# **Ciclo Formativo de Grado Superior Desarrollo de Aplicaciones Web – Curso 2023 / 2024**

## **Proyecto Desarrollo de Aplicaciones Web**

## **Fase 5: ENTREGA DE PROYECTO** Conectar la aplicación con el servidor Node y chequear su funcionamiento

Forma

Descripción generada automáticamente con confianza baja

## Analiza las ventajas y desventajas de de utilizar mecanismos de comunicación asíncrona entre cliente y servidor Web

La **comunicación asíncrona** entre el cliente y el servidor web tiene sus propias **características**, **ventajas** y **desventajas**. Vamos a explorarlas:

1. **Características de la comunicación asíncrona**:
   * **Flexibilidad temporal**: En la comunicación asíncrona, no se requiere una respuesta inmediata. El emisor y el receptor no están sincronizados en tiempo real.
   * **Conservación de la información**: Una vez enviada, la información permanece y puede ser archivada o revisada por el emisor y los receptores.
   * **Mensajes más elaborados**: Los mensajes asincrónicos suelen estar más detallados que los sincrónicos. Permiten el envío de informes, archivos y datos diversos.
   * **Trabajo compartido**: La comunicación asíncrona permite que varias personas trabajen en un mismo proyecto o documento durante un período prolongado, incorporando información de forma gradual.
   * **Respaldos y archivos adjuntos**: Los mensajes asincrónicos pueden ir acompañados de archivos adjuntos como documentos, gráficos o vídeos.
2. **Ventajas de la comunicación asíncrona**:
   * **Sencillez**: No requiere comunicación bidireccional constante.
   * **Flexibilidad para actuar**: Las personas pueden responder en sus propios términos y horarios.
   * **Disponibilidad constante**: Funciona en cualquier momento y lugar.
   * **Registro de comunicación compartida**: Al no ser instantánea, se puede mantener un registro de la comunicación.
   * **Promueve la proactividad**: Las personas pueden tomar la iniciativa en lugar de reaccionar constantemente.
3. **Desventajas de la comunicación asíncrona**:
   * **Menos eficiente**: Comparada con la comunicación sincrónica, la asincrónica puede ser menos eficiente debido a la brecha de tiempo.
   * **Mayor sobrecarga**: Requiere más recursos y planificación para gestionar la comunicación en diferentes momentos.

### Explica en qué consiste el mecanismo de la comunicación asíncrona.

La **comunicación asíncrona** es un mecanismo de intercambio de información entre dos o más partes que no requiere una sincronización inmediata. Aquí están los aspectos clave de este mecanismo:

1. **Retraso temporal**: A diferencia de la comunicación síncrona (como una llamada telefónica), en la comunicación asíncrona no se espera una respuesta inmediata. Los mensajes pueden enviarse y recibirse en momentos diferentes.
2. **Características**:
   * **Flexibilidad**: Los participantes pueden responder cuando les resulte conveniente, sin necesidad de estar disponibles al mismo tiempo.
   * **Persistencia**: Los mensajes asincrónicos permanecen en el sistema hasta que se procesan o se eliminan. Esto permite revisar y archivar la información.
   * **Ejemplos comunes**: Correo electrónico, mensajes de texto, foros en línea y sistemas de tickets de soporte.
3. **Ejemplos de uso**:
   * **Correo electrónico**: Los correos electrónicos se envían y reciben en momentos diferentes, lo que permite a los usuarios gestionar su tiempo de manera eficiente.
   * **Foros en línea**: Los usuarios publican preguntas o comentarios, y otros responden cuando puedan. No es necesario estar en línea simultáneamente.
   * **Sistemas de tickets de soporte**: Los clientes envían solicitudes de ayuda, y los agentes responden cuando estén disponibles.
4. **Ventajas**:
   * **Flexibilidad temporal**: Los participantes no están atados a horarios específicos.
   * **Registro y seguimiento**: Los mensajes quedan registrados, lo que facilita el seguimiento y la resolución de problemas.
   * **Menos presión**: No hay urgencia para responder de inmediato.
5. **Desventajas**:
   * **Menos inmediato**: Puede llevar más tiempo obtener respuestas.
   * **Menos adecuado para conversaciones rápidas**: No es ideal para discusiones en tiempo real.

En resumen, la comunicación asíncrona es valiosa cuando se necesita flexibilidad y persistencia, pero puede no ser la mejor opción para situaciones urgentes o conversaciones rápidas.

## Analiza las propiedades y métodos de los objetos implicados en la comunicación asíncrona. (XMLHTTPrequiest)

La **comunicación asíncrona** entre el cliente y el servidor web se basa en el uso del objeto **XMLHttpRequest** (XHR). Este objeto es fundamental para realizar solicitudes y recibir respuestas sin recargar toda la página web. Veamos sus propiedades y métodos:

1. **Propiedades del objeto XMLHttpRequest**:
   * **onload**: Define una función que se ejecutará cuando la respuesta esté lista. Es útil para procesar los datos recibidos.
   * **onreadystatechange**: Define una función que se llama cuando cambia el estado del objeto. Los valores posibles del estado son:
     + **0**: No inicializado.
     + **1**: Conexión con el servidor establecida.
     + **2**: Solicitud recibida.
     + **3**: Procesando la solicitud.
     + **4**: Solicitud finalizada y respuesta lista.
   * **readyState**: Indica el estado actual del objeto XMLHttpRequest (valores del 0 al 4).
   * **responseText**: Devuelve los datos de la respuesta como una cadena de texto.
   * **responseXML**: Devuelve los datos de la respuesta como datos XML.
   * **status**: Devuelve el código de estado de la solicitud (por ejemplo, 200 para “OK” o 403 para “Prohibido”).
2. **Métodos del objeto XMLHttpRequest**:
   * **new XMLHttpRequest()**: Crea un nuevo objeto XMLHttpRequest.
   * **abort()**: Cancela la solicitud actual.
   * **getAllResponseHeaders()**: Devuelve información sobre las cabeceras de la respuesta.
   * **getResponseHeader(headerName)**: Devuelve información específica de una cabecera.
   * **open(method, url, async, user, password)**:
     + **method**: Especifica el método de la solicitud (GET o POST).
     + **url**: La ubicación del archivo o recurso.
     + **async**: true para asincrónico, false para sincrónico.
     + **user** (opcional): Nombre de usuario.
     + **password** (opcional): Contraseña.
   * **send(data)**:
     + Para solicitudes GET: Envía la solicitud al servidor.
     + Para solicitudes POST: Envía datos al servidor.
   * **setRequestHeader(header, value)**: Agrega una pareja etiqueta/valor a las cabeceras de la solicitud.

En resumen, el objeto **XMLHttpRequest** permite la comunicación asíncrona entre el cliente y el servidor web, facilitando la actualización de partes específicas de una página sin recargarla completamente.

## Clasifica y analiza las librerías actuales que facilitan la incorporación de las tecnologías de actualización dinámica a la programación de páginas Web.

La incorporación de tecnologías de **actualización dinámica** en la programación de páginas web se beneficia enormemente de las **bibliotecas y frameworks de JavaScript**. Estas herramientas permiten a los desarrolladores crear contenido interactivo, mejorar la experiencia del usuario y simplificar tareas de desarrollo. A continuación, clasificaré y analizaré algunas de las bibliotecas más populares:

1. **jQuery**:
   * **Descripción**: jQuery es una de las bibliotecas de JavaScript más conocidas. Se destaca por su simplicidad y versatilidad.
   * **Funcionalidades**:
     + Simplifica la manipulación del DOM.
     + Facilita el manejo de eventos.
     + Permite llamadas AJAX.
     + Ideal para crear elementos interactivos y animaciones.
   * **Uso común:** Ampliamente utilizado en proyectos web para mejorar la interacción del usuario1
2. **React**:
   * **Descripción**: React es un framework JavaScript de código abierto mantenido por Facebook.
   * **Funcionalidades**:
     + Facilita la creación de interfaces de usuario dinámicas.
     + Componentización eficiente.
     + Virtual DOM para optimización de actualizaciones.
   * **Uso común:** Ampliamente utilizado en aplicaciones web y móviles para construir interfaces interactivas y escalables.
3. **Vue.js**:
   * **Descripción**: Vue.js es otro framework JavaScript progresivo.
   * **Funcionalidades**:
     + Enfoque en la capa de vista.
     + Fácil integración con proyectos existentes.
     + Reutilización de componentes.
   * **Uso común:** Ideal para crear contenido web dinámico y aplicaciones de una sola página (SPA).
4. **Angular**:
   * **Descripción**: Angular es un framework completo desarrollado por Google.
   * **Funcionalidades**:
     + Soporte para desarrollo de aplicaciones empresariales.
     + Inyección de dependencias.
     + Enrutamiento y manejo de estados.
   * **Uso común:** Ampliamente utilizado en aplicaciones empresariales y proyectos complejos.
5. **GSAP (GreenSock Animation Platform)**:
   * **Descripción**: GSAP es una biblioteca para animaciones web.
   * **Funcionalidades**:
     + Potente motor de animación.
     + Soporte para animaciones complejas y personalizadas.
     + Amplia comunidad y documentación.
   * **Uso común:** Ideal para crear animaciones fluidas y atractivas en sitios web

En resumen, estas bibliotecas y frameworks permiten a los desarrolladores incorporar actualizaciones dinámicas en sus proyectos web, mejorando la experiencia del usuario y simplificando el desarrollo.