PRÁCTICA 2: PHP Y TYPESCRIPT

Programación Orientada a Objetos

Eduardo Ulibarri Toledo - UO251436

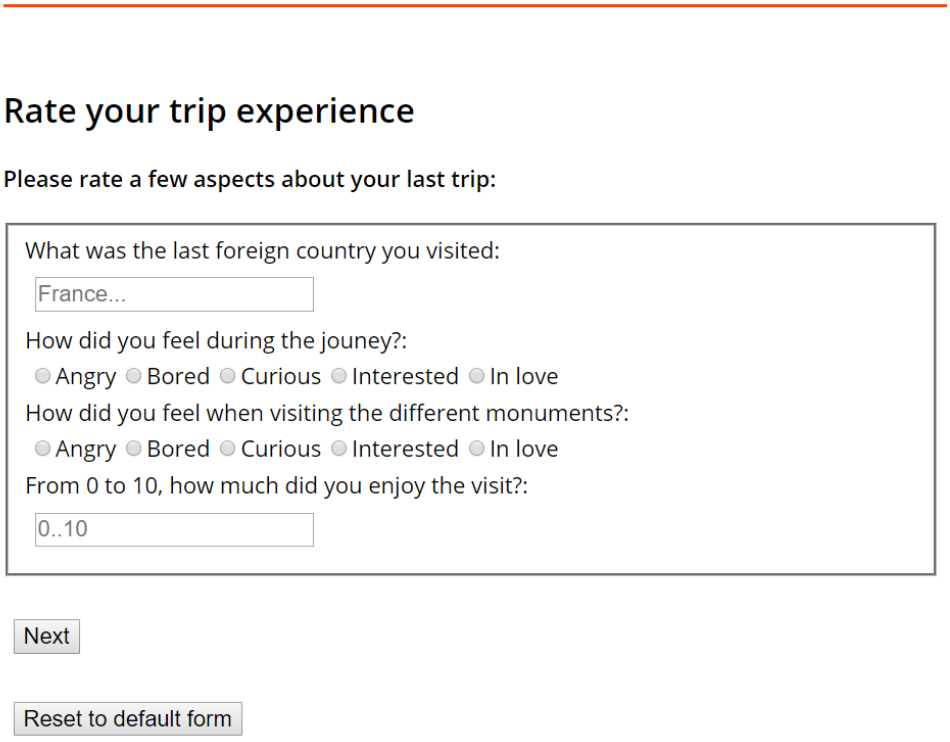
Contenido

1. Descripción general de la aplicación

La aplicación web desarrollada cumple la simple funcionalidad de permitir a los usuarios rellenar los campos de un formulario, ver si los datos que han introducido son válidos y, finalmente, visualizar gráficamente los datos que han introducido (sólo si son válidos).

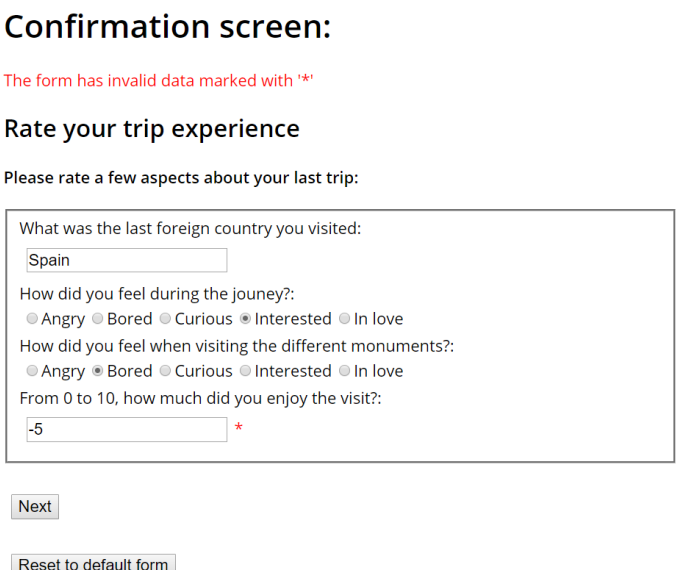
La aplicación consta de 3 pantallas para cumplir con su papel. El código que hace posible su funcionamiento se halla en la carpeta "código". La funcionalidad de cada pantalla es la siguiente:

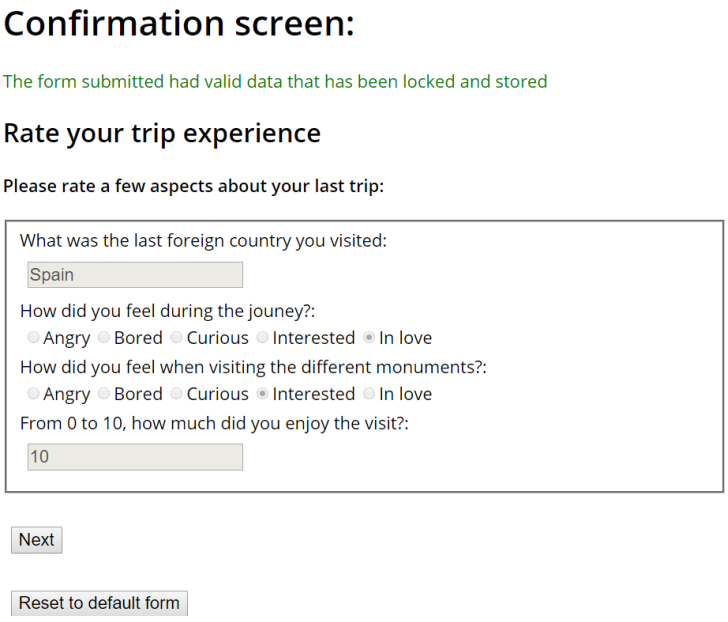
- *Pantalla 1*: Se accede a ella a través del fichero "screen1.php". Esta pantalla muestra al usuario todos los módulos de la aplicación (evaluador, depurador, visualizador...) si bien la parte más relevante es el formulario que muestra al usuario. Si el usuario no ha introducido su propio formulario mediante el módulo evaluador, se le mostrará un formulario preguntándole por su último viaje y su experiencia en el mismo. Cuando se le manda un parámetro "reset" en una petición GET vuelve a mostrar el formulario original fuese cual fuese el anterior.



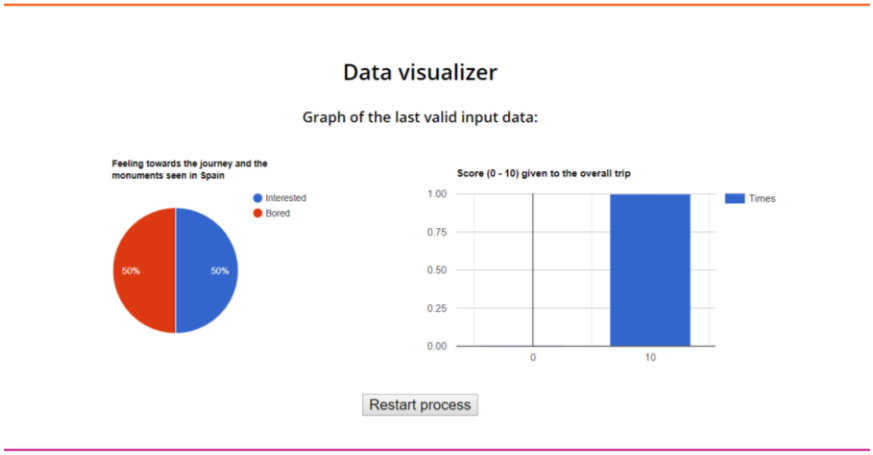
- *Pantalla 2*: Se accede a ella a través del fichero "screen2.php". Tiene la función de hacer de paso intermedio entre la pantalla 1 (introducción de datos) y la pantalla 3 (visualización de datos). En la pantalla 2 se validan los datos introducidos en la pantalla 1. Si hay errores, se le indica al usuario cuales son de forma que si vuelve a subir el cuestionario se reinicia la validación con los nuevos datos válidos que

introduzca. Una vez los datos son válidos, se le muestran al usuario y no se permite su edición a no ser que se vuelva atrás. El botón de continuar le llevará a la pantalla 3.



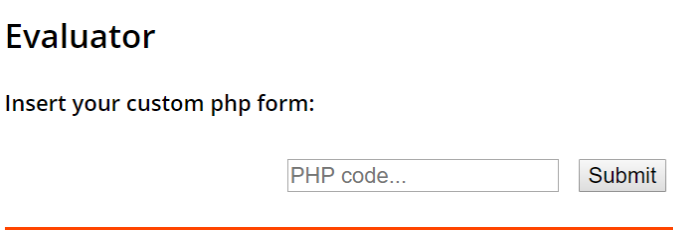


- *Pantalla 3*: Se accede a ella a través del fichero “screen3.php”. Muestra en mayor tamaño dos gráficos relacionados con las respuestas del usuario: cómo de interesante le fue el viaje en general y qué puntuación numérica le dio.



Tal y como se propuso en clase, la aplicación consta de 4 módulo principales:

- *Módulo Evaluador*: Situado en la parte superior de todas las pantallas, ejecuta el código PHP que se le introduzca. Si el código corre sin errores y devuelve un formulario, la aplicación pasará a mostrar el formulario personalizado que haya creado ese código PHP. Su código PHP está en la carpeta “evaluator”.



- *Módulo Formulario*: En las pantallas 1 y 2 (recogida y validación) siempre estará presente a la izquierda de la pantalla una representación gráfica del formulario que PHP está manejando internamente. Su código PHP está en la carpeta “form”.
- *Módulo depurador*: En todas las pantallas aparecerá en la parte inferior una tabla con el nombre, valor y tipo de todas las variables dentro de $_SESSION, $_GET y $_POST de la aplicación PHP. Su código PHP está en la carpeta “debugger”.

Session debugger

2 variable(s) in session.

name	value	type
defaultForm	Form with title: Rate your trip experience	object: Form
validForm	Form with title: Rate your trip experience	object: Form

0 GET request variable(s) in session.

4 POST request variable(s) in session.

name	value	type
place	Spain	string
journey	Curious	string
monuments	Interested	string
score	8	string

- *Módulo visualizador*: Todas las pantallas tiene un espacio reservado para este módulo (la pantalla 3 le dedica más espacio pues en ella no sale el formulario). Este módulo generará y mostrará gráficas de los datos del último formulario que el usuario haya rellenado con datos válidos. Aún si iniciamos un nuevo formulario, hasta que los nuevos datos sean válidos, veremos las gráficas de los viejos. Su código está en la carpeta "js", dentro del fichero "evaluator.ts".
  Sólo se generan gráficas del formulario cuando el formulario no ha sido modificado mediante el módulo evaluador sino que es el formulario por defecto de la aplicación:

Data visualizer

Sorry but graphs are only generated for the default form and not user custom forms.

Restart process

2. Parte PHP

Internamente la aplicación se basa en las características de las sesiones de PHP para funcionar. Está definida en PHP una clase formulario capaz de albergar varios objetos de la clase pregunta. La clase pregunta cuenta con una subclase por cada tipo de pregunta del formulario que representa (pregunta textual, numérica, selección con radio buttons, etc.). Cada pregunta es capaz de traducir su contenido a un campo de un formulario HTML, siendo así el programa capaz de representar el formulario al usuario.

En la sesión de PHP se guarda el objeto formulario al completo, facilitando así su traslado de una pantalla a otra. Se hace uso de **$_POST** para obtener las variables introducidas por el usuario en los formularios y así validar esta información. Se hace uso de **$_GET** para comprobar la existencia de un parámetro específico que reinicie el proceso de formulari.

También se emplea PHP para comprobar condiciones al cargar la página y así mostrar al usuario mensajes informativos en función de lo ocurrido.

El módulo evaluador se basa en la función **eval** de PHP: cuando un usuario crea su propio formulario este nuevo formulario se guarda en sesión y se muestra con prioridad respecto al formulario por defecto. El módulo depurador aprovecha el acceso a la variable **$_SESSION** y el módulo visualizador está contenido en un script de JavaScript.

3. Parte TypeScript

El modulo visualizador está plenamente programado en TypeScript y transpilado a JavaScript dentro de la carpeta "js" del código.

Este módulo primero comprueba si los datos del formulario ya han sido validados o no en función de si PHP dibujó un elemento en pantalla o no. Gracias a PHP el script también tiene acceso al formulario codificado en formato JSON para poder recorrerlo a posteriori.

Si hay acceso a los datos de un formulario validado, se invoca a Google Charts. Mientras tanto, se recorren las respuestas del formulario extrayendo de ellas los datos a representar que finalmente se pasan a la librería de Google. El grueso de la tarea se hace en la función **drawCharts()**.

4. Despliegue

Para el despliegue se recomienda iniciar un servidor PHP en el puerto deseado (*php -S localhost:<<port>>*). Se recomienda iniciar el servidor dentro de la carpeta "código" para más comodidad.

5. Datos adicionales

- Las cadenas de texto que se pueden introducir en el evaluador para probar su funcionalidad pueden encontrarse en la carpeta "código", en el documento "ejemplos_evaluador.txt".
- Se ha seguido la filosofía gradual typing para especificar los tipos con los que se lidiaba en la medida de lo posible (PHP: parámetros y retorno de las funciones principalmente, TypeScript: en menor medida para las variables declaradas a lo largo del script (que no dependiesen de Google Charts) y retorno de las funciones).

- Solo hay un diagrama de clases de alto nivel para la parte PHP puesto que el uso de TypeScript no las ha requerido (se ha usado más como un lenguaje de scripting que automatice la tarea de dibujar gráficos). El diagrama se halla en el fichero “diagrama_clases.png”.
- Todos los tests de hallan en la carpeta “test” dentro de “código”. Se hacen pruebas unitarias del correcto funcionamiento de las clases que manejan los formularios haciendo uso de la utilidad **phpunit** instalada con “composer”.