códigofacilito

Bootcamp: GitHub Actions

Mejores prácticas

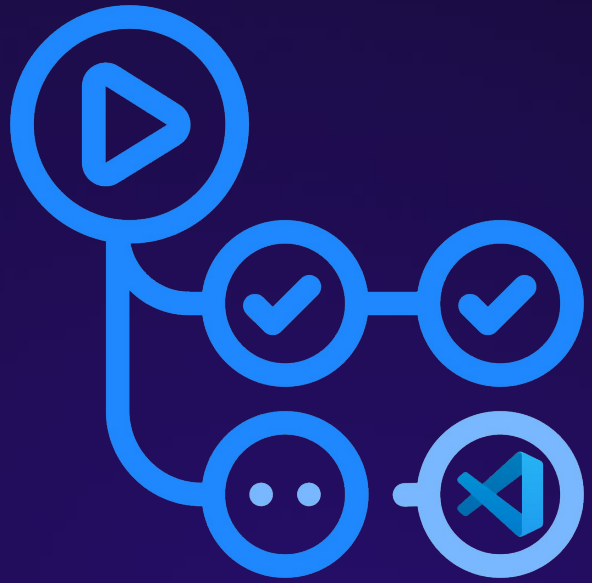
Fernando Garcia

códigofacilito

Organizar los steps de forma clara

Usa nombres descriptivos.

Agrupa por propósito: build, test, deploy.



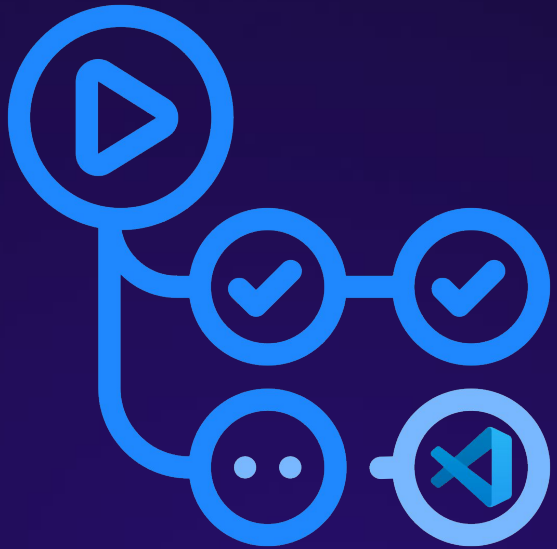
códigofacilito

Nunca harcodear información

Hacer uso de variables.

Hacer uso de secretos.

Evitar exponer cualquier tipo de información.

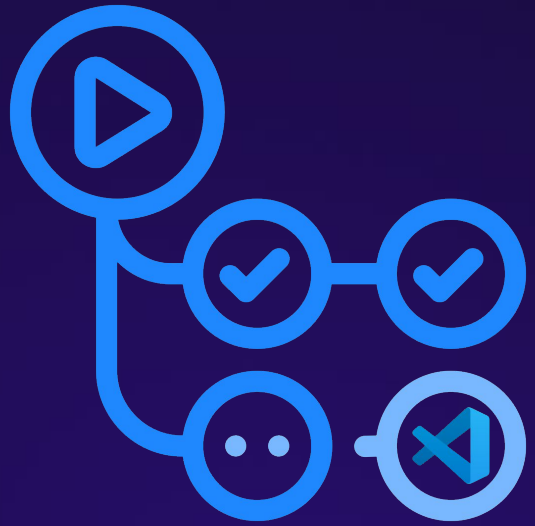


códigofacilito

Trabajar con artefactos

Guarda archivos importantes generados durante el workflow: logs, reportes de pruebas, binarios.

Mejora el debugging: puedes revisar qué salió mal después.

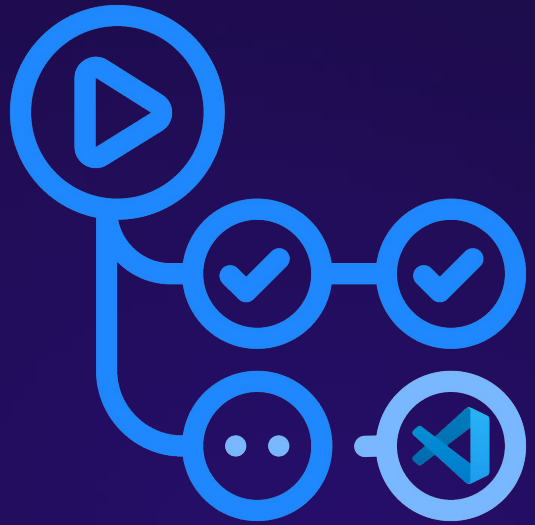


códigofacilito

Usa versiones concretas

Usa versiones concretas para todo.

Evita que actualizaciones automáticas rompan tu pipeline.

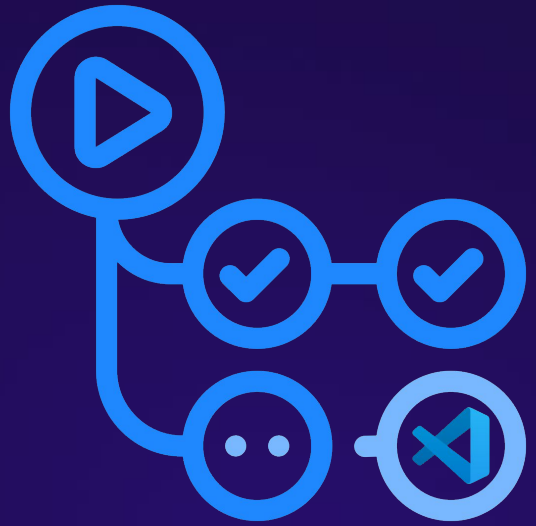


códigofacilito

Define las ramas a escuchar en los eventos

No dispares workflows innecesarios en ramas que no importan.

Usa on: push: branches: o on: pull_request: branches: para limitar la ejecución.



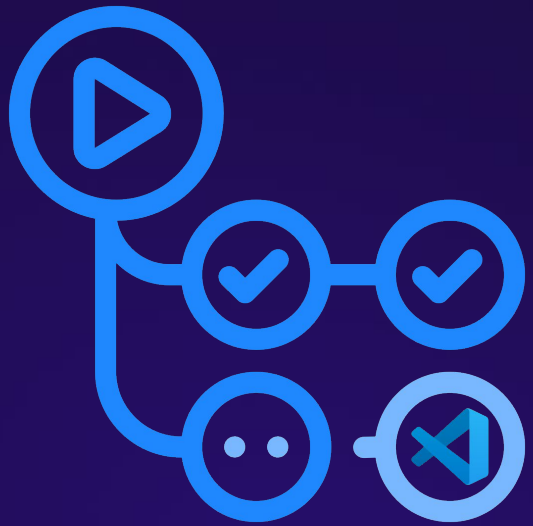
códigofacilito

Crear acciones reusables

Hacer uso de custom actions para evitar duplicidad de lógica.

Encapsulamiento de lógica compleja.

Parámetros flexibles.



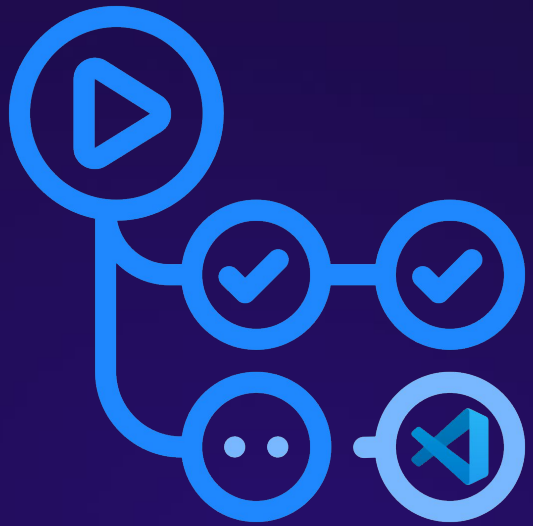
códigofacilito

Crear workflows reusables

Hacer uso de reusable workflows para evitar duplicidad de lógica.

Encapsulamiento de lógica compleja.

Parámetros flexibles.



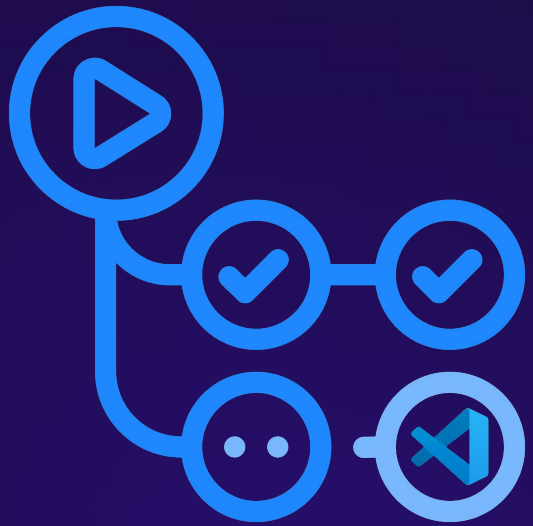
códigofacilito

Revisar las GitHub Actions de terceros

Puedes detectar código malicioso o inseguro.

Verificas si la Action es mantenida activamente, si tiene issues abiertos, o si está desactualizada.

Previene fallos difíciles de diagnosticar en el pipeline.



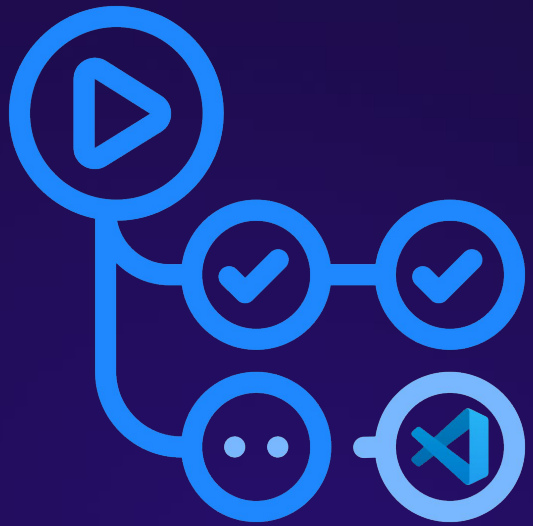
códigofacilito

Uso de Cache

Reduce drásticamente los tiempos de ejecución al evitar reinstalar o recompilar dependencias en cada run.

Evitas descargar paquetes o construir assets que ya están listos.

Al reducir el tiempo total de los jobs, disminuyes el consumo de minutos facturables (en planes privados).

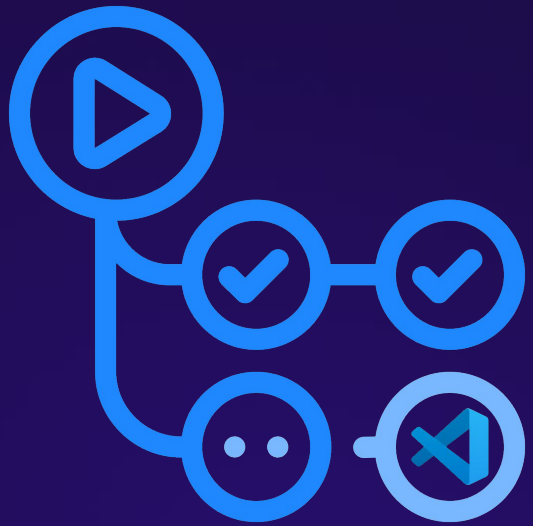


códigofacilito

Configurar permisos mínimos al GITHUB_TOKEN

Reducir los permisos del GITHUB_TOKEN minimiza el riesgo si alguna Action es comprometida o maliciosa.

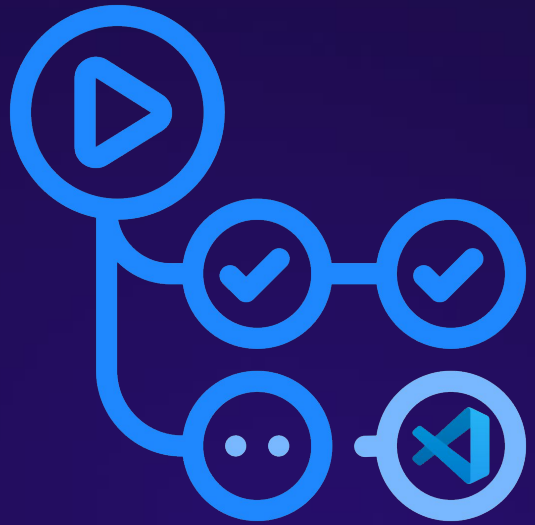
Reduce la superficie de ataque.



códigofacilito

Añadir un badge del status del workflow en tu README

Mejora la visibilidad: todos pueden ver si está fallando o pasando.

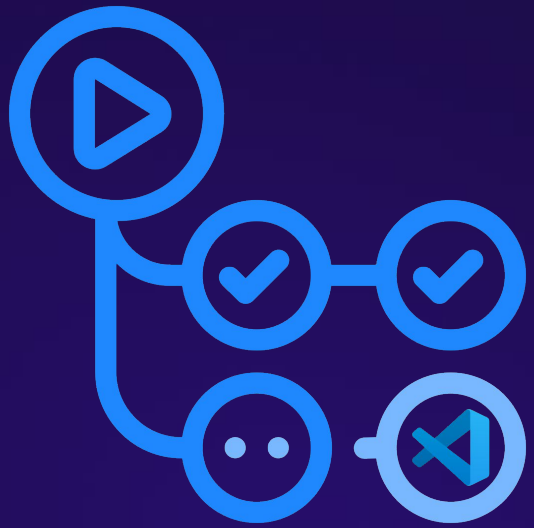


códigofacilito

Recomendaciones de monitoreo y debugging

Usa logs (echo,) en pasos clave.

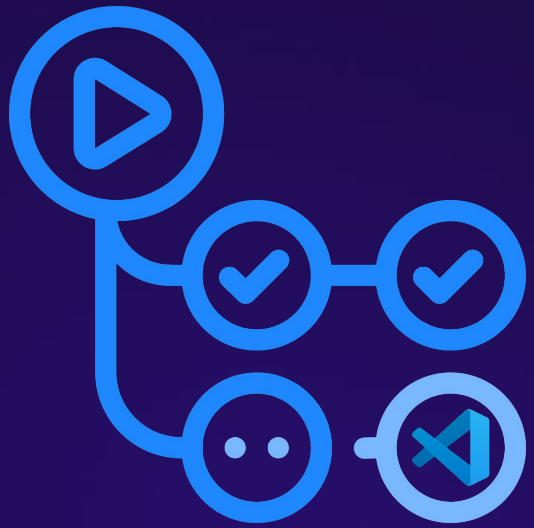
Sube artefactos relevantes como reportes de pruebas.



códigofacilito

Hacer uso de notificaciones

Configura notificaciones (Slack, teams, correo, etc.) para alertas inmediatas.

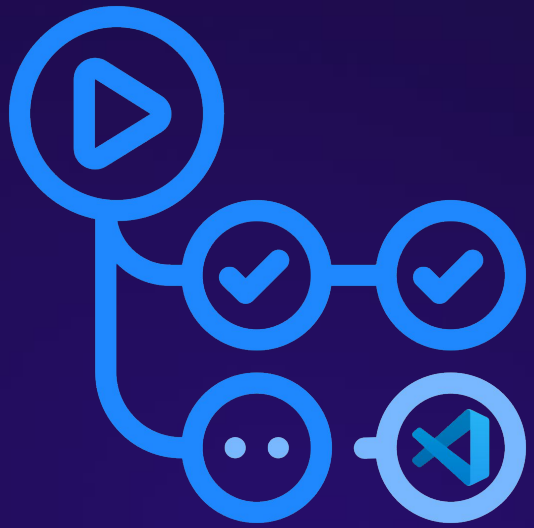


códigofacilito

Hacer uso de paralelismo

Al ejecutar múltiples tareas al mismo tiempo, el workflow termina más rápido.

Ideal cuando tienes muchas pruebas, builds o procesos independientes.



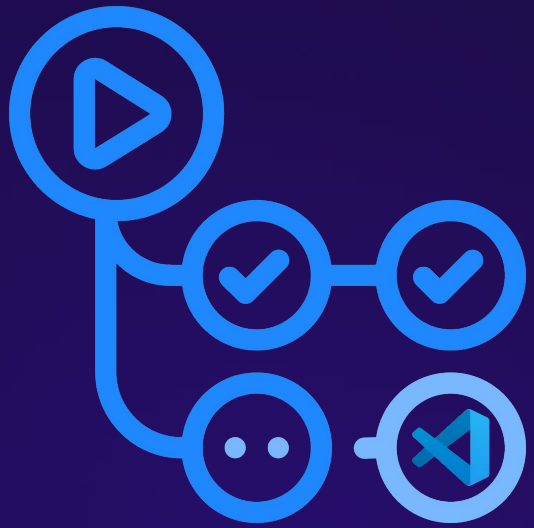
códigofacilito

Hacer uso de matrix

Ejecución paralela automática.

Testing en múltiples entornos.

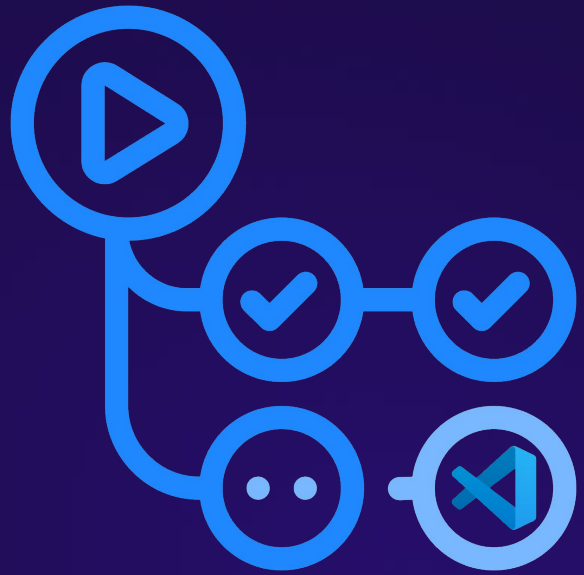
Múltiples combinaciones.



códigofacilito

Utiliza dependabot para actualizar Actions

Habilita dependabot para mantener tus Actions actualizadas:



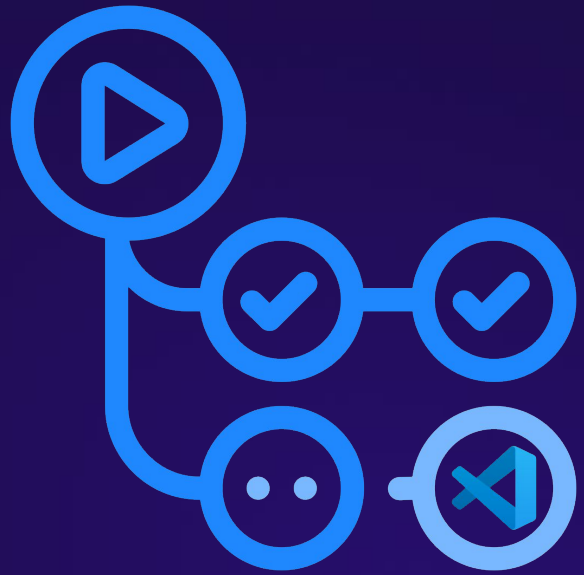
códigofacilito

Hacer uso de concurrencia

Evita ejecuciones innecesarias.

Ahorro de tiempo y recursos.

Prevención de conflictos.

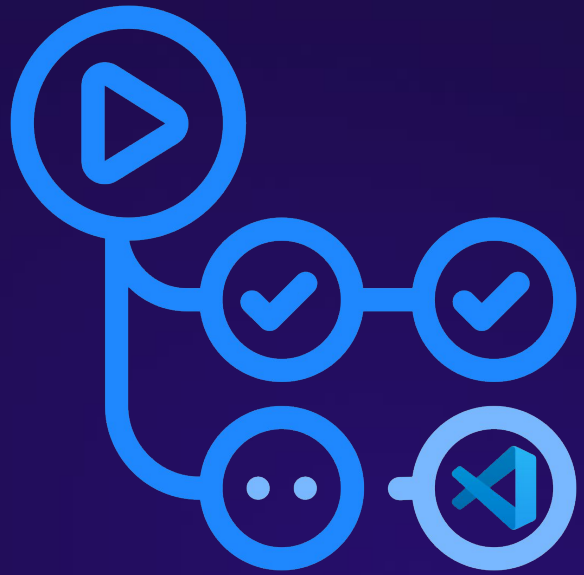


códigofacilito

Hacer uso de servicios

Levantar un contenedor toma unos segundos, mucho más rápido que instalar y configurar servicios manualmente.

Puedes replicar tu entorno de producción fácilmente.



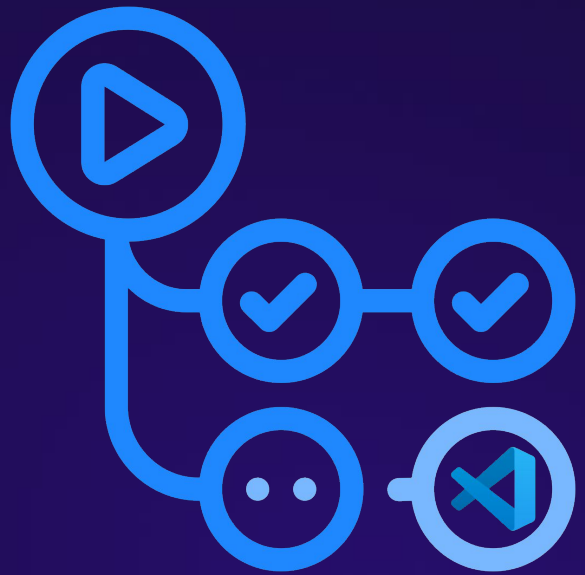
códigofacilito

Usar self hosted runners

Puedes usar máquinas más potentes (más RAM, CPU o SSD) que los runners estándar.

Se eliminan tiempos de espera en colas, especialmente en organizaciones grandes.

Puedes instalar dependencias, herramientas o configuraciones personalizadas sin restricciones.



Ejemplos

códigofacilito

🔗🖼️ Directorio de links

Página: https://codigofacilito.com/

Premium Max : https://codigofacilito.com/max

códigofacilito

Recuerda que

Esta clase está siendo grabada