

DESARROLLO DE INTERFACES

2º DAMV 2023/2024

– PROYECTO FINAL –



# MUSEUM

---

MIGUEL PARDO NAVARRO





## ÍNDICE

DESCRIPCIÓN DEL PROYECTO.....	2
ESTUDIO DE NECESIDADES.....	2
DISEÑO DEL ESQUEMA RELACIONAL .....	3
CASOS DE USO .....	4
ASPECTOS TÉCNICOS.....	6
ETAPAS DE DESARROLLO .....	7
PRUEBAS .....	8
IMPLANTACIÓN .....	15
CONCLUSIONES .....	20
PROPUESTAS DE MEJORA.....	20
ANEXO.....	21

## DESCRIPCIÓN DEL PROYECTO

De cara a este proyecto final que concluirá nuestro paso por la asignatura de Desarrollo de Interfaces, necesitaba un tema con el que me sintiera cómodo, tanto en el sentido de que no fuera algo exageradamente complicado que me resultase un sobreesfuerzo, pero que fuera algo lo suficientemente interesante que me permitiese mostrar mis virtudes creativas y, en definitiva, me motivase a dar lo mejor de mí a la hora de desarrollar una interfaz. Y bajo estos requisitos, qué mejor tema a tratar que el mismísimo **arte**.

Así es como surgió la idea de hacer una aplicación que gestionase un **museo** de arte, centrándome en un principio en algunos de los cuadros más importantes de nuestra historia u otros que considere. No resulta un tema complicado de gestionar, pues al final simplemente trabajamos con las **obras** que formarán parte de nuestro museo y con los **autores** que las han llevado a cabo.

Ambas entidades contarán con una serie de atributos que nos ayudarán a conocer más información sobre ellas: el título y el nombre, la fecha de publicación y la de nacimiento, etc. Un atributo que jugará un papel clave en nuestro programa será el de **imagen**, ya que, si vamos a trabajar con cuadros, qué menos que poder visualizarlos.

Aparte de ser un asunto entretenido sobre el que investigar, una aplicación para un museo puede abrir mucho la puerta a diseñar una interfaz más atractiva y visual de las que estamos acostumbrados a ver en las actividades realizadas durante el curso. De esta manera, no deja de ser una idea asequible pero que supone un reto a la altura de lo esperado, sobre todo en el apartado gráfico y en el manejo de imágenes.

El nombre elegido para el programa es "**MuseumM**", nombre genérico y simple pero elegante y muy estético gracias a su simetría visual. La aplicación usará principalmente colores dorados sobre fondos oscuros que los resalten. La idea de este proyecto es llevar a cabo una aplicación que cumpla con todos los requisitos de funcionalidad propios de un programa informático, pero que no deje de lado la estética y belleza características del arte.

## ESTUDIO DE NECESIDADES

Bien, si nos paramos a estudiar qué debe tener una aplicación para un museo de arte, no tardaremos mucho en llegar a la conclusión de que la presentación visual es un aspecto muy importante. Nuestro programa debe ser tan claro como llamativo, permitiéndonos una gestión sencilla de los elementos de nuestro museo sin olvidarnos de que estamos tratando con asuntos artísticos.

Para ello será muy importante el trato con imágenes en nuestra interfaz, ya que éstas nos permitirán darle a la aplicación un ambiente que nos haga sentir como si de una galería de arte se tratase. A la hora de mostrar las obras almacenadas en nuestro museo, es esencial poder visualizar directamente los cuadros y no solo sus datos.

Puesto que vamos a trabajar con obras de arte, hay un atributo que no puede faltar en ninguna de ellas: su autor. Nuestro programa debe ser capaz de registrar a todo artista que cuente con obras en el museo, y de relacionar cada obra con su correspondiente creador. De la misma manera, si accedemos a los datos de los autores, debemos ver qué cuadros ha pintado cada uno.

## DISEÑO DEL ESQUEMA RELACIONAL

El esquema relacional de la base de datos consistirá en tres tablas: las principales **obra** y **autor**, y una extra que nos permitirá gestionar los **usuarios (user)** que acceden a la aplicación. De manera provisional, las tablas tendrán los siguientes atributos:

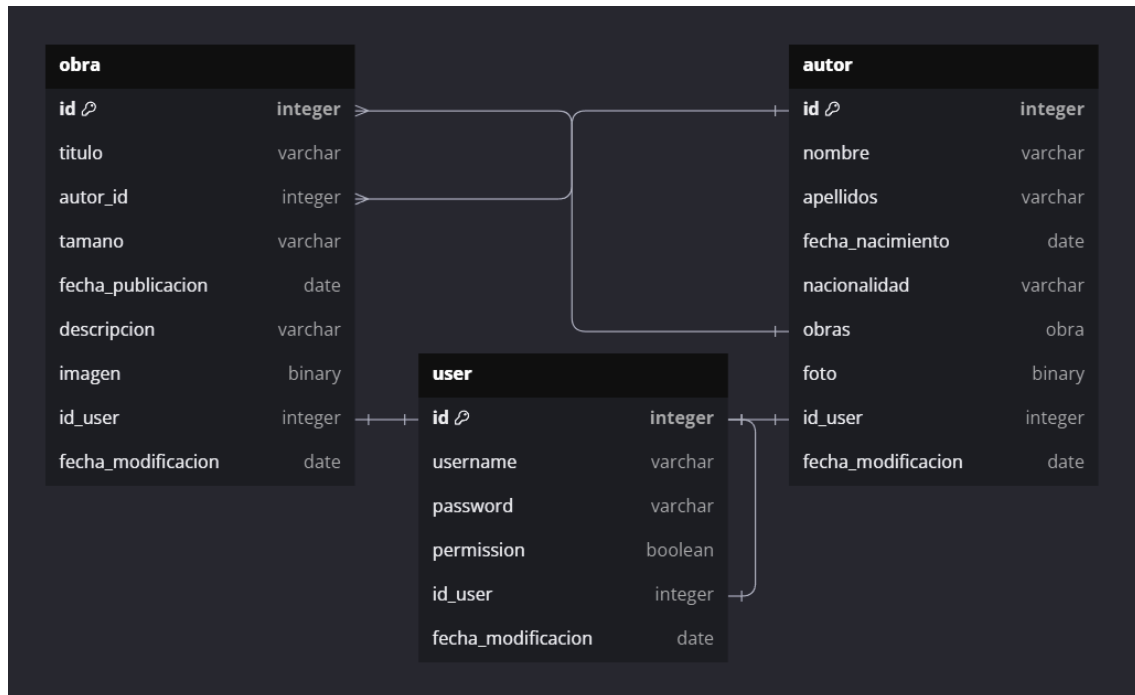
- **Obra:**
  - Id
  - Título
  - Autor (relación Many2One)
  - Tamaño
  - Fecha de Publicación
  - Descripción
  - Imagen
  - Id del usuario que la añade/modifica por última vez (relación One2One)
  - Fecha de la última modificación
- **Autor:**
  - Id
  - Nombre completo
  - Fecha de nacimiento
  - Nacionalidad
  - Foto
  - Id del usuario que lo añade/modifica por última vez (relación One2One)
  - Fecha de la última modificación
- **User:**
  - Id
  - Username
  - Password
  - Permission
  - Id del usuario que lo añade/modifica por última vez (relación One2One)
  - Fecha de la última modificación

Cabe a destacar las **relaciones** entre nuestras tablas, en especial las que relacionan las obras con sus autores y viceversa. Una obra tendrá un único autor, mientras que un autor podrá tener muchas obras.

En cuanto a las relaciones de ambas con la tabla **user**, se almacenará el id del usuario que haya introducido cada registro junto a la fecha en la que esto ocurre. Si la obra o el autor en cuestión son modificados, se sobrescribirán estos dos datos por los que sean pertinentes en dicha ocasión.

Debemos mencionar también que en la tabla de usuarios se registrarán (aparte del id) el nombre de usuario, su contraseña y un atributo *boolean* que nos permitirá diferenciar entre los usuarios

normales y los que tienen permisos de administrador. Un **usuario administrador** será capaz de crear y gestionar usuarios, mientras que aquellos que no gozan de privilegios no serán capaces de realizar esta opción. Igual que para obras y autores almacenamos el id del user que lo añade o modifica y la fecha, aquí haremos lo mismo.

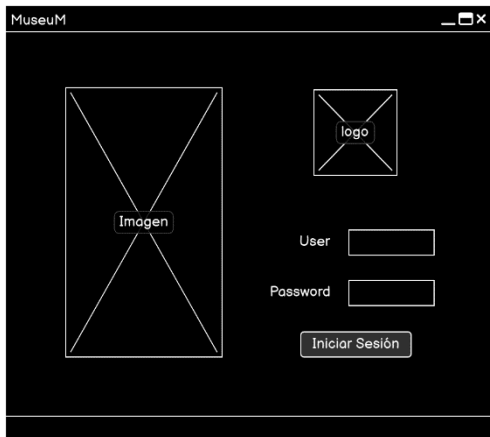


Esquema relacional de las tablas de la base de datos (algunos campos pueden variar). Hecho con dbdiagram.io<sup>1</sup>.

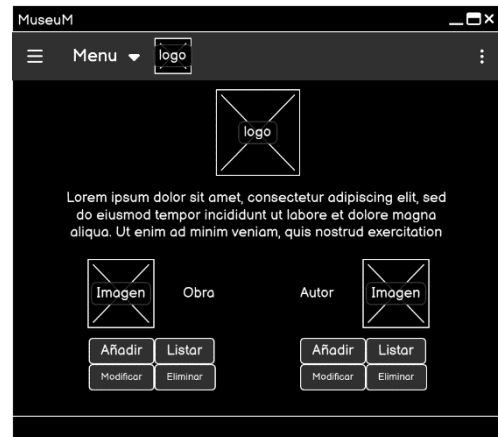
## CASOS DE USO

Los casos de uso del programa serán similares a los de cualquier programa de este estilo, permitiéndonos realizar las acciones CRUD sobre las tablas previamente definidas. Este esquema provisional se llevó a cabo gracias a la herramienta *balsamiq*<sup>2</sup>.

La aplicación comenzará en la pantalla de **inicio de sesión**, donde introduciremos nuestro usuario (normal o administrador) y la contraseña. Una vez hecho el *login*, nos mostrará la pantalla **principal** del programa, que describirá brevemente el proyecto y nos ofrecerá las opciones para realizar las distintas opciones que ofrece.

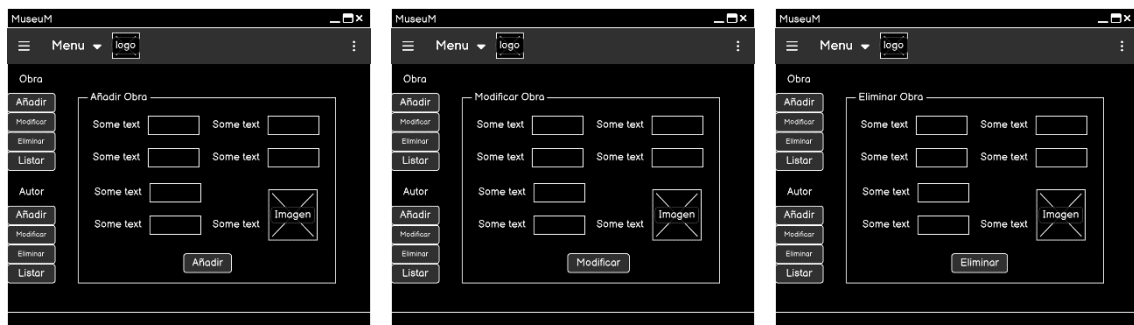


Pantalla de inicio de sesión.



Pantalla principal de la aplicación.

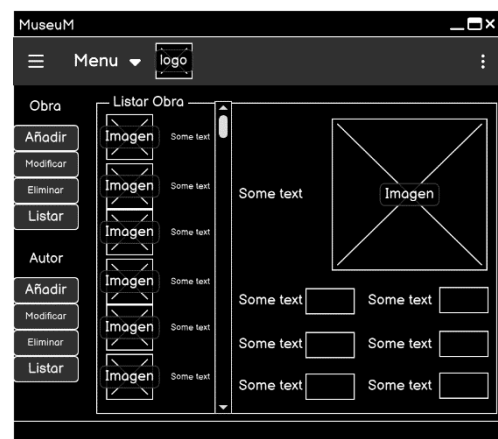
Tendremos para ambas de nuestras tablas sus respectivas pantallas que nos permitirán **añadir**, **modificar**, **eliminar** y listar sus registros. Las tres primeras tendrán una interfaz muy similar que nos permita ver y trabajar con los campos que componen el respectivo registro.



Pantallas de añadir, modificar y eliminar obras (las de autor serán igual).

En cuanto a la opción de **listar**, tendremos una tabla en la que visualizaremos todos los registros de la tabla con su imagen y título/nombre. La idea es poder reconocer las obras más fácilmente al mostrar la imagen directamente.

Una vez seleccionemos una fila, se mostrarán todos sus datos y podremos ver la imagen en grande. En el caso de los autores, se enseñarán todas las obras que tenga almacenadas.



Pantalla de listar obras.



Esto no se trata más que de un esquema provisional que facilitará el inicio del desarrollo, que no obligatoriamente seguirá estas ideas a rajatabla, aunque mantendrá una línea similar.

## ASPECTOS TÉCNICOS

El proyecto *Museum* se ha desarrollado como una aplicación WPF en Visual Studio, usando el lenguaje C# para el código detrás de los archivos XAML. Esta elección tecnológica permitió crear una interfaz gráfica atractiva y funcional, con una estructura de base de datos relacional para gestionar obras, autores y usuarios de manera eficiente.

Mediante la implementación del paquete Nuget de MySQL<sup>3</sup> para proyectos de este estilo, podemos hacer una gestión óptima de nuestra base de datos y los propios datos que le darán forma a la aplicación. Crearemos una clase *Conexion* que nos permitirá el acceso a los datos tras indicar los detalles de la conexión y, mediante clases DAO, escribiremos el código que formulará las consultas necesarias para que la aplicación tenga el funcionamiento esperado.

Hablando de la base de datos, ésta ha sido creada mediante la herramienta XAMPP, que nos permitirá mediante phpMyAdmin la gestión de bases de datos tipo MariaDB, como la que hemos utilizado. La base de datos, *museum.sql*, constará de las tablas, campos y restricciones explicadas en apartados anteriores.

El archivo .sql enlazado contendrá un registro de ambas tablas a modo de prueba (si incluía muchos puede causar problemas al importar debido al tamaño de las imágenes) y dos usuarios: uno con permisos de administrador, que será el usuario maestro **admin**, con contraseña **1234**; y otro sin permisos llamado user1, con la misma contraseña.

Los datos de la conexión son:

```
MySqlConnection conexion = new MySqlConnection();

static string servidor = "127.0.0.1";
static string db = "museum";
static string usuario = "root";
static string password = "";
static string puerto = "3306";

string cadenaConexion = "server=" + servidor + "; port=" + puerto + "; user id=" + usuario
    + "; password=" + password + "; database=" + db + ";";
```

*Datos de la conexión en el archivo Conexión.cs de la carpeta dao.*

Además del proyecto principal que aloja la aplicación, también se ha creado una aplicación auxiliar que lanza informes CrystalReport para mostrar el contenido de las tablas, utilizando los mismos datos de conexión. Esto se debe a las dificultades obtenidas para incluir este tipo de archivos de CrystalReport en el proyecto principal, que encontraron una fácil solución en crear un nuevo proyecto en Visual Studio que sirviese como complemento.

## ETAPAS DE DESARROLLO

Para determinar las etapas y actividades necesarias para el desarrollo del proyecto, me he visto guiado por las entregas parciales que debíamos realizar y las pautas que éstas indicaban. No obstante, el ritmo del proyecto se ha visto afectado por la gran carga de trabajo de este tramo final del curso. No de manera negativa, sino que en previsión a las dificultades que esto pudiera haber provocado, se decidió avanzar lo suficiente en este proyecto en las primeras etapas de su desarrollo para poder centrar la atención en otros proyectos y exámenes posteriores, para finalmente terminar y poner fin a este mismo proyecto.

De esta manera, el inicio del desarrollo comenzó en la última semana de enero de 2024, donde sentarían las bases del proyecto se realizaría el planteamiento lógico de todo lo relativo a este. Se decidió la idea sobre la que trataría, el nombre, el diseño de logos y colores, etc. En especial, debemos mencionar el planteamiento del esquema relacional de la base de datos con la que trabajaríamos, así como el boceto de las distintas pantallas que tendría finalmente la aplicación. En definitiva, esta primera etapa se trata de una etapa de planteamiento sobre la que se desarrollaría el resto.

Una vez estaban claras las ideas que definirían el proyecto, era hora de ponerse manos a la obra y empezar a crear la aplicación. La segunda etapa iniciaría con el objetivo de dar forma a las pantallas básicas de la aplicación: la de inicio de sesión y la pantalla principal de la aplicación sobre la que se cargarán las demás. Además, para asegurar el funcionamiento óptimo del inicio de sesión, será necesario también crear una pantalla para la gestión de los usuarios que se conectarán a la aplicación mediante la base de datos. El objetivo de esta segunda etapa es el de crear la estructura fundamental de la aplicación, que servirá de base para desarrollar los siguientes apartados, ya más concretos.

El comienzo de la tercera etapa no tardó en demorarse después de completar la anterior, aprovechando la inercia creativa que se llevaba en esas mismas semanas. Por lo tanto, con una aplicación sólidamente estructurada, era hora de añadir los apartados que darían funcionalidad al proyecto: las pantallas de gestión de las tablas de nuestra base de datos, primero las de obra y después las de autor. Con esto conseguimos ser capaces de añadir, modificar, eliminar y listar los datos almacenados en ambas entidades, cumpliendo con los requisitos de funcionalidad esperados.

En poco más de dos semanas se había conseguido desarrollar las partes más importantes del proyecto, y, en previsión de las ajetreadas semanas que vendrían en nuestro curso con la llegada de exámenes y otros proyectos, se decidió dejar aparcado el desarrollo de la aplicación durante la segunda mitad de febrero, realizando únicamente pequeñas modificaciones que mejorasen y corrigiesen el rendimiento y el aspecto de la aplicación.

Al retomar la actividad de nuevo, a principios de marzo, ya con lo principal prácticamente cerrado, se puso el centro de atención en los extras que completarían el proyecto: la creación de una aplicación auxiliar encargada de lanzar informes sobre nuestros datos, un vídeo tutorial como ayuda al usuario y demás detalles que otorgan de mayor calidad a *MuseumM*. Esta sería la



última etapa del proyecto, centrada en completar la aplicación añadiendo las funcionalidades y componente que faltaban para que el resultado fuese tal y como se deseaba desde un principio.

Podríamos resumir las etapas en 4 fases, obviando el período de descanso de la actividad: planteamiento, desarrollo de la estructura, desarrollo de la funcionalidad principal e integración de características complementarias.

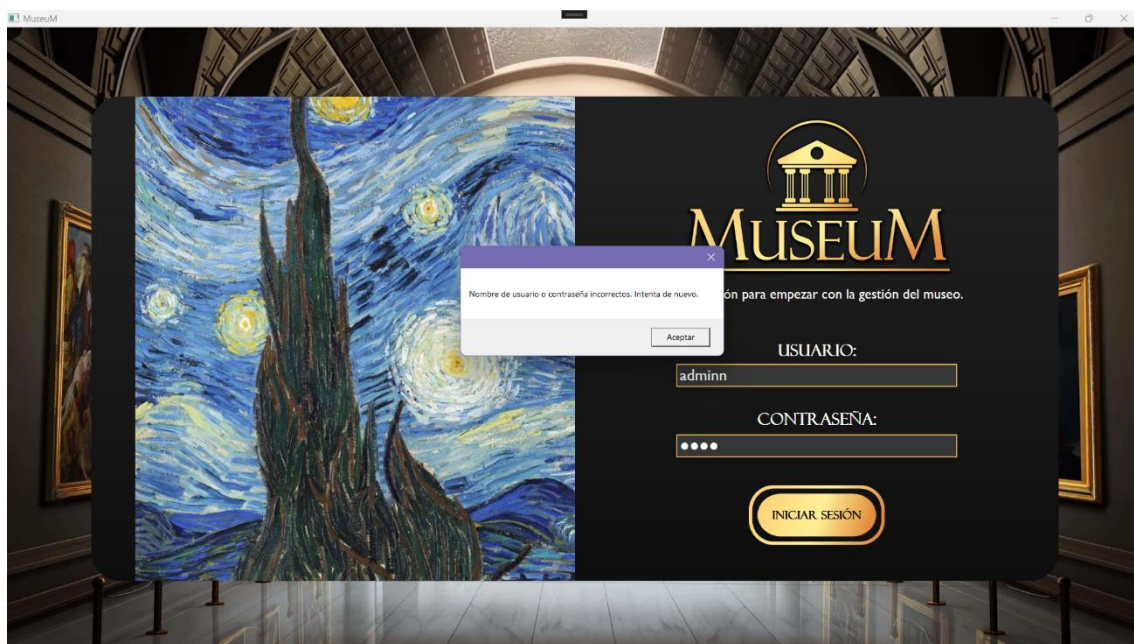
El proyecto se inició aproximadamente en las fechas de la última semana de enero de 2024, y llegó a su conclusión en la segunda semana del mes de marzo. En el siguiente esquema veremos el desarrollo de las diferentes etapas en el tiempo:



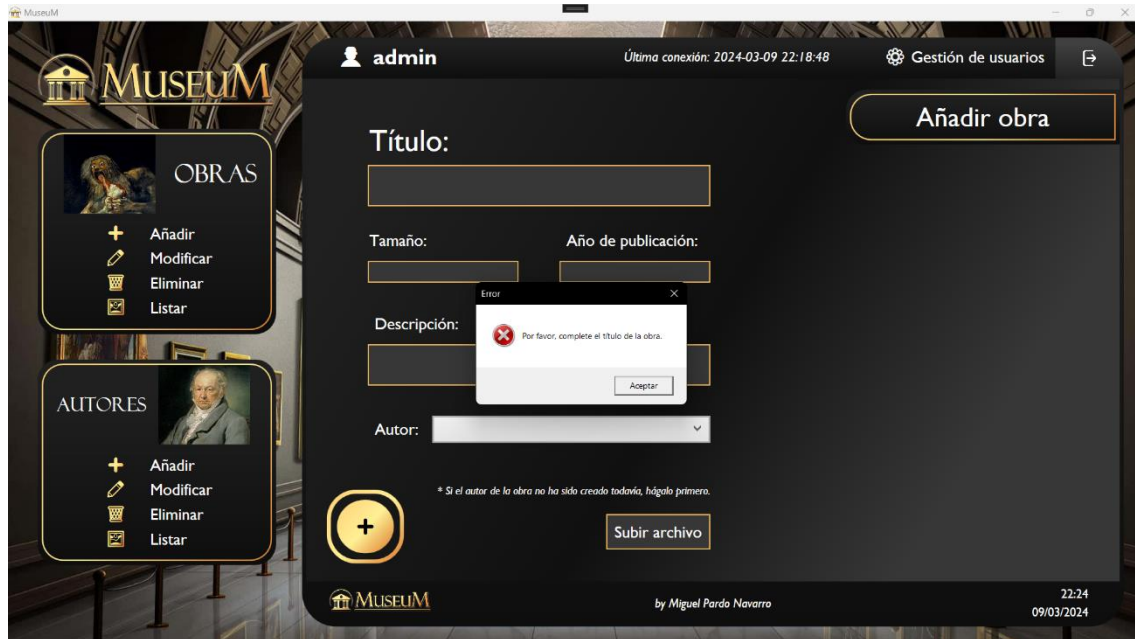
## PRUEBAS

En el apartado de pruebas realizaremos una serie de acciones que nos sirvan para comprobar que nuestra aplicación cumple con las funciones que hemos diseñado para cada elemento de la interfaz. Es decir, probaremos que si necesitamos insertar o modificar una obra o un autor los datos introducidos son los correctos y necesarios para mandarlos a la base de datos.

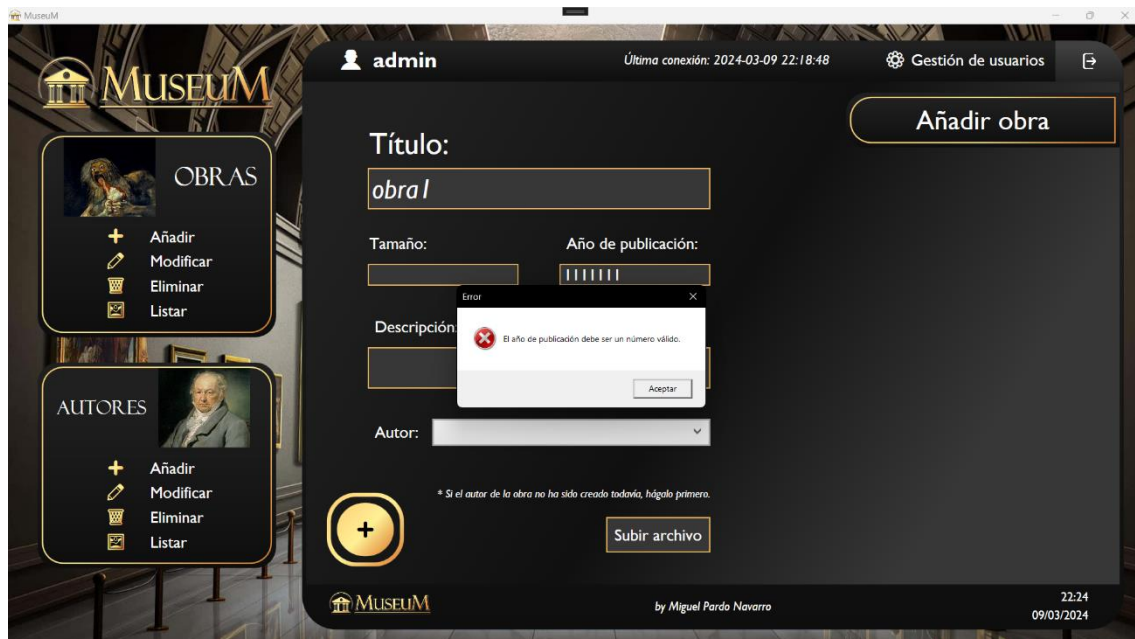
Para empezar, será necesario introducir un login correcto para poder iniciar sesión. El nombre de usuario y la contraseña deben coincidir con el del usuario existente en la base de datos.



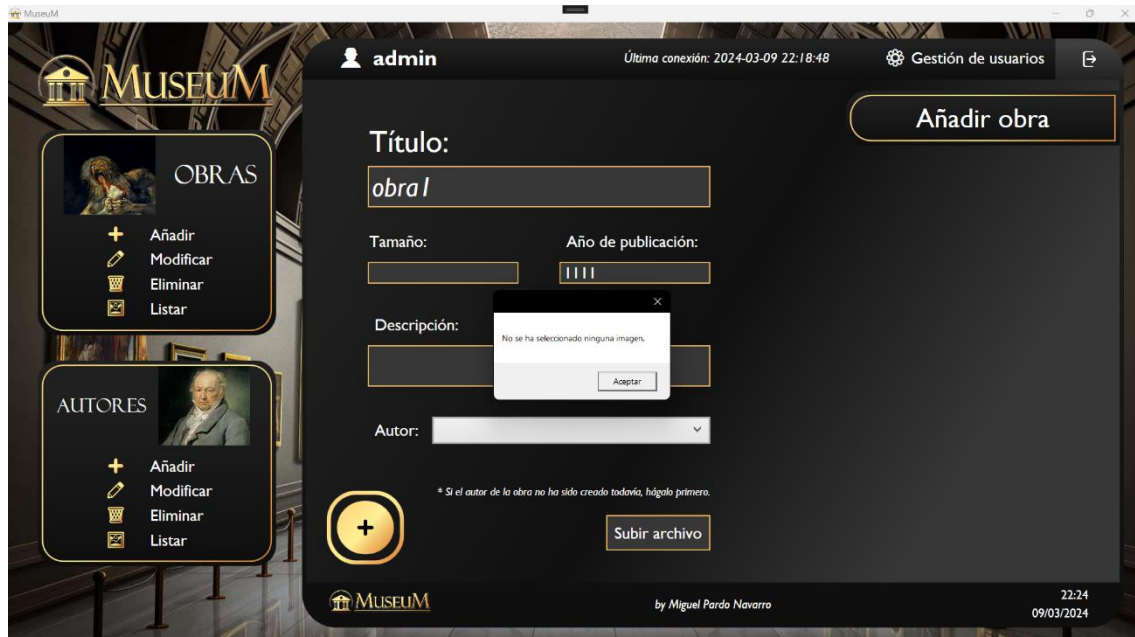
A la hora de insertar una obra, será necesario indicar su título, pues es la manera más sencilla de reconocer la obra y, por lo tanto, un campo obligatorio.



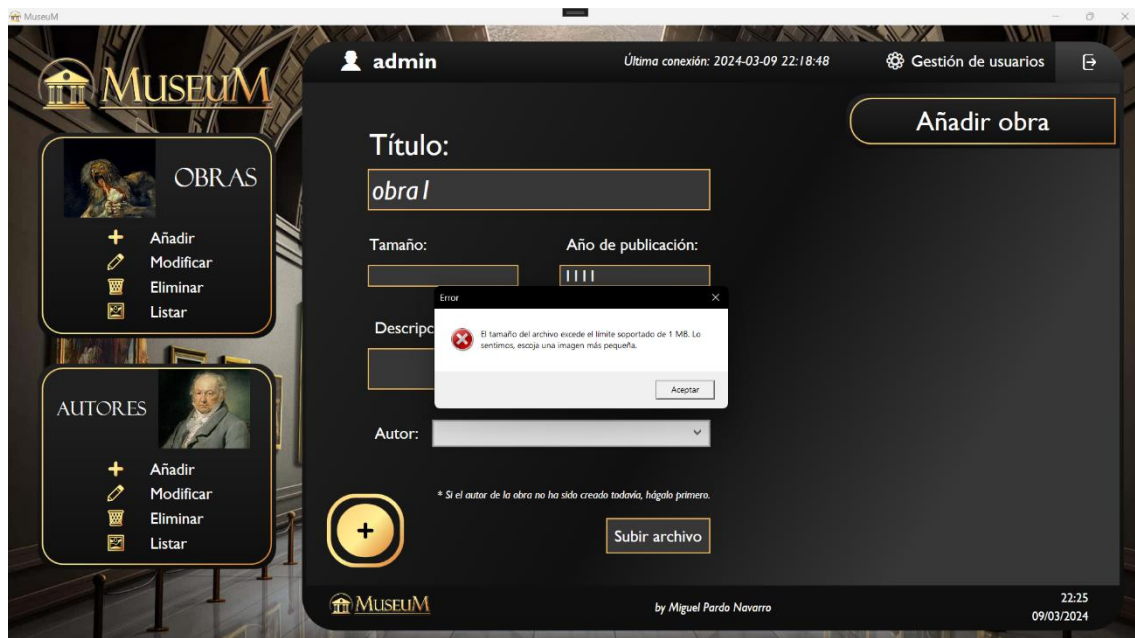
También será necesario introducir un año de publicación válido, es decir, un número entre el 0 y el 2050, para tener un margen de maniobra.



El campo de imagen será también obligatorio, pues es esencial en el funcionamiento de la aplicación el trato con las imágenes de las obras.

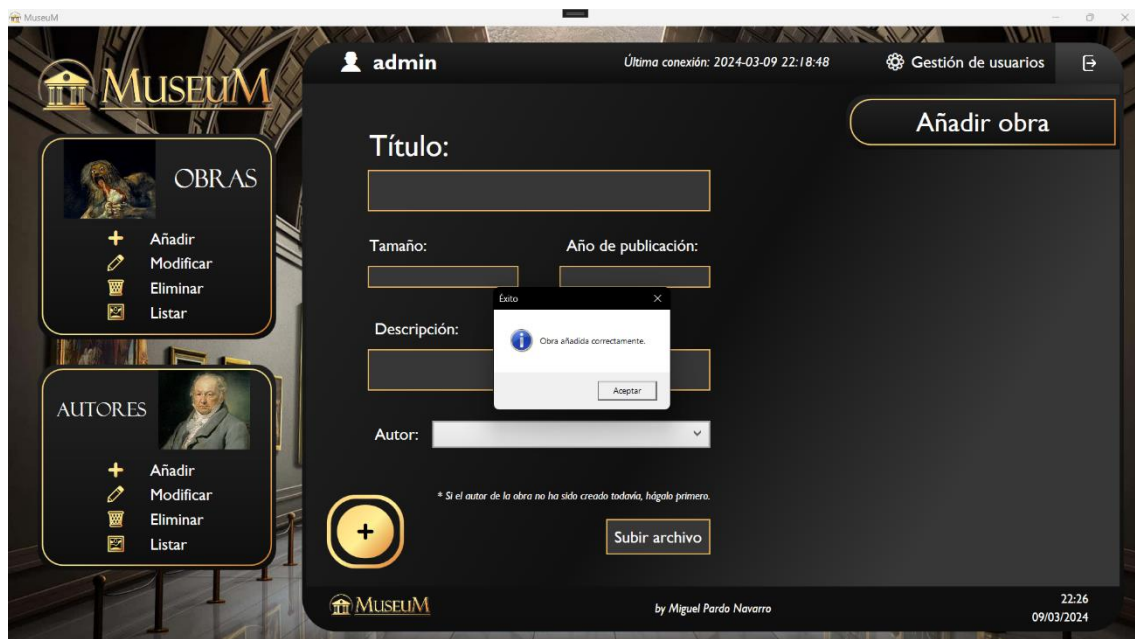


Hay que mencionar que las imágenes que añadamos deben ser siempre de un tamaño inferior a 1 MB, pues un tamaño muy elevado puede provocar errores a la hora de almacenar la imágenes en la base de datos.

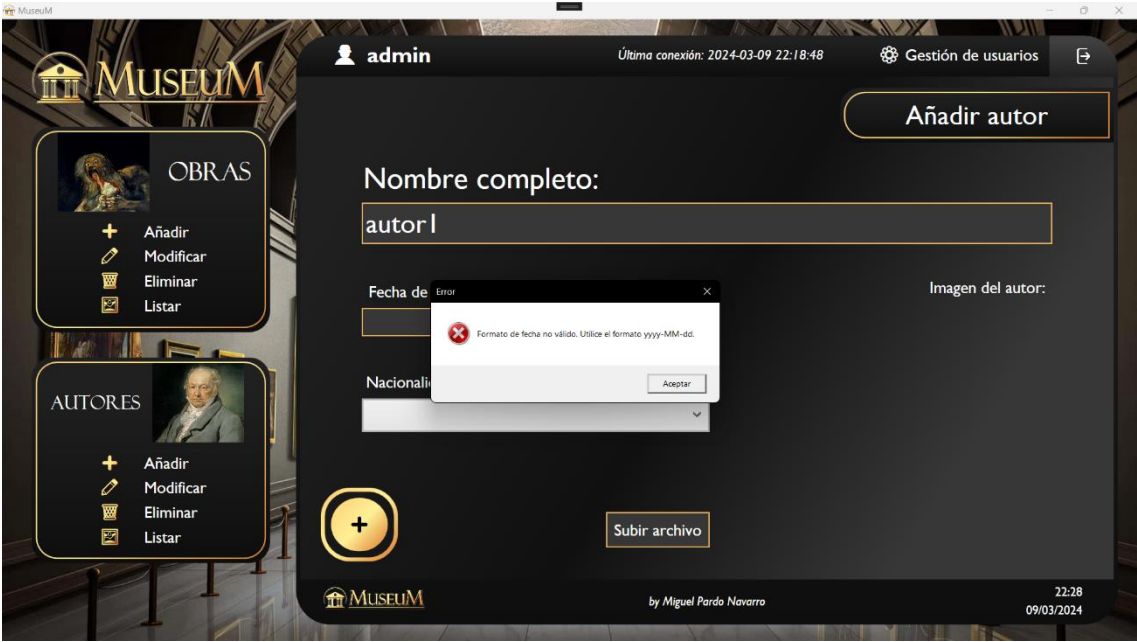
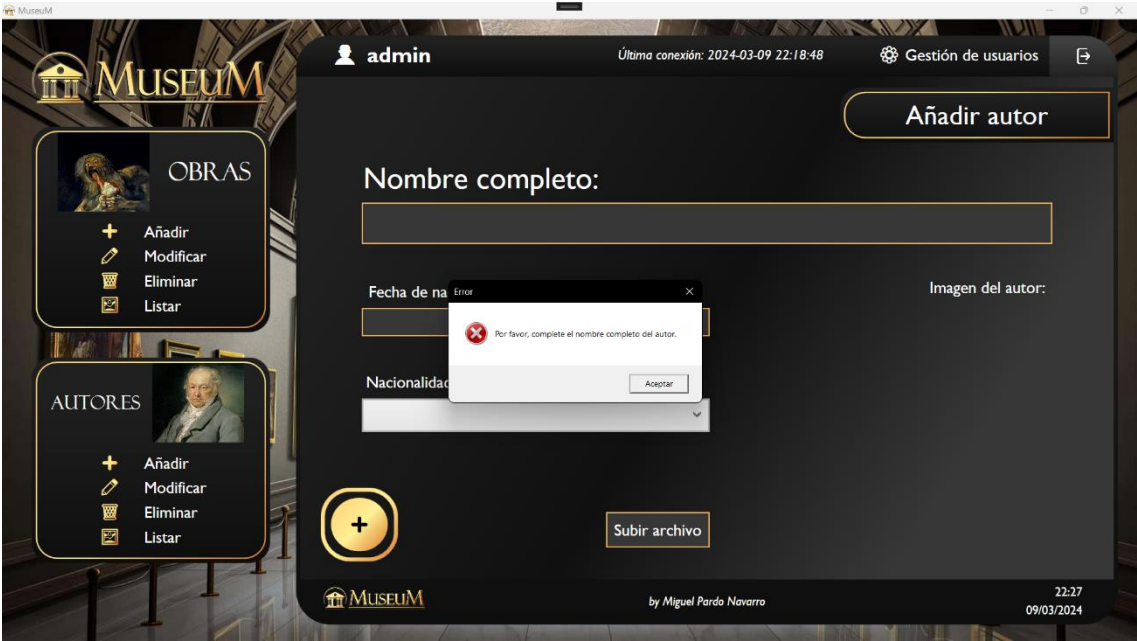


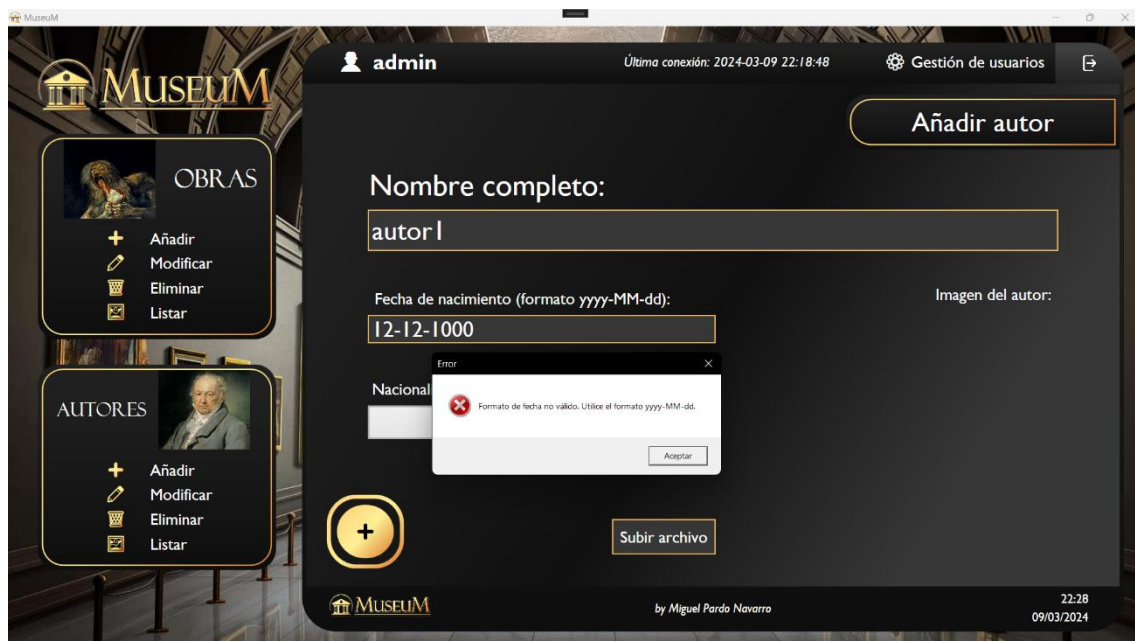


También será necesario escoger el autor en el combo box, y, una vez tengamos los campos obligatorios rellenados de forma correcta, se permitirá la inserción de la obra.



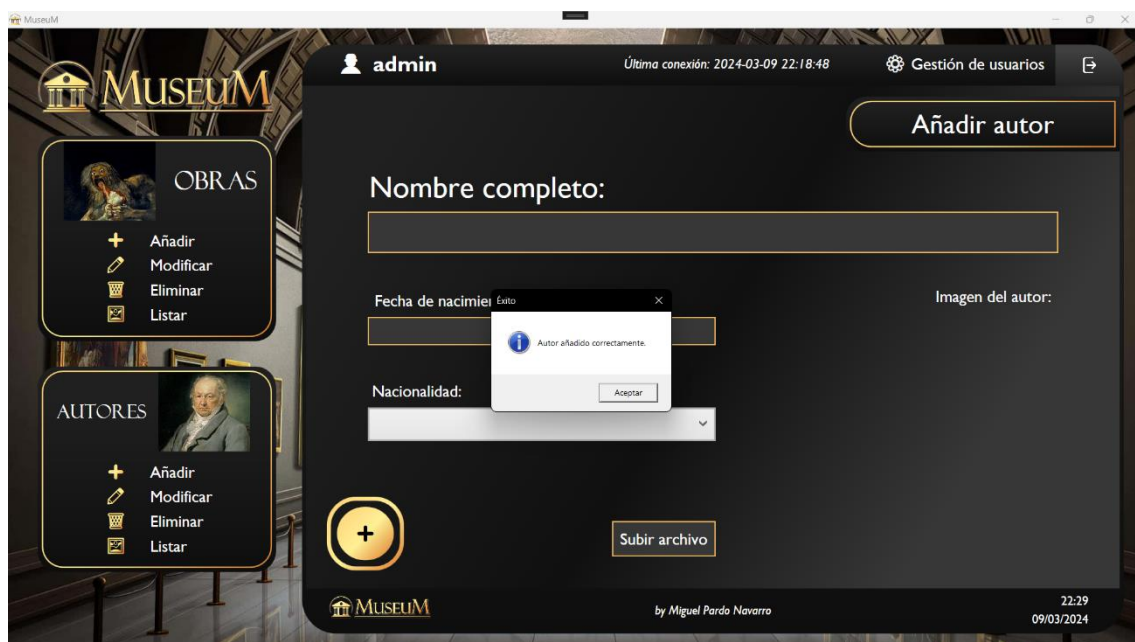
En cuanto a añadir autores, será el mismo procedimiento que para las obras: debemos indicar el nombre, escribir una fecha correcta (en este caso, el formato debe ser año-mes-día) y la imagen tiene que tener un tamaño inferior a 1 MB.





Una vez estén todos los campos obligatorios rellenados con éxito, se podrá añadir sin problema al autor.





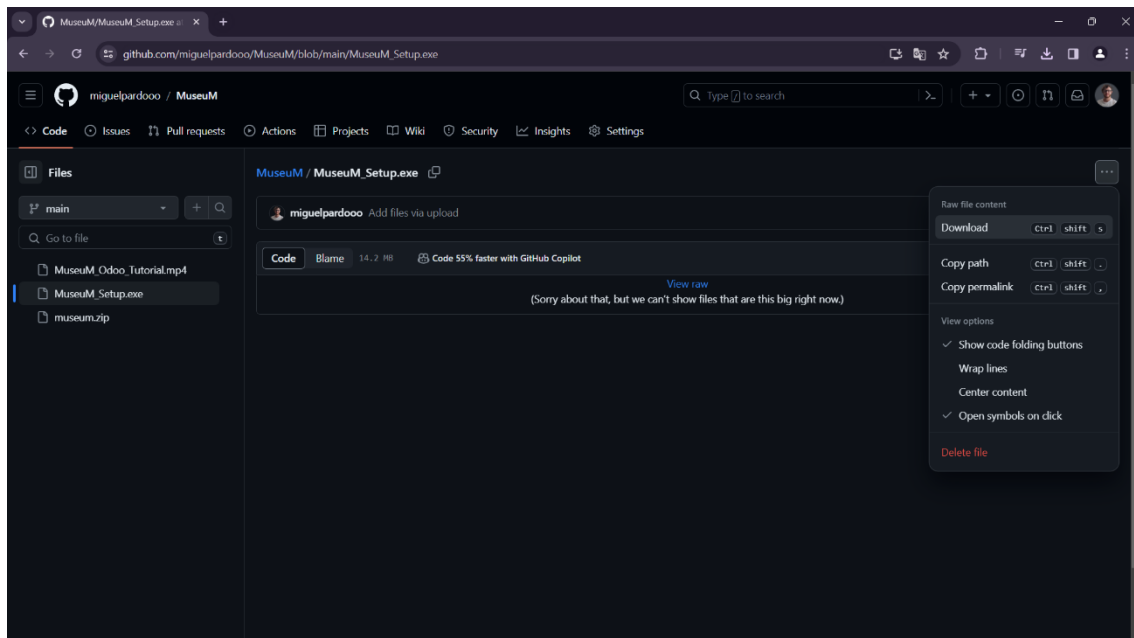
A la hora de modificar los registros existentes se tomarán las mismas restricciones que hemos usado para añadirlos.

## IMPLANTACIÓN

El proceso de implantación de la aplicación es muy rápido y sencillo. Siguiendo los pasos que a continuación explicaremos lograremos utilizar la aplicación y llevar a cabo la gestión de cualquier museo.

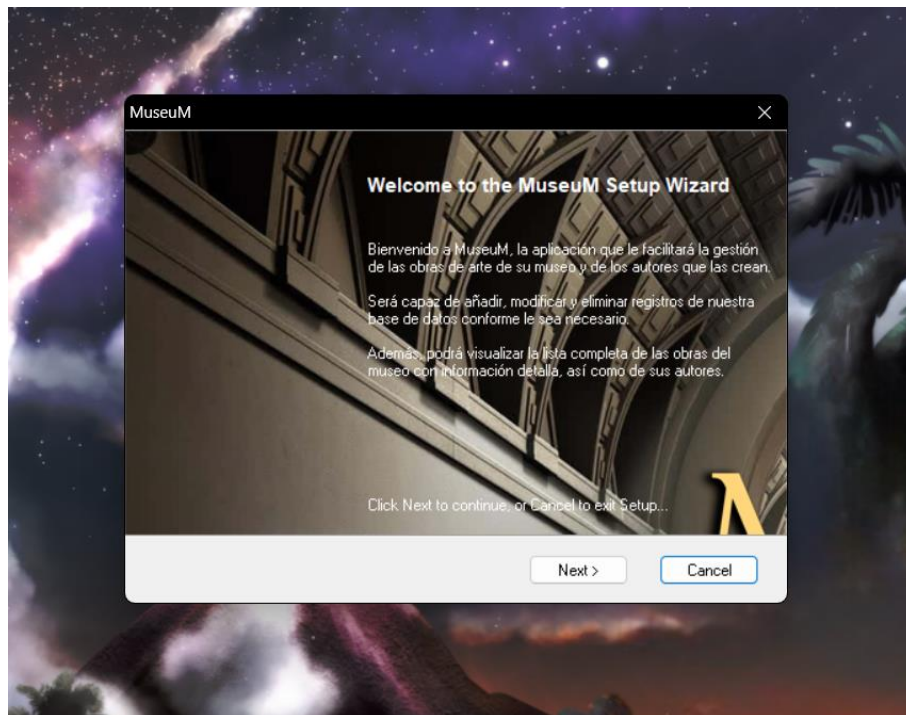
El instalador con el que conseguiremos llevar la aplicación a cualquier ordenador fue creado mediante InstallSimple<sup>4</sup>, programa que facilita la creación de archivos instaladores.

Lo primero que necesitaremos es el archivo ejecutable que nos permita instalar el programa en nuestro ordenador. Para ello, nos dirigiremos a la cuenta de GitHub del creador, @miguelpardooo, donde en el repositorio específico del proyecto *Museum* encontraremos dicho archivo<sup>5</sup>. Hacemos click sobre el y en el botón de los tres puntos de arriba a la derecha, donde aparecerá la opción “Download”.

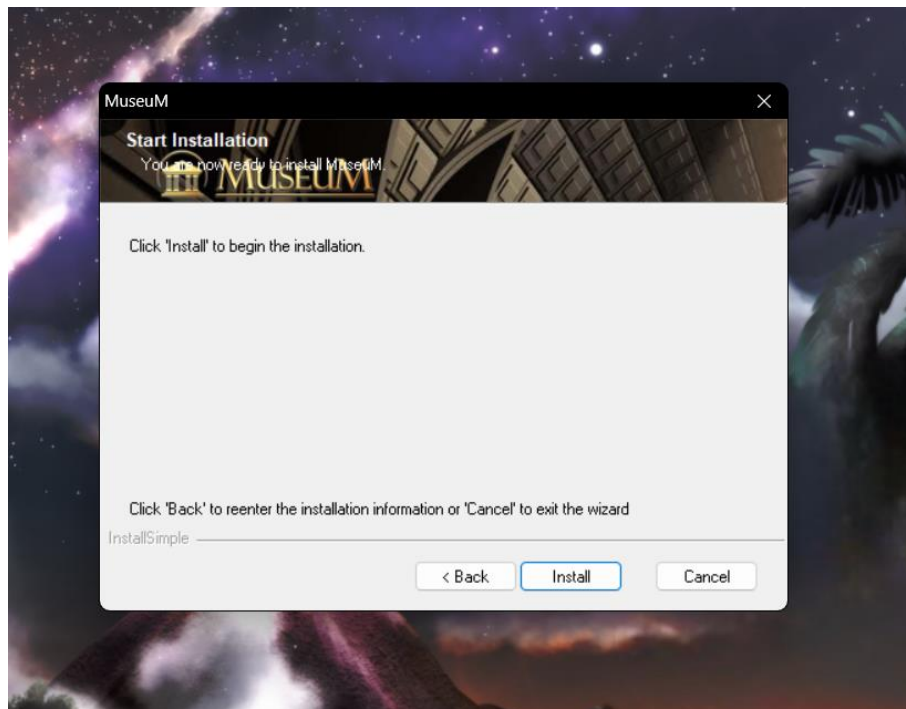


Una vez lo tengamos descargado, debemos ejecutar el archivo “Museum\_Setup.exe”. Es posible que nuestro antivirus advierta de algún riesgo o amenaza, pero esto es nada más que precaución, ejecuta el archivo de todos modos sin ningún miedo.

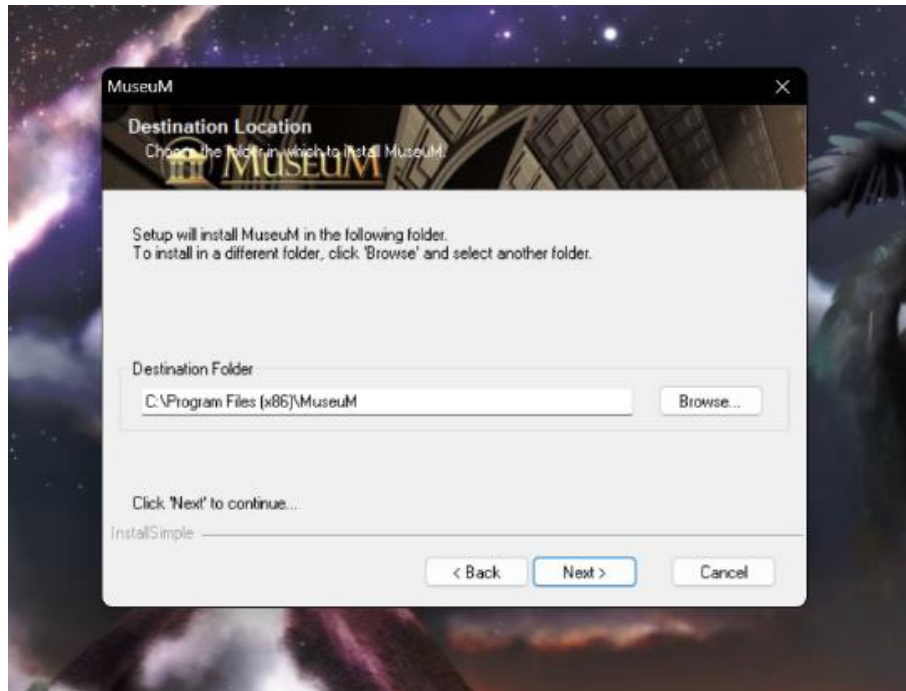
Seguidamente, se abrirá el instalador. Primero veremos la pantalla de inicio de la instalación que nos dará algo de información sobre la aplicación.



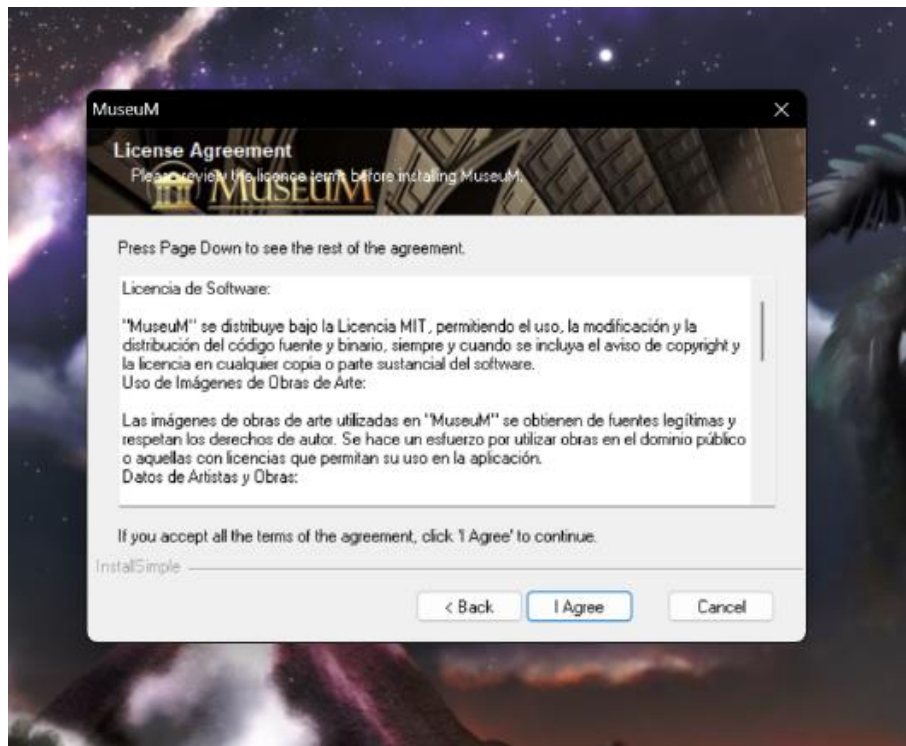
Al darle a siguiente, veremos la pantalla de iniciar la instalación, y entonces le daremos a "Install".



A continuación, nos dejará escoger la ruta donde se instalará el programa. La ruta por defecto será C:\Program Files (x86)\MuseumM.

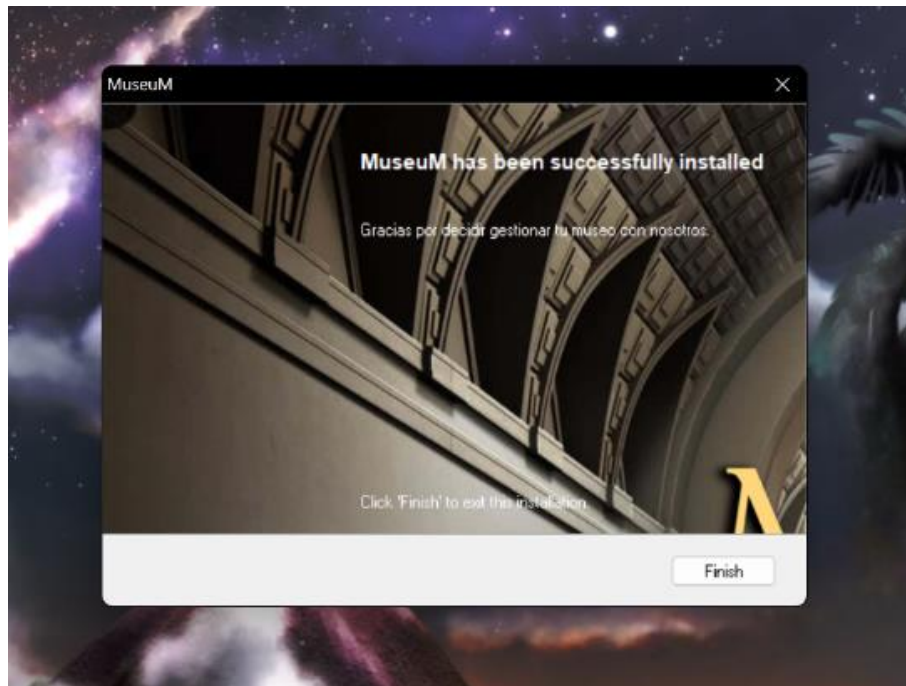


Finalmente, tendremos que aceptar los términos y la licencia de instalación. Conseguiremos esto con simplemente pulsar sobre "I Agree".

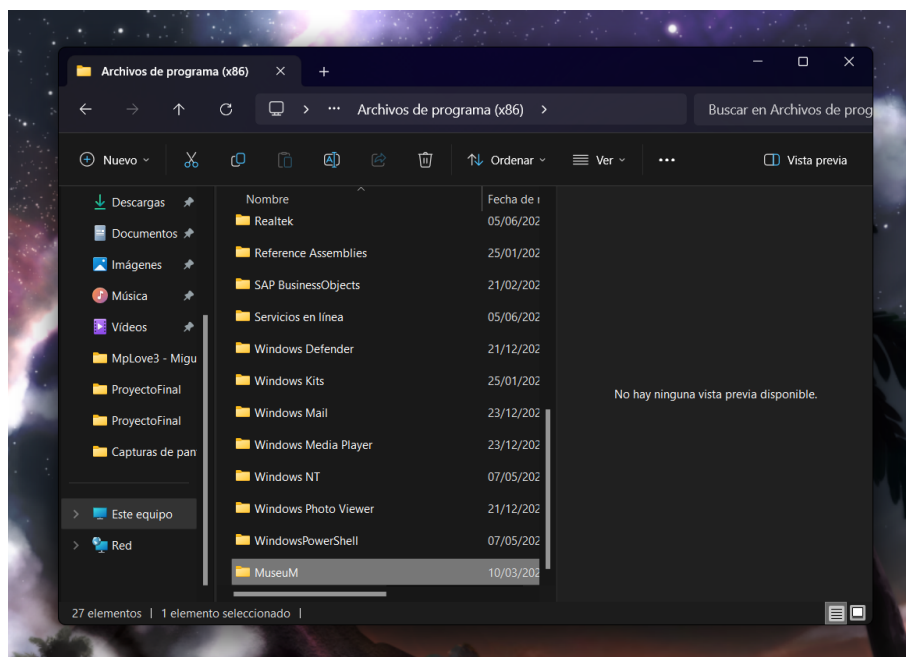




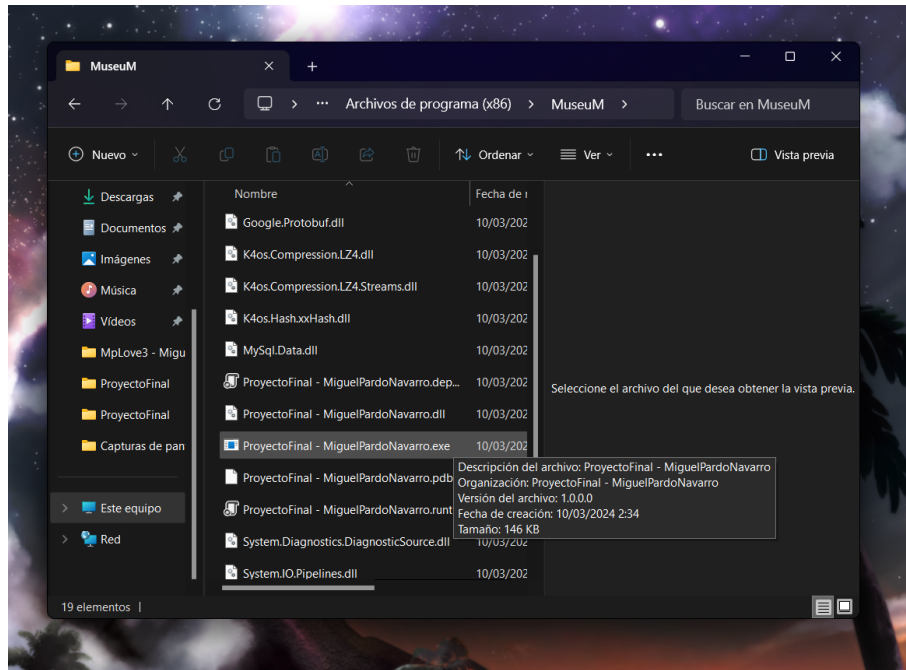
Y con esto ya habremos logrado instalar la aplicación con éxito en nuestro ordenador.



La cuestión ahora es, ¿cómo lo abrimos? La instalación no crea un acceso directo ni en el escritorio ni en Inicio de Windows. Pero que no cunda el pánico, pues solo tendremos que dirigirnos a la ruta que escogimos en la instalación y ahí estará la carpeta perteneciente a nuestra aplicación. Suele ser costumbre no cambiar la ruta de instalación por defecto, por lo tanto iremos a Archivos de programa (x86) en el disco C para encontrarla.



Dentro de la carpeta “MuseuM”, abriremos el archivo ejecutable “ProyectoFinal - MiguelPardoNavarro”, que es el que contiene la aplicación.



Y tras hacer esto se abrirá finalmente la aplicación, y estará todo listo para comenzar la gestión del museo.



Sin embargo, para que la aplicación funcione también será necesario importar la base de datos museum.sql (también descargable en GitHub) en nuestro servidor de base de datos MariaDB. Para ello tendremos que crear la base de datos “museum” vacía, y posteriormente importar el archivo mencionado.



Tras esto, si introducimos el usuario maestro **admin**, con contraseña **1234**, seremos capaces de empezar a usar la aplicación con todas sus funciones. Excepto la de lanzar informes, que va aparte en una aplicación auxiliar.

## CONCLUSIONES

A modo de conclusión, se debe mencionar que el proyecto cumple con creces las expectativas marcadas en el inicio. La aplicación no solo es capaz de realizar la gestión de las obras y los autores de un museo de arte, sino que el resultado final supera lo esperado en cuanto a diseño y posibilidades.

Al comienzo, surgieron ciertas dudas sobre las capacidades que tendría el proyecto para manejar imágenes en la base de datos de manera óptima, pero finalmente se logró completar todo lo esperado con muchos menos problemas de los previstos. La aplicación es capaz de trabajar con las imágenes de obras y autores sin ningún tipo de problema.

En cuanto a la estética, consideramos que la interfaz desarrollada está más que a la altura de la idea inicial de cómo queríamos que fuese. El uso de colores negros y grises adornados con detalles dorados le dan una apariencia elegante y limpia, que es lo que siempre se buscó para *Museum*.

Además, la interfaz es clara e intuitiva, y contiene ciertos elementos que también la dotan de dinamismo como el menú expandible o las galerías de imágenes. De esta manera, la aplicación ofrece al usuario una experiencia cómoda y agradable a la hora de realizar la labor de gestión.

Para finalizar, el proyecto es sin duda alguna un éxito. La aplicación es capaz de hacer todo lo que se esperaba de ella y su diseño no deja nada que desear a aplicaciones profesionales.

## PROPUESTAS DE MEJORA

En cuanto a propuestas de mejora, por mucho que el proyecto sea un éxito, siempre es posible mejorarlo. Sobre todo, en su presentación visual.

La interfaz y estructura principal de la aplicación está muy bien, pero es la misma sobre la que se lleva trabajando desde todo el proceso de desarrollo, haciendo que pueda resultar cargado de tantos elementos que se la han ido añadiendo con el paso del tiempo, y la dificultad de modificar los ya existentes por miedo a tener que cambiar todo. Por lo tanto, es posible limpiarla para obtener un resultado todavía mejor.

En el aspecto de la funcionalidad, el proyecto siempre se planteó para ser capaz de realizar las acciones que hace, siendo estas no muy complejas ni trabajosas, por lo que no sería complicado añadir nuevas funcionalidades que tampoco tengan mucha dificultad como sería ser capaz de trabajar con obras que fuesen más allá de cuadros y se explorasen otros ámbitos artísticos como

la escultura, también propio de un museo. O bien se podría complicar más la funcionalidad existente haciendo posible el trabajo con imágenes de forma aún más dinámica y profesional, como por ejemplo ser capaz de hacer zoom en las obras a nuestro antojo o acciones similares.

En definitiva, algo bueno del tema escogido para el proyecto es que siempre se es capaz de llegar a nuevas ideas que mejoren la presentación en un mundo tan artístico como el de los museos.

## ANEXO

---

<sup>1</sup> *A free database designer for developers and analysts.* (s. f.). <https://dbdiagram.io/d>

<sup>2</sup> *Balsamiq Cloud.* (s. f.). <https://balsamiq.cloud/>

<sup>3</sup> *MySQL.Data 8.3.0.* (s. f.). <https://www.nuget.org/packages/MySQL.Data/>

<sup>4</sup> *Download InstallSimple.* (s. f.). <http://installsimple.com/download.htm>

<sup>5</sup> *Miguelpardooo.* (s. f.-c). *Museum/Museum\_Setup.exe at main · miguelpardooo/Museum.* GitHub. [https://github.com/miguelpardooo/Museum/blob/main/Museum\\_Setup.exe](https://github.com/miguelpardooo/Museum/blob/main/Museum_Setup.exe)