

# El poder oculto de las CLIs

Catedra UV x Capgemini | Pepe Fabra Valverde | 2025

UNIVERSITAT  
DE VALÈNCIA

x  Capgemini



**C&CA** CLOUD &  
CUSTOM  
APPLICATIONS

**SDO**  
Parte de **C&CA**

# Autobús Capgemini

*2 de abril*



*Haz que tu carrera vaya  
sobre ruedas.*

↑ Hoy ▲ ▼ Abril 2025

Lunes	Martes	Miércoles
31 Mar	1 Abr	02

# Pepe Fabra Valverde

Senior Software Engineer

Developer Advocate



# Disclaimer

- Compartiré las slides al final de la sesión
- Son conceptos nuevos
- Puede que sobre tiempo



# Agenda

- Charla
  - CLIs, automatización y más

# Agenda

- Charla
  - CLIs, automatización y más
- Taller práctico
  - Crea tu propia CLI

# Experiencia con comandos

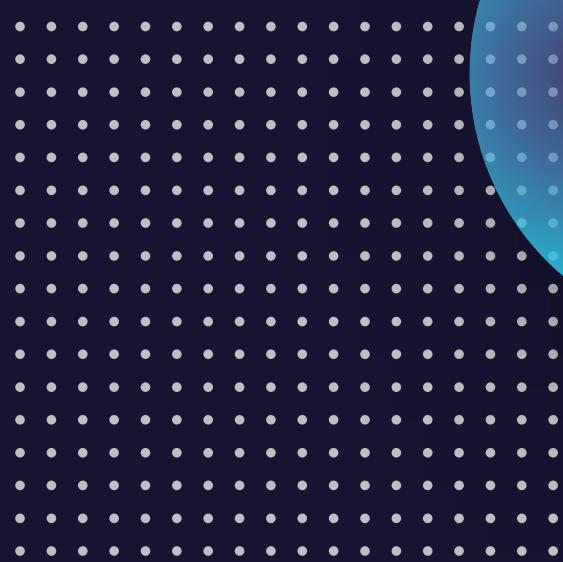
*Qué son, uso... y creación*

Command-line interface

Automatización

Developer Experience

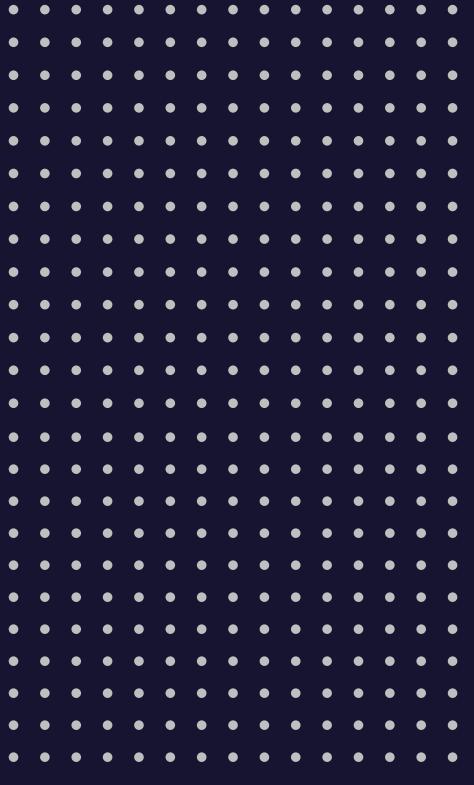
Terraformadores y validadores



Todo el mundo odia los  
comandos

...o les tiene miedo

# Las CLIs abruman



# Son simples

# Ayudan

# Release, code-gen, RPAs

# Entendamos mejor las CLIs

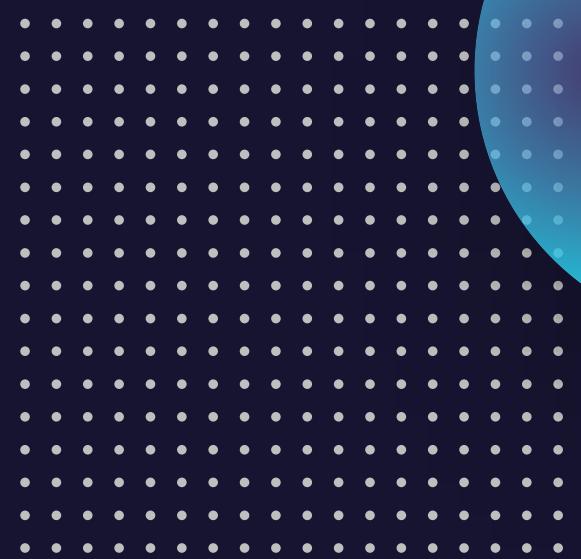
# Command-line interface

*Detalle y visión general*

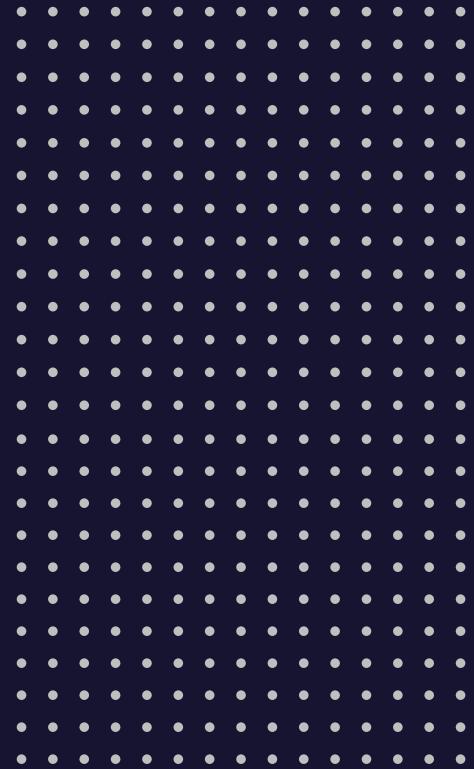
Automatización

Developer Experience

Terraformadores y validadores

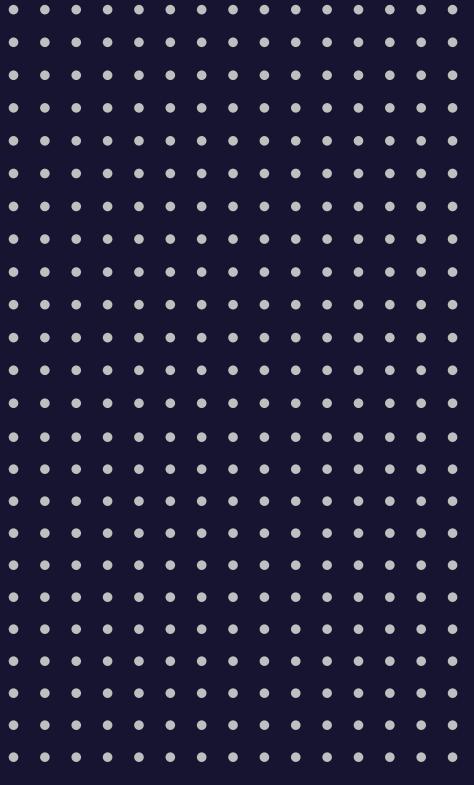


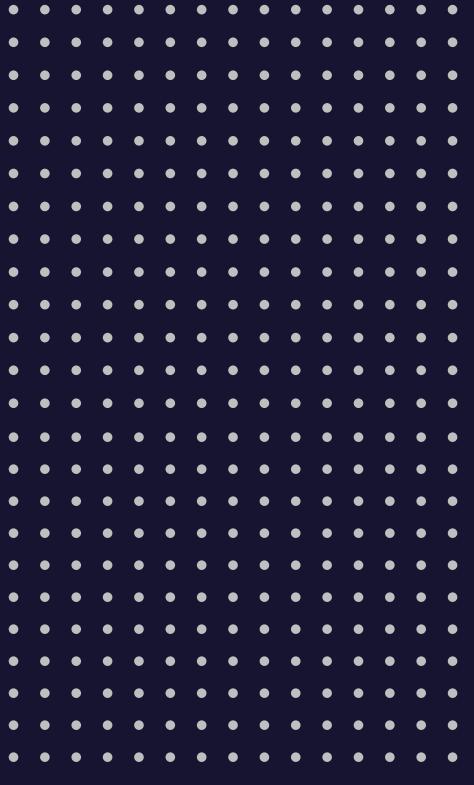
# Instrucciones reutilizables, parametrizables

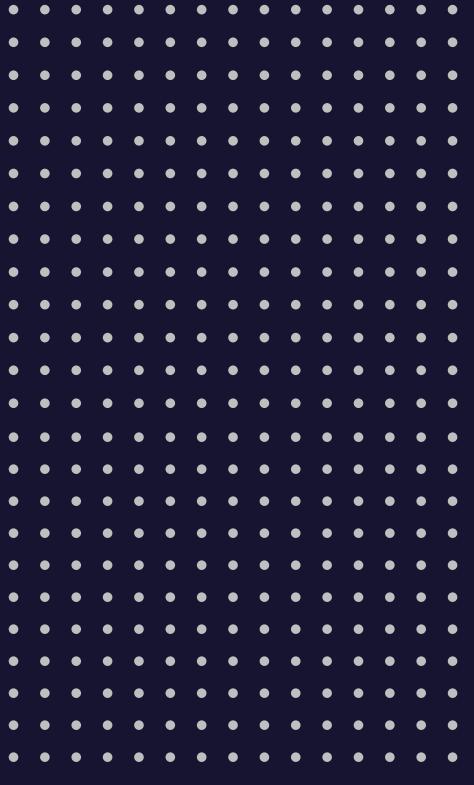


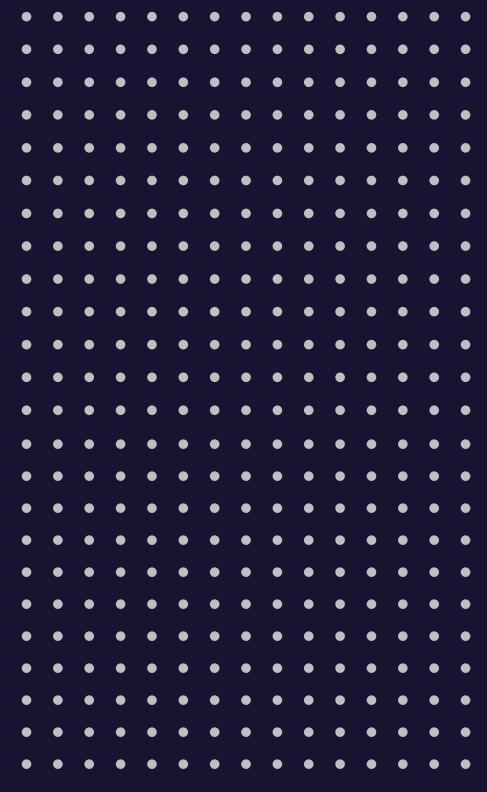


# Son funciones



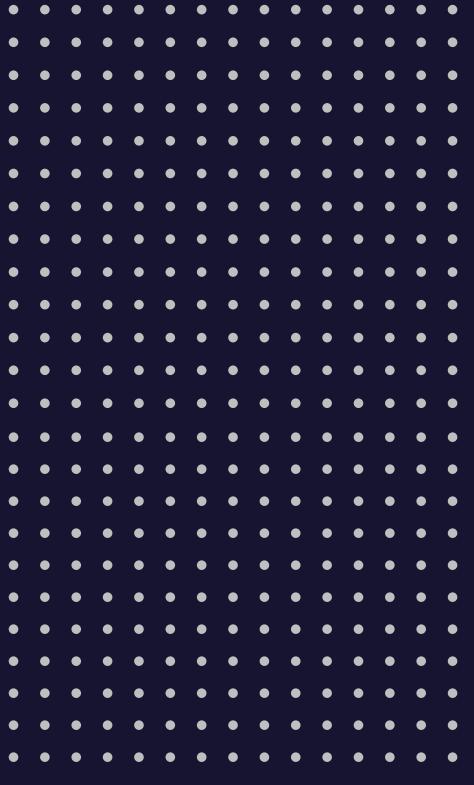
- 
- 
1. *input*
  2. proceso
  3. *output*

- 
1. *input*
  2. proceso
  3. *output*
- 

- 
1. *input*
  2. proceso
  3. *output*
- 

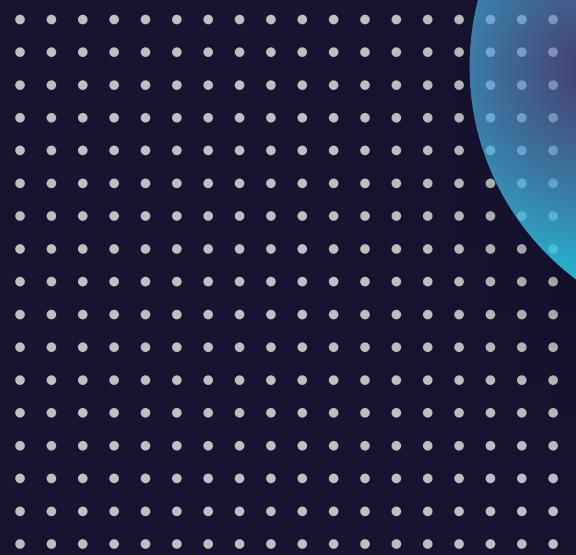


CLIs -> Interactuar  
con sistemas

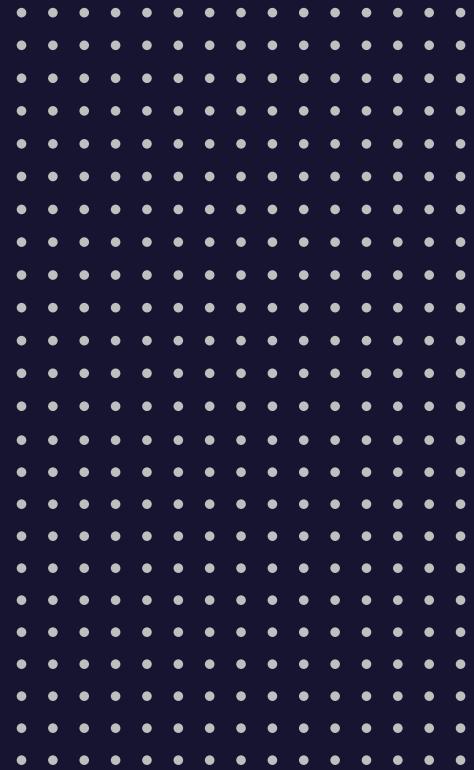


command-line interface

# pathing



# Enrutación



# Acceso a recursos

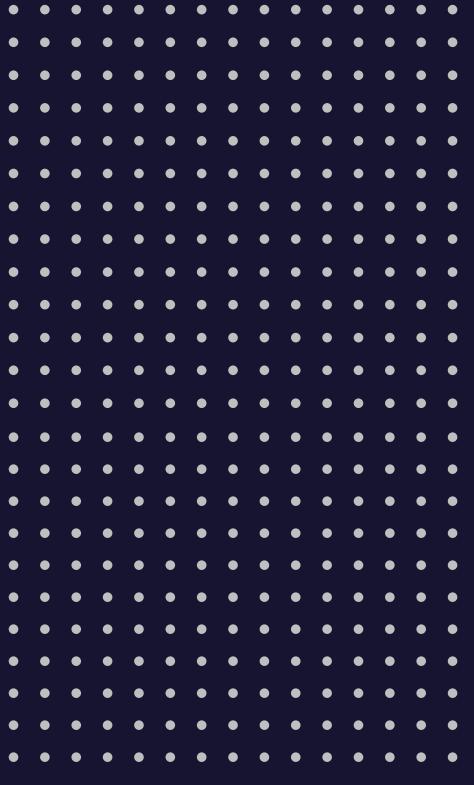
# Origen y destino (no filosofía :D)

command-line interface

args y uri



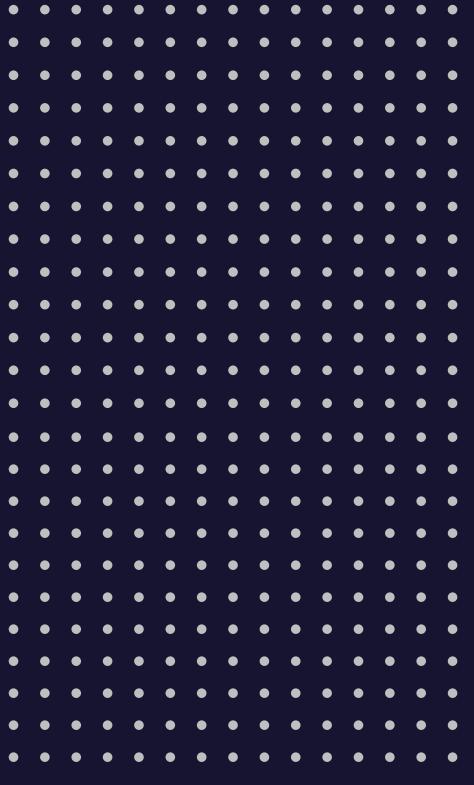
Las funciones tienen  
argumentos



# URI-encoded



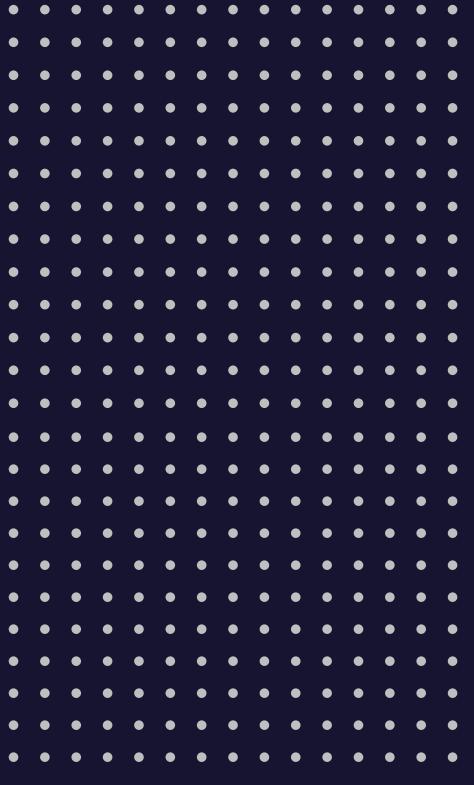
# ¿OS SUENA?



# Unified Resource Identifier



# Existe o no



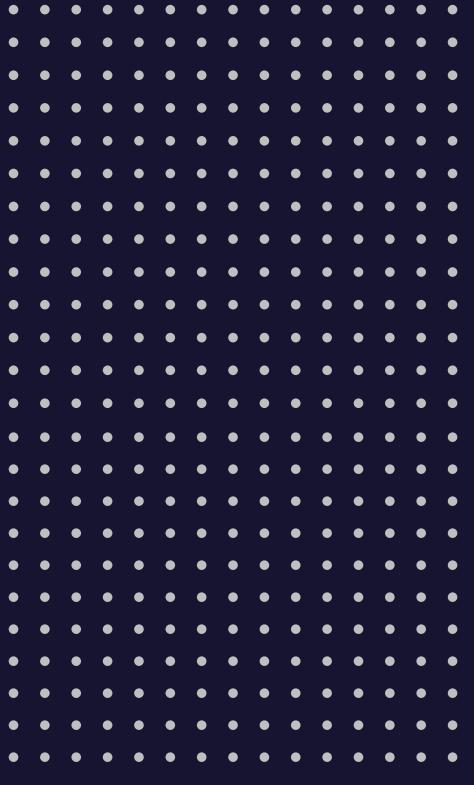
# Físico o abstracto

# Solo admite ASCII

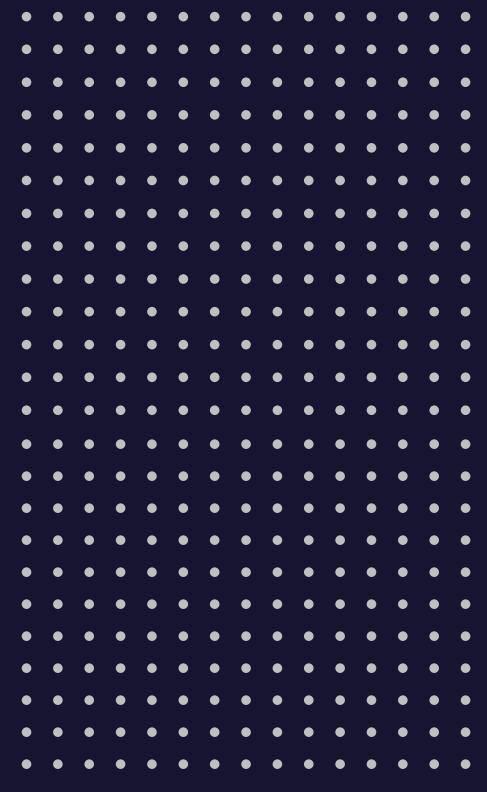
%20 -> ‘/’



# y de argumentos



# Argumentos nombrados



# Verbosos (--) o abreviados (-)

command-line interface

# pipelines

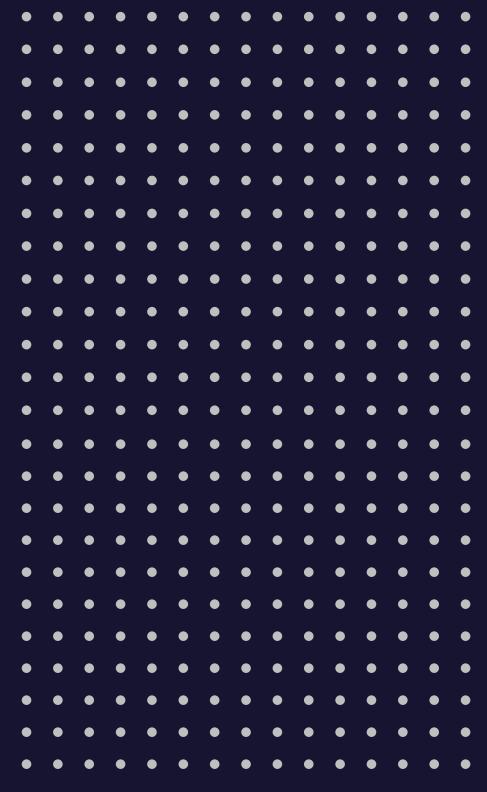
# Concatenar comandos

# Pipe operator ‘|’ encadena

# Paralelización ‘&’ asíncrono



Problemas  
pequeños... pero  
reutilizables



# Eficientes

# Patrón de arquitectura

# Jenkins y CI/CD

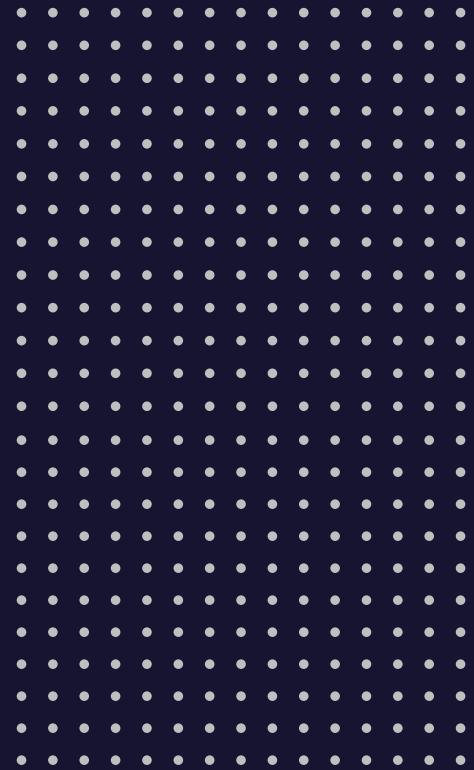
# Pipes en Programación Funcional

# MapReduce, Hadoop, Pipe Operator “|>”

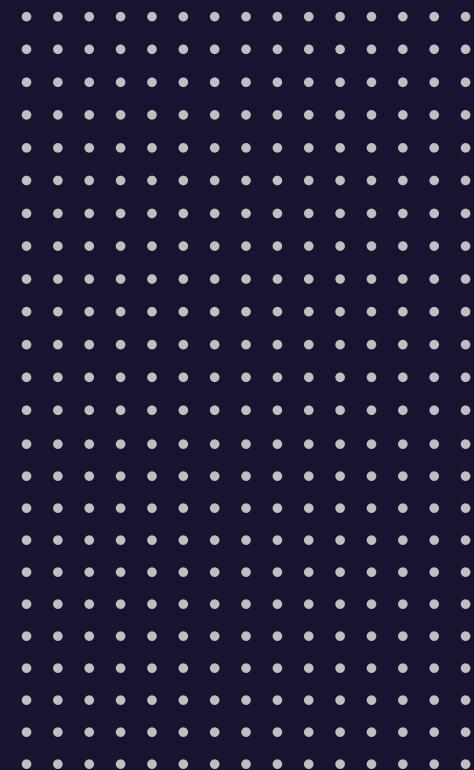
command-line interface

apis

# Application Programming Interface



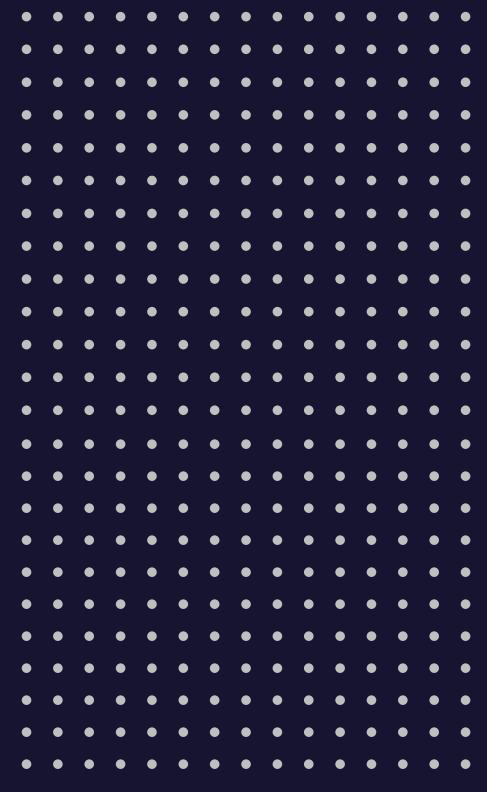
# Interacción con módulo o sistema



# Semántica en programación



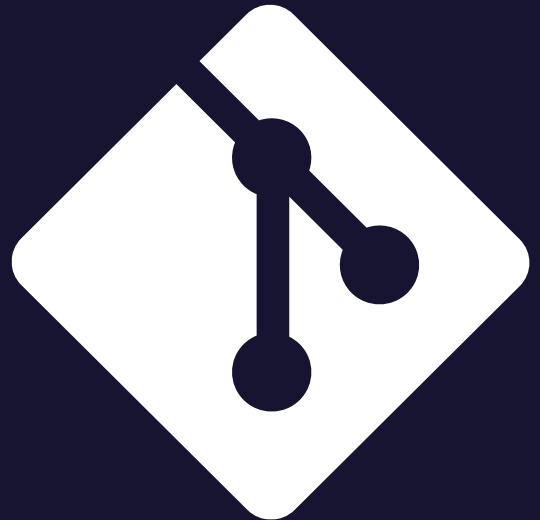
Quién lo va a usar y  
cómo se leerá



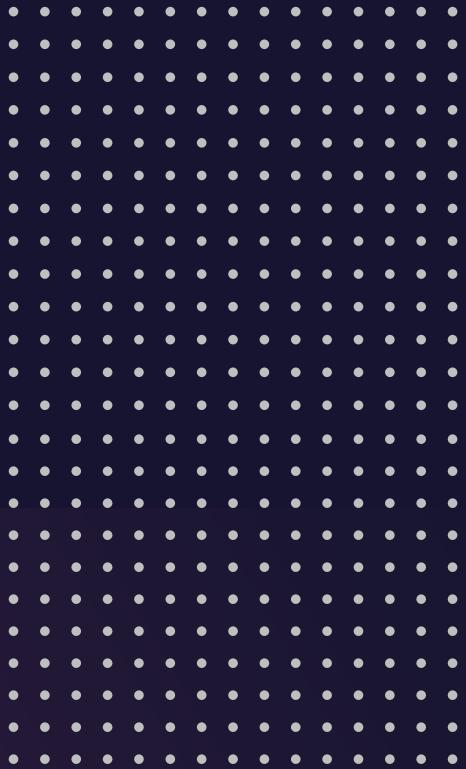
# API-First

# Lo veremos mejor en el taller

# Google Cloud Platform CDK



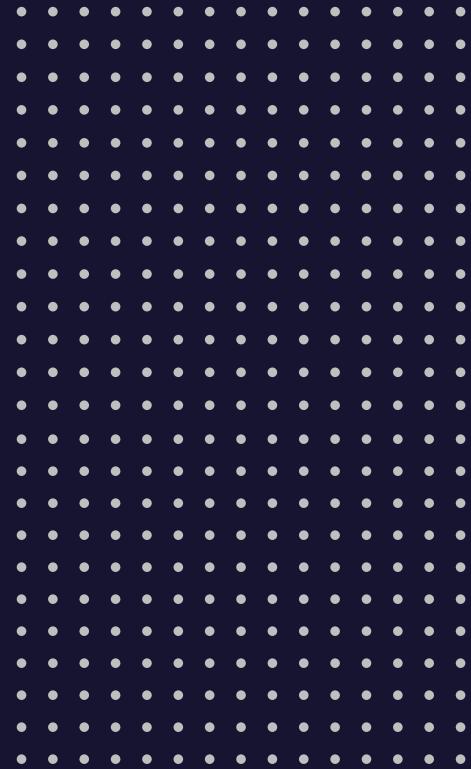
# git



command-line interface

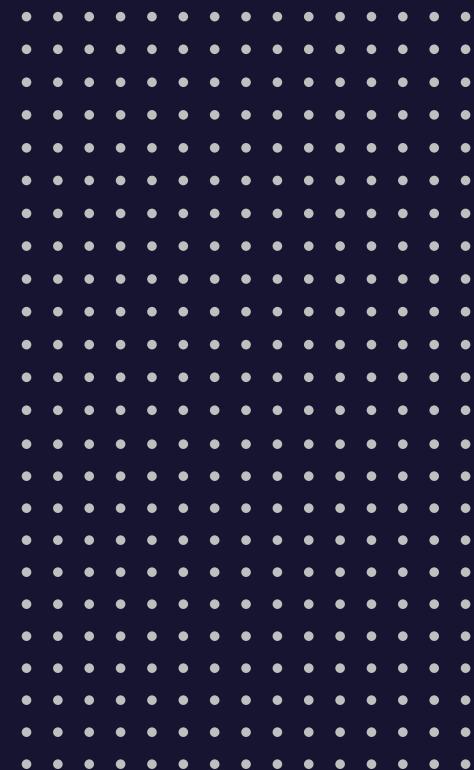
# typescript

# Tipado flexible y ergonómico



# CommonJS y ModuleJS

# NPM



command-line interface

java

# Tipado fuerte

# Build más natural

# Validaciones

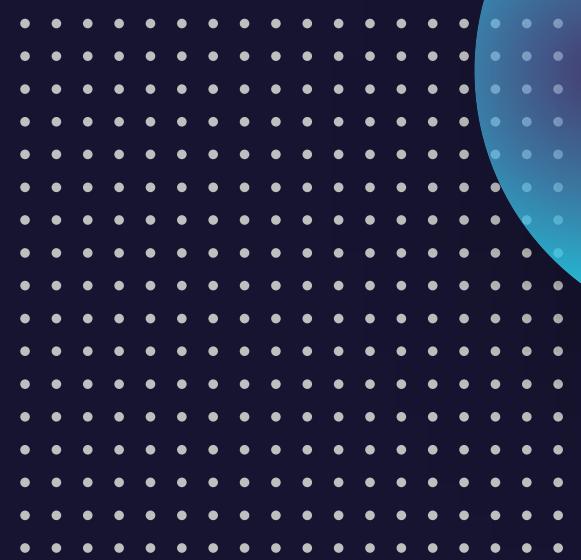
Command-line interface

# Automatizaciones

*DevOps, Arquitectura y comodidad*

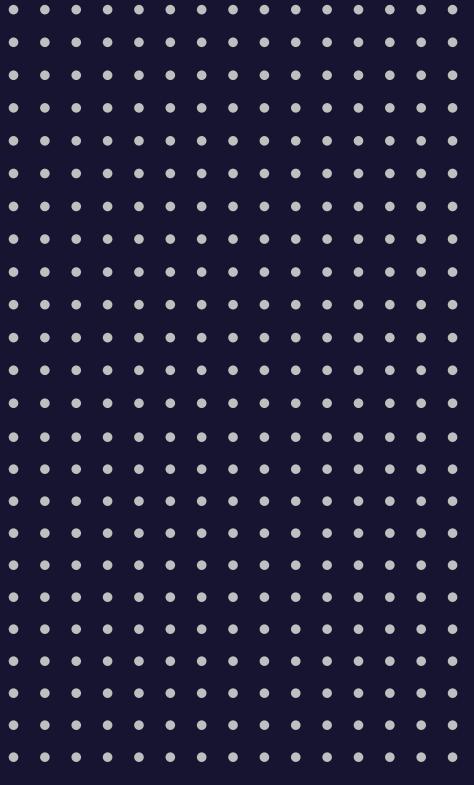
Developer Experience

Terraformadores y validadores



A wise engineering solution  
would produce – or better,  
exploit – reusable parts.

Douglas McIlroy, [leancrew.com/all-this/2011/12/more-shell-less-egg/](http://leancrew.com/all-this/2011/12/more-shell-less-egg/)



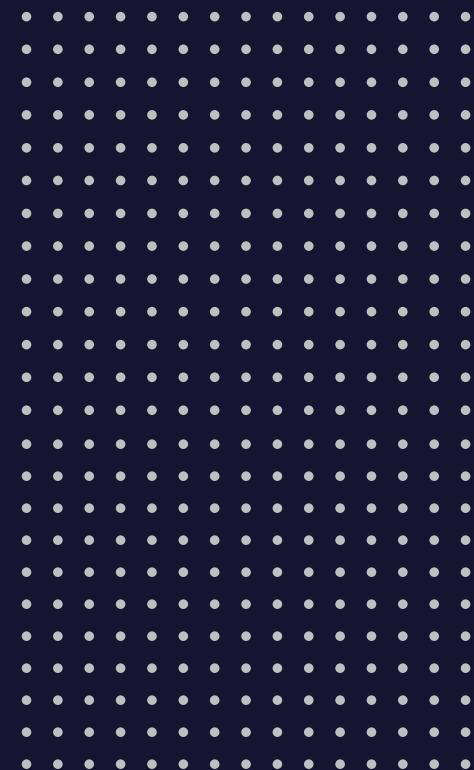
# No repetir procesos

# Ahorro de tiempo

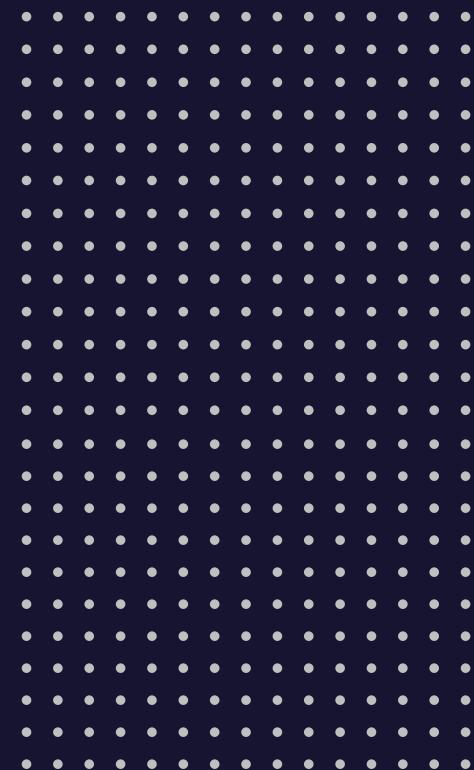
# Web scraping

# Reestructurar directorios

# Procedimientos tediosos



# Procedimientos frágiles

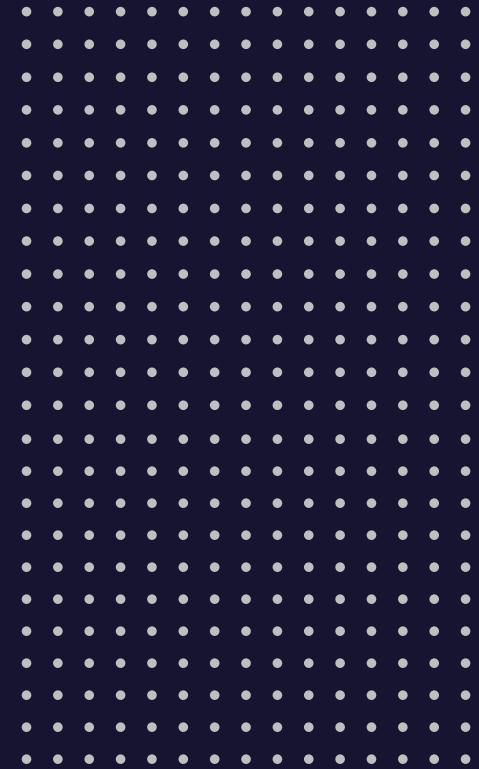


# Reglas de negocio o arquitectura

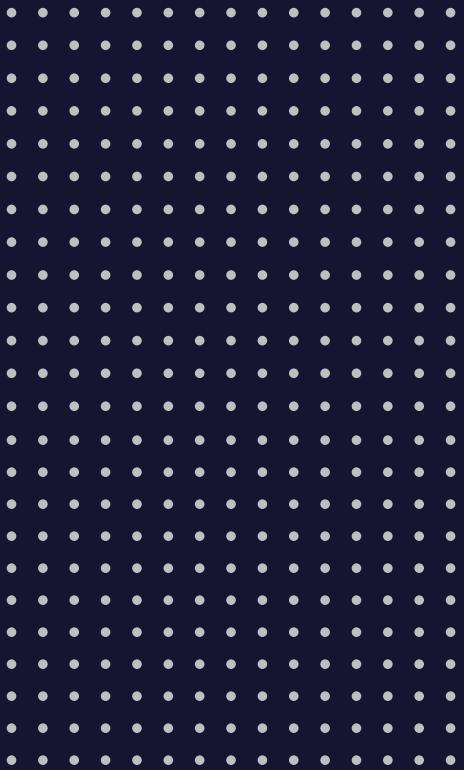
automatizaciones

# crontab

# Gestión y programación de tareas en el tiempo



# Schedulers



\*\*\*\*\*

# Minutos (0-59)

\* \* \* \* \*

# Hora (0-23)

\*\*\*\*\*

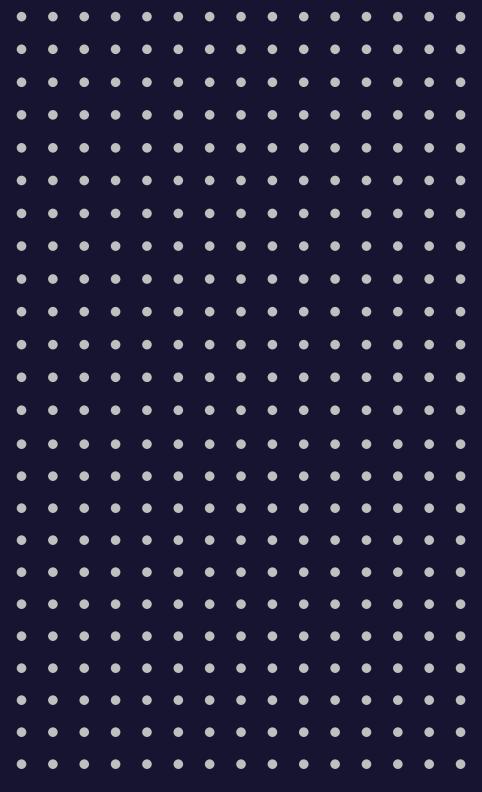
# Día del mes (1-31)

\*\*\*\*\*

# Mes del año (1-12)

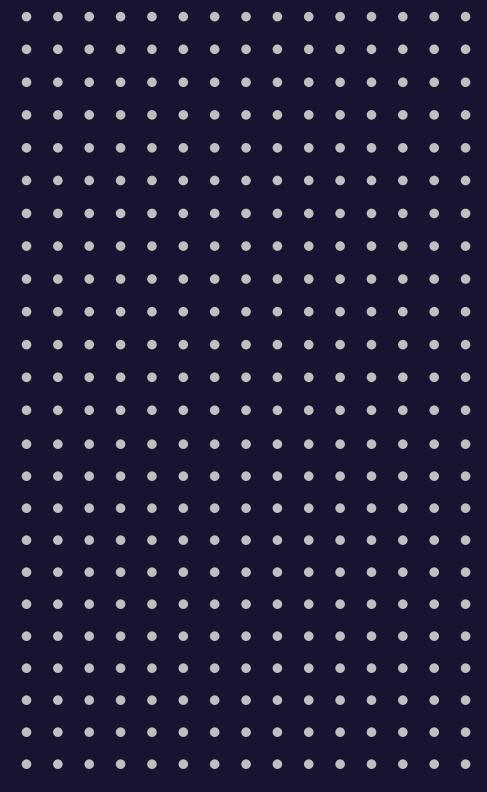
\*\*\*\*\*

# Día de la semana (0-6)





# Se complican más



<https://crontab.guru>

automatizaciones

# hooks y eventos

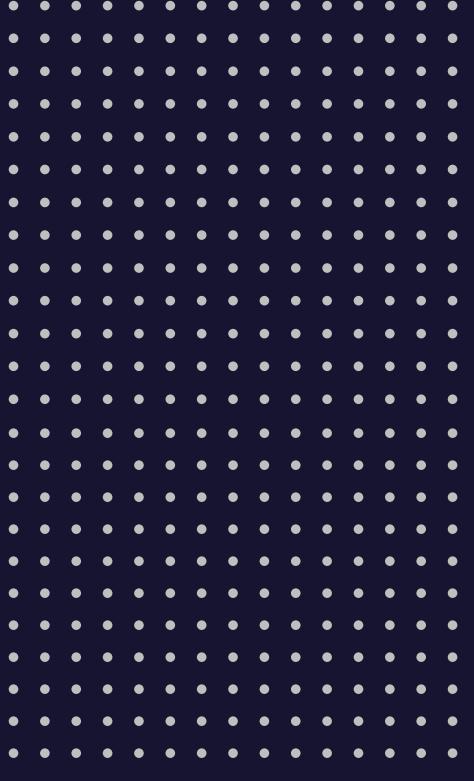
# Eventos concretos

# Git hooks, husky

# Template Method Design Pattern



Automatiza X  
Cada vez que Y



automatizaciones

pereza

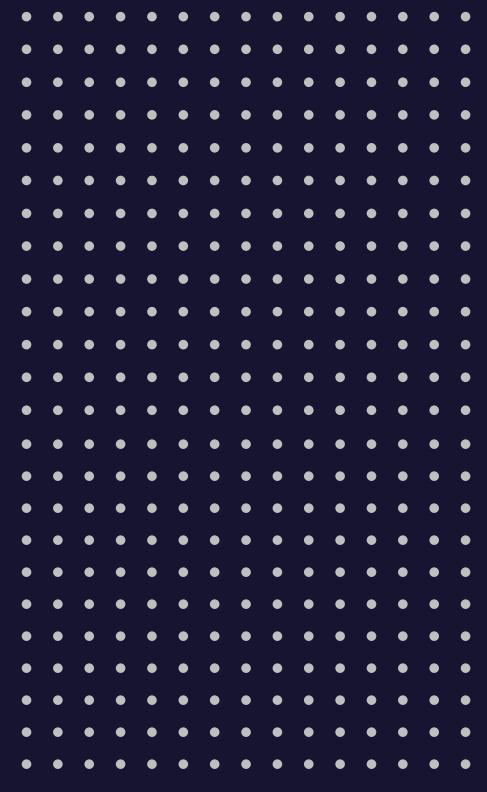
# Hábitos atómicos

# Procrastinación

# Vaguería como motivador de mejora

I choose a lazy person to do a hard job. Because a lazy person will find an easy way to do it

Frank Gilbreth Sr., Clarence Bleicher



# Automatizado es más fácil hacerlo

automatizaciones

# devops y scripting

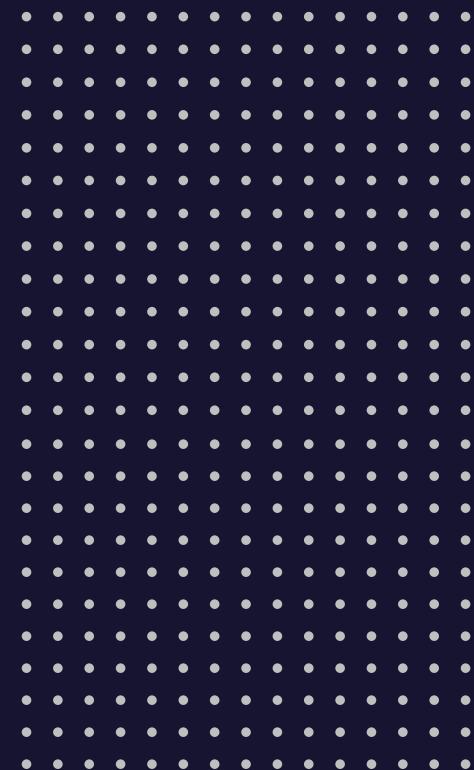


# Cultura DevOps

# CI/CD y scripts

# Validadores de código

# SonarQube



# DevOps y automatizaciones

automatizaciones

# architecture testing

# ADRs

# Fitness Functions

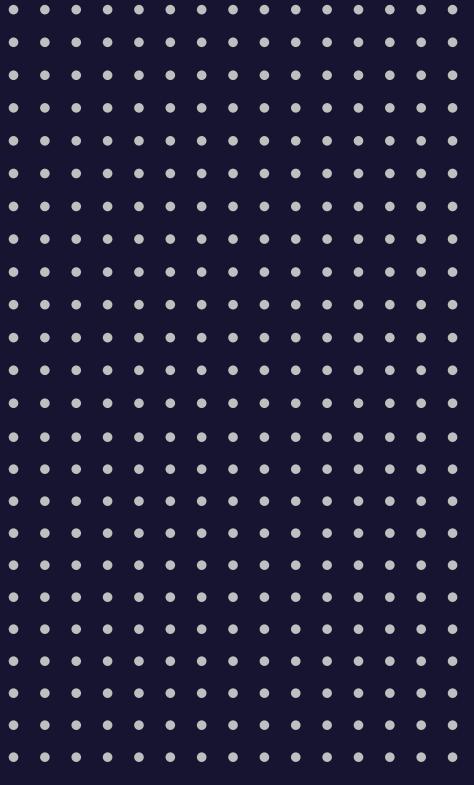
# ArchUnit

# Cliente IoT

# Arquitectura Front

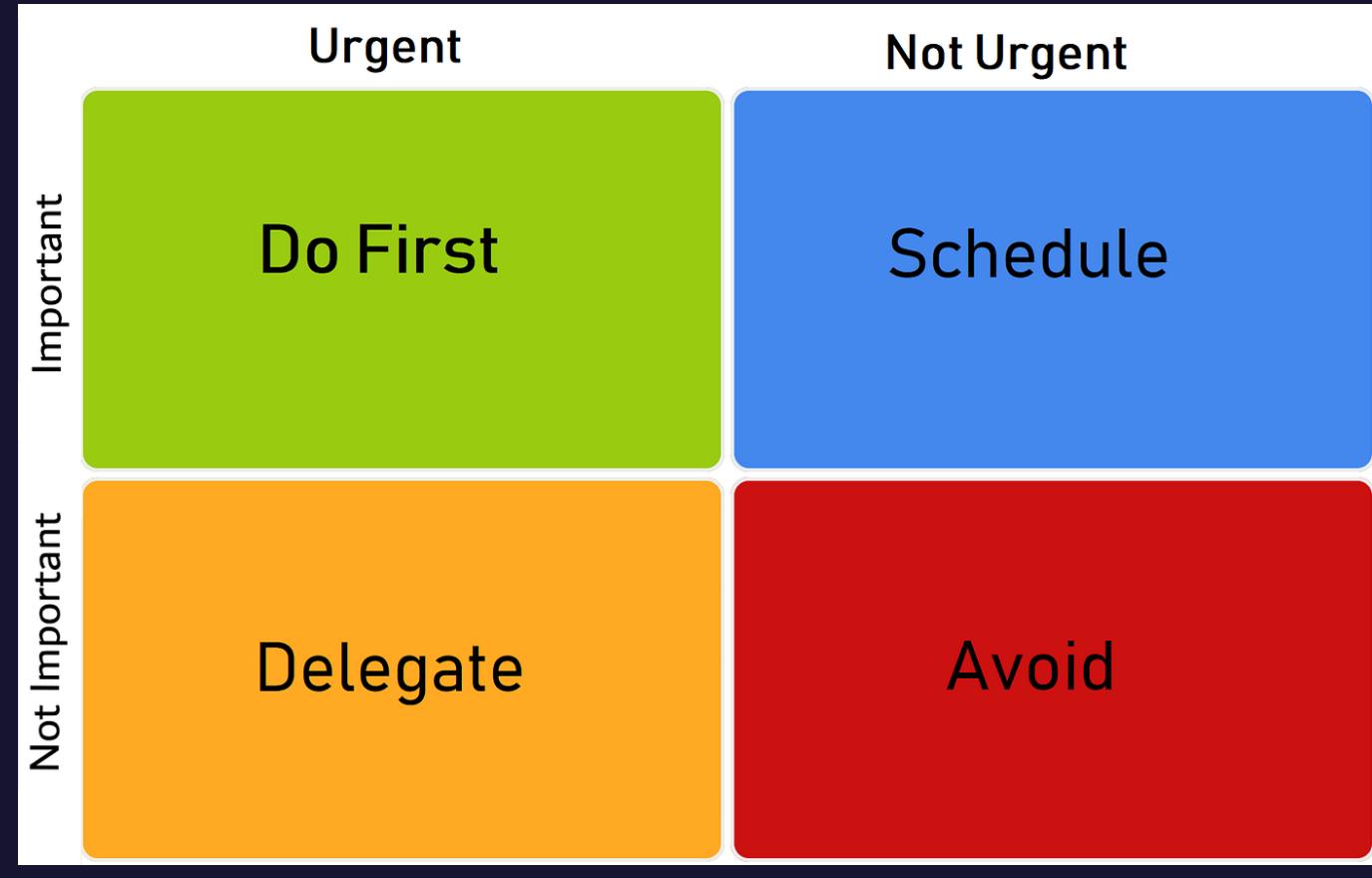
automatizaciones

# matriz de Eisenhower



# ¿qué es?

# Matriz de Eisenhower



# Priorización

# Matriz Eisenhower para automatización



# Matriz Eisenhower para automatización

Muchas veces, poco tiempo

**iñi te lo pienses!!**

# Matriz Eisenhower para automatización

Muchas veces, mucho tiempo

puede...

# Matriz Eisenhower para automatización

Pocas veces, poco tiempo

con hueco...



# Matriz Eisenhower para automatización

Pocas veces, mucho tiempo

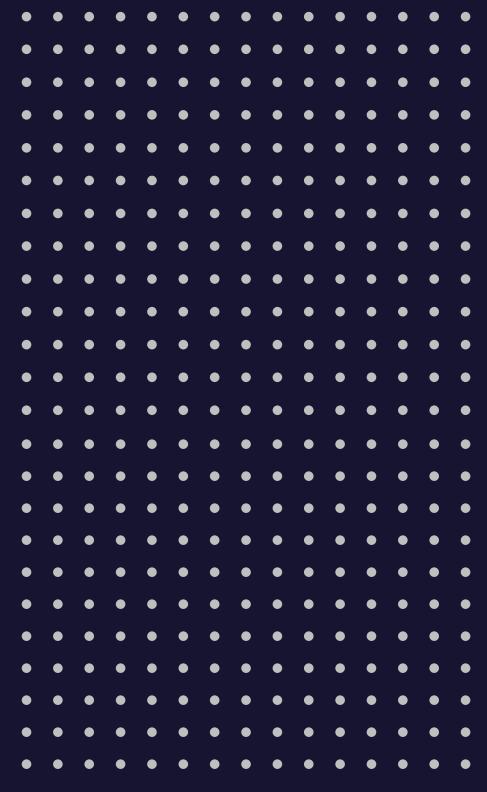
iñi de broma!!

# Matriz Eisenhower para automatización



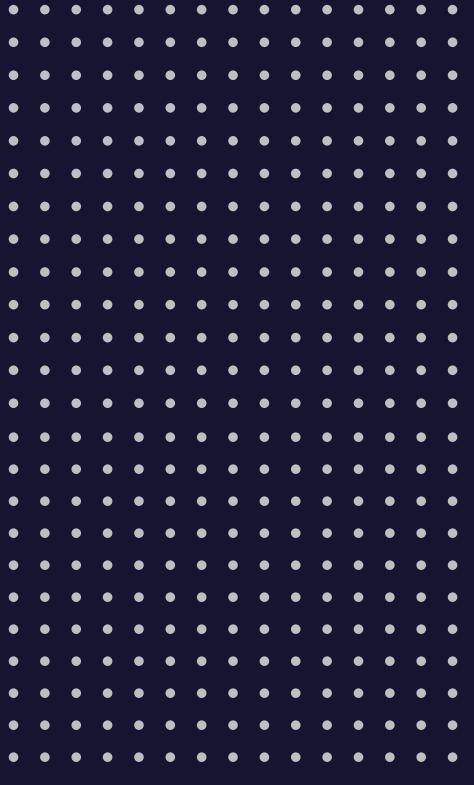


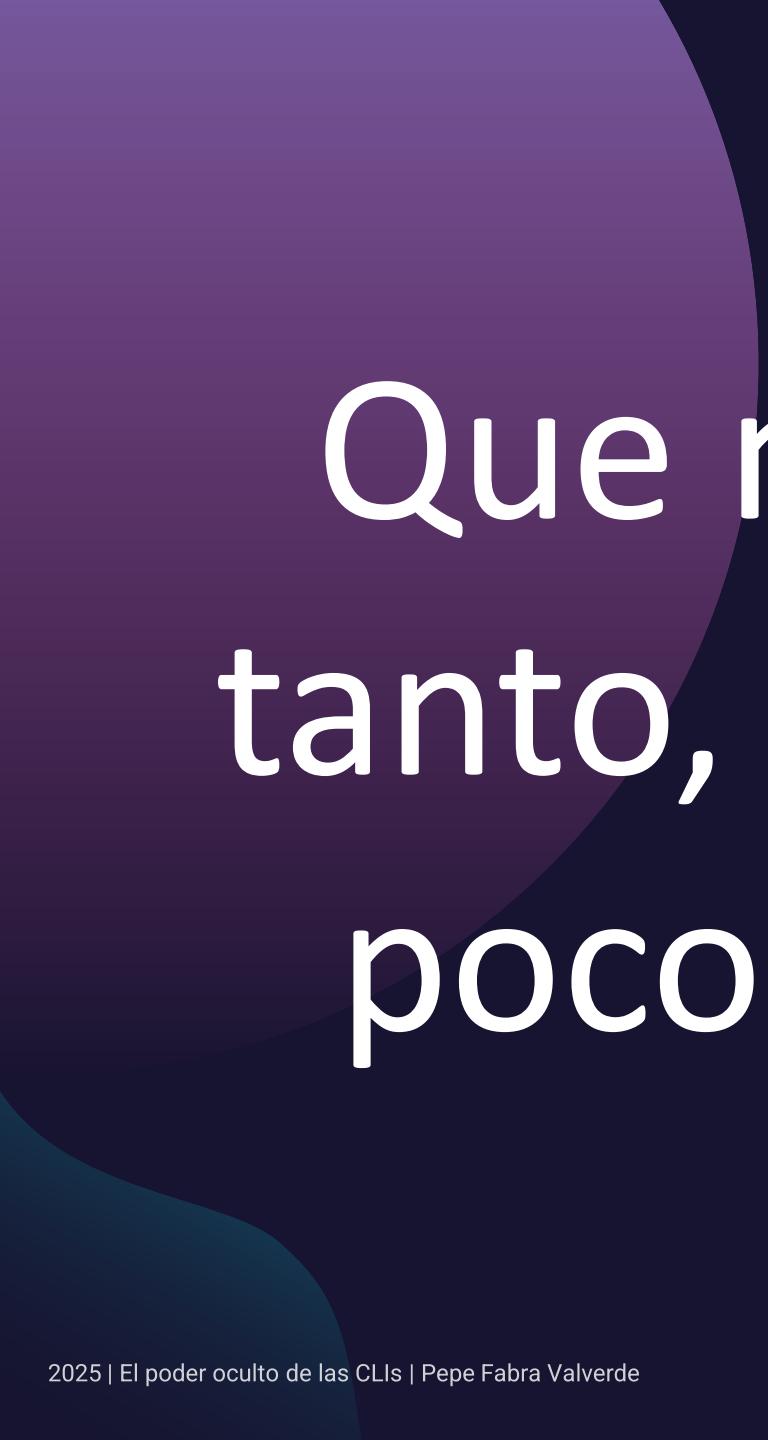
Automatizamos  
cuando es una  
tarea...



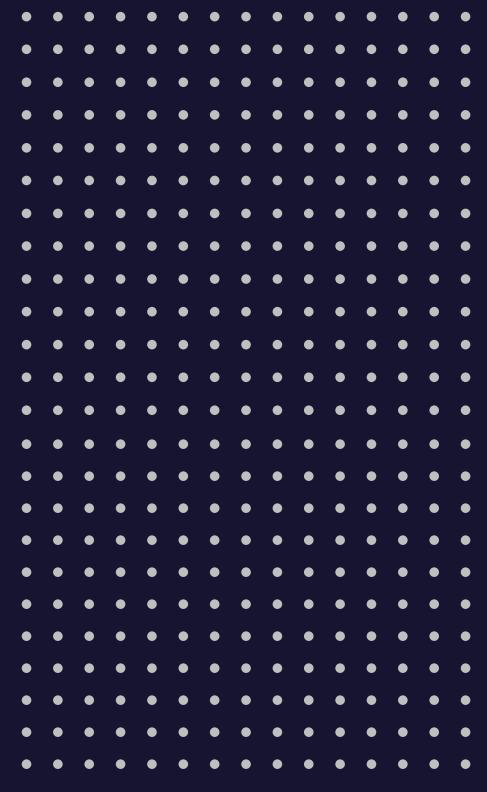


Que repetimos  
mucho



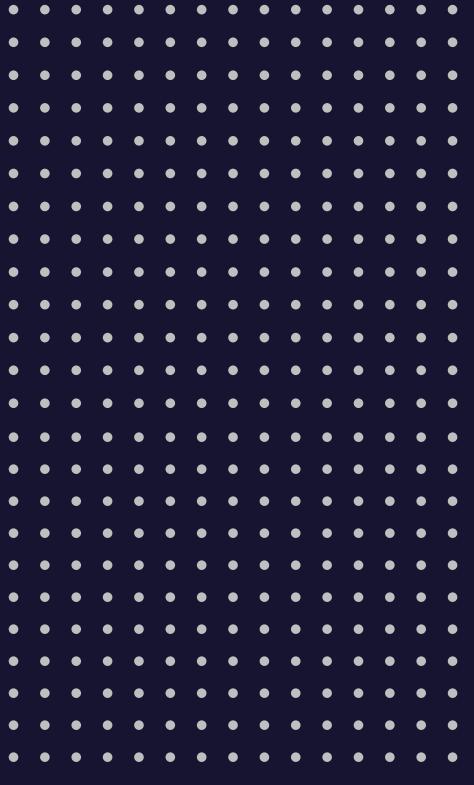


Que no repetimos  
tanto, pero nos lleva  
poco automatizar

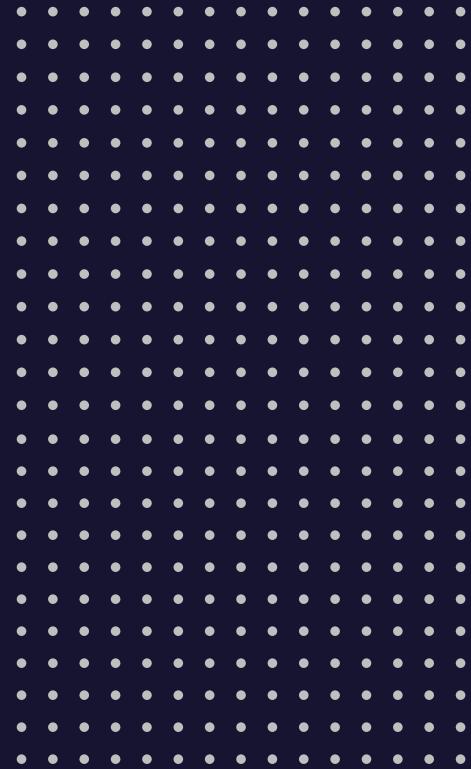


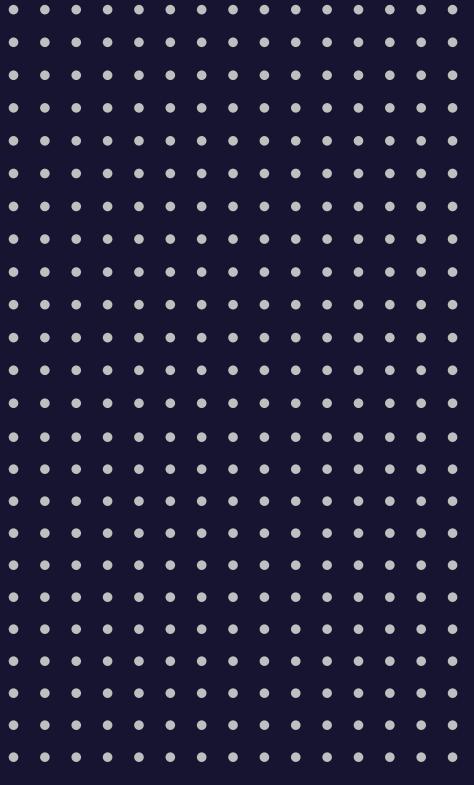


Que es crítico no  
saltarse ningún paso



Que es muy larga  
(riesgo de saltarnos  
pasos)

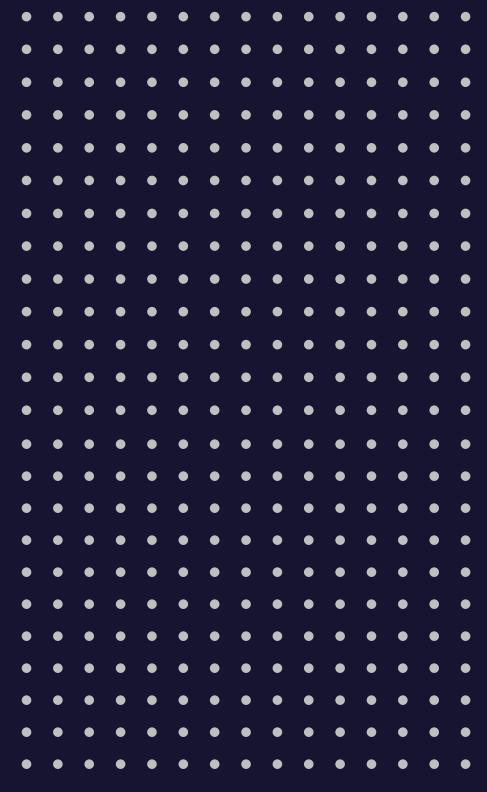




# Automatiza lo que te vaya a ser útil



Cuánto lo usas y el  
tiempo que tardarás  
en desarrollarlo



15 horas para 10  
minutos que harás  
una única vez... no  
merece la pena

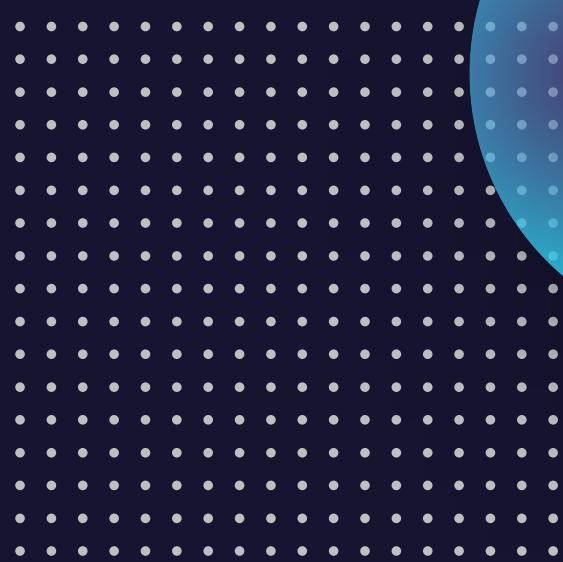
Command-line interface

Automatización

# Developer Experience

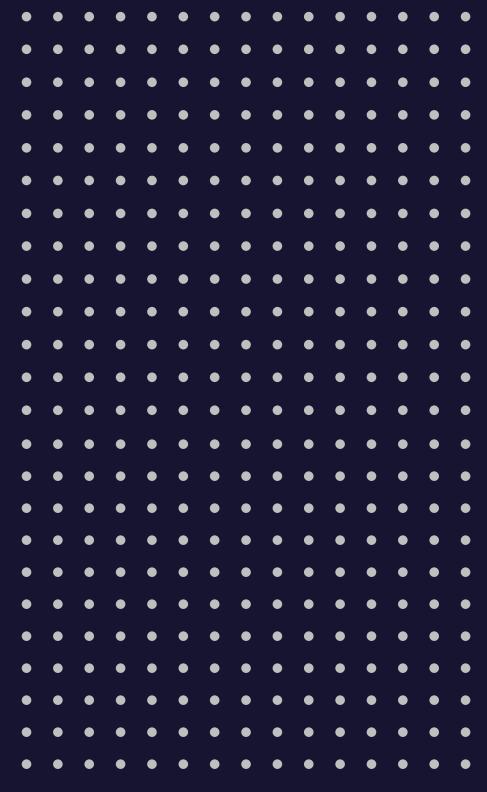
*Qué es y relación con CLIs*

Terraformadores y validadores





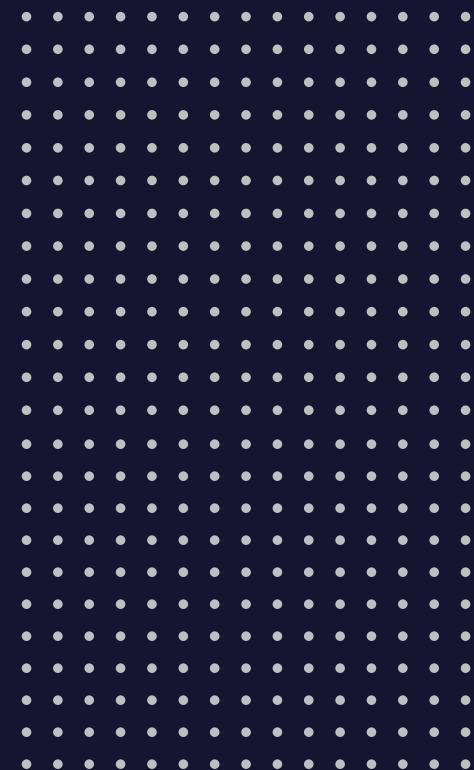
# DX



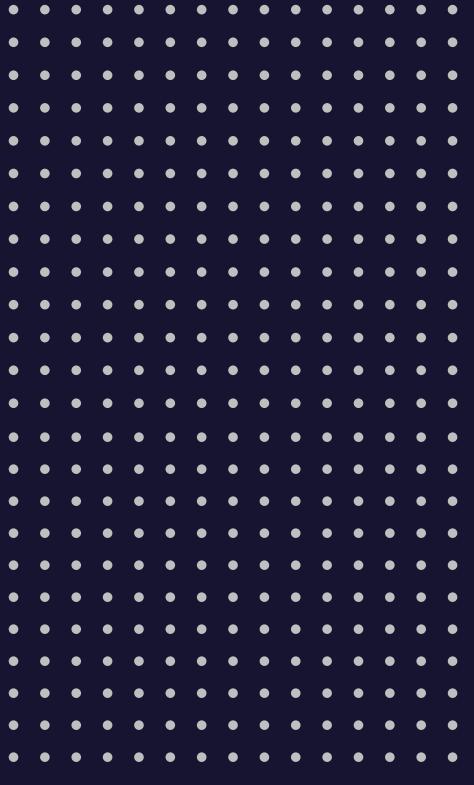
# Utilidades, bibliotecas, frameworks o herramientas

# Plataformas de Ingeniería

# Productividad y confianza



# Desarrollo rápido y seguro



dev-tooling también  
es hacer una CLI

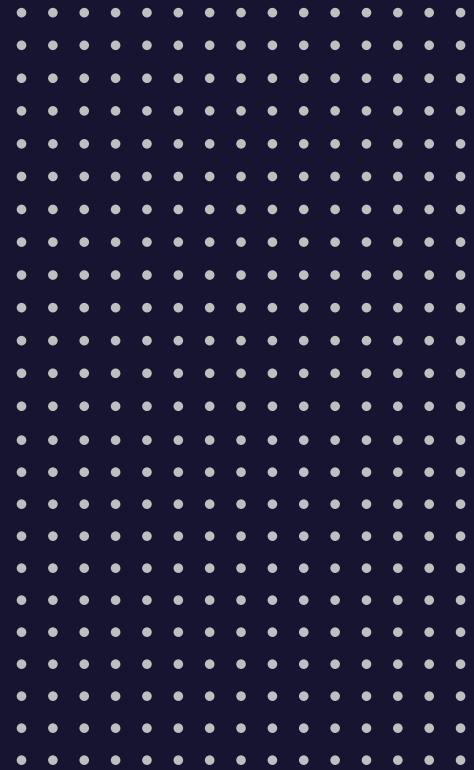
# dev-tooling

# Desarrollas para desarrollar mejor

developer experience

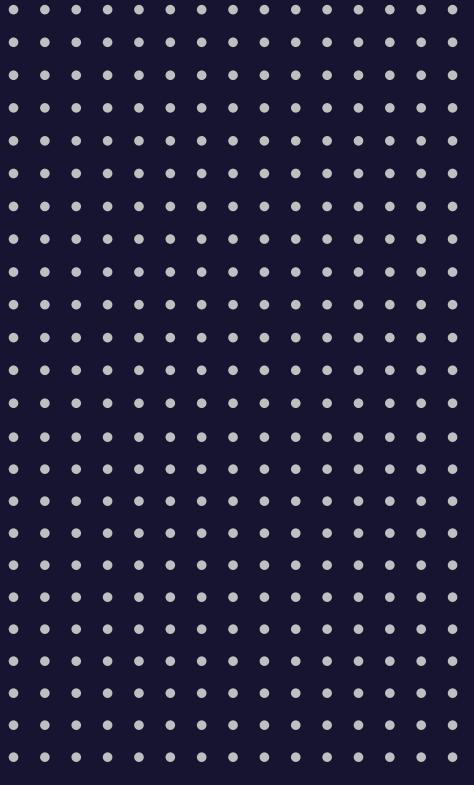
# importancia del DX

# Por qué es importante





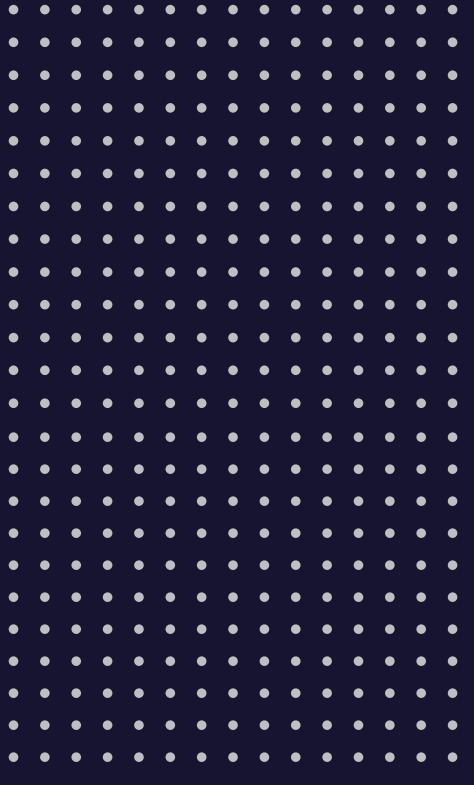
Menos  
para hacer más  
con más confianza



# State of Flow

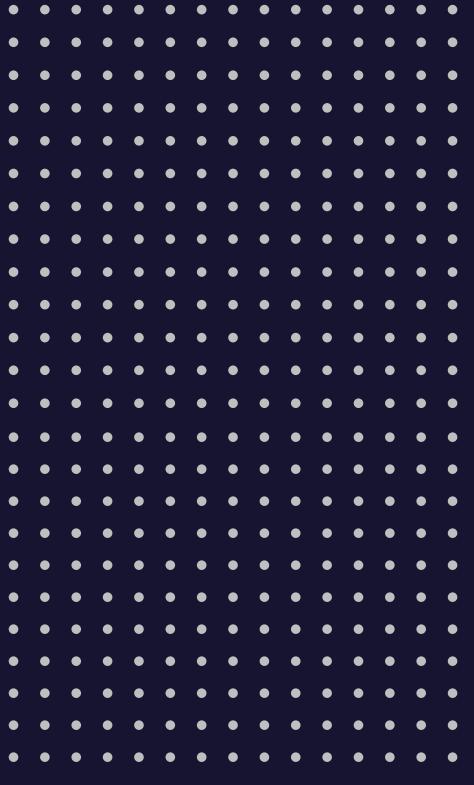


En desarrollo  
importa quien  
desarrolla



developer experience

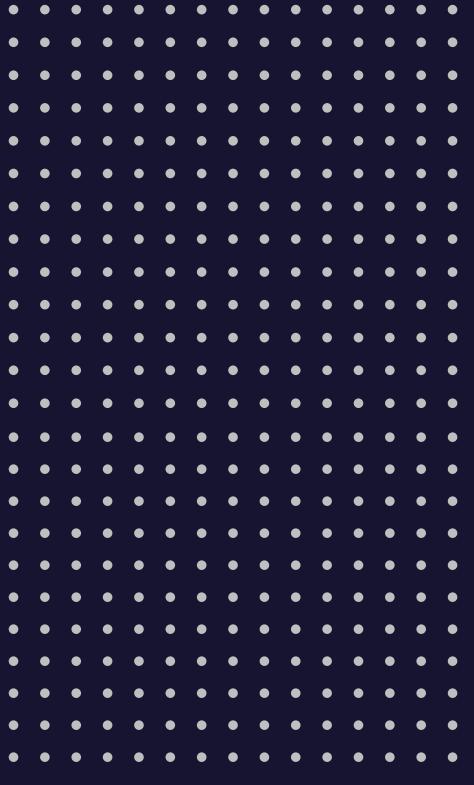
dx <3 cli



# CLIs ayudan a mejorar la DX

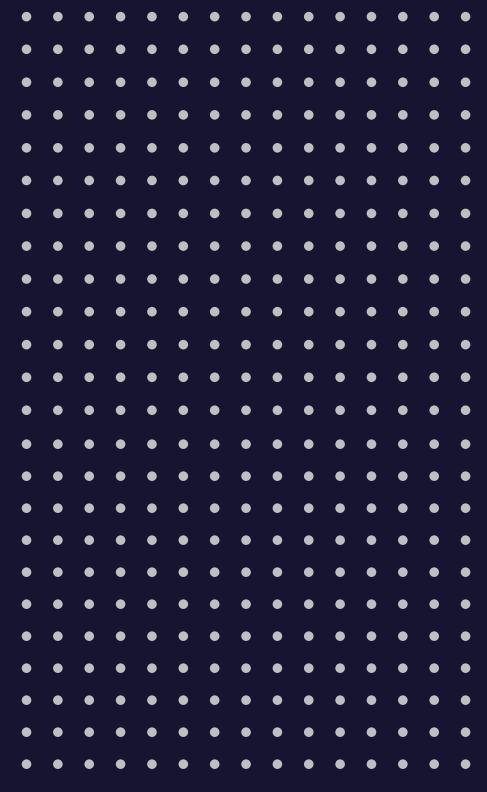


Buena DX incluye  
CLIs





# Son tus manuales



# Acceso a un framework

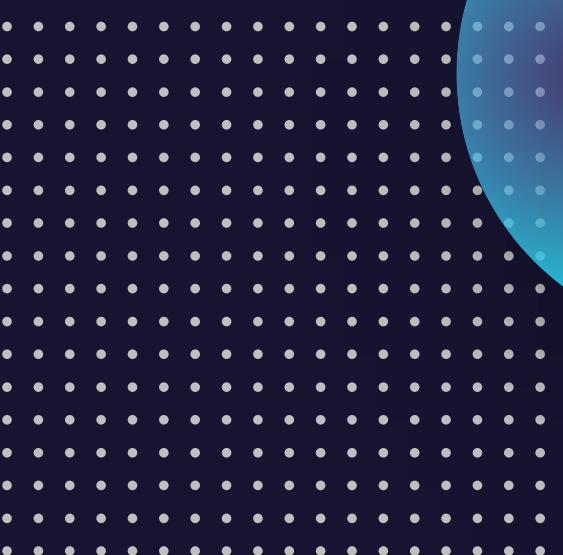
# Laravel y artisan

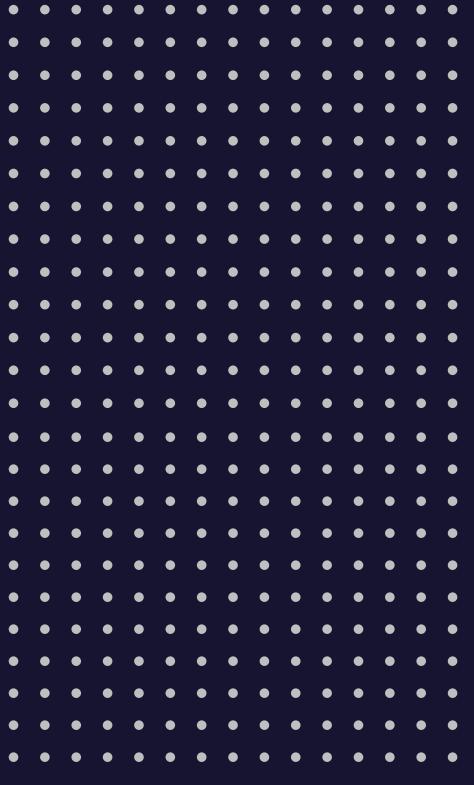
<https://dx.addy.ie/>

Última sección

# Terraformadores y Validadores

*Automatizar cambios de código*



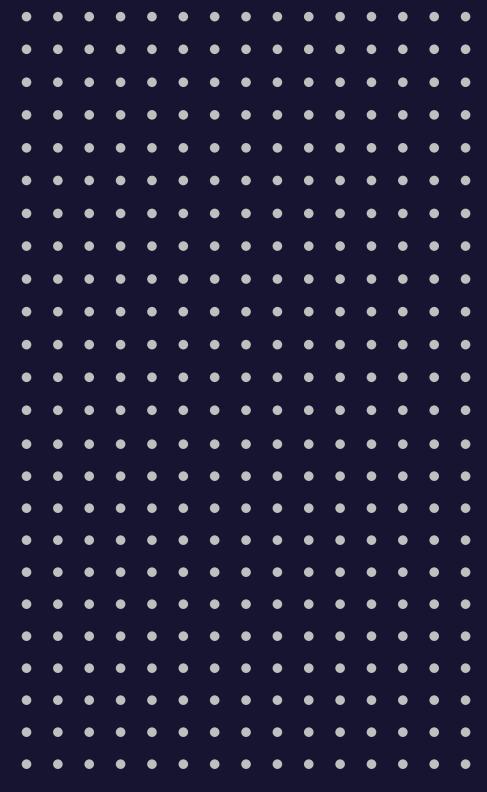


# codebase



# Modificar codebase

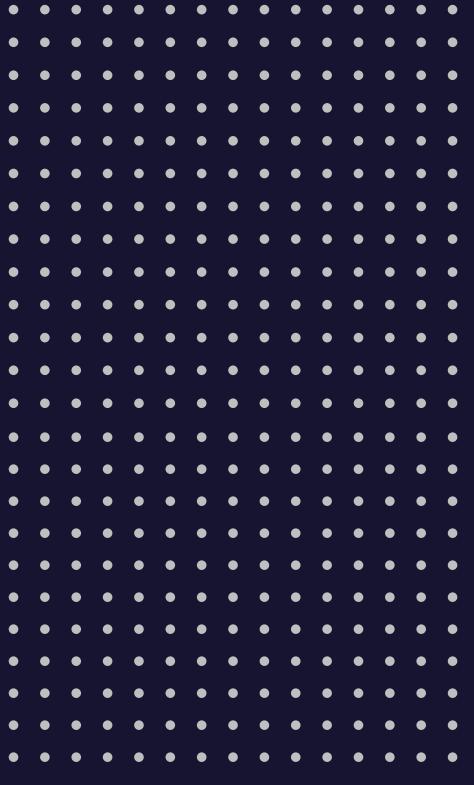
# Modificar terreno



# terraformadores



Validar es comparar  
hipótesis con la  
realidad



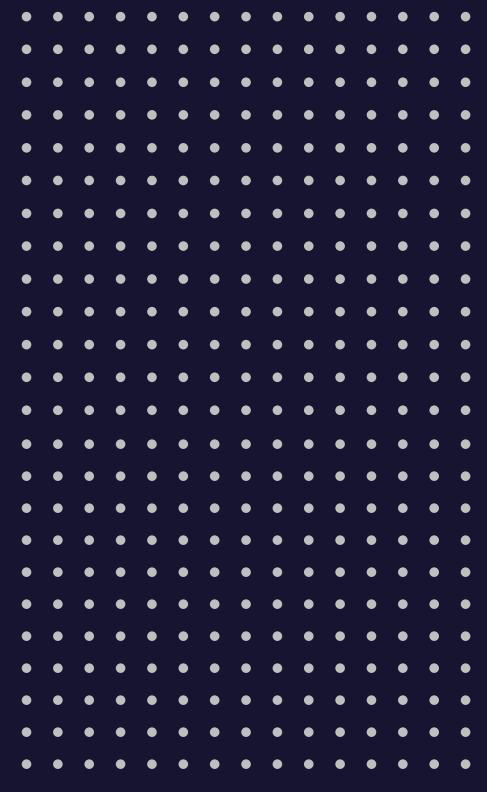
# Expectativa

vs

# Realidad



# linter



# linter --fix

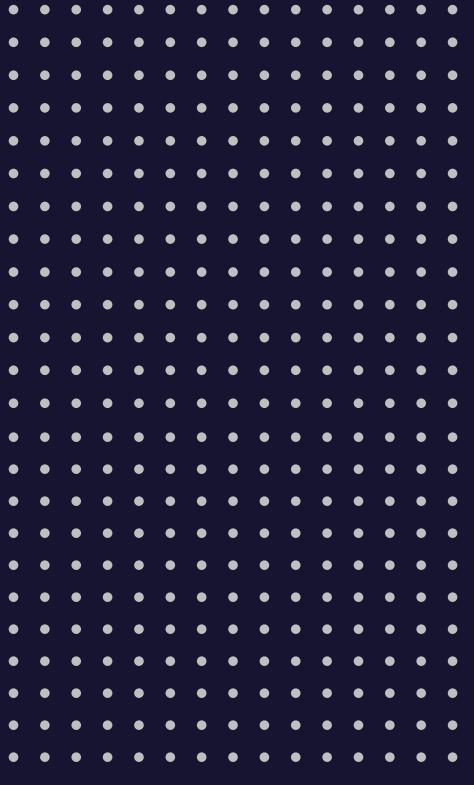
terraformadores y validadores

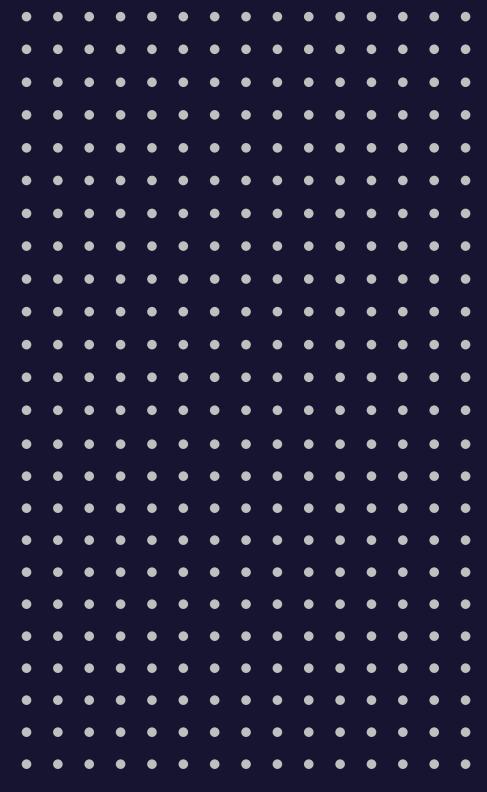
code-gen

# Generación de código



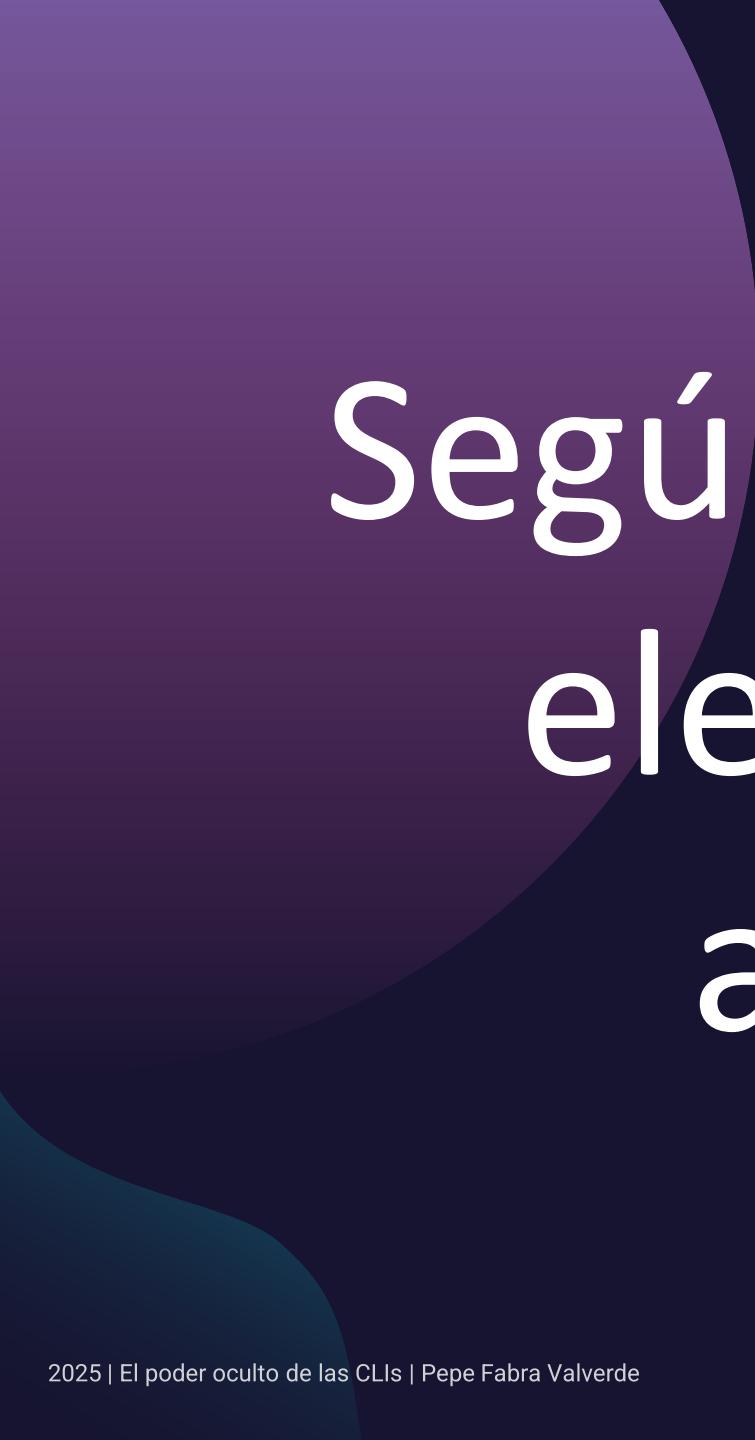
Pero siguiendo X  
estándares



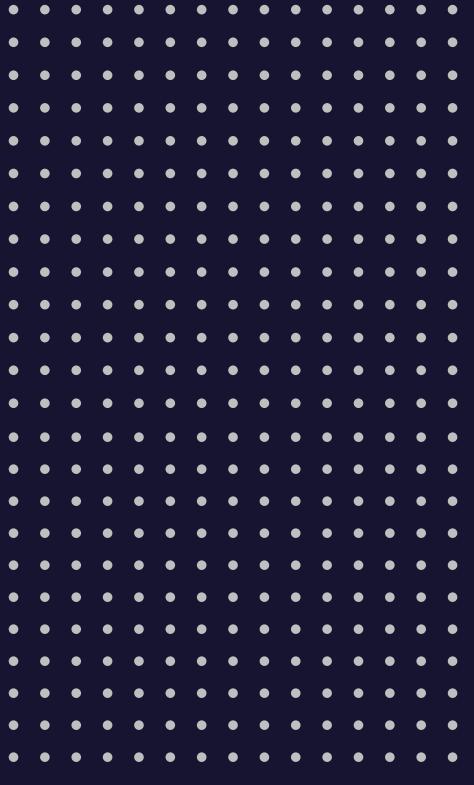


# ¿cómo?

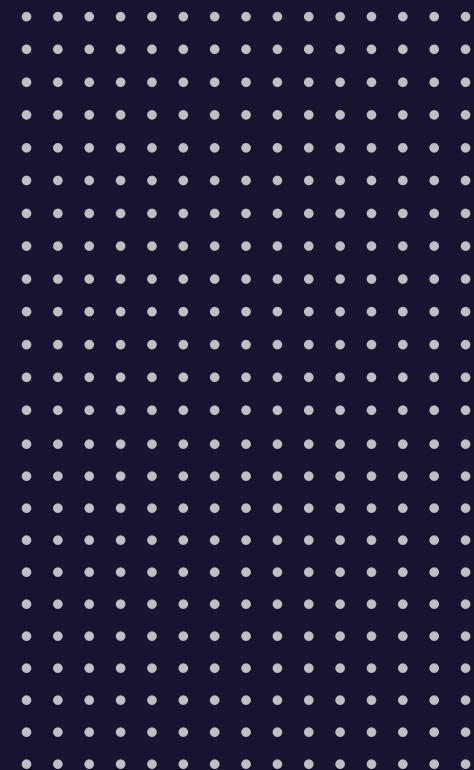
# Usando templates



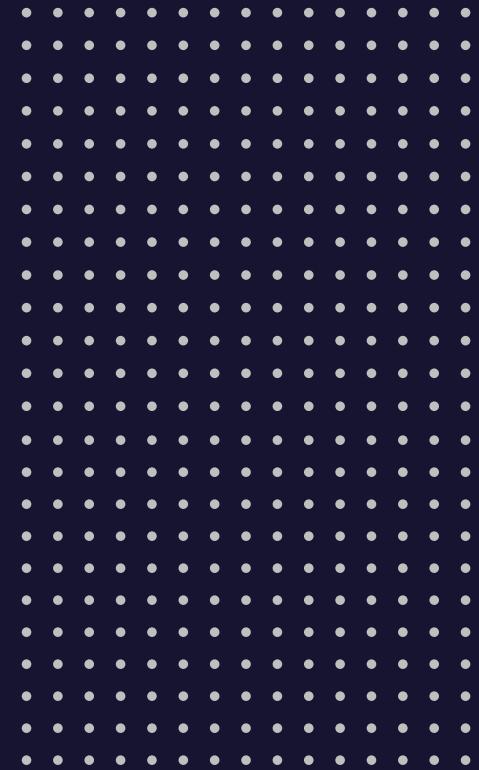
Según el proyecto  
elegiremos el  
approach



# code-gen y fitness functions



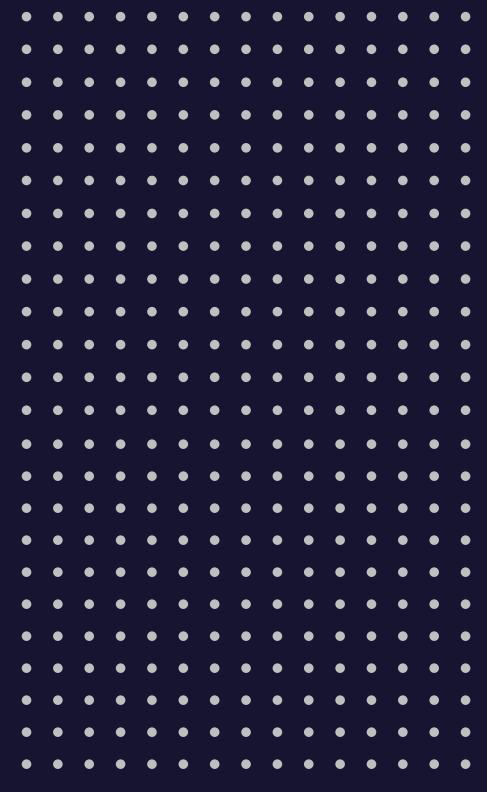
# Abstract Syntax Tree



terraformadores y validadores

# templates

# Guía de cómo se verá el código



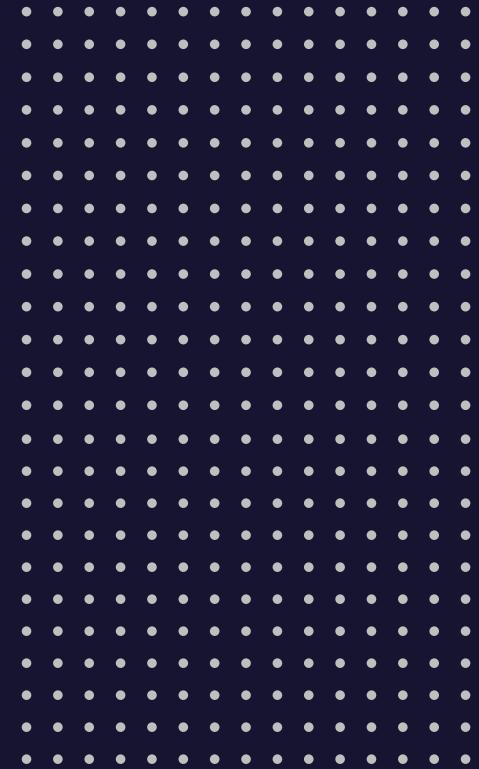
# Es más intuitivo

# Casing como placeholders

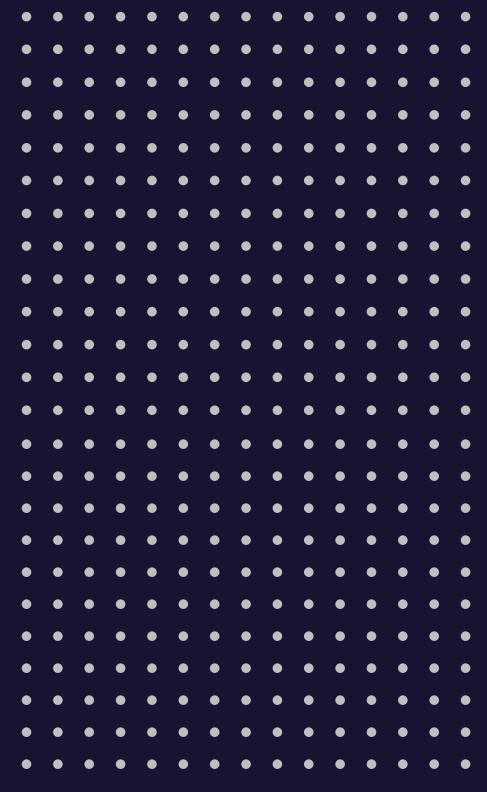
# No hardcodear

# ¿qué es hardcodear?

# Poner el valor a mano



# Usar configuración o argumentos



# Templates también para el pathing

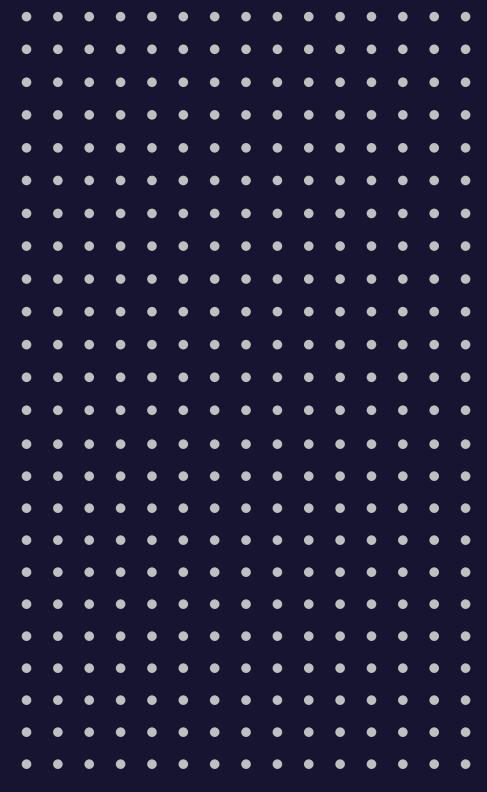
terraformadores y validadores

casing



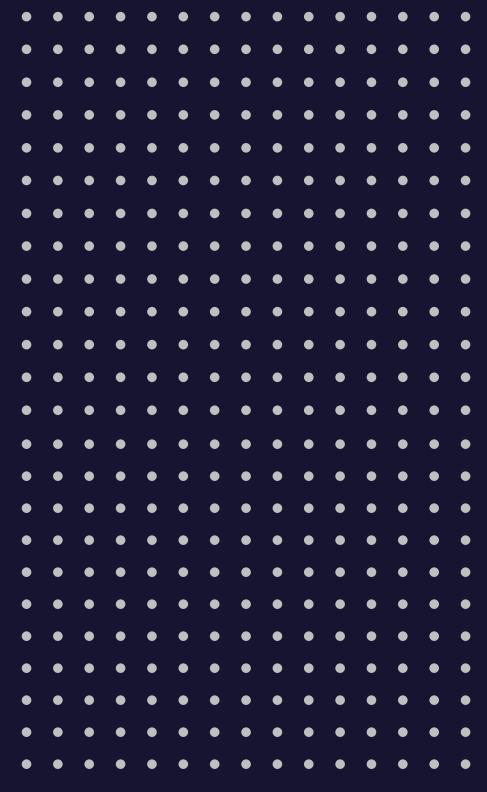


¿os suena alguno?



# camelCase

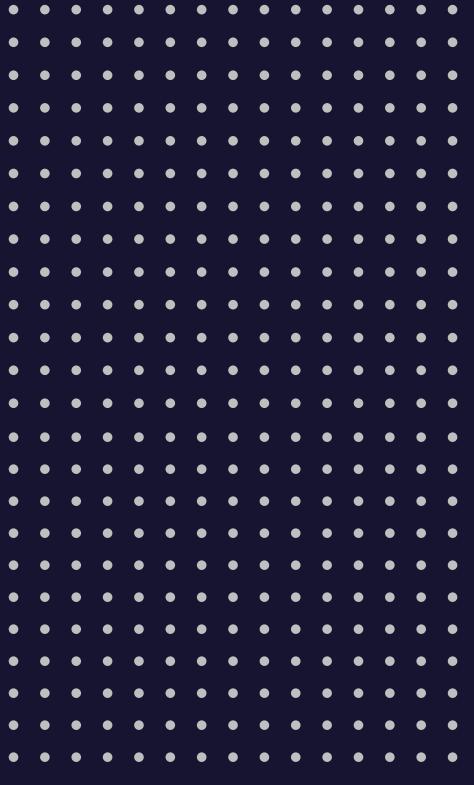
# PascalCase



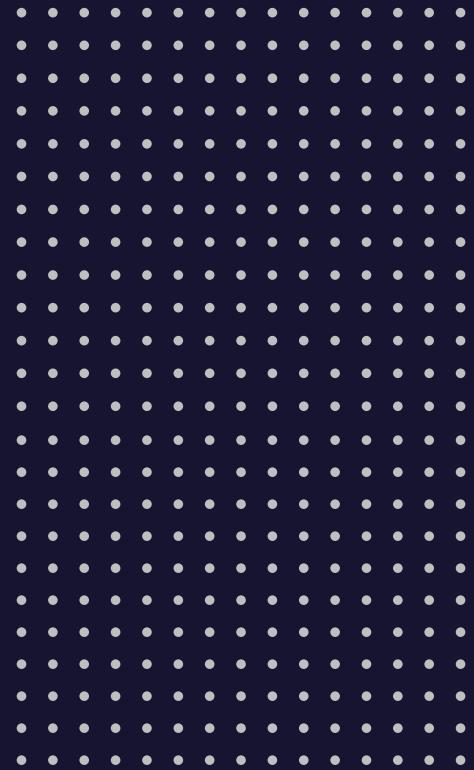
# serpent\_case



# kebab-case



# space case



# UPPER\_CASE

terraformadores y validadores

# terraformador

# code-gen

# Usa templates

# Reestructuraciones

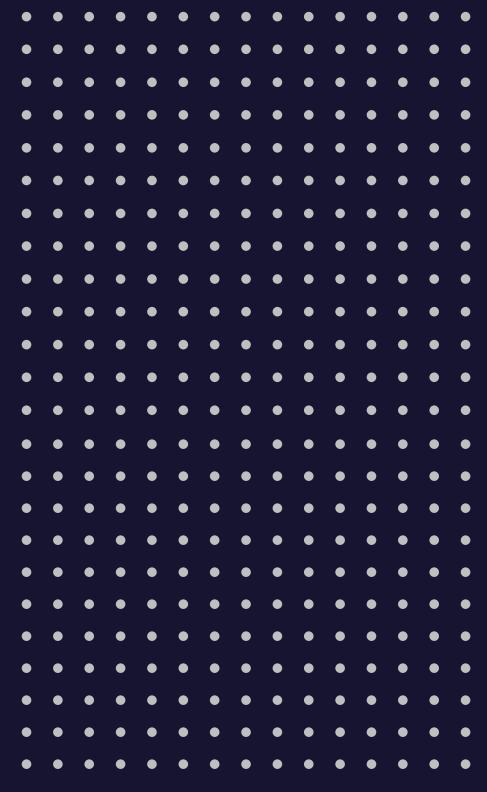
# Templates y/o ADRs con Fitness Functions

# Extensiones de código existente

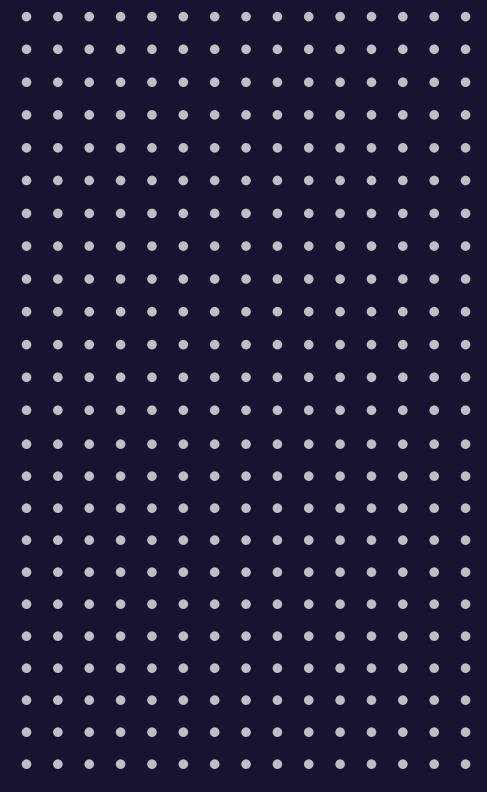
# ADRs con Fitness Functions



Los terraformadores  
cobran sentido con  
volúmenes masivos



# prettier y eslint



# Terraforman el código para adherirlo a reglas

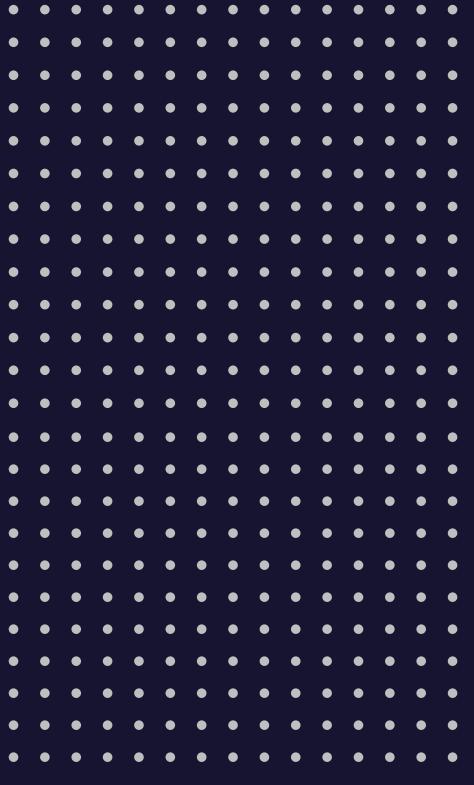
terraformadores y validadores

# validadores

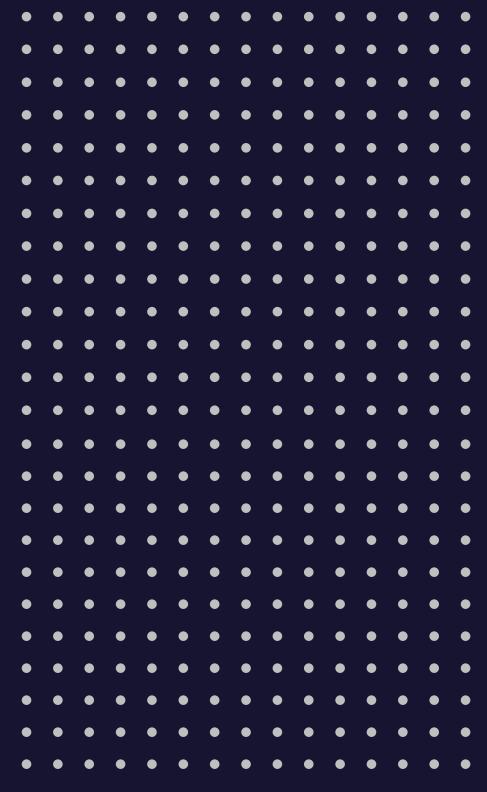
# Policía de las reglas del código

# No sólo el estilo del código

Ej. comprueba los  
nullPointerException



# Usa terraformadores para el diffing



# diffing

# Terraformador(contenido)

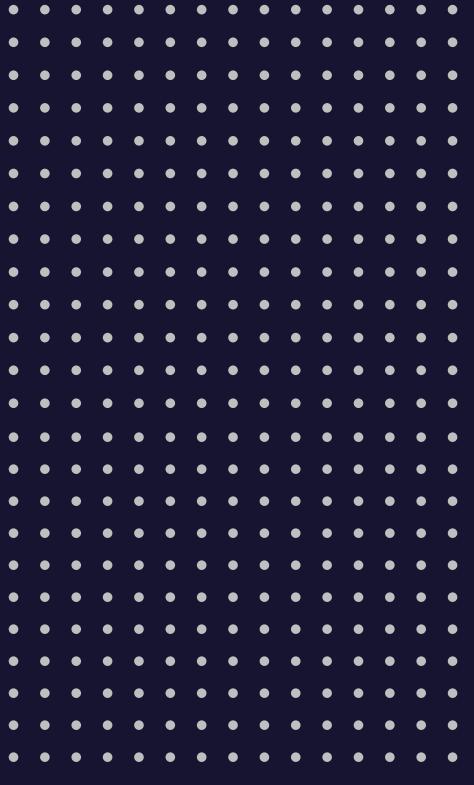
==

## contenido

# Comunicación abierta y honesta

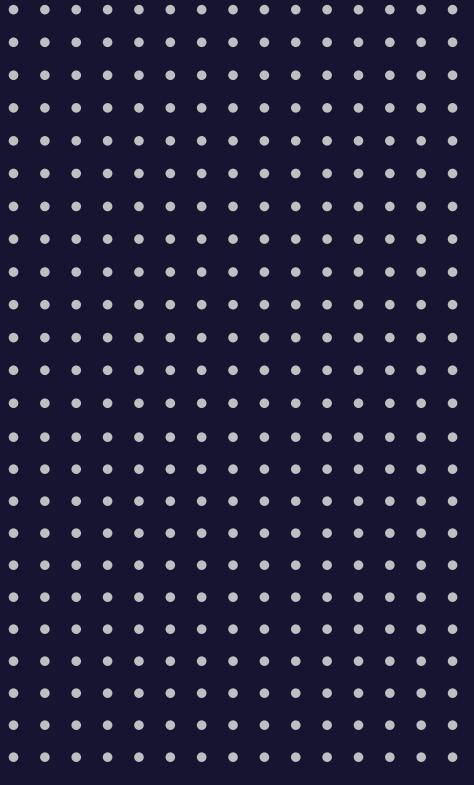


# Dónde has fallado



# Cómo arreglarlo

# O la guía para arreglarlo



# eslint es un validador

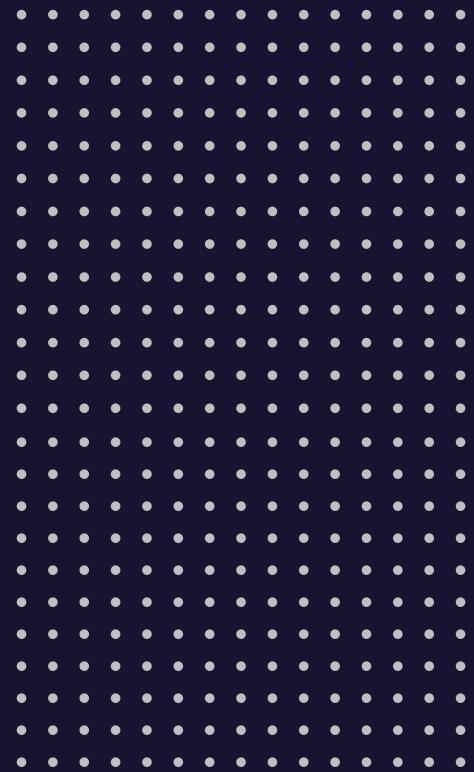
terraformadores y validadores

devops

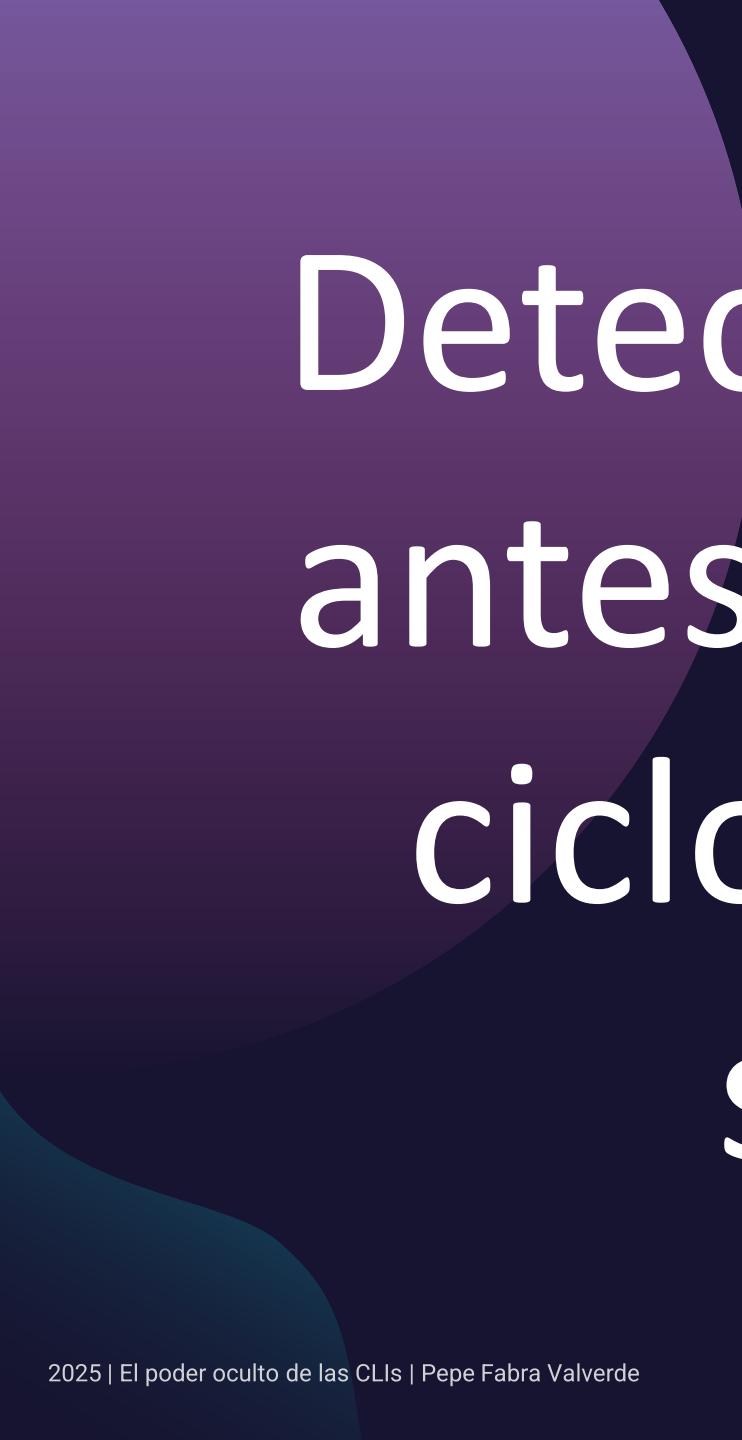
# Validadores en pipelines

# DevOps y Git Hooks

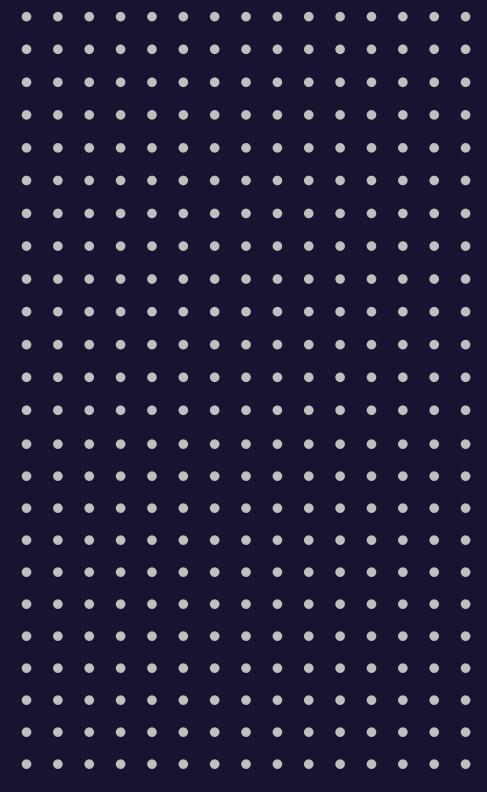
# CI/CD



# Left shifting

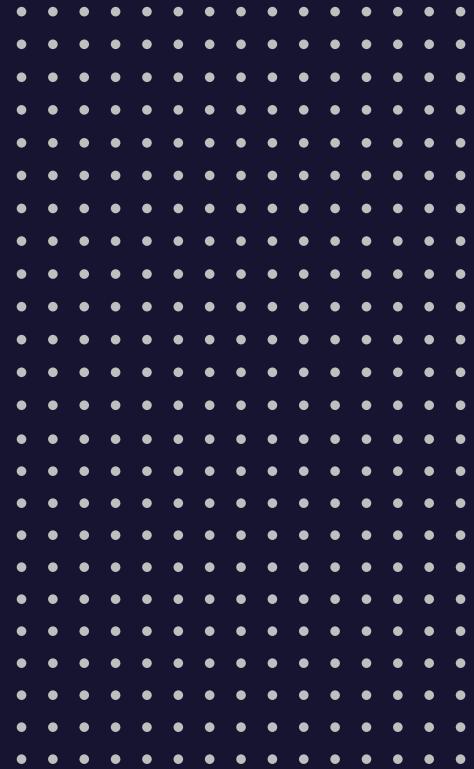


Detectar errores lo  
antes posible en el  
ciclo de vida del  
software



# Cultura de equipo y KTP

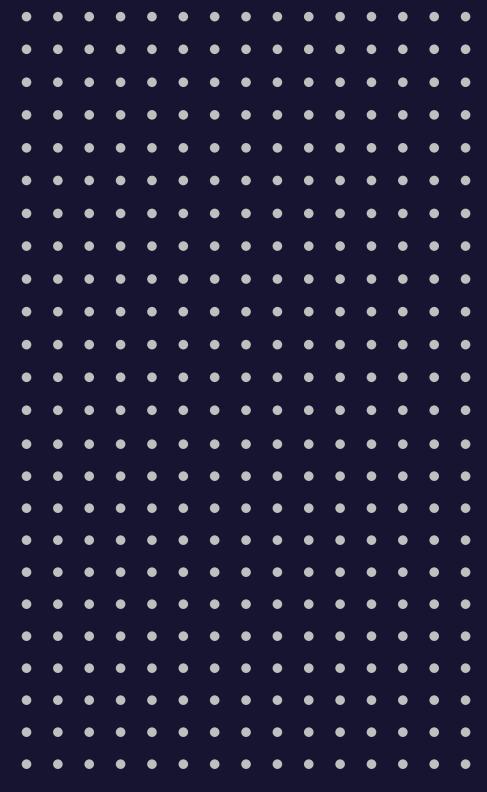
# Knowledge Transfer Partnership



# CLIs en la cultura del equipo



# Son un test más



el poder oculto de las clis

recapitulemos



# Command-line interface

## Automatizaciones

## Developer Experience

## Terraformadores y validadores

Command-line interface  
**Automatizaciones**  
Developer Experience  
Terraformadores y validadores

Command-line interface  
Automatizaciones  
**Developer Experience**  
Terraformadores y validadores

Command-line interface  
Automatizaciones  
Developer Experience  
**Terraformadores y validadores**

# Bibliografía

- Node CLI book
- [github.com/jofaval/node-cli/](https://github.com/jofaval/node-cli/)

# Créditos

- <https://poweredtemplate.com/free-dark-gradient-presentation-template-141850>

# El poder oculto de las CLIs

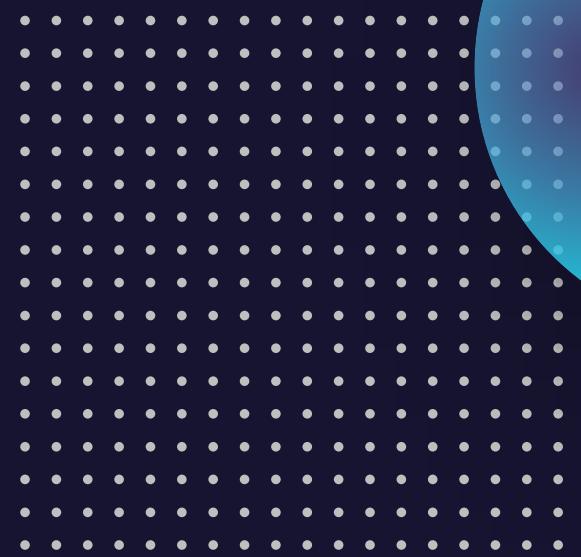
Catedra UV x Capgemini | Pepe Fabra Valverde | 2024

# Preguntas

# Descanso

# Taller

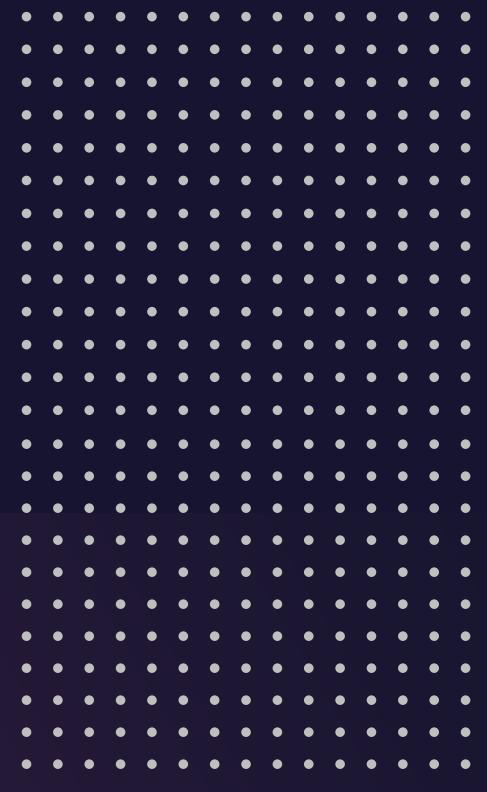
*Crea tu propia CLI*



# magic-cli

- node o java, lo que prefieras
- rellena los huecos

<https://github.com/jofaval/talks-about/tree/master/uv/el-poder-oculto-de-las-clis>



# magic-cli

- node o java, lo que prefieras
- rellena los huecos

[github.com/jofaval/the-hidden-power-of-clis](https://github.com/jofaval/the-hidden-power-of-clis)

main ▾ 1 Branch Tags

Go to file

Code ▾

jofaval

18 Commits

java

node

CHANGELOG.md

README.md

THE\_MORE\_YOU\_KNOW.md

README

# El poder oculto de las CLIs

Taller práctico para la charla de "El poder oculto de las CLIs"

Para saber más contexto de las CLIs y estándares, consulta la sección de [Saber más](#)

## Tabla de Contenido

## CHANGELOG

Registrado en [CHANGELOG.md](#).

## Lenguajes

Java o Node (TypeScript)

## Tiempo

## About

Introductory Workshop to CLIs, best practices, terraforming and validations on the codebase

java cli node typescript maven  
linting validator terraforming

Readme

Activity

1 star

1 watching

0 forks

Report repository

## Releases

No releases published

## Packages

No packages published

## Languages

TypeScript