



# Frontend DevOps

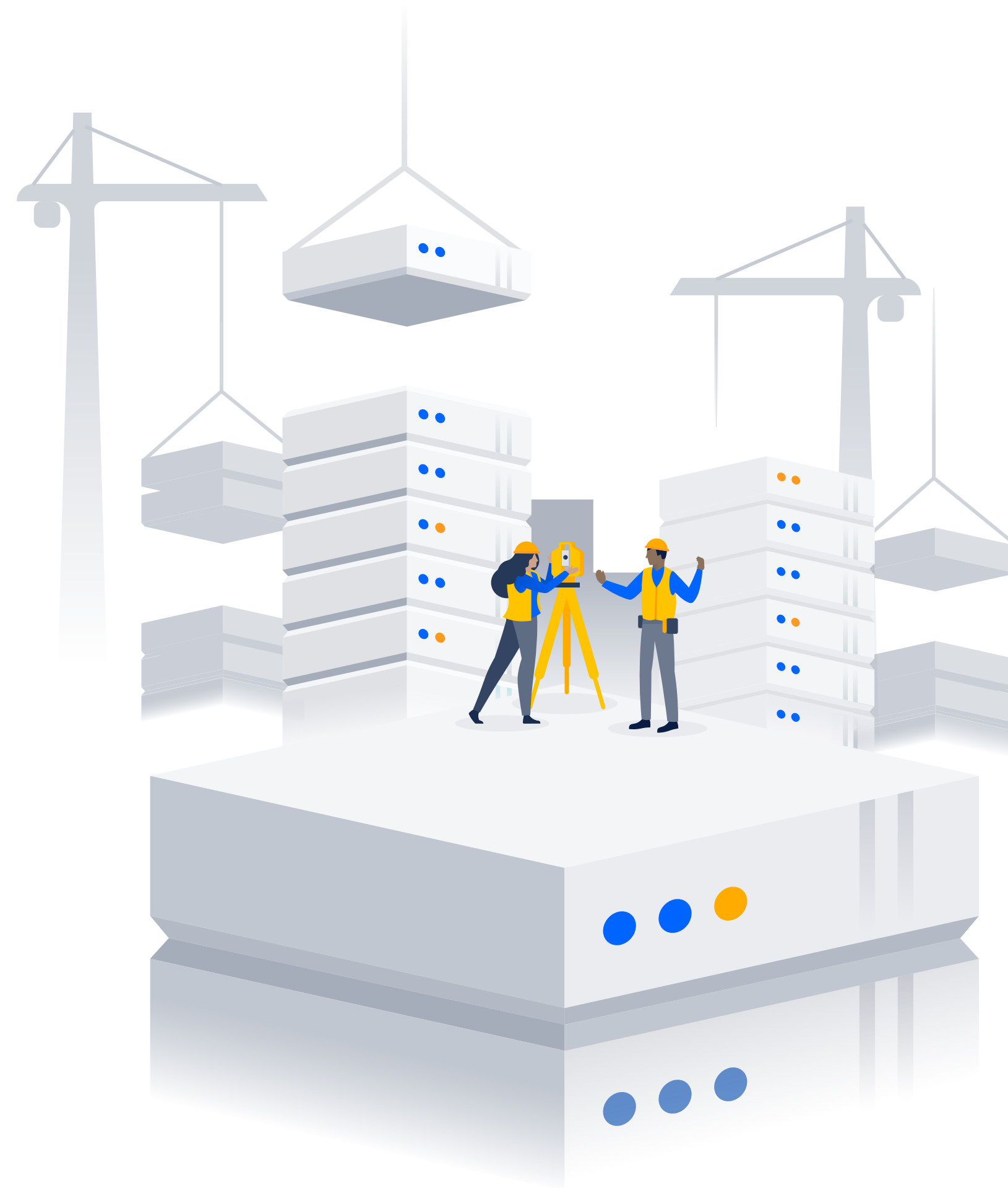
How Jira does Frontend at scale



**SERGIO | PRINCIPAL DEVELOPER | ATlassian | @SCINOS**



# ESCALA DEL PROYECTO



# Tamaño del proyecto



**1,5 millones**

Lineas de código



**80.000**

Tests



**320**

Autores



**600mb**

Código compilado

# Jira Frontend Platform



Mision: acelerar equipos de features



Alexey



Daniel



Jamie



Marcin



Nadia



Sergio

# FRONTEND DEVOPS



**Code**



**Test**



**PR**



**Build**



**Deploy**



**Release**



**Monitor**



## Guia de estilo

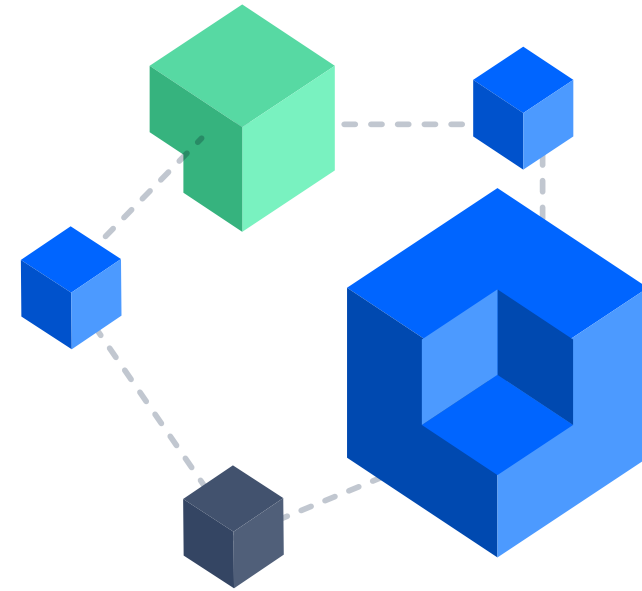
🍊 Tangerine

## Herramientas

- ESLint
- Stricter

Code  
↓  
Test  
↓  
PR  
↓  
Build  
↓  
Deploy  
↓  
Release  
↓  
Monitor

## Desarrollo exploratorio



```
yarn dev start
```

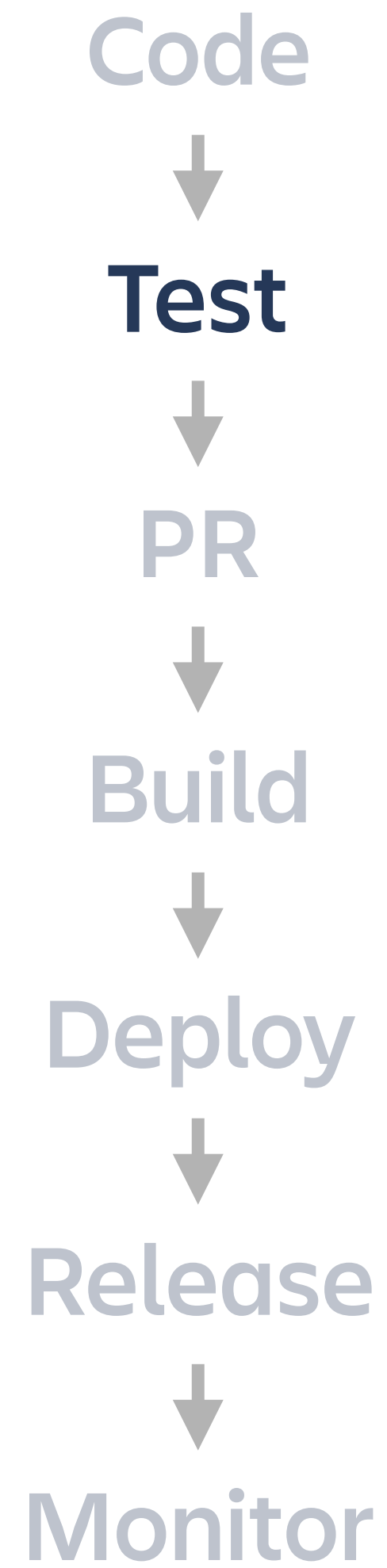
Carga frontend local en  
cualquier instancia  
interna de Jira

## Desarrollo en banco de trabajo



```
yarn storybook start
```

Carga  React Storybook



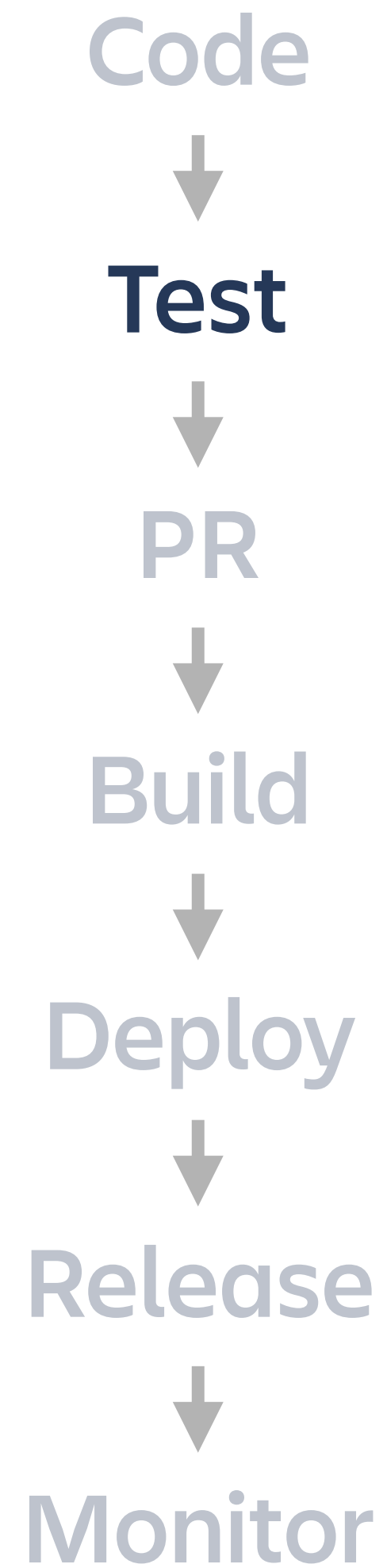
## Tests

- Unit: Jest
- Integration: Cypress

## CI tiene que ser rápido

- Testea solo los cambios
- Reusa caches





## Objetivo

Si los tests pasan en la rama, pasaran en master

## Técnicas

Mergear ramas de una en una

✗ No escala

Mergear master en rama antes de ejecutar tests

✓ Puede valer

Code



Test



PR



Build



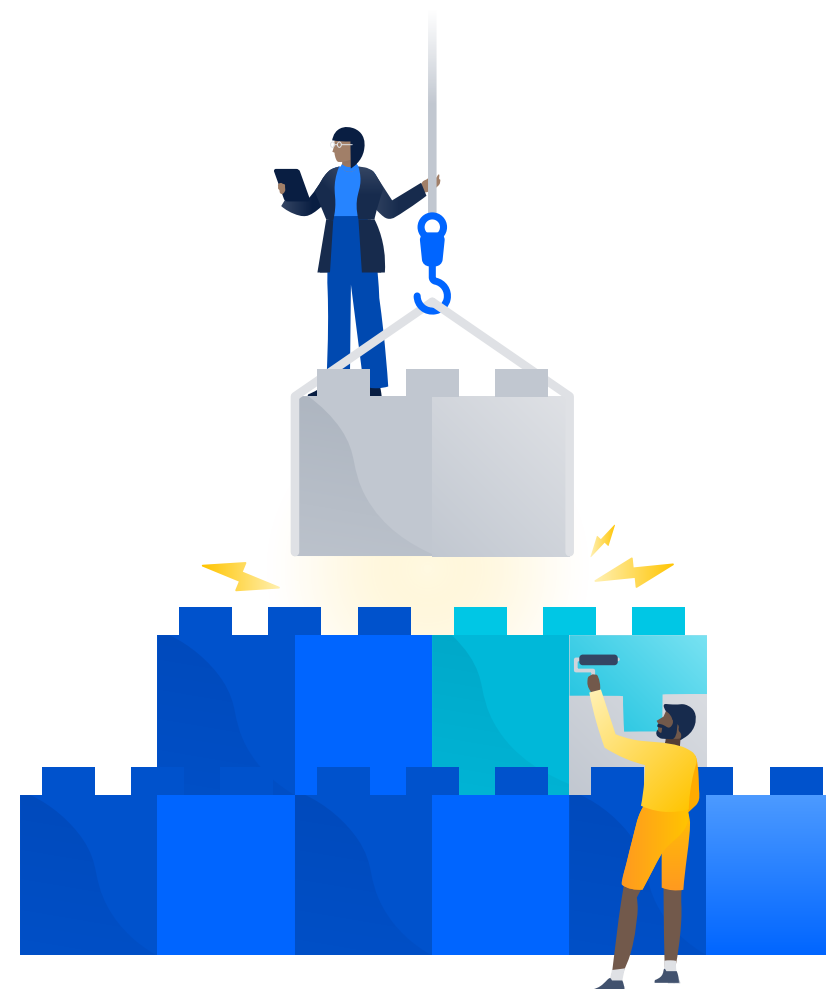
Deploy



Release



Monitor

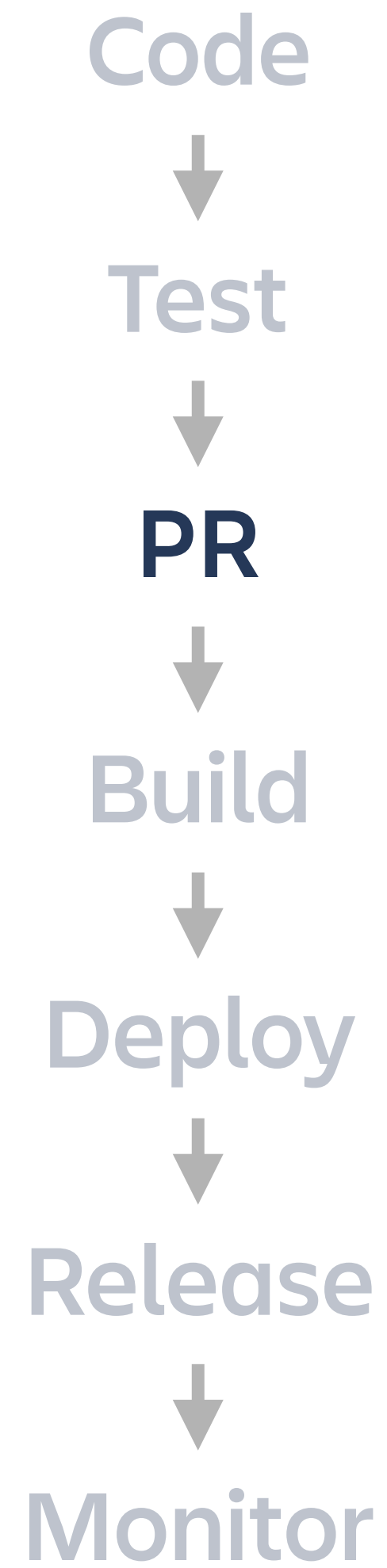


## Revisores por componente

Declarados en `manifest.json`

## Despliegue de rama

Crea version pública con la rama

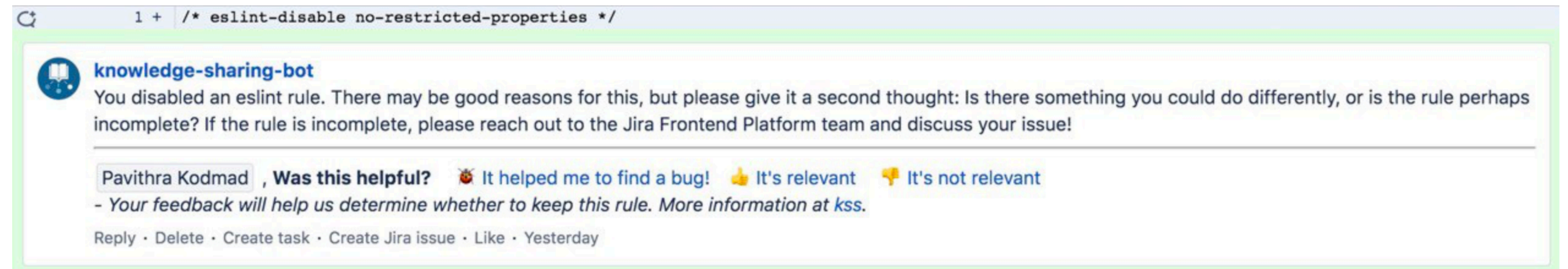


## Anotaciones automáticas en la PR

Archivos muy usados



Patrones comunes



Code



Test



PR



**Build**



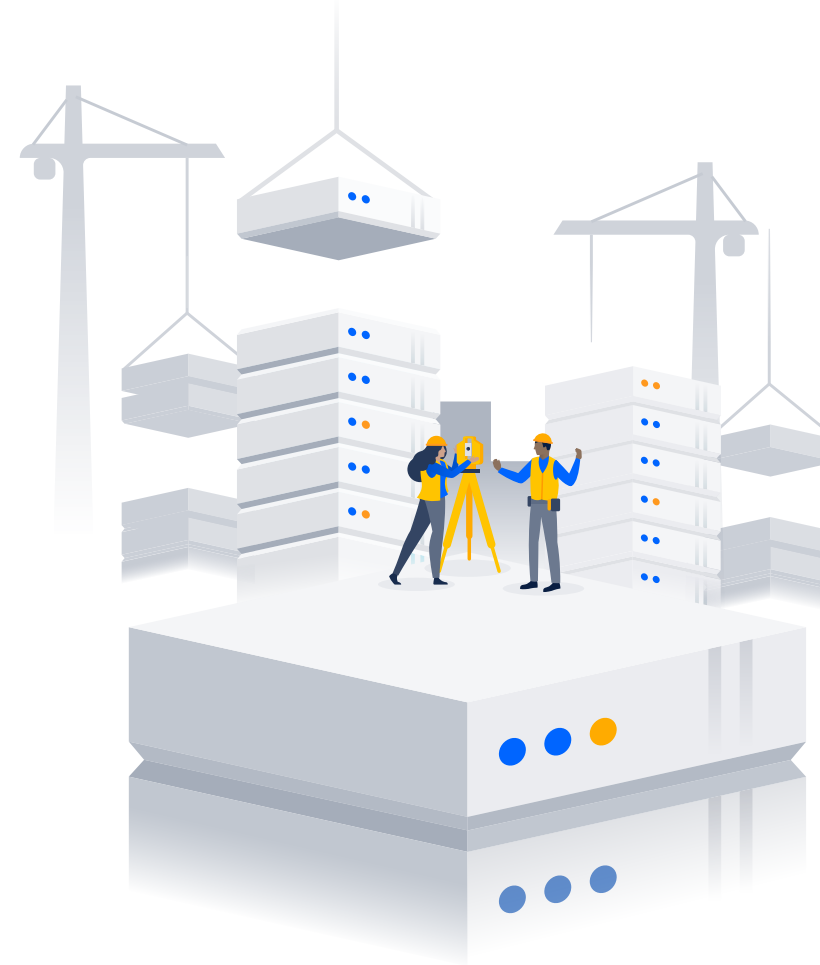
Deploy



Release



Monitor

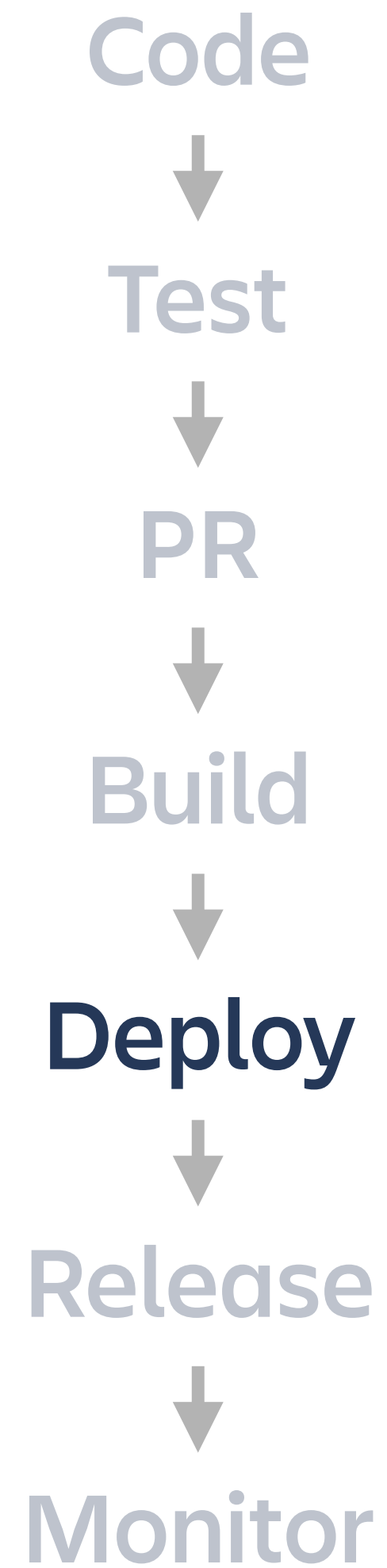


## Webpack hiper-optimizado

- Plugins propios
- Reusar cache

## Build once, deploy everywhere

No hagas build para dev y otra build para producción, usa la misma.

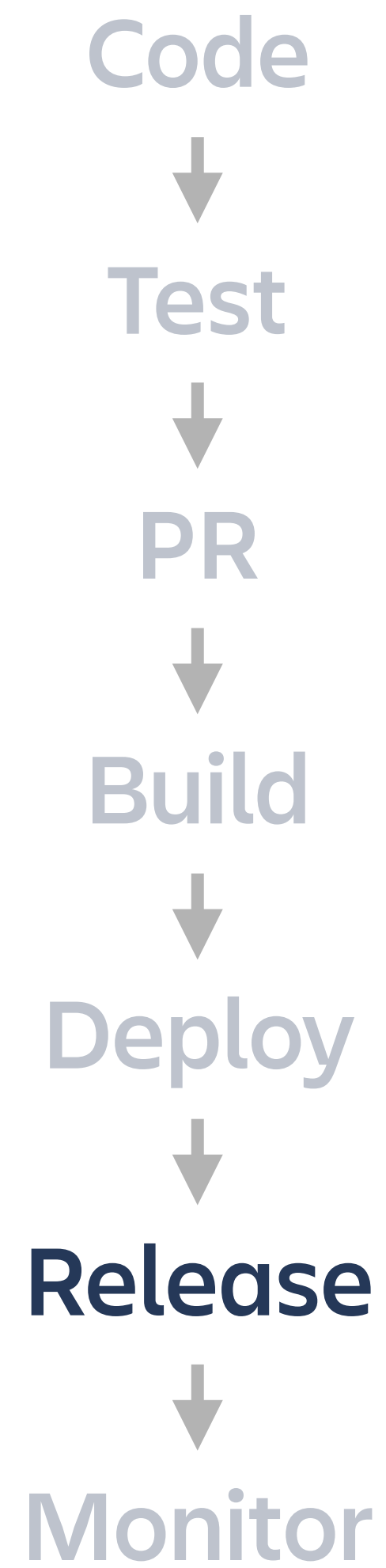


## 100% automatizado

Botón de pánico para parar los despliegues usando Jira

## Despliegue progresivo

- Internal: cada hora
- Asia/Pacific: Internal + 1 hora
- EU/US: AP + 2 horas

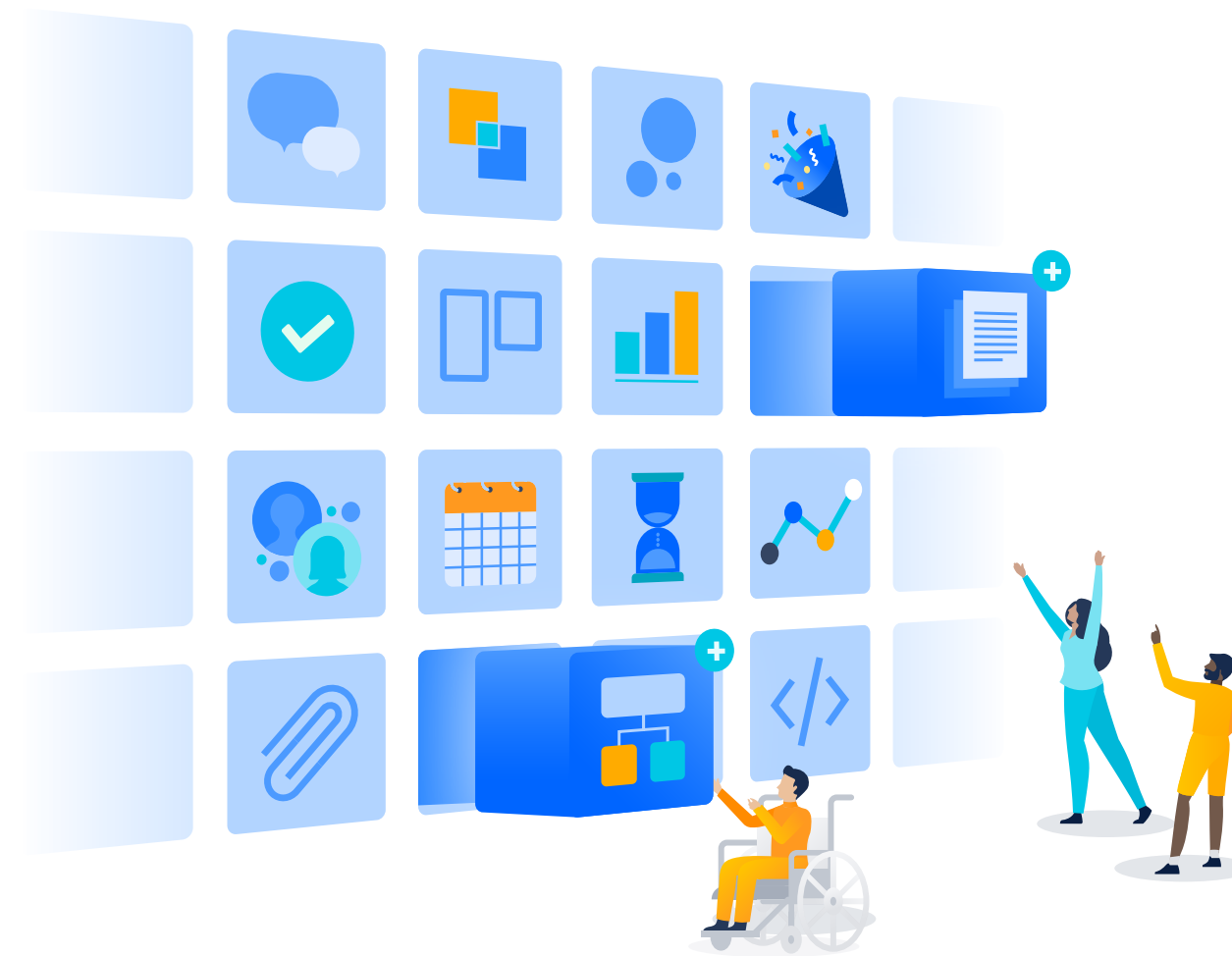
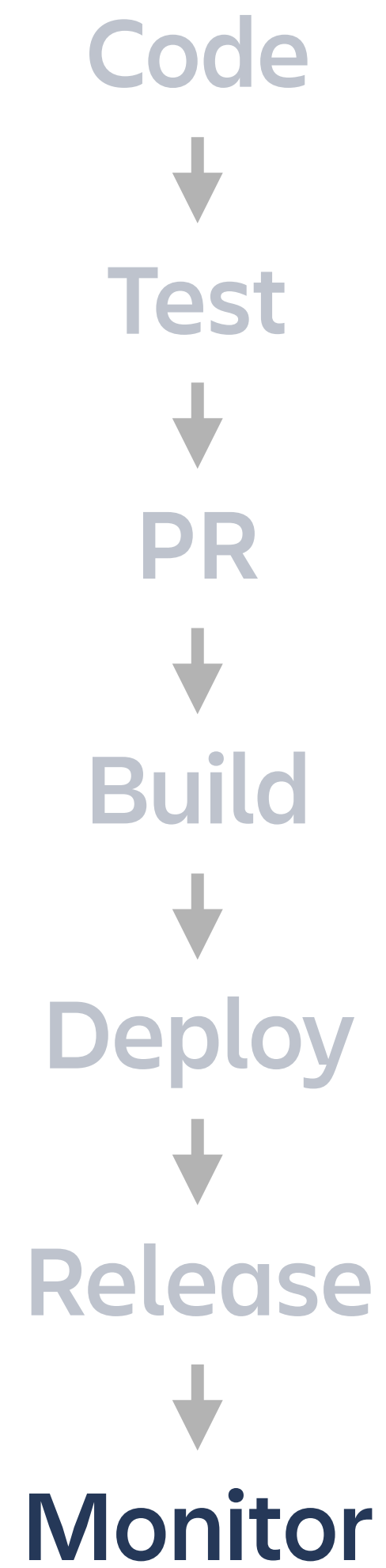


## Release != deploy

- Deploy: “subir” código
- Release: dirigir trafico al nuevo código

## Feature flags





## Frontend == Backend

Frontend no es ciudadano de segunda,  
tiene que ser monitoreado igual que  
backend



Code



Test



PR



Build



Deploy



Release



Monitor

## Errores

- Siempre hay errores
- Lo importante son tendencias



## Eventos

- Objetivo: disparar alertas cuando hay anomalías
- Logea eventos de negocio



## Logs

- Objetivo: diagnosticar errores
- Ojo con logear información privada



## Alertas

- Objetivo: restablecer el servicio cuanto antes.
- Frontend moderno require 24/7







# Thank you!



**SERGIO | PRINCIPAL DEVELOPER | ATlassian | @SCINOS**

