

# Laboratorio III

## AJAX

### Clase 3

Maximiliano Neiner

# Temas a Tratar

- AJAX

# Temas a Tratar

- AJAX

# AJAX (**A**synchronous **J**avaScript **A**nd **X**ML)

- No es un lenguaje de programación sino un conjunto de tecnologías que nos permiten hacer páginas de internet más interactivas.
- AJAX permite que las páginas sean capaces de actualizar partes en segundo plano, sin volver a cargar toda la página.
- AJAX se hizo popular en 2005 por Google, con Google Suggest.

# Tecnologías

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

# Ventajas

- Utiliza tecnologías ya existentes.
- Soportada por la mayoría de los navegadores modernos.
- Interactividad. El usuario no tiene que esperar hasta que lleguen los datos del servidor.
- Portabilidad (no requiere plug-ins como Flash y Applet de Java)
- Mayor velocidad, esto es debido a que no hay que retornar toda la página nuevamente.
- La página se asemeja a una aplicación de escritorio.

# Desventajas

- Problemas con navegadores antiguos que no implementan esta tecnología.
- No funciona si el usuario tiene desactivado el JavaScript en su navegador.
- Requiere programadores que conozcan todas las tecnologías que intervienen en AJAX.

# XMLHttpRequest (1/2)

- La piedra angular de AJAX es el objeto XMLHttpRequest y es utilizado para intercambiar datos con el servidor en segundo plano.
- Para enviar una petición al servidor, se utilizan los métodos **open()** y **send()**.

```
xhttp.open("Método", "Url", Async);
```

- Método: Especifica el tipo de pedido (GET/POST).
- Url: Indica la ubicación del archivo en el servidor.
- Async: true (Asincrónico); false (Sincrónico).



# XMLHttpRequest (2/2)

```
xhttp.send(); //Si es GET  
xhttp.send("string"); //Si es POST
```

- Envía la petición al servidor.

```
var xhttp = new XMLHttpRequest();  
xhttp.open("GET", "ajax_test.php", true);  
xhttp.send();
```

The background is a solid dark blue color. Overlaid on this are several large, rounded rectangular shapes in a lighter shade of blue. These shapes are arranged in a way that they appear to be floating or layered, with some overlapping others. The overall effect is a modern, minimalist design.

Demo

# onreadystatechange (1/2)

- El evento **onreadystatechange** se dispara cada vez que cambia **readyState**.
- La propiedad `readyState` mantiene el estado de `XMLHttpRequest`.
- Existen tres propiedades importantes del objeto `XMLHttpRequest`:

# onreadystatechange (2/2)

- **onreadystatechange:** Almacena una función (o el nombre de una función) que se invoca de forma automática cada vez que cambia de propiedad readyState.
- **readyState:** Mantiene el estado del XMLHttpRequest:
  - 0 : Solicitud no inicializado
  - 1 : Conexión de servidor establecido
  - 2 : Solicitud recibida
  - 3 : Solicitud de procesamiento
  - 4 : Solicitud terminada y respuesta lista
- **status:** 200 → OK; 404 → No encontrado.

Demo



# Ejercitación