

Capítulo 15: Programación del lado del cliente

Capítulo 16: Tecnologías complementarias en .NET

Capítulo 17: Pruebas unitarias



DAT
División de Alta Tecnología

17

Pruebas unitarias

Visual Studio 2015 Developer

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.




Objetivos

- Implementar pruebas unitarias al proyecto creado.

17 - 2

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Agenda

- ¿Por qué las pruebas unitarias?
- Desafíos con las pruebas unitarias
- ¿Cómo se crea un unit test?
- ¿Qué es un unit test?
- Ventajas de los unit test

17 - 3

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



¿Por qué las pruebas unitarias?

- “Inicia con el final en mente.”
-- Stephen Covey (The 7 Habits of Highly Effective People)
- Conjunto de herramientas que ayudan a crear una nueva base para el producto que es altamente estable, cambiabile y utilizable.
- Algo que podemos vender con confianza y todavía dormir por la noche.



17 - 4

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Desafíos con las pruebas unitarias

- ¡No tenemos tiempo para hacer esto!
- No hará una diferencia significativa.
- ¡Mi código no necesita pruebas!
- ¡Sólo haz que funcione ahora!
- ¿Cuántas pruebas de unidad necesitamos?
- ¿Qué conjunto de herramientas usar?
- ¿Hasta dónde llegar? (Por ejemplo: objetos simulados, CI)
- Plazos vs mejora de procesos
- ¿Podemos reducir nuestro personal de pruebas si hacemos la prueba de unidad?

17 - 5

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



¿Cómo se hace un unit test?

- Clase a crear una prueba unitaria:

```
public class MyMath
{
    public int Add(int i, int j)
    {
        return i + j;
    }
}
```

17 - 6

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Definición de unit

- La pieza más pequeña de código de software que se puede probar de forma aislada.



17 - 7

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



¿Esto es una prueba unitaria?

```
public void MyAddTest()
{
    MyMath m = new MyMath();
    Assert.AreEqual(m.Add(2, 3), 5);
}
```

17 - 8

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



¿Qué es un unit test?

Una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado.



17 - 9

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Ventajas de los unit test

- Fomentan el cambio
- Simplifica la integración
- Documenta el código
- Separación de la interfaz y la implementación
- Errores están más acotados y son más fáciles de localizar

17 - 10

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Ejercicio N° 17: Optimizar los unit test creados con xUnit y FluentAssertions

Optimizar los unit test existentes para su fácil lectura.

Al finalizar el laboratorio, el alumno logrará:

- Instalar, configurar y usar xUnit con FluentAssertions para simplificar los unit test.

17 - 11

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Tarea N° 17: Investigar los patrones basados en Unit Test

- Investigar los patrones basados en Unit Test.

17 - 12

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Lecturas adicionales

- Para obtener información adicional, puede consultar los siguientes enlaces:
 - https://es.wikipedia.org/wiki/prueba_unitaria
 - <https://msdn.microsoft.com/en-us/library/hh694602.aspx>

17 - 13

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Resumen

- Unit test son pruebas independientes de pequeños trozos de código creados durante el desarrollo.
- Existen buenos Frameworks de unit test para diferentes lenguajes y entornos de desarrollo.
- Las pruebas unitarias pueden afectar directamente la calidad del código durante la construcción.

17 - 14

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.

