

DESARROLLO GUIADO POR PRUEBAS



CESIÓN DE ESPACIO:

errant
espacio cowork

PATROCINA:



SI2
SOLUCIONES

Alejandro Hernández (@aleherse)



Desarrollador de aplicaciones web, consultor y formador.

Cuando no trabajo para un cliente desarrollo un bot de Telegram para gestionar clubs de juegos de mesa.

+info: <http://about.me/aleherse>

Agile Manifesto

Individuos e interacciones sobre procesos y herramientas

Software funcionando sobre documentación extensiva

Colaboración con el cliente sobre negociación contractual

Respuesta ante el cambio sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

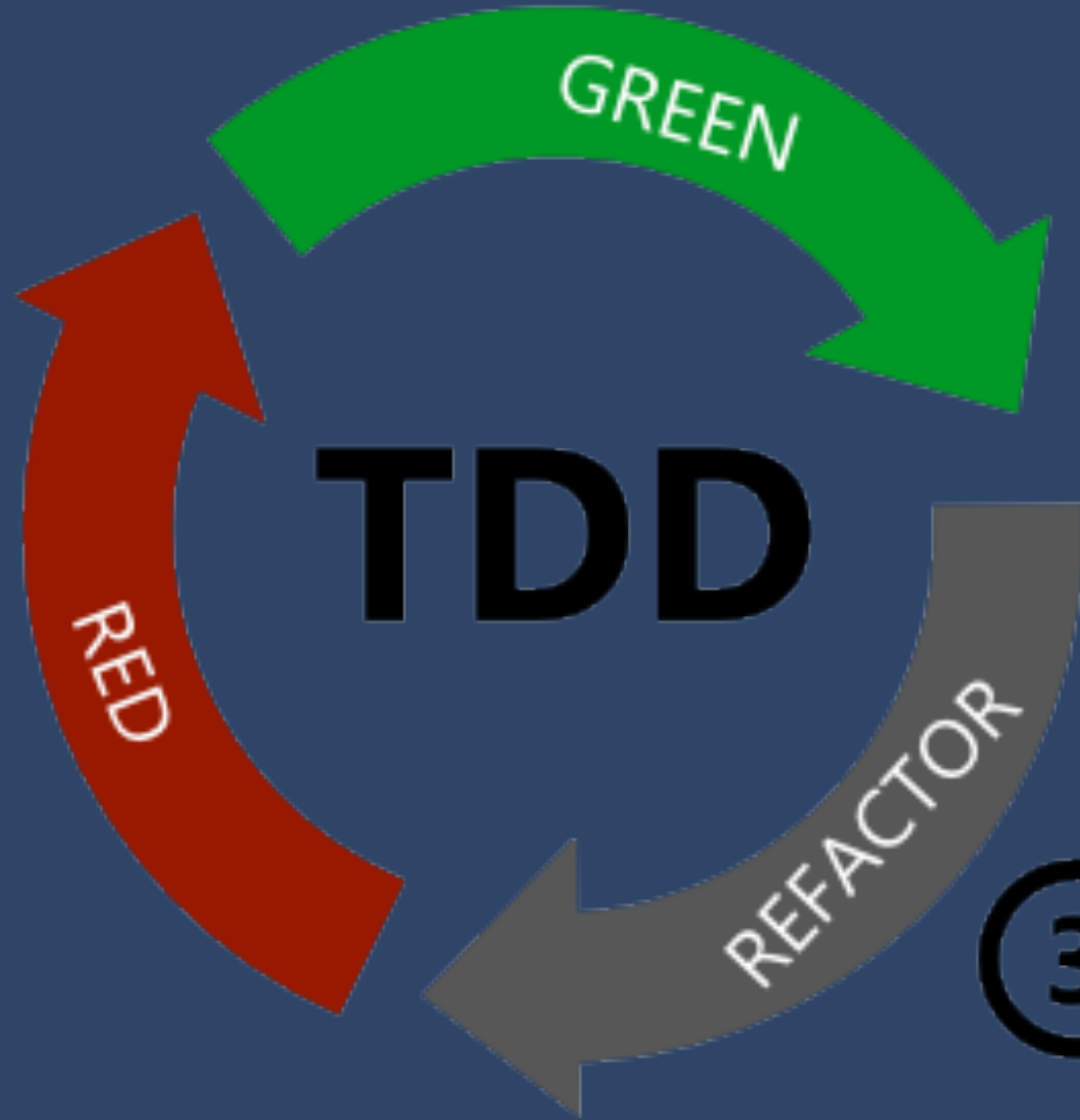
¿Qué es?

Test **D**riven **D**evelopment

Técnica de diseño e implementación de software.

Write a test,
watch it fail.

①



②

Write just enough
code to pass the test.

③

Improve the code without
changing its behavior.

¿Cómo funciona? (fail)

Escribe una prueba (test)

```
function it_converts_from_arabic_1_to_roman_numeral_I()  
{  
    $this->fromArabic(1)->shouldReturn('I');  
}
```

¿Cómo funciona? (fail)

Ejecuta la prueba y mira como falla

```
RomanNumerals
10 ✗ converts from arabic 1 to roman numeral I (173ms)
    expected "I", but got null.

1 specs
1 examples (1 failed)
```

¿Cómo funciona? (pass)

Escribe suficiente código

```
public function fromArabic(int $arabic): string
{
    return 'I';
}
```


¿Cómo funciona? (pass)

Ejecuta la prueba y mira como pasa

```
RomanNumerals
```

```
10  ✓ converts from arabic 1 to roman numeral I (199ms)
```

```
1  specs
```

```
1  examples (1 passed)
```

¿Cómo funciona? (refactor)

Mejora la calidad del código

- Extraer código a un método, clase, colaborador...
- Eliminar repeticiones
- Generalizar la lógica
- Renombrar variables, métodos, clases..

refactoring.guru

Tipos de pruebas

Aceptación

Funcionalidad desde el punto de vista del usuario

Tipos de pruebas

Aceptación

Integración

Nuestro código al interactuar con algo externo

Tipos de pruebas

Aceptación

Integración

Unitarias

Nuestros objetos hacen lo que se espera de ellos

Pilares fundamentales

- Implementar únicamente lo que el cliente necesita

Pilares fundamentales

- Implementar únicamente lo que el cliente necesita
- Minimizar el número de errores que llegan a producción

Pilares fundamentales

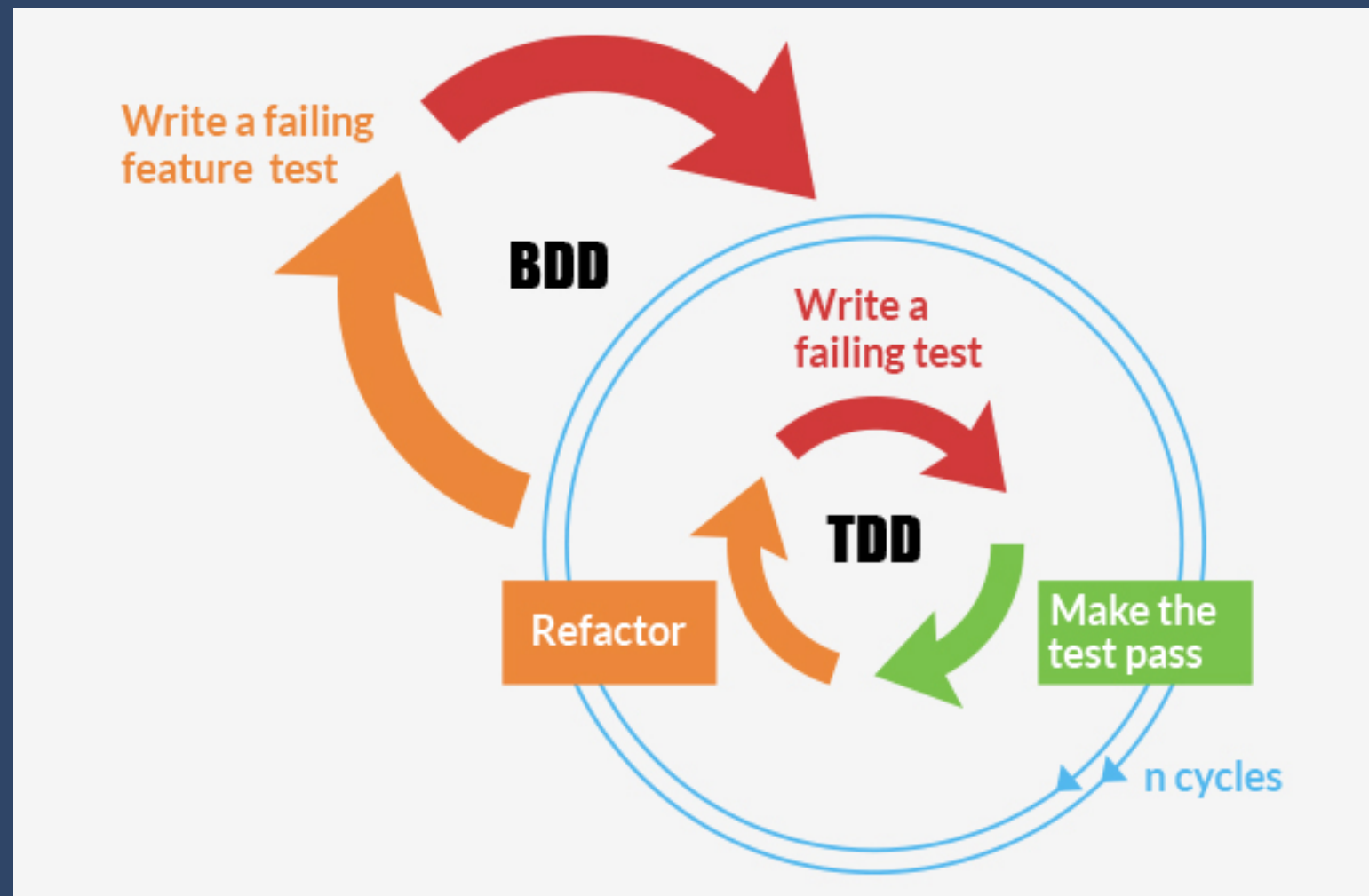
- Implementar únicamente lo que el cliente necesita
- Minimizar el número de errores que llegan a producción
- Implementar código modular, reutilizable y adaptable

Pilares fundamentales

- Implementar únicamente lo que el cliente necesita
- Minimizar el número de errores que llegan a producción
- Implementar código modular, reutilizable y adaptable
- Aumentar la confianza del equipo en el código

**TDD es una
herramienta más**

Desarrollo guiado por comportamiento (BDD)



Desarrollo guiado por comportamiento (BDD)

Scenario: Some determinable business situation

Given some precondition

And some other precondition

When some action by the actor

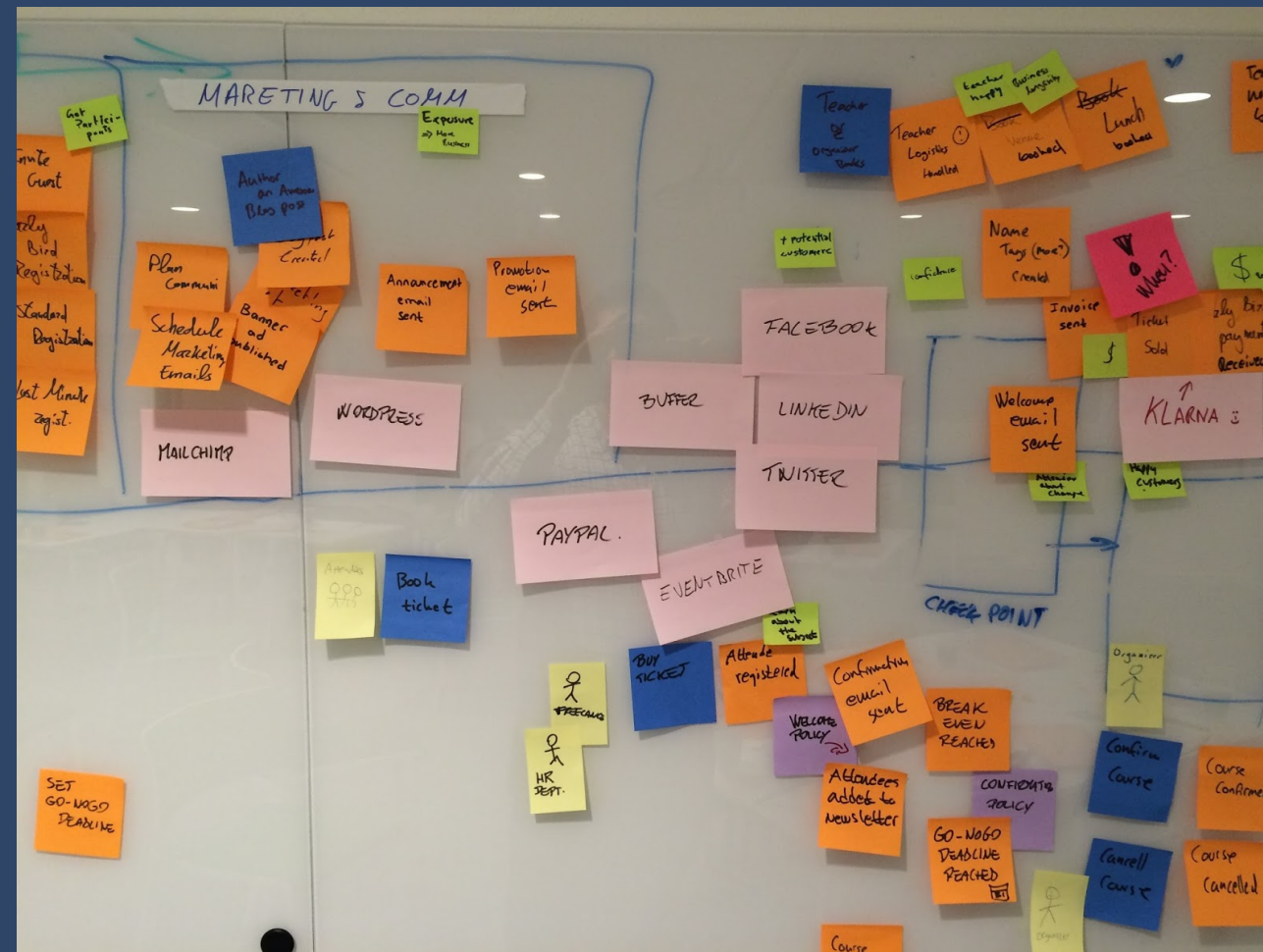
And some other action

And yet another action

Then some testable outcome is achieved

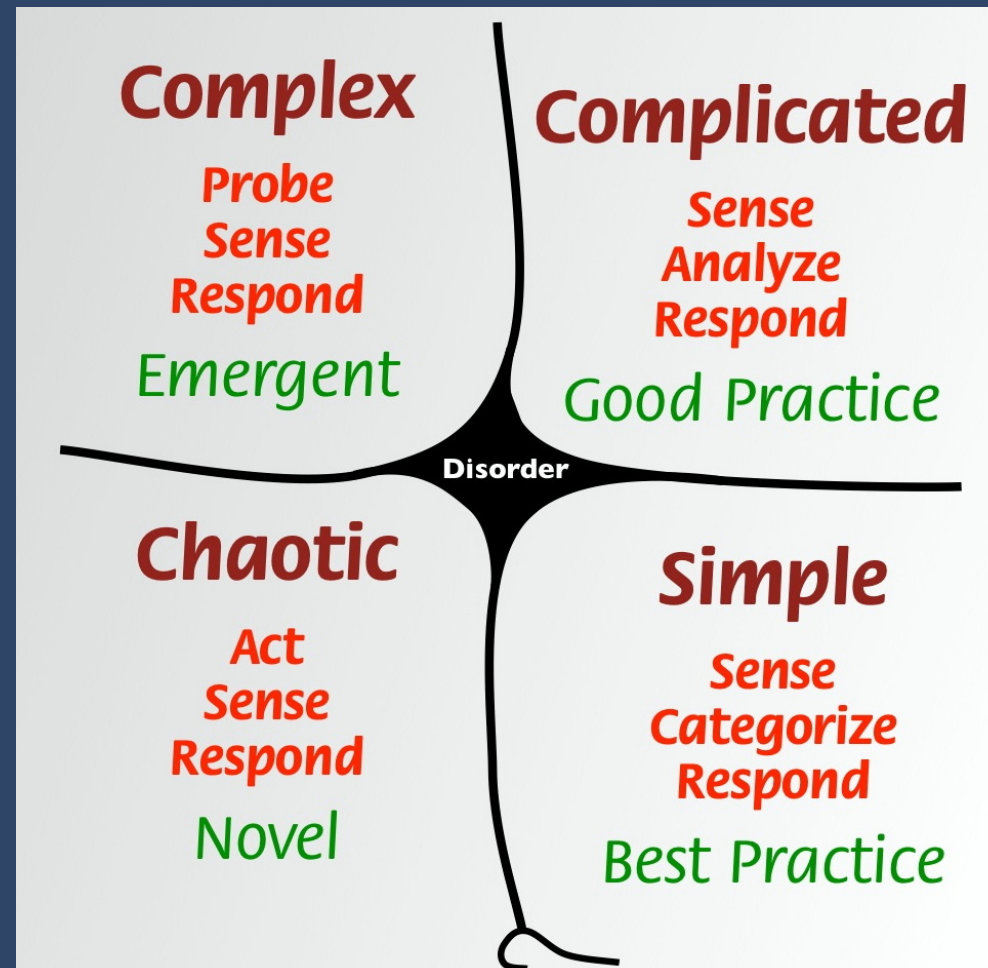
And something else we can check happens too

Event Storming



eventstorming.com

Cynefin



Cynefin for devs by Liz Keogh

Algunos consejos útiles

- Busca ejemplos relevantes

Algunos consejos útiles

- Busca ejemplos relevantes
- Escribe pruebas específicas

Algunos consejos útiles

- Busca ejemplos relevantes
- Escribe pruebas específicas
- Escribe código que sea cada vez más genérico

Algunos consejos útiles

- Busca ejemplos relevantes
- Escribe pruebas específicas
- Escribe código que sea cada vez más genérico
- Refactoriza pensando como un diseñador/arquitecto

Algunos consejos útiles

- Busca ejemplos relevantes
- Escribe pruebas específicas
- Escribe código que sea cada vez más genérico
- Refactoriza pensando como un diseñador/arquitecto
- No tengas miedo a borrar/mover/modificar código

Algunos consejos útiles

- Busca ejemplos relevantes
- Escribe pruebas específicas
- Escribe código que sea cada vez más genérico
- Refactoriza pensando como un diseñador/arquitecto
- No tengas miedo a borrar/mover/modificar código
- Las pruebas son código también, cuídalo

Algunos consejos útiles

"Act locally. Think Globally."

– Ron Jeffries

"Something that is hard to test is badly designed."

– Uncle Bob

Algunos consejos útiles

Demeter Law

Habla únicamente con los colaboradores más próximos

```
$this->getPerson()->getCar()->ignite();
```

Algunos consejos útiles

Demeter Law

Habla únicamente con los colaboradores más próximos

```
$this->getPerson()->getCar()->ignite();
```

debería ser

```
$this->person->startCar();
```

Algunos consejos útiles

Tell don't Ask

- Comportamiento donde la información reside
- Minimiza las interacciones entre objetos

```
if ($user.isAuthenticated()) {  
    echo “hola de nuevo “ . $user.getName();  
} else {  
    echo “bienvenido!”;  
}
```


Algunos consejos útiles

Tell don't Ask

- Comportamiento donde la información reside
- Minimiza las interacciones entre objetos

```
echo $user->welcomeMessage();
```

Algunos consejos útiles

The Transformation Priority Premise

null to constant

unconditional to if

constant to constant+

if to while

constant to scalar

expression to function

statement to statements

Kata de los números romanos

ej: 148 => CXLVIII

Kata de los números romanos

ej: 148 => CXLVIII

5 => V		
1 => I	6 => VI	10 => X
2 => II	7 => VII	20 => XX
3 => III	8 => VIII	30 => XXX
4 => IV	9 => IX	40 => XL
100 => C	500 => D	1000 => M

Conclusiones

- Ten en cuenta la curva de aprendizaje

Conclusiones

- Ten en cuenta la curva de aprendizaje
- Más tiempo al inicio del proyecto

Conclusiones

- Ten en cuenta la curva de aprendizaje
- Más tiempo al inicio del proyecto
- Menos tiempo al modificar/añadir funcionalidad

Conclusiones

- Ten en cuenta la curva de aprendizaje
- Más tiempo al inicio del proyecto
- Menos tiempo al modificar/añadir funcionalidad
- Aporta confianza al equipo de desarrollo

Conclusiones

- Ten en cuenta la curva de aprendizaje
- Más tiempo al inicio del proyecto
- Menos tiempo al modificar/añadir funcionalidad
- Aporta confianza al equipo de desarrollo
- No necesitas el 100% de cobertura

Enlaces de interés

- [Test Driven Development: By Example](#)
- [Growing Object-Oriented Software, Guided by Tests](#)
- [Refactoring: Improving the Design of Existing Code](#)
- [The Transformation Priority Premise](#)
- [Giving up on TDD](#)
- [phpspec](#)