

# Examen: elección de objetivo mediante selectmenu

## Índice

1. Objetivo de la sesión.....	1
2. Pasos.....	1
1.- Añadimos selectmenu “Objetivo”.....	1
2.- Carga de valores en “Objetivo”.....	2
3.- Actualizar pantalla del sonar.....	3
Rúbrica.....	4

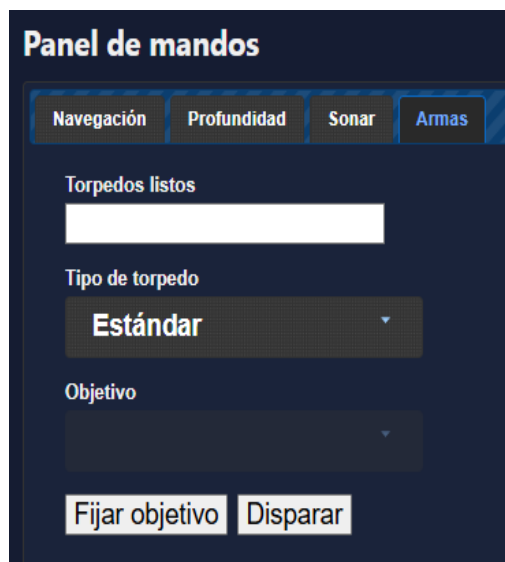
## 1. Objetivo de la sesión

En la pestaña “Armas” vamos a añadir la posibilidad de que el usuario seleccione el barco objetivo mediante un selectmenu (componente de JQuery UI)

## 2. Pasos

### 1.- Añadimos selectmenu “Objetivo”.

- En la pestaña “Armas”, debajo del selectmenu “Tipo torpedo”, añadamos otro selectmenu “Objetivo”. Inicialmente vacío. Y `.selectmenu("disable")`



The screenshot shows a dark-themed control panel titled "Panel de mandos". It has four tabs: "Navegación", "Profundidad", "Sonar", and "Armas". The "Armas" tab is active. Below the tabs, there is a "Torpedos listos" label and an empty text input field. Below that is a "Tipo de torpedo" label and a dropdown menu with "Estándar" selected. Below that is an "Objetivo" label and an empty dropdown menu. At the bottom are two buttons: "Fijar objetivo" and "Disparar".

EN el HTML, **dentro del <select> no poner <option>**, los añadiremos dinámicamente con jquery más tarde.

Algo así:

```
<select id="seleccionar-objetivo">
</select>
```

## 2.- Carga de valores en “Objetivo”

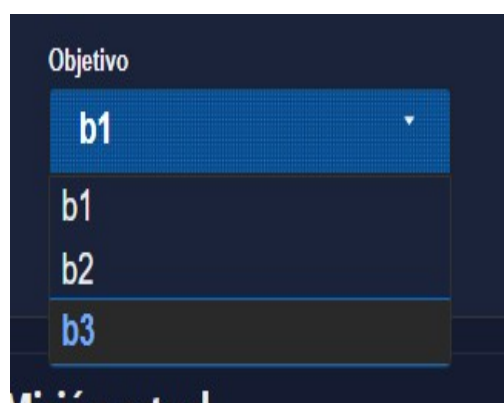
- Cuando el jugador pulse el botón “Escanear” (en la pestaña “Sonar”), ya implementado en prácticas, vamos a llamar la función: **mostrarBarcosEnArmasObjetivo()**:

**Debes tener algo así de la practica, añadiendo ya la función:**

```
$("#btn-escanear")
.button()
.on("click", function () {
    $(".barco").hide();
    $(".barco").fadeIn(4000);

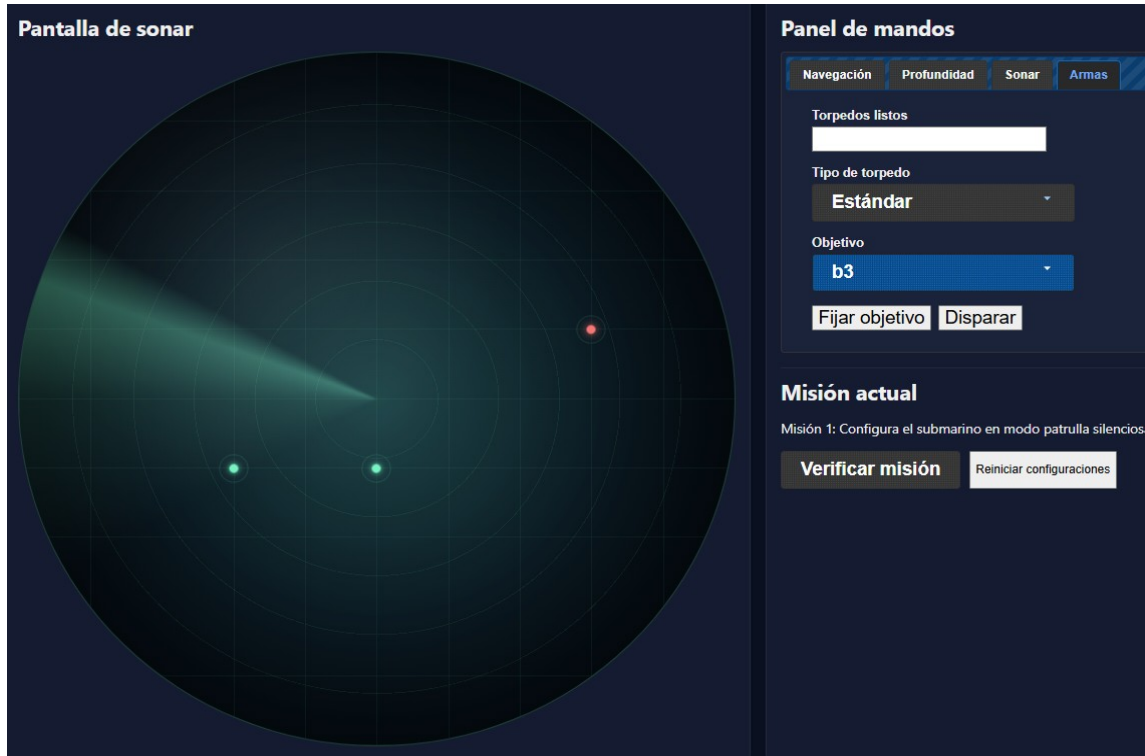
    mostrarBarcosEnArmasObjetivo();
});
```

- La función **mostrarBarcosEnArmasObjetivo()** mostrará los “data-id” de los <div> de los barcos en el nuevo selectmenu “Objetivo”. Para ello:
  - Recupera con jquery todos los barcos que hay dentro de <div id="sonar">.
  - Recorre el array de barcos para crear tantos <option value="bx">bx</option> como barcos alla. Usa “... **.find(“**.barco**”).each(...** “ para recorrer los barcos.
  - Finalmente añadelos a: <select id="seleccionar-objetivo">



### 3.- Actualizar pantalla del sonar

- Dentro de la función inicializarControlesArmas() debemos hacer:
  - Cuando seleccionamos un objetivo en el selectmenu (uso de `.on("selectmenuselect"...)` ) que acabamos de hacer debe ocurrir:
    - Añadir la clase CSS “.barco-seleccionado” al barco seleccionado para que quede marcado en rojo.



# Rúbrica

Criterio (RA)	Excelente (10)	Bien (7)	Suficiente (5)	Insuficiente (0)
<b>RA2-b/c/e/f. Sintaxis, variables, decisiones y bucles en la lógica de “Objetivo”</b> (Evidenciado en: Paso 2 – recorrer barcos con .each() y crear opciones; Paso 3 – seleccionar objetivo y aplicar clase)	Código de JS/jQuery correcto y limpio en todos los apartados del examen: declara adecuadamente variables/constant es, respeta ámbitos, usa decisiones y recorridos sin errores. (1,20 pt)	Pequeños fallos de sintaxis o estructura, pero funcionamiento correcto. (0,84 pt)	Código funcional pero con varios problemas de estilo, ámbitos o decisiones. (0,60 pt)	Errores graves que impiden que el examen funcione. (0 pt)
<b>RA2-g/h. Comentarios y uso de herramientas/entornos</b> (Evidenciado en: código añadido en mostrarBarcosEnArmasObjetivo() y en el manejador selectmenuselect)	Incluye comentarios claros y uso apropiado de herramientas del entorno (jQuery UI, consola). (0,60 pt)	Comentarios útiles pero no completos; uso razonable del entorno. (0,42 pt)	Comentarios escasos o genéricos; poco uso de herramientas de apoyo. (0,30 pt)	Sin comentarios y mal uso del entorno. (0 pt)
<b>RA3-a/b. Objetos predefinidos y DOM: ventanas, documento, #sonar, .barco, #seleccionar-objetivo</b> (Evidenciado en: Paso 2 – localizar barcos dentro de #sonar; Paso 3 – localizar y modificar el barco seleccionado)	Uso óptimo de objetos del DOM y selectores adecuados sin redundancias. (0,80 pt)	Uso correcto con algún selector mejorable. (0,56 pt)	Uso funcional pero confuso de selectores o ámbitos. (0,40 pt)	Selectores incorrectos que impiden el funcionamiento. (0 pt)
<b>RA3-c/d/e. Cambio de aspecto, generación de etiquetas y adaptación al usuario</b> (Evidenciado en: creación dinámica de <option>; habilitar/deshabilitar selectmenu; aplicar .barco-seleccionado)	Opciones generadas sin duplicados, buen control del estado del select y correcto marcado visual del barco. (1,20 pt)	Funciona correctamente con detalles mejorables. (0,84 pt)	Funciona con fallos: duplicados, resaltado defectuoso, etc. (0,60 pt)	No genera opciones o no marca correctamente el barco. (0 pt)
<b>RA3-h. Depuración y documentación del código</b> (Evidenciado en: uso de console.log, comprobaciones durante la carga de opciones, depuración del evento de selección)	Uso consciente de depuración y código final limpio. (0,40 pt)	Alguna depuración y limpieza razonable. (0,28 pt)	Rastro de depuración desordenado o insuficiente. (0,20 pt)	No hay depuración y el código falla sin diagnosticar. (0 pt)
<b>RA4-b/c/d/e. Funciones propias y manejo de colecciones de barcos</b> (Evidenciado en: implementación correcta de mostrarBarcosEnArmasObjetivo() y uso de .each() para recorrer barcos)	Funciones bien estructuradas, manejo correcto de colecciones sin duplicar código. (1,60 pt)	Generalmente correcto, con pequeñas mejoras posibles. (1,12 pt)	Funciona pero mezcla demasiada lógica o usa patrones poco claros. (0,80 pt)	No organiza funciones propias o maneja mal las colecciones. (0 pt)
<b>RA5-a/b/c/d. Manejo de eventos: botón “Escanear” y evento selectmenuselect</b> (Evidenciado en: Paso 1 – evento del botón “Escanear”; Paso 3 – evento selectmenu que marca el objetivo)	Eventos configurados correctamente, flujo robusto y parámetros event	Flujo correcto con algún detalle mejorable. (1,40 pt)	Eventos conectados de forma funcional pero confusa. (1,00 pt)	Eventos mal implementados o sin funcionamiento. (0 pt)

Criterio (RA)	Excelente (10)	Bien (7)	Suficiente (5)	Insuficiente (0)
	y ui bien usados. (2,00 pt)		pt)	
<b>RA6-a/b/c. Acceso a la estructura del documento mediante jQuery</b> (Evidenciado en: recuperar barcos desde #sonar, acceder a atributos data-id, actualizar #seleccionar-objetivo)	Selectores correctos, acceso eficiente al DOM. (1,20 pt)	Acceso correcto con pequeñas imprecisiones. (0,84 pt)	Acceso válido pero poco claro o repetitivo. (0,60 pt)	Acceso incorrecto a elementos del DOM. (0 pt)
<b>RA6-d/e. Creación/modificación de elementos y acciones sobre eventos del modelo</b> (Evidenciado en: añadir <option> al <select>, limpiar contenido previo, aplicar clase .barco-seleccionado al elegir objetivo)	Crea/modifica correctamente elementos y asocia acciones del DOM con precisión. (1,00 pt)	Funciona con alguna pequeña carencia en limpieza o consistencia. (0,70 pt)	Manipulación mínima pero funcional, con errores menores. (0,50 pt)	No crea/modifica elementos o las acciones no afectan al DOM. (0 pt)