**Plan de Testing**

**Objetivos**

Asegurar que el software cumple con los requisitos funcionales y no funcionales.

Identificar y corregir errores en una etapa temprana del desarrollo.

Garantizar la calidad y confiabilidad del software.

**Alcance**

El plan de testing cubrirá las siguientes áreas:

Testing Unitario: Se verificará que cada componente del software funcione de manera aislada y cumpla con sus especificaciones.

Testing de Integración: Se evaluará la interacción entre los diferentes componentes del software.

Testing de Sistema: Se probará el software como un todo, verificando que todas las partes funcionen correctamente.

Testing de Rendimiento: Se medirá el rendimiento y la escalabilidad del software bajo diversas condiciones.

Testing de Seguridad: Se identificarán y eliminaran posibles vulnerabilidades.

**Testing Unitario**

**Objetivo**

Verificar que las funciones y métodos individuales de la aplicación web funcionen correctamente.

**Herramientas de Pruebas:** PHPUnit.

Paso 1: Configuración del Entorno de Pruebas

Configuramos un archivo de pruebas que utilize una base de datos de pruebas en lugar de la de producción.

Paso 2: Identificación de Funciones y Métodos a Probar

Enumeramos las funciones y métodos individuales que precisamos probar de la aplicación.

Paso 3: Creación de Casos de Prueba

Para cada función o método que se necesita probar:

Se crea un caso de prueba unitaria utilizando el marco de pruebas, es este caso PHPunit, creando una clase que herede de PHPUnit\Framework\TestCase y se definen los métodos de prueba.

Paso 4: Ejecución de las Pruebas

Ejecutamos las pruebas unitarias utilizando el comando proporcionado por el marco de pruebas, por ejemplo:

phpunit pruebas\_tutorias.php

Nos aseguramos de que las pruebas pasen sin errores. Si alguna prueba falla, identificamos y solucionamos el problema en la función o método correspondiente.

Paso 5: Informe de Resultados

En este caso todas las pruebas han salido exitosas y no se encontraron errores.