

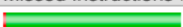
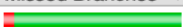
# Ejercicio 8

Realizado por: Álvaro Tapia Muñoz y Christian Berdejo Sánchez

## Jacoco report

avlTree

### avlTree

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
avl		97 %		93 %	8	104	5	237	1	48	0	2
Total	16 of 783	97 %	7 of 111	93 %	8	104	5	237	1	48	0	2





Al inicio, como podemos observar en la imagen, el porcentaje de cobertura de línea es del **97%** y el de cobertura de branch es del **93%**.

Esto se debe a que dentro de la clase `AvlTree` falta por cubrir el método `insert` entero y varias instrucciones de los métodos `deleteNode` y `searchNode`.

Por ejemplo, en el método `searchNode()` hay un error y es que realmente no busca ningún nodo en el árbol y simplemente devuelve el mismo nodo que se le pasa como argumento.

avlTree > avl

### avl

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
AvlTree		97 %		92 %	7	71	5	197	1	28	0	1
AvlNode		100 %		96 %	1	33	0	40	0	20	0	1
Total	16 of 783	97 %	7 of 111	93 %	8	104	5	237	1	48	0	2

av1Tree > avl > Av1Tree

## Av1Tree

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
<a href="#">insert(Object)</a>		0 %		n/a	1	1	3	3	1	1
<a href="#">deleteNode(Av1Node)</a>		93 %		78 %	3	8	1	19	0	1
<a href="#">searchNode(Av1Node)</a>		94 %		91 %	1	7	1	20	0	1
<a href="#">rebalance(Av1Node)</a>		100 %		100 %	0	7	0	17	0	1
<a href="#">searchClosestNode(Av1Node)</a>		100 %		100 %	0	7	0	23	0	1
<a href="#">leftRotation(Av1Node)</a>		100 %		100 %	0	3	0	13	0	1
<a href="#">rightRotation(Av1Node)</a>		100 %		100 %	0	3	0	13	0	1
<a href="#">findSuccessor(Av1Node)</a>		100 %		87 %	1	5	0	10	0	1
<a href="#">deleteLeafNode(Av1Node)</a>		100 %		100 %	0	3	0	8	0	1
<a href="#">getBalance(Av1Node)</a>		100 %		100 %	0	3	0	7	0	1
<a href="#">inOrder(Av1Node)</a>		100 %		100 %	0	2	0	6	0	1
<a href="#">insertAv1Node(Av1Node)</a>		100 %		80 %	1	4	0	9	0	1
<a href="#">deleteNodeWithALeftChild(Av1Node)</a>		100 %		n/a	0	1	0	5	0	1
<a href="#">deleteNodeWithARightChild(Av1Node)</a>		100 %		n/a	0	1	0	5	0	1
<a href="#">insertNodeLeft(Av1Node)</a>		100 %		n/a	0	1	0	4	0	1
<a href="#">insertNodeRight(Av1Node)</a>		100 %		n/a	0	1	0	4	0	1
<a href="#">height(Av1Node)</a>		100 %		100 %	0	2	0	5	0	1
<a href="#">doubleLeftRotation(Av1Node)</a>		100 %		n/a	0	1	0	4	0	1
<a href="#">doubleRightRotation(Av1Node)</a>		100 %		n/a	0	1	0	4	0	1
<a href="#">Av1Tree(Comparator)</a>		100 %		n/a	0	1	0	4	0	1
<a href="#">search(Object)</a>		100 %		n/a	0	1	0	2	0	1
<a href="#">compareNodes(Av1Node, Av1Node)</a>		100 %		n/a	0	1	0	1	0	1
<a href="#">setTop(Av1Node)</a>		100 %		n/a	0	1	0	3	0	1
<a href="#">delete(Object)</a>		100 %		n/a	0	1	0	2	0	1
<a href="#">av1IsEmpty()</a>		100 %		100 %	0	2	0	1	0	1
<a href="#">toString()</a>		100 %		n/a	0	1	0	2	0	1
<a href="#">insertTop(Av1Node)</a>		100 %		n/a	0	1	0	2	0	1
<a href="#">getTop()</a>		100 %		n/a	0	1	0	1	0	1
Total	16 of 625	97 %	6 of 85	92 %	7	71	5	197	1	28

En cambio, como podemos ver en la siguiente imagen, en la clase [Av1Node](#) el porcentaje de cobertura de línea es del **100%** y el de cobertura de branch es del **96%** debido a que falta por cubrir una condición del método [hasOnlyARightChild](#)

av1Tree > avl > Av1Node

## Av1Node

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
<a href="#">updateHeight()</a>		100 %		100 %	0	5	0	8	0	1
<a href="#">Av1Node(Object)</a>		100 %		n/a	0	1	0	8	0	1
<a href="#">isLeaf()</a>		100 %		100 %	0	3	0	1	0	1
<a href="#">hasOnlyALeftChild()</a>		100 %		100 %	0	3	0	1	0	1
<a href="#">hasOnlyARightChild()</a>		100 %		75 %	1	3	0	1	0	1
<a href="#">hasParent()</a>		100 %		100 %	0	2	0	1	0	1
<a href="#">hasLeft()</a>		100 %		100 %	0	2	0	1	0	1
<a href="#">hasRight()</a>		100 %		100 %	0	2	0	1	0	1
<a href="#">setLeft(Av1Node)</a>		100 %		n/a	0	1	0	2	0	1
<a href="#">setParent(Av1Node)</a>		100 %		n/a	0	1	0	2	0	1
<a href="#">setRight(Av1Node)</a>		100 %		n/a	0	1	0	2	0	1
<a href="#">setItem(Object)</a>		100 %		n/a	0	1	0	2	0	1
<a href="#">setHeight(int)</a>		100 %		n/a	0	1	0	2	0	1
<a href="#">setClosestNode(Av1Node)</a>		100 %		n/a	0	1	0	2	0	1
<a href="#">getLeft()</a>		100 %		n/a	0	1	0	1	0	1
<a href="#">getParent()</a>		100 %		n/a	0	1	0	1	0	1
<a href="#">getRight()</a>		100 %		n/a	0	1	0	1	0	1
<a href="#">getItem()</a>		100 %		n/a	0	1	0	1	0	1
<a href="#">getHeight()</a>		100 %		n/a	0	1	0	1	0	1
<a href="#">getClosestNode()</a>		100 %		n/a	0	1	0	1	0	1
Total	0 of 158	100 %	1 of 26	96 %	1	33	0	40	0	20

Resultados en la carpeta **CalidadCodigo**

En el html proporcionado podemos ver que el proyecto cuenta con bastantes warnings, algunos son por uso de frameworks que estan deprecated (en este caso Assert de junit), tambien podemos ver que muchos de ellos se deben a que se han tipado los argumentos explicitamente, lo cual puede ser reemplazado por `<>`. Para finalizar, IntelliJ nos indica muchos warnings por typos, los cuales no son errores de compilación pero si de escritura y por ultimo nos muestra propiedades sin usar en el sonar-project.properties.