

## DISEÑO DE LAS PRUEBAS AUTOMÁTICAS UNITARIAS DE LA CLASE GAME

Prueba N° 1	Objetivo: Probar que el método de ordenamiento por inserción funciona correctamente tomando como criterio el nombre.			
Clase	Método	Escenario	Entradas	Resultado
Game	+ sortPlayerNames(int): void	Ordenar un ArrayList con un solo jugador.	Modo de ordenamiento: <ul style="list-style-type: none"> <li>• 1 para ascendente.</li> <li>• -1 para descendente</li> </ul>	El ArrayList ha sido ordenado.
Game	+ sortPlayerNames(int): void	Ordenar un ArrayList con varios jugadores.	Modo de ordenamiento: <ul style="list-style-type: none"> <li>• 1 para ascendente.</li> <li>• -1 para descendente</li> </ul>	El ArrayList ha sido ordenado.

Prueba N° 2	Objetivo: Prueba que la búsqueda binaria para los nombres funciona.			
Clase	Método	Escenario	Entradas	Resultado
Game	+ searchPlayerName(String): Player	Buscar un jugador que está entre el primer y el último nodo.	Nombre del jugador a buscar.	Jugador encontrado. El jugador que se encontró tiene el mismo nombre que se le pasó al método por parámetro.
Game	+ searchPlayerName(String): Player	Buscar el primer jugador del ArrayList.	Nombre del jugador a buscar.	Jugador encontrado. El jugador que se encontró tiene el mismo nombre que se le pasó al método por parámetro.
Game	+ searchPlayerName(String): Player	Buscar el último jugador del ArrayList.	Nombre del jugador a buscar.	Jugador encontrado. El jugador que se encontró tiene el mismo nombre que se le pasó al método por parámetro.
Game	+ searchPlayerName(String): Player	Buscar un jugador que no esté en el ArrayList.	Nombre del jugador a buscar.	El método devuelve null, ya que el nombre que se pasó por parámetro no le pertenece a ningún jugador.

<b>Prueba N° 3</b>	<b>Objetivo:</b> Prueba la búsqueda binaria para el puntaje.			
<b>Clase</b>	<b>Método</b>	<b>Escenario</b>	<b>Entradas</b>	<b>Resultado</b>
Game	+ searchPlayerScore(int): ArrayList<Player>	Buscar una puntuación que solo la tenga un jugador entre el primer y último nodo.	Puntaje.	El ArrayList no está vacío, ya que se encontró un jugador con ese puntaje.
Game	+ searchPlayerScore(int): ArrayList<Player>	Buscar una puntuación que la tengan dos o tres jugadores entre el primer y último nodo.	Puntaje.	El ArrayList no está vacío, ya que se encontró al menos un jugador con ese puntaje.
Game	+ searchPlayerScore(int): ArrayList<Player>	Buscar la puntuación más baja.	Puntaje.	El ArrayList no está vacío, ya que se encontró al jugador con el menor puntaje.
Game	+ searchPlayerScore(int): ArrayList<Player>	Buscar la puntuación más alta.	Puntaje.	El ArrayList no está vacío, ya que se encontró al jugador con el mayor puntaje.
Game	+ searchPlayerScore(int): ArrayList<Player>	Buscar una puntuación que no la tenga ningún jugador.	Puntaje.	El ArrayList está vacío, ya que ningún jugador obtuvo ese puntaje.

<b>Prueba N° 4</b>	<b>Objetivo:</b> Probar que el método playerExists(String) indica correctamente si un jugador existe en el árbol binario.			
<b>Clase</b>	<b>Método</b>	<b>Escenario</b>	<b>Entradas</b>	<b>Resultado</b>
Game	+ playerExists(String): boolean	Buscar un jugador que si esté en el árbol binario.	Nombre del jugador.	Verdadero. Hay un jugador que sí posee el nombre pasado por parámetro.
Game	+ playerExists(String): boolean	Buscar un jugador que no esté en el árbol binario.	Nombre del jugador.	False. Ningún jugador posee ese nombre.

Prueba N° 5	Objetivo: Probar que se elimina correctamente un jugador del árbol binario de búsqueda.			
Clase	Método	Escenario	Entradas	Resultado
Game	+ deletePlayerFromTree(Player): void	Eliminar la raíz del árbol.	Referencia a un objeto Player el cual es la raíz del árbol.	Falso. El método playerExists(String) indica que ese jugador ya no está en el árbol.
Game	+ deletePlayerFromTree(Player): void	Eliminar una hoja del árbol.	Referencia a un objeto Player el cual es una hoja del árbol.	Falso. El método playerExists(String) indica que ese jugador ya no está en el árbol.
Game	+ deletePlayerFromTree(Player): void	Eliminar un jugador en medio del árbol.	Referencia a un objeto Player que no es la raíz del árbol ni tampoco una hoja.	Falso. El método playerExists(String) indica que ese jugador ya no está en el árbol.