
CASO DE PRUEBAS 01-02: CONSTRUCTORES Y ATRIBUTOS

- PRUEBA 01 - ATRIBUTOS PÚBLICOS

-> MODELO_DEFECTO: Dispositivo Generico
-> TIPO_MOTOR_DEFECTO: Electrico
-> ALTITUD_MAXIMA: 50.0
-> RANGO_OPERACION: 100

- PRUEBA 02 - CONSTRUCTORES Y FÁBRICA (con datos correctos)

Registrando un DispositivoVolador para el dispositivo [D-01,Combustion]

-> Objeto creado con éxito

Registrando un DispositivoVolador para un dispositivo con los parámetros por defecto

-> Objeto creado con éxito

Intentando crear array de 1 objetos.

-> Array de objetos creado con éxito.

Intentando crear array de 20 objetos.

-> Array de objetos creado con éxito.

- PRUEBA 03 - CONSTRUCTORES Y FÁBRICA (con excepciones)

Registrando un DispositivoVolador para el dispositivo [null,Combustion]

-> Se ha producido un error: El modelo del dispositivo no puede ser nulo.

Registrando un DispositivoVolador para el dispositivo [H-01,]

-> Se ha producido un error: El tipo de motor no puede estar vacío.

Intentando crear array de -1 objetos.

-> Se ha producido un error: Número de dispositivos incorrecto: -1. Debe ser mayor o igual que 1 y menor o igual que 20

Intentando crear array de 50 objetos.

-> Se ha producido un error: Número de dispositivos incorrecto: 50. Debe ser mayor o igual que 1 y menor o igual que 20

CASO DE PRUEBAS 03: GETTERS Y MÉTODOS ESTÁTICOS

- PRUEBA 01 - ATRIBUTOS PÚBLICOS (antes de crear objetos)

-> numDispositivos: 0

- PRUEBA 02 - Creación de objetos y uso de getters()

Registrando un DispositivoVolador para el dispositivo [D-01,Electrico]

-> Objeto creado con éxito

Registrando un DispositivoVolador para el dispositivo [H-01,Combustion]

-> Objeto creado con éxito

Registrando un DispositivoVolador para el dispositivo [D-02,Hibrido]

-> Objeto creado con éxito

Array de dispositivos creado con éxito

Leyendo los datos almacenados en el DispositivoVolador

-> getModelo(): D-01

-> getTipoMotor(): Electrico

-> getAltitudActual(): 0.0

-> getPosicionX(): 0

-> getPosicionY(): 0

Leyendo los datos almacenados en el DispositivoVolador

-> getModelo(): H-01

-> getTipoMotor(): Combustion

-> getAltitudActual(): 0.0

-> getPosicionX(): 0

-> getPosicionY(): 0

Leyendo los datos almacenados en el DispositivoVolador

-> getModelo(): D-02

-> getTipoMotor(): Hibrido

-> getAltitudActual(): 0.0

-> getPosicionX(): 0

-> getPosicionY(): 0

- PRUEBA 03 - ATRIBUTOS PÚBLICOS (después de crear objetos)

-> numDispositivos: 3

CASO DE PRUEBAS 04: MÉTODOS DE ACCIÓN

Registrando un DispositivoVolador para el dispositivo [D-01,Electrico]

-> Objeto creado con éxito

Registrando un DispositivoVolador para el dispositivo [D-02,Combustion]

-> Objeto creado con éxito

Registrando un DispositivoVolador para el dispositivo [H-01,Hibrido]

-> Objeto creado con éxito

- PRUEBA 01 - Uso correcto del método de acción desplazar()

Leyendo los datos almacenados en el DispositivoVolador

-> getModelo(): D-01

-> getTipoMotor(): Electrico

-> getAltitudActual(): 0.0

-> getPosicionX(): 0

-> getPosicionY(): 0

-> El dispositivo D-01 se ha desplazado hacia adelante 10 posiciones en el eje X y 20 en el eje Y

Leyendo los datos almacenados en el DispositivoVolador

-> getModelo(): D-01

-> getTipoMotor(): Electrico

-> getAltitudActual(): 0.0

-> getPosicionX(): 10

-> getPosicionY(): 20

-> El dispositivo D-01 se ha desplazado hacia detrás 5 posiciones en el eje X y 5 en el eje Y

Leyendo los datos almacenados en el DispositivoVolador

-> getModelo(): D-01

-> getTipoMotor(): Electrico

-> getAltitudActual(): 0.0

-> getPosicionX(): 5

-> getPosicionY(): 15

- PRUEBA 02 - Uso incorrecto del método de acción desplazar()

El dispositivo D-01 se va a intentar desplazar hacia delante 500 posiciones en el eje X y 20 en el eje Y

-> Se ha producido un error: El dispositivo se ha intentado desplazar fuera de su rango de operación en el eje X.

- PRUEBA 03 - Uso correcto del método de acción calcularConsumo()

Calculando consumos para una distancia recorrida de 100 posiciones

-> El consumo del dispositivo D-01 es 120,00

-> El consumo del dispositivo D-02 es 200,00

-> El consumo del dispositivo H-01 es 150,00

- PRUEBA 04 - Uso incorrecto del método de acción calcularConsumo()

Intentando calcular el consumo del dispositivo D-01 para una distancia recorrida negativa...

-> Se ha producido un error: La distancia recorrida no puede ser negativa.

CASO DE PRUEBAS 05: método toString()

Registrando un DispositivoVolador para el dispositivo [D-01,Electrico]

-> Objeto creado con éxito

- PRUEBA 01 - Visualización del estado del objeto

-> El dispositivo D-01 tiene los siguientes datos iniciales

Datos del dispositivo D-01: [Modelo=D-01, Tipo de motor=Electrico, Altitud=0,00 metros, Posicion en X=0, Posicion en Y=0]

-> El dispositivo D-01 se ha desplazado hacia adelante 50 posiciones en el eje X y 50 en el eje Y

Datos del dispositivo D-01: [Modelo=D-01, Tipo de motor=Electrico, Altitud=0,00 metros, Posicion en X=50, Posicion en Y=50]

-> El dispositivo D-01 modifica su altitud actual a 25 metros y su posicion en el eje X a 15 y en el eje Y a 35

Datos del dispositivo D-01: [Modelo=D-01, Tipo de motor=Electrico, Altitud=25,00 metros, Posicion en X=15, Posicion en Y=35]

-> El dispositivo D-01 intenta establecer su altitud actual a 250 metros y su posicion en el eje X a 300 y en el eje Y a -1000

Datos del dispositivo D-01: [Modelo=D-01, Tipo de motor=Electrico, Altitud=0,00 metros, Posicion en X=0, Posicion en Y=0]