

Convocatoria: Extraordinaria

Ciclo: Desarrollo Aplicaciones Web

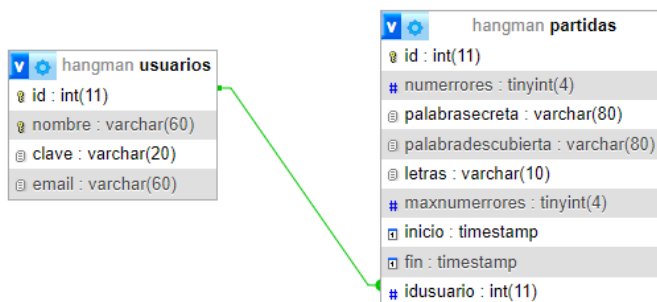
Módulo: Desarrollo Web en Entorno Servidor

JUEGO DEL AHORCADO

La aplicación base consiste en un proyecto de *Apache Netbeans* que corresponde a la aplicación del juego del ahorcado que debe ser actualizada con las modificaciones descritas en este enunciado.

El juego del ahorcado consiste en descubrir una palabra secreta enviando una a una las letras que pensamos que componen dicha palabra. Cada fallo supone añadir una parte del dibujo del ahorcado. Si la figura del ahorcado se completa entonces el jugador pierde la partida. En esta implementación del juego no se usan tildes ni la letra ñ.

Antes de comenzar la prueba asegúrate de que has creado la base de datos *hangman* formada por una tabla *usuarios* que contiene dos filas (1, 'pepe', '123456', 'pepe@gmail.com') y (2, 'luis', '654321', 'luis@gmail.com') y una tabla *partidas* con registros de partidas jugadas anteriormente por los usuarios. Puedes encontrar el esquema de la base de datos en la carpeta *bd* del proyecto suministrado.



La prueba consiste en extender la funcionalidad del juego del ahorcado aportado como punto de partida. **Al final de la prueba se entregará un único proyecto con todos los cambios realizados.** Las modificaciones/extensiones a realizar son las siguientes:

1 Buscador de partidas (9,5 pts)

Se trata de implementar una funcionalidad para buscar partidas almacenadas en la base de datos que cumplan unos criterios de búsqueda dados.

La aplicación base incluye un enlace en la cabecera *Buscador Partidas* que manda una petición al script *juego.php* para mostrar la vista *formbusqueda* que muestra un formulario básico con los campos correspondientes a los criterios de búsqueda.

Se trata de extender dicho código para implementar un buscador de partidas que valide los datos de búsqueda y muestre la lista de partidas encontradas.

La funcionalidad de validación de los datos del formulario y de presentación de las partidas recuperadas se implementará siguiendo estas directrices:

La validación del formulario de criterios de búsqueda se realizará en el servidor mediante código *PHP*, es decir, **no** se validará con atributos *HTML* (los tipos de las etiquetas input es *text*) y/o código *JavaScript*.

- Los campos del formulario son requeridos y deben ser rellenados para que sean valores válidos.
- Los campos del formulario y las restricciones que deben cumplir los datos introducidos en el formulario son los siguientes:
 - Rango de número de letras de la palabra secreta: Número menor y mayor de letras de la palabra secreta de las partidas seleccionadas.
 - Será una cadena del tipo `<número_mínimo>-<número_máximo>`, por ejemplo, `5-10`, ambos números serán menores de 30 y el número mínimo será menor que el número máximo.
 - Si no se cumplen las restricciones o el campo no ha sido rellenado se mostrará el mensaje *Rango inválido*.
 - Letras de la palabra secreta: Letras incluidas en la palabra secreta de las partidas seleccionadas.
 - Será una cadena con letras mayúsculas o minúsculas de longitud 1 a 25.
 - Si no se cumplen las restricciones o el campo no ha sido rellenado se mostrará el mensaje *Secuencia de letras inválida*.

En todos los casos, si los datos del formulario no cumplen con las restricciones descritas se volverá a presentar el formulario relleno con los datos erróneos y con los mensajes de error correspondientes.

- Completar el script de control *juego.php* para que todos los campos sean validados según las reglas descritas anteriormente **en el servidor**. Es importante que se respete el requisito de que la validación debe realizarse en el servidor.
- Extender la vista *formbusqueda* para que se muestren al usuario los mensajes personalizados de posibles errores cometidos en los valores del formulario. Se recomienda que se use la funcionalidad que ofrece *Bootstrap* para realizar validaciones en el servidor y proporcionar retroalimentación de los errores.

Ahorcado

Volver pepe

Buscar Partidas

Rango en el número de letras de la palabra secreta:

Rango inválido

Letras en la palabra Secreta:

Secuencia de letras inválida

Buscar Partidas

- Extender la clase *PartidaDAO* implementando un método *recuperaPorIdUsuario* para obtener los objetos de la clase *Partida* del usuario que está jugando en ese momento. Fíjate que los campos de las fechas de la clase *Partida* (*inicio* y *fin*) son de tipo *string* y que el campo de *IdUsuario* es tipo *int*. Implementa el siguiente método:

public function recuperaPorIdUsuario(int \$idUsuario): array

- Programar una función *obtenerPartidasPorCriteriosBusqueda* que filtre la lista completa de partidas del usuario aplicando los criterios de búsqueda y seleccione solamente aquellas que los cumplen. Implementa la siguiente función:

```
function obtenerPartidasPorCriteriosBusqueda(  
    array $partidas,  
    int $minNumLetras,  
    int $maxNumLetras,  
    string $letrasPalabraSecreta): array
```

- Completar el script de control *juego.php* para que pueda procesar la petición del formulario y validar los valores introducidos en el formulario de búsqueda, utilice el método *recuperaPorIdUsuario* del objeto de la clase *PartidaDAO* para recuperar la lista de partidas del jugador de la tabla de partidas e invoque la función *obtenerPartidasPorCriteriosBusqueda* para filtrar la lista de partidas y devolver solo las que cumplen los criterios de búsqueda. Finalmente invocar la vista *partidasencontradas* para crear la página que permita al usuario visualizar dicha la lista solicitada.
- Crear la vista *partidasencontradas* que muestra las partidas recuperadas de la base de datos. La lista de partidas incluye los siguientes datos sobre cada partida; palabra secreta, letras jugadas, número de errores, fecha de inicio de partida, hora de inicio de partida, fecha de fin de partida y hora de fin de partida, y si la partida ha sido ganada o no. La vista podría ser similar a la que se muestra a continuación. Si no se encuentra ninguna partida se mostrará un mensaje que indique *No hay partidas*. La vista también incluye un enlace para volver a la vista del juego una vez realizada la búsqueda.

Ahorcado

Volver pepe

Partidas Encontradas

Palabra Secreta	Letras	NumErrores	Fecha Inicio	Hora Inicio	Fecha Fin	Hora Fin	Ganada
TRUQUIFLOR	TRUQIFLO	0	2024-05-22	20:29:19	2024-05-22	20:29:57	Si
GRADUAR		0	2024-05-24	19:19:10	2024-05-24	19:19:22	No
SUDORAL		0	2024-06-11	16:43:05	2024-06-11	16:43:18	Si
REGUILETE		0	2024-06-11	16:45:52	2024-06-11	16:45:59	No
GAMBUZA		0	2024-06-11	16:45:59	2024-06-11	16:46:19	No
SISCA		0	2024-06-11	16:46:19	2024-06-11	16:46:25	No
HANSEATICO		0	2024-06-11	16:46:36	2024-06-11	16:47:13	Si
GANDARA		0	2024-06-11	19:27:28	2024-06-11	19:27:32	No
PARESTESIA		0	2024-06-11	19:27:37	2024-06-11	19:27:39	No
EXPLOTABLE		0	2024-06-11	19:27:40	2024-06-11	19:27:42	No
LONCO		0	2024-06-11	20:02:39	2024-06-11	20:25:52	No

Ahorcado

Volver pepe

Partidas Encontradas

Palabra Secreta	Letras	NumErrores	Fecha Inicio	Hora Inicio	Fecha Fin	Hora Fin	Ganada
No hay partidas							

Guía de evaluación

Es muy importante tener en cuenta que, para puntuar en las secciones descritas en la guía de evaluación, el código aportado deberá funcionar de acuerdo con lo especificado en el enunciado. No se tendrá en cuenta código que no implemente la funcionalidad requerida.

Buscador de partidas (9,5 pts)

Código PHP de validación de campos del formulario de búsqueda en el script de control

Código del script *juego.php* que recibe los datos y resuelve si hay un error en el rango de número de letras y el número máximo de errores. Cada campo validado correctamente aporta 0,5 pts.

Puntuación máxima: 1 pto

Formulario con retroalimentación de errores y sin pérdida de datos

Extensión del código de la vista *formbusqueda* que muestra los mensajes de error personalizados al usuario y mantiene los valores introducidos cuando sean erróneos.

Puntuación máxima: 1,5 pts

Recuperación de los objetos partida del usuario de la BBDD

Implementación del método *recuperaPorIdUsuario* de la clase *PartidaDAO* que devuelve un array de objetos *Partida* que pertenecen al usuario que está usando la aplicación. Se penalizará la nota si no se usan consultas preparadas con *PDO* y obtención de objetos con *FETCH_CLASS*.

Puntuación máxima: 1 pto

Filtrado de los objetos partida que cumplen los criterios de búsqueda

Implementación de la función *obtenerPartidasPorCriteriosBusqueda* que recibe el listado completo de partidas del usuario y devuelve solo las partidas que cumplen los criterios de búsqueda.

Puntuación máxima: 2 pts

Creación de la vista con el listado de las partidas recuperadas

Construcción de la vista *partidasencontradas* que muestra la lista de partidas recuperadas de la base de datos. Cada fila de la tabla contiene los datos requeridos en el enunciado.

Puntuación máxima: 2 pts

Código de script de control juego.php que procesa las peticiones para el buscador

Implementación del código del script *juego.php* que recibe y procesa las peticiones del formulario de búsqueda y de la solicitud de búsqueda con los criterios del formulario.

Puntuación máxima: 2 pts