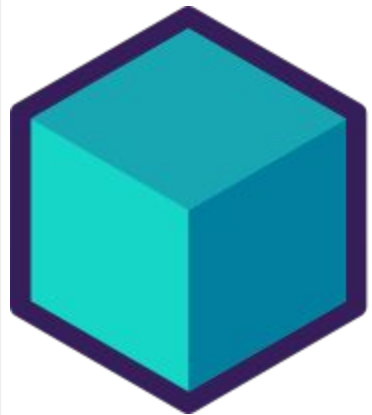


Persistence Tests & Testcontainers

No more excuses



Testcontainers

Jesús María Villar Vázquez

- Software Engineer @ [Celonis](#)
- [Agile & XP Advocate](#)
- Testing Advocate



@geeksusma

GitHub:

<https://github.com/geeksusma>

Twitter:

<https://twitter.com/geeksusma>

Linkedin:

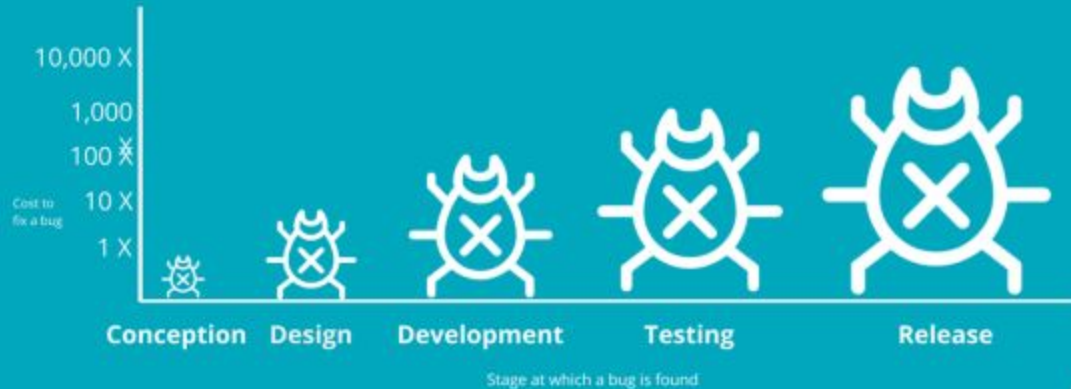
<https://www.linkedin.com/in/geeksusma/>

Por qué son importantes los Tests?

- No nos gustan los bugs
- Queremos detectar bugs temprano
 - Mejor durante el desarrollo
- Porque es más divertido hacer cosas nuevas que corrigiendo “otras cosas”



Resolving bugs early and often reduces associated costs



RAYGUN

<https://azevedorafaela.com/2018/04/27/what-is-the-cost-of-a-bug/>
<https://javi-kata.medium.com/ci-cd-for-dummies-b45ed917f9dc>

Qué es un buen test? FIRST PRINCIPLES

F (fasts)

I (isolated)

R (repeatable)

S (self Validating)

T (thorough)

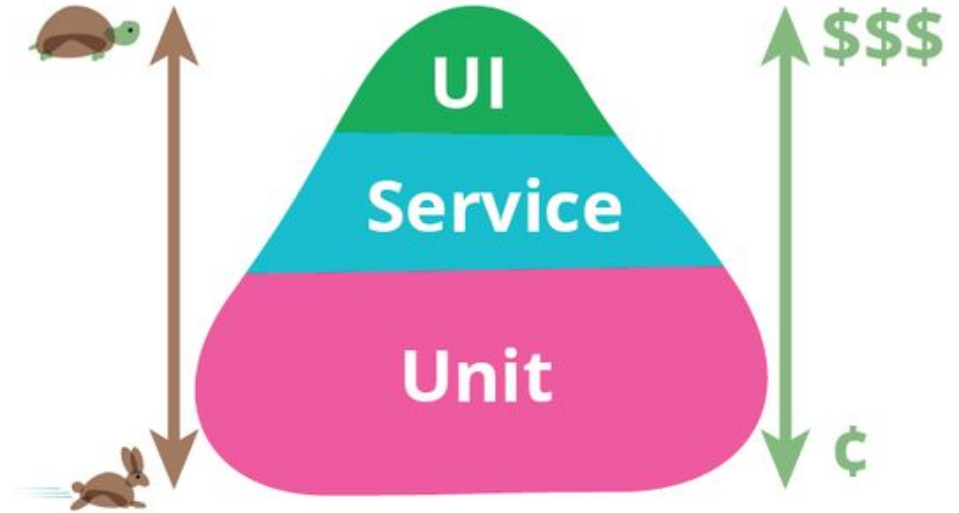


Copyright: Eva Lozano.



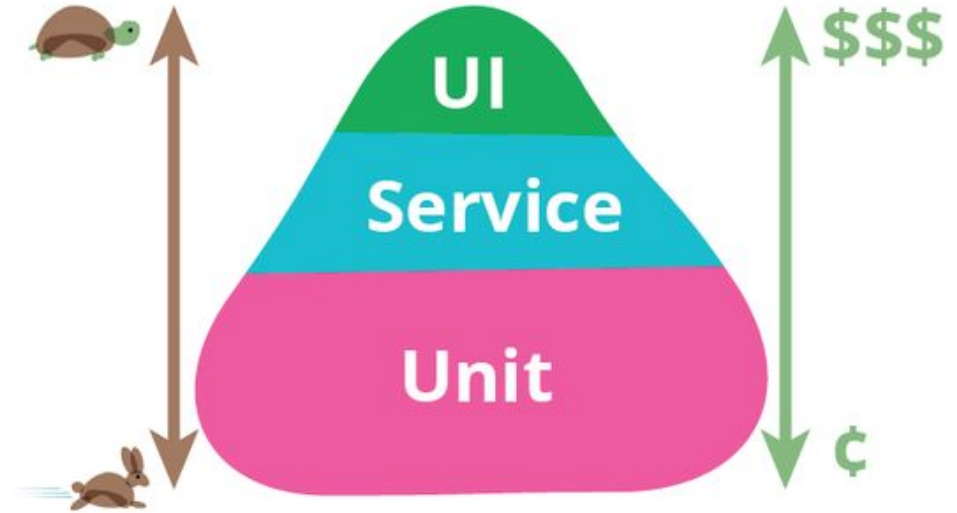
Testing Pyramid

- [Martin Fowler](#)
 - Mientras más lentos, más costosos
 - Mientras más rápidos, más baratos
 - Unit Test vs “Other Tests”



Mi Pirámide (Por Defecto)

- UI
 - Acceptance Tests
 - end2end
- Service
 - In (API Tests)
 - Out (Persistence Tests)
- Unitarios
 - Domain Layer



Y es garantía de cero Bugs?



Por qué hacer Persistence Tests?

Practical Testing Pyramid

- SQL Statements en tiempo de ejecución
- No todo el contexto, sólo la capa de persistencia
- No más todo “arriba” para ajustar una query
- No es necesario end2end
- Automatización
- Permite un enfoque TDD

Por qué con Testcontainers?

- Testcontainers
 - Java based pero adoptada por otros lenguajes
 - Cloud support
 - Levanta/Destruye Contenedores desde un test
 - Facilita mucho el setup
 - Te ayuda a centrarte en escribir el test y no en la infraestructura del test
 - Test en paralelo
 - Fácilmente pueden correr en un pipeline de CI
 - Puedes tener un entorno lo más parecido posible a un entorno productivo



Por qué NO con otras soluciones?

- docker-compose up/down + bash
 - Poner el foco en la capa de infraestructura y no en el test
- Soluciones embebidas
 - Perfectas para pequeñas aplicaciones
 - No 100% fiables en otros entornos
 - Algunas requieren un “mapeo especial” (PostgreSQL vs H2)
- Vagrant & Virtual Box
 - Pesadas
 - Difíciles de correr en pipelines
- CI Environments
 - Requieren supervisión y limpieza
 - Fáciles de romper (falsos positivos)

Shut up and dance!



<https://github.com/geeksusma/persistence-test>