

# SCRIPT DEL TUTORIAL

**Propósito:** Definir exactamente qué dice y hace el tutorial paso a paso

---

## OBJETIVOS DEL TUTORIAL

**Al completar el tutorial, el jugador debe:**

- Saber colocar edificios
- Entender que los haces conectan automáticamente
- Haber creado su primer quark
- Conocer la meta del juego (llegar a ADN)
- Saber que puede presionar F1 para más ayuda

**Duración objetivo:** 2-3 minutos

---

## SECUENCIA DE PASOS

### PASO 0: Introducción

**Trigger:** Al iniciar nueva partida (si `tutorial_completed == false`)

**UI:**

```
|  ¡Bienvenido a Micro Architect! |
| |
| Construye materia desde energía |
| básica hasta crear una cadena |
| de ADN. |
| |
| ¿Quieres hacer el tutorial? |
| |
| [Sí, enséñame] [No, ya sé jugar] |
```

**Acciones:**

- Si "Sí" → Continuar a PASO 1
  - Si "No" → `tutorial_completed = true`, cerrar
- 

### PASO 1: Colocar Siphon

**Trigger:** Despues de aceptar tutorial

**Highlight:** Icono de Siphon en HUD (o menú construcción)

**Texto:**

Paso 1/5: Coloca un SIPHON

Los siphons extraen energía del vacío.

Deben colocarse en losetas AZULES (energía).

Selecciona el Siphon y colócalo en una loseta azul.

**Validación:** Esperar a que se coloque un Siphon

**Al completar:**

- Highlight desaparece
- Sonido de éxito
- Continuar a PASO 2

---

## PASO 2: Entender Haces

**Trigger:** Siphon colocado

**Highlight:** El siphon colocado (con glow)

**Texto:**

Paso 2/5: ¡Bien hecho!

Observa el haz de luz que emite el siphon.  
Los haces conectan edificios automáticamente  
cuando están cerca.

Esto transporta energía de un edificio a otro.

**Validación:** Temporizador de 3 segundos (dar tiempo a leer)

**Al completar:** Continuar a PASO 3

---

## PASO 3: Colocar Compressor

**Trigger:** Despues de 3 segundos

**Highlight:** Icono de Compressor en HUD

**Texto:**

Paso 3/5: Ahora coloca un COMPRESSOR

Los compressors transforman 10 energía básica  
en 1 energía comprimida.

Colócalo CERCA del siphon para que se conecten.

**Validación:** Esperar a que se coloque un Compressor

**Al completar:**

- Verificar si está conectado al Siphon
- Si NO → Mostrar hint: "Muévelo más cerca del siphon"
- Si SÍ → Continuar a PASO 4

## PASO 4: Primera Producción

**Trigger:** Compressor colocado y conectado

**Highlight:** El compressor (con glow)

**Texto:**

Paso 4/5: ¡Conexión establecida!

Los pulsos de energía viajarán del siphon al compressor automáticamente.

Espera a que se cree energía comprimida...

**Validación:** Esperar a que el compressor produzca 1 energía comprimida

**Al completar:**

- Animación de celebración
- Sonido especial
- Continuar a PASO 5

---

## PASO 5: Objetivo Final

**Trigger:** Primera energía comprimida creada

**Highlight:** Ninguno (pantalla completa)

**Texto:**

Paso 5/5: ¡Primera energía comprimida creada!

OBJETIVO DEL JUEGO:

Sigue la cadena de producción hasta crear ADN:

Energía → Quarks → Protones → Átomos →

Moléculas → ADN

Usa más edificios (Mergers, Constructors) para avanzar en la cadena.

Presiona F1 en cualquier momento para ayuda.

No mostrar este tutorial de nuevo

[ ¡Entendido, a construir! ]

**Validación:** Click en botón

**Al completar:**

- Si checkbox marcado → `tutorial_completed = true`
- Guardar preferencia
- Cerrar tutorial

- ¡Juego libre!
- 

## ELEMENTOS VISUALES

### Highlight System

Shader outline/glow:

```
// Aplicar a edificio o UI element
shader_type canvas_item;

uniform vec4 outline_color : hint_color = vec4(1.0, 1.0, 0.0, 1.0);
uniform float outline_width : hint_range(0.0, 10.0) = 2.0;

void fragment() {
    // ... shader code para outline amarillo pulsante
}
```

**Animación:** Pulso suave (1s ciclo)

---

### Panel de Instrucciones

**Posición:** Parte inferior-centro de la pantalla

**Tamaño:** 600x150 px

**Background:** Semi-transparente oscuro (#000000, 80% opacidad)

**Texto:** Blanco, centrado, tamaño 16

**Padding:** 20px

---

### Botón "Saltar Tutorial"

**Posición:** Esquina superior derecha del panel

**Texto:** "Saltar tutorial" (pequeño, 12px)

**Color:** Gris claro

**Acción:** Cierra tutorial inmediatamente, `tutorial_completed = true`

---

## IMPLEMENTACIÓN TÉCNICA

### Estructura de Archivos

```
scenes/ui/tutorial_system.tscn
├── CanvasLayer (overlay oscuro)
│   ├── ColorRect (fondo semi-transparente)
│   ├── PanelContainer (panel de instrucciones)
│   │   ├── VBoxContainer
│   │   │   ├── Label (título "Paso X/5")
│   │   │   ├── RichTextLabel (texto instrucciones)
│   │   │   ├── HBoxContainer (botones)
│   │   │   │   ├── CheckBox ("No mostrar de nuevo")
│   │   │   │   └── Button ("Siguiente" / "Entendido")
```

```
| | | | └ Button ("Saltar tutorial")
| └ [Highlight visual - shader aplicado dinámicamente]
```

### Script Principal: tutorial\_manager.gd

```
# scripts/managers/tutorial_manager.gd
extends Node

signal step_completed(step_number: int)
signal tutorial_finished()

enum TutorialStep {
    INTRO,
    PLACE_SIPHON,
    UNDERSTAND_BEAMS,
    PLACE_COMPRESSOR,
    FIRST_PRODUCTION,
    FINAL_MESSAGE
}

var current_step: TutorialStep = TutorialStep.INTRO
var tutorial_active: bool = false

# Referencias a UI
@onready var overlay = $TutorialOverlay
@onready var instruction_panel = $TutorialOverlay/InstructionPanel
@onready var step_label = $"../StepLabel"
@onready var text_label = $"../TextLabel"

func _ready():
    # Verificar si tutorial ya fue completado
    if SaveSystem.get_value("tutorial_completed", false):
        queue_free() # No mostrar tutorial
        return

    start_tutorial()

func start_tutorial():
    tutorial_active = true
    show_step(TutorialStep.INTRO)

func show_step(step: TutorialStep):
    current_step = step

    match step:
        TutorialStep.INTRO:
            show_intro()
        TutorialStep.PLACE_SIPHON:
            show_place_siphon()
        TutorialStep.UNDERSTAND_BEAMS:
            show_understand_beams()
```

```

# ... etc

func show_intro():
    step_label.text = "¡Bienvenido!"
    text_label.text = "Construye materia desde energía básica hasta ADN.\n¿Quieres hacer el tutorial?"
    # Mostrar botones Sí/No

func _on_building_placed(building_type: String):
    if not tutorial_active:
        return

    # Validar si es el edificio correcto para el paso actual
    match current_step:
        TutorialStep.PLACE_SIPHON:
            if building_type == "siphon":
                complete_current_step()
        TutorialStep.PLACE_COMPRESSOR:
            if building_type == "compressor":
                complete_current_step()

func complete_current_step():
    emit_signal("step_completed", current_step)

    # Avanzar al siguiente paso
    match current_step:
        TutorialStep.INTRO:
            show_step(TutorialStep.PLACE_SIPHON)
        TutorialStep.PLACE_SIPHON:
            show_step(TutorialStep.UNDERSTAND_BEAMS)
        # ... etc
        TutorialStep.FINAL_MESSAGE:
            finish_tutorial()

func finish_tutorial():
    tutorial_active = false

    # Guardar que tutorial fue completado (si checkbox marcado)
    if $CheckBox.button_pressed:
        SaveSystem.set_value("tutorial_completed", true)

    emit_signal("tutorial_finished")
    overlay.hide()
    queue_free()

func skip_tutorial():
    SaveSystem.set_value("tutorial_completed", true)
    finish_tutorial()

```

---

## Integración con SaveSystem

**Modificar scripts/autoload/save\_system.gd :**

```
# Añadir a estructura de guardado
var game_state = {
    "inventory": {},
    "buildings": [],
    "tutorial_completed": false # ← NUEVO
}
```

## TESTING CHECKLIST

- Tutorial se muestra en nueva partida
- Opción "No, ya sé jugar" funciona
- Highlights visuales se muestran correctamente
- Paso 1: Colocar siphon valida correctamente
- Paso 3: Colocar compressor valida proximidad
- Paso 4: Detecta primera energía comprimida
- Botón "Saltar" funciona en cualquier momento
- Checkbox "No mostrar" guarda preferencia
- Tutorial no se muestra en partidas posteriores si se desactivó
- Tutorial es rejuable si NO se marcó checkbox

## TEXTOS ALTERNATIVOS (A/B Testing)

### Versión A: Formal

"Coloca un Siphon en una loseta de energía (azul)"

### Versión B: Casual

"¡Pon un Siphon en una baldosa azul! 🚧"

**Decisión:** Elegir según tono del juego

## EXPANSIÓN FUTURA

Cuando se implementen más mecánicas:

### Tutorial Extendido (opcional):

- Paso 6: Colocar Merger
- Paso 7: Crear primer quark
- Paso 8: Constructor y producción avanzada

### Sistema modular:

- Tutorial básico (pasos 1-5) siempre
- Tutorial avanzado (pasos 6-8) opcional después

---

**Última actualización:** 2025-02-01