

Nombre del alumno o alumna:

Fecha de entrega: domingo, 24 de enero de 2021

Retroalimentación general de tarea de la unidad 4:

Hola ...

a

Calificación Total (SOBRE 10):

Expresión
incorrecta

Ejercicio				Calificación máxima	Calificación obtenida por apartado
Ejercicio 1: clase Opcion.				1,6	1,6
Subítems evaluables	21	Subítems correctos	21		

- (5 items) Se ha creado la interfaz *IGuardable* con los métodos:
 - *guardar*: no estático, parámetro instancia PDO, retorno filas insertadas/modificadas en la base de datos
 - *rescatar*: estático, parámetros instancia PDO e id elemento, retorno null o instancia
 - *borrar*: estático, parámetros instancia PDO e id elemento, retorno num. filas eliminadas.
- Se ha creado la clase *Opcion* con:
 - (2 items) Atributos:
 - *id*: privado, valor por defecto null
 - *idpregunta*: privado, valor por defecto null
 - *textoOpcion*: privado, no puede estar vacío (2-45)
 - *correcta*: privado, boolean
 - (3 items) Métodos:
 - Constructor: público, parámetros texto opción y correcta.
 - *getIdPregunta/setIdPregunta*: públicos
 - *getTextoOpcion/setTextoOpcion*: públicos
 - *getCorrecta/setCorrecta*: públicos
 - *getId* (no hay *setId*): públicos
- (8 items) Se implementan la interfaz *IGuardable*: métodos *guardar* (INSERT o UPDATE según caso), *rescatar* y *borrar*

Penalizaciones:

- (hasta -6 items) Se ha provisto de un mecanismo para modificar inadecuadamente el id de la opción.
- (hasta -4 items) Las clases o interfaces no se han creado en el archivo esperado.
- (hasta -9 items) En uno o más de los métodos se crea una conexión a la base de datos, cuando esa conexión debe pasarse por parámetro.

- (hasta -11 items) La ejecución con parámetros no esperados hace que script falle o entre en un bucle infinito.
- (hasta -6 items) Existen elementos de código innecesario o redundante.
- (hasta la totalidad del ejercicio) El ejercicio realizado no está conforme a lo esperado o lo entregado no corresponde a lo que se pide.

Retroalimentación:

¡Correcto!

Ejercicio 2: probando la clase Opcion				1,6	1,6
Subítems evaluables	30	Subítems correctos	30		

- (10 items) Ejercicio2nueva.php:
 - Si en la sesión no está almacenado el id de opción, se pregunta el texto de la opción y si es correcta o no usando un checkbox a través de un formulario.
 - Si en la sesión está almacenado el id de opción, se rescata la opción de la base de datos usando los métodos de la interfaz IGuardable. Si existe un error al rescatar, se informa del error adecuadamente y se muestra el formulario.
 - Al recibir los datos de formulario, se realiza el guardado (insert o update) en la base de datos usando los métodos de la interfaz IGuardable.
 - Si el guardado se ha realizado con éxito, la información de la opción se guarda en la sesión para ser usada por otros scripts.
- (10 items) Ejercicio2rescate.php:
 - Si en la sesión no está almacenado el id de opción, se pregunta el id de la opción.
 - Si en la sesión hay almacenado un id de opción, se usará dicho id de opción.
 - Se rescatan los datos de la opción usando los métodos de la interfaz IGuardable. En caso de error (id opción no existe, por ejemplo), se muestra el error adecuadamente.
 - Si el id opción existe, se guardara dicha información en la sesión para ser usada por otros scripts.
- (10 items) Ejercicio3borrar.php:
 - Si en la sesión no está almacenado el id de opción, se pregunta el id de la opción.
 - Si en la sesión hay almacenado un id de opción, se usará dicho id de opción.
 - Conocido el id de opción, se preguntará si se desea borrar la opción o no.
 - Una vez confirmada la opción, se procederá al borrado de la opción usando los métodos de la interfaz IGuardable. Si el borrado se llevo con éxito, se informa al usuario. SI el borrado no tuvo éxito (no existe opción), se informa al usuario. En cualquier caso, se elimina información de la opción en la sesión (para evitar errores en la recarga)

Penalizaciones:

- (hasta -10 items) No se hace uso adecuado del archivo settings.php en alguno de los tres scripts (p.e. se han añadido elementos no necesarios en dicho script) o no se declara la constante APP_ROOT_DIR adecuadamente.
- (hasta -10 items) La conexión a la base de datos no se realiza desde una función separada o se realizan múltiples conexiones en cada script, cuando una sola es necesaria.
- (hasta -10 items) Uno o más scripts acaban prematuramente por el uso de exit, die, o cualquier otro método similar.
- (hasta -15 items) La ejecución con parámetros no esperados hace que script falle o entre en un bucle infinito.
- (hasta -10 items) Existen elementos de código innecesario o redundante.
- (hasta la totalidad del ejercicio) En la tarea no se usa Smarty para generar la salida o no se utiliza como se espera (p.e. combinando salida estándar con plantillas smarty).
- (hasta la totalidad del ejercicio) El ejercicio realizado no está conforme a lo esperado o lo entregado no corresponde a lo que se pide.

Retroalimentación:

¡Correcto!

Ejercicio 3: clase Pregunta				1,6	1,6
Subítems evaluables	21	Subítems correctos	21		

- Clase Pregunta
 - (1 item) Es abstracta.
 - (2 items) Tiene los atributos:
 - id: privado (cuyo valor por defecto es null)
 - enunciado: solo accesible por subclases.
 - (7 items) Tiene los métodos:
 - Implementaciones del constructor: público (un único argumento, enunciado). Se establece el valor del atributo enunciado con el valor dado por parámetro.
 - Implementaciones públicas de los métodos getId, getEnunciado, setEnunciado. Se realiza la función esperada en estos métodos.
 - Declaración abstracta del método tipoPregunta.
- (11 items) Se implementan la interfaz IGuardable:
 - método guardar (INSERT o UPDATE según caso): guardando los datos correspondientes y haciendo uso del método de implementación futura “tipoPregunta”. En cualquier caso, se retorna el número de filas insertadas o

actualizadas.

- método borrar (DELETE): retorna el número de filas eliminadas.

Penalizaciones:

- (hasta -6 items) Se ha provisto de un mecanismo para modificar inadecuadamente el id de la pregunta.
- (hasta -4 items) Las clases o interfaces no se han creado en el archivo esperado.
- (hasta -9 items) En uno o más de los métodos se crea una conexión a la base de datos, cuando esa conexión debe pasarse por parámetro.
- (hasta -11 items) La ejecución con parámetros no esperados hace que script falle o entre en un bucle infinito.
- (hasta -6 items) Existen elementos de código innecesario o redundante.
- (hasta la totalidad del ejercicio) El ejercicio realizado no está conforme a lo esperado o lo entregado no corresponde a lo que se pide.

Retroalimentación:

¡Correcto!

Ejercicio 4: subclase <i>PreguntaEligeUna</i>				2	2
Subítems evaluables	21	Subítems correctos	21		

- (2 items) Se implementa la interfaz *IConjuntoOpciones* con los siguientes métodos:
 - *addOption*: un parámetro instancia de clase *Opcion*.
 - *getOpciones*: no parámetros.
 - *delOpcion*: un parámetro, posición numérica dentro del array de opciones de la pregunta.
- Clase *PreguntaEligeUna*:
 - (2 items) Cabecera:
 - Es una subclase de *Pregunta*.
 - Implementa la interfaz *IConjuntoOpciones*.
 - (1 item) Atributos:
 - Tiene un atributo privado para almacenar el conjunto de opciones (*\$opciones*).
 - Métodos:
 - (1 items) Implementa un constructor que hace uso del constructor de la clase padre. El enunciado es opcional (por defecto vacío).
 - (1 item) Implementa el método *tipoPregunta* y retorna “*EligeUna*”.
 - (3 items) Se implementan los métodos de la interfaz *IConjuntoOpciones* y la

funcionalidad esperada (manejar el array de opciones almacenado en el atributo \$opciones)

- (5 items) Se implementa el método estático rescatar, para implementar la pregunta y sus correspondientes opciones, usando para ello el método ya implementados de la clase Opcion. Si no existe la pregunta, se retorna null.
- (6 items) Se sobrescribe el método guardar, haciendo uso de la implementación de la clase padre y del método guardar de la clase Opcion. Se borran las opciones eliminadas haciendo uso del método borrar de la clase Opcion.

Penalizaciones:

- (hasta -6 items) Se ha provisto de un mecanismo para modificar inadecuadamente el id de la pregunta.
- (hasta -4 items) Las clases o interfaces no se han creado en el archivo esperado.
- (hasta -9 items) En uno o más de los métodos se crea una conexión a la base de datos, cuando esa conexión debe pasarse por parámetro.
- (hasta -11 items) La ejecución con parámetros no esperados hace que script falle o entre en un bucle infinito.
- (hasta -6 items) Existen elementos de código innecesario o redundante.
- (hasta la totalidad del ejercicio) El ejercicio realizado no está conforme a lo esperado o lo entregado no corresponde a lo que se pide.

Retroalimentación:

¡Correcto!

Ejercicio 5: probando la clase PreguntaEligeUna.				1,6	1,6
Subítems evaluables	21	Subítems correctos	21		

- (15 items) Ejercicio5nueva.php:
 - Si no hay un id de pregunta almacenado en la sesión se incluye un formulario con los datos de la pregunta: enunciado, texto de 6 opciones, opción correcta (radio button), botón guardar.
 - Se validan los datos del formulario:
 - Enunciado no vacío.
 - Al menos 2 opciones con texto.
 - Opción correcta no vacía.
 - Si los datos son válidos, se almacena en la base de datos usando **solo** el método guardar de la clase Pregunta (se hace uso indirecto del método guardar de la clase Opcion).

- Si el guardado fue correcto, se almacena en la sesión el ID de pregunta.
- Si hay un id de pregunta almacenado en la sesión, se muestra el contenido de la pregunta usando **solo** el método rescatar de la clase Pregunta (se hace uso indirecto del método rescatar de la clase Opcion). Aparece un botón para volver a mostrar el formulario.
- A la hora de guardar o rescatar, si se produce un error, debe mostrarse adecuadamente al usuario.
- (15 items) Ejercicio5borrar.php:
 - Si en la sesión hay almacenado el ID de una pregunta, usarás ese ID para recuperar los datos de la pregunta.
 - Si no hay un ID de pregunta almacenado en la sesión, se mostrará un formulario que pregunte por dicho ID.
 - Cuando hay un ID de pregunta conocido se muestra la pregunta y se pregunta si se desea borrar una opcion no correcta al azar (cuando hay mas dos opciones) o la pregunta (cuando solo quedan dos opciones).
 - Se realizan correctamente las operaciones de borrar opción al azar (usando delOpcion y guardar de la clase PreguntaEligeUna) y la de borrar pregunta según corresponda (usando borrar de la clase PreguntaEligeUna).
 - Si a la hora de rescatar, guardar o borrar se produce un error, debe mostrarse el error adecuadamente al usuario.

Penalizaciones:

- (hasta -10 items) No se hace uso adecuado del archivo settings.php en alguno de los tres scripts (p.e. se han añadido elementos no necesarios en dicho script) o no se declara la constante APP_ROOT_DIR adecuadamente.
- (hasta -10 items) La conexión a la base de datos no se realiza desde una función separada o se realizan múltiples conexiones en cada script, cuando una sola es necesaria.
- (hasta -10 items) Uno o más scripts acaban prematuramente por el uso de exit, die, o cualquier otro método similar.
- (hasta -15 items) La ejecución con parámetros no esperados hace que script falle o entre en un bucle infinito.
- (hasta -10 items) Existen elementos de código innecesario o redundante.
- (hasta la totalidad del ejercicio) En la tarea no se usa Smarty para generar la salida o no se utiliza como se espera (p.e. combinando salida estándar con plantillas smarty).
- (hasta la totalidad del ejercicio) El ejercicio realizado no está conforme a lo esperado o lo entregado no corresponde a lo que se pide.

Retroalimentación:

¡Correcto!

Ejercicio 6: subclase PreguntaEligeVarias.				1,6	1,6
Subítems evaluables	25	Subítems correctos	25		
<ul style="list-style-type: none">(5 ítems) Se implementa la clase PreguntaEligeVarias siguiendo las líneas indicadas en el ejercicio 5.<ul style="list-style-type: none">Se tiene en cuenta que puede haber más de una opción correcta (checkbox).(10 ítems) Ejercicio6nueva.php.(10 ítems) Ejercicio6borrar.php. <p>Penalizaciones:</p> <ul style="list-style-type: none">(hasta -7 ítems) No se hace uso adecuado del archivo settings.php en alguno de los tres scripts (p.e. se han añadido elementos no necesarios en dicho script) o no se declara la constante APP_ROOT_DIR adecuadamente.(hasta -7 ítems) Se ha provisto de un mecanismo para modificar inadecuadamente el id de la pregunta.(hasta -6 ítems) Las clases o interfaces no se han creado en el archivo esperado.(hasta -8 ítems) En uno o más de los métodos se crea una conexión a la base de datos, cuando esa conexión debe pasarse por parámetro.(hasta -13 ítems) La conexión a la base de datos no se realiza desde una función separada o se realizan múltiples conexiones en cada script, cuando una sola es necesaria.(hasta -13 ítems) Uno o más scripts acaban prematuramente por el uso de exit, die, o cualquier otro método similar.(hasta la totalidad del ejercicio) En la tarea no se usa Smarty para generar la salida o no se utiliza como se espera (p.e. combinando salida estándar con plantillas smarty).(hasta la totalidad del ejercicio) El ejercicio realizado no está conforme a lo esperado o lo entregado no corresponde a lo que se pide. <p>Retroalimentación:</p> <p>¡Correcto!</p>					