

# Continuous Integration tool for Godot using Jenkins

---

Se plantea una herramienta de integración continua para Godot usando Jenkins. Se configurará una plantilla de ejemplo en Godot para al menos una plataforma. La integración continua se implementará usando Jenkins. Jenkins corre como proceso local y mediante plugins realiza acciones. Jenkins correrá en una red local, dos ordenadores conectados al mismo router.

## Godot Engine 4.2.2

Motor de código abierto. Nacido en 2001 como código cerrado pero usado desde 2014 para hacer juegos. Notables ejemplos en la saga Deponia, Sonic Colors y el próximo Slay The Spire 2.

- Godot corre en un ejecutable autocontenido (Open Source, no requiere instalación)
- Existe ejecutable para versión GUI y para versión consola de comandos (headless en modo consola)
- Godot crea un proyecto llamado template, este se modifica con el ejecutable (tanto consola como GUI)
- El template puede ser compilado/construido de diferentes formas/maneras para diversas plataformas
- La información de build de template tiene que setearse inicialmente/de manera manual antes de llamar a la build, requiere de una descarga

## Jenkins 2452.1

Se usa en la industria por ser de código abierto y permitir automatización en múltiples sistemas de control de versiones mediante los plugins de la comunidad.

- El repo podría tirar en Github Actions pero no es nuestro (Microsoft)
- La automatización va por Jenkins, primer paso para montar ecosistema propio (Open Source)
- Jenkins necesita correr en un servidor, este puede ser propio (controlamos los datos)
- Jenkins usa lenguaje de scripting, groovy, con sintaxis de Java
- Compatible con centenares de plugins, uno de ellos Github
- El pipeline de Jenkins desencadena los jobs que configuremos

### Jenkins - Documentación inicial

- [Configuración inicial](#)
- [Detección de commits API de Github](#)
- [Configuración plugin Github](#)
- [Pipelines de Jenkins](#)

### Godot Engine - Documentación inicial

- [Exportar plantillas](#) == Incluye codificación para el exe ==
- [Video build desde CMD](#)
- [Correr Godot desde Línea de Comandos](#)
- [Build desde línea de comandos](#)

---

Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo	Planificación
-------	--------	-----------	--------	---------	--------	---------	---------------

---

Lunes	Martes	Miercoles	Jueves	Viernes	Sabado	Domingo	Planificacion
				24	25	26	Repositorio inicial. Plantilla de Godot
26	27	28	29	30	31	1	Godot - Builds por scripting (Windows+plataforma extra)
2	3	4	5	6	7	8	Jenkins - Configuración de plugin Github / Jobs de build
9	10	11	12	13	14	15	Jenkins - Configuración de servidor. Build en local/ servidor
16	17	18	19	20	21	22	Rama secundaria. Jobs en Pull
23	24*	25	26	27	28	29	Contingencias/Presentación
30	1*						Presentación

Todo el proceso será acompañado de la documentación tanto en comentarios de código como en el documento final.

La semana final se terminará la documentación, se creará una infográfica del ciclo de desarrollo y de las pipelines y se preparará la presentación

Idea de montar un sistema que te avise de la rotura de una build por aplicacion de mensajeria externa (mensja en discord)