

# Universidad ORT Uruguay

## Facultad de Ingeniería

### Obligatorio 1 - Diseño de Aplicaciones 2 - Agosto 2023

<https://github.com/IngSoft-DA2-2023-2/194592-275723-246326>

Luis Sanguinetti – 246326

Facundo Rodríguez - 194592

Leandro Olmedo - 275723

Docentes: Ignacio Valle, Marco Fiorito, Matías Salles

2023

# Índice

Clean Code	2
TDD	2
Pruebas automáticas (unitarias y de integración)	3
Reporte de la herramienta de cobertura y análisis del resultado	3

## Clean Code

El clean code es muy importante en el diseño y desarrollo de aplicaciones ya que prioriza la legibilidad, mantenibilidad y eficiencia de código. Al seguir las prácticas del clean code facilitamos la lectura, la comprensión y la modificación del código a futuro. Algunas de las prácticas de clean code utilizadas fueron:

- Nombres Descriptivos
- Funciones Pequeñas
- Uso de comentarios útiles
- Consistencia en la nomenclatura

## TDD

Dentro del contexto de nuestro enfoque en la calidad del código, Clean Code y patrones de diseño en nuestra educación de ingeniería en sistemas, el Test-Driven Development (TDD) emerge como una práctica esencial. TDD impulsa la creación de un código de la más alta calidad al poner un énfasis significativo en la planificación y la exhaustividad de las pruebas.

En nuestra experiencia, hemos abrazado TDD como un medio para asegurar que nuestro código no solo funcione correctamente, sino que también sea robusto y altamente mantenible. Sin embargo, reconocemos que alcanzar una cobertura de código del 90% o superior puede ser un desafío.

Para mejorar nuestra cobertura de código y elevar la calidad de nuestro software, consideramos algunos enfoques clave:

**Revisión y Refactorización:** Revisar nuestro código existente en busca de áreas no probadas y aplicar técnicas de refactorización para hacer que el código sea más testeable.

**Casos de Borde:** Identificar y probar casos de borde y situaciones límite para asegurarnos de que nuestro código sea resistente y confiable en condiciones extremas.

**Automatización de Pruebas:** Explorar herramientas y bibliotecas de automatización de pruebas que faciliten la creación y ejecución de pruebas de manera eficiente.

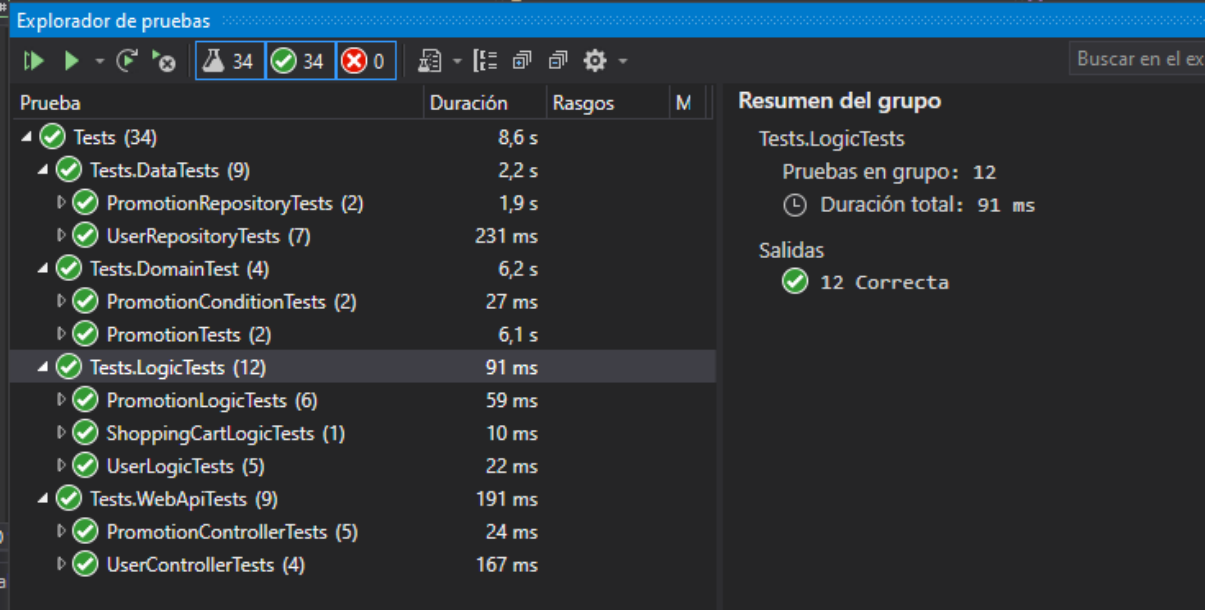
**Pair Programming:** Emplear el enfoque de programación en pareja para fomentar la revisión continua del código y compartir conocimientos entre el equipo.

**Mantener el Equilibrio:** Reconocer que la cobertura del 100% puede no ser siempre el objetivo óptimo, ya que puede conducir a pruebas innecesarias o complejidad excesiva. En su lugar, buscamos un equilibrio entre la cobertura y la eficacia de las pruebas.

El camino hacia una cobertura de código superior al 90% es un viaje continuo. Apreciamos que, en nuestro proceso educativo, estamos aprendiendo a abrazar los principios del Clean Code y a aplicar patrones de diseño de manera efectiva. A medida que avanzamos en nuestra formación, estamos comprometidos con la mejora constante de nuestras habilidades y la calidad de nuestro código.

Este enfoque nos impulsa a crear software que sea no solo funcional, sino también elegante, legible y sostenible a largo plazo, lo que refleja la esencia de la ingeniería en sistemas y la búsqueda de la excelencia en la creación de software.

## Pruebas automáticas (unitarias y de integración)



The screenshot shows the 'Explorador de pruebas' (Test Explorer) window in Visual Studio. At the top, there are icons for running tests: a green play button, a green checkmark with '34', and a red X with '0'. Below this is a table of test results. The table has columns for 'Prueba' (Test), 'Duración' (Duration), 'Rasgos' (Tags), and 'M' (Mark). The tests are organized in a tree structure. On the right, the 'Resumen del grupo' (Group Summary) for 'Tests.LogicTests' is shown, indicating 'Pruebas en grupo: 12' and 'Duración total: 91 ms'. Below this, the 'Salidas' (Outputs) section shows '12 Correcta' (12 Correct).

Prueba	Duración	Rasgos	M
Tests (34)	8,6 s		
Tests.DataTests (9)	2,2 s		
PromotionRepositoryTests (2)	1,9 s		
UserRepositoryTests (7)	231 ms		
Tests.DomainTest (4)	6,2 s		
PromotionConditionTests (2)	27 ms		
PromotionTests (2)	6,1 s		
Tests.LogicTests (12)	91 ms		
PromotionLogicTests (6)	59 ms		
ShoppingCartLogicTests (1)	10 ms		
UserLogicTests (5)	22 ms		
Tests.WebApiTests (9)	191 ms		
PromotionControllerTests (5)	24 ms		
UserControllerTests (4)	167 ms		

**Resumen del grupo**  
Tests.LogicTests  
Pruebas en grupo: 12  
Duración total: 91 ms  
Salidas  
12 Correcta

Nuestro equipo ha dedicado un esfuerzo significativo a la garantía de calidad a lo largo del proyecto, con un enfoque particular en pruebas unitarias e integración. Este compromiso nos ha llevado a desarrollar un conjunto impresionante de 34 pruebas, lo que refleja nuestra determinación por ofrecer un software confiable y libre de defectos.

### Pruebas Unitarias

Las pruebas unitarias son un pilar esencial en nuestra estrategia de pruebas. Se materializan en nuestros "DataTests," donde creamos un entorno simulado de base de datos utilizando mock objects. Este enfoque nos permite enfocarnos en un aspecto crítico: la capacidad de acceder y manipular los datos de manera correcta, independientemente de la implementación subyacente. Hemos trabajado incansablemente para garantizar que nuestros métodos de acceso a datos se comporten según lo esperado, respaldando así la integridad de nuestros datos.

### Pruebas de Integración

Nuestras pruebas de integración son la columna vertebral de la validación de extremo a extremo de nuestro sistema. Estas pruebas, implementadas en nuestros controladores, abarcan todo el flujo de trabajo de la aplicación. Comienzan en el controlador, atraviesan la capa de negocios y finalmente llegan a la capa de acceso a datos. Esta estrategia nos ha permitido garantizar que todos los componentes funcionen armoniosamente juntos, lo que es esencial para la operación sin problemas de nuestra aplicación.

Este enfoque exhaustivo en pruebas refleja nuestra dedicación a la calidad del código y a la entrega de un producto que inspire confianza. Cada prueba es un paso hacia la excelencia, y cada error descubierto es una oportunidad para mejorar. Estamos orgullosos de nuestro compromiso con la calidad y estamos decididos a seguir elevando el estándar de nuestro software.

## Reporte de la herramienta de cobertura y análisis del resultado

Jerarquía	Cubiertos (bloques)	No cubiertos (bloques)	Cubiertas (líneas)	Parcialmente cubiertas (lin	No cubiertas (líneas)	Cubiertos (%bloques)
admin_PC-LEANDRO 2023-10-04 19_05_31.coverage	1543	496	1272	9	631	75,67%
tests.dll	942	7	820	1	5	99,26%
logic.dll	130	12	75	3	0	91,55%
exceptions.dll	4	8	6	0	12	33,33%
servicesfactory.dll	0	8	0	0	10	0,00%
webapi.dll	54	50	47	0	29	51,92%
apimodels.dll	124	18	101	0	15	87,32%
domain.dll	211	52	165	5	43	80,23%
data.dll	78	341	58	0	517	18,62%