Universidad Nacional Abierta y a Distancia

Vicerrectoría Académica y de Investigación

Curso: Programación Web 1

Código: 204035

Guía de actividades y rúbrica de evaluación – Fase 3 Decidir y realización del proyecto

1. Descripción de la actividad

|  |  |
| --- | --- |
| Tipo de actividad: Independiente | |
| Momento de la evaluación: Intermedio | |
| Puntaje máximo de la actividad: 115 puntos | |
| La actividad inicia el:  martes, 4 de octubre de 2022 | La actividad finaliza el: lunes, 31 de octubre de 2022 |
| Con esta actividad se espera conseguir los siguientes resultados de aprendizaje:    Diseñar sitios web haciendo uso de lenguajes de programación del lado del cliente de acuerdo con los requerimientos del usuario y técnicas de codificación del software. | |
| La actividad consiste en:    Continuando con el desarrollo del sitio web de la fase 2, se deben construir e incorporar los nuevos requerimientos.    Parte 1: Scripts con programación PHP.  Parte 2: Construcción del sitio web.    PARTE 1    De manera individual cada estudiante consulta y ejemplifica los siguientes temas sobre lenguaje de programación PHP de acuerdo con las siguientes tablas:    Tabla No. 1. Funciones | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ¿Qué son funciones? | Una función es una agrupación de órdenes que mientras se prolongue el software en su código van a ser ejecutadas un montón de ocasiones. Es debido a aquello que esta agrupación de órdenes se junta en una función. Las funciones es posible que sean invocadas y ejecutadas desde toda parte del software. Fuera de esto, una función le es posible obtener parámetros externos de los que pende la ejecución de aquella función. Lo que significa que de acuerdo al parámetro o parámetros con los que se llame a la función; está mostrará un resultado u otro. Las funciones requieren encontrarse definidas antes de invocan a la función. | |  |
| Identifique algunas funciones | | |
| Denominación | Definición | Ejemplo. |
| round()  floor()  ceil()  rand() strtoupper()strtolower()ucfirst()ucwords()strlen()substr() trim() str\_replace()var\_dump()sort()rsort()shuffle()array\_push()array\_unshift()in\_array()count()explode()impode() | Con esta redondeamos; ósea obtiene un parámetro una cifra y en caso de que la tasación que está a la derecha de la coma es menor a 50 redondeará hacía el número actual de lo contrario lo cambiará por el número siguiente.  Con este redondeamos pero hacia un número menor  Está siempre redondeará al número siguiente  Está ejecuta mostrar una cifra random dentro de los que hay en orden ascendente desde un número hasta otro  Vuelve una sucesión de palabras que forman una oración en mayúscula  Vuelve una sucesión de palabras que forman una oración en minúscula  Vuelve el primer carácter de una sucesión de palabras que forman una oración en mayúscula  Vuelve el primer carácter de todo vocablo de una sucesión de vocablos que forman una oración en mayúscula  Nos mostrará cuantos símbolos tiene una sucesión de vocablo que forman una oración  Está es para escoger la cantidad de símbolos que va a exhibir una sucesión de vocablos que forman una oración el número 1 es la sucesión de vocablos que forman una oración, el número 2 es desde que vocablo deseamos aparezca comienza por el 0 y el último hasta cuál. Esta corta los espacios en una sucesión de vocablos que forman una oración tanto al principio como al final de la mismaRebuscar una sucesión de vocablos que forman una oración en particular y cambiarla por otra o cambiar un carácter por otraEsto nos mostrará el sistema de una variante cualesquiera.Organiza un array en el orden en que las palabras están en el alfabeto; hay que clarificar que la valoración del parámetro se hace por referencia, así que aquel se cambia directamenteHace lo mismo que sort pero la colocación la hace descendenteTrastocar el posicionamiento de un array además de eso se traslada por reseñaAdiciona un posicionamiento inédito en la finalización de un array. Obtiene 2 parámetros; el mismo array y el valor nuevo.Este hace lo del anterior, pero con la diferencia de que adiciona un inédito posicionamiento en el comienzoRebusca una tasación adentro de un array; si lo halla ejecuta true de lo contrario falseMuestra la cuantía de posicionamientos que tiene el arrayTransformará un string en un array. Mostrando como punto de referencia una tasación que aislará los distintos posicionamientos entre sí.Transforma un array en un string y enlazará una sucesión de vocablos que forman una oración con los diferentes posicionamientos por un delimitador. | <?php  $decimal = 1.75;  $decimal\_redondeado = round($decimal);  En este caso el resultado será: “2”  <?php  $decimal = 1.50;  $decimal\_redondeado = floor($decimal);  echo $decimal\_redondeado;  ?>  Esto compilará mostrando : “1”  <?php  $decimal = 1.50;  $decimal\_redondeado = ceil($decimal);  echo $decimal\_redondeado;  ?>  Que ejecutará mostrando= “2”  <?php  $numero\_al\_azar = rand(1, 9);  echo $numero\_al\_azar;  ?>  Puede ejecutar mostrando:” 1 , 2 3 ,4 ,5 ,6 ,7 ,8 o 9”, pero solo uno de ellos y puede ser cualquiera  <?php  $palabra = 'Me encantan las arepas venezolanas.';  $palabra\_mayuscula = strtoupper($palabra);  echo $palabra\_mayuscula;  ?>  lo que ejecutará será mostrar: “ME ENCANTAN LAS AREPAS VENEZOLANAS”  <?php  $palabra = 'No GRITES, habla bajito.';  $palabra\_minuscula = strtolower($palabra);  echo $palabra\_minuscula;  ?>  Lo que ejecutará será mostrar: “no grites, habla bajito.”  <?php  $palabra = 'fernando';  $nombre = ucfirst($palabra);  echo $nombre;  ?>  Lo que ejecutará será mostrar: “Fernando”  <?php  $palabra = 'fernando gonzalo gaitan';  $nombre\_completo = ucwords($palabra);  echo $nombre\_completo;  ?>  Lo que ejcuta es mostrar : “Fernando Gonzalo Gaitan”  <?php  $cancion = 'De aquel amor de música ligera, nada nos libra, nada más queda';  $cantidad\_de\_caracteres = strlen($cancion);  echo $cantidad\_de\_caracteres;  ?>  Y nos mostrará: “64”  <?php  $cancion = 'De aquel amor de música ligera, nada nos libra, nada más queda';  $cancion\_cortada = substr($cancion, 0, 11);  echo $cancion\_cortada;  ?>  Este ejecutará lo siguiente: “De aquel am”  <?php  $cadena = ' Hola ';  $cadena\_cortada = trim($cadena);  echo strlen($cadena) . '<br />';  echo strlen($cadena\_cortada) . '<br />';  ?>  Esto ejecutará lo siguiente: “6  4”;  <?php  $frase = 'Marge\_creo\_que\_odio\_a\_Mickael\_Jackson.';  $frase\_modificada = str\_replace('\_', '-', $frase);  echo $frase\_modificada;  ?>  Lo que ejcutará: “Marge-creo-que-odio-a-Mickael-Jackson.  “  <?php  $cadena = 'Manzana';  $numero = 5;  $valor\_logico = true;  $coleccion = array('Azul', 'Rojo', 'Verde');  echo '<pre>';  var\_dump($cadena) . '<br />';  var\_dump($numero) . '<br />';  var\_dump($valor\_logico) . '<br />';  var\_dump($coleccion) . '<br />';  echo '</pre>';  ?>  Esto ejecutará:  string(7) "Manzana"  int(5)  bool(true)  array(3) {  [0]=>  string(4) "Azul"  [1]=>  string(4) "Rojo"  [2]=>  string(5) "Verde"  }  <?php  $nombres = array('Juan', 'Pedro', 'Maria', 'Jorge', 'Ana');  sort($nombres);  foreach($nombres as $item){  echo $item . '<br />';  }  ?>  Esto ejecutará: “Ana  Jorge  Juan  Maria  Pedro”  <?php  $nombres = array('Juan', 'Pedro', 'Maria', 'Jorge', 'Ana');  rsort($nombres);  foreach($nombres as $item){  echo $item . '<br />';  }  ?>  Esto ejecuta: “Pedro  Maria  Juan  Jorge  Ana “   <?php  $nombres = array('Juan', 'Pedro', 'Maria', 'Jorge', 'Ana');  shuffle($nombres);  foreach($nombres as $item){  echo $item . '<br />';  }  ?>  Esto ejecuta: “Maria  Ana  Jorge  Juan  Pedro”  <?php  $nombres = array('Juan', 'Pedro', 'Maria', 'Jorge', 'Ana');  array\_push($nombres, 'Carlos');  foreach($nombres as $item){  echo $item . '<br />';  }  ?>  Esto ejecutará: Juan  Pedro  Maria  Jorge  Ana  Carlos”  <?php  $nombres = array('Juan', 'Pedro', 'Maria', 'Jorge', 'Ana');  array\_unshift($nombres, 'Carlos');  foreach($nombres as $item){  echo $item . '<br />';  }  ?>  Esto ejecuta: “Carlos  Juan  Pedro  Maria  Jorge  Ana”  <?php  $nombres = array('Juan', 'Pedro', 'Maria', 'Jorge', 'Ana');  $nombre\_a\_buscar = 'Jorge';  if(in\_array($nombre\_a\_buscar, $nombres)){  echo 'El nombre se encuentra en la lista';  }else{  echo 'El nombre NO se encuentra en la lista';  }  ?>  Esto ejecuta: “El nombre se encuentra en la lista si halla el valor si no mostrará:”El nombre No se encuentra en la lista”  <?php  $nombres = array('Juan', 'Pedro', 'Maria', 'Jorge', 'Ana');  $nombres\_total = count($nombres);  echo $nombres\_total;  ?>  Esto ejecuta: “5”  <?php  $paises\_cadena = 'Argentina - Bolivia - Brasil - Chile - Paraguay - Uruguay';  $paises\_arreglo = explode('-', $paises\_cadena);  foreach($paises\_arreglo as $item){  echo $item . '<br />';  }  ?>  Esto ejecutará:  “Argentina  Bolivia  Brasil  Chile  Paraguay  Uruguay”  <?php  $paises\_arreglo = array('Colombia', 'Ecuador', 'Peru', 'Venezuela');  $paises\_cadena = implode(' - ', $paises\_arreglo);  echo $paises\_cadena;  ?>  Esto ejecutará: “Colombia - Ecuador - Peru – Venezuela” |
|  |  |  |
| Fuente: el autor.    Tabla N. 2. Conceptos fundamentales sobre Programación Orientada a Objetos (POO).    ¿Qué es programación Orientada a  Objetos? Es un modelo de codificación que comienza a partir de la concepción de “objetos” como cimiento; estos tienen datos diseño de campos( en ciertas ocasiones igualmente aludidos como: “atributos o propiedades” y codificación a manera de métodos; los objetos tienen la aptitud de interactuar y cambiar los valores englobados en sus campos o atributos(estado) por media de sus métodos(comportamiento). Bastantes de los objetos prediseñados de los lenguajes de máquina modernos posibilitan el agrupamiento en bibliotecas o librerías, pero, bastantes de aquellos lenguajes de máquina posibilitan al usuario la invención de sus librerías. Determinada propiedad de la POO son herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento. Están la mayoría de los lenguajes de máquina de POO fundamentados en las concepciones de clases e instancias  Elementos de una clase y definición en PHP   |  |  |  | | --- | --- | --- | | Elemento | Definición | Ejemplo. | | Clase | Una clase es una plantilla que simboliza una entidad de la realidad y determina propiedades y métodos para aquellas | Esta es la clase Employee (Empleado) en el pedazo que a continuación vamos a mostrarde código, esta simbolizada la entidad del empleado.  <?php  **class** Employee  {  **private** $first\_name;  **private** $last\_name;  **private** $age;    **public** **function** \_\_construct($first\_name, $last\_name, $age)    {      $this->first\_name = $first\_name;      $this->last\_name = $last\_name;      $this->age = $age;    }    **public** **function** getFirstName()    {  **return** $this->first\_name;    }    **public** **function** getLastName()    {  **return** $this->last\_name;    }    **public** **function** getAge()    {  **return** $this->age;    }  }  ?> | | Propiedades | Son variables utilizadas para guardar datos relacionados con el objeto. Generalmente se tiene acceso a las propiedades de las clases por medio de objetos instanciados. Estas propiedades so de la clase “private” lo que quiere decir que únicamente es posible entrar en estas por medio de la parte interna sus clases | El pedazo del código anterior donde dice: **private** $first\_name;  **private** $last\_name;  **private** $age; | | Métodos | Son funciones que realizan operaciones concretas vinculadas con los objetos. Generalmente se utilizan para tener acceso y manejar las propiedades así como realizar acciones vinculadas a ello | del primer código que escribí el método es:    **public** **function** getFirstName()    {  **return** $this->first\_name;    }    **public** **function** getLastName()    {  **return** $this->last\_name;    }    **public** **function** getAge()    {  **return** $this->age;    } | | Objeto | En el código previo estaba el modelo fundamental de un clase en php, y luego instanciaremos la clase para que se posible usarla y de ello resultará un objeto una clase es una estructura y un objeto es algo que existe con el que se puede hacer que el código haga cualquier cosa que desees. | ?php  class eat  {      function comer()      {          echo "Comiendo\_algo.";      }  }  $bar = new eat;  $bar->comiendo\_algo();  ?> |   Fuente: el autor.    Ejercicio de transferencia.      el resultado:  Graphical user interface, text, application, website  Description automatically generated  Graphical user interface, application  Description automatically generatedGraphical user interface, application  Description automatically generated  Graphical user interface  Description automatically generatedA screenshot of a computer  Description automatically generatedA screenshot of a computer  Description automatically generated | | |

|  |
| --- |
| Una vez ha recordado los conceptos fundamentales de la POO y ha revisado su aplicación en PHP, se debe realizar el siguiente ejercicio.    Crear un sitio web donde la interfaz de usuario cuente con los siguientes requerimientos.    En la imagen se visualiza una  interfaz de usuario que contiene en la parte superior el texto Introduzca su número de cedula seguido de un rectángulo. En la línea siguiente se encuentra el texto Elija el servicio que busca del banco:  A continuación, se visualiza tres opciones seleccionables con los nombres de Prestamos, Transferencia o Atención personalizada. En la parte inferior se encuentran dos botones: Enviar y Limpiar.    Requerimientos:     1. El sitio debe contener una clase en PHP dotada de capacidad para generar las cajas de texto <input type="Text"> y las casillas de verificación<input type="checkbox">. 2. Las opciones del elemento <input type="checkbox">, se deben construir a partir del contenido de un array. 3. El elemento <input type="Text">, cambia de color de fondo dependiendo el parámetro de color que reciba. 4. Al hacer clic en el botón enviar, un script PHP, recibe las variables y las imprime por pantalla.     En el tema “Construcción de Formularios Dinámicos en HTML con Programación PHP Orientada a Objetos, Recuperado de [https://repository.unad.edu.co/handle/10596/39422”](https://repository.unad.edu.co/handle/10596/39422), encuentra un video donde se desarrolla un ejercicio similar. Como resultado se deben |

|  |
| --- |
| obtener tres archivos con extensión PHP de la misma forma que lo muestra el video.    Como producto final de la parte 1, se debe crear un documento descrito a continuación:    Primera página: Portada.  Segunda página: Introducción (incluye objetivos).  Tercera página y siguientes: Desarrollo de la Tabla No. 1 y la Tabla No.  2.  Antepenúltima página: Conclusiones.  Ultima Página: Bibliografía.    Script PHP:  Tres scripts PHP del ejercicio de transferencia.    En el repositorio en línea creado en la fase 1, almacenar el documento y los Script en un subdirectorio llamado “EjercicioDeTranferencia(v2).    PARTE 2    Para continuar la construcción del sitio web, se añade los elementos según los nuevos requerimientos establecidos para complementar el sitio web para el banco MySavings.    Requerimientos:    Crear un formulario para solicitar la información del cliente del banco, el cual consta de:   * Número de Documento de Identidad * Tipo de documento de identidad (El cual se debe elegir de una lista) * Nombres y Apellidos completos * Fecha de nacimiento corta * Número celular * Dirección de residencia * Departamento de residencia * Ciudad de residencia * Estado Civil (El cual se debe elegir de una lista) y - Correo Electrónico. |
| Nota: Para el desarrollo de estos requerimientos es necesario usar clases y arreglos en PHP. En esta fase aún no se requiere conexión a bases de datos, por lo que es necesario almacenar los registros en arreglos.    Cada estudiante debe dejar evidencia la trazabilidad del desarrollo en el foro colaborativo de la actividad para recibir comentarios de sus compañeros y tutor. La estructura y organización del desarrollo de la actividad se debe ubicar en el subdirectorio “*PaginaWeb-MySavings(v2)”,* estesubdirectorio debe contener organizados los archivos HTML, SCC, JS y PHP. |
| Para el desarrollo de la actividad tenga en cuenta que:    En el entorno de Información Inicial debe: revisar la agenda del curso para indagar por la unidad académica relacionada con la actividad, la descripción, el tipo de actividad, las fechas de inicio y cierre y el peso evaluativo asignado.    En el Entorno de Aprendizaje debe: Publicar en el foro de la actividad, propuestas en código PHP, para atender los nuevos requerimientos. Posteriormente definir los scripts que se integrarán al desarrollo web.    En el Entorno de Evaluación debe: Cada estudiante debe hacer entrega de su trabajo consolidado con el desarrollo de todos los puntos propuestos en esta guía de actividades. |
| Evidencias de trabajo independiente:  Las evidencias de trabajo independiente para entregar son:   * Participación sustancial en el foro de la actividad. * Documento resultado de la PARTE 1 y 2 con el enlace de los archivos fuentes del proyecto. |
| Evidencias de trabajo grupal:  En esta actividad no se requieren evidencias de trabajo grupal. |

1. Lineamientos generales para la elaboración de las evidencias de aprendizaje a entregar.

|  |
| --- |
| Para evidencias elaboradas independientemente, tenga en cuenta las siguientes orientaciones     1. Documentarse sobre las temáticas propuestas para dar respuesta a las preguntas orientadoras. 2. Realizar comentarios a las respuestas de los compañeros en el foro colaborativo. 3. Dejar trazabilidad del desarrollo del proyecto web en el foro donde se evidencie la interacción con sus compañeros y tutor. 4. Validar que en el repositorio compartido se encuentren todos los archivos fuentes del proyecto. Así mismo, se debe garantizar que el enlace abra correctamente. 5. Entregar el trabajo final en el entorno de evaluación en la fecha establecida en la agenda de actividades. |
| Tenga en cuenta que todos los productos escritos individuales o grupales deben cumplir con las normas de ortografía y con las condiciones de presentación que se hayan definido.  En cuanto al uso de referencias considere que el producto de esta actividad debe cumplir con las normas APA |
| En cualquier caso, cumpla con las normas de referenciación y evite el plagio académico, para ello puede apoyarse revisando sus productos escritos mediante la herramienta Turnitin que encuentra en el campus virtual.    Considere que en el acuerdo 029 del 13 de diciembre de 2013, artículo 99, se considera como faltas que atentan contra el orden académico, entre otras, las siguientes: literal e) “El plagiar, es decir, presentar como de su propia autoría la totalidad o parte de una obra, trabajo, documento o invención realizado por otra persona. Implica también el uso de citas o referencias faltas, o proponer citad donde no haya coincidencia entre ella y la referencia” y liberal f) “El reproducir, o copiar con fines de lucro, materiales educativos o resultados de productos de investigación, que cuentan con derechos intelectuales reservados para la Universidad” |

Las sanciones académicas a las que se enfrentará el estudiante son las siguientes:

1. En los casos de fraude académico demostrado en el trabajo académico o evaluación respectiva, la calificación que se impondrá será de cero puntos sin perjuicio de la sanción disciplinaria correspondiente.
2. En los casos relacionados con plagio demostrado en el trabajo académico cualquiera sea su naturaleza, la calificación que se impondrá será de cero puntos, sin perjuicio de la sanción disciplinaria correspondiente.

3. Formato de Rúbrica de evaluación

|  |  |
| --- | --- |
| Tipo de actividad: Independiente | |
| Momento de la evaluación: Intermedio | |
| La máxima puntuación posible es de 115 puntos | |
| Primer criterio de evaluación:    Presenta el documento de acuerdo con las indicaciones formuladas Configurado en normas APA.    Este criterio representa 25 puntos del total de 115 puntos de la actividad. | Nivel alto: Se tuvieron en cuenta las especificaciones del documento y se aplicaron las normas APA:  Si su trabajo se encuentra en este nivel puede obtener entre 13 puntos y 25 puntos    Nivel Medio: No se tuvieron en cuenta todas las especificaciones del documento y/o no se aplicaron las normas APA.  Si su trabajo se encuentra en este nivel puede obtener entre 6 puntos y 12 puntos    Nivel bajo: No se tomaron en cuenta las especificaciones del documento y se omitió la aplicación de las normas APA.  Si su trabajo se encuentra en este nivel puede obtener entre 0 puntos y 5 puntos |
| Segundo criterio de evaluación: | Nivel alto: Construye el ejercicio propuesto con los tres archivos PHP que dan solución al requerimiento establecido. |
| Comparte en el foro de la actividad los aportes de la parte 1: Scripts en programación PHP.    Este criterio representa 45 puntos del total de 115 puntos de la actividad. | Si su trabajo se encuentra en este nivel puede obtener entre 21 puntos y 45 puntos    Nivel Medio: Construye el ejercicio propuesto con los tres archivos PHP, pero no dan solución al requerimiento establecido.  Si su trabajo se encuentra en este nivel puede obtener entre 6 puntos y 20 puntos    Nivel bajo: No construye el ejercicio propuesto con los tres archivos PHP que dan solución al requerimiento establecido.  Si su trabajo se encuentra en este nivel puede obtener entre 0 puntos y 5 puntos |
| Tercer criterio de evaluación:    Construye scripts en lenguaje PHP para dar solución a los nuevos requerimientos del usuario.    Este criterio representa 45 puntos del total de 115 puntos de la actividad. | Nivel alto: Construye scripts en lenguaje PHP para dar solución a los nuevos requerimientos del usuario.  Si su trabajo se encuentra en este nivel puede obtener entre 21 puntos y 45 puntos    Nivel Medio: Construye scripts en lenguaje PHP, pero no soluciona todos los requerimientos del usuario  Si su trabajo se encuentra en este nivel puede obtener entre 6 puntos y 20 puntos    Nivel bajo: No construye scripts en lenguaje PHP o la solución no resuelve ningún requerimiento del usuario. Si su trabajo se encuentra en este nivel puede obtener entre 0 puntos y 5 puntos |