Universitat de Barcelona

Computació Orientada al Web

Práctica 3

Arthur Font Gouveia - 20222613

Barcelona

2022

Índice

1. Introducción

El objetivo de esta práctica fue añadir las funcionalidades de eventos y sus handlers usando JavaScript y Prototype, separar el código de forma no obstructiva, implementar interacciones con los elementos del DOM, y efectos usando Scriptaculous, y por fin modificar la página usando la tecnología Ajax, basándose en la página web desarrollada en la práctica anterior. Las nuevas funcionalidades fueron codificadas manualmente usando un editor de texto.

2. Apartado 1. Implementar las funcionalidades al cliente utilizando Java Script, Prototype y Scriptaculous

En este apartado, he implementado las funcionalidades al cliente, simulando diversas formas para realizar una búsqueda de hoteles. Para eso, ha sido necesario la creación de un fichero (**funcionalidad.js**).

El fichero funcionalidad.js espera a que el DOM se cargue para ejecutarse mediante la función **window.onload** y relaciona el evento del envío del formulario mediante la función .**observe** y la función formSubmit(), creada para hacer el control de los datos introducidos por parte del cliente. En caso de que algún dato no pase en el control, el envío del formulario se para mediante la llamada a la función **event.preventDefault()**. A continuación están ilustradas las funcionalidades del código más relevantes:

```
window.onload = pageLoad; // Wait until body charges completely
function pageLoad() {
   var form = $("search_form"); // $ equal to document.getElementById()
   form.observe = ("click", formSubmit); // Event handler - Protoype observe
   …
}
function formSubmit(event) {
   if ($F("destination").trim().length == 0 || $F("guests") == 0 ||
      $F("checkin").length == 0 || $F("checkout").length == 0) {
      alert("Please fill out all fields!"); // show error message
      event.preventDefault(); // stop form submission
      return false;
   }
   ….
}
```

El fichero funcionalidad.js también es responsable por realizar interacciones y efectos con elementos del DOM mediante las funciones **grow()** y **shake()**, controladas por el atributo *duration* y responsables por hacer un elemento surgir y sacudir, respectivamente . Además, fue implementada la funcionalidad de autocompletar (dada una lista de opciones) durante la búsqueda de la ciudad de destino mediante la función **Autocompleter.Local()**. A continuación están ilustradas las funcionalidades del código más relevantes:

```
new Autocompleter.Local("destination","destlist",
      ["Barcelona", "Paris", "Porto", "São Paulo", "Brisbane", "Roma", "Rio de Janeiro"],
      {}
);
// Scriptaculous
$("center_header").grow({
    duration: 1.0,
});
$("env_header").shake({
    duration: 1.0,
});
```

3. Apartado 2. Modificar el sitio web utilizando la tecnología Ajax

En este apartado se ha implementado la funcionalidad autocompletar para que aparezca automáticamente la ciudad del hotel tecleando las iniciales, pero esta vez mediante un **XMLHttpRequest()** que recibirá los datos de la base de datos del servidor.

Para eso, he creado el fichero **search_hint.js**, que detecta la entrada del teclado mediante la función **.observe()**, define la función a ser ejecutada cuando reciba la respuesta de la requisición mediante **xmlhttp.onreadystatechange()** y hace las requisiciones a cada cambio detectado mediante las funciones **xmlhttp.open()** y **xmlhttp.send()** y la función request de esta funcionalidad y está ilustrado abajo:

```
document.observe("dom:loaded", function() {
   $("destination").observe("keyup", function() {
      if ($F("destination").length == 0) {
         $("destination").innerHTML = "";
         return;
      } else {
         var xmlhttp = new XMLHttpRequest();
         xmlhttp.onreadystatechange = function() {
            if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
```

```
                $("destination").innerHTML += this.responseText;
            }
        };
        xmlhttp.open("POST", "gethint.php?q=" + $F("destination"), true);
        xmlhttp.send();
    }
  });
});
```

Además, se han implementado las funcionalidades de enseñar en la página inicial la lista de países y también la lista de las zonas (playa, interior o montaña) de hoteles disponibles (en la base de datos) mediante la función **AjaxRequest()**.

En el caso de la lista de países, al pulsar sobre un ítem se busca información sobre el país en el Google. Ya en el caso de la lista de las zonas, al pulsar sobre un ítem se carga los resultados de la búsqueda por zona en la misma página, para eso ha sido importante utilizar A continuación están ilustradas las funcionalidades del código más relevantes

```
// Request to get countries destinations
  new Ajax.Request("getcountries.php", {
    method: "GET",
    onSuccess: countriesSuccessfulResponse,
    onFailure: failedResponse
  });

// Request to get destinations by environment
  new Ajax.Request("getbyenv.php", {
    method: "GET",
    parameters: {env: event.target.innerHTML} ,
    onSuccess: getByEnvSuccessfulResponse,
    onFailure: failedResponse
  });

function getByEnvSuccessfulResponse(ajax) {
  if (ajax.status === 200) {
    var center_body = document.getElementById("center_body");
    …
    center_body.insertAdjacentHTML("beforeend", ajax.responseText);
  }
}
```

Importante resaltar el papel de la función **insertAdjacentHTML()**, que permite añadir la respuesta del servidor a un sitio específico del DOM. En este caso los resultados de la búsqueda fueron insertados justo después de un header central.