

Diseño de pruebas unitarias.

Objetivo: La razón para realizar esta prueba, es verificar que el método “mover” de la clase Ball, permita el recorrido del balón dentro de la cancha sin cruzar los parámetros de la ventana o bordes, etc.

Nombre	Clase	Escenario
Setup1	Ball	Un objeto de la clase Ball con vecX = 800 y vecY = 400
Setup2	Ball	Un objeto de la case Ball con vecX = -200 y vecY = -80

Clase	Método	Escenario	Valores de entrada	Resultado
Ball	Mover	Setup1	X = 800 y Y= 400	X = -800 y Y = -400
Ball	Mover	Setup2	X = -200 y Y = -80	X = 200 y Y = 80

Objetivo: La razón para realizar esta prueba, es verificar que el método “time” de la case Clock, esté haciendo el incremento en segundo de uno en uno. Ejemplo: 1seg, 2seg, 3seg, etc.

Nombre	Clase	Escenario
Setup1	ClockTest	Un objeto de la clase Clock con seconds = 0
Setup2	ClockTest	Un objeto de la clase Clock con seconds = 5
Setup3	ClockTest	Un objeto de la clase Clock con seconds = -8

Clase	Método	Escenario	Valores de entrada	Resultado
Clock	time	Setup1	0	1
Clock	time	Setup2	5	6
Clock	time	Setup3	-8	-7

Objetivo: La razón de esta prueba, es verificar que el método “searchTeam” de la clase Team, se encuentre realizando de manera efectiva la búsqueda del equipo deseado que ingresan los usuarios al jugar.

Nombre	Clase	Escenario
Setup1	TeamTest	Un objeto de la clase Team con name = Brasil
Setup2	TeamTest	Un objeto de la clase Team con name = Peru

Clase	Método	Escenario	Valores de entrada	Resultado
Team	searchTeam	Setup1	El Objeto de tipo team. (El array esta vacio).	“El equipo buscado no existe”
Team	searchTeam	Setup2	El objeto de tipo team. (El array tiene objetos del mismo tipo y el parámetro buscado se encuentra ahí)	Se retorna el objeto buscado.

Objetivo: La razón de esta prueba, es verificar que el método “addUniform” de la clase Team, se encuentre agregando exitosamente los uniformes de los equipos al arrayList de Uniformes.

Nombre	Clase	Escenario
Setup1	TeamTest	firstUniform = null
Setup2	TeamTest	firstUniform ≠ null

Clase	Método	Escenario	Valores de entrada	Resultado
-------	--------	-----------	--------------------	-----------

Team	addUniform	Setup1	colombiaUniform.png	El uniforme se agrega correctamente
Team	addUniform	Setup2	chileUniform.png	El uniforme se agrega correctamente

Objetivo: La razón de esta prueba, es verificar que el método “searchUniform” de la clase Team, funcione correctamente a la hora de realizar la selección de si se desea jugar con uniforme local o visitante, se este seleccionando el uniforme correcto.

Nombre	Clase	Escenario
Setup1	TeamTest	Un objeto de la clase Team con img = caliUniform.png
Setup2	TeamTest	Un objeto de la clase Team con img = colombiaUniform.png

Clase	Método	Escenario	Valores de entrada	Resultado
Team	searchUniform	Setup1	“caliUniform.png”	Este uniforme no existe en el sistema
Team	searchUniform	Setup2	“mexicoUniform.png”	Este uniforme fue encontrado

Objetivo: La razón de esta prueba, es verificar que el método “resultadoPartidos” de la clase Position, realice exitosamente la asignación del equipo que debe ganar el partido entre cada partido.

Nombre	Clase	Escenario
Setup1	PositionTest	Un objeto de tipo Team donde el ganador del partido es team1
Setup2	PositionTest	Un objetito de tipo Team donde el ganador del partido es team2

Clase	Método	Escenario	Valores de entrada	Resultado
Position	resultadoPartidos	Setup1	numGanador = 1	El ganador es team1
Position	resultadoPartidos	Setup2	numGanador ≠ 1	El ganador es team2

Objetivo: La razón de esta prueba, es verificar que el método “addPosition” de la clase Position, se encuentre asignando el lugar adecuado a los equipos dentro del árbol binario.

Nombre	Clase	Escenario
Setup1	PositionTest	Las posiciones del árbol están vacías
Setup2	PositionTest	Por lo menos la primera posición ya está definida

Clase	Método	Escenario	Valores de entrada	Resultado
Position	addPosition	Setup1	left = null y right = null	Se añade correctamente team1 y team2
Position	addPosition	Setup2	Left ≠ null y right ≠ null	Añade dependiendo del lado left o right

Objetivo: La razón de esta prueba, es verificar que el método “ganadorPartido” de la clase Match, realice efectivamente la asignación de goles que marca cada equipo durante el partido que se está jugando .

Nombre	Clase	Escenario
Setup1	MatchTest	golesTeam1 > golesTeam2
Setup2	MatchTest	golesTeam2 < golesTeam1

Clase	Método	Escenario	Valores de entrada	Resultado
Match	ganadorPartido	Setup1	3 > 1	El ganador del partido es el team1
Match	ganadorPartido	Setup2	1 < 2	El ganador del partido es el team2

Objetivo: La razón de esta prueba, es verificar que el método “moverBall” de la clase Match, tenga claras las áreas donde el programa reconoce que ha anotado un gol dentro del partido.

Nombre	Clase	Escenario
Setup1	MatchTest	Un objeto de la clase Match donde la posición en x de ball es = 530
Setup2	MatchTest	Un objeto de la clase Match donde la posición en x de ball es = -80

Clase	Método	Escenario	Valores de entrada	Resultado
Match	moverBall	Setup1	X=530	El equipo 1 ha anotado un gol

Match	moverBall	Setup2	X=-80	El equipo 2 ha anotado un gol
-------	-----------	--------	-------	-------------------------------

Objetivo: La razón de esta prueba, es verificar que el método “organizarEquipos” de la clase Game, ordene de manera idónea los equipos que entran por un criterio de orden alfabético.

Nombre	Clase	Escenario
Setup1	GameTest	Equipo1 = “Colombia”, Equipo2 = “Peru”, Equipo3 = “Argentina”
Setup2	GameTest	Equipo1 = “Usa”, Equipo2 = “Mexico”, Equipo3 = “España”

Clase	Método	Escenario	Valores de entrada	Resultado
Game	organizarEquipos	Setup1	Colombia, Peru, Argentina	1-Argentina, 2-Colombia, 3-Peru.
Game	organizarEquipos	Setup2	Usa, Mexico, España	1-España, 2-Mexico, 3-Usa

Objetivo: La razón de esta prueba, es verificar que el método “loadTeam” de la clase Game, se encuentre cargando de manera efectiva los archivos que me permiten conocer los equipo que están disponibles para elegir antes de jugar.

Nombre	Clase	Escenario
Setup1	GameTest	El archivo de texto no existe
Setup2	GameTest	El archivo de texto no tiene el mismo nombre suministrado
Setup3	GameTest	El archivo de texto existe y concuerda su nombre

Clase	Método	Escenario	Valores de entrada	Resultado
Game	loadTeam	Setup1	null	No se encuentra el archivo a cargar
Game	loadTeam	Setup2	“teams#%.txt”	No se reconoce el archivo que debe ser leído
Game	loadTeam	Setup3	“Teams.txt”	Los datos se han cargado exitosamente

Objetivo: La razón de esta prueba, es verificar que el método “serializableScore” de la clase Game, se encuentre guardando de manera efectiva los datos de los usuarios al terminar cada copa jugada. (nombre y puntos obtenidos durante el campeonato)

Nombre	Clase	Escenario
Setup1	GameTest	La dirección suministrada se creará en los archivos de la aplicación
Setup2	GameTest	La dirección suministrada no existe

Clase	Método	Escenario	Valores de entrada	Resultado
Game	serializableScore	Setup1	“.\data\Scores.txt”	El archivo se creó correctamente
Game	serializableScore	Setup2	“.\address\Scores.txt”	El folder suministrado en la dirección, no existe.