

Developer Experience

Mavi Jiménez @linkita

Hola 🖐️

Mavi Jiménez @linkita



Cenar Palomitas

Dormir



Las alcachofas

La incertidumbre



Cenar Palomitas



Dormir



Las alcachofas

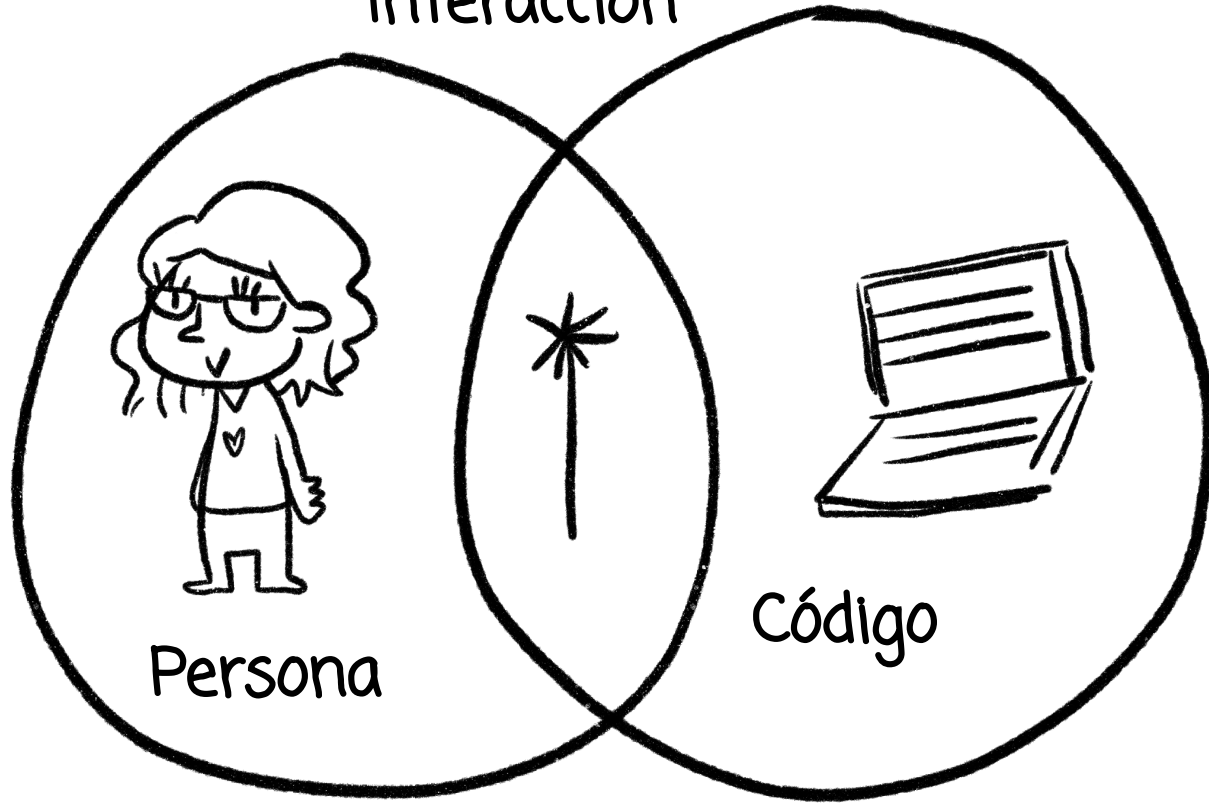


La incertidumbre

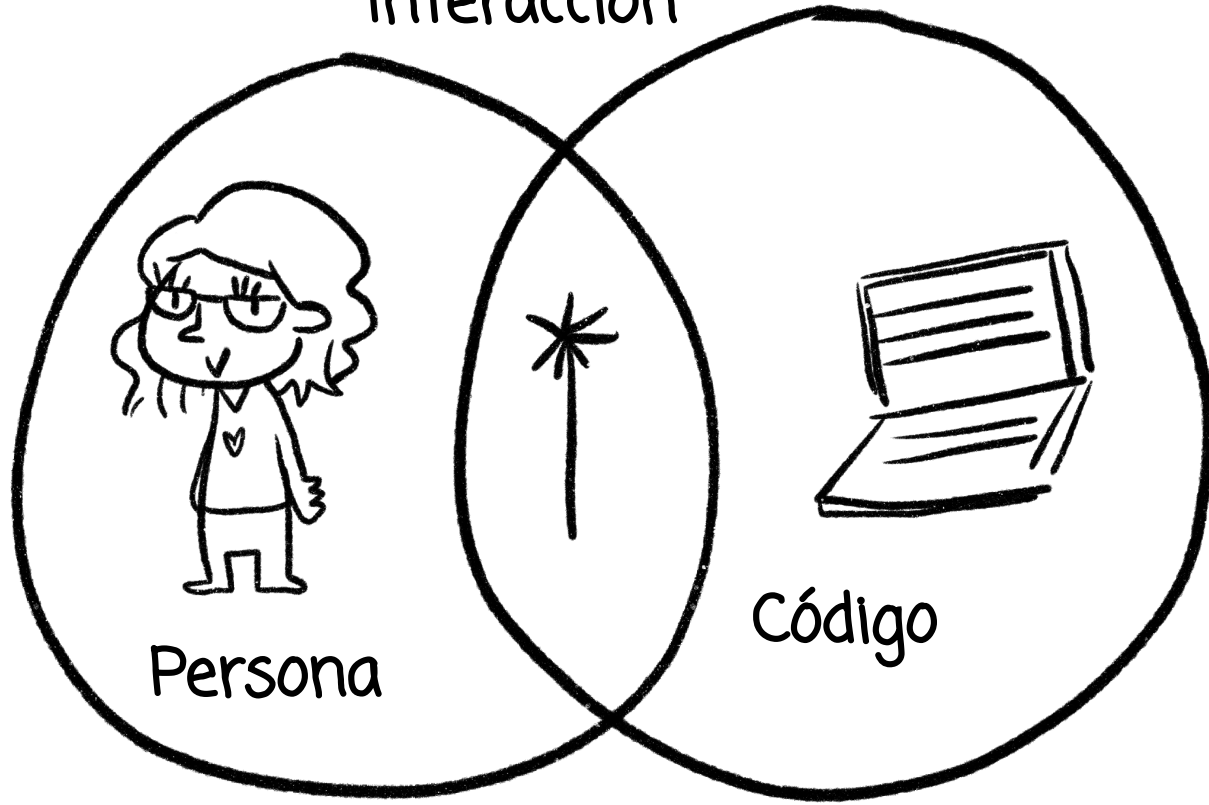


UX

Interacción



Interacción



Personas

Personas

Código

Personas

Interacción

Código

Personas

SESGOS

Personas

SESGOS



Personas

SESGOS

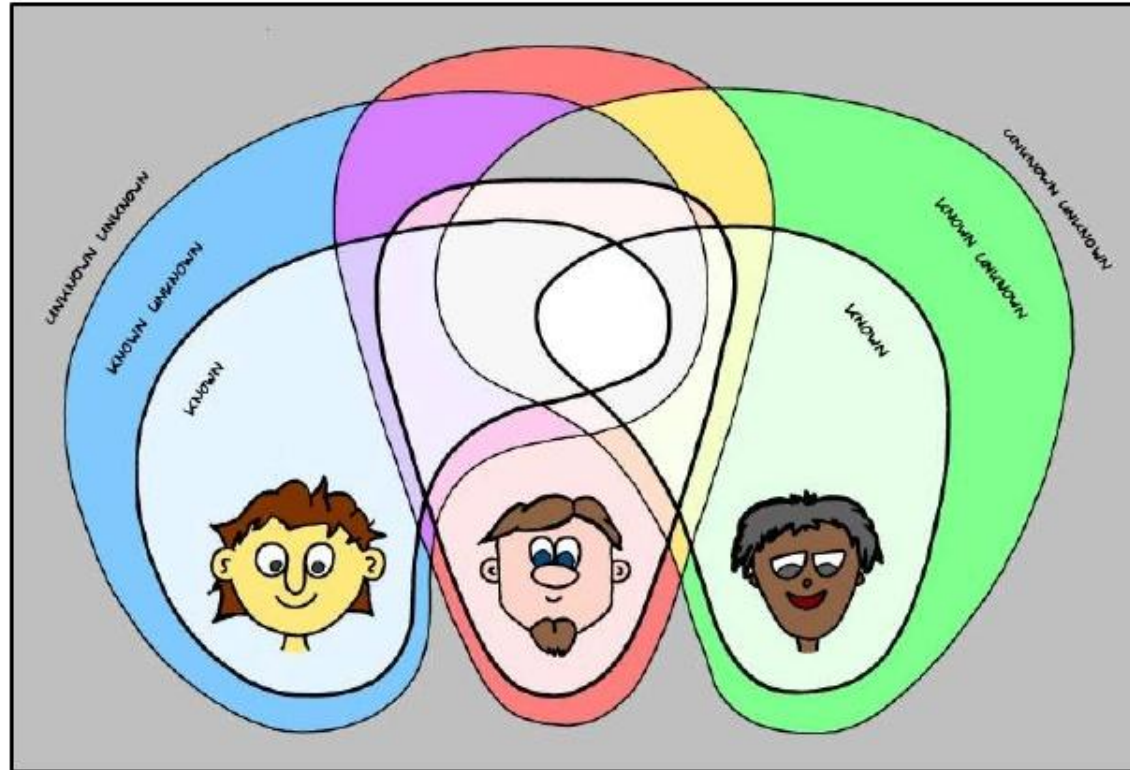
Yo sólo puedo diseñar experiencias para gente como yo misma





@nke_ise

Personas



The unknowns are different per person
That's why people get together and make joint decisions

Personas

EGO

Personas



Personas



Personas



Personas



Personas



¿Cuál es la primera herramienta que puede usar Eva para empezar a programar en el equipo?



Personas

Comunicarse

Personas

Preguntar

Personas

Preguntar reduce la incertidumbre.

Personas



Personas

Lenguaje Ubicuo

Personas

Lenguaje Ubicuo

Personas



Personas

1. Podrá empezar a **identificar** y **explicar** el Dominio los demás:
Developers o Domain Experts.

Personas

1. Podrá empezar a **identificar** y **explicar** el Dominio los demás:
Developers o Domain Experts.
2. Tratar con el dominio e identificar qué **features** son costosas o no.

Personas

1. Podrá empezar a **identificar** y **explicar** el Dominio los demás:
Developers o Domain Experts.
2. Tratar con el dominio e identificar qué **features** son costosas o no.
3. **Elegir sus luchas.**

Personas

1. Podrá empezar a **identificar** y **explicar** el Dominio los demás:
Developers o Domain Experts.
2. Tratar con el dominio e identificar qué **features** son costosas o no.
3. **Elegir sus luchas.**
4. Básicamente hacer **preguntas** sobre el Dominio.

Personas

1. Podrá empezar a **identificar** y **explicar** el Dominio los demás:
Developers o Domain Experts.
2. Tratar con el dominio e identificar qué **features** son costosas o no.
3. **Elegir sus luchas.**
4. Básicamente hacer **preguntas** sobre el Dominio.

Preguntar reduce...

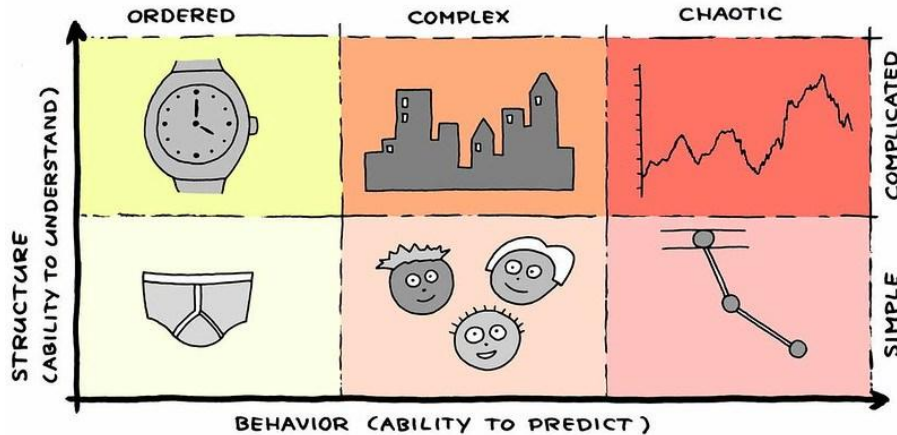
Personas

1. Podrá empezar a **identificar** y **explicar** el Dominio los demás:
Developers o Domain Experts.
2. Tratar con el dominio e identificar qué **features** son costosas o no.
3. **Elegir sus luchas.**
4. Básicamente hacer **preguntas** sobre el Dominio.

Preguntar reduce... la incertidumbre.

Complejidad

Un sistema cuanto más complejo es, más impredecible es



Mantener una estructura simple
(entendible)

Mantener una estructura simple

Código

Patrones de diseño

Mantener una estructura simple

Código

Código semántico

Mantener una estructura simple

Código

Código semántico




```
protected function process(): void
{
    $pendingModifications = $this->modificationRepository->getModificationsInProgress();

    foreach ($pendingModifications as $modification) {
        $modificationCommand = ModificationCommandFactory::build($modification);
        try {
            $this->commandBus->handle($modificationCommand);
        } catch (ORMException $exception) {
            ...
        } catch (Throwable $exception) {
            ...
        }
    }
}
```

```
protected function process(): void
{
    $pendingModifications = $this->modificationRepository->getModificationsInProgress();

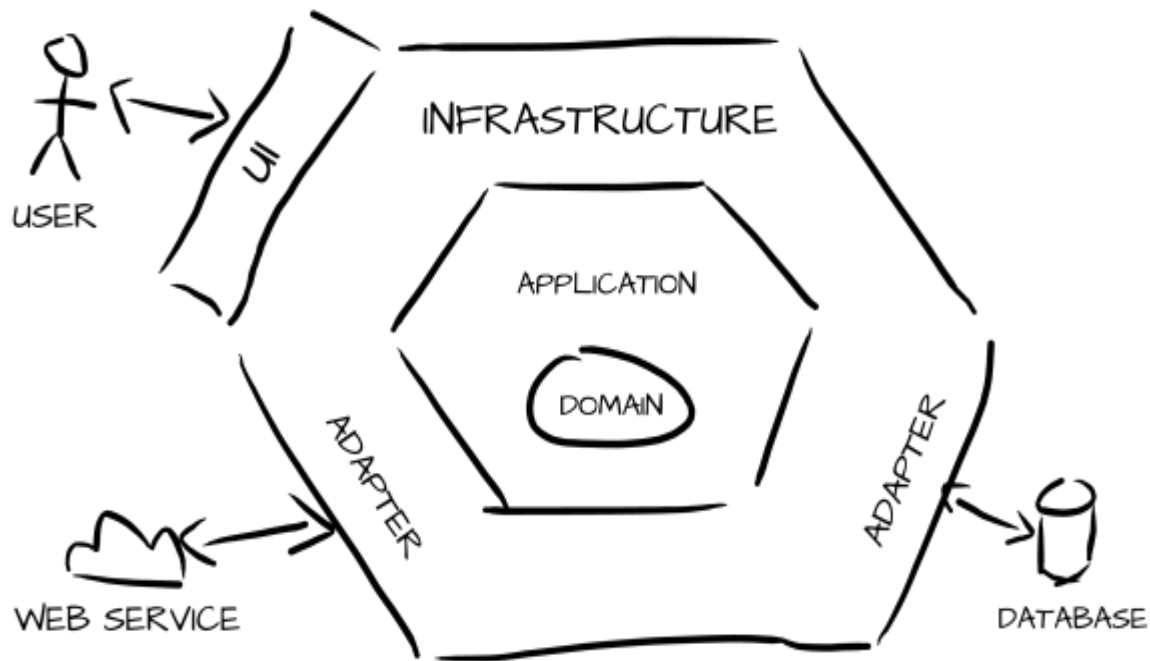
    foreach ($pendingModifications as $modification) {
        $modificationCommand = ModificationCommandFactory::build($modification);
        try {
            $this->commandBus->handle($modificationCommand);
        } catch (ORMException $exception) {
            ...
        } catch (Throwable $exception) {
            ...
        }
    }
}
```

Mantener una estructura simple

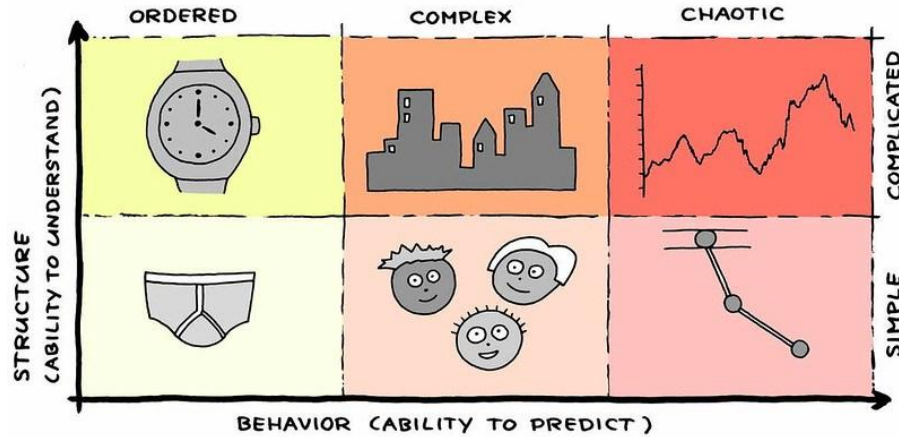
Código

No te inventes tu propio framework

Diseño por capas & Arquitectura Hexagonal



BEHAVIOUR (able to predict)



BEHAVIOUR (able to predict)



TESTS

1. Testea el **100% de tu dominio**. Tu core del negocio tiene que estar testeado, sin excusas.
2. Testea el **100% de la capa de aplicación**.
3. DISEÑA TU CÓDIGO PARA SER TESTEADO

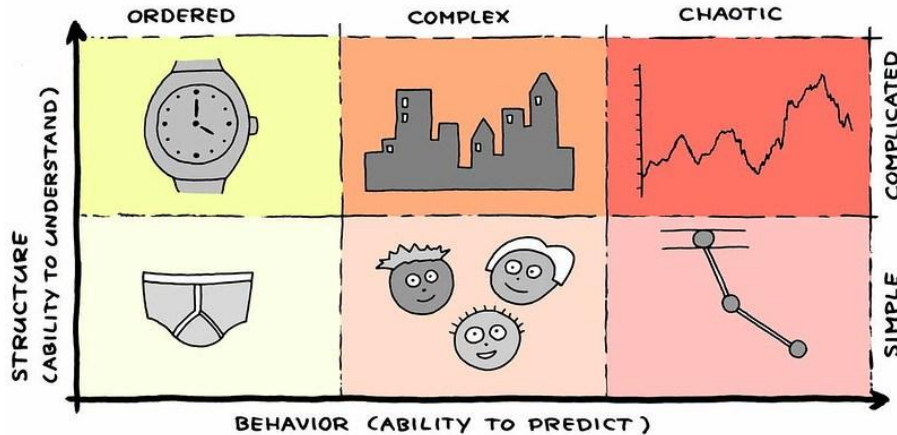


TESTS AUTOMÁTICOS

1. Testea el **100% de tu dominio**. Tu core del negocio tiene que estar testeado, sin excusas.
2. Testea el **100% de la capa de aplicación**.
3. DISEÑA TU CÓDIGO PARA SER TESTEADO

Complejidad

Un sistema cuanto más complejo es, más impredecible es



Personas

- Diversidad
- Lenguaje Ubicuo

Código

Estructura

- Coherencia de código
- Patrones de diseño
- Arquitectura por capas
- Ports & Adapters

Comportamiento

- Tests

Interacción

Interacción

Tabs vs. Spaces

Interacción

Automatización. Otra vez.

(sobre todo las cosas aburridas)

Interacción

Automatización del Despliegue

Interacción

Automatización del Despliegue



Interacción

Fallar rápido

Interacción

"Ever tried. Ever failed. No matter. Try again.
Fail again. Fail better."

Sammuel Better

Interacción

Postmortem sin culpables

Interacción

Postmortem sin culpables

holaluz / **necronomicon** Private

Unwatch 14

★ Star 0

Fork 0

Code

Issues 29

Pull requests 5

Actions

Projects 1

Wiki

Security

Insights

Settings

Branch: master

necronomicon / post-mortems /

Create new file

Upload files

Find file

History



paulajulve Add post-mortem for core downtime due to migrations lock

Latest commit 7793ce4 29 days ago

..



img

2 months ago



clients-zone-login-migration-28-1... Add analisys

11 months ago



core-downtime-04-10-2019-lock... Add post-mortem for core downtime due to migrations lock

3 days ago

Title

Date

> YYYY/MM/DD

Authors

> list of authors who written the postmortem

Status

> Complete or action items in progress

Summary

> One liner explaining the issue. For example: System X went down because of Y

Detection

> Manual detected? Monitoring detection?

Impact

> Persons affected, revenue lost, processes.

Root Causes / Contributing Conditions

> What were the causes that contributed to the end failure. Example: lack of monitoring, known bug, misconfiguration, lack of tests, lack of security

Resolution and recovery

> How the issue was solved, and how we recovered to normal status

Corrective and Preventative Measures

> After issue fixes or automation to prevent from happening again

Lessons Learned

What went well

> Monitoring warn us, autoscaling auto-recovered, etc..

What went wrong

> Autodeploy didn't worked, alerting failed, etc.

Where we got lucky

> We had a backup we didn't know, the issue happened on non-working hours, etc.

Action items

> List of actions to be taken afterwards

Timeline

Time	Description	Status	
-----	-----	-----	
14:51	Explained event	OK / System impaired / Down / whatever	

Interacción

Have fun

Interacción

Have fun



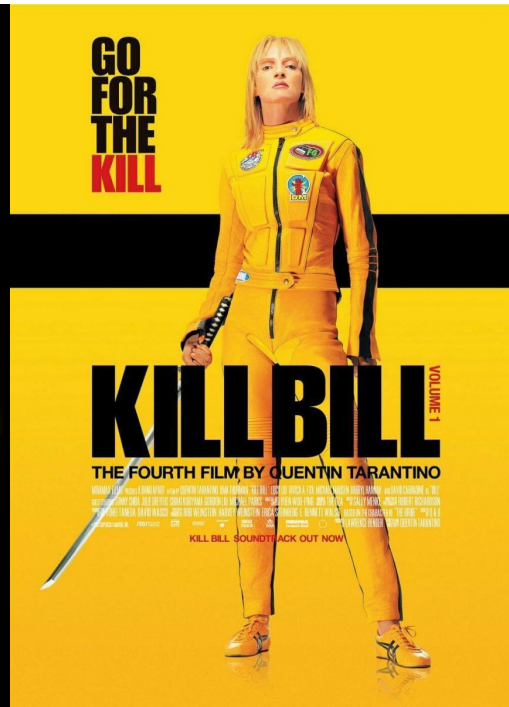
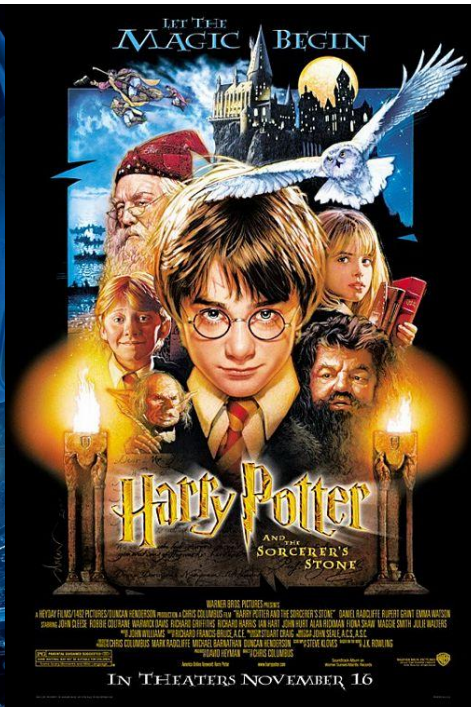
Interacción

Have fun



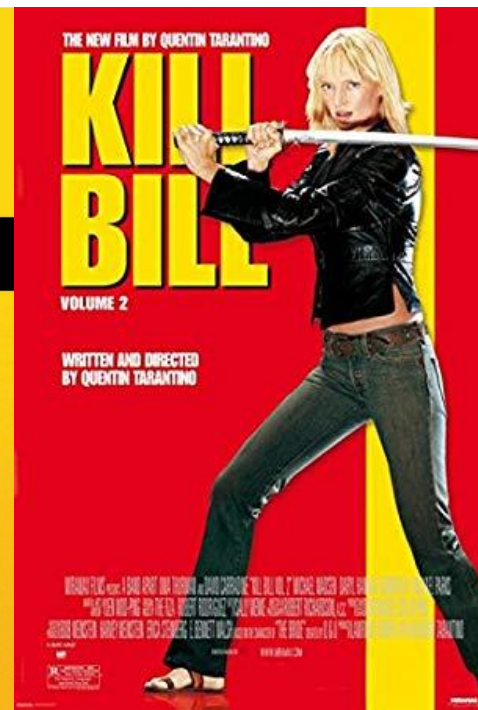
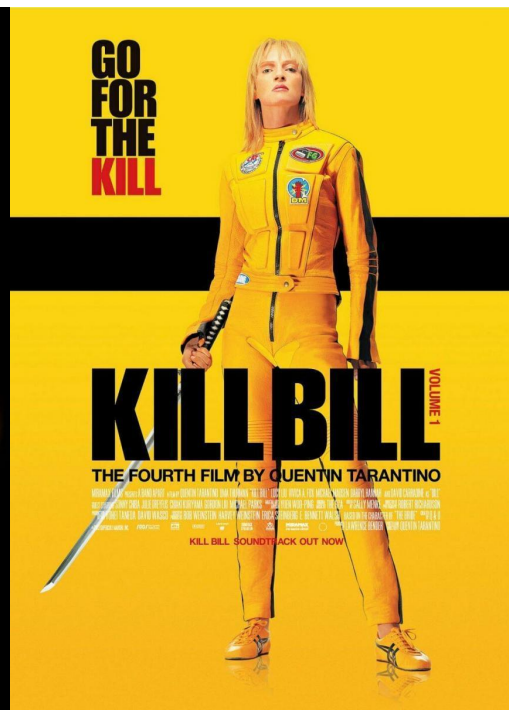
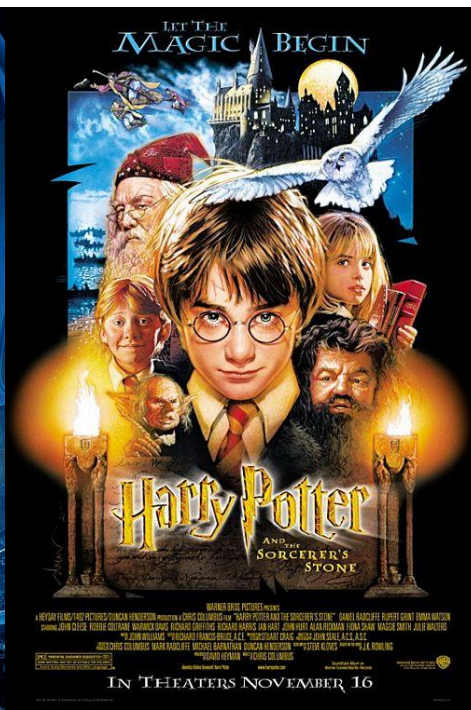
Interacción

Have fun



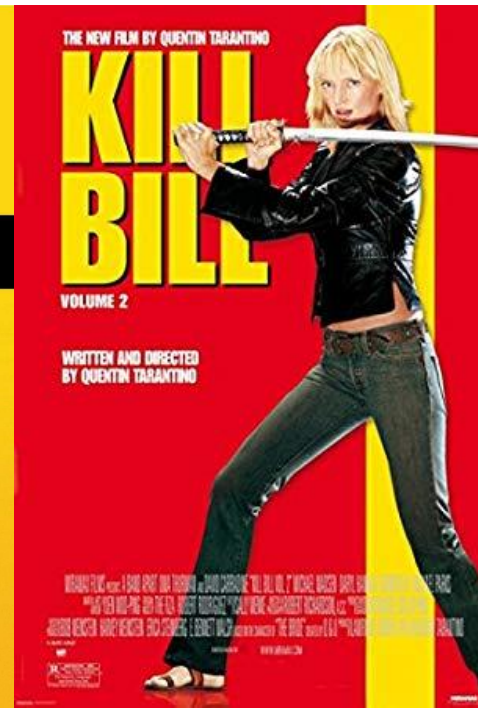
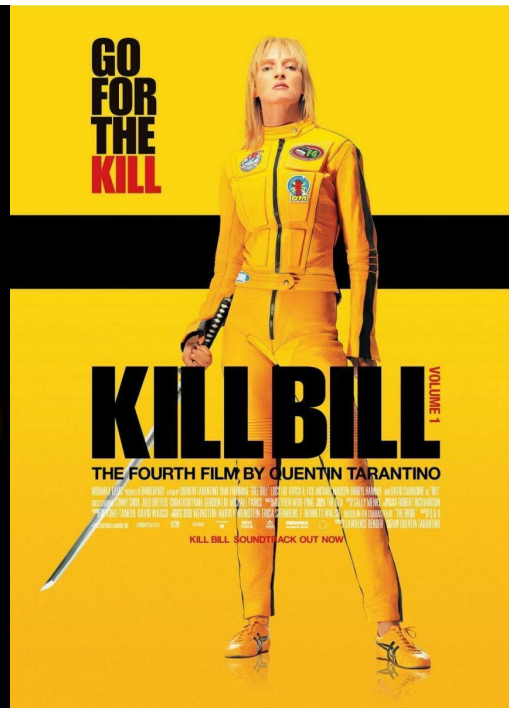
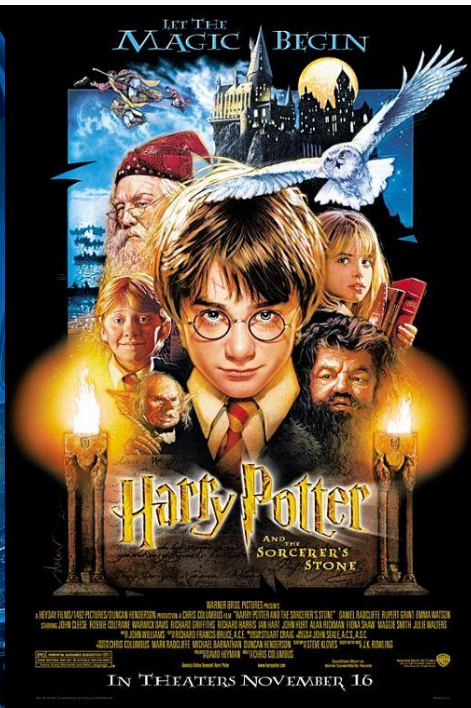
Interacción

Have fun



Interacción

Have fun



Interacción

Have fun

DataNormalizerApp

MailSender

InvoicePrinter

FutureNewInvoicePrinter

Personas

- Diversidad
- Lenguaje ubicuo

Interacción

- Automatización
- Postmortems
- Have fun

Código

Estructura

- Patrones de diseño
- Arquitectura por capas
- Ports & Adapters

Comportamiento

- Tests

Developer Experience



