

# GeneratorPlugin-AI – Documento Técnico Completo

## ## Descripción General

GeneratorPlugin-AI es un plugin avanzado para servidores Minecraft (Paper/Spigot) que permite crear, modificar y ejecutar plugins personalizados directamente desde el juego mediante comandos o una interfaz gráfica. El sistema interpreta instrucciones en lenguaje natural y las traduce en código Java o scripts listos para ejecutarse.

---

## ## Objetivo

Permitir a cualquier jugador con permisos administrativos crear mini-plugins, automatizaciones o mecánicas personalizadas sin necesidad de conocimientos de programación ni reiniciar el servidor.

---

## ## Funcionalidades Clave

### 1. \*\*Generador Natural (IA o Parser interno)\*\*

- Interpreta lenguaje natural ("cuando pase X, haz Y").
- Traduce instrucciones a código Java con eventos Bukkit.
- Genera scripts reutilizables (.js, .py o .yaml).

### 2. \*\*Interfaz GUI In-Game\*\*

- Menú visual para elegir eventos, acciones y condiciones.
- Ejemplo: "Al hacer clic derecho con un bloque de diamante, da 5 esmeraldas".

### 3. \*\*Compilador Integrado\*\*

- Usa `javax.tools.JavaCompiler` para generar código y compilar en vivo.
- Carga los nuevos plugins sin reiniciar el servidor mediante `PluginClassLoader`.
- Guarda plugins generados en `/plugins/GeneratorAI/generated/`.

### 4. \*\*Módulo Script Engine\*\*

- Soporte para scripts ligeros (Groovy, JavaScript o Python).
- Los scripts pueden editarse en tiempo real sin recompilar.
- Sistema de recarga automática con `/gen reload`.

### 5. \*\*Modo Aprendizaje\*\*

- Aprende de las creaciones del usuario.
- Permite modificaciones con comandos tipo "igual que el anterior, pero con 10 diamantes".

### 6. \*\*Seguridad y Control\*\*

- Validación de código antes de compilar.
- Control de permisos por jugador.
- Logs en `/plugins/GeneratorAI/logs/`.

### 7. \*\*Exportador Automático\*\*

- Permite exportar el plugin generado como `.jar` listo para compartir.

---

## ## Ejemplos de Uso

- `/ai createplugin giveemeralds`
- Instrucción: "Cuando el jugador use un bloque de diamante, dale 5 esmeraldas."
- `/ai createplugin fireevent`
- Instrucción: "Cuando alguien diga 'boom' en el chat, que explote TNT cerca."
- `/ai createplugin autofood`
- Instrucción: "Cada 30 minutos, todos los jugadores reciben comida."

---

## ## Arquitectura Técnica

**\*\*Estructura de carpetas:\*\***

...

/GeneratorAI

■

- /generated/ # Plugins creados automáticamente
- /scripts/ # Scripts personalizados
- /config.yml # Configuración general
- /logs/ # Errores y registros
- main.java # Clase principal del plugin

...

**\*\*Clases principales:\*\***

- `Main` – Inicializa el plugin, gestiona comandos y carga dinámica.
- `CommandHandler` – Procesa comandos `/ai` y `/gen`.
- `AllInterpreter` – Interpreta instrucciones naturales y genera código Java.
- `CodeBuilder` – Genera estructuras básicas (eventos, listeners, comandos).
- `PluginCompiler` – Compila y carga el código generado en tiempo real.
- `ScriptManager` – Maneja scripts Groovy/JS/Python.
- `PluginManager` – Controla carga, descarga y recarga de módulos generados.
- `SecurityManager` – Valida sintaxis y permisos.
- `GUIManager` – Interfaz visual dentro del juego.

---

## ## Flujo de Ejecución

1. Usuario ejecuta `/ai createplugin`.
2. Sistema solicita descripción ("¿Qué quieres que haga?").
3. AllInterpreter analiza texto → genera código Java.
4. CodeBuilder crea estructura del plugin.
5. PluginCompiler compila y carga el plugin.
6. El jugador puede usarlo de inmediato sin reiniciar el servidor.

---

## ## Tecnologías Sugeridas

- \*\*Java 17+\*\*
- \*\*PaperMC / Spigot 1.20+\*\*
- \*\*Librerías:\*\*
  - JavaCompiler API (`javax.tools`)
  - Bukkit API

- Optional: Groovy / Nashorn (JS Engine)
- Gson o Jackson (para manejo de JSON)
- **\*\*Base de datos opcional:\*\*** SQLite o MySQL (para historial de creaciones).

---

## ## Ejemplo de Configuración

config.yml:

---

```
auto-save: true
language: "es"
allow-scripts: true
max-plugin-size: 500KB
permissions:
  admin: generator.admin
  create: generator.create
  compile: generator.compile
```

---

---

## ## Potencial de Expansión

- Integración con OpenAI o modelos locales para comprensión natural más avanzada.
- Sistema de plantillas compartidas entre usuarios.
- Marketplace de scripts generados.
- Sandbox de prueba dentro del servidor.
- Generación visual con bloques tipo "plugin-builder".

---

## ## Licencia y Propósito

Este plugin puede distribuirse como Premium o SaaS. Su objetivo es democratizar la creación de plugins dentro de Minecraft, potenciando la creatividad y automatización de los servidores.