**DISEÑO DE PRUEBAS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba: Verifica que el método de addJugador añade correctamente un Jugador | | | | |
| Clase | Método | Escenario | Valores de entrada | Resultado |
| Juego | addJugador(): void | Raizjugador = null  Jugador =  Nickname: “Joan”  Puntos: 100  Nivel: 1 | Ninguna | Raizjugador es igual el jugador |
| Juego | addJugador(): void | Raizjugador= Joan  Jugador =  Nickname: “Camila”  Puntos: 50  Nivel: 1 | Ninguna | Camila es el subárbol izquierdo de Joan |
| Juego | addJugador(): void | Raizjugador= Joan  Jugador =  Nickname: “Sariana”  Puntos: 500  Nivel: 4 | Ninguna | Sariana es el subárbol derecho de Joan |
| Juego | addJuagdor(): void | Raizjugador = Joan  Jugador =  Nickname: “Joan”  Puntos: 200  Nivel: 1 | Ninguna | Lanza JugadorRepetidoException |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba: Verifica que el método de mostrar la lista ordenada por puntaje de manera ascendente y descendente funciona bien. | | | | |
| Clase | Método | Escenario | Valores de entrada | Resultado |
| Juego | ordernarPuntajeAscendente(): ArrayList<Jugador> | Hay un árbol binario donde la raíz es  Nickname: “Julian”  Puntos: 100  Nivel: 3  Y en conjunto el árbol tiene los siguientes elementos  Nickname: “Javier”  Puntos: 200  Nivel: 4  Nickname: “Alejandro”  Puntos: 50  Nivel: 2  Nickname: “Manyolml”  Puntos: 500  Nivel: 7  Nickname: “JuanMa”  Puntos: 400  Nivel: 6 | Ninguna | Alejandro  Julian  Javier  JuanMa  Manyolml |
| Juego | ordernarPuntajeDescendente(): ArrayList<Jugador> | El mismo de arriba | Ninguna | Manyolml  JuanMa  Javier  Julian  Alejandro |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba: Verifica que el método de mostrar la lista ordenada por nombre de manera ascendente y descendente funciona bien. | | | | |
| Clase | Método | Escenario | Valores de entrada | Resultado |
| Juego | ordernarNombreAscencente(): ArrayList<Jugador> | El mismo de arriba | Ninguna | Manyolml  Julian  JuanMa  Javier  Alejandro |
| Juego | ordernarNombreDescencente(): ArrayList<Jugador> | El mismo de arriba | Ninguna | Alejandro  Javier  JuanMa  Julian  Manyolml |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba: Verifica que el método de mostrar la lista ordenada por nivel de manera ascendente y descendente funciona bien. | | | | |
| Clase | Método | Escenario | Valores de entrada | Resultado |
| Juego | ordernarNivelAscencente(): ArrayList<Jugador> | El mismo de arriba | Ninguna | Alejandro  Julian  Javier  JuanMa  Manyolml |
| Juego | ordernarNivelAscencente(): ArrayList<Jugador> | El mismo de arriba | Ninguna | Manyolml  JuanMa  Javier  Julian  Alejandro |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba: Verifica que el método de buscar un jugador por puntaje encuentra al jugador correspondiente | | | | |
| Clase | Método | Escenario | Valores de entrada | Resultado |
| Juego | buscarJugadorPuntos(int puntos): Jugador | El mismo de arriba | 500 | Retorna a Manyo |
| Juego | buscarJugadorPuntos(int puntos): Jugador | El mismo de arriba | 2 | Lanza PuntajeNoExisteException |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba: Verifica que el método de buscar un jugador por puntaje encuentra al jugador correspondiente | | | | |
| Clase | Método | Escenario | Valores de entrada | Resultado |
| Juego | BuscarJugadorNombre(String nombre): Jugador | El mismo de arriba | “Javier” | Retorna al jugador “Javier” |
| Juego | buscarJugadorNombre(String nombre): Jugador | El mismo de arriba | “Karol” | Lanza NombreNoExisteException |