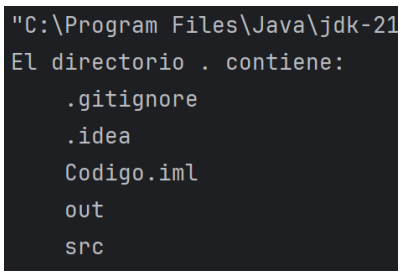
EJERCICIO 1

El programa Main.java permite explorar una ruta en el sistema de archivos. Dependiendo de si la ruta apunta a un archivo o a un directorio, el programa realiza distintas acciones. Si la ruta corresponde a un archivo, muestra su tamaño en bytes mediante length() y los permisos de ejecución, lectura y escritura utilizando canExecute(), canRead() y canWrite().

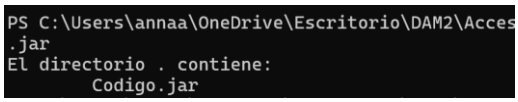
Por otro lado, si la ruta corresponde a un directorio, se listan los archivos que contiene. En caso de que la ruta no exista, el programa informa que el fichero o directorio no está disponible.

Para poder ejecutar la aplicación fuera de IntelliJ IDEA, se generó un artefacto ejecutable. Un artefacto es un paquete que contiene las clases compiladas (.class), los recursos necesarios como imágenes o librerías, y una clase con main() que permite ejecutar la aplicación. En este caso, se creó un JAR ejecutable denominado Codigo.jar.

Durante la ejecución dentro de IntelliJ IDEA, el programa mostró el contenido del directorio del proyecto:



Al ejecutar el JAR desde la terminal con CMD, el directorio por defecto apuntaba a la carpeta donde se encontraba el artefacto, mostrando únicamente el archivo Codigo.jar, que contiene el programa empaquetado. Esto confirma que el JAR es ejecutable de forma independiente y que incluye todos los elementos necesarios para su funcionamiento.



Como conclusión, la práctica ha permitido comprender cómo desarrollar, empaquetar y ejecutar una aplicación Java tanto dentro del IDE IntelliJ IDEA como de manera externa mediante un JAR ejecutable. Se ha comprobado la importancia de mantener la compatibilidad entre la versión del JDK utilizada para compilar y la del runtime que ejecuta el programa, así como la necesidad de generar artefactos correctamente configurados para incluir todas las dependencias. Además, se ha facilitado la comprensión de la estructura de directorios y ficheros de un proyecto Java, y cómo el programa puede interactuar con el sistema de archivos, mostrando información sobre archivos y directorios de manera consistente en ambos entornos de ejecución.