

TAD Player		
sets={Player=<Player>}		
{inv: $ x,y  x \in \text{set} \wedge y \in \text{set}, x \neq y$ }		
primitive operations		
°CreateSet		--> Player
°AddElement	set x player	--> set
°cardinalidad	set	--> int
°DeleteElement	set	--> Player
°searchElement	set x Player	--> player

TAD Player		
sets={Player=<Player>}		
{inv:  x,y  x ∈ set ∧ y ∈ set, x ≠ y}		
primitive operations		
°CreateSet		--> Player
°AddElement	set x player	--> set
°cardinalidad	set	--> int
°DeleteElement	set	--> Player
°searchElement	set x Player	--> player

TAD Player		
sets={Player=<Player>}		
{inv:  x,y  x ∈ set ∧ y ∈ set, x ≠ y}		
primitive operations		
°CreateSet		--> Player
°AddElement	set x player	--> set
°cardinalidad	set	--> int
°DeleteElement	set	--> Player
°searchElement	set x Player	--> player

TAD Player		
sets={Player=<Player>}		
{inv:  x,y  x ∈ set ∧ y ∈ set, x ≠ y}		
primitive operations		
°CreateSet		--> Player
°AddElement	set x player	--> set
°cardinalidad	set	--> int
°DeleteElement	set	--> Player
°searchElement	set x Player	--> player

TAD Player		
sets={Player=<Player>}		
{inv:  x,y  x ∈ set ∧ y ∈ set, x ≠ y}		
primitive operations		
°CreateSet		--> Player
°AddElement	set x player	--> set
°cardinalidad	set	--> int
°DeleteElement	set	--> Player
°searchElement	set x Player	--> player

TAD Player		
sets={Player=<Player>}		
{inv:  x,y  x ∈ set ∧ y ∈ set, x ≠ y}		
primitive operations		
°CreateSet		--> Player
°AddElement	set x player	--> set
°cardinalidad	set	--> int
°DeleteElement	set	--> Player
°searchElement	set x Player	--> player

TAD Player		
sets={Player=<Player>}		
{inv:  x,y  x ∈ set ∧ y ∈ set, x ≠ y}		
primitive operations		
°CreateSet		--> Player
°AddElement	set x player	--> set
°cardinalidad	set	--> int
°DeleteElement	set	--> Player
°searchElement	set x Player	--> player

TAD Player		
sets={Player=<Player>}		
{inv:  x,y  x ∈ set ∧ y ∈ set, x ≠ y}		
primitive operations		
°CreateSet		--> Player
°AddElement	set x player	--> set
°cardinalidad	set	--> int
°DeleteElement	set	--> Player
°searchElement	set x Player	--> player

TAD Player		
sets={Player=<Player>}		
{inv:  x,y  x ∈ set ∧ y ∈ set, x ≠ y}		
primitive operations		
°CreateSet		--> Player
°AddElement	set x player	--> set
°cardinalidad	set	--> int
°DeleteElement	set	--> Player
°searchElement	set x Player	--> player

createSet()
Create a Player set
{pre: Element}
{post: Player}

createSet()
Create a Player set
{pre: Element}
{post: Player}

createSet()
Create a Player set
{pre: Element}
{post: Player}

createSet()
Create a Player set
{pre: Element}
{post: Player}

addElement(set,player)
add new Player in set
{pre: set = Elements $\neq$ Player} $\wedge$ element $\in$ Player
{post: set = element}

addElement(set,player)
add new Player in set
{pre: set = Elements $\neq$ Player} $\wedge$ element $\in$ Player
{post: set = element}

addElement(set,player)
add new Player in set
{pre: set = Elements $\neq$ Player} $\wedge$ element $\in$ Player
{post: set = element}

addElement(set,player)
add new Player in set
{pre: set = Elements $\neq$ Player} $\wedge$ element $\in$ Player
{post: set = element}

cardinalidad(set)

Count cardinalidad in class Player
{pre: set.size= Int}
{post: set.size}

deleteElement(set,player)
Delete element in Player
{pre: set=Elements } $\wedge$ elements $\in$ Player
{post: set=Elements} $\wedge$ elements $\notin$ Player

searchElement(set, player)
search element in Player
{pre: set= Elements} $\wedge$ element $\in$ Player $\vee$ element $\notin$ Player
{post: Null if player $\notin$ set Player de lo contrario}