



Twelve-Factor App

Aplicaciones que escalan y se mantienen

Pepe Fabra Valverde | Septiembre 2024

Disclaimer

- La sesión se grabará
- Se compartirán las diapositivas
- Pregunta cuando quieras



Adam Wiggins

Autor de Twelve-Factor App



Contexto

Contexto

Qué buscamos

- Mejorar la robustez y simpleza de nuestra app
- Cómo nos impacta lo contrario
- Nuestros micros no escalan
- Aplicaciones más fáciles de desplegar

Contexto

Cómo nos impacta no tenerlo

- Cada nuevo desarrollo es más complejo
- No tenemos observabilidad
- La lógica no es consistente entre micros
- Cada despliegue es una locura
 - Y en más de una ocasión con mucho esfuerzo manual

Doce reglas

Una guía práctica y consistente para fáciles de desplegar y que escalan

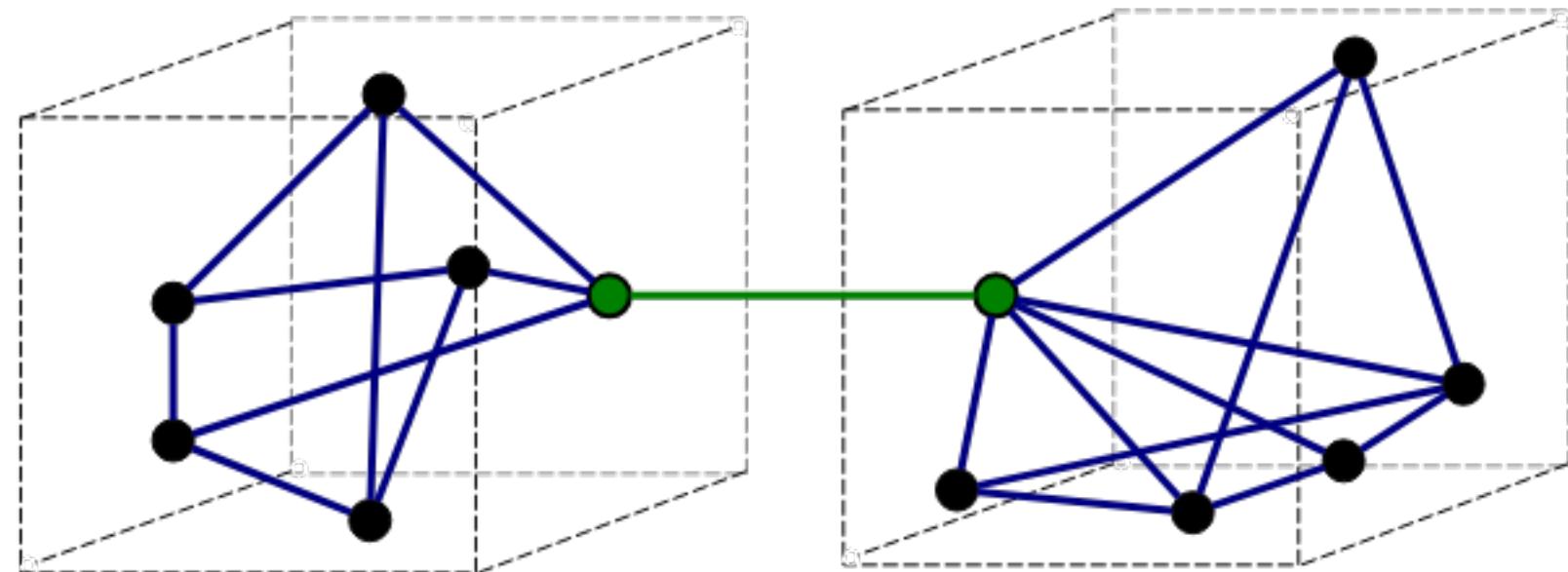
I. Codebase

Los entornos usan diferentes versiones de la misma codebase gestionado por un VCS

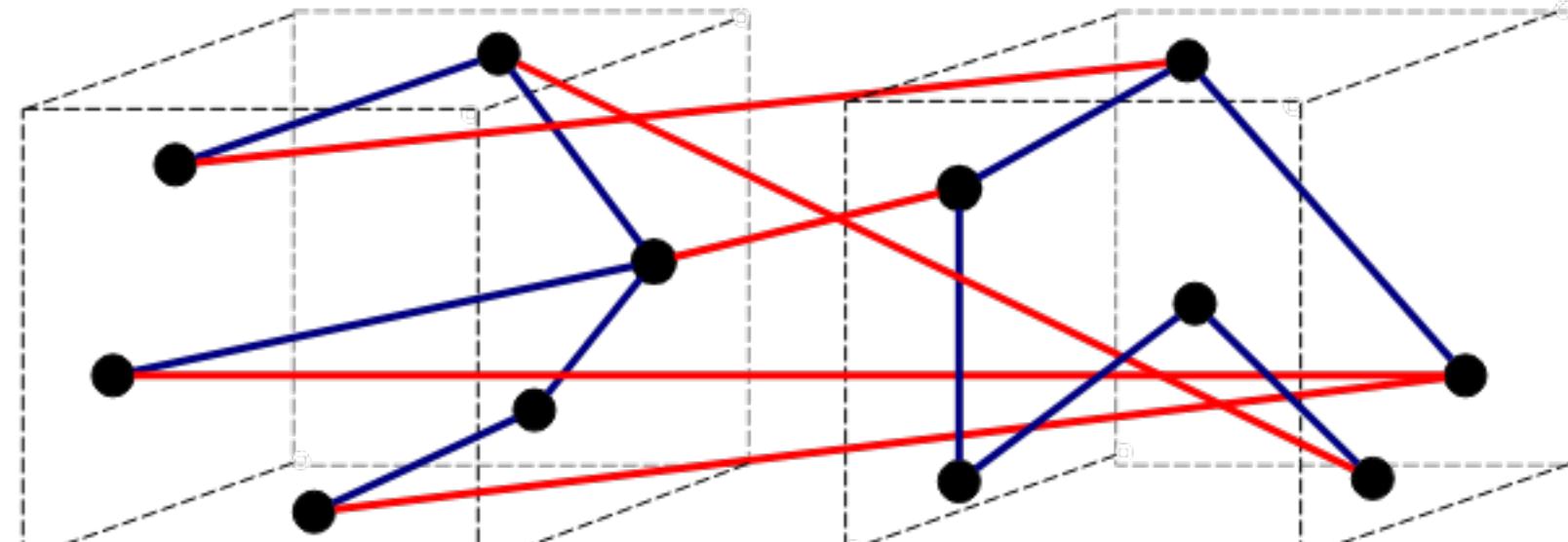
III. Dependencies

Separa y declara todas tus dependencias

**Low coupling
High cohesion**



a) Good (loose coupling, high cohesion)



b) Bad (high coupling, low cohesion)

III. Config

**La configuración que cambia entre entornos
no se almacena en el código, sino en .env**

Config-as-Code

Config as Code

Diferencia entre configuración de entorno y de aplicación

Tipos de configuración de una aplicación:

- Entorno
- Aplicación/Sistema

La configuración que **nunca** cambiará entre entorno **sí** que tiene lugar en nuestro código

IV. Backing services

**Trata los servicios externos como servicios
intercambiables, no te acoples de más**

No hay diferencia entre local y servicios de terceros

V. Build, release, run

Build, run y release son pasos separados

La release es inmutable, cambio nuevo release nueva

VI. Processes

**Los procesos son stateless, todo contexto
necesario no se comparte, se persiste**

Arrancar la app es ejecutar un único script

VIII. Port binding

**La aplicación no se acopla a puertos, porque
son configurables y cambiarán**

VIII. Concurrency

**La carga de trabajo se gestiona siguiendo el
modelo de procesos de UNIX**

Cada tarea es un proceso que no comparte contexto con otros procesos

IX. Disposability

**Arranque ágil, apagado seguro y robusto
ante fallos del sistema**

X. Dev/prod parity

**Mínima diferencia entre los entornos de DEV
y PROD**

XI. Logs

Se envían al stdout y la aplicación no se preocupa de cómo se van a gestionar

Sout en Java, console.log en Node, print en Python, etc.

XIII. Admin processes

Los scripts de administración y gestión del sistema funcionan en un único proceso

Twelve-Factor App

12factor.net

Pepe Fabra Valverde | Septiembre 2024

Gracias

Por vuestra atención

Preguntas