

# **Desarrollo Web Cliente - Ud 6**

**Aprendizaje 3: diferencias entre navegadores**

**Guía para Adaptar Aplicaciones JavaScript a Diversos Navegadores**

**Introducción**

En el desarrollo de aplicaciones web, uno de los retos más importantes es garantizar que el código funcione correctamente en todos los navegadores y versiones posibles. Esto es especialmente relevante cuando se manejan navegadores antiguos como Internet Explorer, que no cumplen completamente con los estándares modernos.

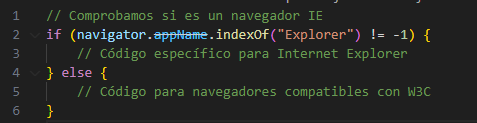
**Pasos para Garantizar la Compatibilidad en JavaScript**

**Identificar los Navegadores Soportados**

* Determinar qué navegadores y versiones deben ser compatibles.
* Herramientas como Google Analytics pueden proporcionar datos sobre los navegadores más utilizados por los usuarios.

**Ejecución Condicional del Código**

En JavaScript, podemos identificar el navegador y ejecutar diferentes bloques de código según sea necesario. Por ejemplo, para comprobar si el navegador es Internet Explorer:

****

Esto permite aplicar soluciones personalizadas para navegadores que no soporten ciertas características.

**Uso de Polyfills**

Los polyfills son scripts que añaden soporte a funcionalidades modernas en navegadores antiguos.

Por ejemplo, si Promise no es compatible, podemos usar un polyfill para asegurar que funcione:

****

**Transpilación con Babel**

* Herramientas como Babel convierten código JavaScript moderno (ES6+) a versiones compatibles con navegadores más antiguos.
* Configurarlo en un proyecto asegura que incluso navegadores como IE11 puedan ejecutar el código.

**Validación y Pruebas**

* Utilizar herramientas como ESLint para detectar errores y advertencias en el código.
* Probar el comportamiento en diferentes navegadores, usando servicios como BrowserStack o Sauce Labs.

**Documentación**

Mantener un registro de los navegadores soportados y las soluciones implementadas para problemas específicos.

**Conclusión**

Adaptar JavaScript a diversos navegadores requiere prever las diferencias de compatibilidad y ajustar el código en consecuencia. Usar técnicas como la detección de navegadores, polyfills, y herramientas como Babel garantiza una experiencia consistente para todos los usuarios.