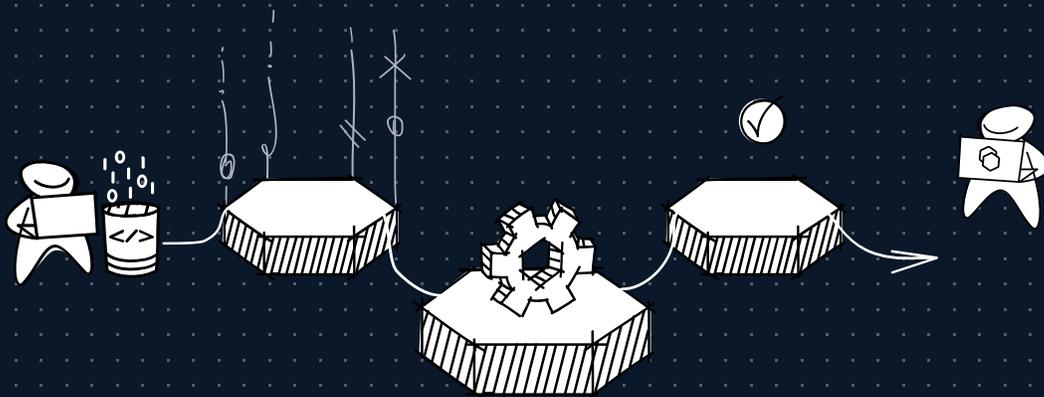




Libro electrónico

# 8

## Patrones de Implementación para transformar su CI/CD





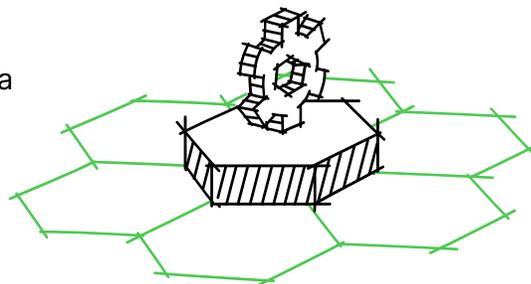
# Índice

<b>Introducción</b>	3
<b>¿Qué Es un Patrón de Implementación?</b>	3
<b>¿Por qué Utilizar Patrones de Implementación?</b>	4
<b>¿Cuál es la Diferencia entre un Buen Patrón de Implementación y uno Excelente?</b>	4
<b>¿Para qué Optimiza sus Implementaciones?</b>	5
<b>Fuerzas Impulsoras de la Selección de Patrones de Implementación</b>	6
<b>8 Patrones de Implementación para Transformar su CI/CD</b>	7
• Céntrico de Resultados	
• Button Push	
• Automatización de Pruebas	
• Multiservicio	
• Entorno de Multiservicio	
• Aprobación a Medias	
• Patrón de Aprobación Completa	
• GitOps	
<b>Conclusión</b>	16

# Introducción

Hoy en día la implementación de patrones CI/CD es fundamental para que sus ideas lleguen a producción de manera rápida y segura. Hay mucho por descifrar en sus patrones, dado que estos replican el ciclo de vida de desarrollo de su software (SDLC) e incluyen opiniones y gobernanza que aportan prudencia a su organización e industria.

En Harness, tenemos el privilegio de ver cientos de organizaciones diferentes que abordan su estrategia de implementación de formas distintas, lo que nos permite destacar algunas tendencias que vale la pena tener en cuenta. En este libro electrónico, compartimos detalles de los patrones de conductos más comunes que observamos, el proceso que se ocupa de determinar cuál podría adaptarse mejor a su organización y la forma en que pueden modernizar sus iniciativas de DevOps.

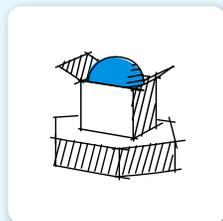


## ¿Qué es un Patrón de Implementación?

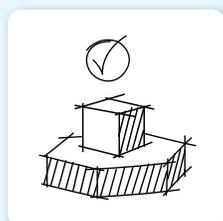
Un patrón de implementación generalmente se compone de uno o más pasos en los cuales el código fuente se convierte en una compilación que se prueba y se implementa en uno o más entornos hasta que termina llegando al usuario final en producción.



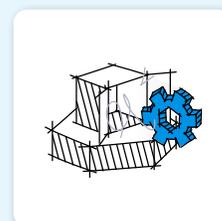
Código fuente



Compilación



Prueba



Implementación

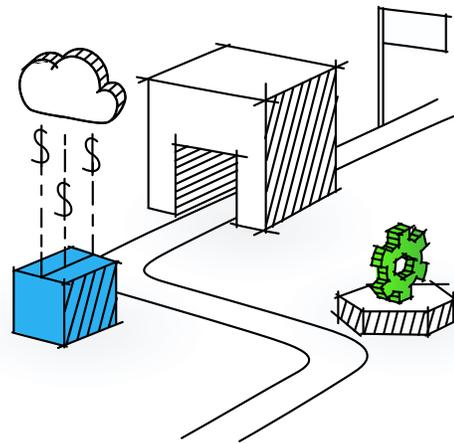
## ¿Por qué adoptar Patrones de Implementación?

A medida que los sistemas se vuelven más complejos, distribuidos y modulares, tenemos la expectativa de que siempre van a funcionar bien. En las entregas de software, contar con una forma segura, rápida y repetible de ingresar sus cambios en la producción es algo esencial.

Por ejemplo, muchas organizaciones usan shell scripts tradicionales para implementar sus aplicaciones. Estos scripts sirven como un conducto simplista para implementar una nueva compilación o versión en un servidor o un entorno objetivo.

Sin embargo, en los ambientes complejos y distribuidos actuales necesitarían ser modificados y ejecutados entre docenas y centenas de veces, además de los cambios comunes que necesitan constante

actualización, como los nombres del servidor, las contraseñas hard-coded o direcciones de IP y datos confidenciales. Eso sin mencionar el retrabajo manual que requeriría conciliar estos cambios.



## ¿Cuál es la Diferencia entre un Buen Patrón de Implementación y uno Excelente?

Básicamente, los buenos patrones de implementación significan conductos rápidos y repetibles mientras que los patrones excelentes significan conductos rápidos, **seguros** y repetibles.

Según el libro “Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations”, los ejecutores de elite tiene un tiempo de ejecución de menos de 1 hora y una tasa de fallas de cambio de menos del 15 % para las implementaciones en producción. Por lo tanto, un patrón excelente se completará en menos de una hora, y capturará el 95 % de las anomalías y regresiones antes de que el código alcance a un usuario final.

Si su código tarda más de una hora en alcanzar la producción, o si fallan más de 2 implementaciones entre 10, es posible que quiera reconsiderar el diseño y la estrategia de su conducto.

La estructura de los patrones de implementación tiende a seguir lo que más impulso le da. En última instancia, la meta de impulso afecta a la estructura de los conductos en sí misma. Por eso, a fin de determinar cuáles estructuras funcionan mejor para su organización en particular, es importante considerar las fuerzas más comunes que ayudan a determinar estas decisiones.

## ¿Para qué Optimiza sus Implementaciones?

Comencemos con la definición de Continuous Delivery de Jez Humble, de la Asociación de Investigación en DevOps (DORA) y coautor del libro Accelerate:

“Continuous Delivery es la capacidad de ingresar cambios de todo tipo (incluidos nuevas funciones, cambios de configuración, arreglos de errores y experimentos) en la producción o en las manos de los usuarios de manera segura y rápida en forma sostenible”.

La manera segura significa que los conductos hacen cumplir con la calidad, las pruebas y un mecanismo para lidiar con los fallos. La manera rápida implica que los conductos promueven

código sin intervención humana, en forma automatizada; y, por último, sostenible significa que los conductos pueden hacer todo esto en forma consistente con poco esfuerzo y recursos.

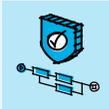
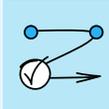
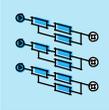
En el mundo real, es raro que un cliente construya conductos seguros, rápidos y sostenibles. Por ejemplo, un conducto que no hace más que recoger un artefacto de algo recobrado y lo implementa en un clúster de Kubernetes puede llevar menos de entre 20 y 30 segundos.

---

¿Pero qué ocurre si ese artefacto tiene errores o afecta a los usuarios finales?

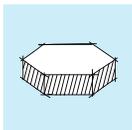
---

### Estas son algunas consideraciones para pensar cuando se diseñan conductos:

	Calidad y cobertura/ automatización de pruebas		Seguridad (p. ej., escaneo de vulnerabilidades)		Configuración del entorno objetivo (p. ej., computación efímera)
	Variables y secretos		Estrategia de liberación (Blue/ Green/Canary)		Aprobaciones
	Auditoría y Compliance		Estrategias de repliegue y falla		Gestión de SLA, SLO, SLI

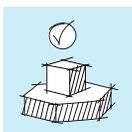
Los conductos pueden ser tan únicos como las organizaciones; pero a medida que nuestra industria avanza hacia la estandarización, comienzan a aparecer patrones.

# Fuerzas Impulsoras de la Selección de Patrones de Implementación



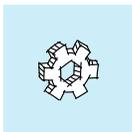
## Entornos

A medida que los sistemas se hacen más distribuidos, la cantidad de ubicaciones en las que tiene que implementarse un servicio aumenta. Si su meta principal consiste en implementarse en múltiples entornos/ubicaciones, los conductos tenderán a ser más céntricos para la implementación, y favorecer la orquestación de todos los entornos que un servicio tiene que atravesar.



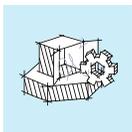
## Pruebas

La orquestación de pruebas es un uso popular de los patrones. Cuando hay que enlazar varias metodologías de prueba diferentes, un lugar natural para la automatización que hace progresar las pruebas es el patrón de implementación. Estos patrones tenderán a tener etapas más largas a medida que la progresión tenga lugar entre los entornos; el rigor de las pruebas aumenta, y es así que es más largo el tiempo por etapa cuanto más se acerca un patrón a la producción.



## Servicios

Con el aumento de los microservicios, las implementaciones tienden a incluir más de un servicio. Si el patrón de implementación se usa para la orquestación de servicios, será necesario implementar varios servicios en forma secuencial o paralela. Estos patrones tienden a utilizarse para la orquestación de varios servicios, lo que asegura la uniformidad entre sus implementaciones.



## Resultado

En última instancia, la función tiene que cumplir con la expectativa. Los patrones que se enfocan en resultados tienden a tener etapas finales más prolongadas y la capacidad de continuar en ejecución cuando se terminó la implementación. Un resultado puede significar que se aseguran los SLA/SLO/SLI y, si se infringen, el patrón es una vía para la restauración/el MTTR. La regresión o un cambio en el resultado pueden ocurrir horas o días después de una implementación.



## Personas

Antes de que hubiera patrones de implementación, las personas se involucraban mucho con las implementaciones progresivas. Si el proceso de implementación sigue siendo mayormente manual y requiere muchas puertas de aprobación (p. ej., manejadas por personas), estos conductos tienden a ser de larga duración y utilizarse como una manera de rastrear el flujo de trabajo humano. Los patrones modernos tratan de evitar esta práctica.

# 8 Patrones de Implementación para Transformar su CI/CD

## Céntrico de Resultados



Puntaje de Patrón

**Agilidad:** Alta**Administración:**  
Media**Complejidad del Patrón:**  
Baja**Riesgo de Falla del Cambio:**  
Bajo

Workloads y la infraestructura modernas permiten que accionar un interruptor para promover sea mucho más fácil. Con una liberación canary, los cambios pueden introducirse de forma segura a lo largo de los entornos y se introducen al cliente de manera segura. Si no está familiarizado con liberación canary, esta introduce cambios como un porcentaje del tráfico o la infraestructura y, en tanto los cambios se aprueban, continúa promoviéndolos hasta que la versión estable quede completamente reemplazada por la del canary. Con el Patrón Céntrico de Resultados, una vez que se cumple con las metas, los cambios están listos para implementarse.

### ¿Qué Tiene de Bueno este Patrón?

Un enfoque muy pragmático para la entrega de cambios en la producción; hay menos énfasis sobre en cuál entorno implementar frente a si tuvieron sentido los cambios y están siendo entregados dentro de los objetivos predefinidos (calidad, seguridad, gobernanza, eficiencia...). Las liberaciones canary, emparejadas con un conjunto de pruebas más estricto o difícil de aprobar, como una prueba de remojo, brindan la confianza de que un cambio cumple con las expectativas.

### ¿Dónde Ve Usted este Patrón?

Este patrón requiere que los equipos tengan una infraestructura madura y una ingeniería de calidad moderna. Ahí donde las aplicaciones son lo básico de una organización, como en el comercio electrónico, los clientes tienden a mostrar este patrón. El dicho de que a sus clientes no les importa cómo lo logró (infraestructura), si no lo que logró (cambios en la aplicación) resplandece con las empresas de eCommerce.

### Oportunidades de Mejora con este Patrón

En función de quién es el cliente interno para este patrón, la fase de pruebas (p. ej., el principal portero de la producción) podría verse como una caja negra para los equipos de desarrollo.



## Button Push



Puntaje de Patrón

**Agilidad:** Baja a Media

**Administración:** Baja

**Complejidad del Patrón:** Baja a Media

**Riesgo de Falla del Cambio:** Baja a Media

Existe una dualidad al embarcarse en un trayecto de conductos: el conducto está completamente automatizado o es completamente manual. El Patrón de Button Push permite la aprobación humana en la línea de llegada en lugar de eliminar el elemento humano por completo. Este método da lugar a una validación más, que atiende razones de cumplimiento, regulaciones o de construcción de confianza.

### ¿Qué Tiene de Bueno este Patrón?

La mezcla de conocimiento humano y de sistema les alivia a los equipos de ingeniería la carga de obtener un candidato a la liberación que está muy cerca de la producción. Luego, en función de los requisitos regulatorios, de cumplimiento u organizacionales, un ser humano puede revisar todos los hechos y hallazgos que se generaron en forma sistemática. Si usted se encuentra con resistencias en su recorrido de automatización de los conductos, el Patrón de Button Push sería un buen término medio, ya que la automatización continúa acumulando confianza.

---

Si usted se encuentra con resistencias en su recorrido de automatización de los conductos, el Patrón de Button Push sería un buen término medio, ya que la automatización continúa acumulando confianza.

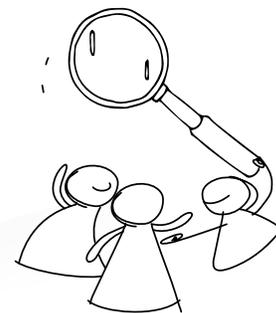
---

### ¿Dónde Ve Usted este Patrón?

Mayormente en organizaciones en las que hay algún tipo de requisito de cumplimiento o regulación que una parte responsable debe firmar. En tanto las organizaciones intenten limitar la cantidad de aprobaciones manuales pero no puedan eliminarlas por completo, el Patrón de Button Push es una buena alternativa.

### Oportunidades de Mejora con este Patrón

Proveer toda la información que se necesita para tomar una decisión y conservarla en un formato auditable constituye un desafío para ciertas plataformas CI/CD que no tienen capacidades listas para usar para los pasos manuales. Por lo general, agregar un paso manual en un proceso automatizado es algo complicado. Una vez que el equipo desarrolla ese patrón, expandir la cantidad de aprobaciones manuales podría ser un obstáculo o un antipatrón que podría inclinarse hacia el Patrón de Aprobación.



## Automatización de Pruebas



Puntaje de Patrón

Agilidad: Media

Administración:  
BajaComplejidad del Patrón:  
MediaRiesgo de Falla del Cambio:  
Bajo

La automatización de pruebas puede cobrar vida propia. Los marcos de prueba modernos imitan el desarrollo de aplicaciones, que tienen necesidades y paquetes de orquestación que se implementan junto con la propia aplicación. Modificar la cobertura de las pruebas para cubrir los cambios de las aplicaciones es una expectativa para las pruebas automatizadas. Y tener un conducto dedicado a la automatización de pruebas puede significar una alternativa prudente.

### ¿Qué Tiene de Bueno este Patrón?

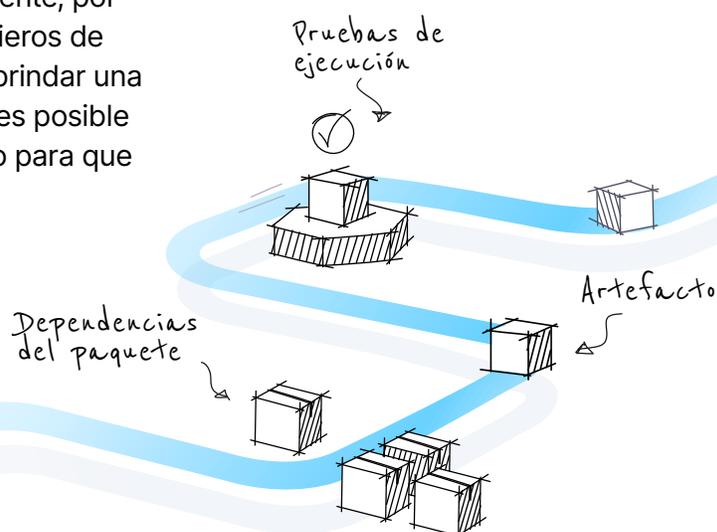
Dado que las organizaciones ponen un mayor énfasis en las pruebas automatizadas, obtener las pruebas correctas requiere de una iteración similar al desarrollo de las aplicaciones. El Patrón de Automatización de Pruebas trata los activos de pruebas/aseguramiento de la calidad como activos de aplicación, y es capaz de implementar y ejecutar conjuntos de pruebas en función de los artefactos de las aplicaciones. El conducto puede ejecutarse en forma independiente; por eso, si los desarrolladores o los ingenieros de aseguramiento de la calidad quieren brindar una retroalimentación temprana, tal cosa es posible con este patrón porque está diseñado para que se ejecute múltiples veces.

### ¿Dónde Ve Usted este Patrón?

Este patrón tiene el potencial de ser una representación de la Ley de Conway. Apreciamos esto con los clientes que construyen sus prácticas de automatización de pruebas o tienden a tener muchas iteraciones en sus conjuntos de pruebas. Varios clientes de diferentes verticales tuvieron variaciones de este patrón donde la automatización de pruebas era orquestada por conductos dedicados.

### Oportunidades de Mejora con este Patrón

Como con cualquier conducto con un propósito específico (si se excluye la vía completa a la producción), la Automatización de Pruebas representa los pasos ideales hacia Continuous Delivery. Según el diseñador del conducto, colocar el aseguramiento de la calidad como un conducto separado para las aplicaciones que atraviesan el patrón puede crear un camino de menor resistencia.



## Multiservicio



Puntaje de Patrón

**Agilidad:** Media a Alta

**Administración:** Media

**Complejidad del Patrón:** Media

**Riesgo de Falla del Cambio:** Baja a Media

Cuando se piensa en un conducto tradicional, viene a la mente la noción de un artefacto por ejecución de un conducto; sin embargo, hay circunstancias válidas donde se necesita implementar más de un artefacto durante un conducto, por ejemplo: al implementar múltiples servicios simultáneamente para un cambio de plataforma o diferentes tipos de artefactos (como artefactos estáticos junto con artefactos de aplicaciones), la meta principal del Patrón de Multiservicio es hacer que los artefactos se implementen.

### ¿Qué Tiene de Bueno este Patrón?

El Multiservicio da lugar a la capacidad de implantar múltiples servicios al mismo tiempo durante una liberación. A medida que los estándares de la empresa se impregnan en la organización y se crean nuevos servicios, un nivel básico de confianza para todos los servicios implementados en el conducto equivale a una liberación más segura. En tanto las aplicaciones y los servicios se vuelven más granulares, agregar servicios adicionales al conducto es algo estándar para el diseño del patrón.

---

A medida que los estándares de la empresa se impregnan en la organización y se crean nuevos servicios, un nivel básico de confianza para todos los servicios implementados en el conducto equivale a una liberación más segura.

---

### ¿Dónde Ve Usted este Patrón?

En organizaciones que están construyendo plataformas; p. ej., tienen más de un artefacto al mismo tiempo para un release. Las interpretaciones anteriores del patrón ejecutarían el mismo conducto una y otra vez, y solo cambiarían el artefacto o el destino para la implementación.

### Oportunidades de Mejora con este Patrón

La orquestación de múltiples implementaciones de servicio puede ser algo complejo, especialmente durante un escenario de fallas parciales. Cada servicio debería ser independiente del otro, y no estar fuertemente acoplados.

Sin embargo, si un servicio dependiera de un resultado de otro servicio, podría haber una demora en la implementación de una nueva función, ya que el conducto necesita poder tener lógica o el sistema subyacente necesita determinar qué ha fallado frente a qué ha servido en el pasado.

## Entorno de Multiservicio

 Puntaje de Patrón

**Agilidad:** Media

**Administración:**  
Media a Alta

**Complejidad del Patrón:**  
Alta

**Riesgo de Falla del Cambio:**  
Bajo

Cuanto más distribuidas crezcan sus aplicaciones, más artefactos y entornos tendrá para orquestar. Si se tienen en cuenta estas dos inquietudes, el Patrón de Entorno de Multiservicio puede ayudar a impulsar cambios entre múltiples entornos con múltiples artefactos.

### ¿Qué Tiene de Bueno este Patrón?

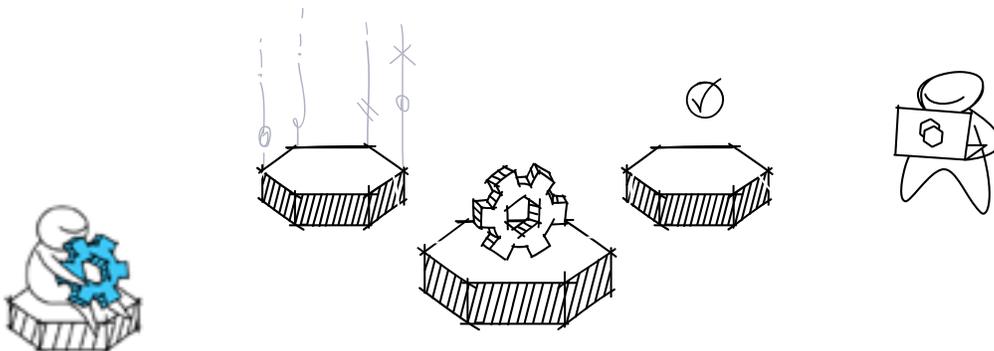
Lo que se había instalado antes de este patrón podría ser una de las orquestaciones más complejas que garantizan la paridad entre los entornos y los múltiples artefactos que tienen que implementarse. Este patrón surgió de la necesidad de automatizar un escenario de implementación complejo. No todos los artefactos y servicios se crean iguales. La interacción con diferentes proveedores de infraestructura de aplicaciones, como una caché o una base de datos, requiere de una estrategia diferente para implementar y agregar a la cantidad de diversos entornos/infraestructura. El Patrón de Entorno de Multiservicio aborda estas dos inquietudes.

### ¿Dónde Ve Usted este Patrón?

Este patrón es común entre las organizaciones que tienen topologías complejas o lidian con la carga significativa que justifica este tipo de topologías complejas. Por ejemplo, los proveedores de comercio electrónico y los de viajes y entretenimiento que lidian con el tráfico a escala de red constantemente.

### Oportunidades de Mejora con este Patrón

Con demasiada frecuencia este patrón se desglosa en conductos diferentes que abarcan la pre y luego la producción. Cuando potencialmente se cruzan múltiples tipos de capacidades en un conducto (p. ej., cambios de aplicación y de caché), el efecto pleno de todos los cambios necesita ser aprobado. Por diseño, esto podría ser una prueba de remojo, o lo que se observó fueron conductos específicos de la producción una vez que se consideró que los entornos inferiores eran apropiados.



## Aprobación a Medias

 Puntaje de Patrón

**Agilidad:** Baja a Media

**Administración:** Media a Alta

**Complejidad del Patrón:** Media

**Riesgo de Falla del Cambio:** Bajo

A veces es difícil quitar por completo el elemento humano de su conducto. Con el Patrón de Aprobación a Medias, se da una buena mezcla de automatización y aprobación humana (cuando resulta necesario). Se representa una combinación de enfoques nuevos con tradicionales.

### ¿Qué Tiene de Bueno este Patrón?

Para las organizaciones sujetas a obligaciones reglamentarias/de cumplimiento o más sensibles a los riesgos, este es un paso hacia la automatización total. Incluso automatizar las piezas de los pasos promocionales podría representar un cambio fundamental en la organización. Al ser pragmático sobre por dónde encarar primero la automatización, las promociones/aprobaciones automatizadas tendrán lugar en los entornos preproductivos. Este patrón puede ser un catalizador del cambio.

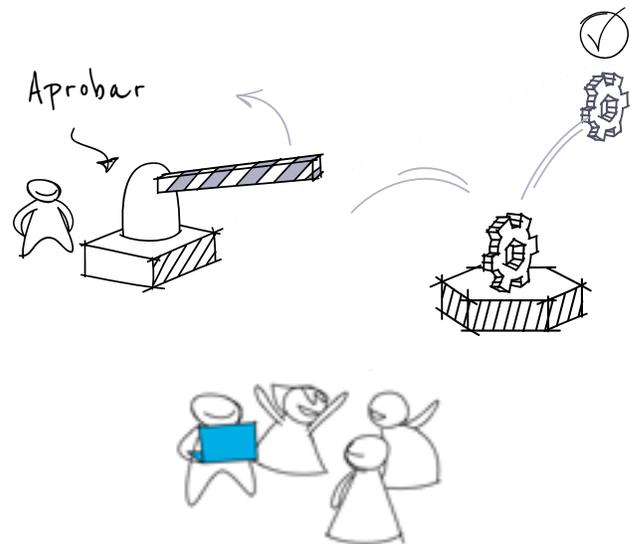
Incluso automatizar las piezas de los pasos de promoción entre ambientes podría representar un cambio fundamental en la organización.

### ¿Dónde Ve Usted este Patrón?

En las empresas de servicios financieros y las industrias reguladas, estos patrones están para ocuparse del impulso para la innovación y la transformación digital. Las organizaciones que también tienen cambios de alto riesgo pueden solo considerar un alcance menor de servicios para implementar ese patrón.

### Oportunidades de Mejora con este Patrón

Si hay demasiadas puertas de aprobación o si hay un retraso en una puerta de aprobación, se formará un cuello de botella. El Patrón de Aprobación Completa toma el Patrón de Aprobación a Medias y agrega una aprobación manual a cada etapa. Como patrón más evolutivo, se harán mejoras a medida que continúe la confianza en el conducto y la necesidad de aprobaciones manuales disminuya.



## Patrón de Aprobación Completa

 Puntaje de Patrón

**Agilidad:** Baja

**Administración:** Alta

**Complejidad del Patrón:**  
Media

**Riesgo de Falla del Cambio:**  
Bajo

¿Recuerda cuando jugaba al teléfono descompuesto de niño? Usted pasaba un mensaje secreto de una persona a otra y, al final de la línea, la frase estaría influenciada por todas las personas por las que pasaba. Eso podría ser un patrón en su conducto. Con el Patrón de Aprobación Completa, la mayoría (si no todos) los pasos de promoción requiere de una aprobación manual.

### ¿Qué Tiene de Bueno este Patrón?

Para ciertas organizaciones que son conservadoras y más sensibles a los riesgos, el Patrón de Aprobación Completa imita al SDLC actual. La resistencia a adoptar este patrón sería baja porque lo más probable es que estuviera ocurriendo antes de un conducto en pasos

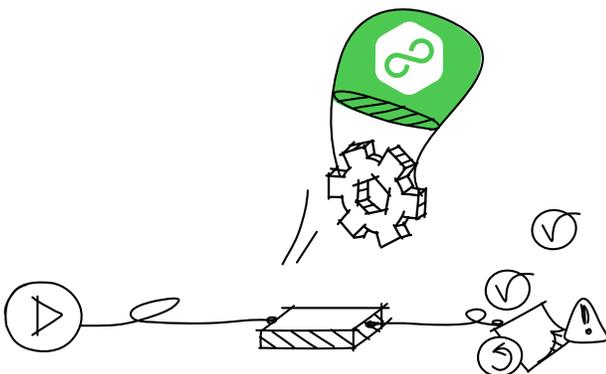
desarticulados en varias plataformas. Contar con este patrón ayudaría a ver dónde ocurren los cuellos de botella y, si ocurriera algún tipo de evolución futura, las áreas en las cuales podrían hacerse mejoras se harían muy evidentes.

### ¿Dónde Ve Usted este Patrón?

Con mayor frecuencia este patrón respalda industrias altamente reguladas como las de infraestructura crítica, inteligencia/ejército y organizaciones de finanzas y atención de la salud. Cuando existe la necesidad de unir fuerzas laborales dispares, como contrataciones múltiples u organizaciones de consultoría para entregar un proyecto, la automatización de la confianza podría no estar disponible de inmediato hasta contar con futuras interpretaciones del proyecto.

### Oportunidades de Mejora con este Patrón

Con mucha frecuencia los pasos de aprobación manual son signos de pruebas manuales. Especialmente si esta es su primera incursión en llevar un elemento hasta la producción, la automatización llegará como aprendizajes de todos los pasos que la implementación de una producción tiene hasta que sale. Este rehospedaje de su proceso manual existente, que un flujo de trabajo o un conducto rastrean, es el primer paso a la automatización. Una vez que se completan todos los pasos, el conducto queda completo. La propiedad de lo que ocurre después de la implementación sigue la estructura organizacional.



## GitOps

 Puntaje de Patrón

**Agilidad:** Alta

**Administración:**  
Media

**Complejidad del Patrón:**  
Media

**Riesgo de Falla del Cambio:**  
Media

En un modelo GitOps, la configuración deseada del sistema se almacena en un sistema de control de versión, como el Git. En lugar de realizar cambios al sistema por medio de un UI o CLI, un ingeniero hace cambios a los archivos de configuración que representan el estado deseado usando Git como una única fuente de verdad. Una diferencia entre el estado deseado guardado en Git y el estado actual del sistema indica que no todos los cambios se han implementado. Estos cambios pueden revisarse y aprobarse por medio de mecanismos de control de versión estándar como pull requests, revisiones de código y uniones al maestro. Cuando los cambios se aprobaron y unieron a la ramificación principal, un proceso de software de operador es responsable de cambiar el estado actual del sistema según la configuración almacenada en Git.

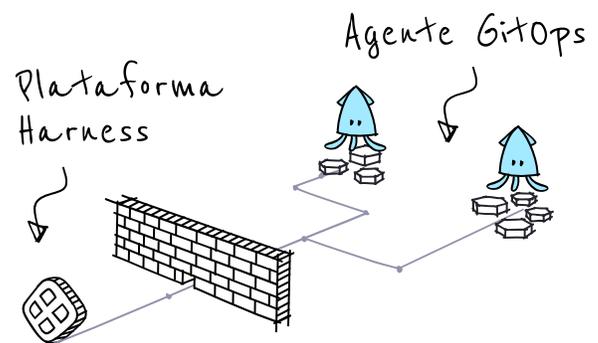
GitOps es el patrón más veloz para que los desarrolladores implementen cambios de código en los entornos.

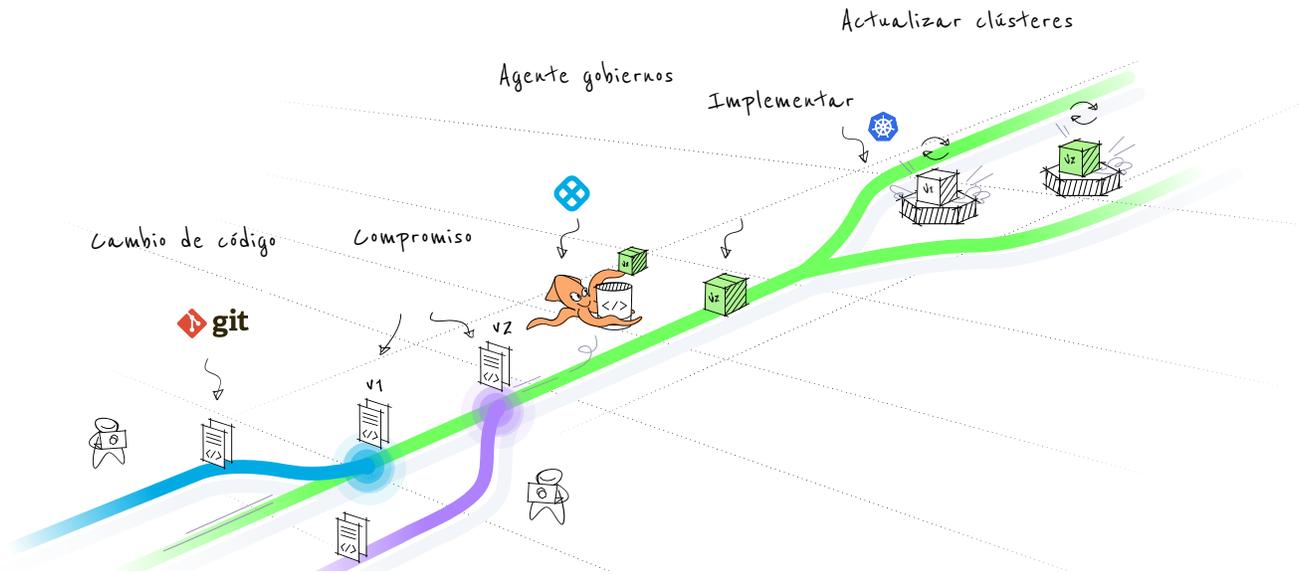
### ¿Qué Tiene de Bueno este Patrón?

GitOps es el patrón más veloz para que los desarrolladores implementen cambios de código en los entornos. A los desarrolladores les gusta GitOps porque les da la libertad de implementar software sin tener que

hacer interfaz con un UI CD tradicional y pueden evitar conductos de implementación extremadamente burocráticos. La mayoría de los desarrolladores conoce el Git, configuración como código, y la infraestructura como código, lo que significa que no se espera que aprendan un sistema de CD completamente nuevo.

A los desarrolladores les gusta GitOps porque les da la libertad de implementar software sin tener que hacer interfaz con un UI CD tradicional y pueden evitar conductos de implementación extremadamente burocráticos.





### ¿Dónde Ve Usted este Patrón?

Este patrón es ideal para entornos no de producción, con altas tasas de cambio y bajas probabilidades de falla, si bien estas limitaciones no son estrictas. GitOps se usa a menudo para probar entornos o para aplicaciones que no son críticas para los negocios.

### Oportunidades de Mejora con este Patrón

GitOps abarca un subconjunto del ciclo de vida del software que se enfoca principalmente en la automatización de la infraestructura. Esto es importante porque las herramientas de GitOps a veces se promocionan como la solución universal que resolverá todos los problemas de liberación, y esto simplemente no es verdad. Antes que nada, GitOps requiere que sus artefactos de implementación ya estén instalados. Esto quiere decir que las tareas del tipo “compilar código, ejecutar pruebas de unidad/integración, escanear seguridad y análisis estadístico...” no son un tema de las herramientas de GitOps y se supone que ya estén ahí.

GitOps tampoco se ocupa de la promoción de liberaciones entre entornos. GitOps usa un modelo de aprobación de pull request, en el que los desarrolladores realizan cambios, crean PR y luego un aprobador puede aceptar el PR. Generalmente esto funciona muy bien hasta que tenemos que incluir aprobaciones de negocio. El personal del negocio puede no estar familiarizado con el Git o los PR. Si un equipo necesita aprobaciones de negocio, tiene que construir un proceso en GitOps. Y esto aumenta drásticamente el plazo de ejecución. GitOps depende de un PR, modelo basado en aprobación, en el cual el aprobador del PR es responsable de cualquier revisión. Una vez que se aprueba un PR, no hay manera de hacer cumplir las reglas de la política de la empresa. Las empresas necesitan formas de controlar lo que pueden hacer los usuarios finales en el clúster, y formas de garantizar que los clústeres cumplen con las políticas de la empresa. Estas políticas pueden figurar para cumplir con requisitos gubernamentales y legales, o para hacer cumplir las mejores prácticas y las convenciones organizacionales.

# Conclusión

En Harness tenemos la oportunidad de ver cómo cientos de organizaciones implementan conductos de entrega de software. Si se trascienden las organizaciones, los patrones que figuran en este libro electrónico sirven como enfoques para llevar ideas a la producción y continuar evolucionando, equilibrando la agilidad y las restricciones reglamentarias. Ningún patrón puede regirlos a todos, ya que cada organización e industria vertical que representa son diferentes. Como los conductos unen capacidades dispares y las aplicaciones son la culminación del trabajo de muchos equipos, los conductos pueden ser altamente únicos y tener características diferentes para cada organización.

Los equipos y las organizaciones pueden tener variaciones o agregados de múltiples patrones. Con la sintonía fina de los procesos respecto de las aprobaciones manuales y la automatización plena, estos patrones respaldan diferentes niveles de agilidad, confianza y sesgo. Si bien su meta específica podría no coincidir al 100 % con uno de los patrones que hemos mencionado, estos modelos pueden actuar como una base para el desarrollo de conductos en su organización.





## **Acerca de Harness**

Harness, the Modern Software Delivery Platform™, brinda una forma simple, segura y protegida para que los ingenieros y los equipos de DevOps liberen con rapidez aplicaciones a la producción. Harness utiliza el machine learning para detectar la calidad de las implementaciones y hacer retroceder automáticamente las fallidas, lo que ahorra tiempo y reduce la necesidad de scripts personalizados y supervisión manual; con esto se les devuelve a los ingenieros los fines de semana. Harness Inc. tiene su base en San Francisco.

**Conozca más en [harness.io](https://harness.io).**