

Manual de modificación e instalación lotería de ciudadanía


Instalación de visual studio code

Entrar al link y descargar la opción de su sistema operativo

<https://code.visualstudio.com/download>

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



↓ **Windows**

Windows 7, 8, 10

User Installer

64 bit

32 bit

ARM

System Installer

64 bit

32 bit


ARM

.zip

64 bit

32 bit

ARM



↓ **.deb**

Debian, Ubuntu

↓ **.rpm**

Red Hat, Fedora, SUSE

.deb

64 bit

ARM

ARM 64

.rpm

64 bit

ARM

ARM 64


.tar.gz

64 bit

ARM

ARM 64

Snap Store



↓ **Mac**

macOS 10.10+

.zip

Universal

Intel Chip

Apple Silicon

Se ejecuta el instalable exe que se descargó y se le da siguiente con todas las opciones predeterminadas

Instalación de node.JS

Entrar al link y descargar la opción LTS

<https://nodejs.org/en/>



[HOME](#) | [ABOUT](#) | [DOWNLOADS](#) | [DOCS](#) | [GET INVOLVED](#) | [SECURITY](#) | [CERTIFICATION](#) | [NEWS](#)

Node.js® is a JavaScript runtime built on [Chrome's V8 JavaScript engine](#).

Download for Windows (x64)

14.17.3 LTS

Recommended For Most Users

16.5.0 Current

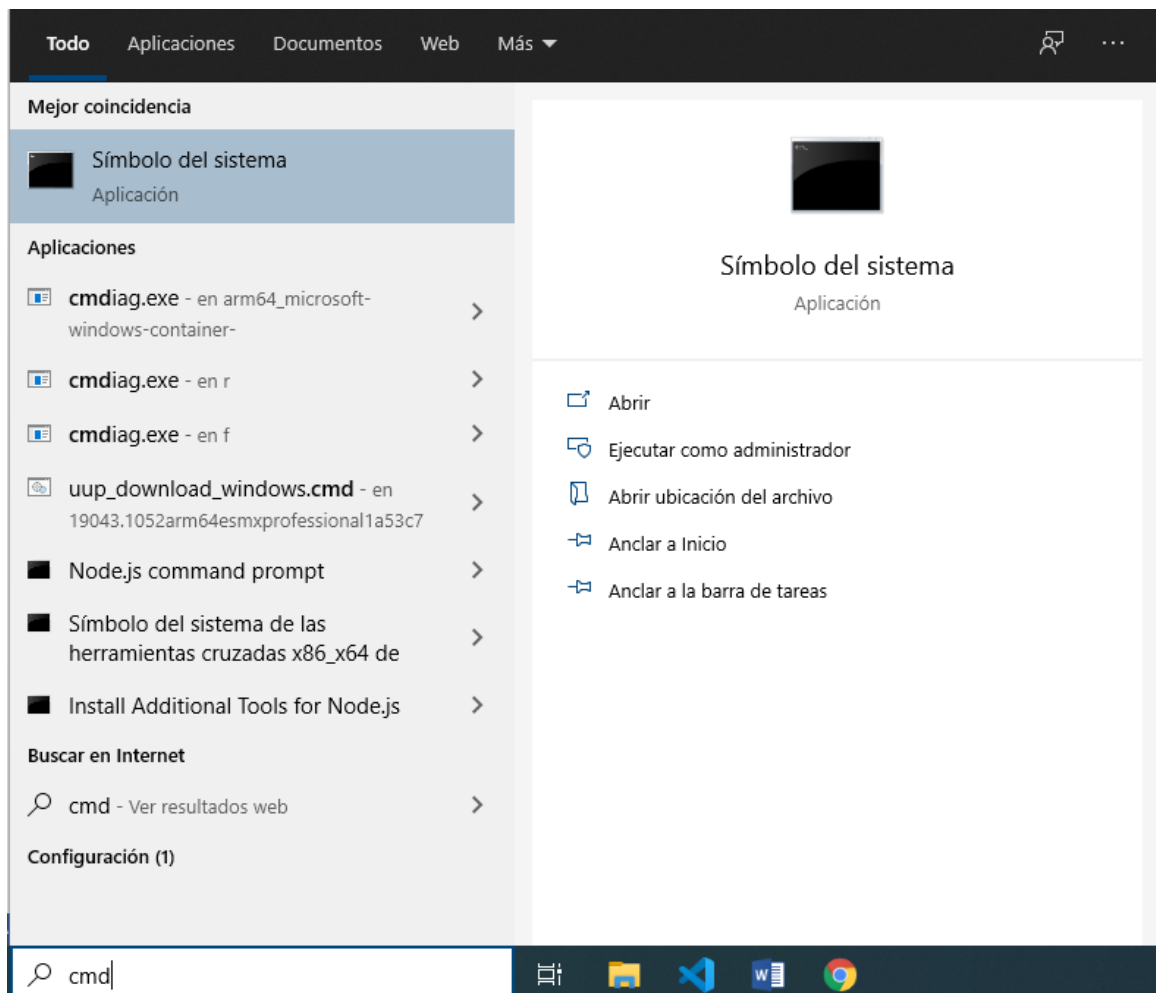
Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#) [Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [Long Term Support \(LTS\) schedule](#).

Y de la misma manera que con Visual Studio Code lo instalamos con todas las opciones predeterminadas.

Para confirmar que se instaló de la manera correcta abrimos la línea de comandos de Windows buscando CMD.



Cuando esto se encuentre abierto corremos el siguiente comando.

```
npm -v
```

Una vez terminado de ejecutarse el comando anterior, ejecutamos el siguiente para tener la última versión.

```
npm install npm -g
```

Nos debe de mostrar la versión que se instaló.

```
C:\Users\Ingenieria y Sistema>npm -v
7.17.0
C:\Users\Ingenieria y Sistema>
```

Instalación de Ionic

Para poder construir el programa necesitamos de Ionic que es un framework sobre el cual se construye la aplicación, para instalarlo debemos de correr el siguiente comando en la línea de comandos.

```
npm install -g cordova ionic
```

Instalación de dependencias

Las siguientes dependencias deben de ser instaladas a través de la línea de comandos o símbolo del sistema.

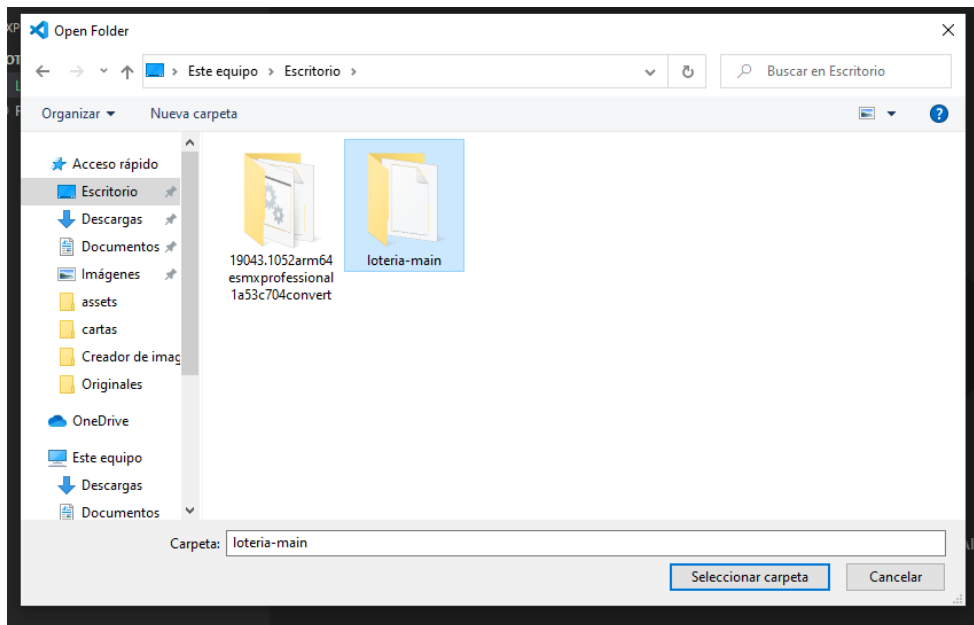
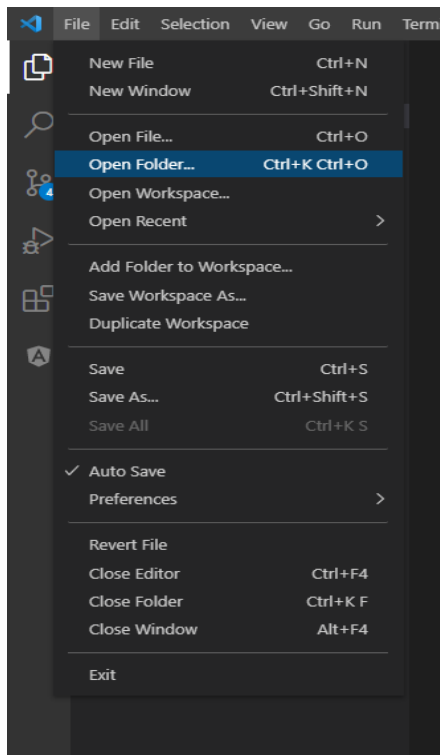
```
npm install electron -g --unsafe-perm=true --allow-root
```

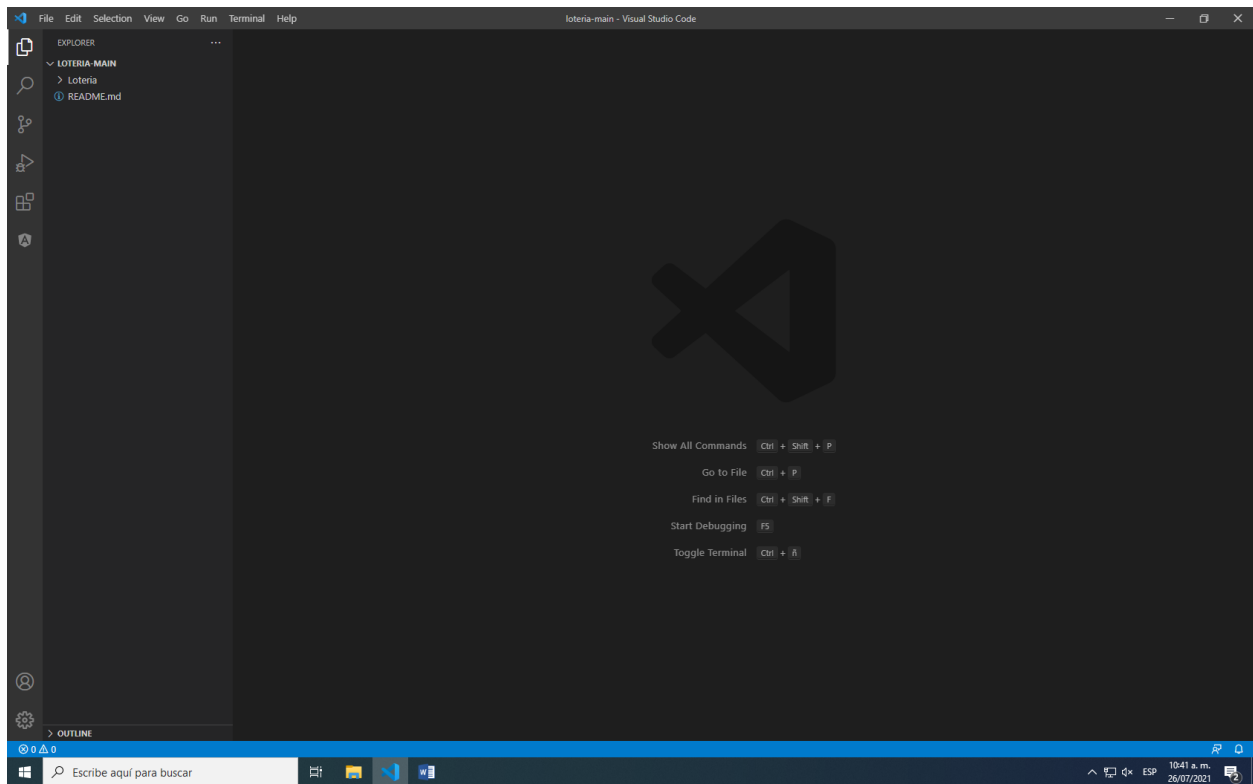
```
npm install electron-packager -g --unsafe-perm=true --allow-root
```

```
npm install electron-installer-dmg -g --unsafe-perm=true --allow-root
```

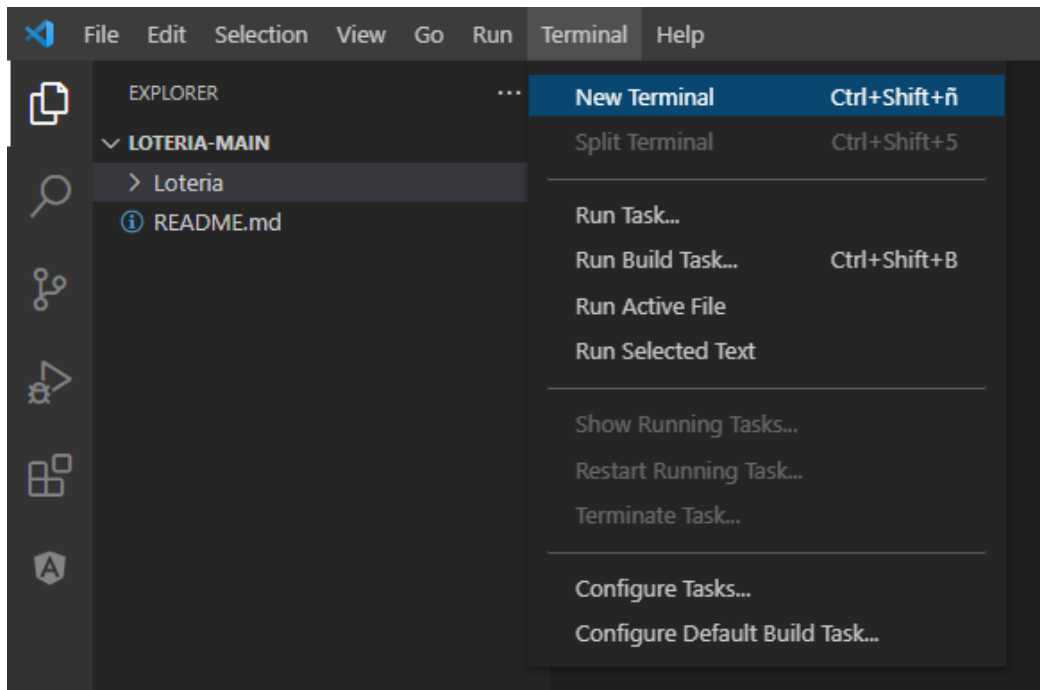
Preparación del ambiente de desarrollo.

Con el archivo .zip que se les fue otorgado descomprimirlo, después de esto abrir el folder en Visual Studio Code.



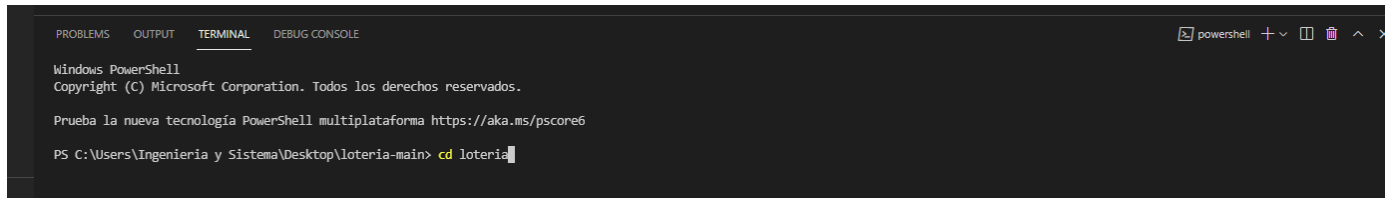


Cuando tengamos el folder abierto abrimos una nueva terminal dentro de Visual Studio Code.



Con la terminal abierta escribimos el siguiente comando para situarnos en la carpeta que debemos hacer el resto de instalaciones del sistema.

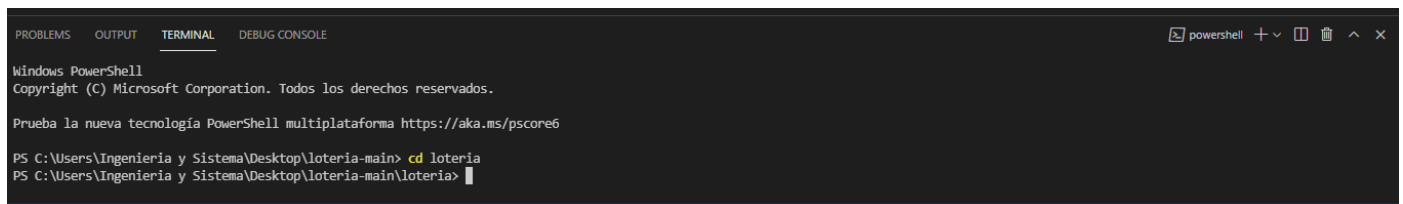
```
cd Loteria
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Ingenieria y Sistema\Desktop\loteria-main> cd loteria
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

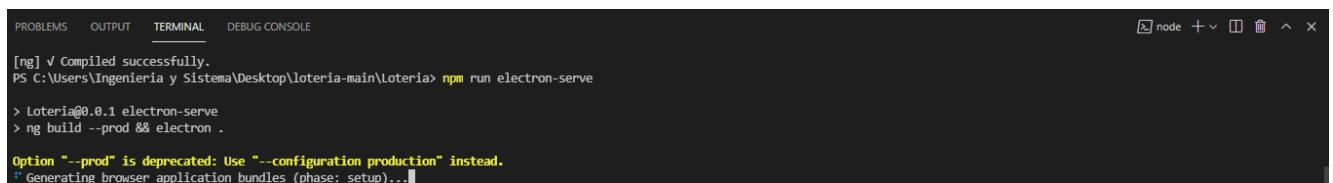
PS C:\Users\Ingenieria y Sistema\Desktop\loteria-main> cd loteria
PS C:\Users\Ingenieria y Sistema\Desktop\loteria-main\loteria>
```

Una vez que nos encontremos situados en esa locación del sistema procederemos a correr los siguientes comandos.

```
npm i
```

Una vez que termine de ejecutarse el comando anterior y para probar que se instalo todo de manera correcta corremos el siguiente comando.

```
npm run electron-serve
```



```
[ng] ✓ Compiled successfully.
PS C:\Users\Ingenieria y Sistema\Desktop\loteria-main\loteria> npm run electron-serve

> loteria@0.0.1 electron-serve
> ng build --prod && electron .

Option "--prod" is deprecated: Use "--configuration production" instead.
Generating browser application bundles (phase: setup)...
```

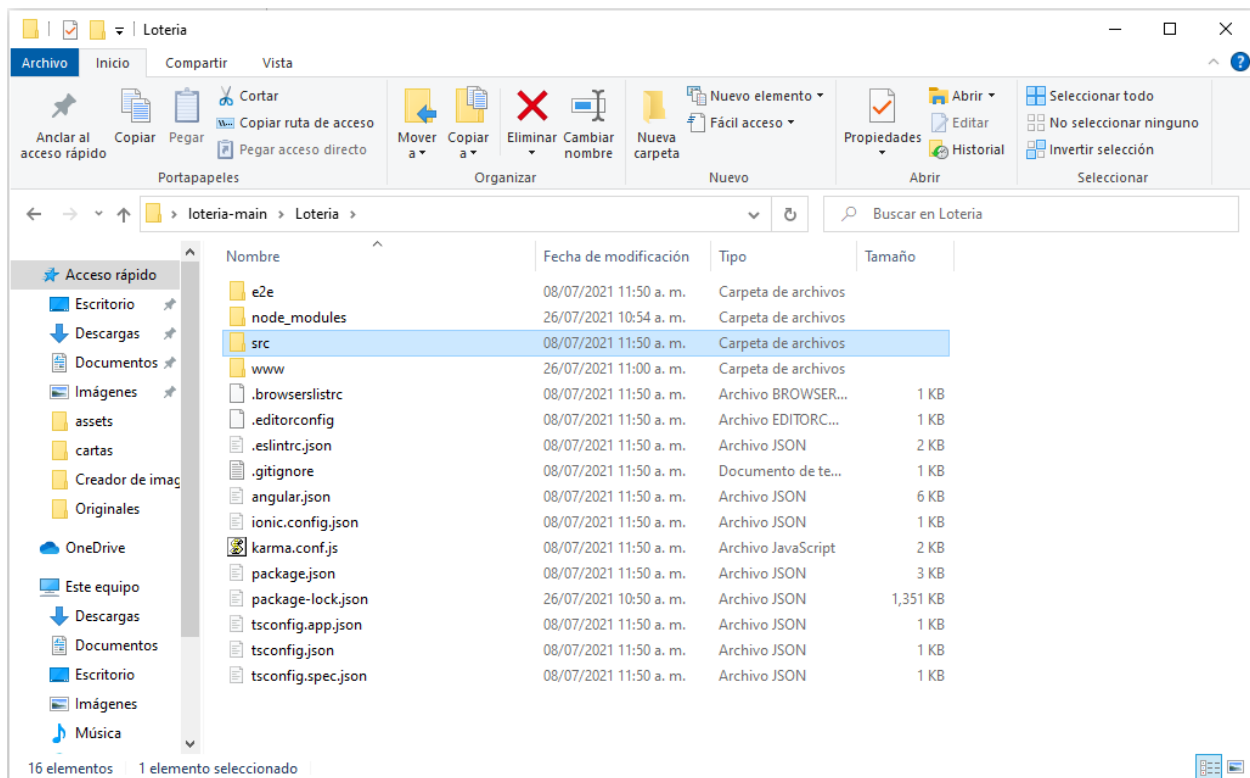
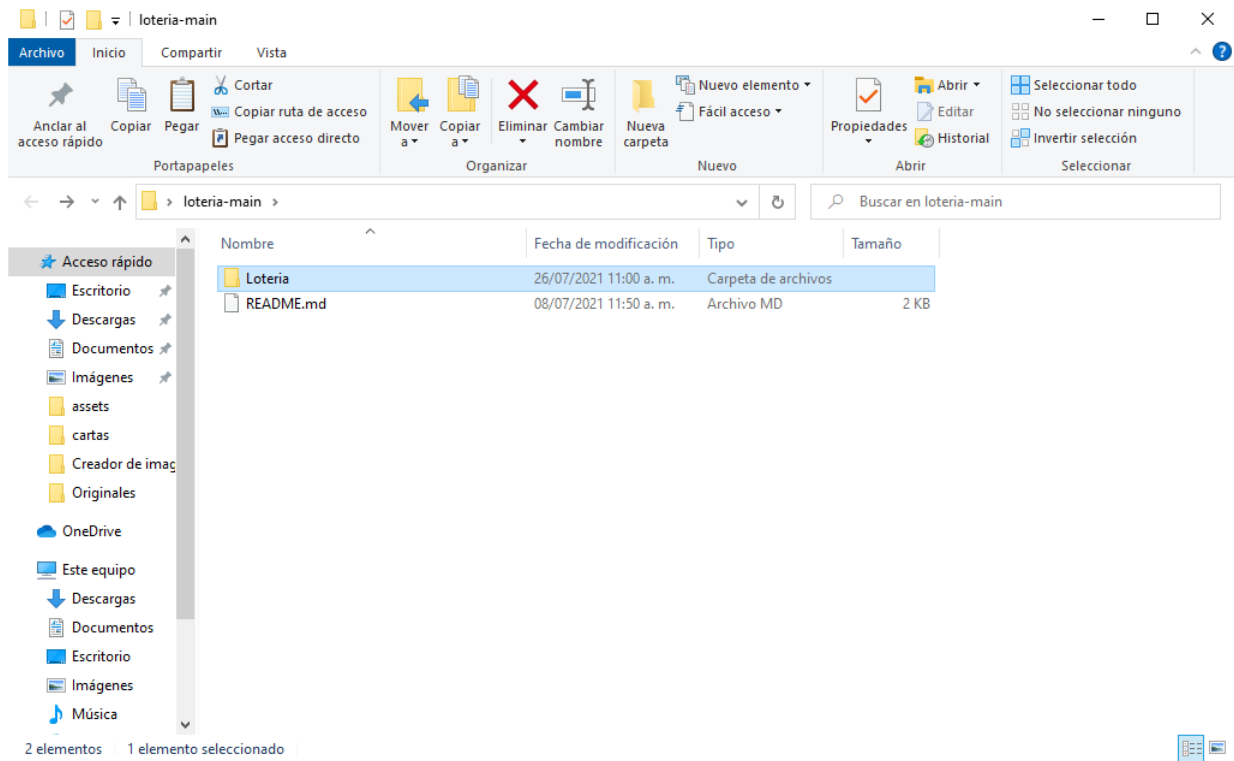
Al finalizar este comando se abrirá una emulación del sistema y esto será una indicación de que todo fue instalado de manera satisfactoria.

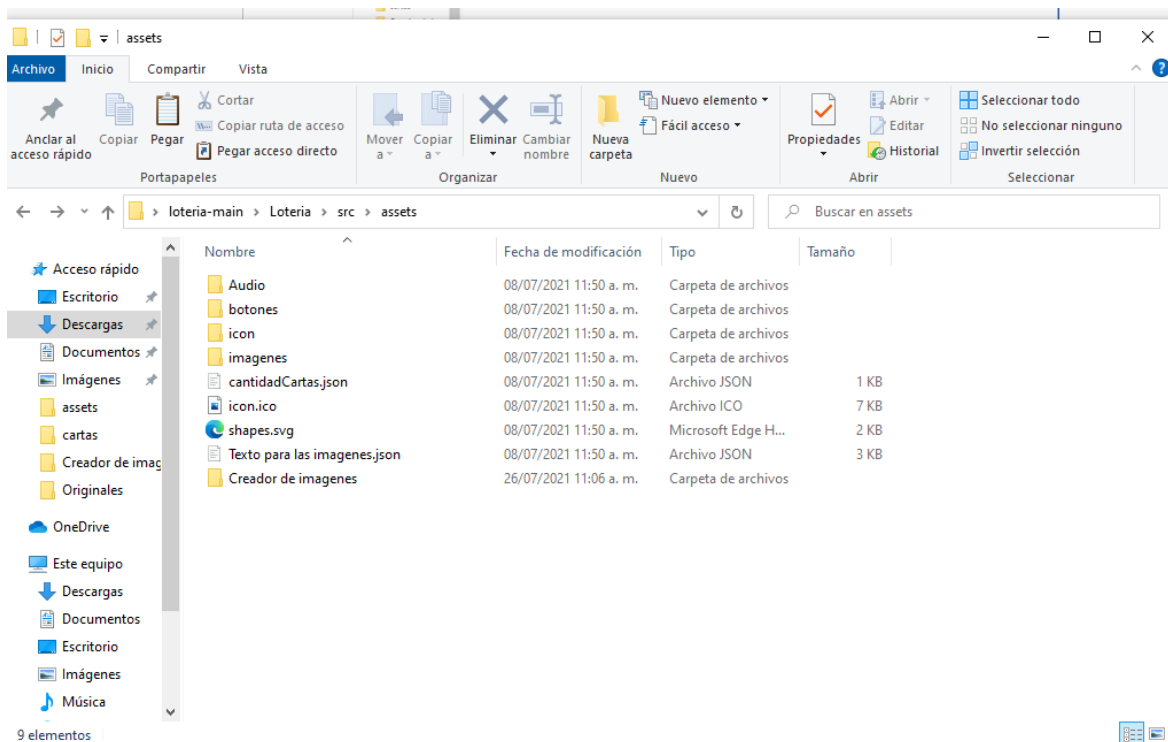


Para cancelar la emulación de la vista previa haga click en la terminal y después presione ctrl+c

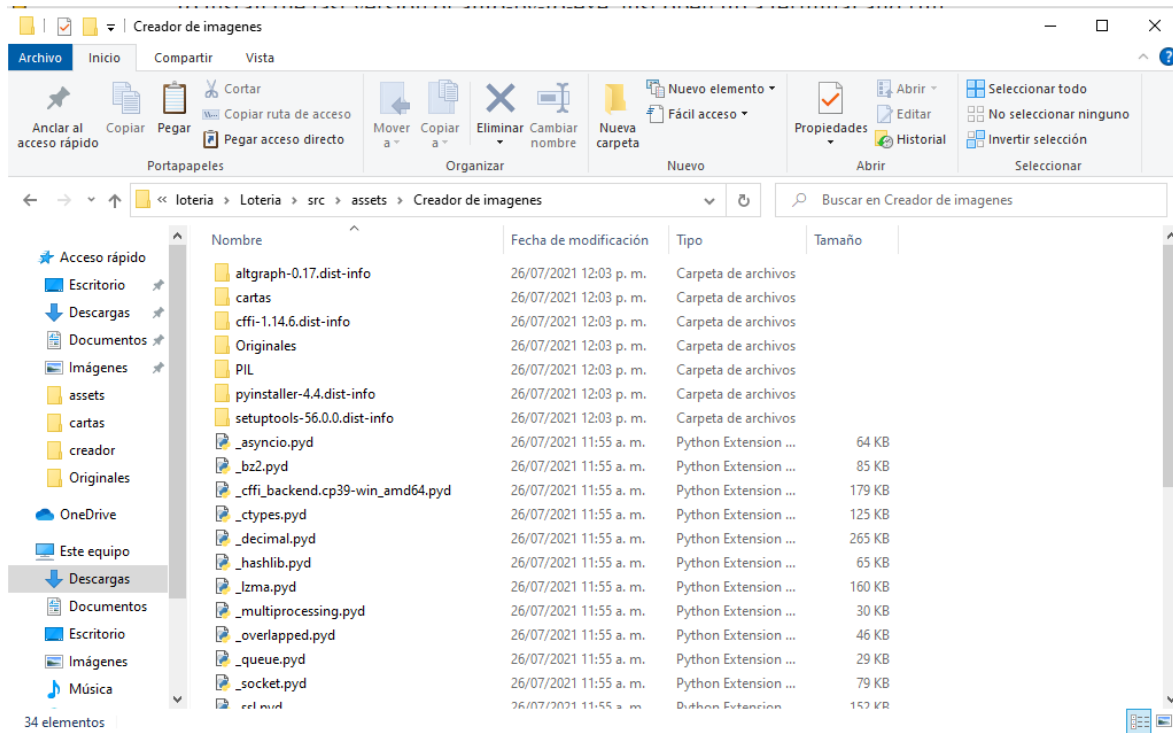
[Modificación de cartas y audios de la lotería](#)

Abrimos la carpeta en donde descomprimos el archivo zip y seguimos el siguiente directorio loteria-main>Loteria>src>assets



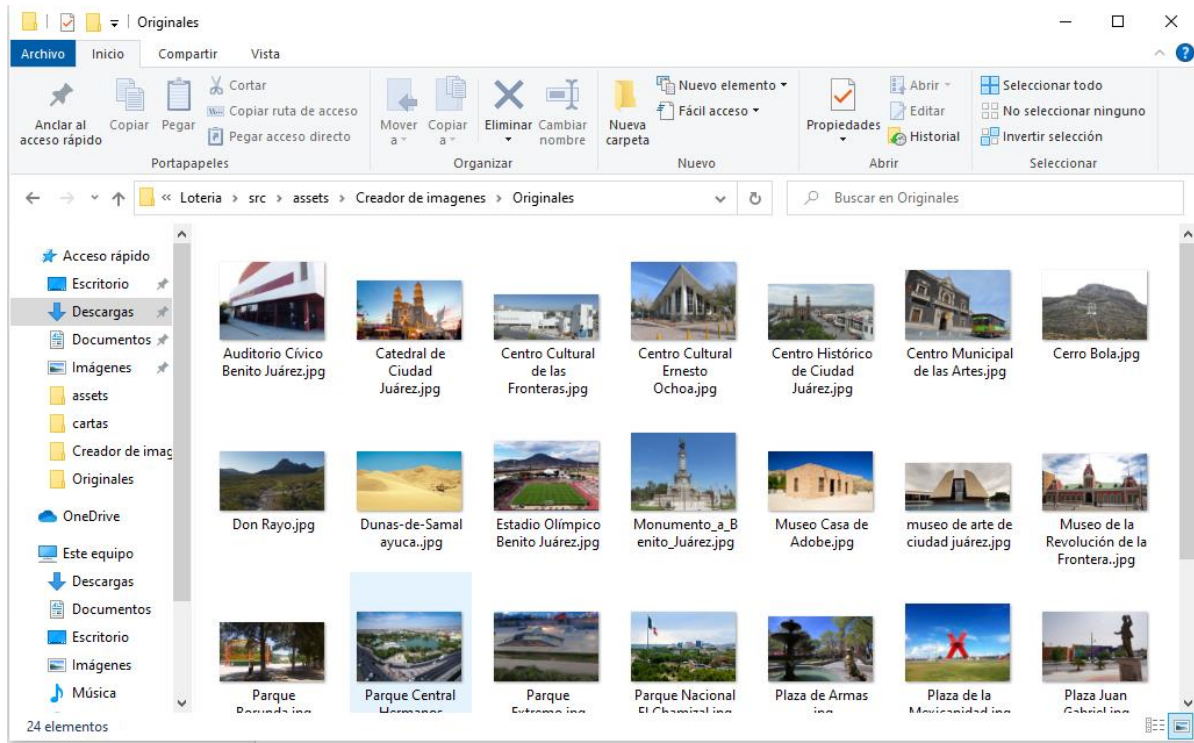


Generación de cartas y texto al lanzar las cartas
Abrimos el creador de imágenes.



Dentro de esta carpeta se encuentran dos carpetas, en la carpeta de Originales deberemos de agregar las imágenes que van a ser utilizadas en la lotería para darles el formato de tamaño adecuado, estas

imágenes deben de estar en formato .jpg y deben de estar nombradas con el texto que se dirá al momento de que la carta sea mostrada.

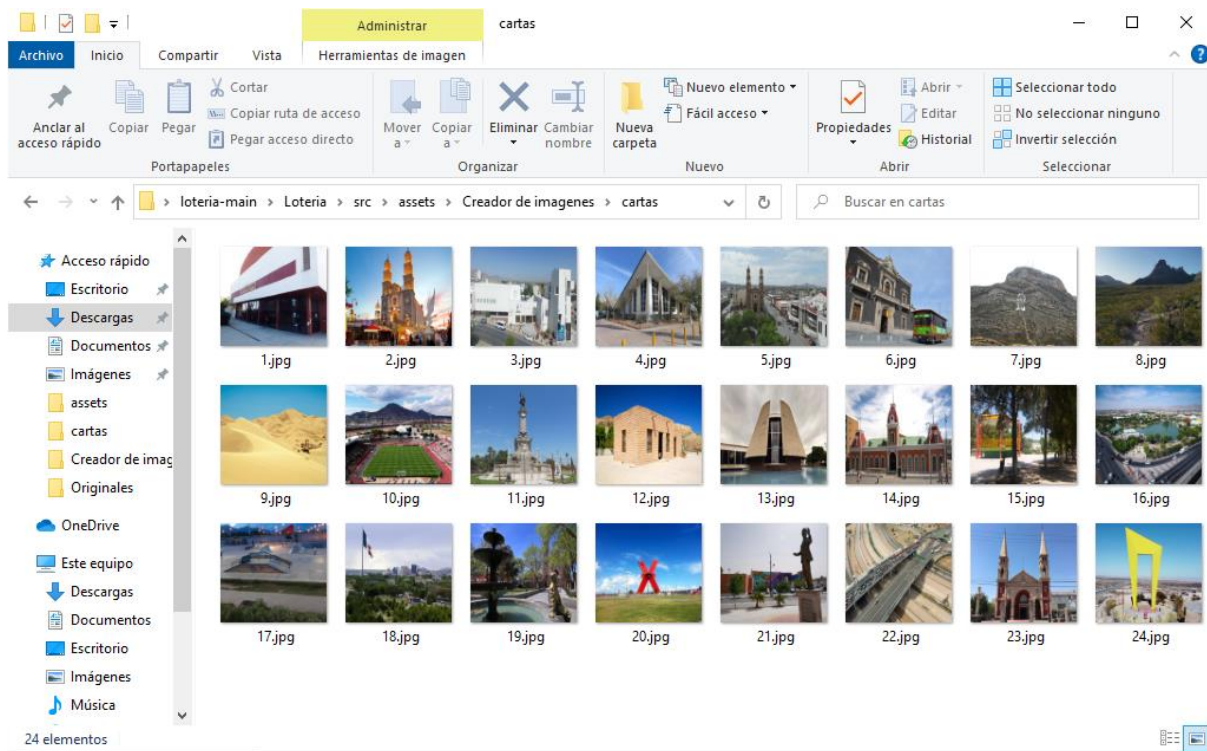


Una vez que las imágenes se encuentren en esta carpeta con las indicaciones anteriores deberemos de asegurarnos que la carpeta de cartas se encuentre vacía.

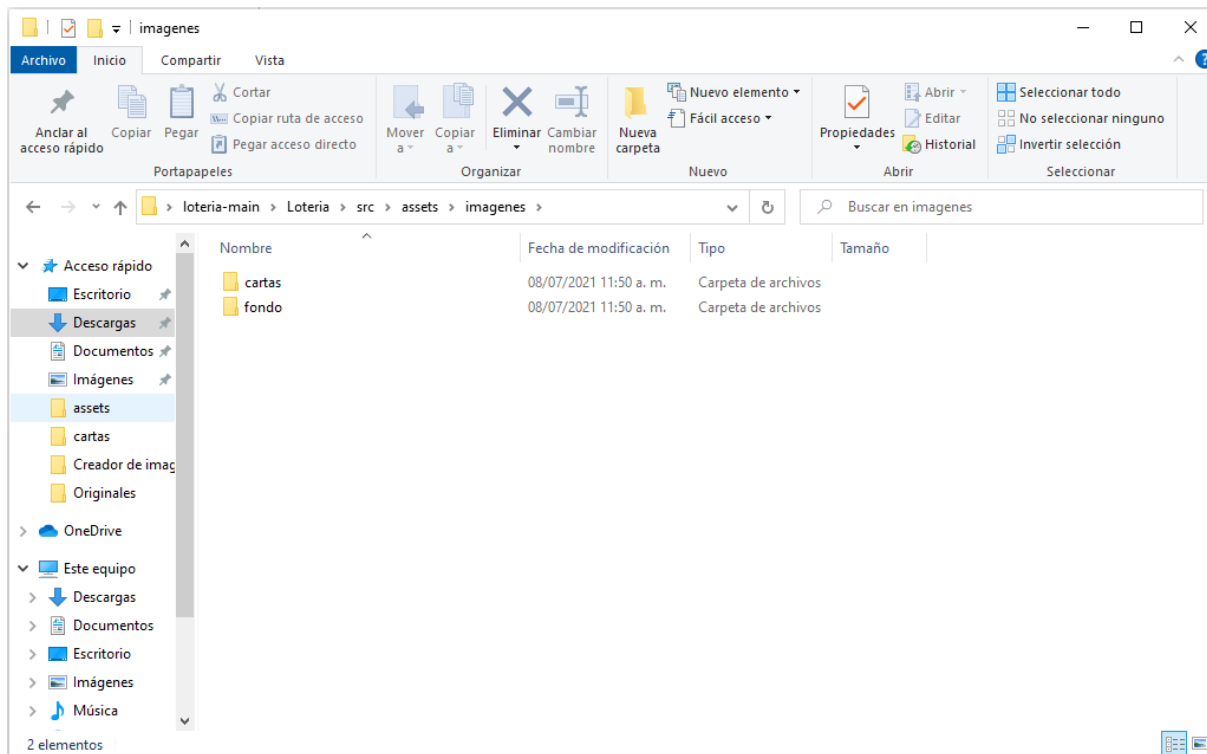
Ya que estemos seguros de que la carpeta esta vacia y que se encuentran en el formato y con el nombre correcto deberemos de ejecutar el archivo imageResize.exe el cual nos va a generar las imágenes con el tamaño y nombre para que la aplicación las reconozca.

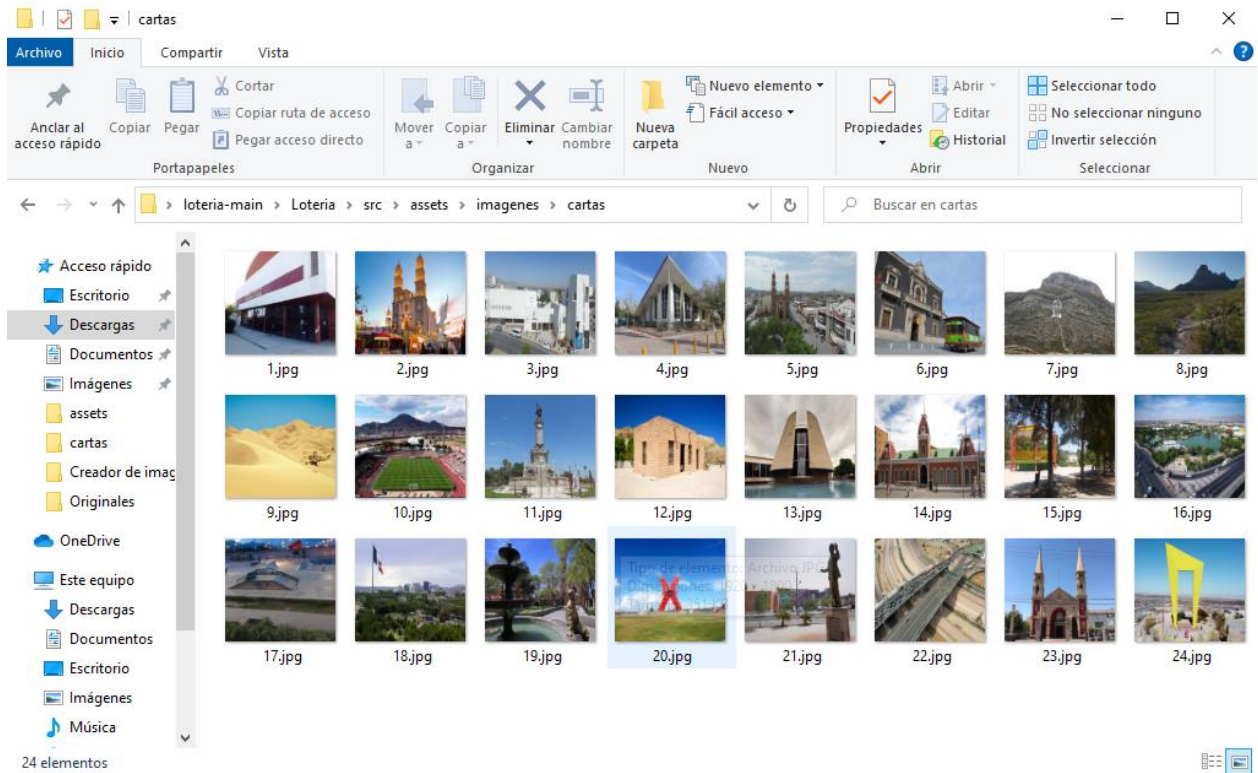


Cuando termine el programa de ejecutarse se cerrara de forma automática y dentro de la carpeta de cartas se crearan las imágenes con el formato necesario.

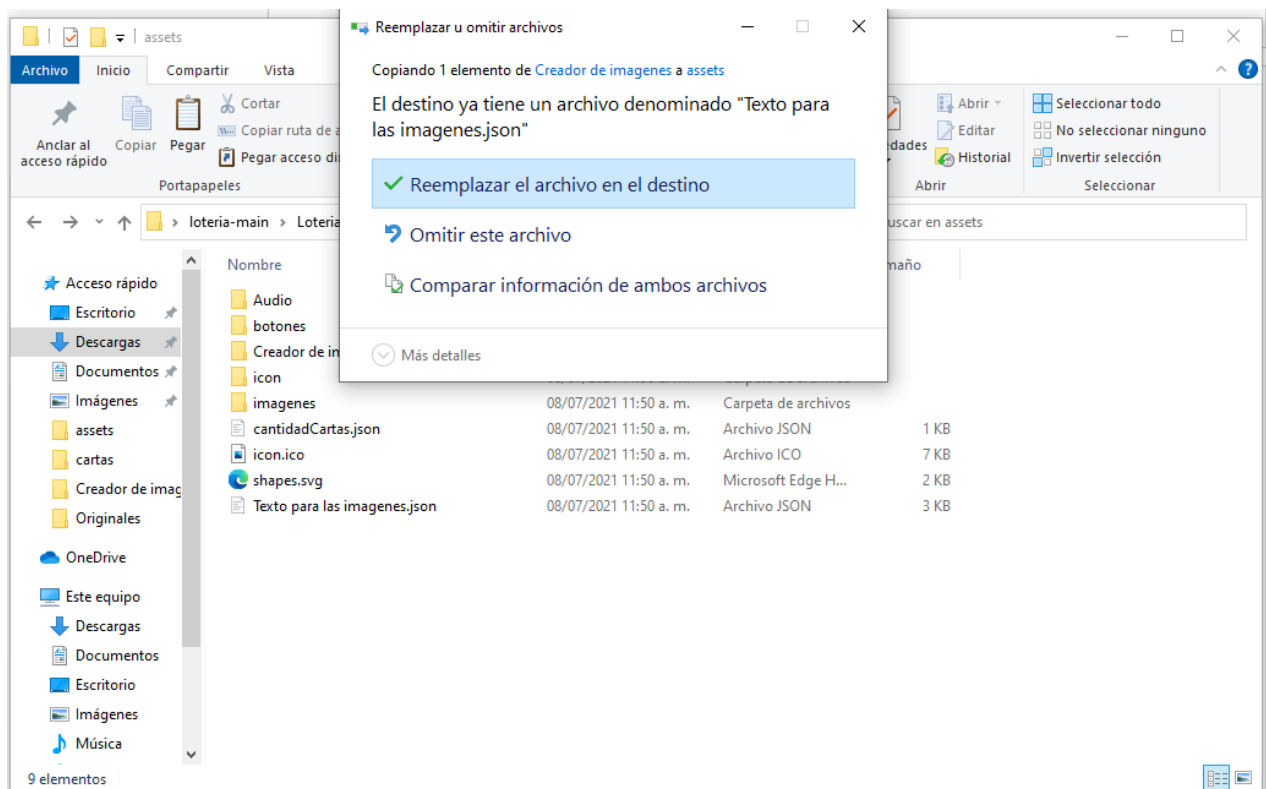


Estas imágenes las copiaremos en la carpeta de imágenes>cartas



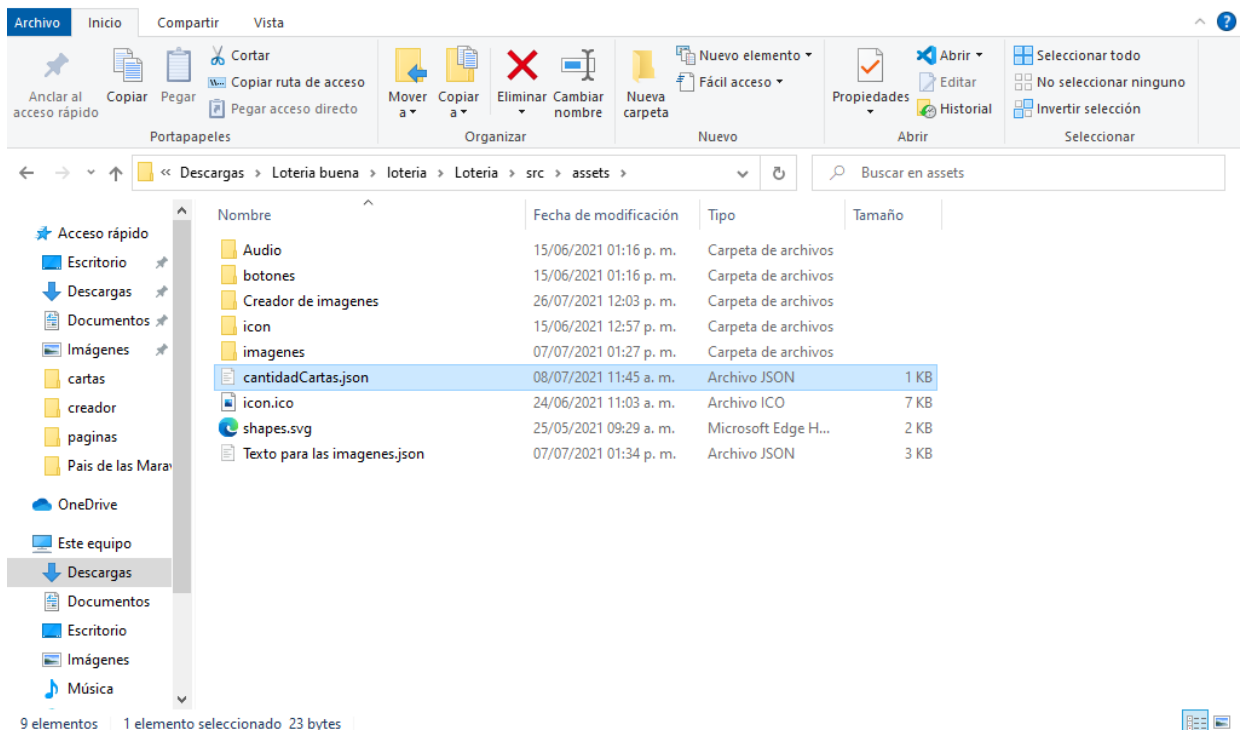


En el creador de imágenes también nos modifica el archivo “Texto para las imágenes.json” que es donde se encuentra el texto que se va a decir para cada imagen cuando sea lanzada. Este archivo lo copiaremos en la carpeta principal de assets y remplazamos los archivos que encuentra en assets.

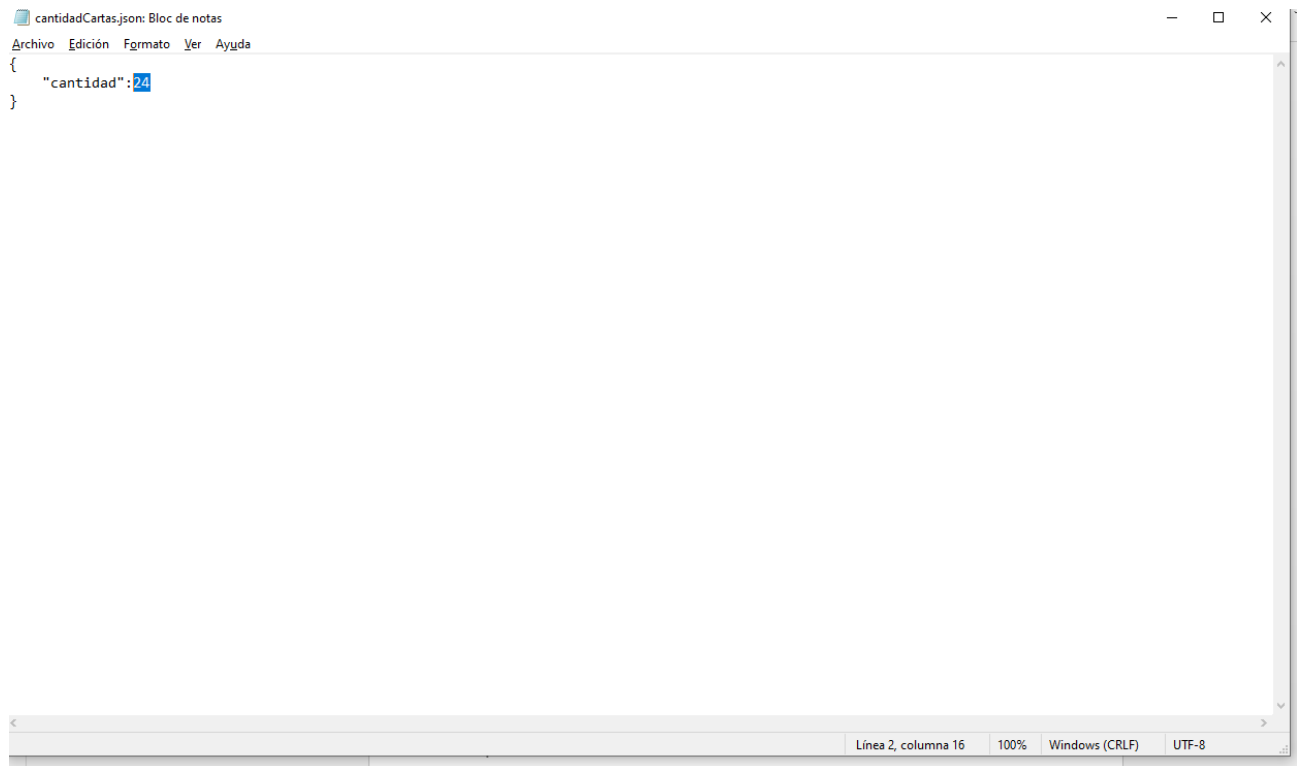


Modificación de cantidad de cartas

Dentro del directorio de asset se encuentra un archivo llamado “cantidadCartas”, dentro de este archivo se debe de modificar la cantidad de cartas que se están utilizando en la lotería de acorde a la cantidad de imágenes que se fueron otorgadas.



El archivo se abra en un bloc de notas y solo deberemos de cambiar el numero que aparece despues de los 2 puntos

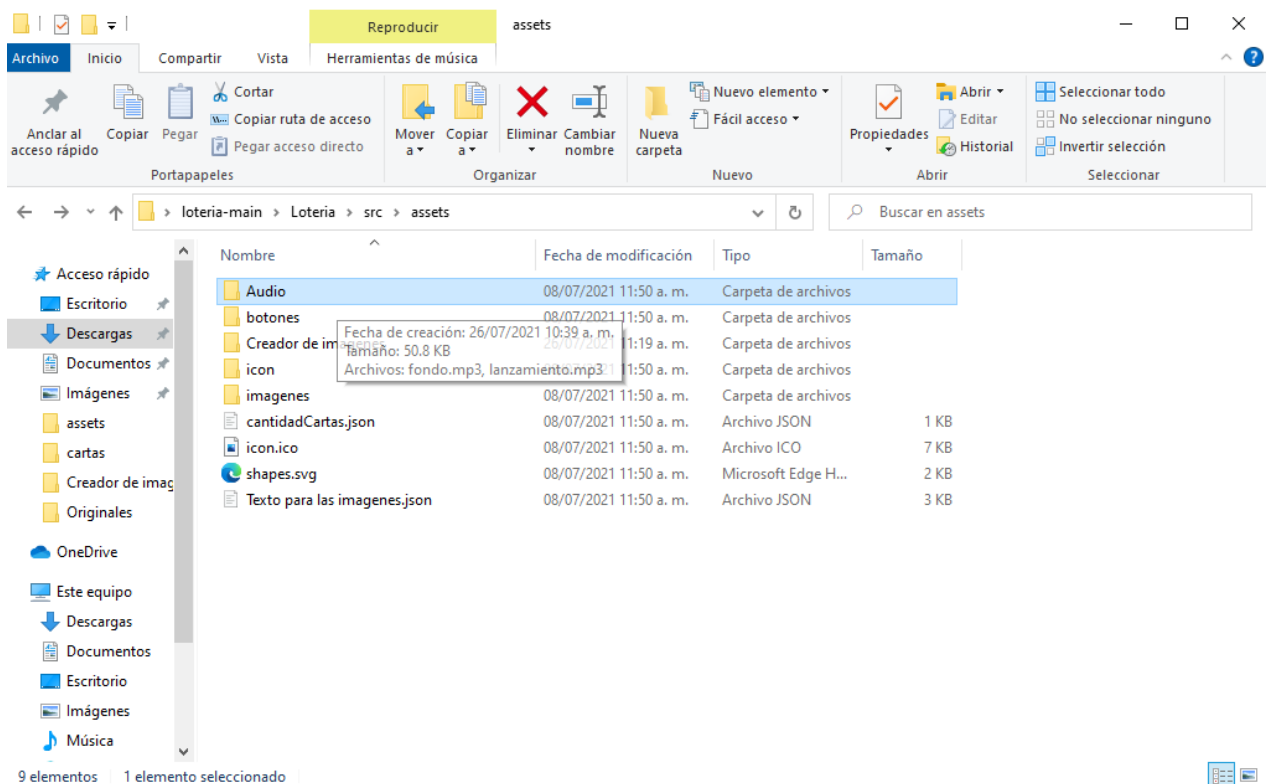


```
cantidadCartas.json: Bloc de notas
Archivo Edición Formato Ver Ayuda
{
  "cantidad": 24
}
```

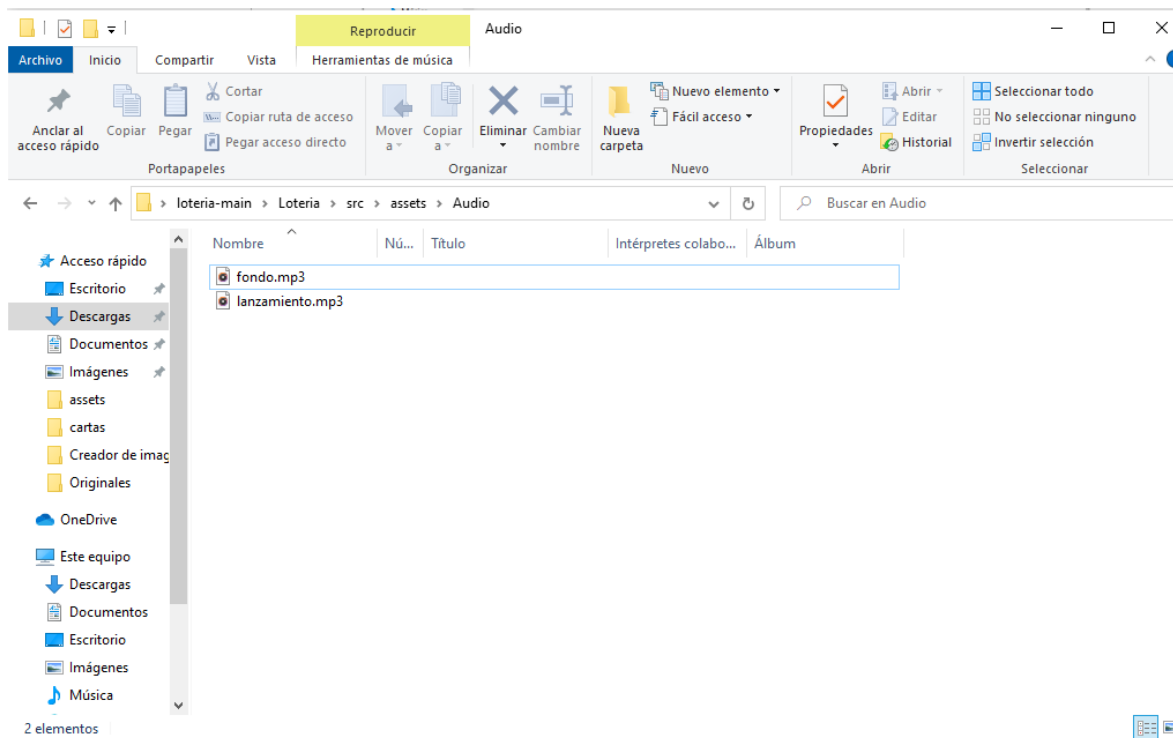
Línea 2, columna 16 100% Windows (CRLF) UTF-8

Modificación de audio de fondo y lanzamiento de carta.

Dentro del directorio de assets encontramos una carpeta llamada Audio.



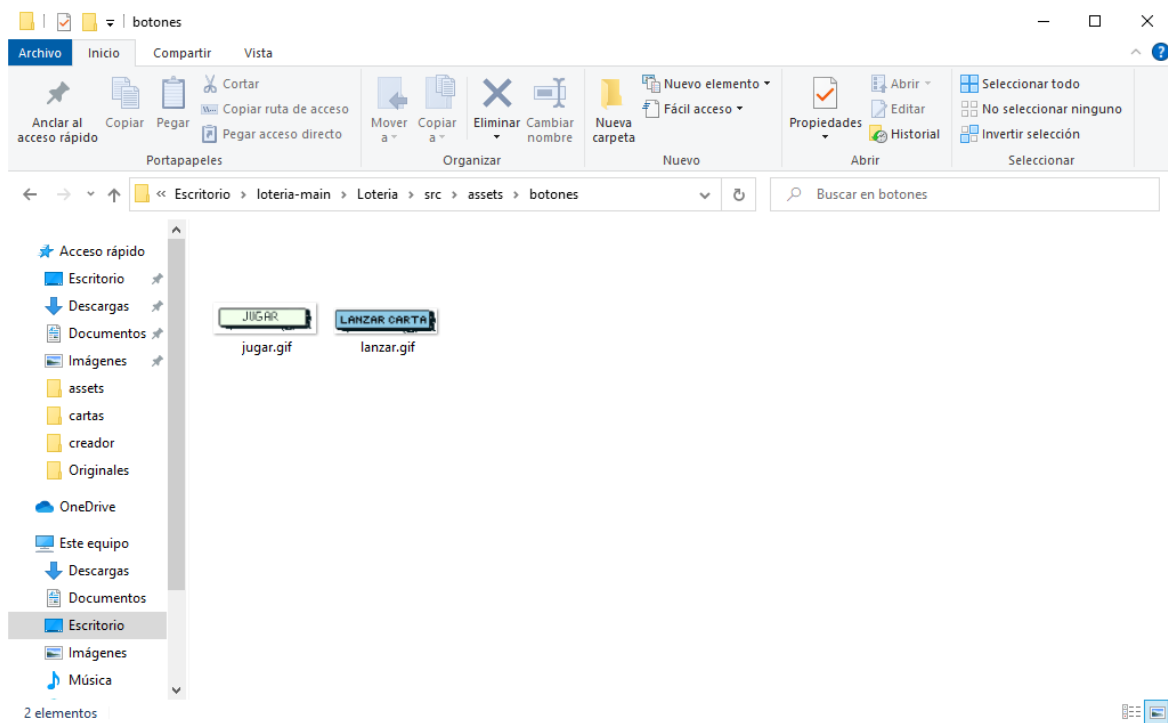
En la cual podremos modificar el audio que se escucha de fondo y el sonido que se produce al lanzar la carta.



Los audios deben de estar en formato .mp3 y deben de estar nombrados como “fondo” y “lanzamiento”.

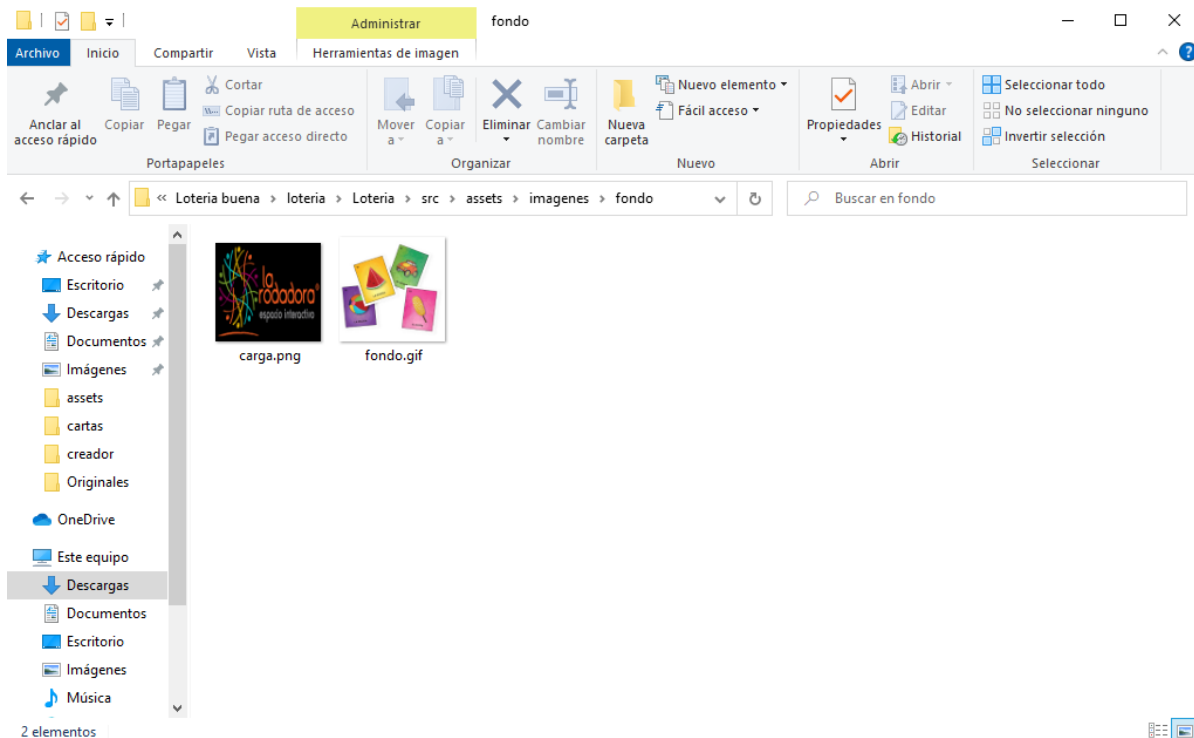
Modificación de botones de jugar y lanzamiento

Dentro de assets se encuentra una carpeta llamada Botones en donde se encuentran los botones que dicen jugar y lanzamiento los cuales para ser modificados deben de tener como nombres “jugar” y “lanzar”, deben de estar en formato .gif



Modificación imagen de fondo e imagen previa de las cartas

Dentro de la carpeta assets encontramos la carpeta imagenes que a su vez contiene una llamada fondo, dentro de fondo se encuentra el archivo que se muestra de fondo con formato .gif y de nombre “fondo”



La imagen con formato .png pertenece a la vista previa antes aparezca la carta en el juego, esta también puede ser modificada y debe de tener el nombre de “carga” con formato .png

Creación de ejecutable

Para crear la carpeta que contenga el archivo ejecutable de la aplicación será necesario correr los siguientes comandos en una terminal de visual studio code.

```
npm run electron-serve
```

Este comando aparte de ayudarnos a pre visualizar la aplicación, es la que hace que se cree una versión previa que ayuda a la construcción del programa sin este comando la aplicación quedara con una versión vieja de la aplicación.

```
npm run windows-release
```

Este comando nos generara una carpeta llamada “Ejecutable” la cual a su vez tendrá una carpeta llamada “lotería”, la cual es donde se encuentran todos los archivos y el ejecutable

```

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Ingeniería y Sistema\Downloads\loteria buena\loteria> cd loteria
PS C:\Users\Ingeniería y Sistema\Downloads\loteria buena\loteria> npm run windows-release

> loteria@0.8.1 windows-release
> electron-packager . loteria --overwrite --asar=true --platform-win32 --arch-ia32 --icon=src/assets/icon.ico --prune=true --out-Ejecutable --version-string.CompanyName=CE --version-string
.FileDescription=CE --version-string.ProductName="loteria"

WARNING: --asar does not take any arguments, it only has sub-properties (see --help)
Packaging app for platform win32 ia32 using electron v13.1.2
  
```

