarse a cabo la implementación de la base de conocimiento desarrollada en un robot social, para lo cual será necesario realizar algunas modificaciones en el código que permitan pasar de la interfaz desarrollada en Visual Studio a la interfaz de usuario que presente el robot; además de incluir el procesamiento de textos en lenguaje natural hablado.

Finalmente, para comprobar la validez y el correcto funcionamiento del sistema, deberán realizarse diferentes pruebas y experimentos con robots que presenten características distintas.

### 5.2.2. Filtrado de información

En el programa desarrollado el sistema almacena toda la información que recibe, basándose únicamente en el nivel de confianza del interlocutor.

Para evitar el almacenamiento de información inútil, podría desarrollarse un sistema de filtrado de información que analice qué datos merecen ser almacenados y cuáles no.

### 5.2.3. Sistema de aprendizaje

En último lugar, se propone la posibilidad de desarrollar un sistema de aprendizaje que permita al programa adquirir conocimientos de manera autónoma mediante la realización de búsquedas en diferentes plataformas.

Esto evitaría la necesidad de una interacción constante con humanos para poder ampliar la base de conocimiento.



### 5.3. VALORACIÓN DE IMPACTOS Y DE ASPECTOS DE RESPONSABILIDAD LEGAL, ÉTICA Y PROFESIONAL

Aunque uno de los objetivos fundamentales de la Inteligencia Artificial es la neutralidad, en muchas ocasiones sus resultados están sesgados. Así, uno de los principales aspectos de responsabilidad ética relacionados con este trabajo es el sesgo algorítmico.

Este fenómeno tiene lugar cuando un sistema informático reproduce los valores de los seres humanos que han formado parte de los procesos de recolección de datos y entrenamiento del algoritmo.

Es decir, no es el algoritmo inicial lo que está sesgado, sino los datos, pudiendo introducir prejuicios culturales y sociales camuflados en el lenguaje humano. De este modo, resulta de vital importancia controlar el contexto en el que se utiliza el algoritmo y los usuarios que pueden interactuar con él.

Puesto que el sesgo algorítmico puede agravar las discriminaciones existentes en la actualidad, es importante trabajar en el diseño de herramientas que detecten los sesgos que pueda presentar la información utilizada en el proceso de entrenamiento del algoritmo; así como la formación en valores de los ingenieros y todo el personal involucrado en el desarrollo de un sistema de Inteligencia Artificial.

## 5.4. PLANIFICACIÓN TEMPORAL

La planificación temporal de este Trabajo Fin de Grado se va a representar mediante una Estructura de Descomposición del Trabajo (EDT) y un Diagrama de Gantt.

### 5.4.1. Estructura de Descomposición del Trabajo (EDT)



Figura 77: Estructura de Descomposición del Trabajo (EDT).

## 5.4.2. Diagrama de Gantt

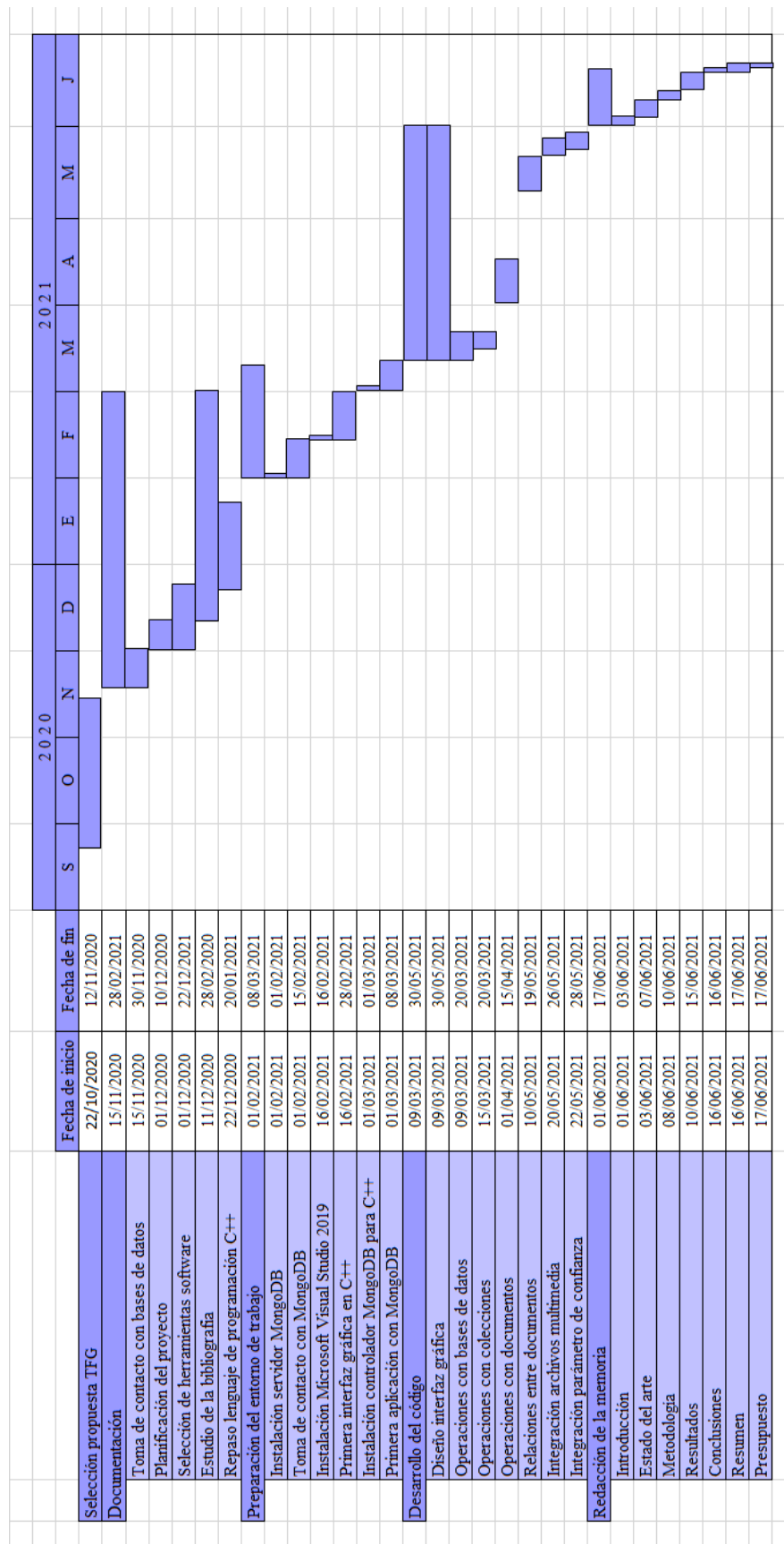


Figura 78: Diagrama de Gantt.

## 5.5. PRESUPUESTO

Para calcular el presupuesto utilizado durante el desarrollo de este proyecto se debe tener en cuenta tanto la materia prima como la mano de obra.

### 5.5.1. Materia prima

	Cantidad (unidades)	Coste unitario (€ / unidad)	Coste total (€)
Ordenador portátil HP	1	600	600
Disco duro Toshiba	1	70	70
Microsoft Office 365	1	69	69
Visual Studio 2019	1	0	0
Servidor MongoDB	1	0	0
			739

Tabla 18: Presupuesto materia prima

### 5.5.2. Mano de obra

	Cantidad (horas)	Coste unitario (€ / hora)	Coste total (€)
Profesor universitario	40	30	1200
Becaria de ingeniería industrial	360	12	4320
			5520

Tabla 19: Presupuesto mano de obra

### 5.5.3. Presupuesto total

	Coste total (€)
Materia prima	739
Mano de obra	5520
	6259

Tabla 20: Presupuesto total

## BIBLIOGRAFÍA

- [1] IONOS, «Linux vs. Windows: análisis comparativo,» *Digital Guide IONOS*, 2019.
- [2] A. J. Pérez Vidal, A. Castro-González, F. Alonso Martín, J. C. Castillo y M. A. Salichs, «Evolución de la robótica social y nuevas tendencias,» de *Actas de las XXXVIII Jornadas de Automática*, Gijón, 2017.
- [3] International Federation of Robotics, «World Robotics Service Robotics,» 2020.
- [4] S. Moriello, «Robots sociales, la nueva generación,» *Tendencias 21*, 2008.
- [5] C. Balkenius, «From Isolated Components to Cognitive Systems,» *ERCIM News*, nº 53, 2003.
- [6] H. Levesque y G. Lakemeyer, «Chapter 23 Cognitive Robotics,» de *Foundations of Artificial Intelligence*, 2008, pp. 869-886.
- [7] S. Manzoor, Y. G. Rocha, S. H. Joo, S. H. Bae, E. J. Kim, K. J. Joo y T. Y. Kuc, «Ontology-Based Knowledge Representation in Robotic Systems: A Survey Oriented toward Applications,» *Applied Sciences*, 2021.
- [8] J. I. Olszewska, M. Barreto, J. Bermejo-Alonso, J. Carbonera, A. Chibani, S. Fiorini, P. Goncalves, M. Habib, A. Khamis, A. Olivares, E. P. de Freitas, E. Prestes, S. V. Ragavan, S. Redfield, R. Sanz, B. Spencer y H. Li, «Ontology for Autonomous Robotics,» de *26th IEEE International Symposium on Robot and Human Interactive Communication*, Lisboa, 2017.
- [9] Universidad Politécnica de Madrid , «Centre for Automation and Robotics,» [En línea]. Available: <https://www.car.upm-csic.es/>. [Último acceso: marzo 2021].
- [10] C. Torras, «Robots sociales: un punto de encuentro entre ciencia y ficción,» *MÈTODE Science Studies Journal*, 2014.
- [11] «MIT Computer Science & Artificial Intelligence Lab,» [En línea]. Available: <http://www.ai.mit.edu/projects/sociable/baby-bits.html>. [Último acceso: marzo 2021].
- [12] S. Domínguez, E. Zalama y J. G. García-Bermejo, «Arisco, un robot social con capacidad de interacción, motivación y aprendizaje,» *Revista Iberoamericana de Automática e Informática Industrial*, vol. 5, nº 2, 2008.
- [13] «AliveRobots.com by Robotrónica,» [En línea]. Available: <https://aliverobots.com/robot-pepper/>. [Último acceso: marzo 2021].
- [14] S. Coradeschi, S. C. Shapiro, M. Asada y C. Breazeal, «Human-Inspired Robots,» *IEE Intelligent Systems*, 2006.

- [15] D. Lala, G. Wilcock, K. Jokinen y T. Kawahara, «ERICA and WikiTalk,» de *28th International Joint Conference on Artificial Intelligence*, Macao, 2019.
- [16] K. Puerto, «Erica es el último robot japonés con aspecto humano, nos imita conversando,» *Xakata*, 2015.
- [17] R. Álvarez, «El impresionante robot con inteligencia artificial 'Erica' será el primero en el mundo en protagonizar una película de ciencia ficción,» *Xakata*, 2020.
- [18] «Hanson Robotics,» [En línea]. Available: <https://www.hansonrobotics.com/sophia/>. [Último acceso: marzo 2021].
- [19] «Tokyo the Robot,» [En línea]. Available: <https://tokyotherobot.com/>. [Último acceso: marzo 2021].
- [20] Equipo GNOSS, «YAGO: Base de conocimiento semántica,» *Next Web*, 2018.
- [21] Tufts University, «Tufts.edu,» [En línea]. Available: <https://access.tufts.edu/tusk>. [Último acceso: marzo 2021].
- [22] I. J. Chang, «Análisis y diseño de sistemas cognitivos para robots autónomos,» 2009.
- [23] K. Irusta, «Aprendizaje en robots sociales,» 2017.
- [24] K. Irusta, «Adquisición de conocimiento de un robot social,» 2019.
- [25] A. Barrientos, L. F. Peñín, C. Balaguer y R. Aracil, *Fundamentos de robótica*, Madrid: Mc Graw Hill, 2007.
- [26] Esneca Business School, «Clasificación de los robots según su función,» 2018.
- [27] T. Christaller, «Cognitive Robotics: a new approach to artificial intelligence,» 1998.
- [28] H. Levesque y G. Lakemeyer, «Cognitive Robotics,» de *Knowledge Representation*, 2008.
- [29] R. Arrabales, «Robótica cognitiva,» *Conscious Robots*, 2007.
- [30] S. Mirete, «Impulso a la inteligencia: la era cognitiva,» *KPMG Tendencias*.
- [31] C. Angulo, Interviewee, *Robótica cognitiva: mitos y realidades en la colaboración humano-robot*. [Entrevista]. 2020.
- [32] Wikipedia, «Robótica cognitiva,» [En línea]. Available: [https://es.wikipedia.org/wiki/Rob%C3%B3tica\\_cognitiva](https://es.wikipedia.org/wiki/Rob%C3%B3tica_cognitiva). [Último acceso: abril 2021].
- [33] A. Serna, E. Acevedo y E. Serna, «Principios de la Inteligencia Artificial en las Ciencias Computacionales,» de *Desarrollo e innovación en ingeniería*, 2017.

- [34] S. Marín, «Ética e inteligencia artificial,» *Cátedra CaixaBank de Responsabilidad Social Corporativa*, 2019.
- [35] J. N. Kok, P. van der Putten, W. A. Kusters y E. J. Boers, «Artificial Intelligence: definition, trends, techniques and cases».
- [36] L. Rouhiainen, *Inteligencia Artificial: 101 cosas que debes saber hoy sobre nuestro futuro*, Planeta S. A., 2018.
- [37] International Federation of Robotics, «Artificial Intelligence in Robotics,» 2018.
- [38] L. E. Munera, «Inteligencia Artificial y sistemas expertos».
- [39] B. G. Buchanan, «A brief history of Artificial Intelligence,» *AI Magazine* , vol. 26, nº 4, 2006.
- [40] View Next, «Aplicaciones de la Inteligencia Artificial en la actualidad,» 2019.
- [41] C. Vázquez, «Cinco aplicaciones de Inteligencia Artificial que utilizas cada día,» *Avansis*, 2019.
- [42] C. F. Reyes, «Análisis de la relación entre la ingeniería del conocimiento y la gestión del conocimiento en base al modelo de Nonaka y Takeuchi,» *Intangible Capital*, vol. 1, nº 9, 2005.
- [43] «Ingeniería del conocimiento,» [En línea]. Available: [https://es.wikipedia.org/wiki/Ingenier%C3%ADa\\_del\\_conocimiento](https://es.wikipedia.org/wiki/Ingenier%C3%ADa_del_conocimiento).
- [44] «Introducción a la ingeniería del conocimiento,» [En línea]. Available: <https://sites.google.com/site/sistemasexperto/introduccion-a-la-ingenieria-del-conocimiento>.
- [45] R. Studer, V. R. Benjamins y D. Fensel, «Knowledge engineering: principles and methods,» *Data and knowledge engineering*, 1998.
- [46] J. Duran, J. Conesa y R. Clarisó, «Introducción a la representación del conocimiento».
- [47] R. E. Ardila, «El conocimiento en Inteligencia Artificial».
- [48] Microsoft, [En línea]. Available: <https://visualstudio.microsoft.com/es/>.
- [49] MongoDB, [En línea]. Available: <https://www.mongodb.com/>.
- [50] D. Rubio, «Modelo ACID y teorema CAP,» *Pandemonio Digital*, 2021.
- [51] A. G. Babucea, «SQL or NoSQL databases? Critical differences,» 2021.
- [52] V. Valverde, N. Portalanza y P. Mora, «Análisis descriptivo de base de datos relacional y no relacional,» *Atlante*, 2019.

- [53] W. Ali, M. U. Shafique, M. A. Majeed y A. Raza, «Comparison between SQL and NoSQL databases and their relationship with Big Data analytics,» *Asian Journal of Research in Computer Science*, 2019.
- [54] C. Strauch, «NoSQL databases».



# ÍNDICE DE FIGURAS

Figura 1: Kismet. MIT. 1998. ....	12
Figura 2: Arquitectura neuronal de Arisco. Universidad de Valladolid. 2008. ....	12
Figura 3: Pepper. SoftBank Robotics. 2014. ....	13
Figura 4: Erica. Hiroshi Ishiguro. 2015. ....	13
Figura 5: Sophia. Hanson Robotics. 2015. ....	14
Figura 6: Tokyo the Robot. Grupo ADD. 2019. ....	14
Figura 7: Watson en el concurso Jeopardy. ....	20
Figura 8: Procesos de la ingeniería del conocimiento. ....	24
Figura 9: Tipos de conocimiento. ....	24
Figura 10: Clasificación de las bases de datos según el teorema CAP. ....	29
Figura 11: Instalación Microsoft Visual Studio 2019. Paso 1. ....	32
Figura 12: Instalación Microsoft Visual Studio 2019. Paso 2. ....	32
Figura 13: Instalación Microsoft Visual Studio 2019. Paso 3. ....	33
Figura 14: Instalación Microsoft Visual Studio 2019. Paso 4. ....	33
Figura 15: Instalación API Formularios de Windows para C++. Paso 1. ....	33
Figura 16: Instalación API Formularios de Windows para C++. Paso 2. ....	34
Figura 17: Instalación API Formularios de Windows para C++. Paso 3. ....	34
Figura 18: Instalación servidor MongoDB. Paso 1. ....	35
Figura 19: Instalación servidor MongoDB. Paso 2. ....	35
Figura 20: Instalación servidor MongoDB. Paso 3. ....	35
Figura 21: Instalación MongoDB Driver C++. Paso 1. ....	36
Figura 22: Instalación MongoDB Driver C++. Paso 2. ....	36
Figura 23: Instalación MongoDB Driver C++. Paso 3. ....	37
Figura 24: Instalación MongoDB Driver C++. Paso 4. ....	37
Figura 25: Instalación MongoDB Driver C++. Instrucción 1. ....	37
Figura 26: Instalación MongoDB Driver C++. Instrucción 2. ....	37
Figura 27: Instalación MongoDB Driver C++. Instrucción 3. ....	37
Figura 28: Preparación Visual Studio 2019. Paso 1. ....	38
Figura 29: Preparación Visual Studio 2019. Paso 2. ....	38
Figura 30: Preparación Visual Studio 2019. Paso 3. ....	39
Figura 31: Estructura sistema de conocimiento. ....	40
Figura 32: Función CrearBaseDeDatos( ). Diagrama de flujo. ....	42
Figura 33: Función EliminarBaseDeDatosExistente( ). Diagrama de flujo. ....	42
Figura 34: Función CrearColeccion( ). Diagrama de flujo. ....	43
Figura 35: Función EliminarColeccionExistente( ). Diagrama de flujo. ....	44
Figura 36: Función CrearDocumento( ). Diagrama de flujo. ....	45
Figura 37: Función EliminarTema( ). Diagrama de flujo. ....	46
Figura 38: Función BuscarDocumentosColeccion( ). Diagrama de flujo. ....	47
Figura 39: Función BuscarDocumentoPorID( ). Diagrama de flujo. ....	48
Figura 40: Función BuscarDocumento( ). Diagrama de flujo. ....	49
Figura 41: Función BuscarDocumentoPorAtributos( ). Diagrama de flujo. ....	50
Figura 42: Función AñadirAtributoADocumento( ). Diagrama de flujo. ....	51
Figura 43: Función ModificarAtributoDocumento( ). Diagrama de flujo. ....	52

Figura 44: Función EliminarAtributoDocumento( ). Diagrama de flujo.....	53
Figura 45: Función EstablecerRelacion( ). Diagrama de flujo. ....	54
Figura 47: Función AñadirArchivoMultimedia( ). Diagrama de flujo.....	56
Figura 48: Función EliminarArchivoMultimedia( ). Diagrama de flujo.....	57
Figura 49: Función ConsultarArchivoMultimedia( ). Diagrama de flujo.....	58
Figura 50: Función ConsultarArchivoMultimediaPorNombre( ). Diagrama de flujo.....	59
Figura 51: Función EstablecerRelaciónMultimedia( ). Diagrama de flujo. ....	60
Figura 52: Panel de inicio. ....	61
Figura 53: Panel lista bases de datos y colecciones. ....	62
Figura 54: Panel lista de bases de datos y colecciones. Ejemplo. ....	62
Figura 55: Panel añadir base de datos. ....	63
Figura 56: Panel eliminar base de datos. ....	64
Figura 57: Panel añadir colección. ....	64
Figura 58: Panel eliminar colección. ....	65
Figura 59: Panel consultar documento. ....	66
Figura 60: Consultar documento por colección. ....	66
Figura 61: Consultar documento por nombre. ....	67
Figura 62: Consultar documento por ID.....	68
Figura 63: Consultar documento por atributos. ....	68
Figura 64: Panel añadir documento.....	69
Figura 65: Panel eliminar documento.....	69
Figura 66: Panel modificar documento.....	70
Figura 67: Panel establecer relación entre documentos.....	71
Figura 68: Panel establecer relación. Ejemplo. ....	71
Figura 69: Panel preguntar curiosidad. ....	72
Figura 70: Panel archivos multimedia. ....	72
Figura 71: Panel añadir archivo multimedia. ....	73
Figura 72: Panel eliminar archivo multimedia.....	73
Figura 73: Panel consultar archivo multimedia. ....	74
Figura 74: Panel establecer relación entre documento y archivo multimedia .....	75
Figura 75: Panel establecer relación entre documento y archivo multimedia .....	75
Figura 76: Paneles interfaz gráfica. ....	77
Figura 77: Estructura de Descomposición del Trabajo (EDT). ....	81
Figura 78: Diagrama de Gantt.....	82

## ÍNDICE DE TABLAS

Tabla 1: Clasificación de los robots atendiendo a su cronología. ....	18
Tabla 2: Aplicaciones de los robots de servicio. ....	19
Tabla 3: Cuadro comparativo modelos de bases de datos relacional y no relacional. ....	27
Tabla 4: Cuadro comparativo tipos de bases de datos no relacionales. ....	30
Tabla 5: Función ListaBasesDeDatosDisponibles( ). Parámetros y funcionalidad. ....	41
Tabla 6: Función ListaColeccionesDisponibles( ). Parámetros y funcionalidad. ....	41
Tabla 7: Función ListaDocumentosDisponibles( ). Parámetros y funcionalidad. ....	41
Tabla 8: Función ListaArchivosMultimediaDisponibles( ). Parámetros y funcionalidad. ....	41
Tabla 9: Función ListaAtributosDisponibles( ). Parámetros y funcionalidad. ....	41
Tabla 10: Nombres paneles y botones de Panel de inicio. ....	61
Tabla 11: Panel lista bases de datos y colecciones. Botones. ....	63
Tabla 12: Panel archivos multimedia. Botones. ....	72
Tabla 13: Objetivos y resultados. ....	76
Tabla 14: Funciones y botones bases de datos. ....	78
Tabla 15: Funciones y botones colecciones. ....	78
Tabla 16: Funciones y botones documentos. ....	78
Tabla 17: Funciones y botones archivos multimedia. ....	78
Tabla 18: Presupuesto materia prima ....	83
Tabla 19: Presupuesto mano de obra. ....	83
Tabla 20: Presupuesto total. ....	83
Tabla 21: Abreviaturas, unidades y acrónimos. ....	91

## ABREVIATURAS, UNIDADES Y ACRÓNIMOS

ACID	Atomicity, Consistency, Isolation, Durability Atomicidad, Consistencia, Aislamiento, Durabilidad
BSON	Binary JSON JSON binario
CAP	Consistency, Availability, Partition Tolerance Consistencia, Disponibilidad, Tolerancia a la partición
CAR	Centro de Automática y Robótica
CRUD	Create, Read, Update, Delete Crear, Leer, Actualizar, Eliminar
ETSII	Escuela Técnica Superior de Ingenieros Industriales
IA	Inteligencia Artificial
IFR	International Federation of Robotics Federación Internacional de Robótica
JSON	JavaScript Object Notation
MIT	Massachusetts Institute of Technology Instituto de Tecnología de Massachusetts
NoSQL	Not only SQL No solo SQL
SQL	Structured Query Language Lenguaje de Consulta Estructurado
TFG	Trabajo Fin de Grado
UPM	Universidad Politécnica de Madrid
YAGO	Yet Another Great Ontology

Tabla 21: Abreviaturas, unidades y acrónimos.

## GLOSARIO

**Atributo** → Cada una de las características que se desean almacenar en un documento.

**Base de conocimiento** → Base de datos destinada a la gestión del conocimiento.

**Base de datos** → Entidad software capaz de almacenar un conjunto de información estructurada que puede ser consultada de forma relativamente sencilla.

**Clave** → Nombre del atributo.

**Colección** → Cada uno de los grupos de documentos en los que se divide una base de datos en función a característica comunes.

**Diagrama de flujo** → Representación gráfica de un algoritmo que muestra, mediante formas y flechas, los pasos a seguir para solucionar un problema.

**Documento** → Conjunto ordenado de claves con su correspondiente valor.

**Documento BSON** → Representación binaria de un documento de tipo JSON, de manera que pueda ser interpretado por un ordenador.

**Documento JSON** → Documento escrito en formato JSON, de tal forma que está constituido por pares nombre-valor.

**Documento GridFS** → Documento de formato GridFS que permite almacenar más de 16M de información, ideal para archivos multimedia.

**Par clave-valor** → Conjunto que recoge el nombre y el valor de un atributo.

**Valor** → Valor asociado a un atributo.