



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# Diseño e implementación de una base de datos para el estudio descriptivo de patologías oculares

Trabajo fin de grado

**Grado en Ingeniería Informática**

**Autor:** Daniel García Molero

**Tutora:** Laura Mota Herranz

Curso 2019-2020



## **Resumen**

El objetivo de este proyecto es el diseño y posterior implementación de una aplicación para la gestión de información de pacientes con patologías oculares para su posterior explotación estadística principalmente.

La aplicación se desarrollará utilizando Microsoft Office Access ya que es el software elegido por los usuarios finales de la aplicación.

El diseño de la aplicación se realizará desde cero tanto en los aspectos estáticos como dinámicos. En la parte estática, se desarrollarán todas sus fases, desde la especificación de requisitos de datos, pasando por la creación del diagrama de clases a partir de la fase anterior, hasta la implementación de la base de datos final. En la parte dinámica, además de los procesos relativos a la base de datos propiamente dicha, también se diseñarán los formularios e informes necesarios para satisfacer la especificación de requisitos de procesos, así como las consultas a la base de datos necesarias para la explotación de dichos formularios e informes.

Debido a que en el presente estudio se van a manejar datos personales de pacientes de carácter sensible, se realizará un análisis del marco legal para tener en cuenta la normativa vigente en relación al tratamiento de dichos datos en todas las fases del desarrollo.

## **Abstract**

The aim of this Project is the development and later implementation of a management application of patients with ocular pathologies to their following mainly statistics use.

The application will be developed using Microsoft Office Access since is the software chosen by the application final users.

The design of the application will be done from scratch in both static and dynamic aspects. In the static part, all its phases will be developed, from the specification of data requirements, through the creation of the class diagram from the previous phase, to the implementation of the final database. In the dynamic part, in addition to the processes related to the database itself, the necessary forms and reports will also be designed to satisfy the process requirements specification, as well as the queries to the database necessary for the exploitation of said processes forms and reports.

Due to the fact that in the present study, personal data of sensitive patients will be handled, an analysis of the legal framework will be carried out to take into account the current regulations regarding the treatment of such data at all stages of development.

## Índice

1	Introducción.....	1
1.1	Motivación .....	1
1.2	Objetivos .....	1
1.3	Metodología.....	2
1.4	Estructura del documento.....	2
2	Contexto Tecnológico.....	4
2.1	Bases de datos.....	4
2.1.1	Tablas .....	8
2.1.2	Relaciones .....	12
2.1.3	Consultas.....	13
2.1.4	Formularios.....	16
2.2	UML.....	17
2.2.1	Diagrama de Clases .....	17
2.2.2	Propuesta de proyecto.....	19
3	Análisis del problema .....	20
3.1	Análisis de Seguridad.....	20
3.2	Análisis del marco legal.....	21
3.2.1	Protección de datos .....	21
3.3	Análisis de riesgos.....	23
3.4	Posibles soluciones .....	23
3.5	Solución propuesta.....	24
3.6	Plan de trabajo .....	24
3.7	Planificación y presupuesto.....	25
4	Diseño de la solución .....	27
4.1	Diagrama UML .....	27
4.2	Esquema lógico de la base de datos.....	29
4.3	Diseño de consultas .....	33
4.4	Diseño de la interfaz gráfica de usuario .....	34
5	Implementación .....	36

5.1 Tablas.....	36
5.2 Relaciones.....	40
5.3 Consultas .....	42
5.4 Formularios .....	48
6 Implantación .....	60
7 Pruebas .....	64
8 Conclusiones .....	68
8.1 Relación del proyecto con los estudios cursados .....	69
8.2 Ventajas obtenidas .....	70
8.3 Desarrollos futuros .....	71
9 Referencias .....	72
10 Glosario: .....	73

## Índice de figuras

Figura 1. Arquitectura de tres niveles de una base de datos. ....	7
Figura 2. Ejemplo de relación uno a uno. ....	12
Figura 3. Ejemplo de relación uno a muchos. ....	13
Figura 4. Ejemplo de relación muchos a muchos. ....	13
Figura 5. Ejemplo de clase UML. ....	18
Figura 6. Diagrama de Gantt .....	25
Figura 7. Diagrama de clases UML del proyecto .....	28
Figura 8. Esquema de tablas y relaciones del proyecto.....	32
Figura 9. Creación de una tabla. Menú Tabla.....	36
Figura 10. Creación de una tabla. Menú Diseño de tabla .....	37
Figura 11. Intercambiar entre tipos de vista de una tabla .....	37
Figura 12. Propiedades de tipo Autonumeración .....	38
Figura 13. Propiedades de tipo Número .....	39
Figura 14. Propiedades de tipo Texto corto .....	39
Figura 15. Propiedades de tipo Sí/No .....	40
Figura 16. Propiedades de tipo Fecha/Hora .....	40
Figura 17. Agregar tablas .....	41
Figura 18. Configuración de la relación. ....	42
Figura 19. Consulta buscar pruebas .....	43
Figura 20. Generador de expresiones .....	44
Figura 21. Consulta buscar pruebas.....	44
Figura 22. Diseño Consulta enfermedades 1 .....	45
Figura 23. Diseño Consulta enfermedades 2.....	45
Figura 24. Tablas de consulta sin relación.....	45
Figura 25. Diseño consulta intervenciones .....	46
Figura 26. Diseño consulta seguimientos .....	47
Figura 27. Diseño Sufre Consulta.....	47
Figura 28. Criterios consulta informe .....	48
Figura 29. Asistente formularios .....	49
Figura 30. Distribución de datos en formulario .....	49
Figura 31. Último paso asistente formularios.....	50

Figura 32. Formulario Principal.....	50
Figura 33. Formulario Gestión de informes.....	51
Figura 34. Formulario Enfermedad .....	51
Figura 35. Opción botón .....	52
Figura 36. Asistente botones .....	52
Figura 37. Formulario Especialidad .....	53
Figura 38. Formulario Pruebas .....	53
Figura 39. Subformulario Foto .....	53
Figura 40. Agregar imagen .....	54
Figura 41. Subformulario Seguimientos.....	55
Figura 42. Subformulario Sufre.....	55
Figura 43. Subformulario P_Realizadas .....	56
Figura 44. Subformulario intervención .....	56
Figura 45. Sub-subformulario observación .....	57
Figura 46. Sub-subformulario quirúrgico.....	57
Figura 47. Sub-subformulario Tratamiento medico.....	57
Figura 48. Formulario Paciente.....	58
Figura 49. Código edición campos .....	59
Figura 50. Código buscar nombre enfermedad .....	59
Figura 51. Apertura en modo exclusivo .....	60
Figura 52. Opción cifrar base de datos .....	60
Figura 53. Configuración aplicación final .....	61
Figura 54. Configuración copia seguridad .....	62
Figura 55. Opciones copia de seguridad .....	63
Figura 56. Datos de paciente.....	64
Figura 57. Fotografías.....	64
Figura 58. Datos de seguimientos .....	65
Figura 59. Datos de enfermedades .....	65
Figura 60. Datos de pruebas médicas .....	66
Figura 61. Datos de intervenciones .....	66
Figura 62. Buscar paciente .....	67

## Índice de tablas

Tabla 1. Tipos de datos en MS Access.....	9
Tabla 2. Combinaciones de “Requerido” y “Permitir longitud cero” .....	10
Tabla 3. Presupuesto del proyecto .....	26



## **1 Introducción**

En este primer capítulo se expone el porqué de este proyecto, es decir, qué necesidad surge y cómo se pretende darle solución paso a paso detallando cada uno de ellos. Asimismo, se establece una planificación temporal del trabajo a realizar y se hace un cálculo del coste económico que conllevaría la realización de este proyecto por un profesional.

### **1.1 Motivación**

Hasta este momento, los estudios estadísticos en el servicio de Oftalmología del Hospital Provincial de Castellón se llevaban a cabo mediante las hojas de cálculo, generalmente usando el Microsoft Excel de la suite Microsoft Office, ya que los datos que se necesitaba explotar no tenían gran complejidad y no requiere conocimientos amplios acerca de su manejo básico. Es por esto por lo que eran los mismos usuarios que realizaban los estudios los que se encargaban de confeccionar la estructura de datos que les permitía llevar a cabo dichos estudios.

Cuando surgió la necesidad de aunar en un mismo estudio diferentes especialidades médicas, se percataron de que el software utilizado hasta el momento tenía limitaciones y no podían almacenar tan fácilmente toda la información que necesitaban sin tener que utilizar distintas hojas de cálculo. Es por ello por lo que se decidió utilizar un software más específico para el almacenamiento y explotación de datos.

La creación de una base de datos utilizando Microsoft Access creo que es una solución sencilla y efectiva para esta problemática, además de ya disponer de este software al estar incluido en la suite Microsoft Office.

### **1.2 Objetivos**

El objetivo de este proyecto es la creación de una aplicación de bases de datos para el almacenamiento de información relativa a pacientes de enfermedades oculares para poder explotar dichos datos posteriormente y realizar estudios acerca de los diversos tratamientos y técnicas quirúrgicas.

La principal meta es unificar sistemas de información en lo relativo a estudios de patologías oftalmológicas en el Servicio de Oftalmología, así como simplificar su uso y aprovechar al máximo las posibilidades que ofrecen las bases de datos.

A pesar de que, una vez creada la base de datos, se pueda utilizar para realizar estudios sobre diversas enfermedades, uno de los propósitos es que no requiera ni mucho tiempo ni mucho esfuerzo para su manejo, es decir que tenga un aprendizaje lo más rápido y sencillo posible. Ya que el mismo proyecto se podrá utilizar para llevar a cabo estudios sobre diferentes enfermedades oculares, se podría optar por un proceso de aprendizaje de su manejo algo más complicado y que requiriera de más tiempo, pero uno de los objetivos es que se pueda comenzar a utilizar prácticamente al inicio de su utilización.

### **1.3 Metodología**

En lo relativo a la metodología se van a llevar a cabo las siguientes fases:

- Identificación del trabajo, necesidades, restricciones y limitaciones que existen en el entorno en el que se va a utilizar la base de datos.
- Diseño de la base de datos, primero en UML y posteriormente en Microsoft Access partiendo del diseño UML realizado.
- Implementación de la base de datos y de la aplicación que accederá a ella a partir de la fase anterior.
- Realización de pruebas para comprobar que la base de datos satisface las necesidades detectadas en la primera fase, además de la comprobación de que se han tenido en cuenta las restricciones y limitaciones.
- Redacción de la memoria.

### **1.4 Estructura del documento**

Este documento se divide en nueve capítulos:

En este primer capítulo se detalla el origen de este proyecto, cómo surge la necesidad de llevarlo a cabo. También se mencionan los objetivos que se tienen en cuenta en el desarrollo del proyecto y la metodología utilizada para su realización.

En el segundo capítulo se plasman los fundamentos teóricos necesarios para realizar este proyecto, tanto generales de las bases de datos relacionales, como más específicos de la herramienta que se va a utilizar para su desarrollo, Microsoft Access.

En el tercer capítulo se lleva a cabo un análisis del problema que supone la realización de este proyecto. Dentro de este apartado se enfocan aspectos como el de la seguridad de los datos almacenados, la normativa a tener en cuenta en relación al tratamiento de datos personales, posibles riesgos que pueden surgir durante el desarrollo del proyecto con sus soluciones respectivas. Por último, dentro de este capítulo se detallará un plan de trabajo para poder realizar una posterior planificación temporal y presupuestaria.

En el cuarto capítulo se muestra el diseño, en todas sus etapas, de la base de datos en su conjunto, incluyendo las tablas, relaciones entre ellas, consultas e interfaz gráfica de usuario.

En el quinto capítulo se podrá observar el proceso de la implementación de la base de datos partiendo del diseño realizado en el capítulo anterior.

En el sexto capítulo se detallará el procedimiento llevado a cabo para la implementación de la base de datos en el sistema en el que se va a utilizar finalmente.

La séptima parte de este documento explicará las pruebas realizadas en el sistema ya implantado para comprobar el correcto funcionamiento y la satisfacción de las necesidades detectadas al inicio.

El octavo capítulo corresponde a las conclusiones en el que se analizan los puntos fuertes y débiles del proyecto y se listan posibles ampliaciones o modificaciones para reforzar los puntos débiles, así como la relación del proyecto realizado con los estudios cursados.

En el noveno y último capítulo se podrán ver las referencias utilizadas para la obtención de la documentación necesaria para la consecución de este documento.

## 2 Contexto Tecnológico

La principal y única herramienta que se utilizan en este proyecto es Microsoft Office Access, ya que no se parte de ningún tipo de sistema de información previo. Se ha elegido esta herramienta al ser la más adecuada de entre las que se dispone, ya que las funcionalidades que ofrece son suficientes para la ejecución del proyecto y no es necesaria la adquisición de otra herramienta de Sistema de Gestión de Base de Datos que aporte nuevas funcionalidades.

### 2.1 Bases de datos<sup>1</sup>

Una base de datos es una colección estructurada de información que debe reflejar fielmente la parte del mundo real que representa, debido a esto, la base de datos debe ser sensible a los sucesos del mundo real y reflejar dichos cambios.

Las empresas y organizaciones las utilizan como sistemas de información porque son capaces de dar servicio a distintos usuarios con diferentes necesidades, a la vez que gestionan gran cantidad de información y aseguran la persistencia de dicha información, es decir, que no se va a perder una vez se ha introducido en el sistema.

Para estructurar las bases de datos se necesita una herramienta que también permita crearlas y manipularlas. Estas herramientas son lo que se conoce como Sistemas de Gestión de Bases de Datos (SGBD).

Existen diferentes familias de SGBD (jerárquicos, en red, relacionales, orientadas a objetos), cada uno con un esquema lógico diferente. En este caso se va a basar la base de datos en el modelo relacional.

En los SGBD relacionales, el esquema lógico de una base de datos se estructura en un conjunto de relaciones, la cuales se pueden ver en forma de tablas, que guardan de manera estructurada los datos necesarios para satisfacer

---

<sup>1</sup> Apuntes de bases de datos y tecnologías de la información de la UPV (Celma, M.; Casamayor, J. C.; Mota, L.; Bases de datos relacionales. Pearson, Prentice Hall, 2003)

las necesidades de la organización en la que esté implantada. El hecho de que estas relaciones se puedan ver en forma de tabla organizadas en filas (tuplas) y columnas (atributos) hace que este modelo sea sencillo, de forma que las filas de una tabla determinada tienen una estructura similar y guardan datos semejantes de objetos del mundo real. Asimismo, cada columna de estas tablas almacena una propiedad de estos objetos. Cabe añadir también que los posibles valores de un atributo en cada tupla de una relación ha de ser del mismo tipo de datos.

Existe la posibilidad de que en la realidad no se disponga de información acerca de un atributo en concreto por lo que no se puede guardar ninguna información en el citado atributo. Este hecho hace que sea necesario introducir el concepto de “valor nulo”.

Además de las características ya mencionadas, también hay que destacar las siguientes:

- Una base de datos puede integrar toda la información de una empresa y dada esta circunstancia, es necesario que los distintos usuarios que accedan a ella lo puedan hacer de forma simultánea.
- Se debe garantizar que la información de una base de datos no se perderá una vez guardada en ella.
- La información debe ser accesible por parte de diferentes usuarios de manera simultánea.
- Existe una independencia entre la base de datos y la implementación de los programas que acceden a ella.
- Cada usuario tendrá una vista parcial definida de la base de datos, es decir, solamente verá la información a la que se le haya permitido acceder.
- Dispone de mecanismos que garantizan la integridad y la seguridad de la información

Todas estas características se pueden resumir en que un Sistema de Gestión de Base de Datos debe garantizar la independencia, la integridad y la seguridad de los datos.

En los sistemas relacionales, el lenguaje de manipulación y definición es el SQL (*Structured Query Language*), el cual se verá con algo más de detalle en otro apartado.

En toda base de datos existe cierto número de limitaciones que aseguran que la información almacenada se represente correctamente respecto a cómo se ha diseñado dicha base de datos. A estas limitaciones se les conoce como restricciones de integridad.

En el modelo relacional existen cuatro restricciones de integridad:

- Restricción de valor no nulo: Indica que, dada esa restricción sobre uno o varios atributos, no debe haber ninguna tupla que tenga el valor nulo en esos atributos.
- Restricción de unicidad: Indica que, dada esa restricción sobre uno o varios atributos, no debe haber dos tuplas que tengan el mismo valor en esos atributos.
- Restricción de clave primaria: Esta restricción definida sobre un conjunto de atributos, sirve para identificar unívocamente cada una de sus tuplas. Implica la restricción de unicidad y a la de valor no nulo.
- Restricción de integridad referencial: También conocida como Clave Ajena, son el mecanismo que proporciona el modelo referencial para expresar asociaciones entre diferentes relaciones.

Existen otro tipo de restricciones que no proporciona el modelo relacional que sirven para expresar situaciones que afectan a uno (sencillas) o varios (general) atributos de una relación.

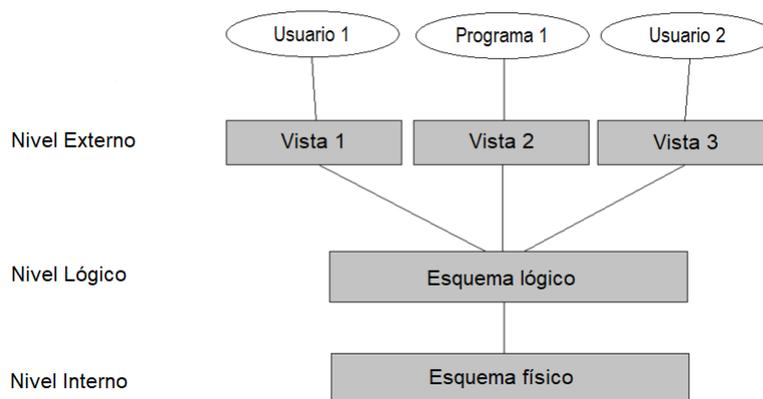
Como se ha mencionado anteriormente, un SGBD debe garantizar la seguridad, la independencia y la integridad de los datos.

En 1977, el grupo de estudio ANSI-SPARC definió un estándar por el que se diseña una base de datos en base a tres niveles de abstracción, el nivel conceptual, el nivel externo y el nivel interno.

- El nivel conceptual define la forma en que se almacenan los datos en la base de datos y las relaciones entre sí. En este nivel existe independencia respecto al SGBD utilizado.

- El nivel externo define las vistas que va a tener cada usuario de la base de datos, es decir, los datos a los que va a tener acceso y de qué manera.
- El nivel interno define la representación física de la base de datos, es decir, de qué manera se va a almacenar en el hardware del equipo.

Además de estos tres niveles, se puede establecer un nivel más, el nivel lógico, en el que se describe la base de datos en función del SGBD que se va a utilizar.



**Figura 1. Arquitectura de tres niveles de una base de datos.**

Esta división en niveles hace que cada uno de ellos sea independiente del resto, por ejemplo, el esquema lógico no depende de cómo se estén almacenados los datos en el disco duro del sistema. Aunque esta independencia de niveles desaparece en el momento en el que es necesario hacer una transformación de un nivel a otro para, por ejemplo, que un usuario haga una operación sobre la base de datos. En la actualidad esta ruptura de la independencia se realiza en cada acceso a la base de datos con lo que, por ejemplo, se pueden realizar cambios a nivel físico/interno sin repercutir en las aplicaciones que acceden a la base de datos, ya que éstas dependen de las vistas externas.

Cuando se habla de la integridad de los datos, puede verse comprometida por el hecho de que diferentes aplicaciones o usuarios acceden y realizan operaciones simultáneamente sobre la base de datos. Para garantizar dicha integridad, es necesario introducir el concepto de transacción, que se define como una secuencia de operaciones de acceso a la base de datos que constituyen una unidad lógica de ejecución.

Otro hecho por el que se puede ver afectada la integridad es por la pérdida de información debido a un fallo en el sistema. Para poner solución a este aspecto, existen las copias de seguridad y el fichero diario en el que se guarda un registro de todas las operaciones iniciadas sobre la base de datos.

En cuanto a la seguridad de una base de datos, se debe garantizar que a una determinada información solamente podrán acceder los usuarios autorizados y de la forma autorizada.

Una vez vistos los conceptos teóricos comunes a toda base de datos, se va a exponer las características propias de los ficheros del sistema Microsoft Access, indicando los valores máximos para cada uno de los atributos:

- Tamaño 2GB. Se puede evitar esta limitación creando vínculos a tablas de otras bases de datos Access
- Número total de objetos en una base de datos: 32.768
- Número de módulos: 1000 (incluyendo formularios e informes)
- Número de caracteres del nombre de un objeto: 64
- Número de caracteres de una contraseña: 14
- Número de caracteres de un nombre de usuario o de grupo: 20
- Número de usuarios simultáneos: 255

### **2.1.1 Tablas**

Como se ha visto, una base de datos almacena información de manera estructurada en forma de tablas, guardando en cada una de ellas datos semejantes sobre objetos de la realidad, siendo cada fila de la tabla la representación de cada uno de estos objetos y cada columna una característica de dicho objeto.

Cada una de estas características puede ser de distinto tipo, como, por ejemplo: de tipo texto, tipo numérico, tipo fecha...

A continuación, se puede ver una tabla con los tipos de datos que ofrece Microsoft Access, indicando para cada uno de ellos su uso y el tamaño máximo que admite.<sup>2</sup>

<b>Tipo de datos</b>	<b>Uso</b>	<b>Tamaño</b>
Texto corto	Datos alfanuméricos.	Hasta 255 caracteres.
Texto largo	Grandes cantidades de datos alfanuméricos.	Hasta 1 GB. Solo se muestran los primeros 64.000 caracteres.
Número	Datos numéricos.	1,2,3,4 o 16 bytes.
Número grande	Datos numéricos.	8 bytes.
Fecha y hora	Fechas y horas.	8 bytes.
Fecha y hora extendida	Fechas y horas	Cadena codificada de 42 bytes.
Moneda	Datos monetarios con 4 decimales de precisión.	8 bytes.
Autonumeración	Valor único generado por Microsoft Access.	4 bytes.
Sí/no	Datos booleanos. 0 para falso y -1 para verdadero.	1 byte.
Objeto OLE	Imágenes, gráficos u otros objetos de ActiveX desde otra aplicación basada en Windows.	Hasta 2 GB.
Hipervínculo	Una dirección de vínculo a un documento o archivo.	Hasta 8192 caracteres.
Datos adjuntos	Imágenes, documentos, hojas de cálculo o gráficos.	Límite el tamaño del fichero Access.
Calculado	Puede crear una expresión que usa datos de uno varios campos.	Depende del tipo de datos del resultado del cálculo (ver resto de tipo de datos).
Asistente para búsquedas	No es un tipo de datos realmente. Cuando se elige esta opción, se inicia el asistente para ayudar a definir una búsqueda simple o complejo.	Depende del tipo de datos del campo de búsqueda.

**Tabla 1. Tipos de datos en MS Access**

Una vez definido el tipo de dato de los mostrados en la Tabla 1, hay que especificar una serie de propiedades en cada uno de ellos:

- **Tamaño del campo:** Sirve para definir la longitud máxima de la información almacenada en los campos de tipo texto, numérico o autonumérico.

<sup>2</sup> Página web de soporte de Microsoft Office (<https://support.office.com/es-es/article/especificaciones-de-access-0cf3c66f-9cf2-4e32-9568-98c1025bb47c>)

- Formato: Define el formato en el que se verán las fechas, horas y textos. Estos formatos difieren para cada tipo de datos.
- Máscara de entrada: Sirve para ayudar al usuario a la introducción de datos controlando la información que éstos pueden introducir. Por ejemplo, se puede establecer una máscara para que el usuario introduzca una fecha de esta manera: \_\_/\_\_/\_\_\_\_.
- Título: Etiqueta que se muestra para el campo sobre el que se establece cuando se usa en una vista.
- Valor predeterminado: Valor que se introducirá en ese campo para los nuevos registros en el caso de que el usuario no lo introduzca.
- Regla de validación: Permite introducir una expresión que limita los posibles valores que se escriben en ese campo.
- Texto de validación: Permite definir el texto de error que aparece cuando no se satisface la regla de validación.
- Requerido: Indica si es obligatorio introducir algún valor para ese campo.
- Permitir longitud cero: Indica si se permite una cadena de texto con cero caracteres como valor para ese campo. Esta opción aparece en los campos de tipo texto, texto largo e hipervínculo.

Seguidamente se introduce una tabla resumen de las distintas combinaciones de valores en los atributos *Requerido* y *Permitir longitud cero* y las restricciones que comportan.

Requerido	Permitir longitud cero	Acción del usuario	Valor guardado
No	No	Pulsa ENTER Pulsa Barra Espaciadora Introduce cadena de longitud cero	Nulo Nulo (No se permite)
No	Sí	Pulsa ENTER Pulsa Barra Espaciadora Introduce cadena de longitud cero	Nulo Nulo Cadena de longitud cero
Sí	No	Pulsa ENTER Pulsa Barra Espaciadora Introduce cadena de longitud cero	(No se permite) (No se permite) (No se permite)
Sí	Sí	Pulsa ENTER Pulsa Barra Espaciadora Introduce cadena de longitud cero	(No se permite) Cadena de longitud cero Cadena de longitud cero

**Tabla 2. Combinaciones de “Requerido” y “Permitir longitud cero”**

- Indexado: Permite establecer un índice sobre ese campo para realizar búsquedas con mayor rapidez. Permite valores No, Sin duplicados y Con duplicados. En el caso de Sin duplicados se prohíbe que existan diferentes registros con el mismo valor en ese campo.
- Compresión Unicode: Indica si permite o no la compresión del texto para ocupar menos espacio en la base de datos.
- Modo IME: Establece el tipo de conversión de caracteres en las versiones asiáticas de Windows.
- Modo de oraciones IME: Establece la conversión de oraciones en las versiones asiáticas de Windows.
- Alineación del texto: Sirve para definir el tipo de alineación del texto para ese campo.

Asimismo, cuando se crea una tabla en Access se puede definir la clave primaria sobre uno o varios campos de ésta y, aunque no es obligatorio, es altamente recomendable ya que, como se ha visto, esta característica hace que cada registro de esa tabla se identifique unívocamente, además de permitir establecer relaciones entre sus datos y los de otra u otras tablas.

A continuación, se detallan las características propias de las tablas de las bases de datos creadas en Microsoft Access indicando los valores máximos admitidos para cada una de ellas:

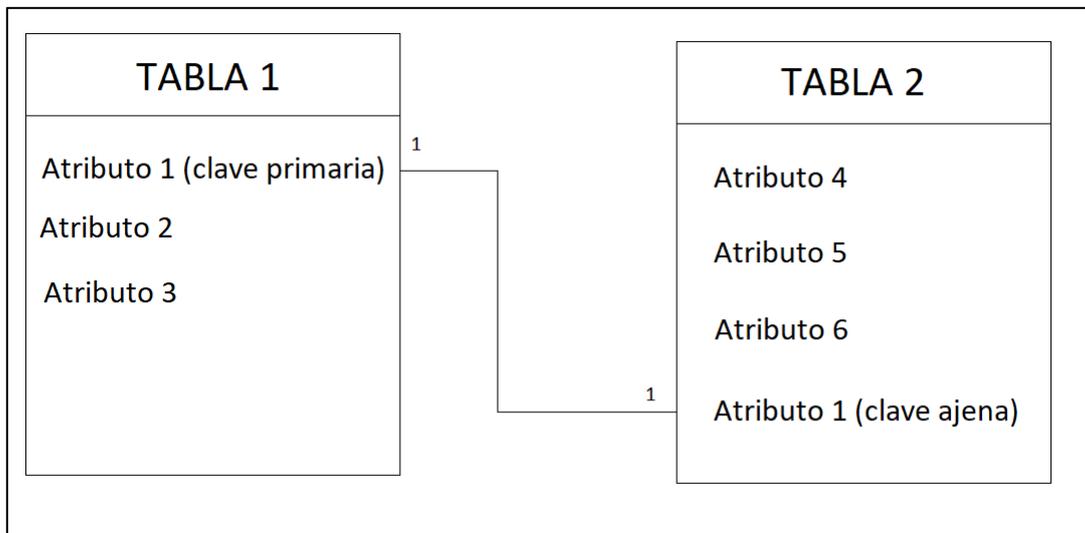
- Número de caracteres de un nombre de tabla: 64
- Número de caracteres de un nombre de campo: 64
- Número de campos en una tabla: 255
- Número de tablas abiertas: 2048
- Tamaño de la tabla: 2GB menos el espacio necesario para los objetos del sistema
- Número de índices de una tabla: 32
- Número de campos en un índice o clave principal: 10
- Número de caracteres en un mensaje de validación: 255
- Número de caracteres en una regla de validación: 2048
- Número de caracteres en una descripción de campo o tabla: 255



### 2.1.2 Relaciones

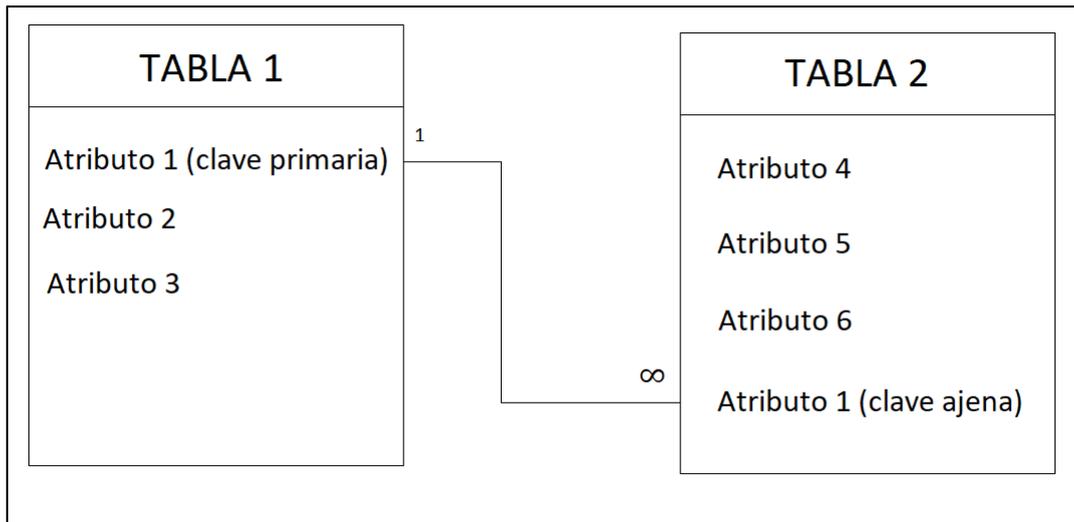
Una vez se han creado las tablas con sus atributos, es necesario indicar de qué manera se relacionan entre sí. Para ello, Microsoft Access proporciona tres maneras de relacionarlas:

Relaciones uno a uno: Este tipo de relaciones indica que cada registro de la TABLA 1 se relaciona únicamente con un registro de la TABLA 2 y viceversa. No es muy común su uso ya que esta situación se podría representar con una única tabla.



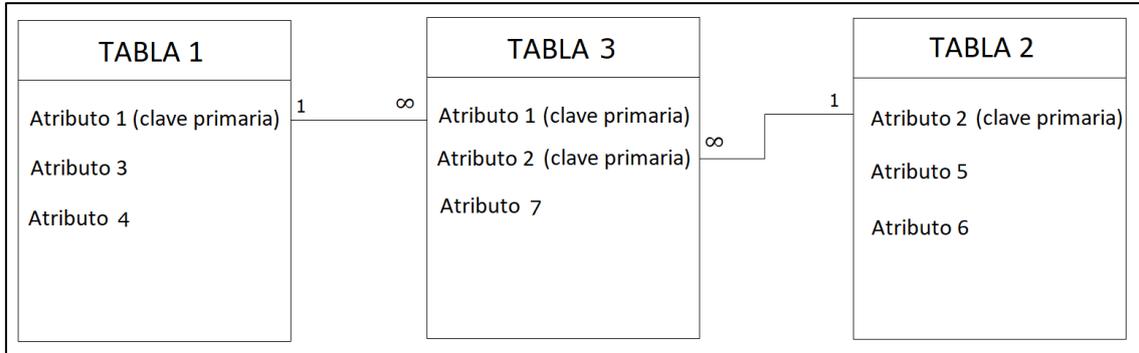
**Figura 2. Ejemplo de relación uno a uno.**

Relaciones uno a muchos: En este caso, un registro de la TABLA 1 se relaciona con muchos registros de la TABLA 2, pero un registro de la TABLA 2 solamente puede relacionarse con un registro de la TABLA 1. Este es el tipo de relaciones más común.



**Figura 3. Ejemplo de relación uno a muchos.**

Relaciones muchos a muchos: En este último tipo de relaciones, un registro de la TABLA 1 se puede relacionar con muchos registros de la TABLA 2, y viceversa. Para representar este tipo de relaciones es necesario crear una tabla adicional que tendrá como mínimo dos campos, que son la clave primaria de cada una de las otras dos tablas.



**Figura 4. Ejemplo de relación muchos a muchos.**

### 2.1.3 Consultas

En los modelos relacionales de bases de datos se utiliza el lenguaje SQL para realizar distintos tipos de operaciones sobre ellas. Se pueden identificar tres grupos:

- Lenguaje de definición de datos.
- Lenguaje de manipulación de datos, que consta, entre otras, de las siguientes instrucciones

- SELECT. Permite realizar consultas para recuperar datos de una o más relaciones de la base de datos.
  - INSERT. Permite insertar una o más tuplas en una relación de la base de datos.
  - UPDATE. Permite modificar los valores de uno o más atributos de una o más tuplas de una relación de la base de datos
  - DELETE. Permite borrar una o más tuplas de una relación de la base de datos.
- Lenguaje de control

Dentro de las operaciones de selección, existe una serie de palabras clave que permiten definir la consulta, en el ejemplo se muestra la sintaxis de éstas:

```
SELECT [all | distinct] {expresion1, expresion2, ...expresionn} | *
FROM tabla
[WHERE expresión_condicional]
[ORDER BY {columna1, columna2, ... columnam}]
[GROUP BY columnai]
[HAVING expresión_condicional_de_función_aritmética]
```

- ALL. Sirve para permitir que aparezcan filas idénticas (Es el valor por defecto).
- DISTINCT. Al contrario que "All", no permite la aparición de filas idénticas en el resultado de la consulta.
- WHERE (expresión\_condicional). Indica la condición que deben cumplir las filas resultado de la ejecución de la consulta. La expresión condicional puede estar formada por una serie de predicados:
  - De comparación. =, <, >, <>, >=, <=.
  - LIKE. Sirve para comparar una cadena de caracteres con otra.
  - BETWEEN. Compara si un número esta entre un rango.
  - IN/EXISTS. Comprueba si un valor se encuentra dentro de un conjunto.
  - IS NULL. Comprueba si el valor es nulo.

Existen otro tipo de funciones llamadas agregadas que permiten obtener el valor medio (AVG), el máximo (MAX), mínimo (MIN) y el sumatorio (SUM) de un conjunto de valores, así como el número de filas (COUNT) resultado de la ejecución de la consulta. Todas estas funciones devuelven únicamente una fila con el resultado de aplicar la función correspondiente, a menos que se utilice la

cláusula GROUP BY, la cual hace que se agrupen los resultados por el atributo deseado antes de realizar la función agregada, proporcionando un resultado de la función por cada agrupación. Adicionalmente, si se desea que esas agrupaciones cumplan una condición, se tiene que usar la palabra reservada HAVING, que tiene un funcionamiento similar al de WHERE.

La manera de ordenar los resultados de la consulta que proporciona el lenguaje SQL es la utilización de la cláusula ORDER BY que permite ordenar de forma ascendente (ASC) o descendente (DESC) los resultados de la operación.

Por último, dentro de la operación de consulta, se puede realizar la operación de concatenar las filas de una columna con las de otra distinta teniendo un atributo común por medio del cual están relacionadas, utilizando la sentencia JOIN.

Asimismo, pueden anidarse consultas, es decir, incluir una cláusula SELECT dentro de una WHERE, pudiendo realizar operaciones de consulta con muchos criterios de búsqueda.

Para la aplicación de este proyecto en concreto, las características a tener en cuenta en el diseño de las consultas necesarias son las siguientes. Los valores que se indican son los máximos para cada una de las características:

- Número de relaciones obligatorias: 32 por tabla
- Número de tablas en una consulta: 32
- Número de combinaciones en una consulta: 26
- Número de campos de un conjunto de registros: 255
- Tamaño del conjunto de registros: 1GB
- Límite de ordenación: 255 caracteres en uno o más campos
- Número de niveles de consultas anidadas: 50
- Número de caracteres de una celda en la cuadrícula de diseño
- Número de caracteres para un parámetro en una consulta de parámetros: 255
- Número de operadores AND en una cláusula WHERE o HAVING: 99
- Número de caracteres en una instrucción SQL: Aproximadamente 64000

#### 2.1.4 Formularios

Para que el usuario pueda explotar la base de datos es necesario proporcionarle una interfaz para poder realizar las operaciones de consulta, modificación, inserción y eliminación fácilmente. Es por ello por lo que surge la necesidad de utilizar los formularios. Estos formularios muestran información extraída de la base de datos a través de consultas, de una forma clara y legible al usuario.

El software Microsoft Access ofrece un asistente para la confección de estos formularios, aunque también se puede hacer manualmente añadiendo los diferentes controles y configurando su aspecto, funcionalidad y origen de datos de cada uno de ellos. Entre los controles disponibles destacan, los cuadros de texto como control básico para mostrar información, los cuadros combinados o los subformularios. Estos últimos, como su propio nombre indica son formularios que están contenidos dentro de un formulario y son muy útiles para ver información acerca del registro del que está mostrando información el formulario principal.

Para finalizar este apartado se enumeran los valores máximos de las características a tener en cuenta en el desarrollo de los formularios en Microsoft Access:

- Número de caracteres de una etiqueta: 2048
- Número de caracteres de un cuadro de texto: 65.535
- Ancho del formulario o informe: 57,79cm
- Alto de la sección: 57,79cm
- Alto de todas las secciones más los encabezados de sección: 508cm
- Número de niveles de formularios o informes anidados: 7
- Número de campos o expresiones que puede ordenar o agrupar un informe: 10
- Número de encabezados y pies de página en un informe: 1
- Número de páginas impresas en un informe: 65.536

## 2.2 UML

A finales de los años 80 y durante todos los años 90, se desarrolló gran número de métodos y lenguajes de programación orientada a objetos, la cual estuvo en auge antes de la introducción del lenguaje UML en este tipo de software. Esta gran variedad de métodos y lenguajes raramente eran compatibles entre sí, lo que hizo que **James Rumbaugh, Grady Booch e Ivar Jacobson** combinaran distintos lenguajes que ya existían y crear un estándar. Cada uno de estos programadores ya había creado un método de desarrollo de software orientado a objetos, James Rumbaugh creó la técnica de modelado de objetos (OMT), Grady Booch creó el método Booch e Ivar Jacobson creó el método de Ingeniería de Software Orientado a Objetos (OOSE). El estándar UML que iban a crear en 1996 (actualmente la versión 2.5.1 desde diciembre de 2017) debía definir un modelado visual y una semántica común para representar los métodos creados por estos tres desarrolladores. Este modelado visual permite ver los estados e interacciones entre diferentes objetos de un sistema como pueden ser:

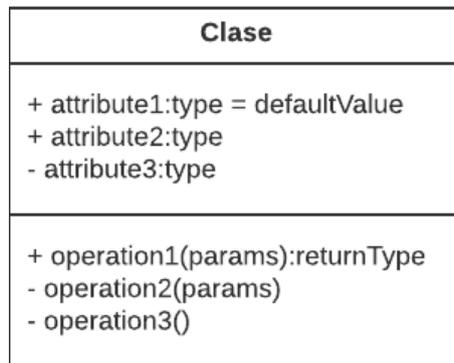
- Objetos. Es el elemento básico.
- Clases. Grupos de elementos con las mismas propiedades.
- Relaciones.
- Actividades.
- Interacciones.

La técnica que se va a utilizar para realizar el diseño de la solución propuesta es el diagrama de clases el cual describe la estructura del sistema viendo cómo se organizan sus clases, dentro de éstas sus atributos, y como se relacionan entre sí.

### 2.2.1 Diagrama de Clases

El diagrama de clases es la estructura estática del sistema, lo que quiere decir que lo que se refleje en él debe de ser válido durante el ciclo de vida del mismo. Las clases se representan con un rectángulo dividido en tres zonas. La primera zona es la destinada al nombre de la clase, la zona intermedia es donde

se definen los atributos propios de la clase y la última zona es en la que se pueden ver los métodos u operaciones que se pueden llevar a cabo en esa clase.



**Figura 5. Ejemplo de clase UML.**

Otro de los elementos de un diagrama de clases son las relaciones, que pueden ser de distinto tipo dependiendo de la relación que exista entre las clases que estén vinculadas. Estos son los distintos tipos de relaciones:

- Herencia.  Se utiliza para indicar que una clase derivada hereda todos los atributos y métodos de la clase principal de la que depende. Además de estos atributos y métodos de la clase principal, puede tener unos propios de la clase derivada.
- Asociación.  Se utiliza para indicar que un objeto de la clase A se relaciona de alguna manera con un objeto de la clase B. En este tipo de relación se pone una etiqueta sobre la misma para indicar de qué manera se relacionan.
- Agregación.  Se utiliza para indicar que un objeto de una clase A puede formar parte de un objeto de otra clase A, pero el objeto de la clase A puede seguir existiendo, aunque el objeto de la clase B desaparezca.
- Composición.  Se utiliza para indicar que un objeto de una clase derivada forma parte de un objeto de la clase principal pero no puede existir sin un objeto de esa clase principal.

Por último, es necesario el concepto de cardinalidad que sirve para indicar cuántos objetos de una clase se relacionan con un objeto de otra y pueden ser:

- Uno y sólo uno: 1

- Cero o uno: 0..1
- Cero o muchos: 0..\*
- Uno o muchos: 1..\*
- Desde N hasta M: N..M (N y M indican un número)

Si la cardinalidad mínima es 0, significa que la relación puede no existir, sin embargo, cuando la cardinalidad mínima es 1, ha de existir obligatoriamente.

### **2.2.2 Propuesta de proyecto**

Como se ha mencionado anteriormente, este proyecto surge de la necesidad de realizar estudios estadísticos sobre enfermedades oculares de pacientes tratados en el servicio de oftalmología del Hospital Provincial de Castellón. Previamente, este tipo de estudios, dada la baja complejidad de la información que se necesitaba almacenar, se llevaba a cabo mediante hojas de cálculo, pero a raíz del abordaje multidisciplinar que se lleva a cabo en algunas enfermedades, resultaba muy complicado seguir haciéndolo con el mismo método.

Para dar respuesta a esta nueva necesidad es necesario utilizar un sistema de información que sea capaz de almacenar de forma estructurada y relacionada toda la información necesaria, por ello se eligió una base de datos como herramienta para tal fin. Esta base de datos almacenará información acerca de los diferentes elementos de los que después se extraerá información y datos estadísticos.

Esta propuesta de solución no se diferencia en cuanto al resultado de lo que se venía utilizando hasta ahora pero sí en cuanto a la estructura de la información y su contenido, proporcionando mayor funcionalidad y posibilidades.

### **3 Análisis del problema**

En el instante anterior a la realización de este proyecto, el servicio de oftalmología del Hospital Provincial de Castellón no disponía de un sistema de información que le fuera útil para gestionar a los pacientes a la vez que realizara estudios estadísticos sobre tratamientos aplicados en ellos.

#### **3.1 Análisis de Seguridad**

Debido a que en la base de datos se va a guardar información personal sobre terceras personas, es necesario realizar un análisis de seguridad para garantizar que esta información no esté disponible para cualquier persona.

El fichero de Microsoft Access que contendrá la base de datos va a estar alojado en un equipo con sistema operativo Microsoft Windows 10 y además no se podrá tener acceso desde otro equipo, aunque esté en la misma red. Esto hace que se simplifique el tema de la seguridad ya que solamente habrá que tener en cuenta los accesos desde un único equipo y, además se sabe que son necesarios un usuario y una contraseña para poder iniciar sesión en él. El fichero necesitará una contraseña para poder tener acceso a la información guardada en él, cifrando todo su contenido, de manera que no legible a no ser que se acceda de la manera mencionada.

Respecto a las copias de seguridad, tanto MS Access como MS Windows proporcionan esta posibilidad. La manera que tiene MS Access para hacerlo es manualmente, de la siguiente manera:

- Una vez abierto el fichero, acceder al menú Archivo – Guardar Como.
- El siguiente paso, en Tipos de archivo seleccionar Guardar base de datos como.
- En la opción Avanzadas, elegir Realizar copia de seguridad de la base de datos y después Guardar como.
- Introducir el nombre que se quiere dar a la copia de seguridad y la localización.
- Y por último hacer click en Guardar.

Las copias de seguridad también están cifradas de la misma manera por lo que se mantiene la misma seguridad que en el fichero principal.

En cuanto a MS Windows, ofrece la posibilidad de programar la realización automática de las copias de seguridad, indicando qué ficheros se van a copiar, dónde se va a almacenar la copia y con qué frecuencia se va a realizar dicha copia.

## **3.2 Análisis del marco legal**

Hay que tener en cuenta a la hora del desarrollo y futura explotación de la solución la normativa vigente en los ámbitos que pueden afectar a la misma. En este sentido, el principal y único punto para tener en cuenta es el de la protección de datos personales.

### **3.2.1 Protección de datos**

La base de datos va a contener información personal de pacientes que están en tratamiento por alguna enfermedad ocular, por lo que hay que revisar las leyes y normativa que lo regula.

A raíz del Reglamento 2016/679 del Parlamento Europeo y del Consejo de 27 de abril de 2016 relativo a la protección de las personas físicas en lo que respecta al tratamiento de los datos personales y a la libre circulación de estos datos, en España se dictó la Ley Orgánica 3/2018 de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales, como así lo refleja en el artículo 1, apartado a):<sup>3</sup>

*“La presente ley orgánica tiene por objeto:*

- a) Adaptar el ordenamiento jurídico español al Reglamento (UE) 2016/679 del Parlamento Europeo y el Consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de sus datos personales y a la libre circulación de estos datos, y completar sus disposiciones.*

---

<sup>3</sup> BOE num 924, de 6 de diciembre de 2018.

*El derecho fundamental de las personas físicas a la protección de datos personales, amparado por el artículo 18.4 de la Constitución, se ejercerá con arreglo a lo establecido en el Reglamento (UE) 2016/679 y en esta ley orgánica.*

Esta misma ley en su artículo 2 establece su ámbito de aplicación:

*Artículo 2. Ámbito de aplicación de los Títulos I a IX y de los artículos 89 a 94.*

- 1. Lo dispuesto en los Títulos I a IX y en los artículos 89 a 94 de la presente ley orgánica se aplica a cualquier tratamiento total o parcialmente automatizado de datos personales, así como al tratamiento no automatizado de datos personales contenidos o destinados a ser incluidos en un fichero.*
- 2. Esta ley orgánica no será de aplicación:*
  - a) A los tratamientos excluidos del ámbito de aplicación del Reglamento general de protección de datos por su artículo 2.2, sin perjuicio de lo dispuesto en los apartados 3 y 4 de este artículo.*
  - b) A los tratamientos de datos de personas fallecidas, sin perjuicio de lo establecido en el artículo 3.*
  - c) A los tratamientos sometidos a la normativa sobre protección de materias clasificadas.*
- 3. Los tratamientos a los que no sea directamente aplicable el Reglamento (UE) 2016/679 por afectar a actividades no comprendidas en el ámbito de aplicación del Derecho de la Unión Europea, se regirán por lo dispuesto en su legislación específica si la hubiere y supletoriamente por lo establecido en el citado reglamento y en la presente ley orgánica. Se encuentran en esta situación, entre otros, los tratamientos realizados al amparo de la legislación orgánica del régimen electoral general, los tratamientos realizados en el ámbito de instituciones penitenciarias y los tratamientos derivados del Registro Civil, los Registros de la Propiedad y Mercantiles.*
- 4. El tratamiento de datos llevado a cabo con ocasión de la tramitación por los órganos judiciales de los procesos de los que sean competentes, así como el realizado dentro de la gestión de la Oficina Judicial, se regirán por lo dispuesto en el Reglamento (UE) 2016/679 y la presente ley orgánica, sin perjuicio de las disposiciones de la Ley Orgánica 6/1985, de 1 julio, del Poder Judicial, que le sean aplicables.*

### **3.3 Análisis de riesgos**

En todo proyecto pueden existir riesgos que afecten a diferentes ámbitos, como el económico y el temporal principalmente. Después de realizar un análisis sobre el proyecto se han detectado ciertos riesgos que pueden afectar a los plazos, al presupuesto o a ambos. Son los siguientes:

- La realización de una planificación incorrecta: Esto puede llevar al incumplimiento de los plazos establecidos para la entrega del proyecto o de los hitos marcados.
- La indisponibilidad de ciertos recursos necesarios para la ejecución de una parte del proyecto: Esta circunstancia se puede dar al necesitar un recurso determinado en un cierto instante, pero no poder disponer de él, haciendo que se retrasen los plazos o teniendo que contratar o adquirir otro recurso para suplir al que no está disponible. En el caso de este proyecto, solamente hay un recurso humano, por lo que cualquier circunstancia en la que se pueda dar este riesgo supondrá un retraso en los plazos.
- El cambio en los requisitos o restricciones por parte del cliente: Este cambio afecta a los plazos de entrega e, indirectamente, al coste económico del proyecto al tener que dedicar durante más tiempo del planeado un recurso a una tarea.

### **3.4 Posibles soluciones**

Una vez identificados los principales riesgos que pueden afectar al conjunto del proyecto, es necesario planificar las posibles soluciones para contrarrestar sus efectos negativos. A continuación, se enumeran las soluciones planteadas para los riesgos citados en el punto anterior respectivamente.

- Para evitar una planificación incorrecta hay que establecer desde un principio las tareas a realizar y los recursos de los que se va a disponer durante la ejecución del proyecto. En un principio puede resultar complicado estimar el tiempo necesario para completar cada una de las tareas de las que se va a componer el proyecto, por lo que se puede recurrir a una técnica que consiste en dividir cada una de ellas en tareas más sencillas, haciendo mucho más sencilla esta estimación.

- Los efectos de la no disponibilidad de recursos en el momento en el que es necesario disponer de ellos se pueden contrarrestar de dos maneras, la primera de ellas es aumentar el presupuesto inicial contemplando la posibilidad de necesitar contratar o adquirir cierto recurso en un momento puntual. La segunda de ellas es la realización de una planificación en la que ya se plantee este retraso, haciendo que este nuevo plazo sea el establecido para la entrega al cliente. En este caso, se ha de optar por la segunda opción ya que no se puede plantear la opción de contratar a alguien para suplir al encargado de la realización del proyecto.
- En la fase de obtención de requisitos hay que preparar un documento en el que queden claros cuales son y que la modificación de cualquiera de ellos supondrá una variación en los plazos de entrega y del presupuesto inicial para que, si se llegara a dar el caso, los posibles retrasos o encarecimiento del proyecto estén cubiertos y su incumplimiento esté avalado por el cliente.

### **3.5 Solución propuesta**

La opción elegida para dar solución al problema inicial consiste en el desarrollo de una aplicación de bases de datos que permita a los usuarios finales el almacenamiento de información relativa a pacientes del servicio de oftalmología del Hospital Provincial de Castellón para la realización de estudios. Teniendo en cuenta todos los requisitos y restricciones presentadas en la especificación de requisitos, se ha optado por el desarrollo con el software Microsoft Access, ya que permite tanto implementación de la base de datos como el desarrollo de la aplicación que posteriormente se utilizará para la introducción de datos en la base de datos de una manera cómoda y fácil, una razón más que he hecho que se decida esta opción es que es el software del que se dispone en el servicio, por lo que no requerirá un gasto económico extra en la adquisición de licencias de ningún otro software.

### **3.6 Plan de trabajo**

Para la ejecución de este proyecto es necesario pasar por diversas etapas o fases, que se detallan a continuación:

- Análisis del problema, obteniendo información sobre los requisitos del cliente y las restricciones que se presentan.
- Selección y justificación del proyecto. A partir de la primera fase se escoge la opción más adecuada que dé solución al problema presentado.
- Análisis de riesgos. En todo proyecto se ha de tener en cuenta posibles problemas que pueden surgir a lo largo del mismo. En esta etapa se analizarán los que puedan aparecer y se plantearán las soluciones posibles para cada uno de ellos.
- Diseño del proyecto. Para todos los apartados del mismo se usarán técnicas de diseño. Por ejemplo, UML en el caso de la base de datos o la aplicación de las leyes de Gestalt para la interfaz gráfica de usuario.
- Implementación de cada una de las partes del proyecto con la herramienta elegida.
- Realización de pruebas para la comprobación de que satisface las necesidades iniciales.
- Implantación del proyecto finalizado en el sistema final.
- Redacción de la memoria.

### 3.7 Planificación y presupuesto

A partir de las etapas definidas en el punto anterior se plantea la planificación detallada en la Figura 6 mediante un diagrama de Gantt.



Figura 6. Diagrama de Gantt

Como se puede observar, excepto la tarea “Redacción de la memoria” todas comienzan cuando finaliza la anterior, esto es porque es necesario que se finalicen para poder comenzar con la siguiente, por ejemplo, es necesario tener el diseño de la solución para poder pasar a la implementación de ésta. En el caso de la redacción de la memoria, ninguna otra tarea depende de su consecución por lo que se ha podido retrasar su inicio a pesar de poder haberse iniciado una vez se ha finalizado la tarea de “Selección y justificación del proyecto”.

En lo que respecta al presupuesto, el cálculo se ha realizado estimando un sueldo de un ingeniero que acaba de conseguir la titulación, que está entorno a los 16.000 euros brutos anuales, lo que supone 1.333 euros brutos mensuales. Haciendo un cálculo de 160 horas mensuales de trabajo, esto supone un coste de 8,33 €/hora. Teniendo en cuenta que el desarrollo del proyecto se ha extendido durante 4 meses, que son un total de 640 horas, el coste total del proyecto asciende a 5.332 €.<sup>4</sup>

<b>Horas totales del proyecto</b>	640 horas
<b>Sueldo bruto anual</b>	16.000 €
<b>Sueldo bruto mensual</b>	1.333 €
<b>Sueldo bruto por hora</b>	8,33 €
<b>Coste total del proyecto</b>	5.332 €

**Tabla 3. Presupuesto del proyecto**

---

<sup>4</sup> Sueldo de ingeniero informático recién graduado ([https://universidadeuropea.es/blog/cuanto-gana-un-ingeniero-informatico#:~:text=El%20salario%20de%20un%20ingeniero,1.000%2D1.200%20euros%20netos%20mensuales\).](https://universidadeuropea.es/blog/cuanto-gana-un-ingeniero-informatico#:~:text=El%20salario%20de%20un%20ingeniero,1.000%2D1.200%20euros%20netos%20mensuales).)

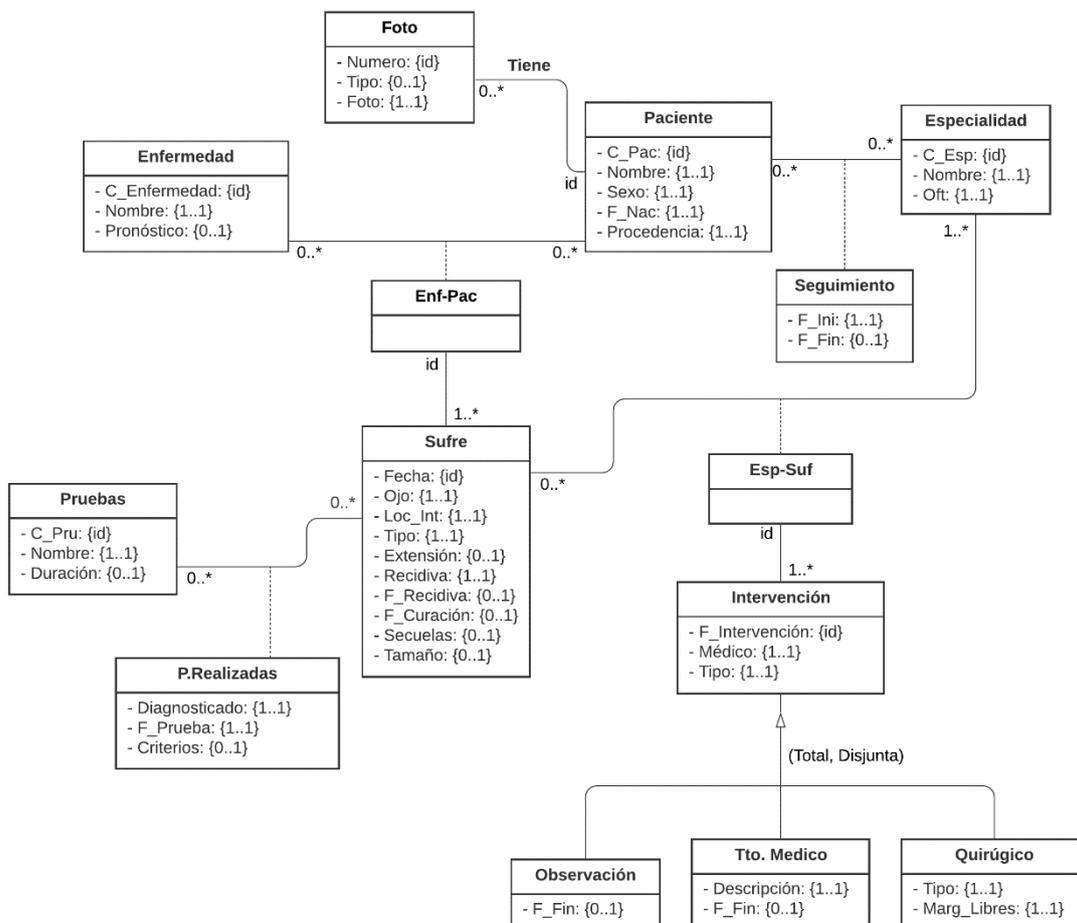
## 4 Diseño de la solución

Para poder diseñar la base de datos, que es el paso previo a la creación de ésta, es necesario obtener la especificación de requisitos, que es una descripción de los requisitos tanto funcionales como no funcionales que debe de satisfacer el futuro sistema de información. Los requisitos funcionales indicarán qué debe hacer y cómo lo debe hacer, aunque también es posible que indique qué no debe hacer, en cambio, los no funcionales indican aspectos relativos a cómo debe ser el sistema como, por ejemplo, restricciones de hardware o de rendimiento.

Los requisitos deben de ser claros, concretos y concisos y en un lenguaje natural, que permitan a cualquier persona sin conocimientos específicos comprender lo que se está pidiendo. Deben priorizarse, o al menos definir qué requisitos son obligatorios y cuales deseables y, por último, deben especificar el comportamiento del sistema, sin definir características relativas a su diseño.

### 4.1 Diagrama UML

A partir de la especificación de requisitos, se realiza el primer paso del diseño, que es el diagrama de clases de UML. En este diagrama se pueden observar los elementos de los que constará la posterior base de datos y las relaciones entre ellos. Para cada uno de estos objetos se pueden ver los atributos de los que constará, y para las relaciones entre los objetos, se puede ver su cardinalidad en la Figura 7. Por ejemplo, un objeto de la clase *Paciente* tendrá como atributos un código de paciente, un nombre, un sexo (Hombre o mujer), una fecha de nacimiento y un servicio médico u hospital de procedencia. En cuanto a las relaciones, se puede apreciar que, por ejemplo, existe una relación llamada *Tiene* entre los objetos de la clase *Paciente* y los de la clase *Foto*, indicando que un objeto de la clase *Foto*, se debe relacionar con un único objeto de la clase *Paciente*, pero no ocurre lo mismo, al contrario, ya que un objeto de la clase *Paciente* se puede relacionar con cero o varios objetos de la clase *Foto*. Esto quiere decir que una foto debe de pertenecer a un paciente y que un paciente puede tener cero o múltiples fotografías.



**Figura 7. Diagrama de clases UML del proyecto**

Además de las restricciones de cardinalidad y de identificación que se aprecian en la Figura 7 cabe destacar que existen otras que aseguran la coherencia de los datos almacenados:

- Coherencia de fechas:
  - ◆ La *fecha* de un objeto de la clase *Sufre* debe ser posterior a la *f\_nac* del objeto de la clase *Paciente* con el que se relaciona.
  - ◆ La *fecha* de un objeto de la clase *Sufre* debe ser anterior a las fechas (*f\_prueba* y *f\_intervención*) de las clases *P\_Realizadas* e *Intervención* de los objetos con los que se relaciona.
  - ◆ El valor de *f\_ini* de un objeto de la clase *Seguimiento* debe ser posterior al valor de *f\_nac* del objeto de la clase *Paciente* con el que se relaciona.

- ◆ En un objeto de la clase *Seguimiento*, *f\_ini* debe ser anterior *f\_fin*.
- ◆ El valor de *f\_intervención* de un objeto de la clase *Intervención* debe de ser anterior al valor de los atributos *f\_fin* de los objetos de las clases *Observación* y *Tto.Médico* en los que se especializa.

## 4.2 Esquema lógico de la base de datos

A partir del diagrama de clases se procede a su transformación para poder implementar su diseño en MS Access. Fruto de esta transformación se obtiene el siguiente esquema relacional:

```

Enfermedad(c_enf: Autonumeración, nombre: Texto corto,
           pronóstico: Texto corto)
  CP: {c_enf}
  VNN: {nombre}

Especialidad (c_esp: Autonumeración, nombre: Texto corto,
             oft: Sí/No)
  CP: {c_esp}
  VNN: {nombre, oft}

Paciente(C_pac: Número, nombre: Texto corto, sexo: Texto corto,
         f_nac: Fecha/Hora, procedencia: Texto corto)
  CP: {C_pac}
  VNN: {nombre, sexo, f_nac, procedencia}

Pruebas(c_pru: Autonumeración, nombre: Texto corto,
        duración: Texto corto)
  CP: {c_pru}
  VNN: {nombre}

Seguimiento(c_pac: Numero, c_esp: Número, f_ini: Fecha/Hora,
            f_fin: Fecha/Hora)
  CP: {c_pac, c_esp}
  CAj: {c_pac} -> Paciente(C_pac)
  CAj: {c_esp} -> Especialidad(c_esp)
  VNN: {f_ini}

Foto(numero: Autonumeración, c_pac: Número, tipo: Texto corto,
     foto: Datos adjuntos)
  CP: {numero, c_pac}
  CAj: {c_pac} -> Paciente(C_pac)
  VNN: {foto}

Sufre(c_pac: Número, c_enf: Número, fecha: Fecha/Hora,
     ojo: Texto corto, loc_interna: Texto corto,
     tipo: Texto corto, extensión: Texto corto,
     recidiva: Sí/No, f_recidiva: Fecha/Hora,
     f_curación: Fecha/Hora, secuelas: Texto corto,
     tamaño: Texto corto)

```

```

CP: {c_pac, c_enf, fecha}
CAj: {c_pac} -> Paciente(C_pac)
CAj: {c_enf} -> Enfermedad(c_enf)
VNN: {ojo, loc_interna, tipo, recidiva}

P_Realizadas(c_pru: Número, c_pac: Número, c_enf: Número,
             fecha: Fecha/Hora, diagnosticado: Sí/No,
             f_prueba: Fecha/Hora, criterios: Texto corto)
CP: {c_pru, c_pac, c_enf, fecha}
CAj: {c_pru} -> Pruebas(c_pru)
CAj: {c_pac, c_enf, fecha} -> Sufre(c_pac, c_enf, fecha)

Intervención(c_esp: Número, c_pac: Número, c_enf: Número,
             fecha: Fecha/Hora, f_intervención: Fecha/Hora, medico:
             Texto corto, tipo: Texto corto)
CP: {c_esp, c_pac, c_enf, fecha, f_intervención}
CAj: {c_pac, c_enf, fecha} -> Sufre(c_pac, c_enf, fecha)
CAj: {c_esp} -> Especialidad(c_esp)
VNN: {medico, tipo}

Observación (c_esp: Número, c_pac: Número, c_enf: Número,
             fecha: Fecha/Hora, f_intervención: Fecha/Hora, f_fin:
             Fecha/Hora)
CP: {c_esp, c_pac, c_enf, fecha, f_intervención}
CAj: {c_esp, c_pac, c_enf, fecha, f_intervención}
->
Intervención(c_esp, c_pac, c_enf, fecha, f_intervención)

Quirúrgico (c_esp: Número, c_pac: Número, c_enf: Número,
            fecha: Fecha/Hora, f_intervención: Fecha/Hora, tipo:
            Texto corto, marg_libres: Sí/No)
CP: {c_esp, c_pac, c_enf, fecha, f_intervención}
CAj: {c_esp, c_pac, c_enf, fecha, f_intervención}
->
Intervención(c_esp, c_pac, c_enf, fecha, f_intervención)
VNN: {tipo, marg_libres}

Tto_medico (c_esp: Número, c_pac: Número, c_enf: Número,
            fecha: Fecha/Hora, f_intervención: Fecha/Hora, tipo:
            Texto corto, marg_libres: Sí/No)
CP: {c_esp, c_pac, c_enf, fecha, f_intervención}
CAj: {c_esp, c_pac, c_enf, fecha, f_intervención}
->
Intervención(c_esp, c_pac, c_enf, fecha, f_intervención)
VNN: {descripción}

```

En la lista siguiente, se incluyen las restricciones de integridad que no pueden expresarse en este esquema:

- Coherencia de fechas:
  - ◆ La *fecha* de una tupla de la tabla *Sufre* debe ser posterior a la *f\_nac* de la tupla de la tabla *Paciente* con la que se relaciona.

- ◆ La *fecha* de un tupla de la tabla *Sufre* debe ser anterior a las fechas (*f\_prueba* y *f\_intervención*) de las tablas *P\_Realizadas* e *Intervención* de las tuplas con las que se relaciona.
- ◆ El valor de *f\_ini* de una tupla de la tabla *Seguimiento* debe ser posteriores al valor de *f\_nac* de la tupla de la tabla *Paciente* con la que se relaciona.
- ◆ En una tupla de la tabla *Seguimiento*, *f\_ini* debe ser anterior *f\_fin*.
- ◆ El valor de *f\_intervención* de una tupla de la tabla *Intervención* debe de ser anterior al valor de *f\_fin* de las tuplas de las tablas *Observación* y *Tto.Médico* en las que se especializa.
- Toda tupla de la tabla *Intervención* se relaciona con una tupla y nada más que una de las tablas *Observación*, *Tto. Médico* o *Quirúrgico*.

La visión gráfica de este esquema se puede obtener en el MS Access y se presenta en la Figura 8.

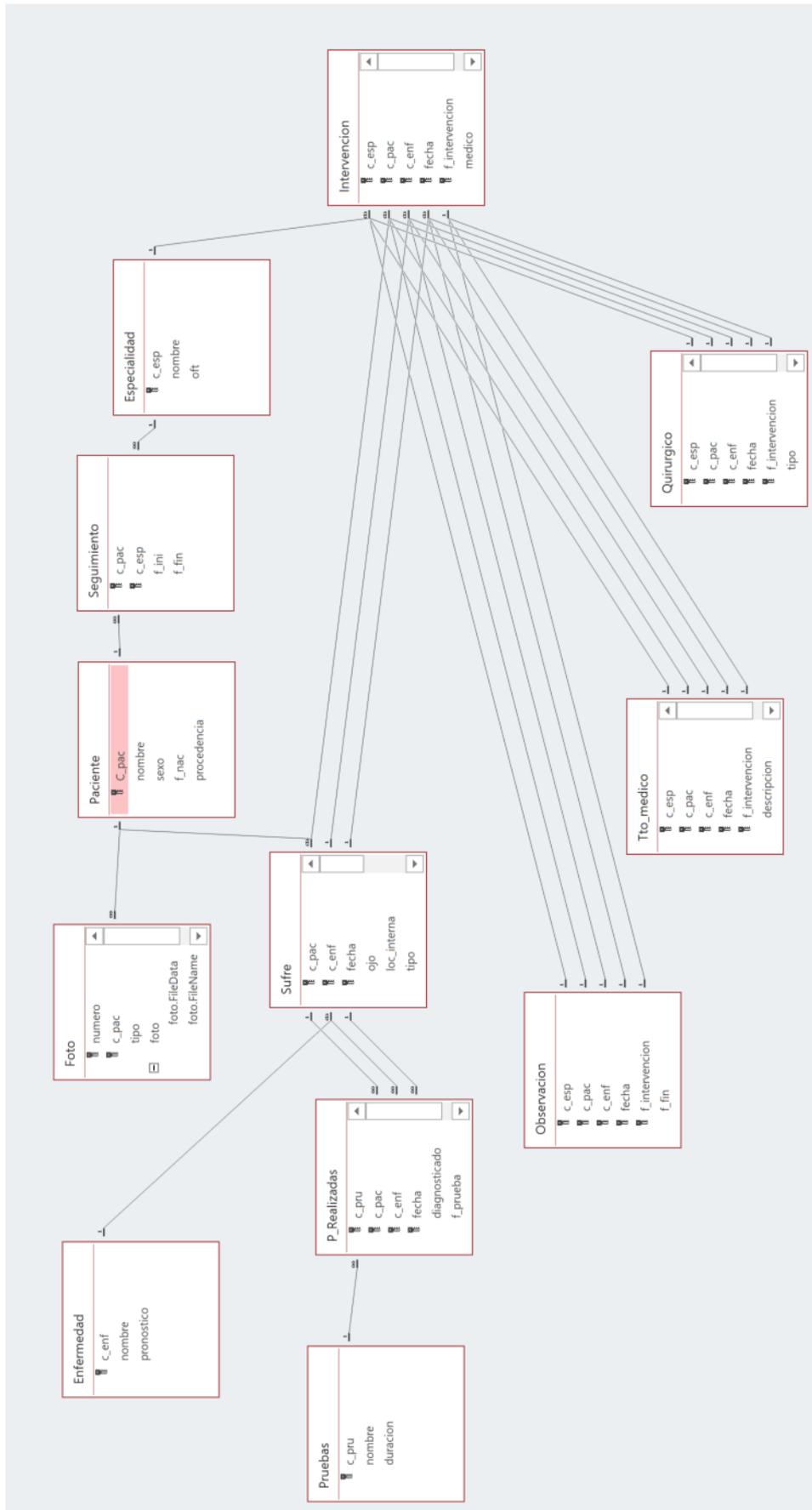


Figura 8. Esquema de tablas y relaciones del proyecto.



### 4.3 Diseño de consultas

Como se ha visto anteriormente, las consultas sirven para recuperar información de un conjunto de tablas con una serie de condiciones. Se han diseñado diferentes consultas para visualizar información en los distintos formularios, así como para recuperar la información requerida en los informes.

Las consultas diseñadas para recuperar la información para los formularios permitirán mostrar la información relativa a un paciente determinado y son las siguientes:

- “Buscar pruebas”: Sirve para recuperar la información de “Pruebas Realizadas” relativas a un paciente. Además, filtra de entre éstas, las que tengan un nombre en concreto.
- “Consulta Enfermedades”: Sirve para recuperar la información de “Enfermedades” que tiene o ha tenido un paciente. Filtra de entre los resultados de la consulta, los que coincidan con una enfermedad en concreto.
- “Consulta Intervenciones”: Recupera de la base de datos las intervenciones que se le ha realizado a un paciente. De entre toda ellas, escoge solamente las coincidentes con una enfermedad concreta.
- “Consulta Seguimientos”: Muestra los seguimientos que tiene o ha tenido un paciente.

Las consultas creadas para mostrar información en los informes han sido las siguientes:

- “Enfermedad Consulta”: Sirve para extraer de la base de datos los pacientes que han sufrido la enfermedad seleccionada en el formulario desde el que se genera el informe correspondiente y que además coincidan con el criterio de recidiva marcado.
- “Sufre Consulta”: Recupera de la base de datos el conjunto de pacientes que han padecido una la enfermedad seleccionada cuadro combinado del formulario.
- “Paciente Consulta Porcentaje”: Sirve para recuperar de la base de datos los pacientes que han sufrido una enfermedad concreta y genera un atributo nuevo en el resultado de la consulta en función de la edad del paciente cuando le fue diagnosticada dicha enfermedad.

#### 4.4 Diseño de la interfaz gráfica de usuario

La parte relativa a la interfaz gráfica de usuario hace referencia a los formularios comentadas en el punto 2.1.4, los cuales son útiles para ver, introducir y modificar toda la información de una manera cómoda para el usuario. Lo más adecuado es que los formularios tengan un diseño y arquitectura sencillos para que el usuario no tenga dificultades en su uso, aun así, es posible que sea necesaria una fase de aprendizaje para su fácil y ágil utilización.

Se ha optado por crear un formulario principal, a modo de menú en el que el usuario puede seleccionar entre distintas opciones, dependiendo de lo que desee hacer. Dentro de este formulario, llamado "*Principal*", existe un acceso a los siguientes formularios:

- "Formulario Enfermedad": Sirve para dar de alta, modificar o eliminar de la base de datos las enfermedades que puede sufrir un paciente.
- "Formulario Pruebas": Se puede introducir, modificar o borrar información relativa a pruebas médicas, las cuales posteriormente serán relacionadas con pacientes.
- "Formulario Especialidad": En este formulario se puede manipular la información referente a las distintas especialidades que se encargarán de tratar las enfermedades.
- "Formulario Paciente": Sirve para dar de alta, modificar o eliminar toda la información relativa a los pacientes. Dentro de este formulario se han creado distintos subformularios para ver la información referente al paciente actual en un instante concreto. Estos son:
  - "Subformulario Foto": Visualiza las fotografías almacenadas y permite añadir, modificar o borrarlas.
  - "Subformulario Seguimiento": Desde este subformulario se puede modificar la información sobre los seguimientos realizados al paciente actual mostrado en el "*Formulario Paciente*".
  - "Subformulario Sufre": Se puede ver y editar la información sobre las enfermedades que padece o ha padecido el paciente actual.

- “Subformulario P\_Realizadas”: Muestra y permite editar la información referente a las pruebas que se realizan para diagnosticar una enfermedad al paciente actual.
- “Subformulario Intervención”: En este último subformulario del “*Formulario Paciente*” se pueden modificar los datos de las diferentes intervenciones realizadas para tratar las distintas enfermedades que haya podido sufrir un paciente. Dentro de este subformulario se han creado tres subformularios más para mostrar los diferentes datos de las intervenciones, dependiendo de su tipo:
  - ◆ “Subformulario Observación”
  - ◆ “Subformulario Quirúrgico”
  - ◆ “Subformulario Tto\_Médico”
- “Formulario informe enfermedades”: Se trata de una pequeña ventana desde la que se seleccionan unos criterios para generar informes.

## 5 Implementación

En esta sección se detalla el proceso de creación de la base de datos, pormenorizando los pasos a seguir para la implementación de las tablas, relaciones, consulta y formularios.

### 5.1 Tablas

Para la implementación de una base de datos y, una vez realizado el diseño, el siguiente paso es la creación de las tablas que van a almacenar la información. Las tablas que hay que definir, son las incluidas en el esquema lógico obtenido en el apartado 4.2.

Para la creación de tablas en Microsoft Access se debe acceder al menú *Crear* y bien pulsar en la opción *Tabla* o bien *Diseño de tabla*. La diferencia entre las dos opciones es que, en el primer caso, se tendrá una vista de la tabla con los datos que pueda tener almacenados, viendo una fila por cada registro y una columna por cada atributo y con la posibilidad de añadir más, eligiendo de una lista el tipo de dato que tendrá ese atributo nuevo (Figura 9), y en el segundo caso no se verán los datos que pueda tener almacenada la tabla, solamente los atributos con el tipo de datos correspondiente de los que consta la misma, además de una serie de opciones que se mencionarán más adelante (Figura 10).

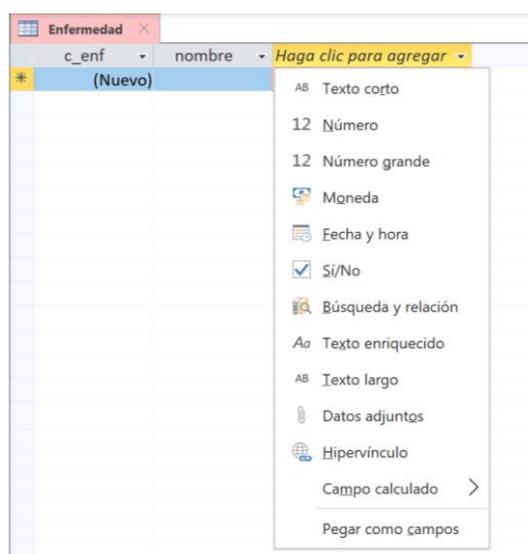


Figura 9. Creación de una tabla. Menú Tabla



A continuación, se pueden ver los diferentes tipos de datos utilizados en este proyecto junto a las propiedades de cada uno de ellos:

- Aut numeración: este tipo de dato sólo se ha utilizado para identificar unívocamente los registros de las tablas que representan objetos de la realidad (Enfermedad, Foto, Especialidad y Pruebas). La única propiedad en la que no se ha dejado el valor por defecto es *Indexado*, el cual sirve para crear un índice que acelera las búsquedas ordenando todos los registros de la tabla por ese atributo. Como contrapartida, esto hace que las actualizaciones sean más lentas ya que tiene que reordenar la tabla en función del nuevo dato guardado.

General	Búsqueda
Tamaño del campo	Entero largo
Nuevos valores	Incrementalmente
Formato	
Título	
Indexado	No
Alineación del texto	General

**Figura 12. Propiedades de tipo Aut numeración**

- Número: Este tipo de dato sirve para almacenar datos de tipo numérico y se ha utilizado principalmente para los atributos de las clases asociación que sirven para relacionar dos clases distintas, por ejemplo, para relacionar los registros de la tabla *Pacientes* y los registros de la tabla *Enfermedades*, surge la clase asociación, que tiene entre otros, los atributos “*c\_pac*” y “*c\_enf*” que son el código de paciente y el código de enfermedad respectivamente, los cuales sirven para establecer la relación antes mencionada (actúan como clave ajena). Cabe destacar que el tipo de datos del conjunto de atributos que actúan como clave ajena de una tabla deben de ser el mismo que la clave principal de la tabla con la que se relaciona. Únicamente se ha establecido la propiedad *Indexado* como *Sí (Con duplicados)* dejando el resto de las propiedades con el valor por defecto, excepto en los atributos “Clave Ajena”, en los que la propiedad *Requerido* se establece como *Sí*.

General	Búsqueda
Tamaño del campo	Entero largo
Formato	
Lugares decimales	Automático
Máscara de entrada	
Título	
Valor predeterminado	0
Regla de validación	
Texto de validación	
Requerido	Sí
Indexado	Sí (Con duplicados)
Alineación del texto	General

**Figura 13. Propiedades de tipo Número**

- **Texto corto:** Este tipo de dato es útil para almacenar información con caracteres alfanuméricos, tales como una palabra o una descripción. Las propiedades destacables son *Tamaño del campo* y *Permitir longitud cero*. La primera de ellas sirve para establecer el número de caracteres máximo que puede tener el dato almacenado en ese atributo para un registro de la tabla y el segundo permite introducir una cadena de texto con longitud cero cuando se establece esta propiedad como *Sí*. En nuestro caso, dependiendo de cada atributo se configuran sus propiedades en función de las necesidades. Más adelante se detalla.

General	Búsqueda
Tamaño del campo	255
Formato	
Máscara de entrada	
Título	
Valor predeterminado	
Regla de validación	
Texto de validación	
Requerido	Sí
Permitir longitud cero	No
Indexado	No
Compresión Unicode	Sí
Modo IME	Sin Controles
Modo de oraciones IME	Nada
Alineación del texto	General

**Figura 14. Propiedades de tipo Texto corto**

- **Sí/No:** Tipo de dato booleano, almacena valores *Verdadero* y *Falso*. Todas sus propiedades se dejan con su valor por defecto.

General	Búsqueda
Formato	Sí/No
Título	
Valor predeterminado	No
Regla de validación	
Texto de validación	
Indexado	No
Alineación del texto	General

**Figura 15. Propiedades de tipo Sí/No**

- Fecha/Hora: Este tipo de dato permite guardar fechas y horas. Como aspecto destacable, existe una propiedad que permite mostrar un selector de fecha permitiendo elegir un día concreto de una pequeña vista de calendario. Su formato se establece como *Fecha corta*.

General	Búsqueda
Formato	
Máscara de entrada	
Título	
Valor predeterminado	
Regla de validación	
Texto de validación	
Requerido	Sí
Indexado	No
Modo IME	Sin Controles
Modo de oraciones IME	Nada
Alineación del texto	General
Mostrar el Selector de fecha	Para fechas

**Figura 16. Propiedades de tipo Fecha/Hora**

## 5.2 Relaciones

Los objetos que nos proporciona MS Access para establecer las claves ajenas son las relaciones, un término que puede llevar a confusión ya que, en el modelo relacional de bases de datos, una relación es lo que en MS Access se conoce como tabla.

Para crear estas relaciones hay que acceder al menú *Herramientas de base de datos* y dentro de él, la opción *Relaciones*, mostrándose una ventana vacía. Una vez en ella, hay que añadir las tablas que se quiere relacionar eligiendo la opción *Agregar Tablas* dentro del menú *Diseño*.



**Figura 17. Agregar tablas**

Aparecerá un listado de las tablas creadas en la base de datos, de la cual se debe elegir las que formarán parte de las distintas relaciones, en este caso en particular se seleccionan todas las tablas. Una vez añadidas las tablas necesarias, para crear cada una de las relaciones, hay que seleccionar los atributos que forman la clave principal de una tabla A y arrastrarlos hasta otra tabla B que tiene como clave ajena sobre la A esos mismos atributos, apareciendo una ventana como la que se puede ver en la Figura 18 en la que en la parte izquierda se pueden ver los atributos seleccionados que forman la clave principal de A y en la parte de la derecha se deben elegir los atributos de la tabla B que forman la clave ajena sobre A. El orden en el que aparecen los atributos es relevante ya que deben de corresponderse en las dos tablas, por ejemplo, si el primer atributo de la tabla A es el código del paciente y el segundo es el código de la enfermedad, el orden en el que aparecen los atributos de la tabla B, debe de ser el mismo.

Asimismo, existen tres opciones de configuración de la relación que se está creando:

- *Exigir integridad referencial.* Es conveniente marcar esta opción, ya que, por ejemplo, no es posible que se esté llevando el seguimiento de un paciente por parte de una especialidad que no existe.
- *Actualizar en cascada los campos relacionados.* También conviene marcar esta opción porque si se modificara algún dato relativo a un paciente, automáticamente se modificaría ese dato en el resto de las tablas que estuvieran relacionadas con él.
- *Eliminar en cascada los campos relacionados.* Esta opción también es importante marcarla ya que, si se eliminara la información personal de un paciente, no es necesario mantener todo lo relativo a ese paciente, por lo que al tener activada esta opción, se borraría toda la información relacionada con ese paciente de forma automática.



**Figura 18. Configuración de la relación.**

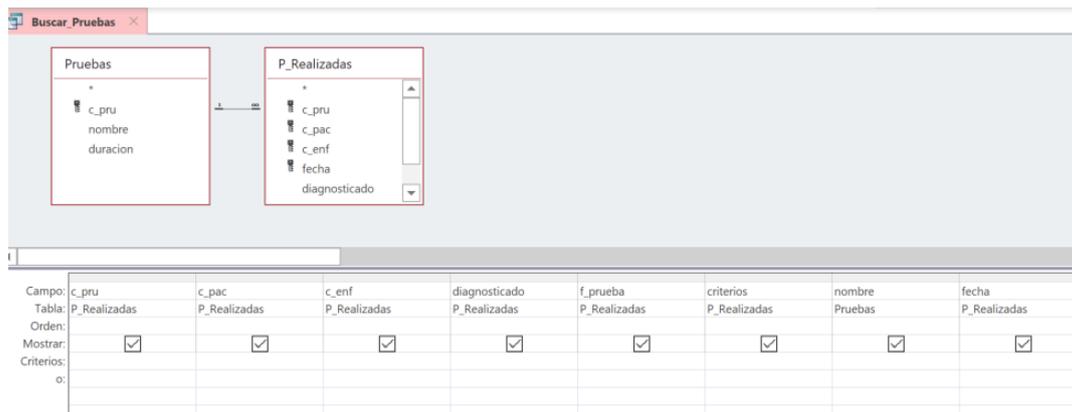
Todo el proceso descrito anteriormente se debe realizar para cada una de las claves ajenas existentes en el diseño lógico presentado en el punto 4.2, quedando como resultado de la creación de todas ellas un esquema como el de la Figura 8.

### 5.3 Consultas

Como se ha indicado en un punto anterior en el que se trataba el diseño de las consultas, se han creado consultas para recuperar información para plasmarla en los formularios y, por otra parte, otras se han creado para crear los informes.

El proceso de creación de una consulta ha comenzado por acceder a la opción “*Diseño de consulta*” del menú “*Crear*”. Posteriormente, del listado que aparece a la derecha de la ventana, se seleccionan las tablas de las que se desea obtener información, mostrándose las posibles relaciones existentes entre ellas. A continuación, se detalla el proceso de creación de cada una de las consultas:

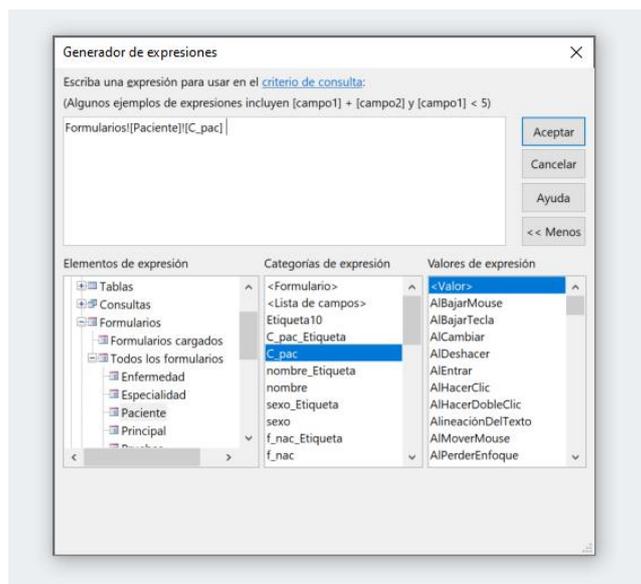
“*Buscar Pruebas*”: Una vez seleccionada la tabla *Pruebas*, se selecciona el atributo nombre para que aparezca como resultado de la consulta. Asimismo, se seleccionan de la tabla *P\_Realizadas* todos sus atributos (*c\_pru*, *c\_pac*, *c\_enf*, *diagnosticado*, *f\_prueba* y  *criterios*). El aspecto en este instante del diseño de la consulta es el de la Figura 19.



**Figura 19. Consulta buscar pruebas**

Ya que se pretende que solamente muestre la información del paciente actual en ese momento, se añade un criterio en el atributo *c\_pac* para que solamente aparezca la información indicada cuando el código del paciente actual del “Formulario Paciente” coincida con este atributo. También se aplica un filtro para que solamente aparezcan las pruebas realizadas relativas a una enfermedad en concreto, por lo que se añade otro criterio en el atributo “*nombre*”, haciéndolo coincidir con un valor elegido de un cuadro combinado existente en el “Subformulario P\_Realizadas”.

Para añadir estos criterios de forma más sencilla, hay que acceder a la opción “Generador” del menú, apareciendo una ventana en la que hay que explorar entre los formularios para seleccionar el campo que contiene el valor que debe de coincidir con el de los registros devueltos por la consulta, como se puede observar en la Figura 20.



**Figura 20. Generador de expresiones**

Una vez introducidos estos criterios, el aspecto del diseño de la consulta es el que se muestra en la Figura 21.

Campo:	c_pru	c_pac	c_enf	diagnosticado	f_prueba	criterios	nombre	fecha
Tabla:	P_Realizadas	P_Realizadas	P_Realizadas	P_Realizadas	P_Realizadas	P_Realizadas	Pruebas	P_Realizadas
Ordenar:								
Mostrar:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
Criterios:		[Formularios![Paciente][C_pac]					[Formularios![Paciente][Subformulario P_Realizadas][Formulario][cc_pruebas]	
o:								

**Figura 21. Consulta buscar pruebas**

“Consulta Enfermedades”: El proceso a seguir para esta consulta es exactamente igual que en la anterior, pero en este caso se selecciona la tabla *Enfermedad*, de la que se quiere recuperar el atributo *nombre*, y la tabla *Sufre*, de la que se quieren recuperar todos sus atributos (*c\_pac*, *c\_enf*, *ojo*, *loc\_interna*, *tipo*, *extensión*, *recidiva*, *f\_recidiva*, *f\_curacion*, *secuelas*, *tamaño* y *fecha*). En esta consulta también se añaden criterios en los atributos *c\_pac* y *nombre* para hacerlos coincidir con el código del paciente del “Formulario Paciente” y con el nombre de la enfermedad por la cual se quiere filtrar, respectivamente. El resultado de este diseño es el que se puede observar en la Figura 22 y Figura 23.

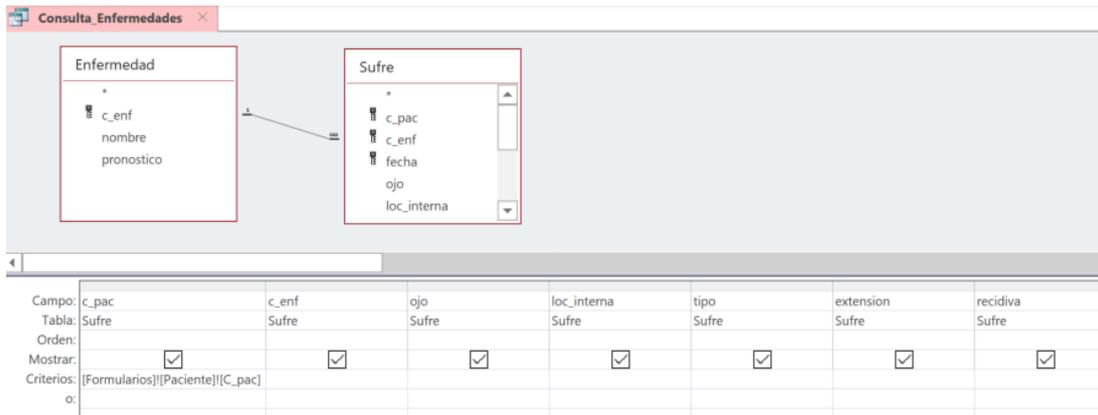


Figura 22. Diseño Consulta enfermedades 1

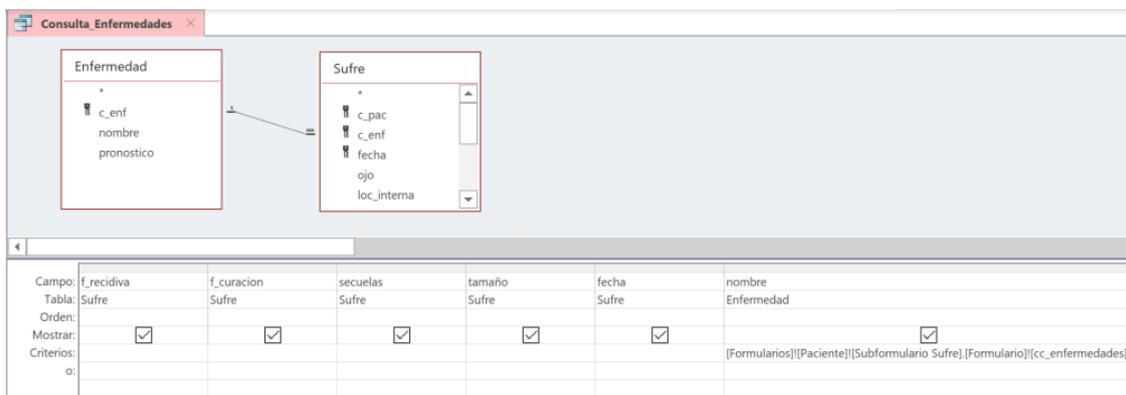


Figura 23. Diseño Consulta enfermedades 2

“Consulta Intervenciones”: En este caso se añaden al diseño de la consulta la tabla *Enfermedad*, de la que se selecciona el atributo *nombre*, la tabla *Intervención*, de la que se seleccionan todos sus atributos (*c\_esp*, *c\_pac*, *c\_enf*, *fecha*, *f\_intervención*, *médico* y *tipo*) y la tabla *Sufre* de la que no se selecciona ningún atributo, pero que es necesaria para poder relacionar las dos anteriores. En el caso de que no se añadiera esta última tabla, el diseño de la consulta quedaría como la Figura 24, en el que se puede observar que no existe relación alguna entre las dos tablas.

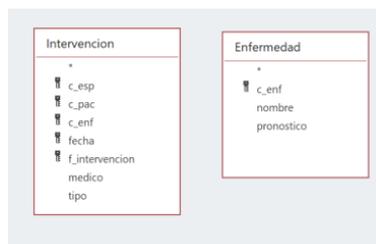
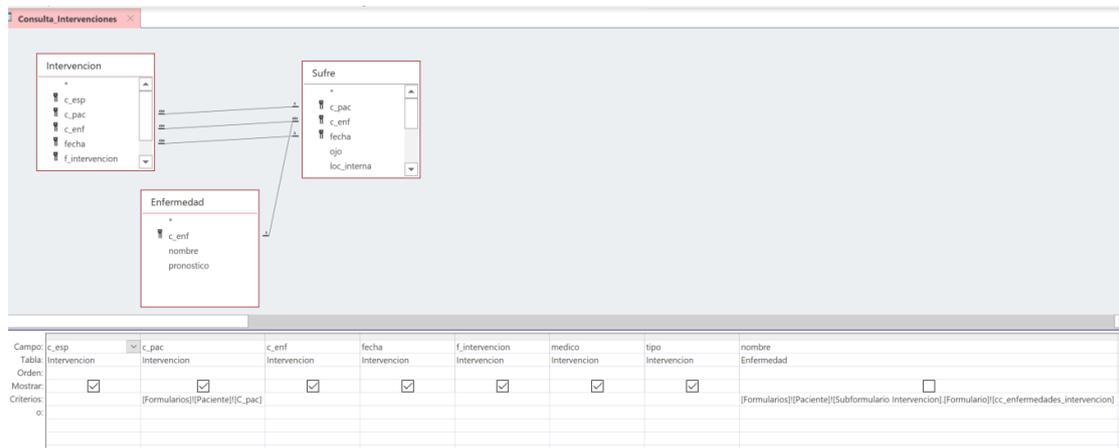


Figura 24. Tablas de consulta sin relación

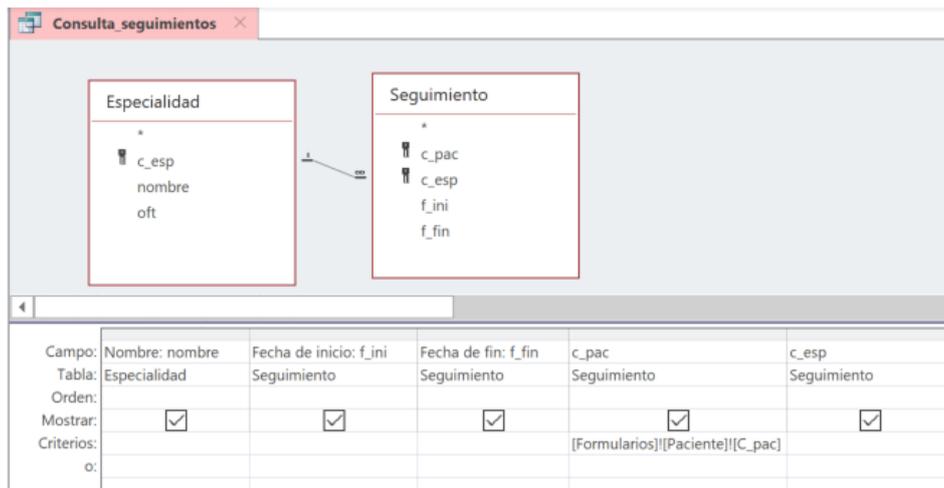
En esta consulta también se añaden dos criterios, el primero que coincida *c\_pac* con el código del paciente del “Formulario Paciente” y el segundo que hace de filtro para que solamente se vean las intervenciones relativas a una enfermedad en concreto seleccionada. El diseño resultante se puede observar en la Figura 25.



**Figura 25. Diseño consulta intervenciones**

“Consulta Seguimientos”: En esta consulta se desea recuperar información de las tablas *Seguimiento*, de la que se recuperan todos sus atributos (*f\_ini*, *f\_fin*, *c\_pac* y *c\_esp*), y *Especialidad* de la que se recupera únicamente el atributo *nombre*. En este caso solamente se establece un criterio, que coincida el código del paciente con el atributo *c\_pac*.

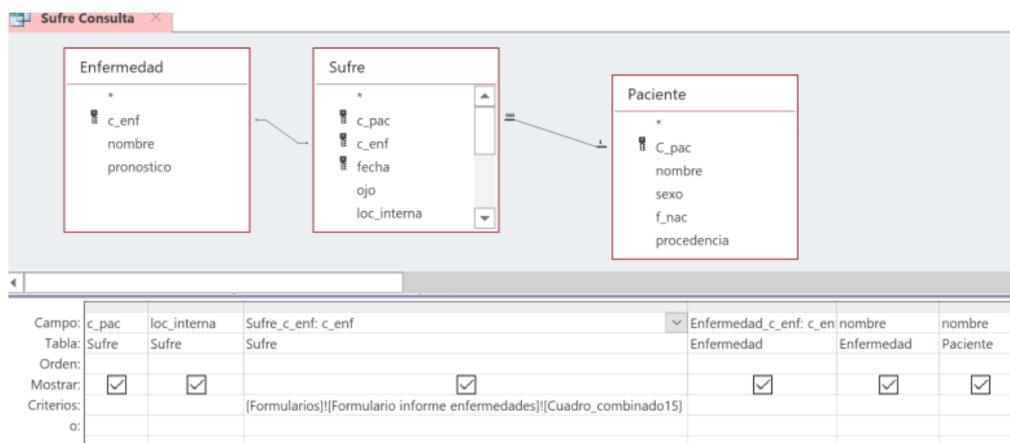
Asimismo, dado que se quiere mostrar la información de manera distinta a los casos anteriores, se opta por darle un nombre de visualización distinto a los atributos *nombre*, *f\_ini* y *f\_fin*, dejándolos como *Nombre*, *Fecha de inicio* y *Fecha de fin* respectivamente. Esta acción se lleva a cabo introduciendo el texto deseado precedido de dos puntos antes del nombre del atributo correspondiente. Se puede ver cómo queda el diseño de la consulta en la Figura 26.



**Figura 26. Diseño consulta seguimientos**

Las consultas creadas para poder generar los informes han sido principalmente centradas en las enfermedades que ha sufrido un paciente.

“Sufre Consulta”: Devuelve los atributos *c\_pac*, *loc\_interna* y *c\_enf* de la tabla *Sufre*, el nombre de la tabla *Paciente* y los atributos *c\_enf* y *nombre* de la tabla *Enfermedad*. El único criterio que se aplica a esta consulta es que el código de la enfermedad de los registros seleccionados debe coincidir con el de la enfermedad seleccionada del cuadro combinado desde el que se genera el informe que utiliza esta consulta.



**FIGURA 27. DISEÑO SUFRE CONSULTA**

“Enfermedad Consulta”: En esta consulta se recuperan los atributos *c\_enf* y *nombre* de la tabla *Enfermedad*, los atributos *c\_pac*, *c\_enf*, *fecha*, *f\_recidiva* y *recidiva* de la tabla *Sufre* y los atributos *c\_pac* y *nombre* de la tabla *Paciente*. Como criterios de la consulta se ha establecido que el código de la enfermedad

tiene que coincidir con el código de la enfermedad correspondiente seleccionada del cuadro combinado que está ubicado en el formulario desde el que se genera el informe que utiliza esta consulta. Asimismo, el atributo *recidiva* también tiene que coincidir con el campo correspondiente del formulario anterior. Es aspecto de esta consulta es muy similar a las vistas anteriormente por lo que a continuación únicamente se muestran los criterios de selección de ésta.

Enfermedad_c_enf: c_enf	recidiva
Enfermedad	Sufre
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Formularios][Formulario informe enfermedades][Cuadro_combinado15]	[Formularios][Formulario informe enfermedades][recidiva]

**Figura 28. Criterios consulta informe**

“Paciente Consulta Porcentaje”: En esta consulta se recuperan los atributos *c\_pac* y *nombre* de la tabla *Paciente*, el código del paciente de la tabla *Sufre* y el nombre y el código de la enfermedad de la tabla *Enfermedad*. Además de todo ello, se ha generado un nuevo atributo en la consulta en función de la edad del paciente en el momento del diagnóstico de la enfermedad seleccionada (de la misma manera que en las dos consultas inmediatamente anteriores). En este nuevo atributo llamado *Rango* se asignan los valores “Menos de 40”, “Entre 40 y 65” o “Más de 65” en función de la diferencia en años entre la fecha de nacimiento de cada paciente y la fecha de diagnóstico (atributo *fecha* de la tabla *Sufre*). La expresión que genera este atributo es la siguiente:

```
Rango: Conmutador (DifFecha("aaaa"; [f_nac]; [fecha]) <40; "Menos de 40"; DifFecha("aaaa"; [f_nac]; [fecha]) >65; "Mas de 65"; DifFecha("aaaa"; [f_nac]; [fecha]) >= 40 Y DifFecha("aaaa"; [f_nac]; [fecha]) <=65; "Entre 40 y 65")
```

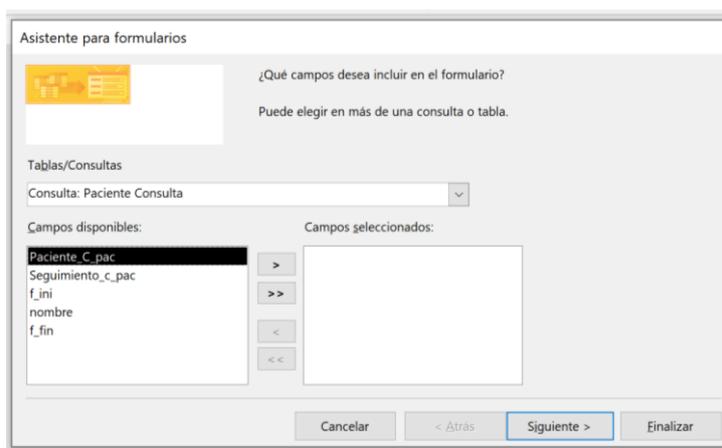
Por ejemplo, a un paciente nacido el 01/01/2000 y diagnosticado de la enfermedad elegida el día 06/03/2020, en el atributo *Rango* se le asignaría el valor “Menos de 40” ya que su edad en el momento del diagnóstico es de 20 años.

## 5.4 Formularios

Como se ha visto en uno de los puntos anteriores, los formularios representan la interfaz gráfica con la que el usuario va a interactuar,

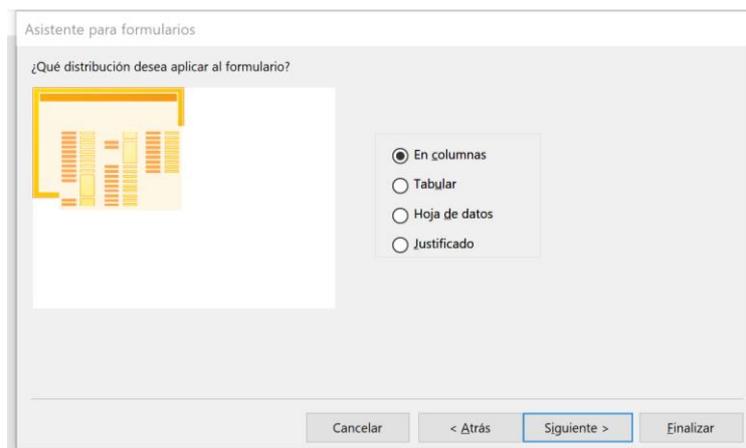
permitiéndole introducir, consultar y modificar la información almacenada en la base de datos de una forma más cómoda y comprensible.

La manera más cómoda de confeccionar los formularios es a través del asistente que proporciona Microsoft Access. Dentro del menú *Crear*, seleccionando la opción *Asistente para formularios* mostrándose una ventana en la que hay que elegir el origen de los datos y qué campos aparecerán en ese formulario Figura 29.



**Figura 29. Asistente formularios**

Una vez seleccionados los datos que aparecerán en el formulario, el asistente da la opción de elegir la distribución que tendrán estos datos, pudiendo elegir entre las opciones que aparecen en la Figura 30.



**Figura 30. Distribución de datos en formulario**

Por último, se ha de introducir el nombre del formulario y seleccionar si queremos que abra el formulario para comenzar a introducir información o, por el contrario, se quiere modificar el diseño.

**Figura 31. Último paso asistente formularios**

Una vez visto el proceso de creación de los formularios utilizando el asistente, se continuará viendo la creación de los formularios mencionados en un punto anterior de este documento.

El formulario *Principal* que es el primero que verá el usuario al iniciar la aplicación consta de dos partes, en la primera, situada a la izquierda de del formulario, se pueden ver cuatro botones que sirven para abrir los formularios correspondientes a los textos que aparecen en cada uno de ellos. La parte de la derecha corresponde a un pequeño formulario que nos permitirá generar distintos informes.

**Figura 32. Formulario Principal**

El formulario *Informe Especialidades* es el que se abre al pulsar el botón de *Gestión de informes* y en él se puede observar un cuadro combinado que nos

ofrece una lista de enfermedades y una casilla de verificación con el texto “Recidiva”. Estos dos objetos sirven para establecer los criterios sobre los que se va a generar el informe deseado pulsando el botón correspondiente (Figura 33).



**Figura 33. Formulario Gestión de informes**

El formulario *Enfermedad* sirve para poder introducir información sobre enfermedades que, posteriormente pueden ser vinculadas a pacientes. En él se pueden ver dos campos de texto en los que se introducen el nombre y el pronóstico de las enfermedades. Asimismo, se han añadido botones para poder agregar y eliminar registros, así como para guardar los cambios realizados, buscar un registro en concreto y por último un botón para cerrar el formulario.



**Figura 34. Formulario Enfermedad**

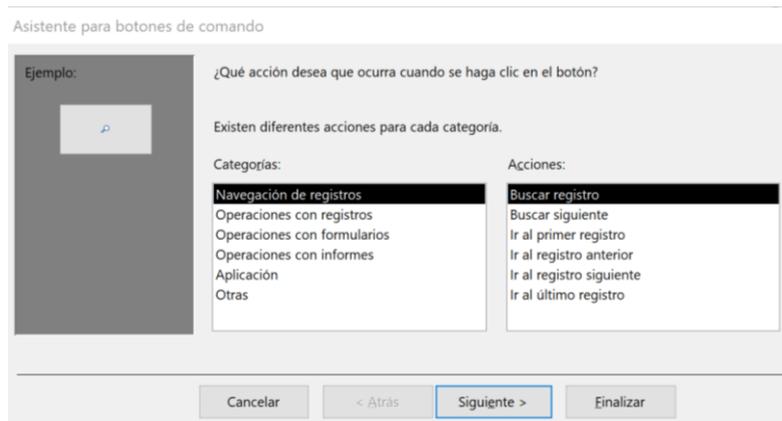
El proceso para la creación de los botones mediante el asistente que proporciona Microsoft Access es el siguiente:

En el menú *Diseño*, se elige la opción *botón* de la barra de herramientas.



**Figura 35. Opción botón**

Se elige el lugar donde se quiere posicionar el botón haciendo click y posteriormente aparece una ventana como la de la Figura 36 en la que se puede elegir el comportamiento del botón que se va a añadir.



**Figura 36. Asistente botones**

En los casos de los botones para añadir un registro, borrar el registro actual y guardar registro, se debe seleccionar de la lista de la izquierda la opción *Operaciones con registros* y posteriormente, las opciones *Agregar nuevo registro*, *Eliminar registro* y *Guardar registro* de la lista de la derecha respectivamente. Para el botón de buscar un registro, se debe seleccionar de la categoría *Navegación de registros* la acción *Buscar registro* y, por último, para el botón de cerrar formulario, se ha de seleccionar de la categoría *Operaciones con formularios*, la acción *Cerrar formulario*. Para finalizar la creación del botón, se introduce el texto que se quiere que muestre en el botón creado.

A continuación, se puede ver el aspecto del resto de formularios creados usando el mismo procedimiento que para el *Formulario Enfermedad*.

Edición de Especialidades

Nombre de especialidad

Oftalmológica

Nuevo Eliminar Guardar Buscar

Cerrar

**Figura 37. Formulario Especialidad**

Edición de Pruebas Médicas

Nombre

Duracion

Nuevo Eliminar Guardar Buscar

Cerrar

**Figura 38. Formulario Pruebas**

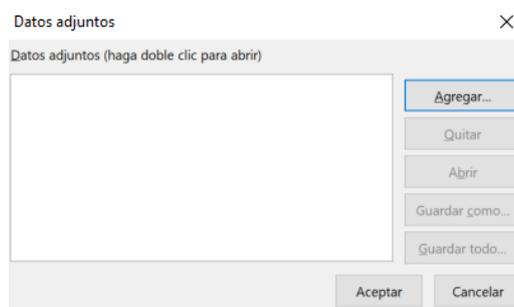
Antes de ver el formulario *Paciente*, se va a mostrar el aspecto final de los subformularios que están contenidos dentro de éste.

Guardar cambios Foto anterior Siguiete foto Nueva foto Eliminar foto

Tipo de foto

**Figura 39. Subformulario Foto**

En el subformulario para introducir fotografías, igualmente existen botones para añadir, borrar y guardar registros, pero también se han añadido dos botones para explorar entre las diferentes fotografías que pueda tener asociadas un paciente. La manera de agregar una fotografía es hacer doble click sobre el recuadro grande que posteriormente mostrará la imagen, viendo una ventana como la de la Figura 40.



**Figura 40. Agregar imagen**

Al pulsar sobre el botón “Agregar...” se explorará entre las carpetas del equipo para seleccionar la imagen deseada, pulsando *Aceptar* una vez se haya hecho.

En el *Subformulario Seguidimientos* se puede observar una lista en la que aparecerán los seguimientos que se le está realizando a un paciente en concreto, indicando los datos de cada columna en su encabezado correspondiente. Al pulsar sobre cualquiera de las filas de la lista, se mostrarán en los campos de la izquierda, los datos correspondientes a la fila seleccionada. Asimismo, se pueden ver botones para añadir, guardar o borrar registros, así como un botón para editar la información de los campos de texto que, al pulsar sobre él habilitará la edición de éstos.

**Figura 41. Subformulario Seguidimientos**

En el *Subformulario Sufre* lo primero que llama la atención es la existencia de un cuadro combinado el cual sirve de filtro para recuperar únicamente las enfermedades del tipo elegido, sufridas por el paciente actual. Igualmente se pueden ver botones para gestionar la edición de registros y dos botones más para la exploración entre los registros coincidentes con el filtro del cuadro combinado.

**Figura 42. Subformulario Sufre**

En el caso del *Subformulario P\_Realizadas* se pueden ver los datos relativos a las pruebas médicas que se le ha realizado al paciente que se muestra en ese momento. Al igual que en el *Subformulario Sufre* existe un cuadro combinado para realice un filtro y solamente muestre las pruebas realizadas del tipo seleccionado.

**Figura 43. Subformulario P\_Realizadas**

En el *Subformulario Intervención* aparece información referente a las intervenciones de distintos tipos que se le haya realizado al paciente actual. También se ha incluido un cuadro combinado para que aparezcan solamente las pruebas relacionadas con la enfermedad elegida y se han creado botones para gestionar la información de cada prueba realizada, así como para explorar entre los registros resultado de realizar el filtro del cuadro combinado.

**Figura 44. Subformulario intervención**

En este subformulario se ha creado además un nuevo botón que muestra información adicional en función del tipo de intervención, mostrándose un subformulario dentro de éste.

**Figura 45. Sub-subformulario observación**

**Figura 46. Sub-subformulario quirúrgico**

**Figura 47. Sub-subformulario Tratamiento medico**

En el caso del formulario *Paciente*, la información relativa al mismo es muy amplia por lo que se ha optado por crear los subformularios anteriores dentro de éste y gestionarlos mediante pestañas, conteniendo cada uno de ellos información concerniente a su área respectiva. El aspecto del formulario *Paciente* es el que se puede observar en la Figura 48.

**Figura 48. Formulario Paciente**

Todos los subformularios contenidos dentro de éste solamente mostrarán información relativa al paciente actual ya que, al utilizar las consultas definidas y detalladas anteriormente para cada uno de los subformularios, hacen coincidir el código del paciente del formulario *Paciente*, con el código del paciente de los subformularios respectivos, campo que se ha mantenido oculto en todos ellos al no ser de interés para el usuario que utilizará la aplicación.

En cada uno de los formularios y subformularios ha sido necesario programar funcionalidad para eventos que se puedan producir dentro de ellos, por ejemplo, cada vez que se pulsa el botón para editar la información de un formulario, es necesario habilitar o inhabilitar los campos de texto de éste. Para ello se ha creado un campo de texto oculto en cada formulario que guardará un valor dependiendo de si se puede editar o no y se ha añadido un evento en el botón correspondiente que lleva a la ejecución del siguiente código de Visual Basic para Aplicaciones (VBA). En la Figura 49 se puede ver el ejemplo del

código creado para el *Subformulario Sufre* en el que también se modifica el texto del interior del botón en función de si está habilitada o no la edición.

```
Private Sub boton_editar_Click()  
    Dim bloqueado As Boolean  
    bloqueado = Me.txt_bloqueo.Value  
  
    bloqueado = Not (bloqueado)  
    Me.txt_bloqueo = bloqueado  
  
    If bloqueado = True Then  
        Me.cc_enfermedad_sufre.Enabled = False  
        Me.ojo.Enabled = False  
        Me.loc_interna.Enabled = False  
        Me.tipo.Enabled = False  
        Me.extension.Enabled = False  
        Me.recidiva.Enabled = False  
        Me.f_recidiva.Enabled = False  
        Me.f_curacion.Enabled = False  
        Me.secuelas.Enabled = False  
        Me.tamaño.Enabled = False  
        Me.fecha.Enabled = False  
        Me.boton_editar.Caption = "Editar"  
    Else  
        Me.cc_enfermedad_sufre.Enabled = True  
        Me.ojo.Enabled = True  
        Me.loc_interna.Enabled = True  
        Me.tipo.Enabled = True  
        Me.extension.Enabled = True  
        Me.recidiva.Enabled = True  
        Me.f_recidiva.Enabled = True  
        Me.f_curacion.Enabled = True  
        Me.secuelas.Enabled = True  
        Me.tamaño.Enabled = True  
        Me.fecha.Enabled = True  
        Me.boton_editar.Caption = "Cancelar Edición"  
    End If  
End Sub
```

**Figura 49. Código edición campos**

Otro ejemplo de código generado para manejar eventos es el de la Figura 50, en el que en el *Subformulario P\_Realizadas*, cada vez que se selecciona una prueba en el cuadro combinado que sirve para filtrar, busque el nombre de la enfermedad para la que se realiza esa prueba y lo ponga en el campo Enfermedad de ese subformulario.

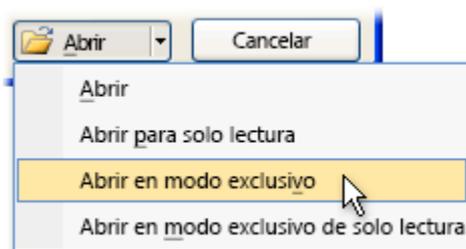
```
Private Sub cc_pruebas_AfterUpdate()  
    Me.Requery  
    Me.cc_enfermedad_pruebas = DLookup("[nombre]", "[Enfermedad]", "c_enf=" & Me.c_enf & "")  
    Me.cc_prueba_pruebas = Me.cc_pruebas  
End Sub
```

**Figura 50. Código buscar nombre enfermedad**

Para todos los formularios creados se han ocultado los botones de navegación y selectores de registro, ya que al haber creado los botones de exploración “*Anterior*” y “*Siguiente*”, no son necesarios, pudiendo llevar a confusión al usuario. Asimismo, la vista predeterminada para todos ellos se ha establecido en “*Un único formulario*”, mostrando solamente un formulario para cada registro. Otra de las configuraciones de los formularios ha sido no permitir la vista “*Hoja de datos*” que permite ver toda la información del formulario actual en forma de tabla, pudiendo también llevar a error o confusión del usuario.

## 6 Implantación

La implantación de la aplicación de bases de datos en el sistema final en el que se va a utilizar es simple y no tiene demasiada complicación. Se trata de trasladar el fichero de MS Access que contiene ambas partes del proyecto y depositarlo en el equipo del usuario, pero antes de ello, hay que proteger la aplicación con una contraseña que determinará el usuario que podrá tener acceso. Hay que abrir el fichero de MS Access en modo exclusivo entrando en el menú *Archivo*, opción *Abrir* y seleccionando dicha opción



**Figura 51. Apertura en modo exclusivo**

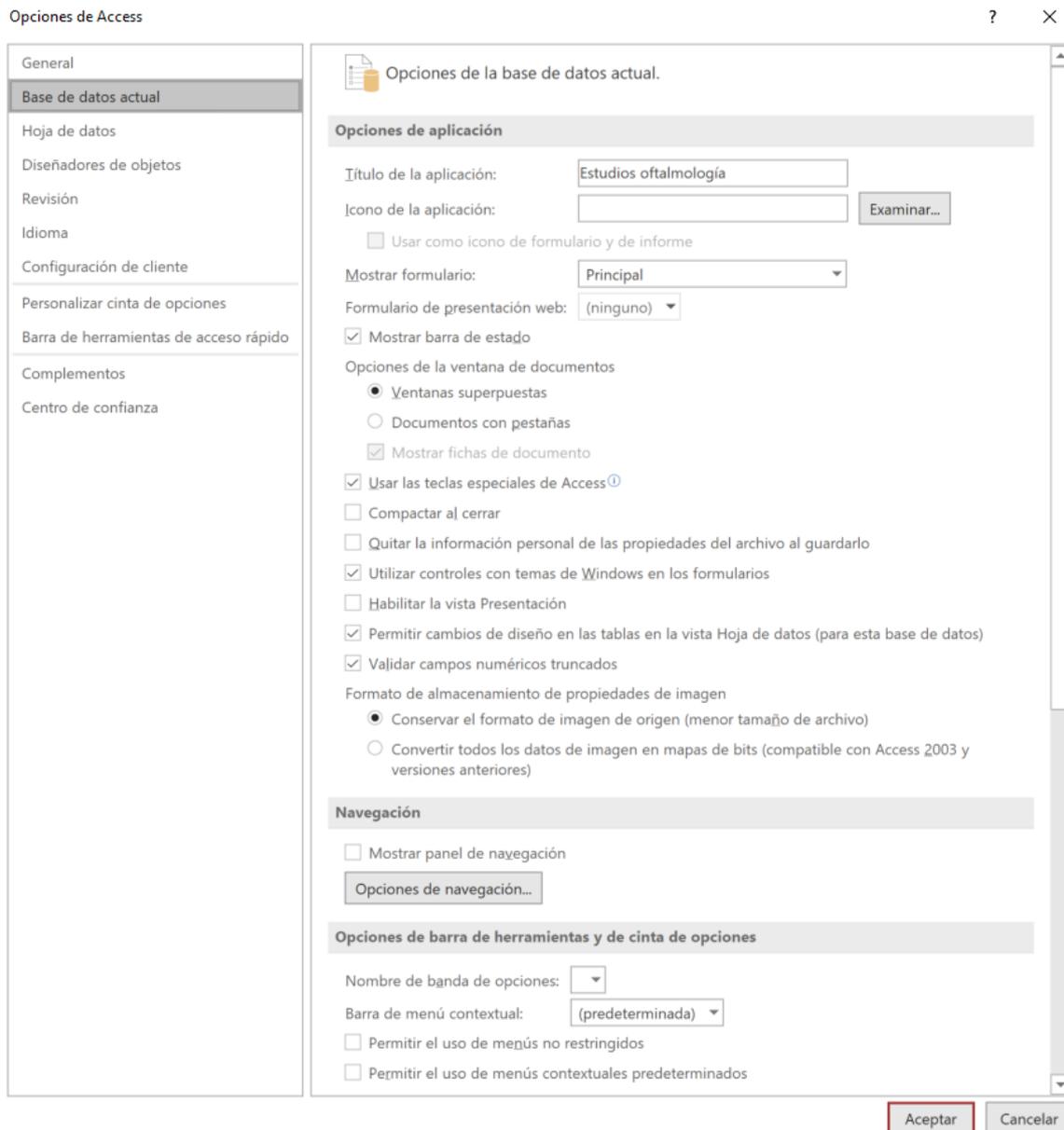
Cuando esté abierta, nuevamente en el menú *Archivo*, en la opción *Información*, se puede establecer una contraseña para proteger el fichero, introduciendo la palabra clave deseada y confirmándola.



**Figura 52. Opción cifrar base de datos**

Una vez protegida la base de datos hay que configurar el fichero de la aplicación de bases de datos en modo aplicación para que no se pueda modificar ningún aspecto del diseño ni de la configuración de la misma. Para este proceso, hay que acceder al menú *Archivo* y seleccionar la opción *Opciones* apareciendo una ventana de configuración. En esa ventana, en el panel de la izquierda se elige *Base de datos actual* y, en el panel de la derecha, se debe elegir el nombre

que tendrá la aplicación, el formulario que se abrirá al iniciar la misma (en el caso del presente proyecto es el formulario Principal) así como una serie de opciones que eviten que el usuario pueda variar el diseño o configuración como hemos citado anteriormente. El aspecto de esta ventana debe ser similar al de la Figura 53.



**Figura 53. Configuración aplicación final**

Una vez hecho, se programan las copias de seguridad para la recuperación de una posible pérdida de información o avería del sistema de almacenamiento en el que se ha dejado la aplicación. El proceso para la programación de éstas es el siguiente.

Hay que acceder a la opción de *Copia de Seguridad* que ofrece Microsoft Windows 10 y activar la realización de las copias de seguridad automática (cabe destacar que es necesario que haya un dispositivo de almacenamiento además del que contiene la aplicación que se desea respaldar) tal y como se indica en la Figura 54.

## Copia de seguridad

### Copia de seguridad con Historial de archivos

Realiza una copia de seguridad de tus archivos en otra unidad y restáuralos si los originales se han perdido, están dañados o se han eliminado.

Realizar una copia de seguridad automática de mis archivos

Activado

[Más opciones](#)

### **Figura 54. Configuración copia seguridad**

Una vez activado, entrando en *Más opciones* se selecciona el destino de la copia de seguridad, así como su frecuencia, el tiempo que se mantendrán dichas copias y las ubicaciones del disco que se va a respaldar (Figura 55).

## 🏠 Opciones de copia de seguridad

### Información general

Tamaño de la copia de seguridad: 0 bytes

Espacio total en Respaldo (D:) (D:): 14,8 GB

Todavía no se ha hecho ninguna copia de seguridad de los datos.

Hacer ahora una copia de seguridad

Realizar una copia de seguridad de mis archivos

Cada hora (predeterminado) ▾

Mantener las copias de seguridad

Para siempre (predeterminado) ▾

Hacer una copia de seguridad de estas carpetas



Agregar una carpeta



Oftalmología  
C:\Users\theau\Desktop

**Figura 55. Opciones copia de seguridad**

De esta manera, la aplicación queda lista para su utilización por parte del usuario final y quedando cubierta por cualquier incidencia que pueda ocasionar pérdida de información.

## 7 Pruebas

Para comprobar el correcto funcionamiento, tanto de la base de datos como de la aplicación que accede a ella, se ha realizado una prueba consistente en dar de alta un nuevo paciente en la base de datos, introduciendo sus datos personales.



**Datos de Paciente**

Número de historia: 1234

Nombre: Daniel Garcia Molero Buscar paciente

Sexo: Hombre Nuevo paciente Borrar paciente

Fecha de nacimiento: 26/10/1985

Procedencia: Casa Cancelar Edición Guardar datos paciente

**Figura 56. Datos de paciente**

Continuando por cada una de las pestañas que separan la información relativa al paciente actual en las distintas áreas.



**Datos de Paciente**

Número de historia: 1234

Nombre: Daniel Garcia Molero Buscar paciente

Sexo: Hombre Nuevo paciente Borrar paciente

Fecha de nacimiento: 26/10/1985

Procedencia: Casa Editar Guardar datos paciente

Fotos **Seguimientos** Enfermedades Pruebas Intervenciones

Guardar cambios Foto anterior Siguiete foto Nueva foto Eliminar foto

Tipo de foto: Imagen de prueba

**IMAGEN DE PRUEBA**

**Figura 57. Fotografías**

### Datos de Paciente

Número de historia:

Nombre:  Buscar paciente

Sexo:  Nuevo paciente Borrar paciente

Fecha de nacimiento:

Procedencia:  Cancelar Edición Guardar datos paciente

Fotos | Seguimientos | Enfermedades | Pruebas | Intervenciones

Especialidad:

Fecha inicio seguimiento:

Fecha fin seguimiento:

Nombre	Fecha de inicio	Fecha de fin
e3	01/06/2020	16/06/2020

Nuevo Seguimiento
Cancelar Edición
Guardar
Borrar seguimiento

**Figura 58. Datos de seguimientos**

### Datos de Paciente

Número de historia:

Nombre:  Buscar paciente

Sexo:  Nuevo paciente Borrar paciente

Fecha de nacimiento:

Procedencia:  Cancelar Edición Guardar datos paciente

Fotos | Seguimientos | Enfermedades | Pruebas | Intervenciones

Seleccione una enfermedad para filtrar:

Enfermedad:

Fecha diagnóstico:

Ojo:

Localización:

Tipo:

Extensión:

Recidiva:

Fecha de recidiva:

Fecha de curación:

Secuelas:

Tamaño:

Registro anterior
Registro siguiente

Nueva enfermedad
Cancelar Edición
Guardar enfermedad
Borrar enfermedad

**Figura 59. Datos de enfermedades**

**Datos de Paciente**

Número de historia:

Nombre:  Buscar paciente

Sexo:

Fecha de nacimiento:  Nuevo paciente Borrar paciente

Procedencia:  Cancelar Edición Guardar datos paciente

Fotos | Seguimientos | Enfermedades | Pruebas | Intervenciones

Seleccione un tipo de prueba para filtrar:

Enfermedad:

Fecha de la enfermedad:

Prueba:

Diagnosticado:

Fecha de la prueba:

Criterios

Registro anterior Registro siguiente

Nueva prueba Editar Guardar prueba Borrar prueba

**Figura 60. Datos de pruebas médicas**

**Datos de Paciente**

Número de historia:

Nombre:  Buscar paciente

Sexo:

Fecha de nacimiento:  Nuevo paciente Borrar paciente

Procedencia:  Cancelar Edición Guardar datos paciente

Fotos | Seguimientos | Enfermedades | Pruebas | Intervenciones

Seleccione una enfermedad para filtrar

Enfermedad:

Especialidad:

Fecha de intervención:

Médico responsable:

Tipo de intervención:

Registro anterior Registro siguiente

**Detalles tipo Quirúrgico**

Tipo:

Márgenes libres:

Guardar Cerrar Borrar Cirugia

Cancelar Edición Nueva Intervención Ver detalles Guardar intervención Borrar intervención

**Figura 61. Datos de intervenciones**

Una vez introducido todos los datos se comprueba que se han guardado correctamente, para ello se cierra la aplicación volviéndola a abrir y buscando el



paciente por número de historia y, una vez encontrado, se revisa que toda la información guardada anteriormente aparece en cada uno de los subformularios.

The image shows a web application interface for patient data management. The main form is titled "Datos de Paciente" and contains the following fields: "Número de historia" (1234), "Nombre" (Daniel Garcia Molero), "Sexo" (Hombre), "Fecha de nacimiento" (26/10/19...), and "Procedencia" (Casa). There are buttons for "Buscar paciente", "Nuevo paciente", and "Borrar paciente". A "Guardar cambios" button is also visible. Below the form are tabs for "Fotos", "Seguimientos", "Enfermedades", and "Pruebas". A modal window titled "Buscar y reemplazar" is open, showing search options: "Buscar" (1234), "Buscar en:" (Documento actual), "Coincidir:" (Hacer coincidir todo el campo), "Buscar:" (Todos), and checkboxes for "Coincidir mayúsculas y minúsculas" and "Buscar los campos con formato". Buttons for "Buscar siguiente" and "Cancelar" are also present in the modal.

Figura 62. Buscar paciente

## 8 Conclusiones

El objetivo de este proyecto era la creación de una aplicación de bases de datos mediante la cual se pudiera guardar información relativa a una población de pacientes oftalmológicos para posteriormente poder realizar estudios sobre enfermedades, tratamientos o técnicas quirúrgicas.

El objetivo se ha cumplido, ya que se ha diseñado una base de datos capaz de almacenar la información necesaria para llevar a cabo distintos estudios oftalmológicos, se ha desarrollado una aplicación que permita la introducción y modificación de la información en la base de datos y se han creado informes que obtienen información necesaria para los estudios mencionados. Todo ello implica una mejora importante respecto al método de trabajo anterior en el que se utilizaban hojas de cálculo para almacenar información y llevar a cabo pequeños estudios estadísticos con las limitaciones que el uso de este tipo de aplicaciones conlleva.

Para poder realizarlo ha sido necesario pasar por una serie de procesos, empezando por la especificación de requisitos, pasando por las distintas fases del diseño de la base de datos y de la aplicación, la posterior implementación de ambas y terminando en la implantación del sistema en el equipo final. Después de todo el proceso, considero que las fases más importantes son las dos primeras. Una buena especificación de requisitos es fundamental, ya que, si no se recogen todos los requisitos o se hacen de manera incorrecta, hará que se realice un mal diseño y posterior implementación, lo cual llevará a tener que realizar cambios tanto en el diseño como en la implementación con el consiguiente sobrecoste. Una vez se tiene la base de una buena especificación de requisitos, es crucial realizar un buen diseño ya que es sobre lo que se va a basar la posterior implementación. Un mal diseño puede llevar a una implementación que no satisfaga los requisitos del cliente

En cuanto a la implementación, Microsoft Access ha sido el software elegido ya que es del que actualmente dispone el servicio de oftalmología que explotará toda la información almacenada. El hecho de que ya se disponga de la suite Microsoft Office conlleva un ahorro importante por parte del cliente, que no tiene que adquirir una licencia de software. Microsoft Access proporciona



asistentes que facilitan en gran medida la creación de, por ejemplo, formularios, expresiones, consultas o macros. A pesar de estos asistentes, ha sido necesario modificar el diseño de algunas soluciones proporcionadas por éstos, por lo que se ha tenido que aprender ciertos aspectos que se desconocían de Microsoft Access, como pueden ser el lenguaje de programación Visual Basic para Aplicaciones o la sintaxis de las expresiones que puede resultar un tanto compleja en un principio. Sobre estos aspectos como también, algún otro como puede ser lo que MS Access llama *Relaciones*, no se tenía ningún conocimiento previo, ya que nunca se había trabajado con este software ni con ninguna de las tecnologías que utiliza por lo que ha sido necesario un proceso de aprendizaje de todas ellas, profundizando en algunas funciones a medida que era necesaria su utilización.

La aplicación creada puede verse como una aplicación clásica en el sentido que no se conecta a ningún servidor web ni está alojada en la nube, sino que se implanta en un equipo y se trabaja únicamente desde él, pero es el tipo de aplicación que el cliente necesita, no siendo necesario el uso de esas tecnologías u otras más actuales.

### **8.1 Relación del proyecto con los estudios cursados**

En el desarrollo de este proyecto he tenido que aplicar conocimientos adquiridos a lo largo de los cursos del Grado de Ingeniería Informática.

En primera instancia, en la gestión del proyecto he tenido que aplicar conocimientos de la asignatura con el mismo nombre, por ejemplo, realizar una planificación tanto temporal como de recursos de este, un presupuesto teórico teniendo en cuenta el número de horas inicialmente estimadas, así como la identificación de posibles riesgos que pudieran afectar a la planificación inicial y sus posibles soluciones.

En la fase de especificación de requisitos, los conocimientos aplicados de la asignatura “Diseño centrado en el usuario”, en la que se enseñan técnicas para la identificación de los requisitos tanto funcionales como no funcionales.

En el diseño e implementación de la base de datos, las asignaturas implicadas en aportarme conocimientos fueron “Bases de datos y sistemas de

información” y “Tecnologías de bases de datos”, las cuales me han servido para poder realizar un diseño correcto de la base de datos pudiendo identificar las relaciones entre los distintos objetos, determinar el tipo de datos necesario para cada atributo o establecer restricciones para algún atributo en concreto.

En el diseño interfaz gráfica, o lo que es lo mismo, los formularios de Microsoft Access intervinieron las asignaturas “Diseño centrado en el usuario” e “Interfaces Persona-Computador”, aportando a los distintos formularios un aspecto homogéneo y haciendo que la aplicación sea usable, lo que quiere que la información se lee de manera rápida y los menús y la funcionalidad es sencilla con lo que el usuario está cómodo y satisfecho con su uso.

En el desarrollo de ciertas funcionalidades de la aplicación ha sido necesario aplicar conocimientos relativos a la programación, en lo que han intervenido varias asignaturas, pero la principal de ellas ha sido “Introducción a la informática y a la programación” que ha proporcionado los principios de la programación. En menor medida, pero también me ha aportado conceptos como el de *evento* han influido las asignaturas “Tecnología de sistemas de información en la red” o “Desarrollo web”.

## **8.2 Ventajas obtenidas**

Una vez implantada la aplicación de bases de datos, se obtienen varios beneficios respecto a los métodos utilizados anteriormente, los cuales se basaban en la utilización de Microsoft Excel para almacenamiento de información y posterior explotación estadística. El principal beneficio es que es posible almacenar información relativa a distintos campos de la oftalmología en el mismo sistema, el cual es capaz de mantener una relación entre todos ellos, pudiendo así realizar estudios de forma sencilla en el que se vean implicados más de uno.

Otro de los aspectos de mejora es el ámbito de la búsqueda de la información relativa a un paciente ya que, buscando bien por número de historia clínica o bien por nombre, de un vistazo y con la sencilla exploración entre pestañas se pueden ver todos los datos referentes a ese paciente, ahorrando tiempo en este proceso.

### **8.3 Desarrollos futuros**

Los futuros desarrollos o ampliaciones que se pueden desarrollar sobre este proyecto pueden consistir en la creación de distintos informes con su correspondiente consulta que muestre la combinación de datos requerida en cada caso. Esto no supondría una complejidad excesiva ya que la base de datos ya está creada y la información ya está relacionada de forma que se pueden recuperar todos los datos almacenados de la forma deseada.

Otra posible ampliación puede consistir en ampliar el ámbito de aplicación de este proyecto abarcando a más especialidades médicas. En este cambio para añadir nuevas especialidades, habría que agregarlas a la tabla de Especialidades médicas, aunque es posible que hubiera que almacenar información no planteada en el esquema actual lo que supondría la modificación de alguna de las tablas o incluso de las relaciones entre ellas, dependiendo de los cambios realizados.

## 9 Referencias

- Apuntes de bases de datos y tecnologías de la información de la UPV (Celma, M.; Casamayor, J. C.; Mota, L.; Bases de datos relacionales. Pearson, Prentice Hall, 2003)
- Página web de soporte de Microsoft Office (<https://support.office.com/es-es/article/especificaciones-de-access-0cf3c66f-9cf2-4e32-9568-98c1025bb47c>)
- BOE num 924, de 6 de diciembre de 2018.
- Sueldo de ingeniero informático recién graduado (<https://universidadeuropea.es/blog/cuanto-gana-un-ingeniero-informatico#:~:text=El%20salario%20de%20un%20ingeniero,1.000%2D1.200%20euros%20netos%20mensuales>).

## 10 Glosario:

- SGBD: Sistema de gestión de bases de datos.
- UML: Unified Modeling Language.
- SQL: Structured Query Language.
- OMT: Object modeling technique.
- OOSE: Object oriented software engineering.
- Relación: Conjunto de datos estructurados propios del Modelo Relacional.
- Relación (MS Access): Vínculo que sirve de enlace entre dos tablas.