

Explicación del fallo en el test: **public void** testEsPosicionExtremo() {

Este el método ubicado en la clase mapa:

```
public boolean esPosicionExtremo(Posicion unaPosicion){  
  
    double x1 = unaPosicion.getCoordenadaX();  
    double y1 = unaPosicion.getCoordenadaY();  
  
    boolean bordelzq= ((x1==0)||((y1>=0)&&(y1<=this.dimension)));  
    boolean bordeSup=((y1==0)||((x1>=0)&&(x1<=this.dimension)));  
    boolean bordeInf=((y1==this.dimension)&&((x1>=0)&&(x1<=this.dimension)));  
    boolean bordeDer=((x1==this.dimension)&&((y1>=0)&&(y1<=this.dimension)));  
  
    return (bordelzq || bordeDer || bordeSup || bordeInf);  
}
```

Este el método de prueba:

```
@Test  
public void testEsPosicionExtremo() {  
  
    // Este sería el centro del mapa. (Dimension=500)  
    Posicion posicionBordelzq = new Posicion(250, 250);  
  
    //Deberia pasar el test, sin embargo, el metodo del modelo parece estar mal planteado,  
    //debido a que considera como un extremo la posicion (250,250).  
    assertFalse(mapa.esPosicionExtremo(posicionBordelzq));  
}
```

Dado que la posición (250, 250) está dentro del rango de una dimensión de 500, pero no está en los bordes (izquierdo, derecho, superior o inferior), el resultado esperado es false. La posición no se encuentra en ninguna de las líneas que forman los límites (bordes) de la dimensión.

Sin embargo, el método de prueba falla.

Esto se debe a que en esta línea se tiene:

```
boolean bordelzq= ((x1==0)||((y1>=0)&&(y1<=this.dimension)));
```

- x1 es 250, no es igual a 0, por lo que la primera parte `x1 == 0` es falsa.
- La segunda parte `(y1 >= 0) && (y1 <= this.dimension)` es verdadera porque `y1 = 250` está en el rango `[0, 500]`.

El resultado es que se toma a `bordelzq` como verdadero, cuando en realidad eso es incorrecto, ya que la posición no está en el borde izquierdo.

Algo similar ocurre con la línea de código:

```
boolean bordeSup=((y1==0)||((x1>=0)&&(x1<=this.dimension)));
```

Resultado final del código:

El código de la clase Mapa evaluará (bordeIzq || bordeSup || bordeInf || bordeDer).

Debido a que las variables bordeIzq y bordeSup son verdaderos, el resultado final será verdadero, lo cual es incorrecto. La posición (250, 250) no debería considerarse como un borde del mapa.

El código original dará un resultado incorrecto (ya que devolverá true) cuando se pase la posición (250, 250) con un mapa de dimensión 500.