

Beneficios de la Infraestructura como código

DigitalHouse >
Coding School



**Certified Tech
Developer**
The Ultimate Degree

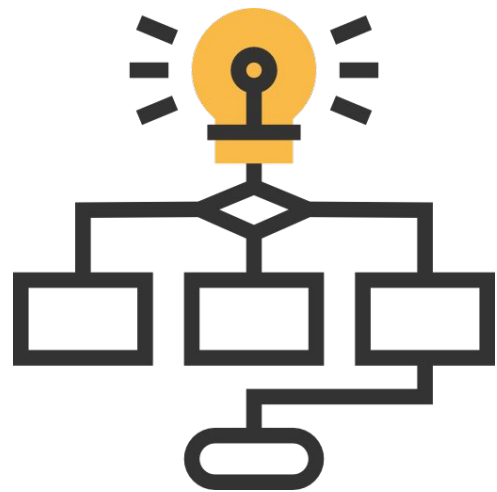


A continuación, les compartimos diez beneficios de la **infraestructura como código**.



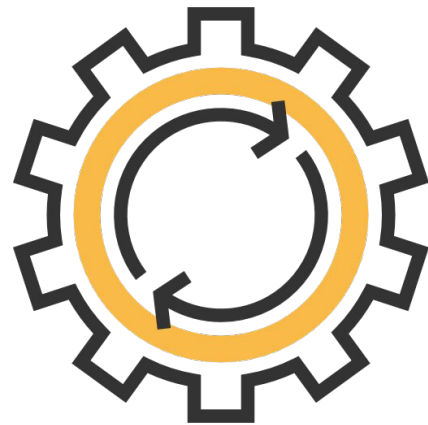
1. Reducción del **error humano**

Minimizamos el riesgo de equivocarnos cuando seguimos una serie de pasos. Mediante procedimientos claros y ordenados podemos evitar guardar una mala configuración o borrar algo que no debíamos. Esto va a aumentar la confianza que tengan en la infraestructura que brindemos.



2. Repetibilidad y predictibilidad

Cuando sabemos que el contexto de nuestra aplicación funciona, vamos a poder repetir la cantidad de pasos que sean necesarios y ser capaces de predecir el resultado, ya que siempre será el mismo. Esto nos da —como resultado— una infraestructura más testeable y estable.



3. Tiempos y reducción de desperdicios

El encargado de ejecutar nuestra infraestructura va a poder hacerlo en cuestión de minutos y sin necesidad de instalar algún componente extra. ¡Siempre vamos a poder activar el proceso para que haga lo suyo!





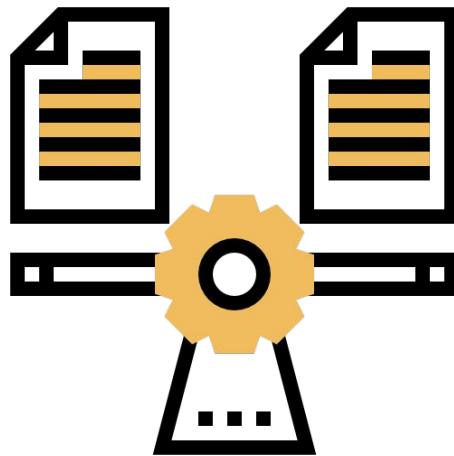
Al ejecutar las tareas más rápido, vamos a poder ayudar a otros equipos a que también trabajen en menos tiempo y de manera más eficiente.



4. Control de versiones

Nuestra infraestructura se va a encontrar definida en archivos, por lo que vamos a poder versionar —al igual que el código fuente de nuestra aplicación— en **templates** o plantillas.

¡Saquémosle el jugo a los templates! Podemos utilizar parámetros para escribir nuestro código de la manera más genérica posible. Luego, al ejecutarlo, vamos a poder enviarle datos distintos en forma de parámetros para que nuestro código los reciba.



5. Reducción de **costos**

Al automatizar procesos, podemos enfocarnos en otras tareas y mejorar lo ya hecho. Esto aporta a la flexibilidad de los equipos de infraestructura para abarcar más tareas.



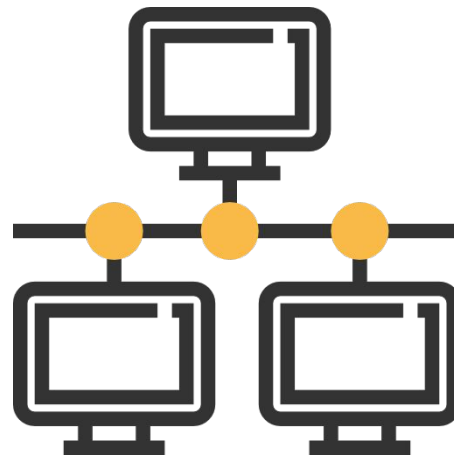
6. Testeos

La infraestructura como código permite que los equipos de infraestructura puedan realizar **pruebas de las aplicaciones** en cualquier entorno (incluso producción) al principio del ciclo de desarrollo.



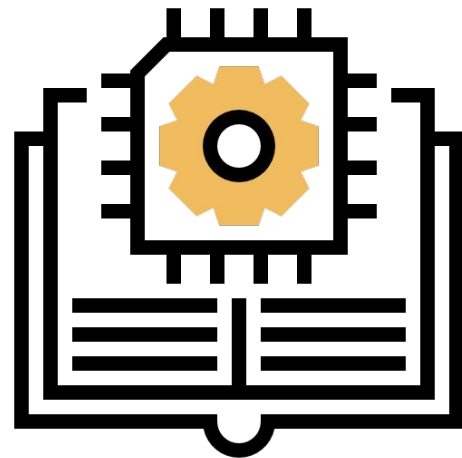
7. Entornos **estables y escalables**

Al evitar configuraciones manuales, la falta de dependencias y al obtener el estado final de infraestructura que necesitamos para nuestras aplicaciones, vamos a ofrecer **entornos estables y escalables**.



8. Estandarización de la configuración

Estandarizar las configuraciones y el despliegue de la infraestructura nos permite evitar cualquier problema de incompatibilidad con nuestra infraestructura y que las aplicaciones se ejecuten con el **mejor rendimiento posible**.



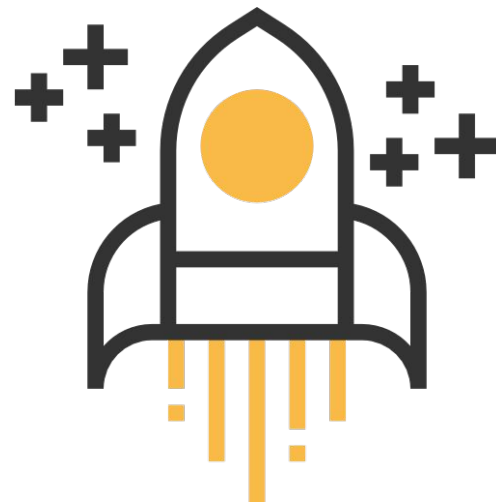
9. Documentación

Al aportar a la documentación de los procesos internos de nuestros equipos vamos a mejorar tiempos y costos. Como ya vimos, podemos **versionar nuestras automatizaciones**. Esta característica nos permite que cada cambio se encuentre documentado, registrado por usuario y con una vuelta atrás (rollback) rápida si encontramos errores en los despliegues, al igual que el código fuente.



10. Más rapidez sin descuidar la **seguridad**

Al momento de mejorar nuestra infraestructura, nunca hay que dejar de pensar en la seguridad que la compone. Al momento de estandarizar la ejecución de la infraestructura, también podemos **estandarizar los grupos de seguridad** con los permisos mínimos, pero necesarios para que todos los equipos puedan trabajar y evitar tareas manuales por parte de los equipos de seguridad.



DigitalHouse>
Coding School