

INFRASTRUCTURE AS CODE

Trabajo Práctico Especial

Ingeniería de Software II

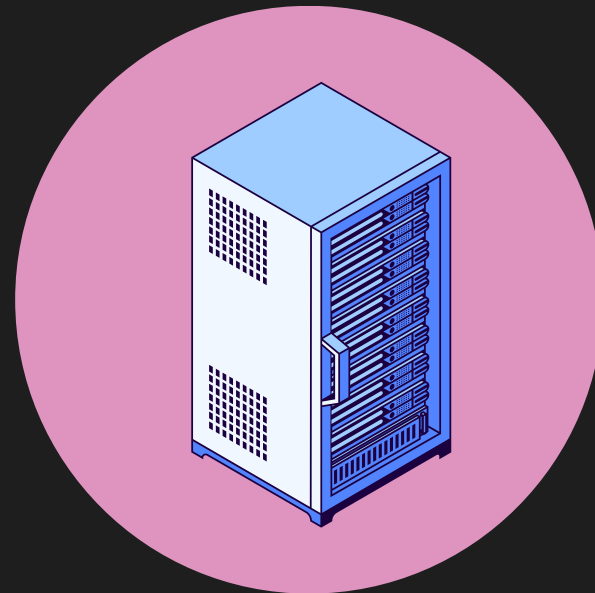
De Simone, Franco - 61100

Dizenhaus, Manuel - 61101

Caeiro, Alejo - 60692

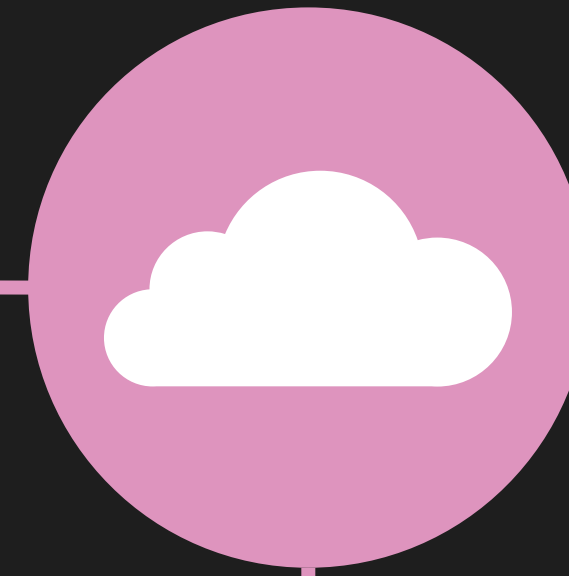
Catino, Kevin - 61643

¿Por qué nace IaC?



Iron age
90s

- Dev y Ops conceptos separados
- Gestión manual de hardware
- Problemas de agilidad



Cloud age
2010

- Mudanza hacia la nube
- Filosofía DevOps
- Mejora CI/CD



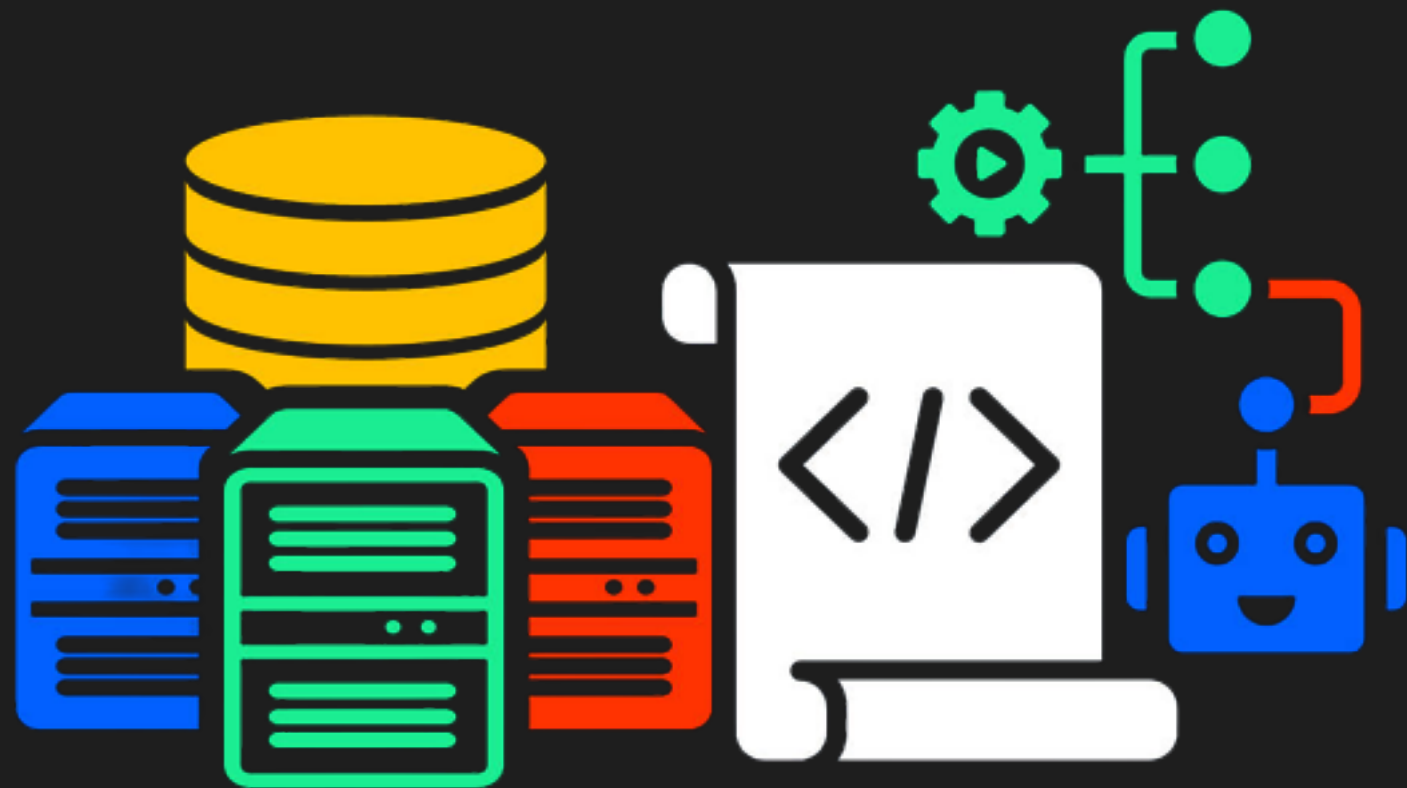
¿Qué es Infrastructure as Code?

"The process of provisioning and managing infrastructure defined through code, instead of doing so with a manual process."



Ventajas de IaC

Why learn a bunch of new languages and tools and encumber yourself with yet more code to manage?



01 SELF-SERVICE

02 SPEED AND SAFETY

03 VALIDATION

04 REUSE

05 DOCUMENTATION & VERSION CONTROL

Declarativo vs Imperativo

DECLARATIVO

- El servicio debe comparar el estado actual de la arquitectura con el código.
- No se definen los pasos o dependencias entre recursos.
- Se define el estado deseado del sistema.
- Es idempotente.

IMPERATIVO

- Se definen las instrucciones y el orden en el que se deben llevar a cabo para lograr la infraestructura deseada.
- Difícil de reutilizar
- No idempotente.

Algunos stacks



Terraform

- Herramienta de código abierto desarrollada por HashiCorp.
- Es declarativo.
- Soporta varios proveedores de nube, como AWS, Azure, Google.
- Utiliza archivos de configuración en formato HCL.



AWS CloudFormation

- Servicio de AWS.
- Es declarativo.
- Permite definir y desplegar recursos en la nube utilizando JSON o YAML.



CHEF™

- Herramienta de código abierto.
- Es mayormente imperativo.
- Utilizado para manejo de configuración de infraestructura.
- Platform-agnostic



Demostración práctica



**MUCHAS
GRACIAS**