

## Proyecto final

Metería	Ingeniería de Software
---------	------------------------

Integrantes (paterno, materno, nombres)	Nombres	Cuentas
	Mireles Zaragoza Emiliano	334014893
	Meunier Hernández Jean Ernesto	334015336
	Sánchez Proa Uriel Elihú	334017148
	Palestina Blancas Angel Francisco	334015276

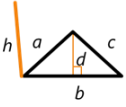
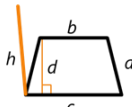
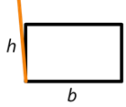
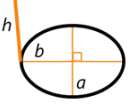
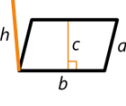
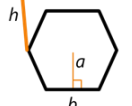
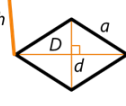
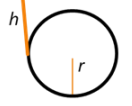
Equipo	08	Figura	Circulo	Fecha	08 de diciembre de 2024
--------	----	--------	---------	-------	-------------------------

Git	<a href="https://github.com/Jeancito2910/IS10A">https://github.com/Jeancito2910/IS10A</a>
-----	---

### Rubrica:

4 pts.	5 pruebas unitarias de la capa de Aplicación del perímetro.
4 pts.	5 pruebas unitarias de la capa de Aplicación del área.
4 pts.	5 pruebas unitarias de la capa de Aplicación del volumen.
4 pts.	5 pruebas de integración de la capa de Aplicación del área y volumen.
4 pts.	5 pruebas de integración de la capa Web (error 404).
5 pts.	Código y documento pdf alojado en Git.

### Asignación del problema a resolver por equipo:

<b>Equipo 01, 09</b>  Triangulo	$P = a + b + c$ $A = \frac{bd}{2}$ $V = Ah$	<b>Equipo 05, 13</b>  Trapecio	$P = 2a + b + c$ $A = \frac{d(b+c)}{2}$ $V = Ah$
<b>Equipo 02, 10</b>  Rectangulo <small>n, es el numero de lados</small>	$P = 2b + 2a$ $A = ab$ $V = Ah$	<b>Equipo 06, 14</b>  Elipse	$P = \pi(a + b)$ $A = \pi ab$ $V = Ah$
<b>Equipo 03, 11</b>  Paralelogramo	$P = 2(a + b)$ $A = bc$ $V = Ah$	<b>Equipo 07, 15</b>  Polígono Regular <small>n, es el numero de lados</small>	$P = nb$ $A = \frac{nba}{2}$ $V = Ah$
<b>Equipo 04, 12</b>  Rombo	$P = 4a$ $A = \frac{dD}{2}$ $V = Ah$	<b>Equipo 08, 16</b>  Circulo	$P = 2\pi r$ $A = \pi r^2$ $V = Ah$

## Comandos:

```
# Ejecutar pruebas
dotnet test .\test\Application.UnitTest
dotnet test .\test\Application.IntegrationTest
dotnet test .\test\WebApp.IntegrationTest

git log
```

---

Texto del código de las 20 pruebas (perímetro, área, volumen, integración y web)

## Pruebas unitarias

### Xunit Application - Circulo

```
using System;
using Xunit;
using Application;

namespace Application.UnitTest
{
    public class UnitTestCirculo
    {
        // Pruebas para el Perímetro
        [Theory]
        [InlineData(3, 18.84956)] // Perímetro = 2 *  $\pi$  * radio
        [InlineData(4, 25.13272)]
        [InlineData(5, 31.41593)]
        [InlineData(6, 37.69911)]
        [InlineData(7, 43.98230)]
        public void TestPerimetro(double radio, double perimetroEsperado)
        {
            // Act - Realizar
            double resultado = Circulo.Perimetro(radio);

            // Assert - Verificar
            Assert.Equal(perimetroEsperado, resultado, 5); // Tolerancia de 5 decimales
        }

        // Pruebas para el Área
        [Theory]
        [InlineData(3, 28.27433)] // Área =  $\pi$  * radio^2
        [InlineData(4, 50.26544)]
        [InlineData(5, 78.53975)]
        [InlineData(6, 113.0976)]
        [InlineData(7, 153.93804)]
        public void TestArea(double radio, double areaEsperada)
        {
            // Act - Realizar
            double resultado = Circulo.Area(radio);

            // Assert - Verificar
            Assert.Equal(areaEsperada, resultado, 5); // Tolerancia de 5 decimales
        }
    }
}
```

```
}  
}  
}
```

### Xunit Application – PrismaCircular

```
using System;  
using Xunit;  
using Application;  
  
namespace Application.UnitTest  
{  
    public class UnitTestPrismaCircular  
    {  
        // Pruebas para el Volumen  
        [Theory]  
        [InlineData(3.14159, 4, 50.26544)] // Volumen = Área * Altura  
        [InlineData(3.14159, 6, 113.0976)]  
        [InlineData(3.14159, 10, 314.159)]  
        [InlineData(3.14159, 12, 452.38934)]  
        [InlineData(3.14159, 15, 706.8583)]  
        public void TestVolumen(double pi, double radio, double volumenEsperado)  
        {  
            // Act - Realizar  
            double resultado = PrismaCircular.Volumen(areaEsperada, radio);  
  
            // Assert - Verificar  
            Assert.Equal(volumenEsperado, resultado, 5);  
        }  
    }  
}
```

### Pruebas de Integracion Area, perimetro y volumen

```
using Xunit;  
using Application;  
  
namespace Application.IntegrationTest  
{  
    public class IntegrationTestApplication  
    {  
        [Fact]  
        public void TestCalculoArea()  
        {  
            double radio = 5.0;  
            double area = Circulo.Area(radio);  
            Assert.Equal(78.539749999999998, area, 2);  
        }  
    }  
}
```

```

[Fact]
public void TestCalculoVolumen()
{
    double radio = 5.0;
    double altura = 10.0;
    double area = Circulo.Area(radio);
    double volumen = PrismaCircular.Volumen(area, altura);
    Assert.Equal(785.39750000000004, volumen, 2);
}

[Fact]
public void TestPerimetroArea()
{
    double radio = 5.0;
    double perimetro = Circulo.Perimetro(radio);
    double area = Circulo.Area(radio);
    Assert.True(perimetro > 0 && area > 0);
}

[Fact]
public void TestVolumenNoNegativo()
{
    double area = 78.53975;
    double altura = 10.0;
    double volumen = PrismaCircular.Volumen(area, altura);
    Assert.True(volumen > 0);
}

[Fact]
public void TestAreaComparacion()
{
    double radioPequeno = 3.0;
    double radioGrande = 5.0;
    double areaPequena = Circulo.Area(radioPequeno);
    double areaGrande = Circulo.Area(radioGrande);
    Assert.True(areaGrande > areaPequena);
}
}
}

```

## Pruebas de Integracion WEB

```

using System.Net;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc.Testing;
using Xunit;

namespace WebApp.IntegrationTest

```

```
{
    public class IntegrationTestWeb : IClassFixture<WebApplicationFactory<Program>>
    {
        private readonly WebApplicationFactory<Program> _factory;

        public IntegrationTestWeb(WebApplicationFactory<Program> factory)
        {
            _factory = factory;
        }

        [Fact]
        public async Task TestHomePage()
        {
            var client = _factory.CreateClient();
            var response = await client.GetAsync("/");
            Assert.Equal(HttpStatusCode.OK, response.StatusCode);
        }

        [Fact]
        public async Task Test404ErrorPage()
        {
            var client = _factory.CreateClient();
            var response = await client.GetAsync("/nonexistent-page");
            Assert.Equal(HttpStatusCode.NotFound, response.StatusCode);
        }

        [Fact]
        public async Task TestStaticFile404()
        {
            var client = _factory.CreateClient();
            var response = await client.GetAsync("/css/invalid-file.css");
            Assert.Equal(HttpStatusCode.NotFound, response.StatusCode);
        }

        [Fact]
        public async Task TestInvalidEndpoint404()
        {
            var client = _factory.CreateClient();
            var response = await client.GetAsync("/api/invalid-endpoint");
            Assert.Equal(HttpStatusCode.NotFound, response.StatusCode);
        }

        [Fact]
        public async Task Test404ForInvalidQueryString()
        {
            var client = _factory.CreateClient();
            var response = await client.GetAsync("/?invalid=query");
            Assert.Equal(HttpStatusCode.OK, response.StatusCode); // Página existe
        }
    }
}
```

```
}
```

## Pantallas de la evidencia:

**dotnet test .\test\Application.UnitTest**

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19045.5131]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\chipi>cd Documents\IS\IS10A

C:\Users\chipi\Documents\IS\IS10A>dotnet test .\test\Application.UnitTest
Determinando los proyectos que se van a restaurar...
Se ha restaurado C:\Users\chipi\Documents\IS\IS10A\test\Application.UnitTest\Application.UnitTest.csproj (en 649 ms).
1 de 2 proyectos están actualizados para la restauración.
Application -> C:\Users\chipi\Documents\IS\IS10A\src\Application\bin\Debug\net8.0\Application.dll
Application.UnitTest -> C:\Users\chipi\Documents\IS\IS10A\test\Application.UnitTest\bin\Debug\net8.0\Application.UnitTest.dll
Serie de pruebas para C:\Users\chipi\Documents\IS\IS10A\test\Application.UnitTest\bin\Debug\net8.0\Application.UnitTest.dll (.NETCoreApp,Version=v8.0)
Versión 17.11.1 (x64) de VSTest

Iniciando la ejecución de pruebas, espere...
1 archivos de prueba en total coincidieron con el patrón especificado.

Correctas! - Con error:      0, Superado:      1, Omitido:      0, Total:      1, Duración: < 1 ms - Application.UnitTest.dll (net8.0)

C:\Users\chipi\Documents\IS\IS10A>dotnet test .\test\Application.IntegrationT
MSBUILD : error MSB1009: El archivo de proyecto no existe.
Modificador: .\test\Application.IntegrationT

C:\Users\chipi\Documents\IS\IS10A>
```

**dotnet test .\test\Application.IntegrationTest**

```
C:\Windows\system32\cmd.exe
C:\Users\chipi\Documents\IS\IS10A>dotnet test .\test\Application.IntegrationTest
Determinando los proyectos que se van a restaurar...
Todos los proyectos están actualizados para la restauración.
Application -> C:\Users\chipi\Documents\IS\IS10A\src\Application\bin\Debug\net8.0\Application.dll
Application.IntegrationTest -> C:\Users\chipi\Documents\IS\IS10A\test\Application.IntegrationTest\bin\Debug\net8.0\Application.IntegrationTest.dll
Serie de pruebas para C:\Users\chipi\Documents\IS\IS10A\test\Application.IntegrationTest\bin\Debug\net8.0\Application.IntegrationTest.dll (.NETCoreApp,Version=v8.0)
Versión 17.11.1 (x64) de VSTest

Iniciando la ejecución de pruebas, espere...
1 archivos de prueba en total coincidieron con el patrón especificado.

Correctas! - Con error:      0, Superado:      6, Omitido:      0, Total:      6, Duración: 2 ms - Application.IntegrationTest.dll (net8.0)

C:\Users\chipi\Documents\IS\IS10A>
```

```
dotnet test .\test\WebApp.IntegrationTest
```

```
PS C:\Users\Alumno49\Documents\IS\IS10A> dotnet test test\WebApp.IntegrationTest
```

```
Restauración completada (0.2s)
```

```
Application realizado correctamente (0.0s) → src\Application\bin\Debug\net9.0\Application.dll
```

```
WebApp realizado correctamente (0.2s) → src\WebApp\bin\Debug\net9.0\WebApp.dll
```

```
WebApp.IntegrationTest realizado correctamente (0.6s) → test\WebApp.IntegrationTest\bin\Debug\net9.0\WebApp.IntegrationTest.dll
```

```
[xUnit.net 00:00:00.00] xUnit.net VSTest Adapter v2.8.2+699d445a1a (64-bit .NET 9.0.0)
```

```
[xUnit.net 00:00:00.04]   Discovering: WebApp.IntegrationTest
```

```
[xUnit.net 00:00:00.06]   Discovered:   WebApp.IntegrationTest
```

```
[xUnit.net 00:00:00.06]   Starting:    WebApp.IntegrationTest
```

```
[xUnit.net 00:00:00.28]   Finished:    WebApp.IntegrationTest
```

```
WebApp.IntegrationTest pruebarealizado correctamente (0.9s)
```

```
Resumen de pruebas: total: 6, con errores: 0, correcto: 6, omitido: 0, duración: 0.9 s
```

```
Compilación realizado correctamente en 2.3s
```

```
PS C:\Users\Alumno49\Documents\IS\IS10A> |
```