

Tarea online PPS01.

Título de la tarea: Evaluación de software

Unidad: 1

Ciclo formativo y módulo: Curso de especialización en ciberseguridad en entornos de las tecnologías de la información - Puesta en Producción Segura.

Curso académico: 2023-2024

¿Qué contenidos o resultados de aprendizaje trabajaremos?

Resultados de aprendizaje

- ✓ **RA1.** Entender cómo funciona los sistemas de evaluación y testeo de software.
- ✓ **RA2.** Ser capaz de desarrollar pruebas unitarias sobre distintos elementos de nuestro código.

Contenidos

- 1.- Introducción.
 - 1.1.- Fundamentos de la programación.
 - 1.2.- Modelos de ejecución de software.
 - 1.3.- Elementos básicos.
 - 1.4.- Pruebas de software.
 - 1.5.- Evaluación de lenguajes de programación.

1.- Descripción de la tarea.



Caso práctico

La mayoría de los lenguajes de programación disponen de librerías que simplifican el proceso de creación y ejecución de las pruebas unitarias o de examen de unidad. Por ejemplo, en PHP está PHPUnit, en Java JUnit y en Python unittest. Los pasos a seguir para crear el programa de pruebas suelen ser siempre los mismos:

- Importar la librería de test de software (PHPUnit en PHP)
- Definir una clase donde incluiría la función para comprobar esta parte específica del código (normalmente el nombre debe comenzar con test).
- Realizar los casos de prueba necesarios (para los cuales se suele utilizar el método assertEquals)

Julián ha empezado a trabajar con un nuevo equipo de desarrollo que trabaja con Visual Studio Code como entorno de desarrollo y PHP como lenguaje. Ha instalado Visual Studio Code, PHP y ha configurado el IDE para poder trabajar de la mejor forma posible con PHP, incluyendo poder realizar las pruebas unitarias (ha sido necesario instalar PHPUnit).

Para comprobar que todo está bien ha creado el siguiente programa php que ha llamado **ejemplo.php**:

```
<?php
function obtenerPrimerosNumeros($n)
{
    $numeros = array();
    for ($i = 1; $i <= $n; $i++) {
        $numeros[] = $i;
    }
    return $numeros;
}
?>
```

Y para realizar las pruebas ha creado el siguiente programa php que ha llamado **ejemploTest.php**:

```
<?php
require 'ejemplo.php';

// Importa el archivo con la función a probar
use PHPUnit\Framework\TestCase;
```

```
// utiliza la librería con PHPUnit

class EjemploTest extends TestCase {

    // crea una clase que hereda de TestCase

    public function testObtenerPrimerosNumeros() {

        // crea un método para realizar las pruebas

        $n = 5;

        $resultado_esperado = [1, 2, 3, 4, 5];

        $resultado_obtenido = obtenerPrimerosNumeros($n);

        $this->assertEquals($resultado_esperado, $resultado_obtenido);

        // comprueba que el resultado esperado es igual al obtenido

    }

}

?>
```

Ha ejecutado las pruebas unitarias y el resultado es correcto así que ya está preparado para empezar a trabajar en el nuevo proyecto.

En el nuevo equipo le han pedido desarrollar un pequeño programa que **devuelve la serie de Fibonacci hasta un número dado**. Como es lógico, debe hacer un test del software de tipo examen de unidad o prueba unitaria, es decir quiere verificar que una parte del código se comporta de forma esperada.

Julián ha pensado que como caso de prueba comprobará si el quinto número de la serie Fibonacci es el número 3, en cuyo caso es correcto (la serie Fibonacci es 0,1,1,2,3,5,8,13...).

¿Qué te pedimos que hagas?

Antes de nada reflejaremos la sucesión de Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34... Tienes mas información sobre esta interesante sucesión y su origen [aquí](#).

✔ Apartado 1: Configurar el entorno

Para realizar la tarea los pasos son:

- Instalar PHP junto con la librería PHPUnit
- Instalar Visual Studio Code y configurar el entorno para poder trabajar con PHP y poder realizar las pruebas unitarias en el mismo.

✔ Apartado 2: Rellenar la siguiente tabla utilizando el código del programa de pruebas que se ha dado como ejemplo

Indica una línea en qué aparece el elemento del lenguaje correspondiente utilizando el archivo **ejemploTest.php** que se ha dado como ejemplo

ELEMENTO DEL LENGUAJE	LINEA
-----------------------	-------

Comandos de pre-procesamiento	
Comentario	
Variable	
Función/método	
Sentencia	

✔ Apartado 3: Crear un programa PHP que genere la secuencia de Fibonacci.

Se pide crear un pequeño programa PHP para generar la secuencia de Fibonacci. Se ha escogido PHP porque se va a utilizar en las unidades 2 y 3 y además, es uno de los lenguajes más utilizados en las aplicaciones web

Para realizar la tarea los pasos serán:

- Crearemos el archivo **fibonacci.php**
- Debes crear una función de nombre fibonacci donde incluirás:
 - Una línea de comentario donde escribirás tu nombre.
 - Otros comentarios en las líneas que creas importantes
 - Utilizar variables
 - Utilizar alguna estructura de control
 - Utilizar el control de errores (mira este enlace <https://www.php.net/manual/en/language.exceptions.php>).

✔ Apartado 4: Creación del programa de pruebas unitarias

Crearemos un programa principal donde definiremos una clase llamada Test para probar nuestro código (tienes un ejemplo en el enunciado de la tarea)

Para realizar la tarea los pasos serán:

- Crearemos un archivo de nombre **testFibonacci.php**
- Importaremos la librería de testeo de software en este caso PHPUnit.
- Importaremos el archivo donde se ha definido la función fibonacci
- Crearemos nuestra clase (del tipo TestCase)
- Dentro de esta clase definiremos una función (es recomendable un nombre ilustrativo que empiece por test)
- Dentro de la función reflejaremos el tipo de testeo que vamos a realizar. En este caso nuestro objetivo es verificar si la posición X de la sucesión de Fibonacci coincide con el resultado esperado, es decir coincide con la posición X que devuelve nuestro programa. Es decir queremos verificar que el quinto número de la sucesión, que es 3, coincide con el quinto número que devuelve nuestra secuencia.
 - Para hacer esta comprobación usaremos el tipo de comprobación **assertEquals** que comprueba si dos valores son iguales
 - Mediante esta función comprobaremos si la posición quinta de nuestra función coincide con el valor esperado (el valor esperado es la quinta posición de la sucesión que es 3). Recuerda que los arrays en PHP empiezan en la posición 0

✔ Apartado 5: Verificación de software

Ejecutamos la prueba y comprobamos que no hay error

Pasos a realizar:

- Ejecutaremos el programa final y verificaremos si realmente nuestro programa que calcula la sucesión de Fibonacci se comporta como esperamos.
- Si el programa que comprueba el código detecta un error nos reflejará que dato está esperando y que ha recibido

✔ Apartado 6: Responder a la pregunta

Dado el siguiente código PHP que comprueba la conexión a una base de datos MYSQL y que si se ejecuta funciona:

```
1 <?php
2 require_once 'pdoconfig.php';
3 $servername = "db1";
4 $database = "databasename";
5 $username = "username";
6 $password = "password";
7
8 try {
9     $conn = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
10    echo "Conectado a $dbname de $host con exito.";
11 } catch (PDOException $pe) {
12    die("No es posible la conexión a $dbname ." . $pe->getMessage());
13 }
14 ?>
```

Si realizamos un análisis estático de código nos va a devolver un error del tipo **"Hard-coded credentials are security-sensitive"** ¿Por qué? Razona la respuesta. Indica la(s) línea(s) que habría que cambiar

NOTA IMPORTANTE

Para todos los apartados es necesario entregar las capturas de pantalla de los principales pasos realizados, explicando el proceso seguido en cada uno de ellos. Las capturas de pantalla realizadas deben tener como fondo de pantalla la plataforma con tu usuario mostrando claramente la foto de tu perfil. Aquellos apartados/subapartados que no cumplan esta condición no serán corregidos.

2.- Información de interés.

Recursos necesarios y recomendaciones

Recursos necesarios

- ✓ Ordenador personal con, al menos, 4 Gigabytes de memoria RAM
- ✓ Conexión a Internet para consultar ejemplos de la Unidad 1.
- ✓ Sistemas Operativos Windows 10, Ubuntu 18.04, Ubuntu 20.04
- ✓ Navegador web.
- ✓ Software para edita código [Microsoft VisualStudio Code.](#)

Recomendaciones

- ✓ Antes de abordar la tarea:
 - ➡ lee con detenimiento la unidad, consulta los enlaces para saber más, examina el material proporcionado por el profesor y aclara las dudas que te surjan con él.
 - ➡ Realiza el examen online de la unidad, y consulta nuevamente las dudas que te surjan. Solo cuando lo tengas todo claro, debes abordar la realización de la tarea.
- ✓ El lenguaje a utilizar será PHP puesto que se va a utilizar también en las dos próximas unidades.
- ✓ No olvides elaborar el documento explicativo.



Indicaciones de entrega

Una vez realizada la tarea, el envío se realizará a través de la plataforma. El archivo se nombrará siguiendo las siguientes pautas:

Apellido1_Apellido2_Nombre_PPS_Tarea01

3.- Evaluación de la tarea.

Criterios de evaluación implicados

Criterios de evaluación RA1

- ✓ a. Se ha valorado la importancia de entender la necesidad de la evaluación del software para garantizar la calidad del mismo.
- ✓ b. Se ha entendido qué componentes tenemos dentro de nuestro código fuente.
- ✓ c. Se ha comprendido cómo se relacionan los distintos elementos del código fuente (importaciones, sentencias, funciones)
- ✓ d. Se ha entendido como funcionan los lenguajes interpretados.
- ✓ e. Se ha comprobado cómo afecta un intérprete en la ejecución de software.
- ✓ f. Se ha entendido el proceso de creación de un programa.

Criterios de evaluación RA2

- ✓ a. Se han entendido como integrar los sistemas de evaluación de software en nuestro código.
- ✓ c. Se ha comprendido los distintos tipos de prueba de software, sus necesidades, sus resultados y las necesidades que genera.
- ✓ d. Se ha entendido todo el proceso de creación de software y evaluación del mismo.
- ✓ e. Se ha entendido cómo el software en ejecución puede generar valores no esperados.

¿Cómo valoramos y puntuamos tu tarea?

Rúbrica de la tarea

Apartado 1: Realiza la instalación de PHP junto con la librería PHPunit, y del IDE Visual Studio Code. Además, configura el IDE para poder trabajar con PHP y realizar las pruebas unitarias directamente.

1,5 puntos

Apartado 2: Rellena la tabla identificando los distintos elementos del lenguaje en el programa de ejemplo	1 puntos
Apartado 3: Crea el programa solicitado con la función que genera la serie de Fibonacci. El código es claro y descriptivo y tiene los elementos que se le solicitan	1,5 puntos
Apartado 4: Crea el programa para realizar las pruebas importando las librerías correctas, tanto para hacer las pruebas unitarias como para importar la función creada anteriormente.	1 puntos
Apartado 4: Crea e instancia una clase de manera correcta. El código es claro y conciso y tiene una estructura ordenada que permita realizar no solamente la prueba que buscamos sino realizar pruebas adicionales.	1 puntos
Apartado 4: Invoca la función exacta que buscamos (assertEqual) de una manera clara con los valores que necesitamos comprobar	1 puntos
Apartado 4: La estructura del código permite realizar mas pruebas de forma sencilla mediante la parametrización de las variables	1 puntos
Apartado 5: La prueba de software es exitosa	1 puntos
Apartado 6: Responde a las preguntas de forma razonada.	1 puntos

NOTA IMPORTANTE

Aquellos apartados/subapartados en los que las capturas de pantalla no sean claras o no tengan como fondo de pantalla la plataforma con tu usuario mostrando claramente la foto de tu perfil, no serán corregidos.