




Nombre: \_\_\_\_\_ Clase \_\_\_\_\_ Equipo: \_\_\_\_\_

**Importante:** Compruebe las instrucciones que se dan para realizar la entrega, siga las mismas, márquelas cuando las haya comprobado y firme en la casilla correspondiente. Cualquier entrega que no cumpla las reglas de nombrado, el formato de compresión del archivo o el contenido de los archivos del mismo, será penalizada con 2 puntos sobre 10 por cada incumplimiento. Pasado el límite de entrega se aceptará el envío del trabajo, con una penalización de un 2 puntos sobre 10 de la calificación por cada minuto o fracción de retraso a partir del tercer minuto.

<ul style="list-style-type: none"><li><input type="checkbox"/> Restricciones: Las rutas de los ficheros empleados serán relativas, a fin de que su solución pueda ser examinada en cualquier equipo.</li><li><input type="checkbox"/> Material a entregar: Los archivos <code>funcionesRuletaSuerte.js</code> y <code>jQueryRuletaSuerte.js</code>.</li><li><input type="checkbox"/> Formato de entrega: Archivo comprimido ZIP.</li><li><input type="checkbox"/> Nombre del archivo de entrega: El nombre del fichero tendrá un formato específico dictado por el nombre de cada alumno. Por ejemplo, para un alumno llamado "José María Núñez Pérez" el fichero se nombrará como <code>NunyezPerezJM.zip</code>. Obsérvese que las tildes son ignoradas y las eñes sustituidas.</li><li><input type="checkbox"/> Vía entrega: La entrega del trabajo se hará a través de la tarea correspondiente en el Campus Virtual y se completará cuando se reciba por correo electrónico resguardo de la entrega. Conserve dicho correo como prueba del envío.</li><li><input type="checkbox"/> Comprobación de la entrega: Los alumnos deben comprobar que el archivo depositado en la plataforma es correcto descargándolo de la misma, descomprimiéndolo y comprobando su contenido.</li></ul>	He realizado todas las comprobaciones indicadas. Firma del alumno: _____
---	--

### Objetivos

- Demostrar los conocimientos adquiridos en el desarrollo de aplicaciones empleando JavaScript y jQuery.

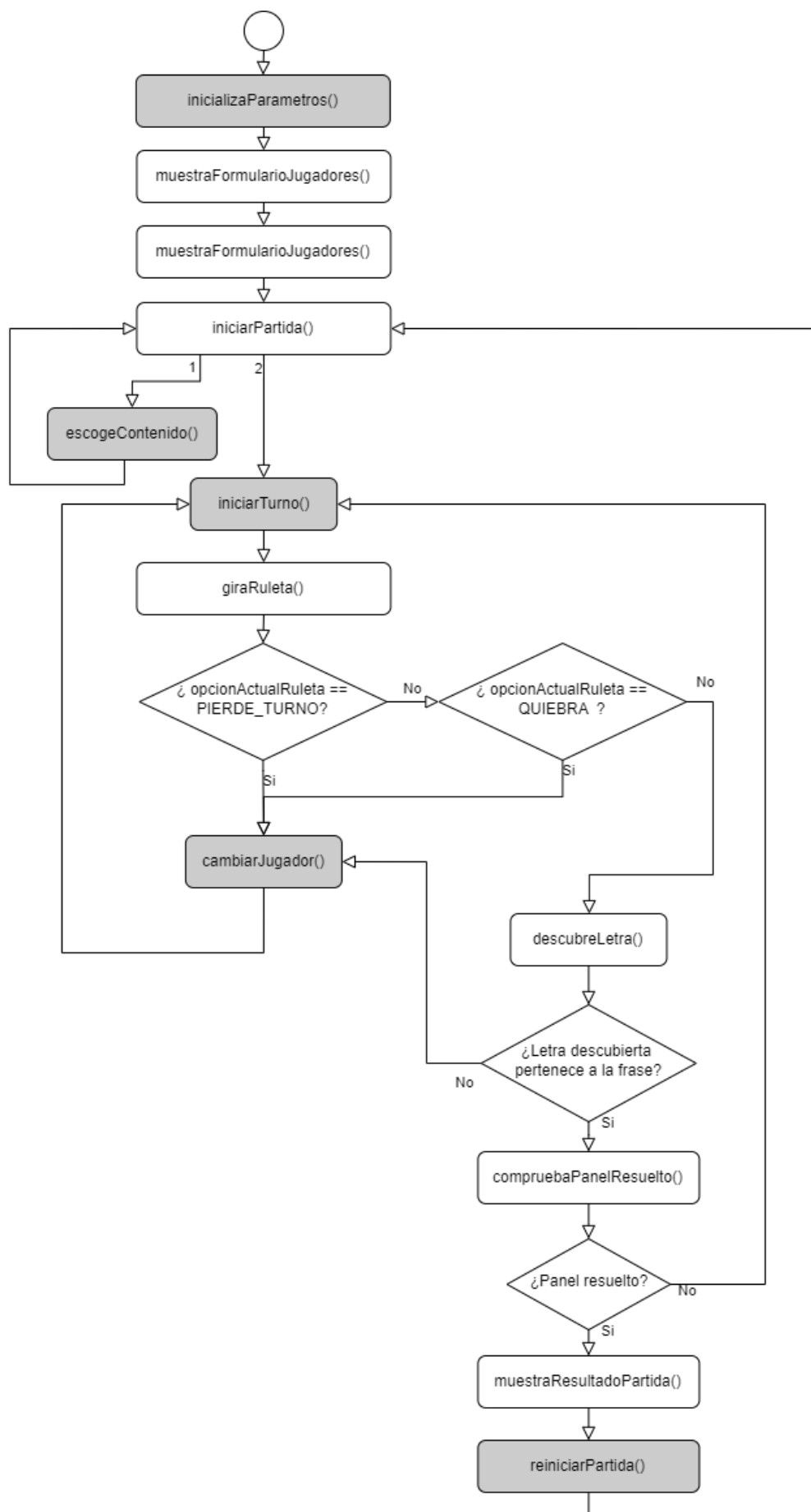
### Descripción del ejercicio propuesto

#### IMPORTANTE:

- Recuerde que el uso del atributo `innerHTML` no se tolera y su uso invalidará el ejercicio entregado.
- Los archivos proporcionados en el material adicional **NQ** deben ser modificados de ningún modo para implementar las funcionalidades requeridas.
- Todas las funciones JavaScript descritas deben ir en el fichero `funcionesRuletaSuerte.js`. Los comportamientos y plugins jQuery irán en el fichero `jQueryRuletaSuerte.js`.

Se va a implementar el juego de la ruleta de la suerte simplificado. No se comprarán vocales, ni se resolverá directamente, además solo se contemplarán letras mayúsculas de la A a la Z, excluyendo la letra Ñ, que será tratada como un signo de puntuación más. La finalidad es adivinar una frase a base de ir seleccionando posibles letras que la formarán.

Para ello, se ha diseñado un algoritmo en el que interaccionan una serie de funciones desarrolladas en JavaScript. En el siguiente diagrama puede observar cómo se relacionan estas funciones entre sí. Tenga en cuenta que las que aparecen sombreadas se encuentran definidas e implementadas en el fichero `funcionesProporcionadas.js`:





La descripción del contenido disponible en el fichero *funcionesProporcionadas.js* se detalla a continuación:

- Definición de una serie de variables globales y constantes que sirven para gestionar todo el juego. Ver, en los distintos comentarios, ejemplos de cada una de ellas, así como, en el método inicialización, como se inicializa el resto de variables necesarias.
- Del mismo modo, se definen las siguientes funciones que tendrá que usar para lograr que la aplicación funcione tal y como se ha indicado:
  - *escogeContenido()*. Obtiene aleatoriamente un elemento de *contenidosDisponibles* y lo almacena en la variable *contenidoSeleccionado*.
  - *iniciarTurno()*. Muestra los datos del jugador que posee el turno, habilita el botón *girarRuleta* y oculta el elemento con id *accionLetras*, añadiendo al mismo la clase CSS *oculto*.
  - *cambiarJugador()*. Pasa el turno al siguiente jugador y ejecuta la función *iniciarTurno*.
  - *muestraMensajeError(texto)*. Función que recibe una cadena de texto y la muestra como mensaje de error al usuario.
  - *muestraMensajeAcierto(texto)*. Función que recibe una cadena de texto y la muestra como mensaje de acierto al usuario.
  - *reiniciarPartida()*. Función que reinicia la aplicación para proceder a jugar una nueva partida, con los jugadores (y sus respectivas puntuaciones) tal y como han finalizado la última partida.

**EJ. 1 [8 puntos].** Partiendo de lo anteriormente expuesto, y de la estructura HTML definida en el fichero *ruletaSuerte.html*, implemente todo el código necesario en el fichero *funcionesRuletaSuerte.js* para dotar a la página de las funcionalidades descritas a continuación haciendo uso exclusivamente de JavaScript :

- a. **[1.25 puntos].** Función *muestraFormularioJugadores()*. Obtendrá el número de jugadores del campo *select* con id *numeroJugadores*. Se asegurará de que en el nodo *infoJugadores* no cuelga ningún otro nodo. Tras esto, añada tantos campos de texto al elemento con id *infoJugadores* como número de jugadores haya (usará estos campos de texto para recoger el nombre de cada uno de los jugadores). Por último, si el número de jugadores añadidos es mayor que cero, habilite el botón *iniciarPartida*.
- b. **[0.5 puntos].** Función *cargarInfoJugadores()*. Inicialice la variable global *infoJugadores*. Por cada jugador, se almacenará el nombre (obtenido de los campos de texto que añadió al nodo *infoJugadores*) y la puntuación inicial del mismo, que estará fijada en 100. Tras esto, llame a la función *iniciarPartida()*, definida en el siguiente apartado.
- c. **[1 punto].** Función *iniciarPartida()*. Inicialice el índice del jugador actual a 0. Llame a la función *escogeContenido()*, tras esto, usando el vector *contenidoSeleccionado*:
  - Reemplace todas las letras de la 'A' a la 'Z' de la frase (primera posición) por guiones (\_), salvo la letra Ñ y los signos de puntuación que pudiera haber (use la el método más adecuado de los que localizará en la documentación), y asigne el valor obtenido al elemento con id *panel*.
  - Asigne al elemento con id *pista* el valor de la pista (segunda posición de *contenidoSeleccionado*).
  - Asigne al elemento con id *valorPanel* el valor del panel (tercera posición de *contenidoSeleccionado*).

Oculte el elemento con id *configurarPartida*, añadiéndole la clase *oculto* para ello. Muestre el elemento con id *partida*, eliminando la clase *oculto* para lograr esto. Por último llame a la función *iniciarTurno()*.

- d. **[1.25 puntos].** Función *girarRuleta()*. En primer lugar, deshabilite el botón con id *girarRuleta*. A continuación, escoja de forma aleatoria un elemento del vector *opcionesRuleta*, *almacénalo en la variable global opciónActualRuleta* y muestre su valor en el elemento con id *resultadoRuleta*. En función al resultado obtenido, implemente la siguiente lógica:



- Si la opción obtenida es *Quiebra*: El turno del jugador ha finalizado, estableciendo su puntuación actual a 0. Muestre un mensaje de error indicando esto (use la función *muestraMensajeError()* para esto) y llame a la función *cambiarJugador()*.
  - Si la opción obtenida es *Pierde turno*: El turno del jugador ha finalizado. Muestre un mensaje de error indicando esto (use la función *muestraMensajeError()* para esto) y llame a la función *cambiarJugador()*.
  - En cualquier otro caso: Se ha obtenido un valor numérico. Muestre un mensaje de acierto (use la función *muestraMensajeAcierto()* para esto) y haga visible el elemento con id *accionLetras* (elimine la clase CSS *oculto* para lograr esto).
- e. **[1.75 puntos]**. Función *descubreLetra()*. Esta función recibe el botón que se ha pulsado al realizar la llamada (Ver líneas 53 a 78 del HTML). Deshabilite dicho botón y, a continuación, obtenga el contenido de texto de éste, que corresponderá a la letra pulsada.
- Si la frase del panel (primera posición del vector *contenidoSeleccionado*) no contiene esta letra, muestre un mensaje de error indicando que la letra no está en la frase (use la función *muestraMensajeError()* para esto) y llame a la función *cambiarJugador()*.
  - En caso de que la letra esté contenida en la frase a resolver, muestre la letra seleccionada en cada una de las posiciones en que aparece dentro de la frase en el elemento con id *panel*, cambiándola por el guión que corresponda, y cuente cuántas veces aparece. Tras esto, multiplique la opción que obtuvo al tirar la ruleta (almacenada previamente en *opcionActualRuleta*) por el número de repeticiones y súmele el valor obtenido a la puntuación del jugador actual. Por ejemplo, si la opción obtenida es 75 y el número de repeticiones de la letra es 4, se le ha de sumar a la puntuación del jugador  $75 \cdot 4 = 300$ . Por último, llame a la función *compruebaPanelResuelto()* (descrita a continuación).
- f. **[0.75 puntos]**. Función *compruebaPanelResuelto()*. Compruebe el valor mostrado en el elemento con id *panel* con la frase del vector *contenidoSeleccionado* (primera posición) y:
- Si no son iguales, el jugador puede seguir tirando la ruleta y descubriendo letras, por lo que debe llamar a la función *iniciarTurno()*.
  - En caso contrario, oculte el elemento con id *accionLetras* (añada la clase *oculto* para ello), sume el valor de resolver el panel (tercera posición del vector *contenidoSeleccionado*) a la puntuación del jugador actual, muestre un mensaje de acierto indicando el nombre del jugador que ha resuelto el panel (use la función *muestraMensajeAcierto()* para esto) y por último llame a la función *muestraResultadoPartida()* (Descrita a continuación).
- g. **[1.5 puntos]**. Función *muestraResultadoPartida()*. Muestre el elemento con id *resumenPartida* (elimine la clase CSS *oculto* para conseguir este efecto). A continuación, muestre el listado de jugadores ordenado de mayor a menor puntuación. Para ello, ordene el vector *infoJugadores* por la puntuación que ha obtenido cada jugador (segundo campo del vector) y muéstrelo en dicho orden como elementos de tipo *LI* dentro del nodo *OL* con id *ranking*.

**EJ. 2 [2 puntos]**. En el fichero *jQueryRuletaSuerte.js*, defina:

- a. **[1.25 puntos]** *Plugin* jQuery para validar campos de texto. Cree un plugin para validar que todos los campos de tipo texto tengan contenido y:
- En caso de detectar algún campo vacío, establezca color de fondo rojo. En caso contrario establezca color de fondo blanco.
  - Si todos los campos de texto tienen contenido, el *plugin* debe devolver el valor *true*, y en caso contrario muestre un mensaje de *log* por consola indicando que se han detectado errores y devolver valor *false*.

Haga uso de este *plugin* en la función *cargarInfoJugadores()* de forma que solo se ejecute la funcionalidad de dicha función en caso de que el *plugin* devuelva el valor *true*.



- b. **[0.75 puntos]** Comportamiento jQuery: Volver a Jugar. Defina el comportamiento jQuery necesario para ejecutar la función *reiniciarPartida()* en caso de hacer clic sobre el botón con id *volverAJugar*.

### Material suministrado

---

Como material adicional dispondrá del fichero "Material Adicional.7z" que contendrá:

- Carpeta *demo*, en la que localizará un ejemplo totalmente funcional de la aplicación propuesta.
- Carpeta documentación
  - *expresionRegular.txt*. Encontrará aquí una expresión regular para identificar las letras de la A a la Z.
  - *String.prototype.replace* - JavaScript.pdf. Descripción del método *replace*.
  - *String.prototype.replaceAll* - JavaScript.pdf. Descripción método *replaceAll*.
  - *String.prototype.search* - JavaScript.pdf. Descripción del método *search*.
  - *String.prototype.slice* - JavaScript .pdf. Descripción del método *slice*.
- Carpeta *base*:
  - *funcionesProporcionadas.js* Implementación de las variables globales, constantes y funciones suministradas, descritas anteriormente, algunas de ellas ofuscadas.
  - *funcionesRuletaSuerte.js* donde debe implementar las funciones JavaScript solicitadas. Se ha descrito cada una de ellas para que solo tenga que completar cada apartado.
  - *jQueryRuletaSuerte.js* donde debe implementar el *plugin* y el comportamiento jQuery requeridos.
  - *ruletasuerte.css* Hoja de estilo.
  - *ruletaSuerte.html* Fichero HTML de partida.
  - *jquery-3.5.1.js* Librería jQuery.