

Nº	Modulo/Función	Objetivo	Precondición	Postcondición	Implementación
1	Player/player_create	Prueba si se crea correctamente un jugador	Se crea el jugador	Un puntero no nulo al jugador creado	test1_player_create();
2	Player/player_set_name	Prueba si se le asigna correctamente un nombre a un jugador	El nombre del jugador	La salida tiene que ser el puntero al jugador	test1_player_set_name();
3	Player/player_set_name	Prueba asignar un nombre a un jugador sin inicializar	El jugador al que establecer el nombre es un puntero a NULL	La salida debe ser NULL	test2_player_set_name();
4	Player/player_set_name	Prueba asignarle un nombre nulo a un jugador	La cadena que se quiere asignar es un puntero a NULL	La salida tiene que ser NULL	test3_player_set_name();
5	Player/player_set_id	Prueba asignar un id a un jugador en condiciones normales	Al jugador se le asigna un id cualquiera mayor que cero	La salida tiene que ser el puntero al jugador	test1_player_set_id();
6	Player/player_set_id	Prueba asignar un id a un jugador sin inicializar	El jugador al que establecer el nombre es un puntero a NULL	La salida esperada es NULL	test2_player_set_id();
7	Player/player_set_id	Prueba asignar un id a un jugador	El id que se quiere asignar es NO_ID	La salida que se espera es NULL	test3_player_set_id();
8	Player/player_set_location	Prueba asignarle una localización a un jugador en condiciones	El jugador ha sido previamente inicializado y la localización que se le asigna es valida	La salida es el puntero al jugador	test1_player_set_location();
9	Player/player_set_location	Prueba asignarle una localización a un jugador sin inicializar.	El jugador no ha sido inicializado previamente	La salida que se espera es NULL	test2_player_set_location();
10	Player/player_set_location	Prueba asignarle una localización invalida a un jugador.	La localización que se quiere asignar al jugador es es negativa	La salida esperada es NULL	test3_player_set_location();
11	Player/player_get_id	Prueba leer el id de un jugador	Al jugador se le ha asignado previamente el id 12	La salida esperada es el id asignado antes (12)	test1_player_get_id();
12	Player/player_get_id	Prueba leer el id de un jugador	El jugador no ha sido inicializado previamente	La salida esperada es NO_ID	test2_player_get_id();

13	Player/player_get_id	Intenta leer el id del jugador	Al jugador no se le ha asignado ninguna id todavia	La salida esperada es NO_ID	test3_player_get_id();
14	Player/player_get_name	Intenta leer el nombre de un jugador	Al jugador se le ha asignado previamente el nombre "Bob"	La salida esperada es el nombre asignado antes : "Bob"	test1_player_get_name();
15	Player/player_get_name	Prueba leer el nombre de un jugador	El jugador está sin inicializar	La salida debe ser NULL	test2_player_get_name();
16	Player/player_get_name	Prueba leer el nombre de un jugador	Al jugador no se le ha dado ningun nombre previamente	La salida esperada es una cadena vacía	test3_player_get_name();
17	Player/player_get_location	Leer la localización de un jugador	Al jugador se le ha asignado previamente la localización 12	La salida esperada es la localización previamente asignada (12)	test1_player_get_location();
18	Player/player_get_location	Prueba leer la localización del jugador	El jugador no ha sido inicializado	La salida esperada es NO_ID	test2_player_get_location();
19	Player/player_get_location	Prueba leer la localización del jugador	Al jugador no se le ha asignado ninguna localización todavía	La salida que se espera es NO_ID	test3_player_get_location();
20	Player/player_add_object	Prueba añadir un objeto a un jugador	El jugador está sin inicializar	La salida que se espera es ERROR	test1_player_add_object();
21	Player/player_add_object	Prueba añadir un objeto a un jugador en condiciones normales	El jugador ha sido correctamente inicializado y el id del objeto es valido	La salida debe ser OK	test2_player_add_object();
22	Player/player_add_object	Prueba añadir un objeto al inventario del jugador	El objeto tiene id -1	La salida que se espera es ERROR	test3_player_add_object();
23	Player/player_remove_object	Prueba quitar un objeto al jugador en condiciones normales	Al jugador se le añade un objeto	La salida que se espera es OK	test1_player_remove_object();
24	Player/player_remove_object	Prueba quitar un objeto a un jugador	Al jugador no se le ha añadido todavía ningun objeto	La salida que se espera es ERROR	test2_player_remove_object();
25	Player/player_remove_object	Prueba quitar un objeto a un jugador	El jugador está sin inicializar	La salida que se espera es ERROR	test3_player_remove_object();
26	Player/player_remove_object	Prueba quitar un objeto al jugador	El objeto ya ha sido borrado anteriormente	La salida que se espera es ERROR	test4_player_remove_object();

27	Player/player_set_max_objects	Prueba poner el número máximo de objetos de un jugador	Las condiciones son normales: jugador inicializado, número de objeto dentro de limites.	La salida debe ser el puntero al jugador	test1_player_set_max_objects();
28	Player/player_set_max_objects	Prueba poner el número máximo de objetos de un jugador	El número de objetos está por encima del limite permitido	La salida debe ser NULL	test2_player_set_max_objects();
29	Player/player_set_max_objects	Prueba poner el número máximo de objetos de un jugador	El jugador está sin inicializar	La salida debe ser NULL	test3_player_set_max_objects();
23	Player/player_set_max_objects	Prueba poner el numero máximo de objetos de un jugador	El número de objetos es un número negativo	La salida debe ser NULL	test4_player_set_max_objects();
31	Player/player_has_object	Prueba si el jugador tiene un objeto en condiciones normales	Se le añade un objeto al jugador	La salida debe ser TRUE	test1_player_has_object();
32	Player/player_has_object	Prueba si el jugador tiene un objeto	El jugador está sin inicializar	La salida debe ser FALSE	test2_player_has_object();
33	Player/player_has_object	Prueba si el jugador tiene un objeto	El jugador está inicializado pero no tiene ningun objeto	La salida debe ser FALSE	test3_player_has_object();