



UNIVERSIDAD DE
GUADALAJARA

Red Universitaria e Institución Benemérita de Jalisco

Herramientas para manejar errores parte 2

29/Agosto/2023

Hector Emmanuel Garcia Garcia

217585177

Michel Emanuel López Franco

Computación tolerante a fallas 2023B

En esta segunda parte enseñare un ejemplo en php con una interfaz de usuario en HTML en un local host, simplemente es un formulario sencillo que pedirá 2 números para dividirlos y probaremos si los elementos del documento anterior nos facilitan la tarea.

Ejemplo sin el type number

En este caso en el código se puede ver que los inputs que reciben el código no tienen tipo número, por lo que significa que por defecto están esperando un texto.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Excepciones</title>
</head>
<body>
  <h3>Divisiones</h3>
  <h4>Elige 2 numeros para dividirlos y tener la respuesta</h4>
  <form method="post" name="division" id="division">
    <label for="divisor">Elige un divisor</label>
    <input id="divisor" name="divisor" /> <!--Sin type number-->
    <label for="dividendo">Elige un dividendo</label>
    <input id="dividendo" name="dividendo" /> <!--Sin type number-->
    <input type="submit" name="boton" id="boton" value="Calcular"/>
  </form>
  <br>

  <?php
    if (isset($_POST["boton"])) {
      $num1 = $_POST["divisor"];
      $num2 = $_POST["dividendo"];

      echo "El resultado es: " . ($num2 / $num1);
    }
  ?>
</body>
</html>
```

Como se puede ver a continuación el formulario es muy sencillo y aunque si funciona puede tener errores en tiempo de ejecución pues no hay nada que evite que el usuario coloque texto e incluso podría hacer una inyección SQL si hubiera una base de datos.

Ejemplo funcional.

Divisiones

Elige 2 numeros para dividirlos y tener la respuesta

Elige un divisor Elige un dividendo

El resultado es: 10

Ejemplo con error.

Divisiones

Elige 2 numeros para dividirlos y tener la respuesta

Elige un divisor Elige un dividendo

Fatal error: Uncaught TypeError: Unsupported operand types: string / string in C:\xampp\htdocs\Excepciones\index.php:23 Stack trace: #0 {main} thrown in C:\xampp\htdocs\Excepciones\index.php on line 23

Con un simple cambio al agregar **type = "number"** hacemos que el campo solo acepte números, pero aún es posible la división entre 0 y ese es otro error.

Divisiones

Elige 2 numeros para dividirlos y tener la respuesta

Elige un divisor Elige un dividendo

El resultado es: 30

Error.

Divisiones

Elige 2 numeros para dividirlos y tener la respuesta

Elige un divisor Elige un dividendo

Fatal error: Uncaught DivisionByZeroError: Division by zero in C:\xampp\htdocs\Excepciones\index.php:23 Stack trace: #0 {main} thrown in C:\xampp\htdocs\Excepciones\index.php on line 23

Al agregar el **try catch** nos libramos de ese problema.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Excepciones</title>
</head>
<body>
    <h3>Divisiones</h3>
    <h4>Elige 2 numeros para dividirlos y tener la respuesta</h4>
    <form method="post" name="division" id="division">
        <label for="divisor">Elige un divisor</label>
        <input id="divisor" name="divisor" type="number"/> <!--Con type number-->
        <label for="dividendo">Elige un dividendo</label>
        <input id="dividendo" name="dividendo" type="number"/> <!--Con type number-->
        <input type="submit" name="boton" id="boton" value="Calcular"/>
    </form>
    <br>

    <?php
        if (isset($_POST["boton"])) {
            try {
                $num1 = $_POST["divisor"];
                $num2 = $_POST["dividendo"];

                echo "El resultado es: " . ($num2 / $num1);
            } catch (\Throwable $th) {
                echo "No es posible la division entre 0";
            }
        }
    ?>
</body>
</html>

```

Divisiones

Elige 2 numeros para dividirlos y tener la respuesta

Elige un divisor Elige un dividendo

No es posible la division entre 0

Este ejemplo también es replicable en C# con Windows forms, aunque con sus respectivos componentes de Windows forms, pero la idea de controlar lo que el usuario puede agregar es muy útil para tener control sobre nuestro software.

Bibliografía

- *TODOS los tipos de INPUT para FORMULARIOS en HTML.* (s/f). Kikopalomares.com. Recuperado el 28 de agosto de 2023, de <https://kikopalomares.com/clases/todos-los-tipos-de-input-para-formularios-en-html>
- *PHP: Excepciones - Manual.* (s/f). Php.net. Recuperado el 28 de agosto de 2023, de <https://www.php.net/manual/es/language.exceptions.php>