

Guía de usuario de AppImage Launcher Creator

Creación de lanzadores gráficos para aplicaciones AppImage en Linux Mint

Linux Mint | AppImage | Tkinter | Menú de aplicaciones | 2026

Empresa	TECPROG WORLD E.I.R.L.
RUC	20608743252
Tipo de contribuyente	Empresa Individual de Responsabilidad Limitada
Estado del contribuyente	Activo
Condición del contribuyente	Habido
Domicilio fiscal	Mza. C Lote 43 Urb. Los Nísperos, San Martín de Porres, Lima, Lima
Actividad principal	7110 – Actividades de arquitectura e ingeniería y actividades conexas de consultoría técnica
Sitio web	https://tecprog-world-store.github.io/

Documento técnico de soporte para usuarios de Linux, software libre, escritorio XFCE y aplicaciones portables.

4 de junio de 2026

Índice

1. Organización de archivos para compilar esta guía	3
2. Objetivo del software	3
3. Problema que resuelve	3
4. Requisitos del sistema	3
5. Ejecución del programa	4
6. Descripción general de la interfaz	4
7. Sección 1: Archivo AppImage	5
8. Sección 2: Ícono de la aplicación	5
9. Sección 3: Datos del lanzador	6
9.1. Nombre visible	6
9.2. ID interno	6
9.3. Nombre genérico	6
9.4. Categorías	6
9.5. Comentario	7
9.6. Argumentos Exec	7
10. Sección 4: Opciones de instalación	7
10.1. Ejecutar en terminal	7
10.2. Activar StartupNotify	7
10.3. Sobrescribir lanzador si ya existe	7
10.4. Copiar AppImage a ~/.local/opt	7
11. Sección 5: Rutas que se crearán	7
12. Botones principales	8
12.1. Validar datos	8
12.2. Crear lanzador	8
12.3. Probar lanzador	8
12.4. Salir	8
13. Ejemplo completo: crear lanzador para Immersed	8
14. Estructura generada por el software	9
15. Contenido típico de un archivo .desktop	9
16. Solución de problemas frecuentes	9
16.1. La aplicación no aparece en el menú	9
16.2. El AppImage no abre	9
16.3. Error relacionado con FUSE	10
16.4. El ícono no aparece	10
16.5. El botón Probar lanzador falla	10
17. Buenas prácticas	10

18. Proyección técnica	10
19. Conclusión	10

1 Organización de archivos para compilar esta guía

Para que las figuras se inserten correctamente en este documento, se recomienda crear la siguiente estructura de carpetas:

```
GuiaAppImageLauncher/  
|-- guia_appimage_launcher_creator.tex  
|-- figuras/  
    |-- appimage_launcher_creator_01.png  
    |-- appimage_launcher_creator_02.png
```

Ubicar la primera captura de pantalla de la interfaz principal como:

```
figuras/appimage_launcher_creator_01.png
```

Ubicar la segunda captura de pantalla, donde se observa la sección de rutas y botones inferiores, como:

```
figuras/appimage_launcher_creator_02.png
```

Compilar el documento desde la carpeta principal con:

```
pdflatex guia_appimage_launcher_creator.tex  
pdflatex guia_appimage_launcher_creator.tex
```

2 Objetivo del software

AppImage Launcher Creator es una aplicación gráfica desarrollada en Python con Tkinter para crear lanzadores `.desktop` de aplicaciones distribuidas en formato **AppImage**. El propósito del software es evitar que el usuario tenga que escribir manualmente archivos de configuración, mover ejecutables de forma desordenada o depender de carpetas temporales como **Descargas**. Con esta herramienta, el usuario selecciona el archivo `.AppImage`, asigna un ícono, completa los metadatos de la aplicación y crea un lanzador visible en el menú de Linux Mint.

3 Problema que resuelve

Muchas aplicaciones modernas para Linux se distribuyen como archivos **AppImage**, los cuales pueden ejecutarse sin instalación tradicional. Sin embargo, si el archivo permanece en **Descargas**, puede eliminarse, renombrarse o moverse accidentalmente, provocando que el acceso directo deje de funcionar. Además, el usuario común puede no conocer la estructura XDG para crear lanzadores en `$HOME/.local/share/applications`, por lo que una interfaz gráfica simplifica el proceso y reduce errores operativos.

4 Requisitos del sistema

El software está orientado a Linux Mint y distribuciones compatibles con escritorio gráfico que respeten archivos `.desktop`. Para usarlo, se recomienda disponer de:

- Linux Mint, Ubuntu, Debian o una distribución derivada.
- Python 3 instalado.
- Tkinter instalado.
- Un archivo `.AppImage` válido.

- Una imagen de ícono en formato PNG, SVG, XPM, JPG, JPEG o WEBP.

Si Tkinter no se encuentra instalado, ejecutar:

```
sudo apt update
sudo apt install python3-tk
```

Para mejorar la previsualización de imágenes en formatos como JPG o WEBP, instalar Pillow:

```
sudo apt install python3-pil python3-pil.imagetk
```

5 Ejecución del programa

El programa puede ejecutarse directamente con Python:

```
python3 appimage_launcher_creator.py
```

También se puede conceder permiso de ejecución:

```
chmod +x appimage_launcher_creator.py
./appimage_launcher_creator.py
```

6 Descripción general de la interfaz

La interfaz está organizada en secciones numeradas. Cada sección solicita información específica para construir el lanzador de forma ordenada. La figura 1 muestra la parte superior de la aplicación, donde se selecciona el archivo AppImage, el ícono y los principales datos de identificación.

Figura 1: Interfaz principal de AppImage Launcher Creator para seleccionar el archivo AppImage, el ícono y los datos del lanzador.

La figura 2 muestra la parte inferior del programa, donde se previsualizan las rutas de instalación y se encuentran los botones de validación, creación y prueba del lanzador.

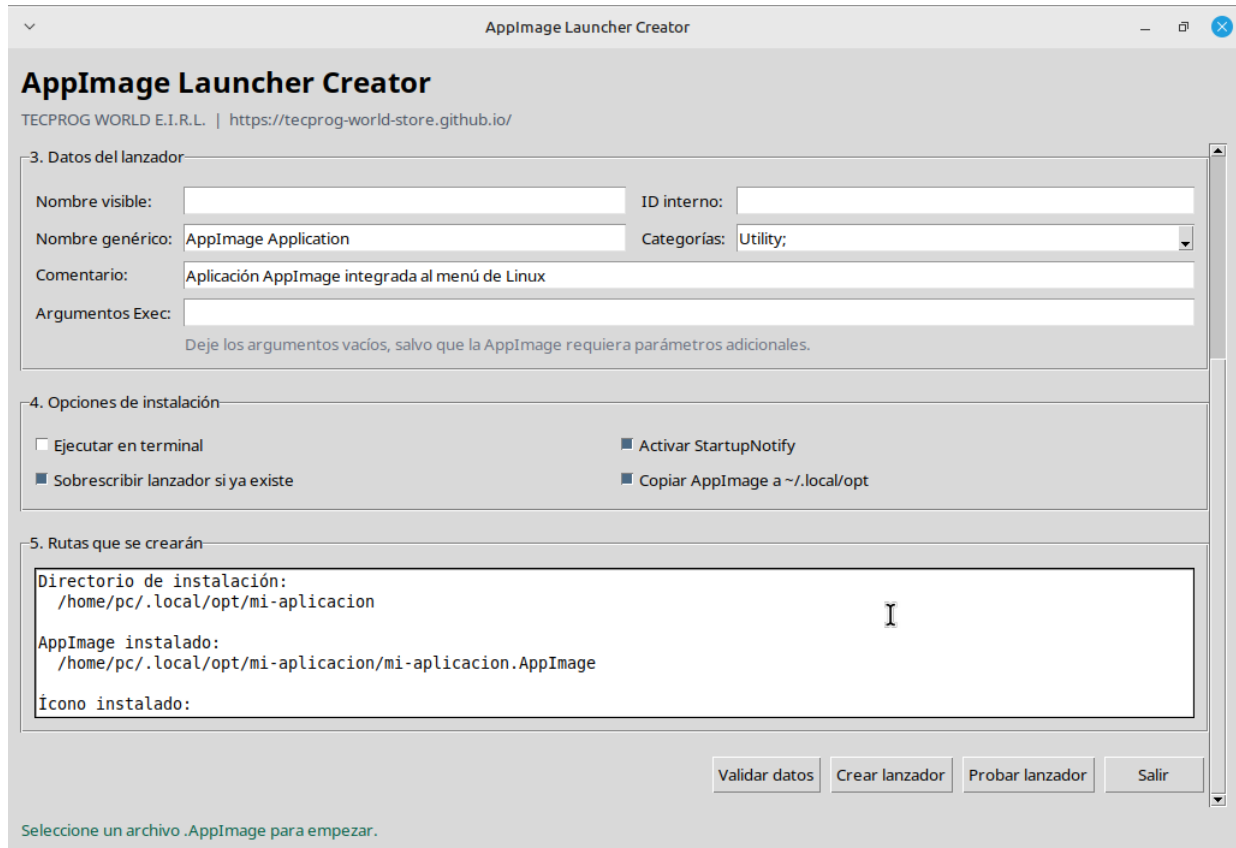


Figura 2: Sección de rutas generadas, opciones finales y botones principales del software.

7 Sección 1: Archivo AppImage

La primera sección permite seleccionar el ejecutable portable de la aplicación. El botón **Examinar** abre el explorador de archivos del sistema para ubicar un archivo con extensión `.AppImage`. Al seleccionar el archivo, el software puede proponer automáticamente un nombre visible basado en el nombre del ejecutable.

Ejemplo de ruta:
`/home/pc/Descargas/Immersed-x86_64.AppImage`

El archivo seleccionado debe existir y tener una extensión válida. Si el archivo no es reconocido como `AppImage`, la validación mostrará un mensaje de error.

8 Sección 2: Ícono de la aplicación

La segunda sección permite seleccionar la imagen que será usada como ícono en el menú de aplicaciones. El programa acepta formatos frecuentes como `PNG`, `SVG`, `XPM`, `JPG`, `JPEG` y `WEBP`. Para una integración más limpia en Linux Mint, se recomienda usar una imagen `PNG` cuadrada, por ejemplo de `256x256` o `512x512` píxeles.

Ejemplo de ruta:
`/home/pc/Descargas/immersed.png`

Cuando el ícono es `PNG`, Tkinter suele mostrar una vista previa básica. Para otros formatos, puede ser necesario instalar Pillow.

9 Sección 3: Datos del lanzador

Esta sección define los metadatos del archivo `.desktop`. Estos campos son importantes porque determinan cómo se verá y clasificará la aplicación en el menú.

9.1 Nombre visible

El campo **Nombre visible** corresponde al nombre que aparecerá en el menú de Linux Mint. Por ejemplo:

```
Immersed
QGIS Portable
Mi Aplicacion Cientifica
```

9.2 ID interno

El **ID interno** se usa para crear los nombres técnicos de carpetas y archivos. Debe ser corto, sin espacios y preferentemente en minúsculas.

```
immersed
qgis-portable
mi-aplicacion-cientifica
```

Este identificador será usado para construir rutas como:

```
~/.local/opt/immersed/
~/.local/share/applications/immersed.desktop
```

9.3 Nombre genérico

El campo **Nombre genérico** describe la clase de aplicación. Puede ser algo general como:

```
AppImage Application
Remote Desktop Client
Scientific Software
Development Tool
```

9.4 Categorías

Las categorías permiten que el entorno gráfico ubique la aplicación en una sección del menú. Algunos ejemplos válidos son:

```
Utility;
Network;RemoteAccess;Utility;
Development;
Graphics;
AudioVideo;
Office;
Education;
Science;Education;
System;
```

Para una aplicación como Immersed, una categoría razonable es:

```
Network;RemoteAccess;Utility;
```

9.5 Comentario

El comentario describe brevemente la función de la aplicación. Este texto puede aparecer como ayuda contextual en algunos entornos de escritorio.

```
Cliente Immersed para escritorio remoto, productividad y realidad virtual
```

9.6 Argumentos Exec

El campo **Argumentos Exec** debe dejarse vacío en la mayoría de casos. Solo se completa cuando la aplicación requiere parámetros adicionales al ejecutarse.

```
Ejemplo común:  
campo vacío
```

```
Ejemplo avanzado:  
--no-sandbox
```

10 Sección 4: Opciones de instalación

La sección de opciones controla el comportamiento final del lanzador.

10.1 Ejecutar en terminal

Esta opción debe activarse solamente si la aplicación requiere abrirse dentro de una terminal. Para aplicaciones gráficas AppImage, normalmente se mantiene desactivada.

10.2 Activar StartupNotify

Esta opción permite que el escritorio muestre retroalimentación visual al iniciar la aplicación. Se recomienda mantenerla activada.

10.3 Sobrescribir lanzador si ya existe

Esta opción permite reemplazar un lanzador anterior con el mismo ID interno. Es útil cuando se desea actualizar una aplicación o corregir su ícono.

10.4 Copiar AppImage a ~/.local/opt

Esta opción copia el ejecutable AppImage a una ruta estable dentro del usuario. Se recomienda mantenerla activada porque evita depender de la carpeta **Descargas**.

11 Sección 5: Rutas que se crearán

El programa muestra una vista previa de las rutas que se generarán. Esto permite verificar el resultado antes de crear el lanzador.

```
Directorio de instalacion:  
/home/pc/.local/opt/immersed  
  
AppImage instalado:  
/home/pc/.local/opt/immersed/immersed.AppImage  
  
Icono instalado:  
/home/pc/.local/share/icons/immersed.png
```

```
Lanzador .desktop:
/home/pc/.local/share/applications/immersed.desktop
```

Estas rutas siguen una organización local del usuario y no requieren permisos de administrador.

12 Botones principales

12.1 Validar datos

El botón **Validar datos** revisa que el archivo **AppImage** exista, que el nombre visible no esté vacío, que el ID interno sea válido y que la imagen seleccionada tenga un formato permitido.

12.2 Crear lanzador

El botón **Crear lanzador** copia el **AppImage**, copia el ícono, genera el archivo **.desktop**, concede permisos y actualiza la base local de aplicaciones.

12.3 Probar lanzador

El botón **Probar lanzador** intenta ejecutar el acceso creado mediante:

```
gtk-launch nombre-del-id
```

Para el caso de Immersed:

```
gtk-launch immersed
```

12.4 Salir

El botón **Salir** cierra el programa.

13 Ejemplo completo: crear lanzador para Immersed

Para instalar Immersed como aplicación de menú, completar los campos así:

Campo	Valor recomendado
Archivo AppImage	/home/pc/Descargas/Immersed-x86_64.AppImage
Ícono	/home/pc/Descargas/immersed.png
Nombre visible	Immersed
ID interno	immersed
Nombre genérico	Immersed Remote Desktop
Categorías	Network;RemoteAccess;Utility;
Comentario	Cliente Immersed para escritorio remoto, productividad y realidad virtual
Argumentos Exec	vacío

Luego presionar **Validar datos**, después **Crear lanzador** y finalmente buscar **Immersed** en el menú de aplicaciones.

14 Estructura generada por el software

Después de crear el lanzador, el sistema tendrá una estructura similar a la siguiente:

```
/home/pc/.local/opt/immersed/
|-- immersed.AppImage

/home/pc/.local/share/icons/
|-- immersed.png

/home/pc/.local/share/applications/
|-- immersed.desktop
```

15 Contenido típico de un archivo .desktop

El lanzador generado tiene una estructura similar a:

```
[Desktop Entry]
Version=1.0
Type=Application
Name=Immersed
GenericName=Immersed Remote Desktop
Comment=Cliente Immersed para escritorio remoto, productividad y realidad virtual
Exec="/home/pc/.local/opt/immersed/immersed.AppImage"
Icon=/home/pc/.local/share/icons/immersed.png
Terminal=false
StartupNotify=true
Categories=Network;RemoteAccess;Utility;
MimeType=
```

Este archivo permite que Linux Mint reconozca la aplicación y la muestre en el menú.

16 Solución de problemas frecuentes

16.1 La aplicación no aparece en el menú

Actualizar la base de datos de aplicaciones:

```
update-desktop-database ~/.local/share/applications
```

En XFCE, también puede reiniciarse el panel:

```
xfce4-panel -r
```

Si el problema continúa, cerrar sesión y volver a ingresar.

16.2 El AppImage no abre

Verificar que tenga permisos de ejecución:

```
chmod +x ~/.local/opt/immersed/immersed.AppImage
```

Ejecutarlo manualmente:

```
~/.local/opt/immersed/immersed.AppImage
```

16.3 Error relacionado con FUSE

Algunos AppImage requieren compatibilidad con FUSE. En Linux Mint moderno puede ser necesario instalar:

```
sudo apt update
sudo apt install libfuse2t64
```

16.4 El ícono no aparece

Verificar que el ícono exista:

```
ls -lh ~/.local/share/icons/immersed.png
```

Si se modificó el ícono, actualizar el lanzador o volver a crearlo con la opción de sobrescritura activada.

16.5 El botón Probar lanzador falla

Verificar que `gtk-launch` pueda ubicar el archivo:

```
gtk-launch immersed
```

Si no lo encuentra, revisar que exista:

```
ls -lh ~/.local/share/applications/immersed.desktop
```

17 Buenas prácticas

- Usar nombres de ID interno simples, sin espacios ni tildes.
- Mantener el AppImage instalado en `~/.local/opt`.
- Usar íconos en formato PNG cuadrado para mayor compatibilidad.
- No eliminar manualmente archivos dentro de `~/.local/opt` si el lanzador depende de ellos.
- Crear un respaldo del AppImage antes de actualizar una aplicación.

18 Proyección técnica

El software puede evolucionar hacia un paquete instalable `.deb`. Para ello, se recomienda definir un ícono oficial de la aplicación, instalar el ejecutable en `/usr/bin` o `/opt`, instalar su lanzador en `/usr/share/applications` y distribuir los recursos gráficos en `/usr/share/icons`. Esta evolución permitiría que *AppImage Launcher Creator* se instale como una herramienta regular dentro de Linux Mint.

19 Conclusión

AppImage Launcher Creator simplifica la integración de aplicaciones portables en Linux Mint al automatizar la creación de archivos `.desktop`, la copia de ejecutables y la asociación de íconos. La herramienta permite que usuarios técnicos y no técnicos organicen sus aplicaciones AppImage sin depender de la carpeta `Descargas` ni escribir manualmente configuraciones del menú. Este flujo mejora la trazabilidad del software instalado, reduce errores de ruta y facilita el uso cotidiano de aplicaciones portables dentro del escritorio Linux.

Contacto institucional

Empresa	TECPROG WORLD E.I.R.L.
RUC	20608743252
Actividad principal	Actividades de arquitectura e ingeniería y actividades conexas de consultoría técnica
Ubicación	San Martín de Porres, Lima, Perú
Sitio web	https://tecprog-world-store.github.io/