

Convocatoria: Extraordinaria

Ciclo: Desarrollo Aplicaciones Web

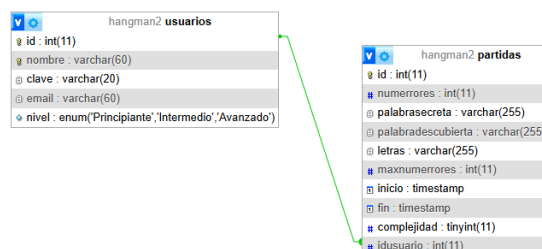
Módulo: Desarrollo Web en Entorno Servidor

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_25_2* formada por una tabla *usuarios* que contiene dos filas (1, 'pepe', '123456', 'pepe@gmail.com', 'Principiante') y (2, 'luis', '654321', 'luis@gmail.com', 'Avanzado') y una tabla *partidas* vacía. 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 Adecuación de la complejidad de la partida al nivel del usuario (9,5 pts)

Se trata de establecer la complejidad de las palabras jugadas al nivel de habilidad del usuario. Se definen tres niveles de capacidad: 'Principiante', 'Intermedio' y 'Avanzado'.

Cada jugador podrá establecer su nivel en el formulario de perfil que aparece cuando abre una sesión con la aplicación utilizando sus credenciales.

La complejidad de la palabra de una partida se califica con un entero que puede ir del 0 al 4, siendo 0 la complejidad más baja y 4 la más alta.

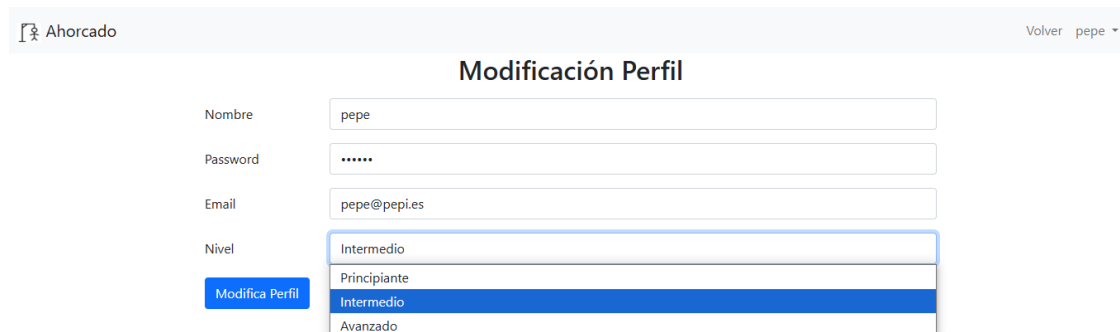
La correspondencia entre los niveles de los usuarios y las complejidades de las palabras con las que deben jugar será la siguiente:

| | Principiante | Intermedio | Avanzado |
|-------------|--------------|------------|----------|
| Complejidad | 0-1 | 2-3 | 4 |

Las partidas incluirán una propiedad adicional llamada complejidad de tipo entero que representará el valor de la complejidad de la partida. Esta información de la partida también se almacenará en la base de datos.

La funcionalidad de adecuación de la complejidad de la partida al nivel del usuario se implementará siguiendo estas directrices:

- Crear un tipo enumerado con valores asociados llamado *Nivel* que contenga los posibles valores del nivel del usuario: Principiante, Intermedio y Avanzado. Los valores asociados serán los mismos valores en formato cadena, es decir, 'Principiante', 'Intermedio', 'Avanzado'.
- Adapta el script *index.php* y la vista *formperfil* para que un usuario pueda establecer su perfil con las opciones posibles de *Nivel*. El elemento *select* es el más adecuado para completar el formulario de perfil con un menú desplegable. La vista de perfil se mostrará como en la imagen:



- Adapta el método *modifica* de la clase *UsuarioDAO* para que la información de nivel configurada por el usuario se persista en la BBDD.
- Crea una clase llamada *AnalizadorComplejidad* en una nueva carpeta del proyecto llamada *Servicios* cuyo objeto permita establecer la complejidad de una palabra en una escala de 0 a 4. Recuerda ubicar la clase en el espacio de nombres correspondiente. Las reglas para asociar una complejidad a una palabra serán las siguientes:
 - 0 => palabra de 5 a 8 letras, contiene solo las consonantes m, n, t, s, l, c, r, b, d, p y termina en ar, er o ir.
 - 1 => palabra 1 a 8 letras sin secuencias de 2 vocales seguidas.
 - 2 => palabra de 1 a 8 letras que no cumplen las reglas de la complejidad 0 y 1.
 - 3 => palabra de más de 8 letras sin z, x, q, k, h, y, w.
 - 4 => resto de palabras.

En dicha clase se define un método llamado *complejidadPalabra* que recibe una palabra y devuelve su complejidad como un número en la escala de 0 a 4. La especificación el método será:

public function complejidadPalabra(string \$palabra): int

- Añade una propiedad llamada *complejidad* de tipo entero a la clase *Partida* para almacenar la complejidad de la partida.
- Cambia el constructor de la clase *Partida* para que reciba dos nuevos parámetros de entrada correspondientes a un objeto de la clase *AnalizadorComplejidad* y una cadena que represente la complejidad requerida para la nueva partida. La cadena de complejidad puede representarse con un dígito o con dos dígitos separados por un guion para representar un rango de valores. Por ejemplo: '4' => Representa el valor 4, '0-2' => Representa los valores 0, 1, 2.

El nivel del usuario determinará la complejidad de las palabras con las que va a jugar de acuerdo con la tabla anterior que puedes incluir en la aplicación como un array asociativo en *juego.php*.

- Adapta el script *juego.php* para que suministre al constructor de la clase *Partida* un objeto de la clase *AnalizadorComplejidad* y la cadena de complejidad adecuada al jugador.
- Adapta el método *crea* de la clase *PartidaDAO* para que la propiedad adicional de *complejidad* se persista en los objetos de la clase *Partida*.

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 valorará código que no implemente correctamente la funcionalidad requerida.

Adecuación de la complejidad de la partida al nivel del usuario (9,5 pts)

Creación de tipo enum *Nivel* para el usuario

Creación del tipo enum *Nivel* con los valores adecuados para describir el nivel del usuario.

Puntuación máxima: 1 pto

Proceso de acción *botonproperfil*

Extensión del script *index.php* que recibe un campo adicional de nivel asociado al perfil del usuario y persiste el cambio.

Puntuación máxima: 0,5 pts

Adaptación de la vista de *formperfil*

Extensión de la vista *formperfil* para que muestre un menú desplegable para que cada usuario pueda configurar su perfil. Se penalizará si los valores del menú no se obtienen del tipo enumerado.

Puntuación máxima: 2 pts

Adaptación del método *modifica* de la clase *UsuarioDAO*

Modificación del método *modifica* de la clase *UsuarioDAO* para que el nivel del usuario se persista en la BBDD.

Puntuación máxima: 0,5 pts

Creación de la clase *AnalizadorComplejidad*

Implementación de la clase *AnalizadorComplejidad* con el método *complejidadPalabra* para que se realice el cálculo de la complejidad de una palabra. Se puntuará 0.5 pts cada rango de complejidad de la palabra bien identificado.

Puntuación máxima: 3 pts

Modificación del constructor de la clase *Partida*

Modificación del constructor de la partida para que construya partidas adecuadas al nivel del jugador.

Puntuación máxima: 2 pts

Adaptación del método *crea* de la clase *PartidaDAO*

Modificación del método *crea* de *PartidaDAO* para que se persista la propiedad *complejidad* de la partida.

Puntuación máxima: 0,5 pts