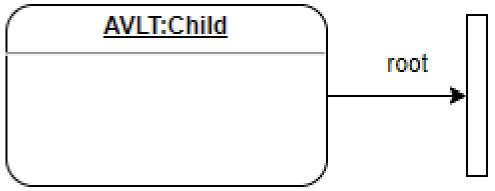
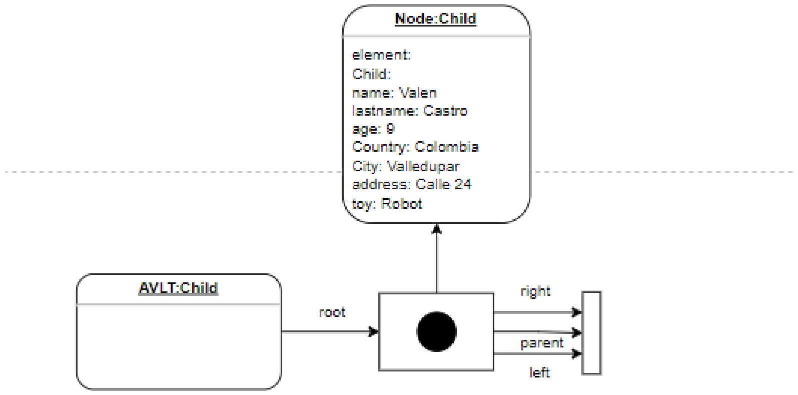
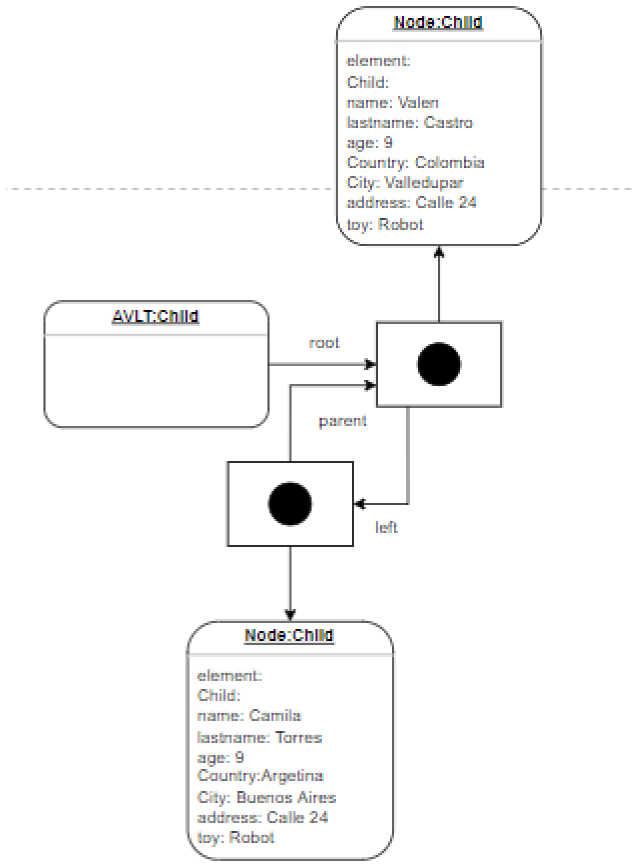
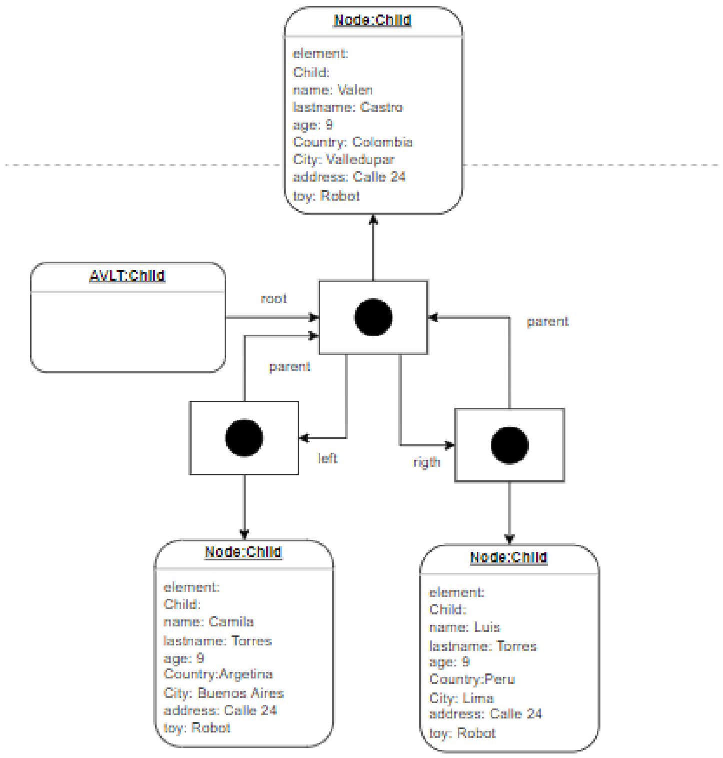
TEST CASES
AVL

Scenario configuration

Name	Class	Scenario
setUpScenaryI ()	AVLTest	
setUpScenary2()	AVLTest	 Node:Child element: Child: name: Valen lastname: Castro age: 9 Country: Colombia City: Valledupar address: Calle 24 toy: Robot AVLT:Child root right parent left

TEST CASES
AVL

setUpScenary3()	AVLTest	 Node:Child element: Child: name: Valen lastname: Castro age: 9 Country: Colombia City: Valledupar address: Calle 24 toy: Robot AVLT:Child root parent left Node:Child element: Child: name: Camila lastname: Torres age: 9 Country:Argetina City: Buenos Aires address: Calle 24 toy: Robot
setUpScenary4()	AVLTest	 Node:Child element: Child: name: Valen lastname: Castro age: 9 Country: Colombia City: Valledupar address: Calle 24 toy: Robot AVLT:Child root parent parent left rigth Node:Child element: Child: name: Camila lastname: Torres age: 9 Country:Argetina City: Buenos Aires address: Calle 24 toy: Robot Node:Child element: Child: name: Luis lastname: Torres age: 9 Country:Peru City: Lima address: Calle 24 toy: Robot

TEST CASES
AVL

Test Cases Design

Test Objective	Validate that the method add is correctly working			
Class	Method	Scenario	Input Values	Result
AVLTest	add	setUpScenary1 ()	name: “Valen” lastname: “Castro” age: 9 Country: “Colombia” City: “Valledupar” address: Calle 24 toy: “Robot”	The element of type child is added to the AVL, Now eoot is equal to this element.

Test Objective	Validate that the method add is correctly working			
Class	Method	Scenario	Input Values	Result
AVLTest	add	setUpScenary2()	name: “Camila” lastname: “Torres” age: 9 Country: “Argentina” City: “Buenos Aires” address: Calle 24 toy: “Robot”	The new element of type Child is added to the AVL, so now, left of the root is pointed to this new element .

Test Objective	Validate that the method dequeue is correctly working			
Class	Method	Scenario	Input Values	Result
QueueTest	dequeue	setUpScenary3()	name: “Luis” lastname: “Torres” age: 9 Country: “Perú” City: “Lima” address: Calle 24 toy: “Robot”	The element is added to the AVL, but, as is minor that the root this element will replace the root.

TEST CASES
AVL

Test Objective	Validate that the method delete is correctly working			
Class	Method	Scenario	Input Values	Result
AVLTest	delete	setUpScenary2()	name: “Valen” lastname: “Castro” age: 9 Country: “Colombia” City: “Valledupar” address: Calle 24 toy: “Robot”	When we ask for the root the element will return null.

Test Objective	Validate that the method delete is correctly working			
Class	Method	Scenario	Input Values	Result
AVLTest	delete	setUpScenary3()	name: “Camila” lastname: “Torres” age: 9 Country: “Argentina” City: “Buenos Aires” address: Calle 24 toy: “Robot”	When we ask for the left node of the root will return null

Test Objective	Validate that the method search is correctly working			
Class	Method	Scenario	Input Values	Result
AVLTest	search	setUpScenary2()	name: “Valen” lastname: “Castro” age: 9 Country: “Colombia” City: “Valledupar” address: Calle 24 toy: “Robot”	The method will return the element of the root.

TEST CASES
AVL

Test Objective	Validate that the method search is correctly working			
Class	Method	Scenario	Input Values	Result
AVLest	search	setUpScenary1()	name: “Valen” lastname: “Castro” age: 9 Country: “Colombia” City: “Valledupar” address: Calle 24 toy: “Robot”	The method will return null, the AVL isEmpty.

Test Objective	Validate that the method search is correctly working			
Class	Method	Scenario	Input Values	Result
AVLest	search	setUpScenary3()	name: “Camila” lastname: “Torres” age: 9 Country: “Argentina” City: “Buenos Aires” address: Calle 24 toy: “Robot”	The method will return the element that which is asked for.