



UNIVERSITAT DE
BARCELONA

Facultad de Matemáticas e Informática
Grado en Ingeniería Informática

Computación Orientada a la Web

Entrega Sección 4-5

Alumno: William D. Sotto

Memoria de la Práctica – Sección 4–5

Introducción

El proyecto “Reservas de Hoteles” que desarrollé inicialmente, seguía una estructura clásica cliente/servidor basada en HTML, CSS y PHP. En la primera fase se implementaron funcionalidades del lado del cliente con JavaScript, aprovechando las librerías Prototype y Scriptaculous para agregar interactividad y efectos visuales (como animaciones de aparición). Posteriormente, en el segundo apartado decidí mejorar la experiencia del usuario implementando comunicación asíncrona con Ajax. Esto permitió, por ejemplo, filtrar el listado de hoteles en tiempo real y evitar la recarga completa de la página en operaciones clave.

Apartado 1 – Funcionalidades con JavaScript, Prototype y Scriptaculous

Implementación en el Código Inicial:

Efectos Visuales y Animaciones:

En la página principal (index.php) se integró Prototype y Scriptaculous para añadir efectos de “fade in” a los elementos de la interfaz, en particular a las tarjetas de hoteles (clase `hoteles-card`) y a la imagen de fondo (clase `bg-hotel`).

```
document.observe('dom:loaded', function() {
    $$('.hoteles-card').each(function(cardElement) {
        cardElement.hide();
        new Effect.Appear(cardElement, { duration: 2.0 });
    });
    $$('.bg-hotel').each(function(headerElement) {
        headerElement.hide();
        new Effect.Appear(headerElement, { duration: 2.0 });
    });
});
```

- Con este código, se consigue que, una vez cargado el DOM, los elementos se oculten y posteriormente aparezcan suavemente en pantalla, cumpliendo con el objetivo de una mejor experiencia.

Resumen del Apartado 1:

Usé animaciones y validaciones en el cliente mediante el uso de Prototype y Scriptaculous, cumpliendo con los lineamientos del enunciado. Esta parte se centró en mejorar la experiencia visual y la interactividad sin alterar la estructura principal del sitio.

Apartado 2 – Implementación de Ajax para Interacción Dinámica

Cambios Realizados en el Código:

Para el segundo apartado se realizaron las siguientes modificaciones:

1. Filtrado Dinámico de Hoteles:

Se añadió un campo de búsqueda en el index.php que permite al usuario filtrar los hoteles por nombre, ciudad o zona.

Implementación con Ajax:

Creé el archivo **filtrar_hoteles.php**, el cual recibe un parámetro GET (**q**) y consulta la base de datos para devolver el HTML de las tarjetas de hoteles filtradas.

1. Si el campo está vacío, se muestran todos los hoteles.
2. Si contiene un valor, se aplica una búsqueda con comodines (LIKE) en los campos relevantes.

El código JavaScript en index.php intercepta el evento **keyup** en el input de búsqueda y realiza la llamada Ajax mediante Prototype:

```
function cargarHoteles(query) {
  new Ajax.Request('filtrar_hoteles.php', {
    method: 'get',
    parameters: { q: query },
    onSuccess: function(response) {
      $('hotelesContainer').update(response.responseText);
    },
    onFailure: function() {
      $('hotelesContainer').update('<div class="alert alert-danger">Error al cargar los hoteles.</div>');
    }
  });
}
```

```
$('#hotelSearch').observe('keyup', function(event) {  
    var query = event.element().value;  
    cargarHoteles(query);  
});
```

3. Esta implementación permite que el listado de hoteles se actualice en tiempo real conforme el usuario escribe, mejorando significativamente la usabilidad y la experiencia.

2. Intercepción del Evento POST en Formularios:

En el formulario de registro (sign-up.php), se decidió cambiar la forma en que se envían los datos.

Antes:

El formulario se enviaba de manera tradicional, provocando la recarga completa de la página y mostrando mensajes de error o éxito tras la respuesta del servidor.

Ahora:

Se intercepta el envío del formulario mediante un evento `submit` y se utiliza una petición Ajax (usando Prototype) para enviar los datos al servidor de forma asíncrona. Esto evita la recarga y permite actualizar la interfaz con el resultado (mensaje de registro exitoso o error) de forma inmediata.

```
$('#registrationForm').observe('submit', function(event) {  
    event.stop();  
    new Ajax.Request('create_user.php', {  
        method: 'post',  
        parameters: Form.serialize('registrationForm'),  
        onSuccess: function(response) {  
            $('#message').update(response.responseText);  
        },  
        onFailure: function() {  
            $('#message').update('Error en el registro.');        }  
    });  
});
```

Esta modificación no solo mejora la interactividad, sino que también permite una mayor modularidad en el manejo de errores y la validación en el lado del servidor.

Resumen del Apartado 2:

Se adaptó el sitio web para incorporar tecnología Ajax, logrando que:

- El listado de hoteles se filtre de manera dinámica en función de la entrada del usuario.
- Se intercepte el envío de formularios (el registro de usuario) para procesar la información de forma asíncrona.

Estas mejoras permiten una experiencia de usuario mucho más fluida, eliminando recargas completas de página y ofreciendo una retroalimentación inmediata en la interfaz.

Conclusión

El desarrollo del proyecto se estructuró en dos grandes fases:

- **Apartado 1:** Se implementaron funcionalidades interactivas y efectos visuales utilizando Prototype y Scriptaculous. Se mejoró la presentación del contenido sin alterar la arquitectura del sitio.
- **Apartado 2:** Se introdujo Ajax para modernizar la comunicación cliente/servidor. Con este cambio se logró, por ejemplo, que el listado de hoteles se actualice dinámicamente en función de la búsqueda del usuario y que el formulario de registro se procese sin recargar la página.

Ambos apartados se integraron de manera coherente en el sitio, logrando una experiencia mejorada, y cumpliendo con los requerimientos del enunciado.