

Epifanías de developer



kamstrup

W U N
D E R
D O G

TOXOTA



METADEV

plain
concepts

Ortus Solutions
A Software Revolution

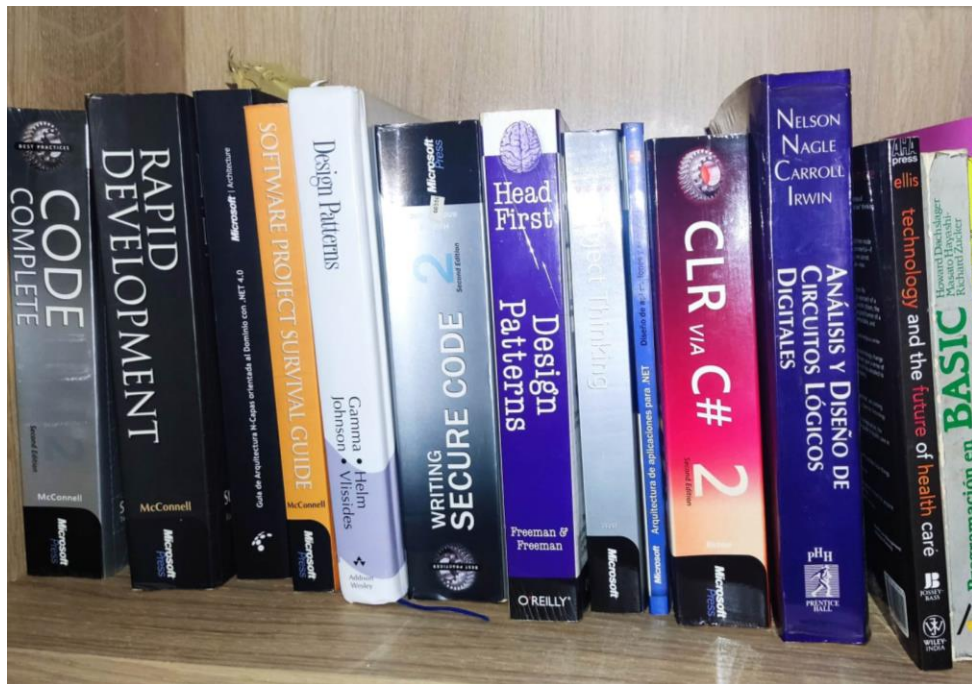


[@SevillaNetDevs](https://twitter.com/SevillaNetDevs)



[@ndoteclados](https://twitter.com/ndoteclados)

Leer es importante

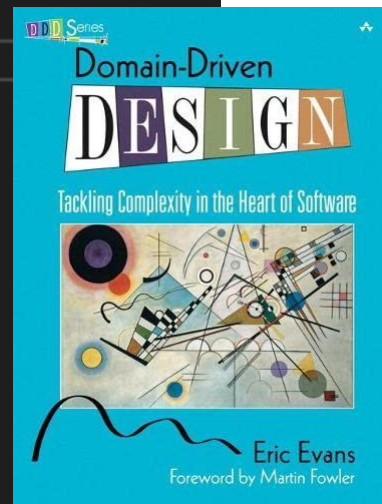


Interfaz o clase

```
private readonly IOfflineModeSettingsProvider offlineModeSettingsProvider;  
private readonly IConfigurationService configurationService;  
private readonly IConnectivity connectivityService;  
private readonly IActionsRepository actionsRepository;  
private readonly IServersChecker serversService;  
private readonly ILogService logService;  
private bool isOffline;  
private bool thereArePendingActionsToSend;  
private bool serversAvailable;
```

1 reference | Juan María Laó Ramos, 193 days ago | 1 author, 1 change

```
public OfflineModeService(  
    IOfflineModeSettingsProvider offlineModeSettingsProvider,  
    IConfigurationService configurationService,  
    IConnectivity connectivityService,  
    IActionsRepository actionsRepository,  
    IServersChecker serversService,  
    ILogService logService)
```



- Domain-Driven Design: Tackling Complexity in the Heart of Software (de Evans Eric)

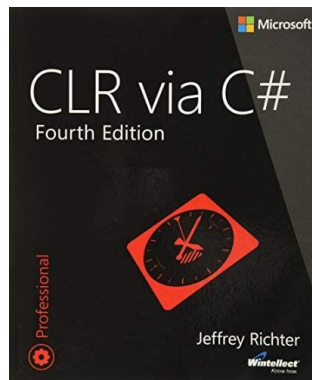
- Cambiemos una por ejemplo:
 - OfflineModeSettingsProvider



- Cambiemos una por ejemplo:
 - OfflineModeSettingsProvider

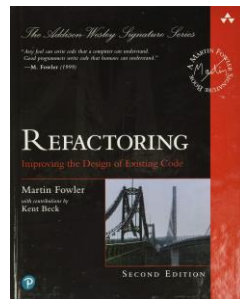
- ¿Porqué virtual?

- CLR via C# (Jeffrey Richter)



- Cambiemos una por ejemplo:
 - OfflineModeSettingsProvider

- Refactoring (Martin Fowler)



- ¿Porqué virtual?

- Ahh menos mal, hay teses que me van a avisar de si he roto algo.

- Código Sostenible (Carlos Ble)



• ¿Pero que es eso del Builder?

8 references | Juan María Laó Ramos, 199 days ago | 1 author, 1 change

```
public class Builder
{
    private readonly Mock<OfflineModeSettingsProvider> offlineModeSettingsProvider;
    private readonly Mock<IConfigurationService> configurationService;
    internal readonly Mock<IConnectivity> ConnectivityService;
    private readonly Mock<IActionsRepository> actionsRepository;
    internal readonly Mock<IServersChecker> ServersService;
    internal readonly Mock<ILogService> LogService;
    private bool withPendingActions = false;
```

1 reference | Juan María Laó Ramos, 199 days ago | 1 author, 1 change

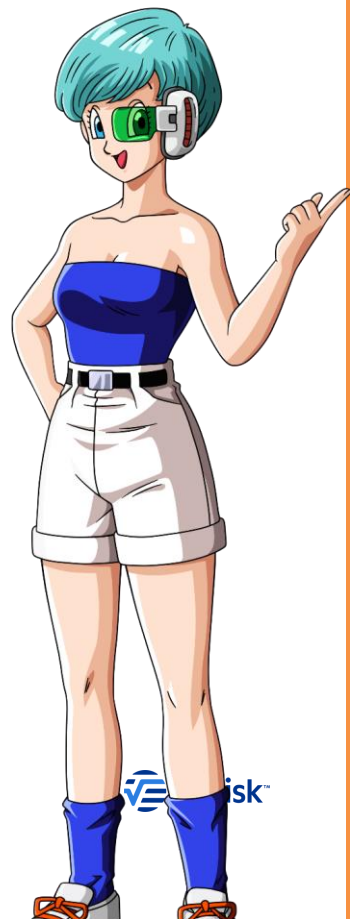
```
public Builder()
{
    this.offlineModeSettingsProvider = new Mock<OfflineModeSettingsProvider>();
    this.configurationService = new Mock<IConfigurationService>();
    this.ConnectivityService = new Mock<IConnectivity>();
    this.actionsRepository = new Mock<IActionsRepository>();
    this.ServersService = new Mock<IServersChecker>();
    this.LogService = new Mock<ILogService>();
}
```

12 references | 32/32 passing | Juan María Laó Ramos, 199 days ago | 1 author, 1 change

```
public OfflineModeService Build()...
```

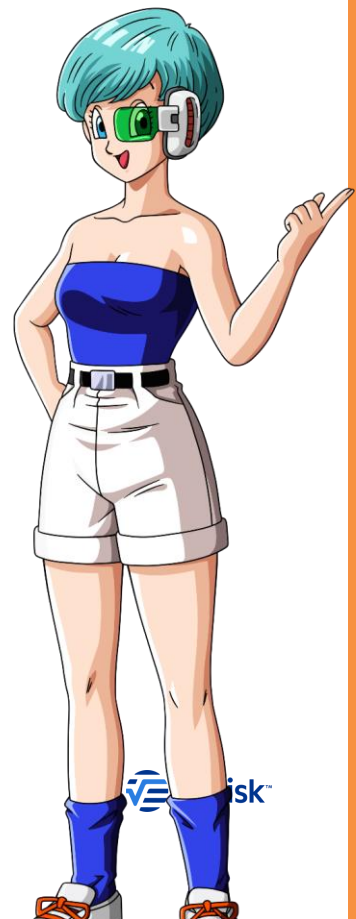
10 references | 29/29 passing | Juan María Laó Ramos, 199 days ago | 1 author, 1 change

```
internal Builder EnabledFeature(bool offlineFeatureEnabled = true)...
```



- ¿Pero que es eso del Builder?

```
12 references | 32/32 passing | Juan María Laó Ramos, 198 days ago | 1 author, 1 change  
public OfflineModeService Build()  
{  
    return new OfflineModeService(  
        this.offlineModeSettingsProvider.Object,  
        this.configurationService.Object,  
        this.ConnectivityService.Object,  
        this.actionsRepository.Object,  
        this.ServersService.Object,  
        this.LogService.Object);  
}
```

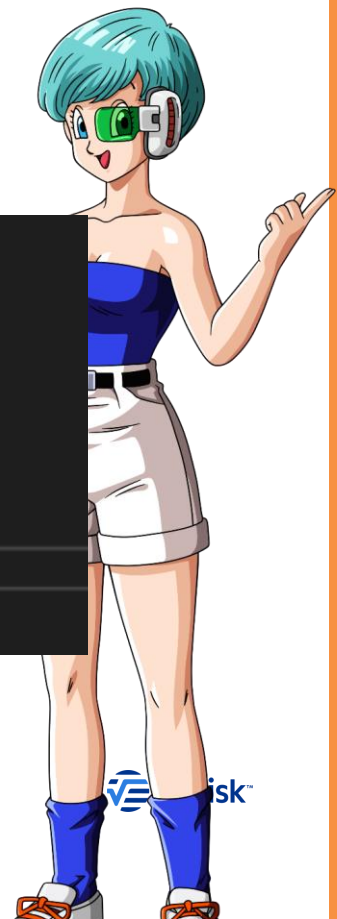


- ¿Pero que es eso del Builder?

10 references | 29/29 passing | Juan María Laó Ramos, 198 days ago | 1 author, 1 change

```
internal Builder EnabledFeature(bool offlineFeatureEnabled = true)
{
    this.configurationService
        .SetupGet(c => c.OfflineModeFeatureEnabled)
        .Returns(offlineFeatureEnabled);

    return this;
}
```



• ¿Pero que es eso del Builder?

[Fact]

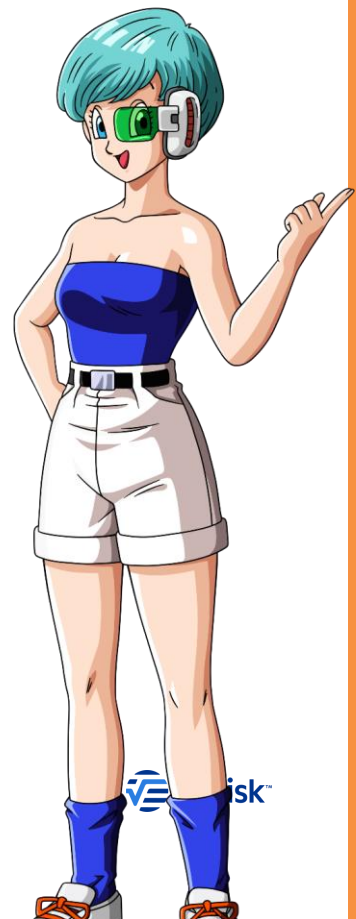
0 references | Juan María Laó Ramos, 198 days ago | 1 author, 1 change

```
public void ReturnToOnlineWhenServersBecomeAvailable()
{
    service = builder
        .WithServersAvailable(false)
        .WithInternet(NetworkAccess.Internet)
        .EnabledFeature(true)
        .SwitchIsActive(true)
        .Build();

    service.IsOffline = true;

    service.ServersAvailable = true;

    Assert.False(service.IsOffline);
}
```



- Se puede mejorar algo ahí

1 reference | 1/1 passing | Juan María Laó Ramos, 198 days ago | 1 author, 1 change

```
internal Builder WithPendingActions()
{
    this.actionsRepository
        .Setup(c => c.ThereAreActionsToSendAsync())
        .ReturnsAsync(true);
    return this;
}
```



Buenos nombres

- ¿Codigo legible = menos bugs?
- ¿Como lo hacemos?
- Decisión de diseño

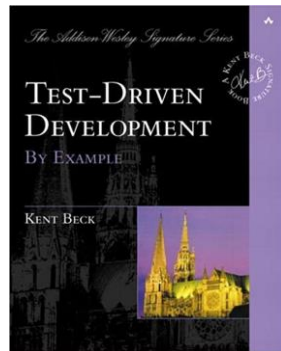


Falla pronto, falla rápido

- Probar las decisiones
- Si no gusta, tomamos otra decisión
- Y la probamos de nuevo
- Baby steps

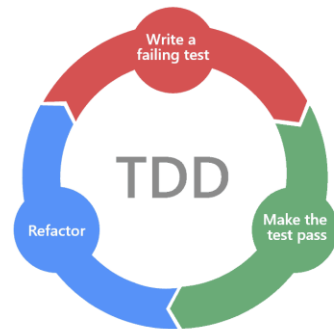
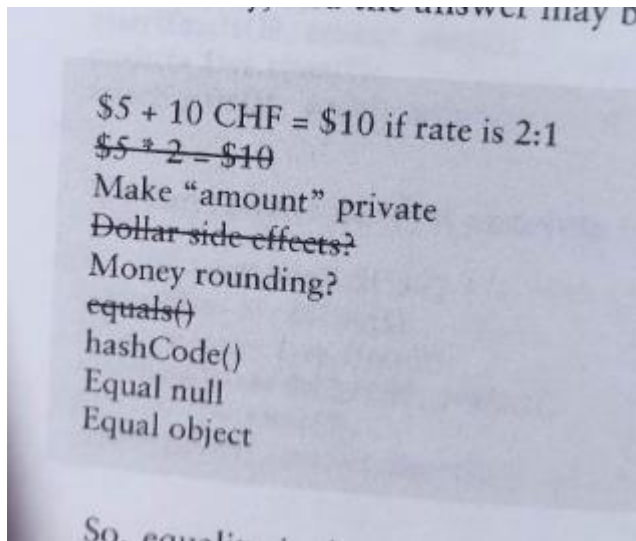


Test-Driven Development By example (Kent Beck)

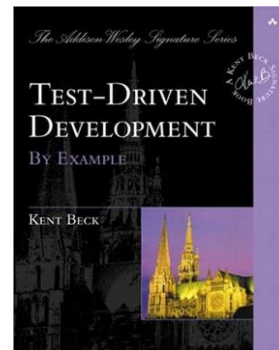


Falla pronto, falla rápido

- Cuidado con perder el foco



Test-Driven Development By example (Kent Beck)



Tip: Arrange-Act-Assert?

- O Assert-Act-Arrange?

```
[Fact]
```

```
• | 0 references | Juan María Laó Ramos, 199 days ago | 1 author, 1 change
```

```
public void patata()
```

```
{
```

```
    service = builder
```

```
        .WithServersAvailable(false)
```

```
        .WithInternet(NetworkAccess.Internet)
```

```
        .EnabledFeature(true)
```

```
        .SwitchIsActive(true)
```

```
        .Build();
```

```
    service.IsOffline = true;
```

```
    service.ServersAvailable = true;
```

```
    Assert.False(service.IsOffline);
```



IT WORKS ON MY MACHINE

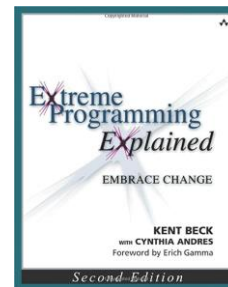
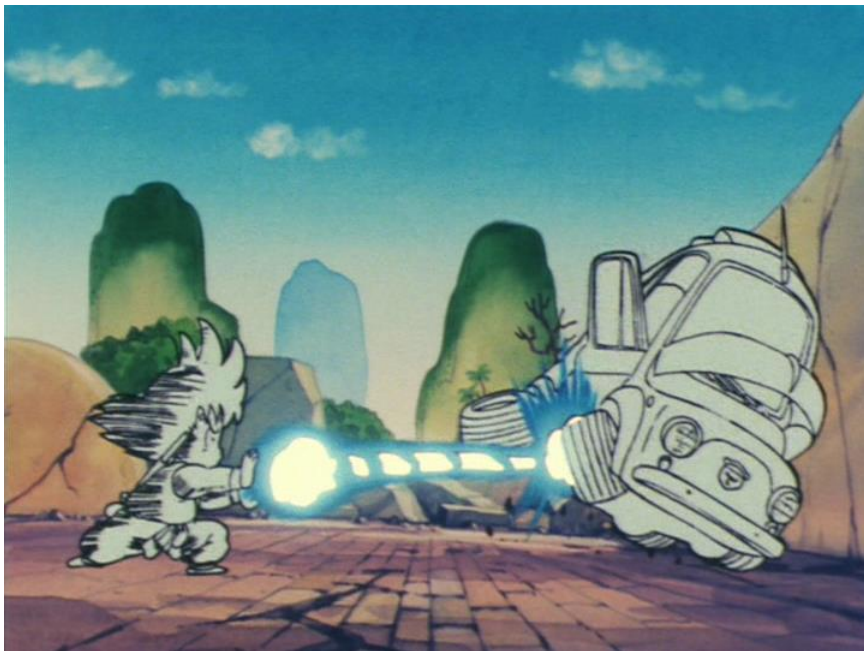


THEN WE'LL SHIP YOUR MACHINE



AND THAT IS HOW DOCKER WAS BORN

CI/CD Integración y entrega continua



Extreme
Programing
Explained
(Kent Beck)

No montar CI/CD

- ¿Más rápido?
 - A corto plazo
- Más seductor
- Pero no más poderoso
- No te obsesiones con la cobertura



Test	Duration	Trait
RealProject (32)	136 ms	
Tests.Features.Offline (32)	136 ms	
OfflineModeServiceShould (32)	136 ms	
No Traits (32)	136 ms	
BeOfflineWhenThereArePendingActions	< 1 ms	
CorrectlyEvaluatelsOfflineDependingOnSeveralParameters (9)	112 ms	
EnsuresOfflineChangedIsInvokedWhenConnectivityChanged (4)	10 ms	
EnsuresOfflineChangedIsInvokedWhenIsActiveChanged (8)	2 ms	
EvaluatePendingActionsWhenEvaluatinOffline	< 1 ms	
InvokeOfflineChangedWhenPendingActionsAreSetted	< 1 ms	
LogEvaluatelsOfflineOnlyWhenSwitchIsActive (2)	12 ms	
MainTainlsOfflinelSensorsAreNotAvailable	< 1 ms	

+

CI/CD



¿Hay tiempo para mas?



Historias de usuario

- Criterios de aceptación

Historias de usuario

- Criterios de aceptación
 - Dado que X Cuando Y Entonces Z

Nos dice si la historia es manejable

Identificamos información que falta

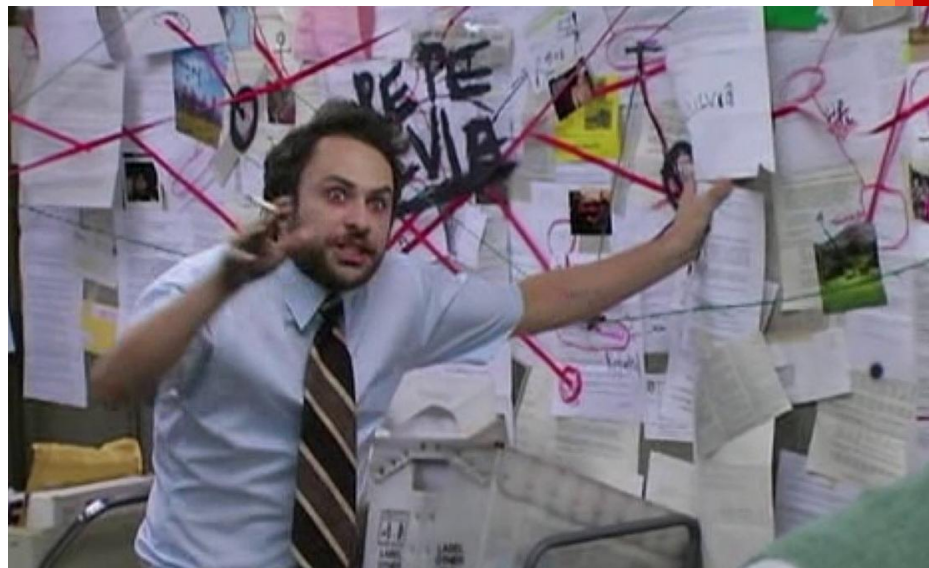
Más información para cuando vayamos a implementar

Menos sorpresas

Historias más cortas

Ramas de Git viven menos

Las revisiones por pares (Pull Request) son menos “pesadas”



#NoAlCodigoHomeopático

- Sin pruebas
- Sin revision por pares

Algunos recursos

- Domain-Driven Design: Tackling Complexity in the Heart of Software (de Evans Eric)
- CLR via C# (Jeffrey Ritcher)
- Código Sostenible (Carlos Ble)
- Test-Driven Development By example (Kent Beck)
- Extreme Programing Explained (Kent Beck)
- [How I Met Testing \(@SevillaNetDevs\)](#)
 - [Código en github](#)
- [@ndoteclados](#)



DOT NET MLG



kamstrup

W U N
D E R
D O G

TOYOTA



METADEV

plain
concepts

Ortus Solutions
A Software Revolution