

# **Desarrollo Web - Cliente- Act- Ud 1**

# **Elección de tecnologías y primeras pruebas**

Una vez superada la primera unidad nos vemos con las suficientes fuerzas para empezar a crear nuestros propios proyectos de programación web en entorno cliente. Charlando con unos compañeros del módulo nos han surgido varias dudas sobre las tecnologías a utilizar.

1. **Modelos de ejecución de código vía web**

**Cliente/Servidor**

El cliente solicita recursos al servidor, y este responde con los datos necesarios, como páginas HTML, imágenes, o archivos JSON.

**Ventajas**

-*Seguridad:* El servidor centralizado permite un control más estricto de los datos.

-*Facilidad de mantenimiento:* Los datos se almacenan en el servidor, lo que facilita la actualización y administración.

**Inconvenientes**

-*Dependencia del servidor:* Si el servidor falla, los clientes no pueden acceder a los recursos.

-*Mayor latencia:* Cada solicitud requiere una respuesta del servidor, lo que puede añadir retardo.

**P2P (Peer-to-Peer)**

Los usuarios pueden actuar tanto como clientes como servidores, compartiendo recursos entre ellos sin necesidad de un servidor centralizado.

**Ventajas**

-*Descentralización:* No hay un único punto de fallo.

-*Escalabilidad:* A medida que más usuarios se unen a la red, los recursos disponibles aumentan.

**Inconvenientes**

-*Seguridad:* Mayor dificultad para controlar el acceso y proteger los datos.

-*Complejidad*: La implementación y el mantenimiento de una red P2P pueden ser más complicados.

**Conclusión**

Creo que usaremos mas el modelo ***cliente/servidor*** que cubre las necesidades de la mayoría de las aplicaciones web, se suele preferir este modelo por su simplicidad y seguridad, especialmente en entornos donde el control de los datos es fundamental.

1. **Comparación de IDEs**

**Sublime Text**

**Ventajas**

-Ligero y rápido.

-Multiplataforma (Windows, macOS, Linux).

-Interfaz sencilla y fácil de personalizar.

-Soporta múltiples lenguajes.

**Inconvenientes**

-La versión completa es de pago.

-Carece de características avanzadas sin la instalación de plugins.

**Visual Studio Code (VSCode)**

**Ventajas**

-Gratuito y de código abierto.

-Gran cantidad de extensiones y plugins disponibles.

-Integración con sistemas de control de versiones como Git.

-Depurador integrado.

-Amplia comunidad de soporte.

**Inconvenientes**

-Consumo de más recursos en comparación con otros editores ligeros.

**Notepad++**

**Ventajas**

-Ligero y rápido.

-Gratuito y de código abierto.

-Fácil de usar, especialmente para scripts y archivos de texto simples.

-Múltiples lenguajes soportados.

**Inconvenientes**

-No es tan potente ni tan personalizable como VSCode o Sublime.

-Funcionalidades limitadas sin plugins.

**Conclusión**

El IDE a elegir dependerá del balance entre simplicidad y funcionalidad. Si quieres un entorno muy completo y versátil, ***Visual Studio Code*** puede ser la mejor opción. Si prefieres un editor más sencillo pero eficiente, Sublime Text o Notepad++ podrían ajustarse mejor.

1. **Comparación de lenguajes de script**

**JavaScript**

**Ventajas**

-Ejecutado en todos los navegadores, sin necesidad de plugins.

-Ideal para el desarrollo en el lado del cliente.

-Gran comunidad y documentación.

-Soporte para programación asíncrona.

**Desventajas**

-Manejo de errores y debugging más complicados en comparación con lenguajes compilados.

-No es el mejor lenguaje para tareas intensivas en CPU.

**Python**

**Ventajas**

-Sintaxis sencilla y fácil de aprender.

-Ideal para el lado del servidor, con frameworks como Django o Flask.

**Desventajas**

-No se ejecuta directamente en el navegador (requiere backend).

**PHP**

**Ventajas**

Muy utilizado en desarrollo web del lado del servidor.

Amplia compatibilidad con bases de datos.

**Desventajas**

Se está quedando atrás en comparación con tecnologías más modernas.

El desarrollo frontend necesita tecnologías adicionales.

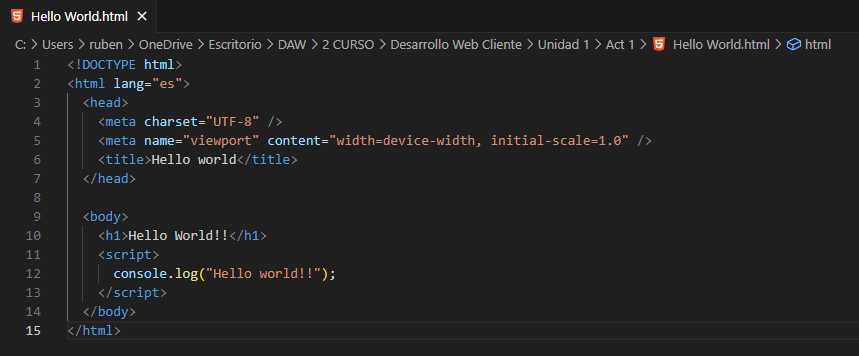
**Conclusión**

Para la mayoría de los proyectos web modernos, ***JavaScript*** es la opción más recomendable. No solo es el lenguaje nativo de los navegadores, sino que también ofrece una curva de aprendizaje razonable, con una gran cantidad de recursos de aprendizaje y soporte de la comunidad. Además, su capacidad para desarrollarse tanto en el cliente como en el servidor lo convierte en una elección versátil y poderosa.

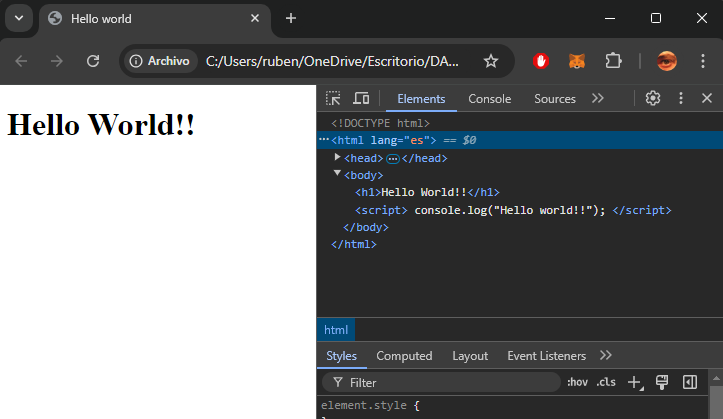
1. **Prueba “Hello World”**

Se pide insertar el código JavaScript en un fichero HTML sencillo y ejecutarlo (en al menos 3 navegadores), para mostrar el resultado. La página debe mostrar “Hello World” en el navegador.

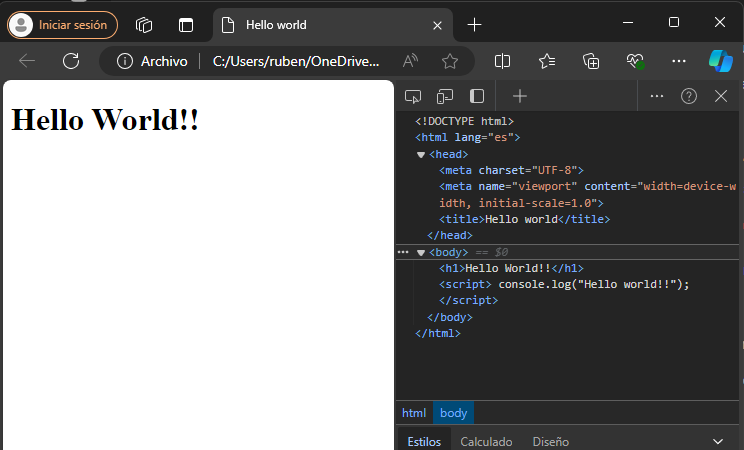
Por último, abriremos la consola de nuestro navegador, para poder ver nuestro código fuente, en al menos 3 navegadores diferentes.

****

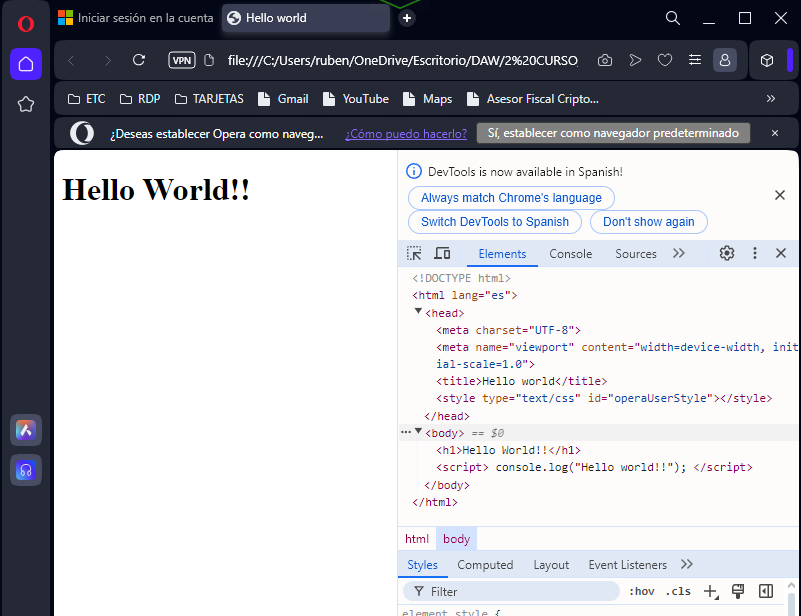
*Codigo HTML de mensaje por consola de “Hello World”*



*Codigo fuente en “Chrome”*



*Codigo fuente en “Microsoft Edge”*



*Codigo fuente en “Opera”*