

ROUPY

*DOCUMENTACION
LENGUAJE DE PROGRAMACION*

Roupy Core - Etapa 1: Desarrollo Inicial

Meta principal:

Roupy Core - Etapa 1 tiene como propósito sentar las bases de un lenguaje de programación propio, diseñado desde cero con un enfoque accesible, funcional y adaptable. Esta etapa inicial se centra en desarrollar una versión mínima pero operativa de *Roupy*, capaz de ejecutar instrucciones básicas, declarar variables simples y mostrar resultados en pantalla.

El objetivo es que **Roupy pueda ser utilizado como lenguaje principal** en dos contextos muy distintos pero complementarios:

1. Desarrollo de videojuegos 2D simples

Esto implica dotar al lenguaje de herramientas que permitan controlar sprites, físicas simples, lógica de juego, ciclos de ejecución y más adelante integración con un motor gráfico. Desde esta primera etapa, la intención es construir una sintaxis clara, personalizable y con orientación a la interacción visual y lógica de juegos.

2. Automatización de tareas básicas y scripting

Se busca también que Roupy pueda usarse en el desarrollo de scripts simples que automaticen procesos cotidianos o repetitivos, como generación de archivos, respuestas automatizadas, manipulación de texto, o más adelante tareas programadas. Esto lo vuelve útil no solo para nichos como los videojuegos, sino también para uso personal o profesional en entornos de productividad.

En resumen:

La **meta principal** de esta primera etapa de desarrollo es **crear un lenguaje propio, con sintaxis sencilla**, enfocado en ser:

- Intuitivo para programadores principiantes.
- Suficientemente robusto como para crecer hacia motores gráficos propios.
- Adaptable para otros tipos de proyectos a mediano plazo.
- Una herramienta educativa, personalizable y 100% abierta.

Esta base no solo servirá como proyecto de aprendizaje profundo, sino también como el pilar sobre el cual se construirá una suite completa de herramientas: IDE, motor gráfico, y más adelante lenguajes hermanos (como Rouj o RouPlus) unificados en un ecosistema de desarrollo llamado **RouEngine**.

HISTORIAL DE VERSIONES

Versión 0.01 – Sistema de salida básico (output)

Fecha estimada de implementación: 26/04/2025

Características implementadas:

- *Se habilitó la capacidad de mostrar texto estático en pantalla.*
- *Se introdujo el comando show, encargado de imprimir texto plano o cadenas estáticas.*

(Codigo .rp)

```
show = "Hola, mundo"
```

(Salida)

```
Hola, mundo
```

Objetivo técnico alcanzado:

Establecer el primer *parser básico* del lenguaje, capaz de interpretar una línea de instrucción simple con reconocimiento de cadenas (string) e impresión estándar.

Esta versión permitió confirmar que la extensión .rp podía ejecutarse a través del intérprete personalizado hecho en Python.

Versión 0.02 – Declaración y almacenamiento de variables (string e int)

Fecha estimada de implementación: 26/04/2025

Características implementadas:

- *Capacidad para declarar y almacenar variables de tipo entero (int) y texto (string).*
- *Mapeo de variables a través de un diccionario interno en Python.*

(Codigo .rp)

```
x = 15
```

```
nombre = "Roussel"
```

Objetivo técnico alcanzado:

Agregar almacenamiento dinámico de datos utilizando estructuras internas del intérprete (dict) y permitir referencias posteriores a variables en otras instrucciones. Fue la base para futuras operaciones y manipulación de datos.

Versión 0.03 – Lectura y visualización de variables (dynamic output)

Fecha estimada de implementación: 27/04/2025

Características implementadas:

- *Se extendió la funcionalidad del comando show para aceptar tanto texto como variables.*
- *El intérprete puede ahora concatenar y mostrar texto combinado con valores dinámicos.*

(Codigo .rp)

nombre = "Roussel"

edad = 20

show = "Hola, me llamo: ", nombre, " y tengo ", edad, " años"

(Salida)

Hola, me llamo: Roussel y tengo 20 años

-

3.

4.

