

Sistema de Modelado de Figuras Geométricas en 2D y 3D

María Fernanda Giraldo Duque

Juan Sebastián Arévalo Vásquez

Emerick Ríos Villa

Nicolas González Toro

Universidad de San Buenaventura Cali

Técnicas Avanzadas de Programación

Iván Mauricio Cabezas Troyano

19 de agosto de 2025

Tabla de Contenido

Introducción	3
Metodología ADIM.....	3
Análisis	3
<i>Historias De Usuario Y Criterios BDD</i>	<i>4</i>
Diseño	6
<i>Diagrama De Clases.....</i>	<i>6</i>
<i>Diagramas De Secuencia</i>	<i>7</i>
<i>Justificación De La Escogencia Del Patrón Aplicado En Cada Caso.....</i>	<i>8</i>
Factory	8
Unidad Adapter.....	8
Implementación	9
<i>Git Como Control De Versiones</i>	<i>9</i>
<i>BDD</i>	<i>10</i>
<i>Analizadores Estático De Código</i>	<i>12</i>
<i>Contenedores Docker Para Facilitar El Desarrollo</i>	<i>13</i>
Medición.....	13
<i>Ejecución del código</i>	<i>13</i>

Introducción

El presente documento describe el proceso de desarrollo de un software educativo orientado al cálculo de propiedades geométricas como el área, perímetro y volumen, aplicando la metodología ADIM (Análisis, Diseño, Implementación y Medición). El sistema fue desarrollado utilizando los lenguajes de programación Java, Python y C++, aprovechando los principios de la programación orientada a objetos para una estructura modular, reutilizable y escalable.

Metodología ADIM

A continuación, se detalla cada una de las fases seguidas en el proceso de desarrollo.

Análisis

En esta fase se identificaron las nuevas necesidades del sistema, con base en la evolución del software del primer taller. Ahora el sistema debe incorporar las siguientes funcionalidades:

El sistema debe permitir la creación de las figuras círculo, cuadrado, cubo y esfera, a partir de la longitud de su componente de área o volumen (lado o radio).

El sistema permite la selección de una unidad de medida (centímetros, metros, pulgadas, pies) para el despliegue de las medidas de las figuras.

El sistema debe asignar un identificado único a cada figura creada.

El sistema debe permitir el almacenamiento de las figuras creadas, habilitando su coexistencia en memoria de manera simultánea.

El sistema debe permitir la consulta y despliegue de las figuras que se han creado, empleando una unidad de medida especificada.

El sistema debe permitir el eliminar una figura específica, con base en su identificador.

El sistema debe permitir la persistencia de las figuras que existan en un momento dado.

El sistema debe permitir la recuperación y carga de las figuras creadas en la ejecución previa del programa para su manipulación.

Restricciones:

El sistema debe ejecutarse desde el CLI, mediante un menú.

El sistema debe implementar POO.

El Sistema debe realizarse en los lenguajes de programación Python, C++, Java.

Historias De Usuario Y Criterios BDD

Id de la Historia de Usuario: HU01	Título: Crear figuras	Prioridad: Alta
Historia de Usuario: Como usuario Quiero crear figuras geométricas ingresando sus medidas Para calcular automáticamente sus propiedades (área o volumen)		
Criterios BDD: Dado que el usuario está en la opción "Crear figura" Cuando selecciona el tipo de figura e ingresa su medida correspondiente Entonces el sistema calcula sus propiedades (área o volumen), asigna un identificador único y muestra la confirmación.		

Id de la Historia de Usuario: HU02	Título: Selección de una unidad de medida	Prioridad: Media
Historia de Usuario: Como usuario Quiero seleccionar una unidad de medida Para visualizar las propiedades (área o volumen) de las figuras en la unidad correspondiente		
Criterios BDD: Dado que el sistema gestiona diferentes unidades de medida Cuando el usuario selecciona una unidad específica Entonces el sistema convierte y muestra las medidas de las figuras en la unidad seleccionada.		

Id de la Historia de Usuario: HU03	Título: Asignar un identificador único	Prioridad: Alta
Historia de Usuario: Como sistema Quiero asignar un identificador único a cada figura creada Para garantizar su correcta identificación		
Criterios BDD: Dado que el usuario crea una figura Cuando el sistema registra la figura Entonces se asigna un identificador único y se muestra en la confirmación		

Id de la Historia de Usuario: HU04	Título: Almacenamiento de las figuras creadas	Prioridad: Media
Historia de Usuario: Como usuario Quiero que las figuras creadas permanezcan en memoria durante la sesión Para poder consultarlas y gestionarlas		
Criterios BDD: Dado que el usuario ha creado una o más figuras Cuando solicita visualizar las figuras almacenadas Entonces el sistema muestra la lista de todas las figuras registradas en la sesión		

Id de la Historia de Usuario: HU05	Título: Consulta y despliegue de las figuras que se han creado	Prioridad: Media
Historia de Usuario: Como usuario Quiero consultar las figuras creadas con sus datos registrados Para obtener información actualizada		
Criterios BDD: Dado que existen figuras registradas Cuando el usuario solicita la consulta de figuras Entonces el sistema muestra la lista completa con identificador, tipo y medidas.		

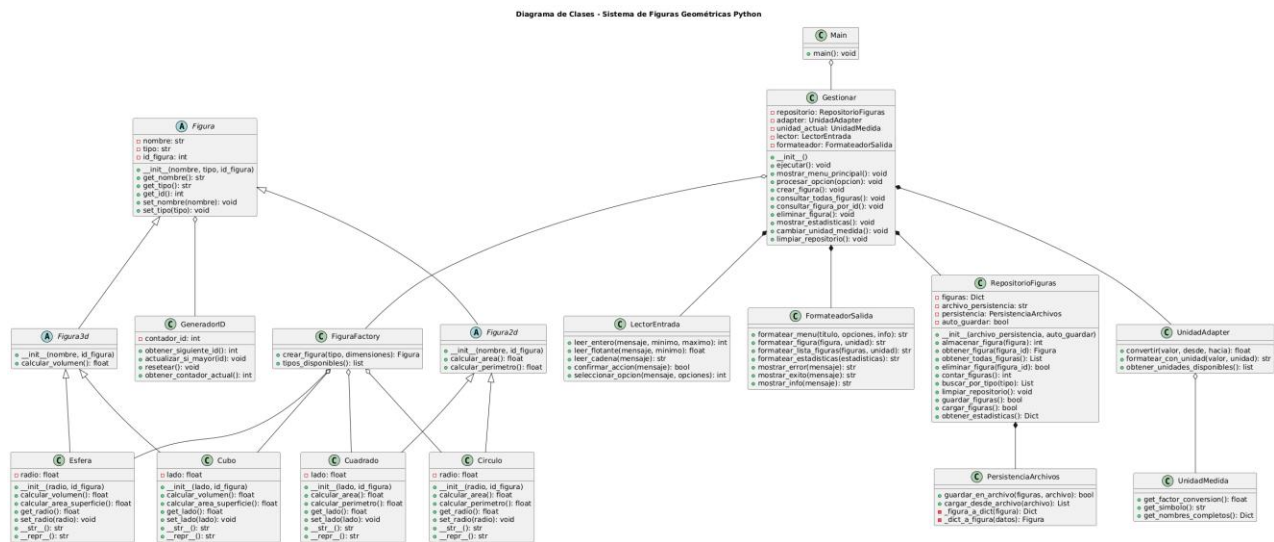
Id de la Historia de Usuario: HU06	Título: Eliminar una figura específica	Prioridad: Media
Historia de Usuario: Como usuario Quiero eliminar una figura utilizando su identificador Para mantener solo las que necesito		
Criterios BDD: Dado que existe una figura registrada Cuando el usuario indica su identificador y confirma la eliminación Entonces el sistema elimina la figura y muestra un mensaje de confirmación		

Id de la Historia de Usuario: HU07	Título: Persistencia de las figuras que existan	Prioridad: Alta
Historia de Usuario: Como usuario Quiero guardar las figuras creadas en un archivo Para recuperarlas en futuras sesiones		
Criterios BDD: Dado que existen figuras registradas Cuando el usuario solicita guardar las figuras Entonces el sistema persiste los datos y muestra un mensaje de confirmación		

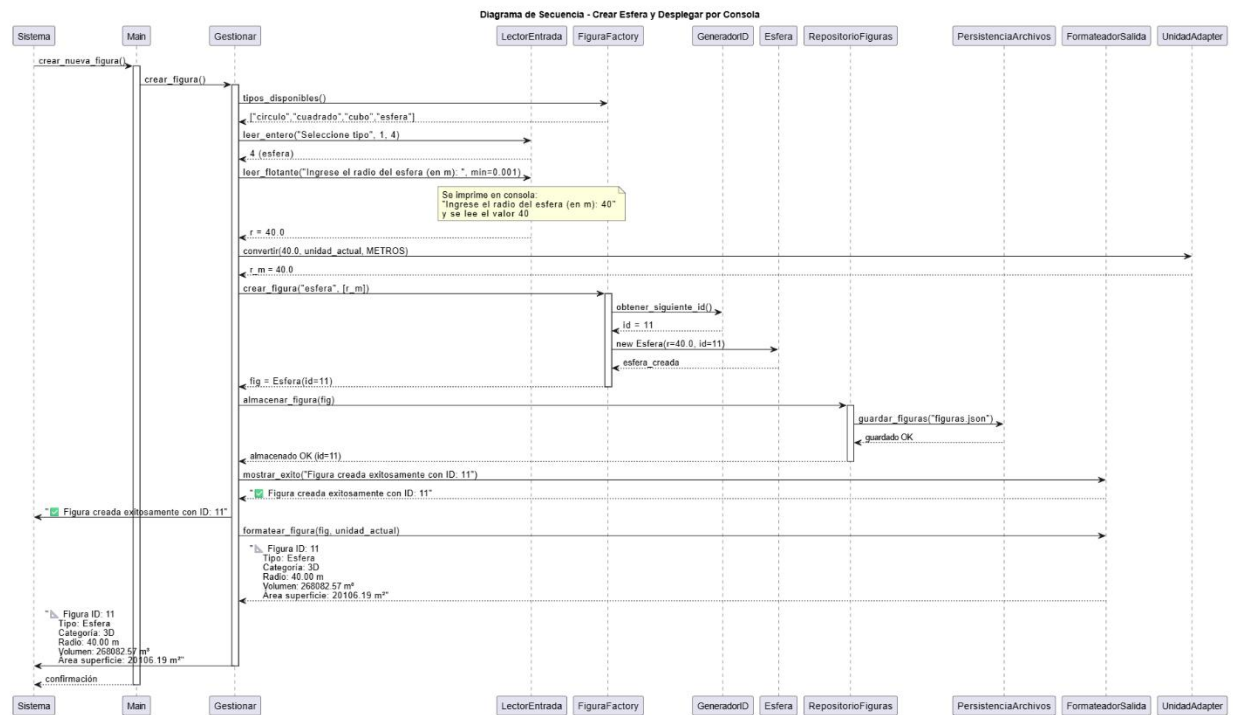
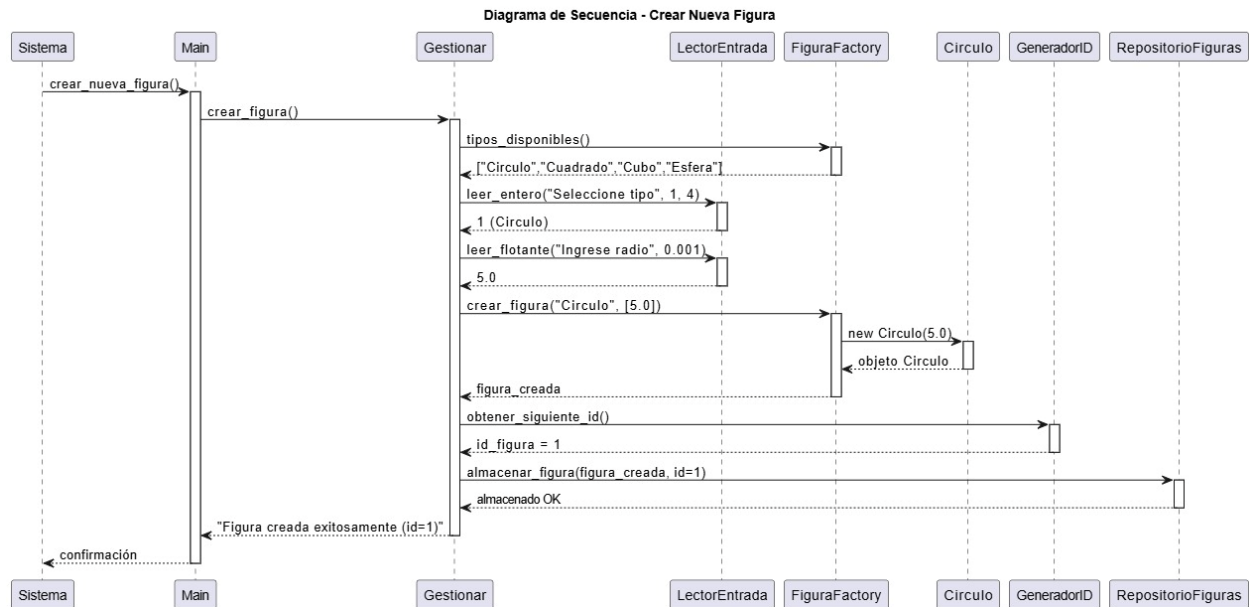
Id de la Historia de Usuario: HU08	Título: Recuperación y carga de las figuras creadas	Prioridad: Alta
Historia de Usuario: Como usuario Quiero cargar las figuras almacenadas previamente en un archivo Para restaurar el estado anterior y seguir trabajando con ellas		
Criterios BDD: Dado que existe un archivo con figuras previamente guardadas Cuando el usuario selecciona la opción "Cargar figuras" Entonces el sistema lee el archivo, recupera las figuras y las muestra en la lista actual.		

Diseño

Diagrama De Clases



Diagramas De Secuencia



Justificación De La Escogencia Del Patrón Aplicado En Cada Caso

Factory

Es un patrón que centraliza la creación de objetos. En vez de que cada parte del programa haga `new Circulo(...)` o `new Cubo(...)`, le pedimos a una fábrica que, según un “tipo”, devuelva la instancia correcta.

Utilizamos este patrón de diseño creacional porque nuestro flujo trabaja con tipos de figuras que llegan desde entrada/archivo, y tener una sola puerta de creación nos da orden, mantenibilidad y OCP real haciendo que el sistema quede abierto a extender nuevas figuras y cerrado a modificar lo ya probado.

En el proyecto, eso vive en FiguraFactory.py: se le pasa el tipo y parámetros, y te regresa el objeto correcto (Circulo, Cuadrado, Cubo, Esfera).

Unidad Adapter

Es un patrón que adapta/convierte una interfaz o valores para que una parte del sistema no tenga que cambiar. En nuestro caso: convierte unidades (m, cm, mm, in, ft) para que el dominio trabaje siempre “limpio”.

Utilizamos este patrón de diseño estructural porque necesitamos aceptar datos en distintas unidades y responder en la que el usuario pida, sin acoplar la lógica de conversión a las clases Círculo, Cubo, etc. Con el adapter Cumplimos Responsabilidad Única (SRP): conversión aquí, cálculos allá.

En el proyecto, esto se ve en UnidadAdapter.py (y UnidadMedida.py): recibe valores en una unidad, los pasa a la estándar para calcular, y si hace falta, los vuelve a convertir para la salida.

Implementación

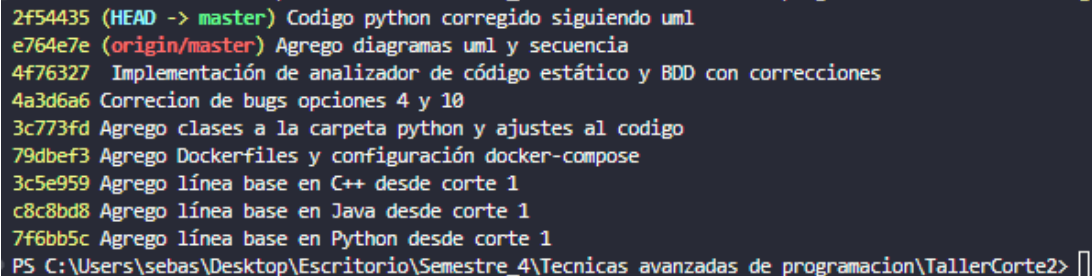
En esta parte de implementación se utiliza Git como herramienta de control de versiones, lo cual permite gestionar de forma organizada los cambios, mantener un historial del proyecto y facilitar la colaboración.

También se implementó Behavior Driven Development (BDD), que guía la creación del código a partir de criterios de aceptación expresados como comportamientos esperados.

Para garantizar el cumplimiento de dichos criterios, se implementa el ciclo rojo-verde-refactor, con el fin de escribir pruebas que inicialmente fallen, hacerlas pasar con la mínima implementación y finalmente optimizar el código.

Además, se incorporan analizadores estáticos de código, que permiten detectar errores potenciales y problemas de estilo antes de la ejecución, contribuyendo así a la construcción de un software más robusto y confiable.

Git Como Control De Versiones



```

2f54435 (HEAD -> master)Codigo python corregido siguiendo uml
e764e7e (origin/master)Agrego diagramas uml y secuencia
4f76327 Implementación de analizador de código estático y BDD con correcciones
4a3d6a6 Correccion de bugs opciones 4 y 10
3c773fd Agrego clases a la carpeta python y ajustes al codigo
79dbef3 Agrego Dockerfiles y configuración docker-compose
3c5e959 Agrego línea base en C++ desde corte 1
c8c8bd8 Agrego línea base en Java desde corte 1
7f6bb5c Agrego línea base en Python desde corte 1
PS C:\Users\sebas\Desktop\Escritorio\Semestre_4\Tecnicas avanzadas de programacion\TallerCorte2>
  
```

En la figura se observa el historial de commits del proyecto, donde cada cambio significativo fue registrado con un mensaje descriptivo. Esto permite mantener trazabilidad y control de versiones durante el desarrollo.

BDD

collected 18 items

```
test_figuras_bdd.py::TestCreacionFigurasConID::test_figura_tiene_id_unico ERROR [ 5%]
test_figuras_bdd.py::TestCreacionFigurasConID::test_multiples_figuras_ids_diferentes ERROR [ 11%]
test_figuras_bdd.py::TestFactory::test_factory_crea_circulo ERROR [ 16%]
test_figuras_bdd.py::TestFactory::test_factory_crea_cuadrado ERROR [ 22%]
test_figuras_bdd.py::TestFactory::test_factory_valida_tipo_invalido ERROR [ 27%]
test_figuras_bdd.py::TestFactory::test_factory_valida_parametros_negativos ERROR [ 33%]
test_figuras_bdd.py::TestUnidadAdapter::test_conversion_longitud_metros_a_centimetros ERROR [ 38%]
test_figuras_bdd.py::TestUnidadAdapter::test_conversion_misma_unidad ERROR [ 44%]
test_figuras_bdd.py::TestUnidadAdapter::test_validacion_unidad_invalida ERROR [ 50%]
test_figuras_bdd.py::TestRepositorioFiguras::test_almacenar_figura ERROR [ 55%]
test_figuras_bdd.py::TestRepositorioFiguras::test_multiples_figuras_coexisten ERROR [ 61%]
test_figuras_bdd.py::TestRepositorioFiguras::test_eliminar_figura ERROR [ 66%]
test_figuras_bdd.py::TestRepositorioFiguras::test_eliminar_figura_inexistente ERROR [ 72%]
test_figuras_bdd.py::TestRepositorioFiguras::test_persistencia_guardar ERROR [ 77%]
test_figuras_bdd.py::TestRepositorioFiguras::test_persistencia_cargar ERROR [ 83%]
test_figuras_bdd.py::TestValidacionEntradas::test_radio_negativo_circulo ERROR [ 88%]
test_figuras_bdd.py::TestValidacionEntradas::test_lado_cero_cuadrado ERROR [ 94%]
test_figuras_bdd.py::TestValidacionEntradas::test_tipo_dato_incorrecto ERROR [100%]
```

```
Escenario: Calcular volumen de figuras 3D # features/figuras.feature:44
  Dado que tengo un sistema de figuras geométricas inicializado # features/steps/figuras_steps.py:41 0.031s
  Y que tengo un repositorio de figuras vacío # features/steps/figuras_steps.py:52 0.006s
  Dado que tengo un cubo con lado 3.0 # features/steps/figuras_steps.py:128 0.008s
  Cuando calculo el volumen del cubo # features/steps/figuras_steps.py:178 0.005s
  Entonces el volumen debe ser 27.0 # features/steps/figuras_steps.py:234 0.000s
  Y dado que tengo una esfera con radio 3.0 # None
  Cuando calculo el volumen de la esfera # None
  Entonces el volumen debe ser aproximadamente 113.10 # None
```

```
Escenario: Almacenar y recuperar figuras del repositorio # features/figuras.feature:52
  Dado que tengo un sistema de figuras geométricas inicializado # features/steps/figuras_steps.py:41 0.046s
  Y que tengo un repositorio de figuras vacío # features/steps/figuras_steps.py:52 0.009s
  Dado que tengo un repositorio de figuras # features/steps/figuras_steps.py:256 0.000s
  Y que creo un círculo con radio 2.5 # features/steps/figuras_steps.py:264 0.012s
  Cuando almaceno la figura en el repositorio # features/steps/figuras_steps.py:271 0.010s
  Entonces la figura debe ser almacenada con un ID válido # features/steps/figuras_steps.py:279 0.000s
  Y cuando busco la figura por su ID # None
  Entonces debo poder recuperar la figura correctamente # None
  Y la figura recuperada debe tener radio 2.5 # None
```

```
Escenario: Convertir unidades de medida # features/figuras.feature:61
  Dado que tengo un sistema de figuras geométricas inicializado # features/steps/figuras_steps.py:41 0.027s
  Y que tengo un repositorio de figuras vacío # features/steps/figuras_steps.py:52 0.010s
  Dado que tengo un valor de 100.0 en centímetros # features/steps/figuras_steps.py:318 0.023s
  Cuando convierto el valor a metros # features/steps/figuras_steps.py:326 0.011s
  Entonces el resultado debe ser 1.0 metros # features/steps/figuras_steps.py:335 0.000s
  Y cuando convierto 1.0 metros a milímetros # None
  Entonces el resultado debe ser 1000.0 milímetros # None
```

```

Escenario: Convertir unidades de medida                                # features/figuras.feature:61
  Dado que tengo un sistema de figuras geométricas inicializado      # features/steps/figuras_steps.py:41 0.025s
  Y que tengo un repositorio de figuras vacío                         # features/steps/figuras_steps.py:52 0.005s
  Dado que tengo un valor de 100.0 en centímetros                    # features/steps/figuras_steps.py:318 0.011s
  Cuando convierto el valor a metros                                # features/steps/figuras_steps.py:326 0.005s

```

```

Escenario: Almacenar y recuperar figuras del repositorio            # features/figuras.feature:52
  Dado que tengo un sistema de figuras geométricas inicializado      # features/steps/figuras_steps.py:41 0.024s
  Y que tengo un repositorio de figuras vacío                         # features/steps/figuras_steps.py:52 0.005s
  Dado que tengo un repositorio de figuras                          # features/steps/figuras_steps.py:256 0.000s
  Y que creo un círculo con radio 2.5                               # features/steps/figuras_steps.py:264 0.005s
  Cuando almaceno la figura en el repositorio                       # features/steps/figuras_steps.py:271 0.006s
  Entonces la figura debe ser almacenada con un ID válido           # features/steps/figuras_steps.py:279 0.000s
  Y cuando busco la figura por su ID                                # features/steps/figuras_steps.py:494 0.005s
  Entonces debo poder recuperar la figura correctamente             # features/steps/figuras_steps.py:294 0.000s
  Y la figura recuperada debe tener radio 2.5                       # features/steps/figuras_steps.py:303 0.000s

```

```

Escenario: Calcular volumen de figuras 3D                          # features/figuras.feature:44
  Dado que tengo un sistema de figuras geométricas inicializado      # features/steps/figuras_steps.py:41 0.025s
  Y que tengo un repositorio de figuras vacío                         # features/steps/figuras_steps.py:52 0.006s
  Dado que tengo un cubo con lado 3.0                               # features/steps/figuras_steps.py:128 0.006s
  Cuando calculo el volumen del cubo                                # features/steps/figuras_steps.py:178 0.005s
  Entonces el volumen debe ser 27.0                                  # features/steps/figuras_steps.py:234 0.000s
  Y dado que tengo una esfera con radio 3.0                         # features/steps/figuras_steps.py:482 0.006s
  Y cuando calculo el volumen de la esfera                          # features/steps/figuras_steps.py:488 0.005s
  Entonces el volumen debe ser aproximadamente 113.10              # features/steps/figuras_steps.py:243 0.000s

```

```

1 feature passed, 0 failed, 0 skipped
11 escenarios passed, 0 failed, 0 skipped
80 steps passed, 0 failed, 0 skipped

```

En las siguientes figuras se presentan los escenarios definidos con el enfoque BDD. Cada escenario está redactado en términos de Dado–Cuando–Entonces, representando el comportamiento esperado del sistema. Inicialmente las pruebas fallaron (fase Rojo), luego fueron implementadas hasta pasar exitosamente (fase Verde), y finalmente se realizó refactorización del código manteniendo las pruebas en verde (fase Azul).

Analizadores Estático De Código

```
PS C:\Users\sebas\Desktop\Escritorio\Semdocker-compose run --rm python-app pylint Figuras/main.py
***** Module main
Figuras/main.py:9:0: C0303: Trailing whitespace (trailing-whitespace)
Figuras/main.py:15:0: C0303: Trailing whitespace (trailing-whitespace)
Figuras/main.py:4:0: E0402: Attempted relative import beyond top-level package (relative-beyond-top-level)
Figuras/main.py:7:0: R0903: Too few public methods (1/2) (too-few-public-methods)
Figuras/main.py:21:4: W0104: Statement seems to have no effect (pointless-statement)

-----
Your code has been rated at 0.00/10
```

```
PS C:\Users\sebas\Desktop\Escritorio\Semestre_4\Técnicas avanzadas de programacion\TallerCorte2> docker-compose run --rm python-app pylint Figuras/Figura.py
***** Module Figura
Figuras/Figura.py:9:0: C0303: Trailing whitespace (trailing-whitespace)
Figuras/Figura.py:22:0: C0303: Trailing whitespace (trailing-whitespace)
Figuras/Figura.py:26:0: C0303: Trailing whitespace (trailing-whitespace)
Figuras/Figura.py:30:0: C0303: Trailing whitespace (trailing-whitespace)
Figuras/Figura.py:34:0: C0303: Trailing whitespace (trailing-whitespace)
Figuras/Figura.py:38:0: C0303: Trailing whitespace (trailing-whitespace)
Figuras/Figura.py:41:0: C0304: Final newline missing (missing-final-newline)
Figuras/Figura.py:1:0: C0103: Module name "Figura" doesn't conform to snake_case naming style (invalid-name)
Figuras/Figura.py:4:0: W0611: Unused abstractmethod imported from abc (unused-import)

-----
Your code has been rated at 4.38/10
```

```
PS C:\Users\sebas\Desktop\Escritorio\Semestre_4\Técnicas avanzadas de programacion\TallerCorte2> docker-compose run --rm python-app pylint Figuras/Figura.py
***** Module Figura
Figuras/Figura.py:41:0: C0304: Final newline missing (missing-final-newline)
Figuras/Figura.py:1:0: C0103: Module name "Figura" doesn't conform to snake_case naming style (invalid-name)

-----
Your code has been rated at 8.75/10
```

```
-----
Your code has been rated at 3.25/10
```

```
-----
Your code has been rated at 9.44/10
```

Para reforzar la calidad del software, se aplicó un analizador estático sobre el código fuente. Este proceso permitió identificar errores y malas prácticas de programación antes de la ejecución.

Contenedores Docker Para Facilitar El Desarrollo

```
PS C:\Users\sebas\Desktop\Escritorio\Semdocker-compose run --rm python-app pylint Figuras/main.py
***** Module main
Figuras/main.py:9:0: C0303: Trailing whitespace (trailing-whitespace)
Figuras/main.py:15:0: C0303: Trailing whitespace (trailing-whitespace)
Figuras/main.py:4:0: E0402: Attempted relative import beyond top-level package (relative-beyond-top-level)
Figuras/main.py:7:0: R0903: Too few public methods (1/2) (too-few-public-methods)
Figuras/main.py:21:4: W0104: Statement seems to have no effect (pointless-statement)

-----
Your code has been rated at 0.00/10
```

```
PS C:\Users\sebas\Desktop\Escritorio\Semestre_4\Técnicas avanzadas de programación\TallerCorte2> docker-compose run --rm python-app pylint Figuras/Figura.py
***** Module Figura
Figuras/Figura.py:9:0: C0303: Trailing whitespace (trailing-whitespace)
Figuras/Figura.py:22:0: C0303: Trailing whitespace (trailing-whitespace)
Figuras/Figura.py:26:0: C0303: Trailing whitespace (trailing-whitespace)
Figuras/Figura.py:30:0: C0303: Trailing whitespace (trailing-whitespace)
Figuras/Figura.py:34:0: C0303: Trailing whitespace (trailing-whitespace)
Figuras/Figura.py:38:0: C0303: Trailing whitespace (trailing-whitespace)
Figuras/Figura.py:41:0: C0304: Final newline missing (missing-final-newline)
Figuras/Figura.py:1:0: C0103: Module name "Figura" doesn't conform to snake_case naming style (invalid-name)
Figuras/Figura.py:4:0: W0611: Unused abstractmethod imported from abc (unused-import)

-----
Your code has been rated at 4.38/10
```

```
PS C:\Users\sebas\Desktop\Escritorio\Semestre_4\Técnicas avanzadas de programación\TallerCorte2> docker-compose run --rm python-app pylint Figuras/Figura.py
***** Module Figura
Figuras/Figura.py:41:0: C0304: Final newline missing (missing-final-newline)
Figuras/Figura.py:1:0: C0103: Module name "Figura" doesn't conform to snake_case naming style (invalid-name)

-----
Your code has been rated at 8.75/10
```

```
-----
Your code has been rated at 3.25/10
```

```
-----
Your code has been rated at 9.44/10
```

Medición

Ejecución del código

Una vez implementado el sistema, se realizaron pruebas de validación para asegurar que cada figura calcule correctamente sus propiedades geométricas. Se usaron múltiples entradas y se verificaron manualmente los resultados esperados para:

```
=====
i Figuras almacenadas: 3 | Unidad: m
```

1. Crear nueva figura
 2. Consultar todas las figuras
 3. Consultar figura por ID
 4. Eliminar figura
 5. Ver estadísticas
 6. Cambiar unidad de medida
 7. Limpiar repositorio
 8. Salir
- ```
=====
```

```
=====
Seleccione una opción: 1
```

```
=====
 CREAR NUEVA FIGURA
=====
```

```
Tipos de figuras disponibles:
```

1. Circulo
2. Cuadrado
3. Cubo
4. Esfera

```
Seleccione una opción: 2
```

```
Ingrese el lado del cuadrado (en m): 5
```


```
✓ Figura creada exitosamente con ID: 6
```


```
▴ Figura ID: 6
 Tipo: Cuadrado
 Categoría: 2D
 Lado: 5.00 m
 Área: 25.00 m²
 Perímetro: 20.00 m
```


```
Presione Enter para continuar...|
```


```
=====
Seleccione una opción: 2
=====
```


```
TODAS LAS FIGURAS
=====
```

```
 Total de figuras: 4
=====
```

```
1.  Figura ID: 2
 Tipo: Cuadrado
 Categoría: 2D
 Lado: 4.00 m
 Área: 16.00 m2
 Perímetro: 16.00 m
```

```
2.  Figura ID: 3
 Tipo: Cuadrado
 Categoría: 2D
 Lado: 2.00 m
 Área: 4.00 m2
 Perímetro: 8.00 m
```

```
3.  Figura ID: 5
 Tipo: Círculo
 Categoría: 2D
 Radio: 2.00 m
 Área: 12.57 m2
 Perímetro: 12.57 m
```

```
4.  Figura ID: 6
 Tipo: Cuadrado
 Categoría: 2D
 Lado: 5.00 m
 Área: 25.00 m2
 Perímetro: 20.00 m
```

```
=====
Seleccione una opción: 3
=====
```

```
CONSULTAR FIGURA POR ID
=====
```

```
Ingrese el ID de la figura: 2
```

```
▴ Figura ID: 2
 Tipo: Cuadrado
 Categoría: 2D
 Lado: 4.00 m
 Área: 16.00 m²
 Perímetro: 16.00 m
```

```
Ingrese el ID de la figura a eliminar: 5
```

```
Figura a eliminar:
```

```
▴ Figura ID: 5
 Tipo: Círculo
 Categoría: 2D
 Radio: 2.00 m
 Área: 12.57 m²
 Perímetro: 12.57 m
```

```
¿Está seguro de que desea eliminar esta figura? (s/n): S
```

```
✅ Figura eliminada exitosamente.
```



```
=====
Seleccione una opción: 5
```

```
=====
ESTADÍSTICAS
```

```
=====
📊 ESTADÍSTICAS DEL REPOSITORIO
```

```
=====
Total de figuras: 3
```

```
Distribución por tipo:
```

- Cuadrado: 3 (100.0%)

```
Presione Enter para continuar...|
```

```
=====
Seleccione una opción: 6
```

```
=====
CAMBIAR UNIDAD DE MEDIDA
```

```
=====
Unidad actual: m
```

```
Unidades disponibles:
```

1. m - metros
2. cm - centímetros
3. mm - milímetros
4. in - pulgadas
5. ft - pies

```
Seleccione una opción: 3
```

```
✅ Unidad cambiada a: mm
```

```
=====
 CREAR NUEVA FIGURA
=====

Tipos de figuras disponibles:

1. Circulo
2. Cuadrado
3. Cubo
4. Esfera
Seleccione una opción: 5
Por favor, seleccione un número entre 1 y 4
Seleccione una opción: 4

Ingrese el radio del esfera (en m): hola
Por favor, ingrese un número válido.

Ingrese el radio del esfera (en m): |
```

```
Ingrese el radio del esfera (en m): -0.001
El valor debe ser mayor o igual a 0.001
```

```
=====
 SISTEMA DE FIGURAS GEOMÉTRICAS
=====
i Figuras almacenadas: 4 | Unidad: m

1. Crear nueva figura
2. Consultar todas las figuras
3. Consultar figura por ID
4. Eliminar figura
5. Ver estadísticas
6. Cambiar unidad de medida
7. Limpiar repositorio
8. Salir
=====

Seleccione una opción: 9
El valor debe ser menor o igual a 8

Seleccione una opción: |
```

```
=====
 SISTEMA DE FIGURAS GEOMÉTRICAS
=====
i Figuras almacenadas: 5 | Unidad: m

1. Crear nueva figura
2. Consultar todas las figuras
3. Consultar figura por ID
4. Eliminar figura
5. Ver estadísticas
6. Cambiar unidad de medida
7. Limpiar repositorio
8. Salir
=====

Seleccione una opción: 3

=====
CONSULTAR FIGURA POR ID
=====
Ingrese el ID de la figura: 20
❌ Error: No se encontró una figura con ID: 20
```

El sistema cumplió satisfactoriamente con los objetivos planteados y demostró ser funcional, eficiente y mantenible.