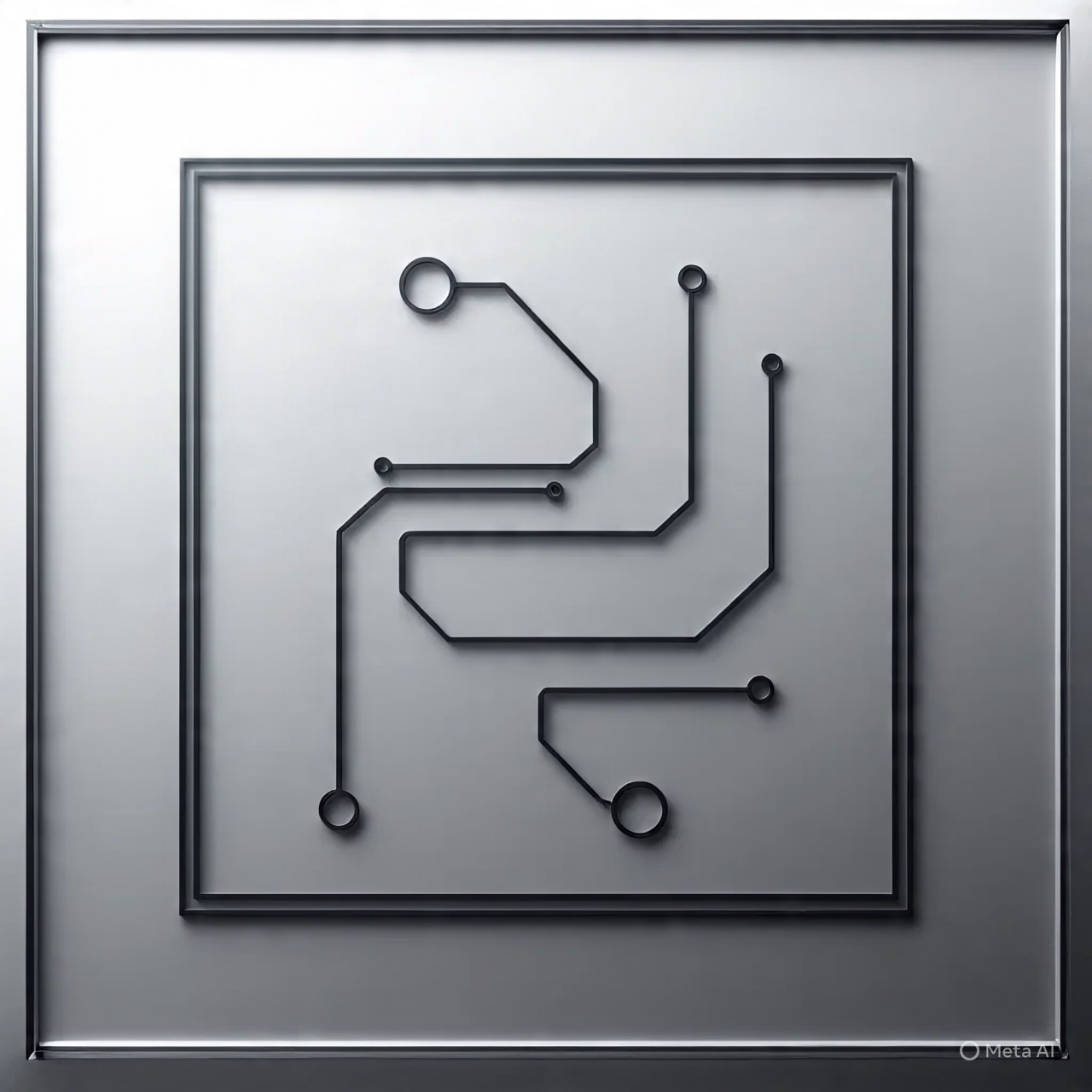
Manual Tecnico  
**Pensamiento de Programador: Explorando la Lógica de la Programación**



Logotipo Logic Programming

Universidad Tecnologica de Pereira   
luis Eduardo Muñoz Guerrero  
Agosto 2025

# Tabla de Contenido

* [Portada](#portada)
* [Tabla de Contenido](#tabla-de-contenido)
* [Descripción General](#descripción-general)
* [Requisitos del Sistema](#requisitos-del-sistema)
* [Instalación y Ejecución](#instalación-y-ejecución)
* [Estructura del Proyecto](#estructura-del-proyecto)
* [Arquitectura y Tecnologías](#arquitectura-y-tecnologías)
* [Configuración de Build y Empaquetado](#configuración-de-build-y-empaquetado)
* [Gestión de Recursos y Datos](#gestión-de-recursos-y-datos)
* [Solución de Problemas](#solución-de-problemas)
* [Bibliografía](#bibliografía)

# Descripción General

Logic Programming es una aplicación de escritorio educativa desarrollada en Angular y Electron, orientada a la enseñanza de lógica de programación mediante teoría, ejemplos y ejercicios interactivos con bloques visuales (Blockly). El usuario puede practicar y validar sus soluciones de manera autónoma y visual. La aplicación está pensada para funcionar sin conexión a internet y ser distribuida como ejecutable (.exe) en Windows.

# Requisitos del Sistema

* Windows 10/11 (x64)
* RAM: 2GB mínimo
* Procesador dual-core o superior
* No requiere conexión a internet para uso estándar

# Instalación y Ejecución

1. Descarga el instalador o el archivo .exe desde el repositorio oficial o medio proporcionado.
2. Haz doble clic en logic-programming.exe para iniciar la aplicación.
3. No requiere instalación adicional ni dependencias externas.



Pantalla de inicio

# Estructura del Proyecto

El proyecto está organizado siguiendo la arquitectura estándar de Angular y Electron. A continuación se muestra la estructura principal de carpetas y archivos:

logic-programming/  
├── dist/ # Archivos compilados Angular  
├── main.js # Entrada principal Electron  
├── electron-builder.yml # Configuración de empaquetado  
├── src/  
│ ├── app/ # Componentes, módulos y servicios  
│ ├── assets/ # Imágenes y archivos JSON  
│ └── ...  
├── documentos/ # Documentación y manuales

# Arquitectura y Tecnologías

* **Angular 16**: Framework principal para la interfaz y lógica de usuario.
* **Electron**: Empaquetado y ejecución como app de escritorio.
* **PrimeNG y TailwindCSS**: Estilos modernos y responsivos.
* **Blockly**: Motor de ejercicios de programación visual.
* **Archivos JSON**: Para cargar dinámicamente ejercicios y módulos.

# Configuración de Build y Empaquetado

El archivo de configuración electron-builder.yml define el empaquetado y el icono de la app:

win:  
 target:  
 - target: nsis  
 arch: [x64]  
 icon: icono-logic-programing.ico

El icono de la aplicación se encuentra en la raíz del proyecto como icono-logic-programing.ico y se utiliza tanto para el ejecutable como para los accesos directos.

Para generar el ejecutable, se usa el comando:

npm run electron:dist

# Gestión de Recursos y Datos

Los ejercicios y módulos se almacenan en archivos JSON dentro de src/assets/jsons-base/. Las imágenes y recursos visuales están en src/assets/img/ y documentos/img-documentacion/. No se requiere base de datos externa; toda la información se gestiona localmente.

# Solución de Problemas

* Si la app no abre, verifica que tu antivirus no bloquee el archivo.
* Si la interfaz se ve en blanco, asegúrate de que los archivos de recursos no hayan sido movidos o eliminados.
* Para soporte técnico, contactar al desarrollador vía GitHub.

# Bibliografía

American Psychological Association. (2020). *Publication manual of the American Psychological Association* (7th ed.). https://doi.org/10.1037/0000165-000

Angular. (2025). Documentación oficial. https://angular.io/

Electron. (2025). Documentación oficial. https://www.electronjs.org/docs

Blockly. (2025). Documentación oficial. https://developers.google.com/blockly

Repositorio del proyecto: https://github.com/Napssters/logic-programming