

Pruebas de Software



TP Integrador 2020

Docente a cargo: Eduardo Diez

Instructor Ayudante: Pablo San Román

Alumno: Oscar Ruina

DNI: 35639606

Link al código de los test: <https://github.com/OscarRuina/TP-pruebas-de-software-2020>

2-Planilla de especificación de casos de prueba

Test Unitarios.

ID	Nombre	Condiciones de Entrada	Pre-Condición	Resultado Esperado	Resultado
1	testGetValordeDificultad	-Objeto Dificultad creado. -3 variables tipo entero creadas	Inicializo las tres variables con los valores 10,15 y 7 Instancio el objeto dificultad pasando como parámetro las variables.	True, al verificar que el método getValorDeDificultad() devuelva un valor igual a la suma de las tres variables anteriores.	True
2	testIgualA	-2 Objetos de tipo Posición creados. -4 variables numéricas creadas	-inicializo las variables (los pares x y con igual valor en las 4 variables). -Instancio los objetos de tipo Posición con las variables.	True, al verificar que el método igualA() devuelve true.	True
3	testDistanciaHasta	-2 Objetos de tipo Posición creados. -5 variables numéricas creadas	-Inicializo las variables, calculo la distancia entre las variables. -Instancio los objetos de tipo Posición con las variables.	True, al verificar que el método distanciaHasta() devuelve un valor igual a la variable distancia calculada anteriormente	True
4	testPuedeAterrizarHelicoptero	-Objetos Posición, Helipuerto, Helicóptero creados. -2 variables numéricas creadas	-inicializar las variable con valores numéricos. -instanciar los objetos de tipo Posición y Helipuerto.	True, al verificar que el método puedeAterrizarHelicoptero() devuelve true al ser un objeto tipo Helicoptero el que puede aterrizar.	True

5	testPuedeAterrizarAvionSimple	-Objetos Posición, Helipuerto, Avión Simple creados. -2 variables numéricas creadas	-inicializar las variable con valores numéricos. -instanciar los objetos de tipo Posición y Helipuerto.	False, al verificar que el método puedeAterrizarAvionSimple() devuelve false al ser un objeto tipo AvionSimple el que no puede aterrizar.	False
---	-------------------------------	--	--	---	-------

Test de Integración.

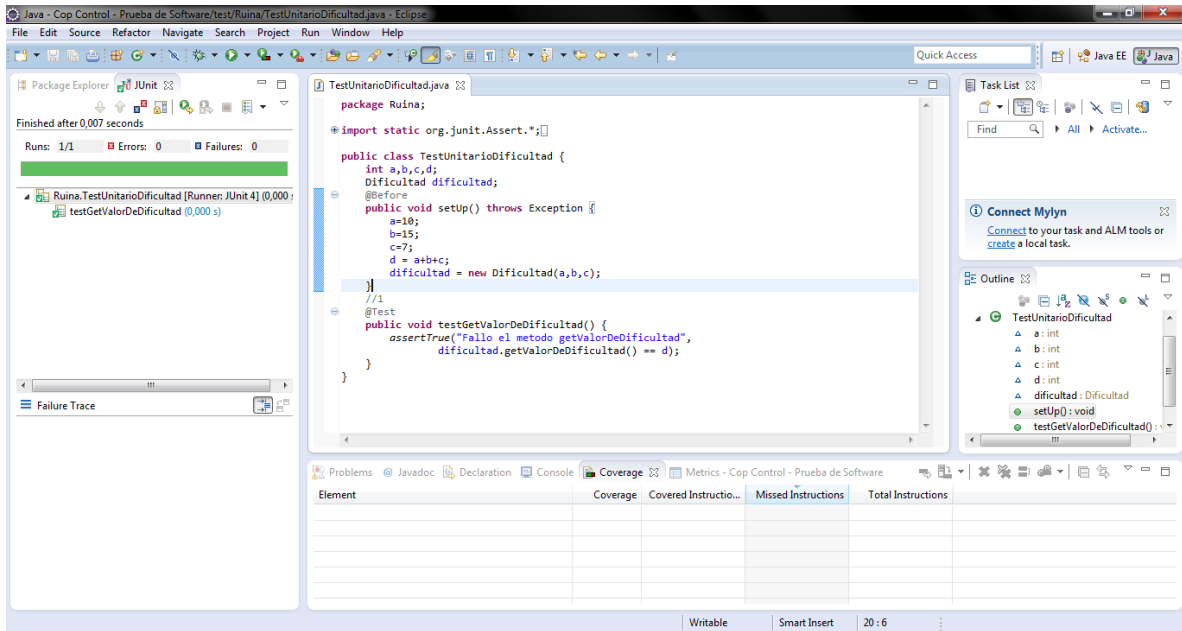
ID	Nombre	Condiciones de Entrada	Pre-Condición	Resultado Esperado	Resultado
6	testEstaEnZonaDeAterrizaje	-Objetos de Posicion, Mapa, PistaDoble y AvionSimple creados	Inicializar los objetos creados	True, el método verifica que el objeto Avion está en la zona de aterrizaje	True
7	testPuesAterrizarAvionSimple	-Objetos de Posicion, Mapa, y AvionSimple creados	Inicializar los objetos	True, el método verifica que el objeto Avion puede aterrizar en una pista	True
8	testAterrizarAviones	-Objetos de clase creados: Mapa – Dificultad – Posición - Avión – Pista – Lista de Pistas.	Inicializo los objetos, agrego una pista a la lista de pistas, y coloco aviones en el aire	True, el método aterrizarAviones() devuelve un 2 que es la cantidad de aviones que puse en el aire	True
9	testTienePistaAdecuada	-Objetos de clase creados: Mapa – Dificultad – Posición - Avión – Pista – Lista de Pistas.	Inicializo los objetos, agrego una pista a la lista de pistas, y coloco aviones en el aire, la pista y el avión son compatibles a la hora de aterrizar	True, el método tienePistaAdecuada() devuelve true si el avión a aterrizar puede hacerlo en la pistaSimple	True
10	testTieneAvionesVolando	-Objetos de clase creados: Mapa – Dificultad – Posición - Avión – Pista – Lista de Pistas.	Inicializo los objetos, coloco aviones en el aire	True, el método tieneAvionesVolando() devuelve true ya que anteriormente coloque dos aviones en el aire	True

3-Captura de Pantalla de los resultados

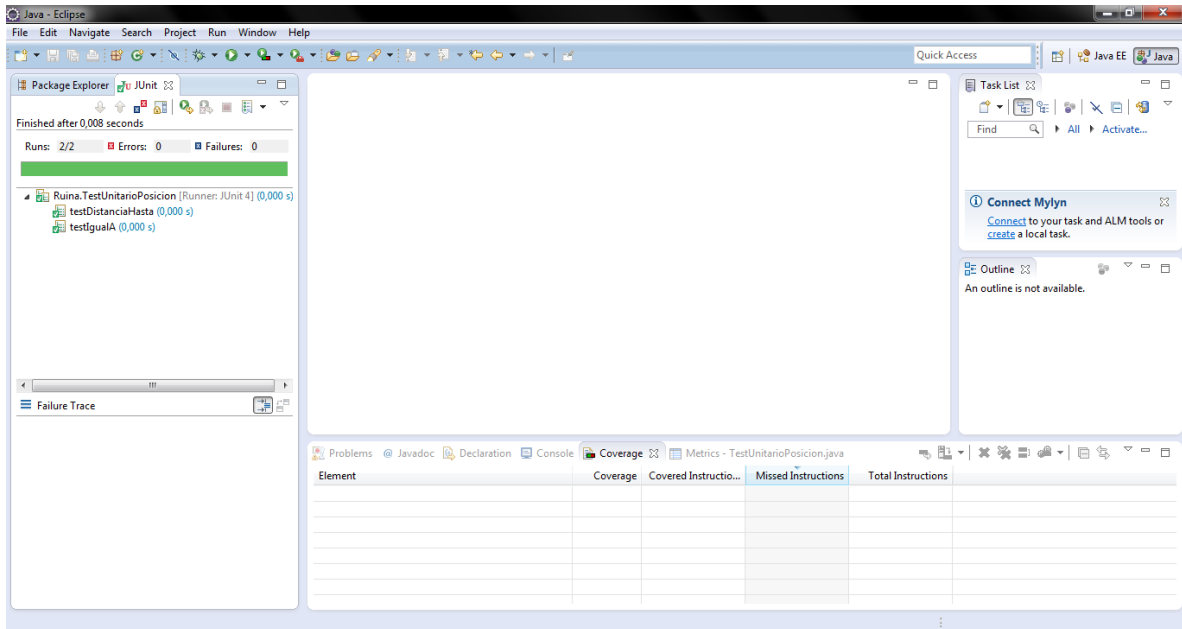
Coloco el resultado del test más el código.

Pruebas de Unidad:

Test 1:



Test 2 y Test 3:



```

Java - Cop Control - Prueba de Software/test/Ruina/TestUnitarioPosicion.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

package Ruina;

import static org.junit.Assert.*;

public class TestUnitarioPosicion {
    Posicion posicion;
    Posicion otraPosicionIgual;
    Posicion otraPosicionDistancia;
    int i,j,k,l,m,n;
    double distancia;

    @Before
    public void setUp() throws Exception {
        i = 5;
        j = 5;
        k = 5;
        l = 5;
        posicion = new Posicion(i,j);
        otraPosicionIgual = new Posicion(k,l);
        m = 10;
        n = 15;
        otraPosicionDistancia = new Posicion(m,n);
        distancia = Math.hypot(Math.abs(otraPosicionDistancia.getCoordenadaY() - posicion.getCoordenadaY()),
            Math.abs(otraPosicionDistancia.getCoordenadaX() - posicion.getCoordenadaX()));
    }

    //2
    @Test
    public void testIgualA() {
        assertTrue("Fallo el metodo igualA(Posicion otraPosicion)",
            posicion.igualA(otraPosicionIgual) == true);
    }

    //3
    @Test
    public void testDistanciaHasta(){
        assertTrue("Fallo el metodo distanciaHasta(Posicion otraPosicion)",
            posicion.distanciaHasta(otraPosicionDistancia) == distancia);
    }
}

```

Test 4 y 5

Finished after 0,009 seconds

Runs: 2/2 Errors: 0 Failures: 0

Ruina.TestUnitarioHelipuerto [Runner: JUnit 4] (0,000 s)

- testPuedeAterrizajeHelicoptero (0,000 s)
- testPuedeAterrizajeAvionSimple (0,000 s)

Failure Trace

Task List

Find All Activate...

Connect Mylyn

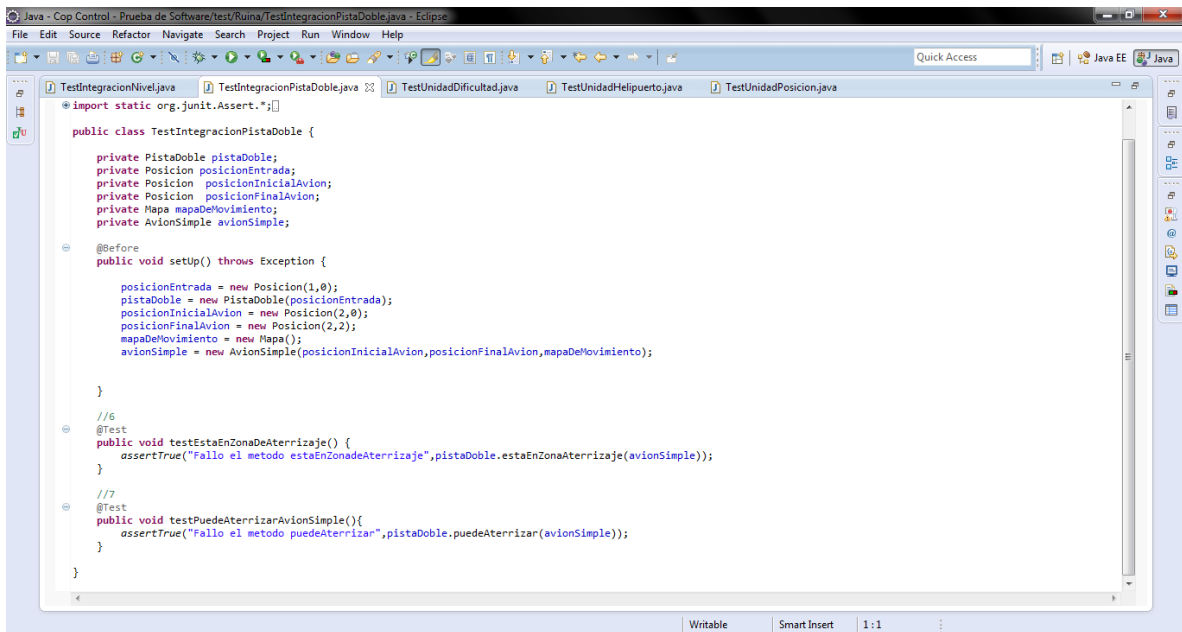
Connect to your task and ALM tools or create a local task.

Outline

An outline is not available.

Problems Javadoc Declaration Console Coverage Metrics - Cop Control - Prueba de Software - McCabe Cyclomatic Complexity (avg/max per m...)

Element	Coverage	Covered Instruction...	Missed Instructions	Total Instructions



```
import static org.junit.Assert.*;

public class TestIntegracionPistaDoble {

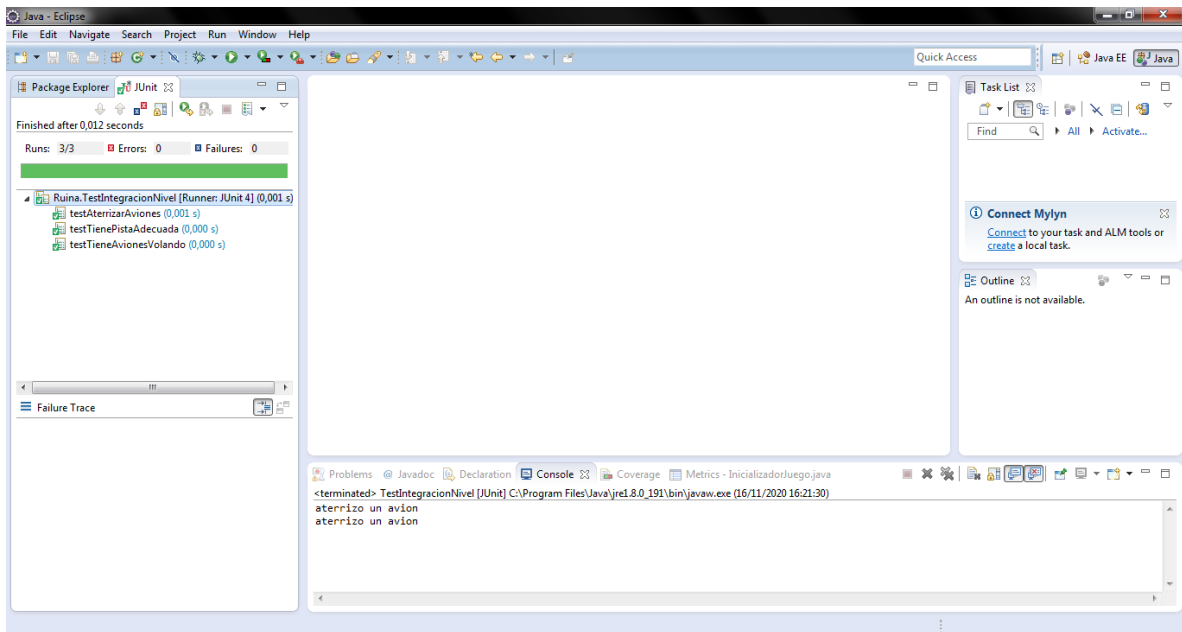
    private PistaDoble pistaDoble;
    private Posicion posicionEntrada;
    private Posicion posicionInicialAvion;
    private Posicion posicionFinalAvion;
    private Mapa mapaDeMovimiento;
    private AvionSimple avionSimple;

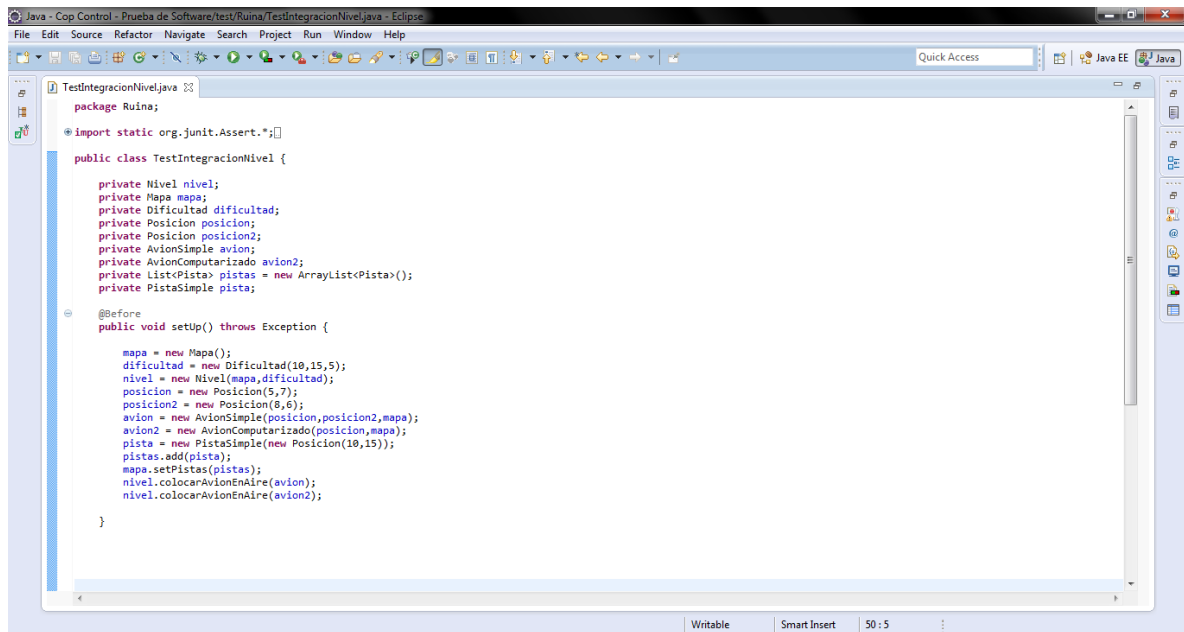
    @Before
    public void setUp() throws Exception {
        posicionEntrada = new Posicion(1,0);
        pistaDoble = new PistaDoble(posicionEntrada);
        posicionInicialAvion = new Posicion(2,0);
        posicionFinalAvion = new Posicion(2,2);
        mapaDeMovimiento = new Mapa();
        avionSimple = new AvionSimple(posicionInicialAvion,posicionFinalAvion,mapaDeMovimiento);
    }

    //6
    @Test
    public void testEstaEnZonaDeAterrizaje() {
        assertTrue("Fallo el metodo estaEnZonadeAterrizaje",pistaDoble.estaEnZonaAterrizaje(avionSimple));
    }

    //7
    @Test
    public void testPuedeAterrizaraAvionSimple(){
        assertTrue("Fallo el metodo puedeAterrizar",pistaDoble.puedeAterrizar(avionSimple));
    }
}
```

Test 8,9 y 10





```
package Ruina;

import static org.junit.Assert.*;

public class TestIntegracionNivel {

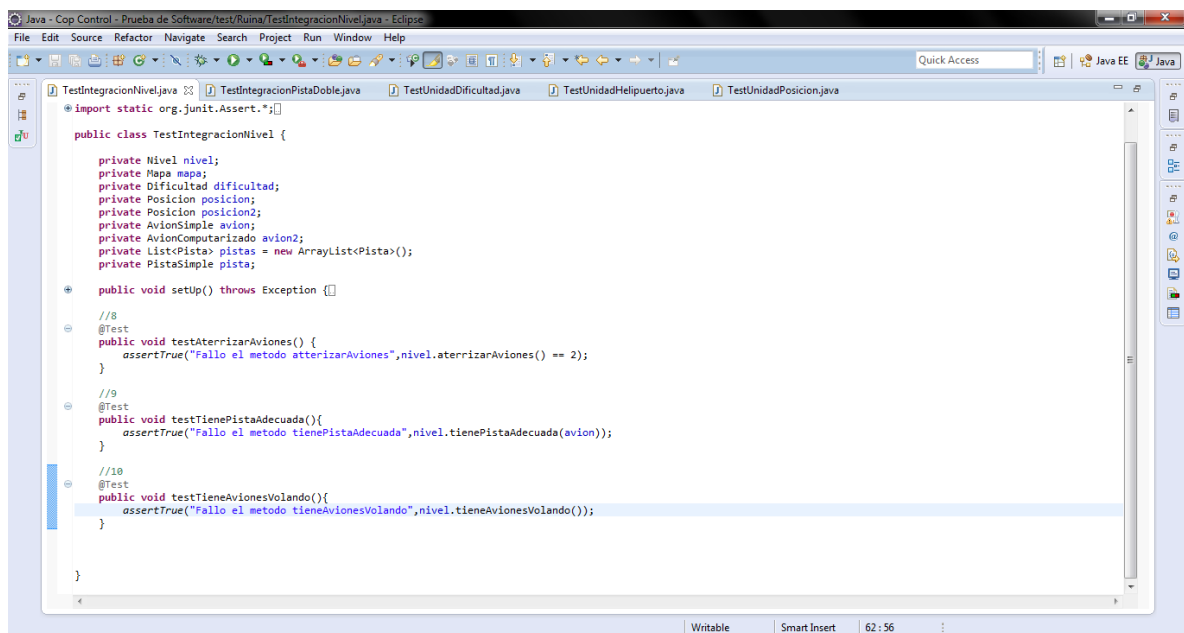
    private Nivel nivel;
    private Mapa mapa;
    private Dificultad dificultad;
    private Posicion posicion;
    private Posicion posicion2;
    private AvionSimple avion;
    private AvionComputarizado avion2;
    private List<Pista> pistas = new ArrayList<Pista>();
    private PistaSimple pista;

    @Before
    public void setUp() throws Exception {

        mapa = new Mapa();
        dificultad = new Dificultad(10,15,5);
        nivel = new Nivel(mapa,dificultad);
        posicion = new Posicion(5,7);
        posicion2 = new Posicion(8,6);
        avion = new AvionSimple(posicion,posicion2,mapa);
        avion2 = new AvionComputarizado(posicion,mapa);
        pista = new PistaSimple(new Posicion(10,15));
        pistas.add(pista);
        mapa.setPistas(pistas);
        nivel.colocarAvionEnAire(avion);
        nivel.colocarAvionEnAire(avion2);

    }

}
```



```
package Ruina;

import static org.junit.Assert.*;

public class TestIntegracionNivel {

    private Nivel nivel;
    private Mapa mapa;
    private Dificultad dificultad;
    private Posicion posicion;
    private Posicion posicion2;
    private AvionSimple avion;
    private AvionComputarizado avion2;
    private List<Pista> pistas = new ArrayList<Pista>();
    private PistaSimple pista;

    @Before
    public void setUp() throws Exception {}

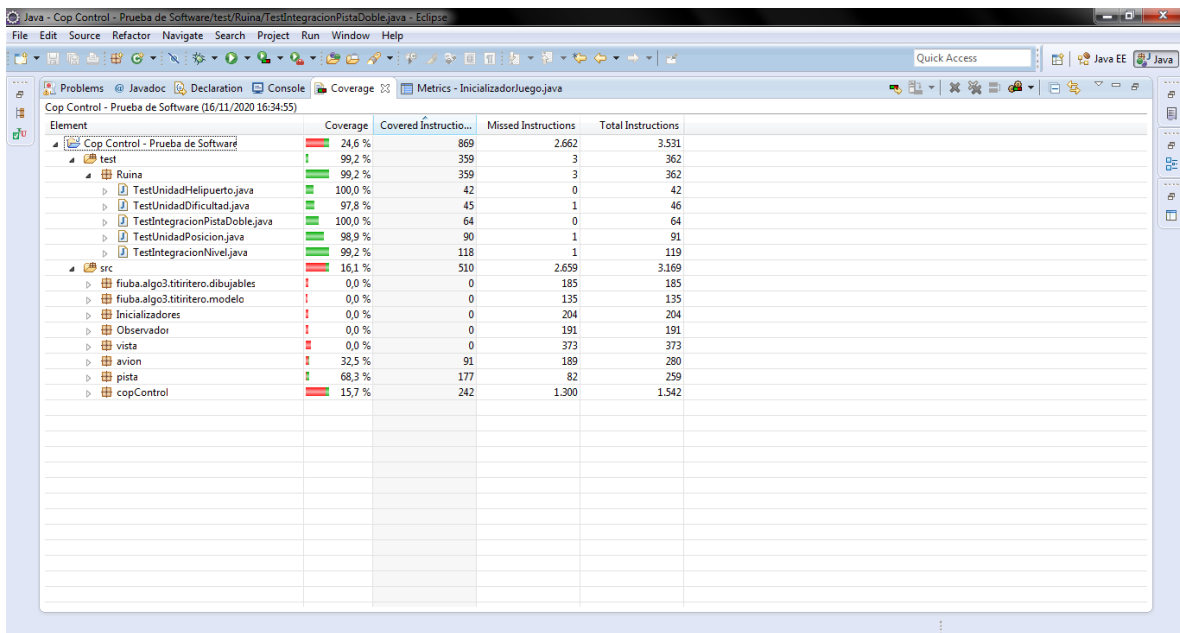
    //8
    @Test
    public void testAterrizzarAviones() {
        assertTrue("Fallo el metodo aterrizzarAviones",nivel.aterrizzarAviones() == 2);
    }

    //9
    @Test
    public void testTienePistaAdecuada(){
        assertTrue("Fallo el metodo tienePistaAdecuada",nivel.tienePistaAdecuada(avion));
    }

    //10
    @Test
    public void testTieneAvionesVolando(){
        assertTrue("Fallo el metodo tieneAvionesVolando",nivel.tieneAvionesVolando());
    }

}
```


4- Porcentaje del código del modelo



Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
Cop Control - Prueba de Software	24,6 %	869	2.662	3.531
test	99,2 %	359	3	362
Ruina	99,2 %	359	3	362
TestUnidadHelipuerto.java	100,0 %	42	0	42
TestUnidadDificultad.java	97,8 %	45	1	46
TestIntegracionPistaDoble.java	100,0 %	64	0	64
TestUnidadPosicion.java	98,9 %	90	1	91
TestIntegracionNivel.java	99,2 %	118	1	119
src	16,1 %	510	2.659	3.169
fruba.algo3.titirtero.dibujables	0,0 %	0	185	185
fruba.algo3.titirtero.modelo	0,0 %	0	135	135
Inicializadores	0,0 %	0	204	204
Observador	0,0 %	0	191	191
vista	0,0 %	0	373	373
avion	32,5 %	91	189	280
pista	68,3 %	177	82	259
copControl	15,7 %	242	1.300	1.542

5- Metodos con más de 24 líneas de código

En la Clase **Jugador** el método **public void moverHacia(int x, int y)** -> 30 líneas de código.

En la Clase **Posición** el método **private List<Posicion> getVecinos()** -> 24 líneas de código.

En la Clase **Mapa** el método **public Posicion generarPosicionExtremoAlAzar()** -> 24 líneas de código.

6- Metodos con más de 4 parámetros

En la Clase **Nivel** el método **private boolean intersects(double x1, double y1, double r1, double x2, double y2, double r2)** -> tiene 6 parámetros.

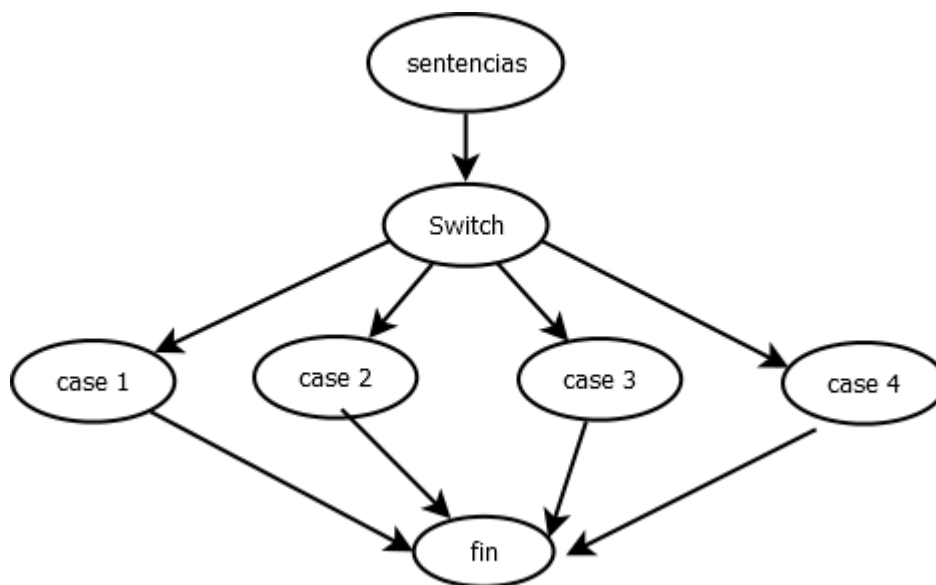
En la Clase **Pista** el método **protected boolean intersects(double x1, double y1, double r1, double x2, double y2, double r2)** -> tiene 6 parámetros.

7-Metodos con complejidad ciclomatica igual 5

Formula de Complejidad Ciclomatica : $V(G) = \text{Aristas} - \text{Nodos} + 2$

En la Clase Mapa el método `public void generarPosicionExtremaAlAzar()` .

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
Method Lines of Code (avg/max per method)	867	3,387	4,455	30	/Cop Control - Prueba de Software/src/copControl/I...	moverHacia
Number of Parameters (avg/max per method)		0,57	0,868	6	/Cop Control - Prueba de Software/src/copControl/NL...	intersects
McCabe Cyclomatic Complexity (avg/max per method)		1,297	0,913	8	/Cop Control - Prueba de Software/src/copControl/I...	moverHacia
src		1,315	0,938	8	/Cop Control - Prueba de Software/src/copControl/I...	moverHacia
copControl		1,646	1,387	8	/Cop Control - Prueba de Software/src/copControl/I...	moverHacia
Jugador.java		2,75	3,031	8	/Cop Control - Prueba de Software/src/copControl/I...	moverHacia
Trayectoria.java		2,1	1,64	6	/Cop Control - Prueba de Software/src/copControl/T...	generarPosicionExtremo...
Mapa.java		1,462	1,151	5	/Cop Control - Prueba de Software/src/copControl/I...	generarPosicionExtremo...
Mapa		1,462	1,151	5	/Cop Control - Prueba de Software/src/copControl/I...	generarPosicionExtremo...
generarPosicionExtremoAlAzar	5					
getPosPistaAdecuada	3					
Mapa	1					
Mapa	1					
getAvionesEnAire	1					
getPistas	1					
setPistas	1					
tieneAvionesVolando	1					
getPosicionesExtremos	1					
colocarAvionEnAire	1					
getDimension	1					
getPosicionAleatoria	1					
esPosicionExtremo	1					
Juego.java	1,789	1,321		5	/Cop Control - Prueba de Software/src/copControl/I...	crearAvionAlAzar
Nivel.java	1,6	1,114		5	/Cop Control - Prueba de Software/src/copControl/NL...	chocoConOtroAvion
Dificultad.java	1	0		1	/Cop Control - Prueba de Software/src/copControl/D...	Dificultad
Posicion.java	1	0		1	/Cop Control - Prueba de Software/src/copControl/P...	Posicion
Observador	1,9	1,578		5	/Cop Control - Prueba de Software/src/observador/A...	nuevaVistaDePista
fruba.algo3.tiritero.modelo	1,133	0,499		3	/Cop Control - Prueba de Software/src/fruba/algo3/tl...	run
Inicializadores	1,273	0,445		2	/Cop Control - Prueba de Software/src/inicializadores...	pistaLargainicializada

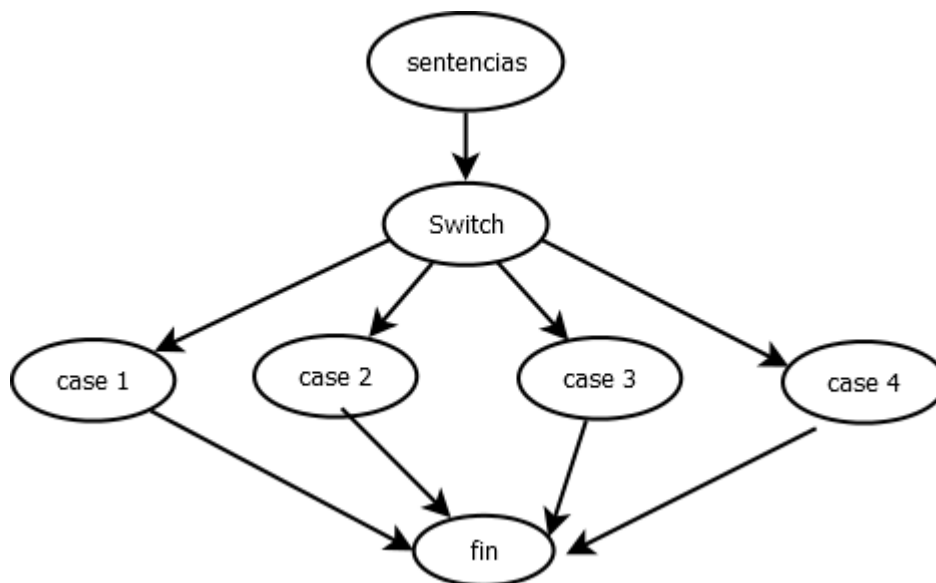


$$V(G) = 9 - 7 + 2 = 4$$

En la Clase **Juego** el método **private Avion crearAvionAlAzar()** .

The screenshot shows the Eclipse IDE with the 'Metrics' view open. The table displays McCabe Cyclomatic Complexity metrics for various classes and methods. The 'Juego' class is highlighted, showing a complexity of 1,789. The 'crearAvionAlAzar' method is also highlighted, showing a complexity of 5.

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
Number of Parameters (avg/max per method)	0,57	0,868		6	/Cop Control - Prueba de Software/src/copControl/N...	intersects
McCabe Cyclomatic Complexity (avg/max per method)	1,297	0,913		8	/Cop Control - Prueba de Software/src/copControl/J...	moverHacia
src	1,315	0,938		8	/Cop Control - Prueba de Software/src/copControl/J...	moverHacia
copControl	1,646	1,387		8	/Cop Control - Prueba de Software/src/copControl/J...	moverHacia
Jugador.java	2,75	3,031		8	/Cop Control - Prueba de Software/src/copControl/J...	moverHacia
Trayectoria.java	2,1	1,64		6	/Cop Control - Prueba de Software/src/copControl/T...	generarPosicionExtremo...
Mapa.java	1,462	1,151		5	/Cop Control - Prueba de Software/src/copControl/J...	generarPosicionExtremo...
Juego.java	1,789	1,321		5	/Cop Control - Prueba de Software/src/copControl/J...	crearAvionAlAzar
Juego	1,789	1,321		5	/Cop Control - Prueba de Software/src/copControl/J...	crearAvionAlAzar
crearAvionAlAzar	5					
colocarAvion	4					
esLuegoGanado	4					
vivir	4					
huboChoque	2					
avanzarNivel	2					
Juego	1					
getNivelActual	1					
estaJugandose	1					
getCantidadAvionesAterrizados	1					
getJugador	1					
getNiveles	1					
chequearAterrizajes	1					
actualizarMovimientoDeAvion	1					
reiniciarCiclos	1					
cantidadDeCiclos	1					
esUltimoNivel	1					
pasoUnCiclo	1					
getVelocidadActual	1					
Nivel.java	1,6	1,114		5	/Cop Control - Prueba de Software/src/copControl/N...	chocoConOtroAvion



$$V(G) = 9 - 7 + 2 = 4$$

En la Clase **Nivel** el método **private boolean chocoConOtroAvion (Avion avionActual,List<Avion> avionesEnAire) .**

Java - Cop Control - Prueba de Software/src/Observador/Adm.../ControladorDeVistas.java - Eclipse

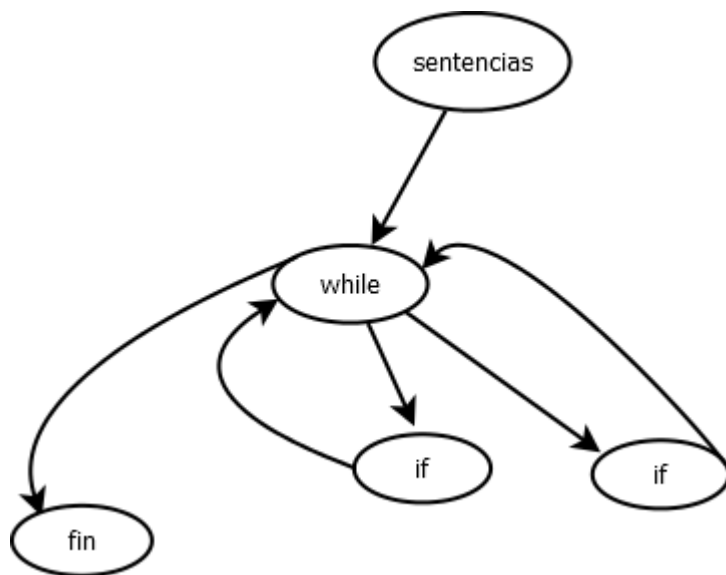
File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Java EE Java

Problems Javadoc Declaration Console Coverage Metrics - Cop Control - Prueba de Software - McCabe Cyclomatic Complexity (avg/max per method)

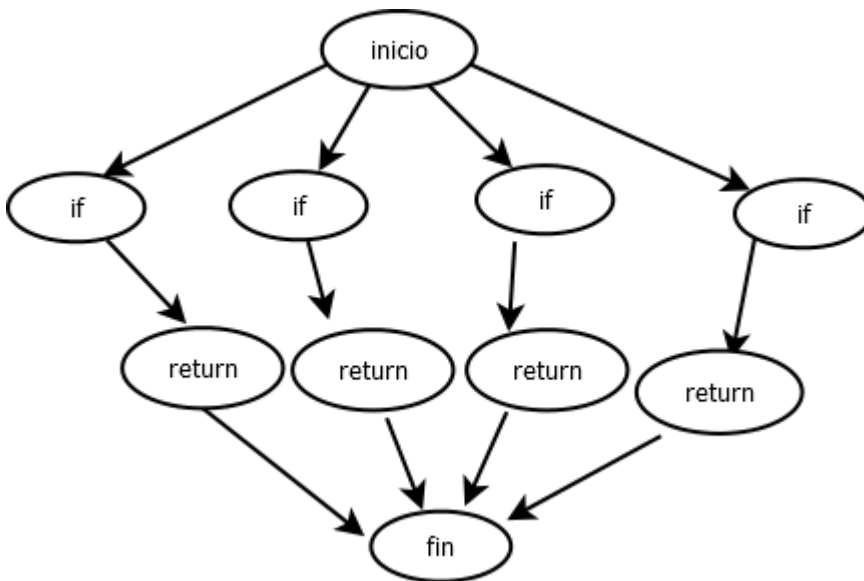
Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
src	1,315	0,938	8	/Cop Control - Prueba de Software/src/copControl/...	moverHacia	
▲ src						
copControl	1,646	1,387	8	/Cop Control - Prueba de Software/src/copControl/...	moverHacia	
> Jugador.java	2,75	3,031	8	/Cop Control - Prueba de Software/src/copControl/...	moverHacia	
> Trayectoria.java	2,1	1,64	6	/Cop Control - Prueba de Software/src/copControl/T...	generarPosicionExtremo...	
> Mapa.java	1,462	1,151	5	/Cop Control - Prueba de Software/src/copControl/...	generarPosicionExtremo...	
> Juego.java	1,789	1,321	5	/Cop Control - Prueba de Software/src/copControl/...	crearAvionAlAzar	
▲ Nivel.java	1,6	1,114	5	/Cop Control - Prueba de Software/src/copControl/NL...	chocoConOtroAvion	
▲ Nivel	1,6	1,114	5	/Cop Control - Prueba de Software/src/copControl/NL...	chocoConOtroAvion	
chocoConOtroAvion	5					
getAvionEnPosicion	3					
tienePistaAdecuada	3					
huboChoque	3					
avanzarAvionesEnAire	3					
Nivel	1					
aterizarAviones	1					
getPosicionesExtremos	1					
colocarAvionEnAire	1					
tieneAvionesVolando	1					
choqueAvion	1					
intersects	1					
getAvionesVolando	1					
getPosPistaAdecuada	1					
getMapa	1					
getFrecuenciaDeAparicionDt	1					
getCantidadDeAvionesPorAt	1					
seGano	1					
estaGanado	1					
getVelocidadDeAviones	1					
> Dificultad.java	1	0	1	/Cop Control - Prueba de Software/src/copControl/D...	Dificultad	



$$V(G) = 6 - 5 + 2 = 3$$

En la Clase **Administrador de Vistas** el método **private ObjetoDibujable nuevaVistaDePista(Pista pista)** .

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
src		1,315	0,938	8	/Cop Control - Prueba de Software/src/copControl/I...	moverHacia
copControl		1,646	1,387	8	/Cop Control - Prueba de Software/src/copControl/I...	moverHacia
Jugador.java		2,75	3,031	8	/Cop Control - Prueba de Software/src/copControl/I...	moverHacia
Trayectoria.java		2,1	1,64	6	/Cop Control - Prueba de Software/src/copControl/T...	generaPosicionExtremo...
Mapa.java		1,462	1,151	5	/Cop Control - Prueba de Software/src/copControl/I...	generaPosicionExtremo...
Juego.java		1,789	1,321	5	/Cop Control - Prueba de Software/src/copControl/I...	crearAvionAlAzar
Nivel.java		1,6	1,114	5	/Cop Control - Prueba de Software/src/copControl/NL...	chocoConOtroAvion
Dificultad.java		1	0	1	/Cop Control - Prueba de Software/src/copControl/D...	Dificultad
Posicion.java		1	0	1	/Cop Control - Prueba de Software/src/copControl/P...	Posicion
Observador		1,9	1,578	5	/Cop Control - Prueba de Software/src/Observador/A...	nuevaVistaDePista
AdministradorDeVistas.java		2,5	1,803	5	/Cop Control - Prueba de Software/src/Observador/A...	nuevaVistaDePista
AdministradorDeVistas		2,5	1,803	5	/Cop Control - Prueba de Software/src/Observador/A...	nuevaVistaDePista
nuevaVistaDePista	5					
nuevaVistaDeAvion	5					
actualizar	2					
AdministradorDeVistas	1					
actualizarVistasPistas	1					
actualizarVistasAviones	1					
Observable.java		1	0	1	/Cop Control - Prueba de Software/src/Observador/O...	Observable
Observable.java		0	0			
fiuba.algo3.titirero.modelo		1,133	0,499	3	/Cop Control - Prueba de Software/src/fiuba/algo3/ti...	run
Inicializadores		1,273	0,445	2	/Cop Control - Prueba de Software/src/inicializadores...	pistaLargaIniciada
avion		1,056	0,229	2	/Cop Control - Prueba de Software/src/avion/Avion.j...	esPosicionContenida
fiuba.algo3.titirero.dibujables		1,053	0,223	2	/Cop Control - Prueba de Software/src/fiuba/algo3/ti...	limpiar
pista		1,025	0,156	2	/Cop Control - Prueba de Software/src/pista/Pista.java	Pista
vista		1,2	0,4	2	/Cop Control - Prueba de Software/src/vista/VistaAvi...	getColor
test		1	0	1	/Cop Control - Prueba de Software/test/Ruina/TestU...	setUp
Number of Attributes (avg/max per type)	91	2,167	2,645	10	/Cop Control - Prueba de Software/test/Ruina/TestU...	
Number of Static Attributes (avg/max per type)	4	0,095	0,294	1	/Cop Control - Prueba de Software/src/avion/Proxim...	



$$V(G) = 12 - 10 + 2 = 4$$

En la Clase **Administrador de Vistas** el método **private ObjetoDibujable nuevaVistaDeAvion(Avion avion)** .

Java - Cop Control - Prueba de Software - Observador/AdministradorDeVistas.java - Eclipse

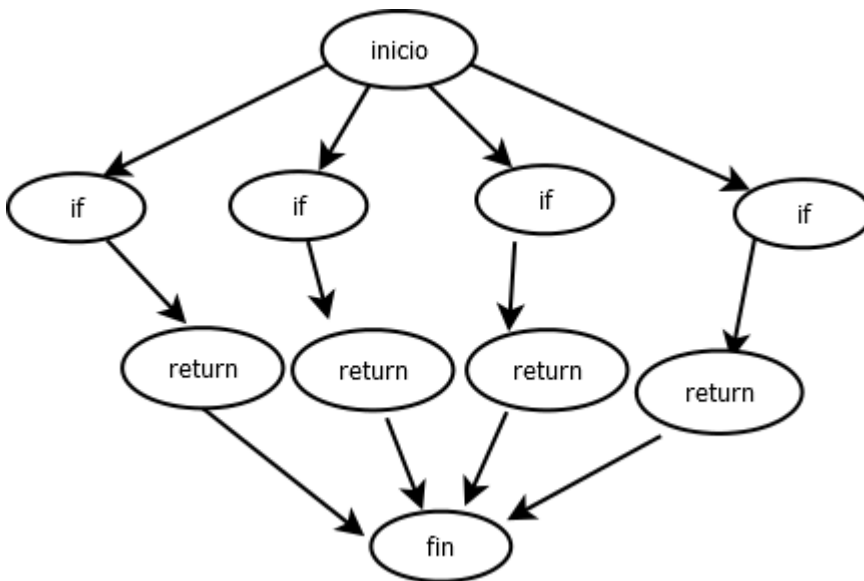
File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Java EE Java

Problems Javadoc Declaration Console Coverage Metrics - Cop Control - Prueba de Software - McCabe Cyclomatic Complexity (avg/max per method)

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
src		1,315	0,938	8	/Cop Control - Prueba de Software/src/copControl/J...	moverHacia
copControl		1,646	1,387	8	/Cop Control - Prueba de Software/src/copControl/J...	moverHacia
Jugador.java		2,75	3,031	8	/Cop Control - Prueba de Software/src/copControl/J...	moverHacia
Trayectoria.java		2,1	1,64	6	/Cop Control - Prueba de Software/src/copControl/T...	generarPosicionExtremo...
Mapa.java		1,462	1,151	5	/Cop Control - Prueba de Software/src/copControl/I...	generarPosicionExtremo...
Juego.java		1,789	1,321	5	/Cop Control - Prueba de Software/src/copControl/I...	crearAvionAlAzar
Nivel.java		1,6	1,114	5	/Cop Control - Prueba de Software/src/copControl/NL...	chocoConOtroAvion
Dificultad.java		1	0	1	/Cop Control - Prueba de Software/src/copControl/D...	Dificultad
Posicion.java		1	0	1	/Cop Control - Prueba de Software/src/copControl/P...	Posicion
Observador		1,9	1,578	5	/Cop Control - Prueba de Software/src/Observador/A...	nuevaVistaDePista
AdministradorDeVistas.java		2,5	1,803	5	/Cop Control - Prueba de Software/src/Observador/A...	nuevaVistaDePista
AdministradorDeVistas		2,5	1,803	5	/Cop Control - Prueba de Software/src/Observador/A...	nuevaVistaDePista
nuevaVistaDePista	5					
nuevaVistaDeAvion	5					
actualizar	2					
AdministradorDeVistas	1					
actualizarVistasPistas	1					
actualizarVistasAviones	1					
Observable.java	1	0	0	1	/Cop Control - Prueba de Software/src/Observador/O...	Observable
Observador.java	0	0	0			
fiuba.algo3.titiritero.modelo	1,133	0,499	3	/Cop Control - Prueba de Software/src/fiuba/algo3/ti...	run	
Inicializadores	1,273	0,445	2	/Cop Control - Prueba de Software/src/Inicializadores...	pistaLargaIniciada	
avion	1,056	0,229	2	/Cop Control - Prueba de Software/src/avion/Avion.j...	esPosicionContenida	
fiuba.algo3.titiritero.dibujables	1,053	0,223	2	/Cop Control - Prueba de Software/src/fiuba/algo3/ti...	limpiar	
pista	1,025	0,156	2	/Cop Control - Prueba de Software/src/pista/Pista.java	Pista	
vista	1,2	0,4	2	/Cop Control - Prueba de Software/src/vista/VistaAvi...	getColor	
test	1	0	1	/Cop Control - Prueba de Software/test/Ruina/TestU...	setUp	
Number of Attributes (avg/max per type)	91	2,167	2,645	10	/Cop Control - Prueba de Software/test/Ruina/TestU...	
Number of Static Attributes (avg/max per type)	4	0,095	0,294	1	/Cop Control - Prueba de Software/src/avion/Proxim...	



$$V(G) = 12 - 10 + 2 = 4$$