

TABLA COMPARATIVA

| | TIPO | VENTAJA | DESVENTAJAS | CASOS DE USO |
|------------|--------------------------|---|--|---|
| HTML | Lenguaje de marcado | <ul style="list-style-type: none">• Es el estándar para la web.• Fácil de aprender y usar.• Compatible con todos los navegadores. | <ul style="list-style-type: none">• No es un lenguaje de programación.• Necesita CSS y JavaScript para funcionalidades avanzadas. | Creación de páginas web estáticas, estructura de sitios web y aplicaciones web. |
| CSS | Lenguaje de estiloso | <ul style="list-style-type: none">• Separa contenido y diseño.• Permite personalizar el aspecto de las páginas.• Compatible con animaciones y transiciones. | <ul style="list-style-type: none">• Puede volverse complejo en proyectos grandes sin preprocesadores como SASS o LESS.• Inconsistencias en la renderización entre navegadores. | Diseño y estilización de sitios web, desarrollo de interfaces responsivas. |
| JAVASCRIPT | Lenguaje de programación | <ul style="list-style-type: none">• Lenguaje universal en la web (frontend y backend con Node.js).• Gran comunidad y ecosistema de librerías (React, Angular, Vue). | <ul style="list-style-type: none">• Código ejecutado en el navegador es vulnerable a ataques si no se maneja correctamente.• Puede ser difícil de depurar en proyectos grandes. | Desarrollo frontend interactivo, aplicaciones web dinámicas, backend con Node.js. |

| | TIPO | VENTAJA | DESVENTAJAS | CASOS DE USO |
|---------|---------------------|---|---|---|
| REACT | Biblioteca frontend | <ul style="list-style-type: none">• Reutilización de componentes.• Virtual DOM mejora el rendimiento.• Compatible con React Native para apps móviles. | <ul style="list-style-type: none">• Curva de aprendizaje moderada.• Necesita herramientas adicionales como Webpack y Babel.• Puede ser excesivo para proyectos pequeños. | Aplicaciones de una sola página (SPA), interfaces dinámicas, dashboards. |
| ANGULAR | Framework frontend | <ul style="list-style-type: none">• Arquitectura escalable con TypeScript.• Soporte a largo plazo de Google.• Seguridad integrada y herramientas avanzadas.• Compatible con PWA y SSR. | <ul style="list-style-type: none">• Curva de aprendizaje alta.• Mayor peso en comparación con otras tecnologías frontend.• Menos flexible para proyectos pequeños. | Aplicaciones empresariales, aplicaciones de gran escala con alta seguridad. |
| VUE.JS | Framework frontend | <ul style="list-style-type: none">• Sintaxis sencilla y fácil de aprender.• Ligero y rápido en comparación con React y Angular.• Se puede integrar fácilmente en proyectos existentes. | <ul style="list-style-type: none">• Menos soporte empresarial en comparación con React y Angular.• Comunidad más pequeña.• No es tan eficiente en proyectos muy grandes como Angular. | Aplicaciones interactivas, dashboards, proyectos de tamaño medio. |

| | TIPO | VENTAJA | DESVENTAJAS | CASOS DE USO |
|---------|--------------------------|--|--|---|
| NODE.JS | Entorno backend | <ul style="list-style-type: none">• Basado en JavaScript• No bloqueante, permite alta concurrencia.• Compatible con Express.js para construir APIs rápidamente. | <ul style="list-style-type: none">• No es la mejor opción para tareas que requieren alto uso de CPU.• Callbacks anidados pueden dificultar la lectura del código. | APIs REST, aplicaciones en tiempo real (chats, websockets). |
| DJANGO | Framework backend | <ul style="list-style-type: none">• Basado en Python, fácil de aprender y escalable.• Seguridad integrada (protección contra XSS, CSRF).• Modelo MVC bien estructurado. | <ul style="list-style-type: none">• Puede ser demasiado pesado para proyectos pequeños.• Menos flexible para desarrollos microservicio comparado con Flask o FastAPI. | Aplicaciones seguras y escalables, comercio electrónico, paneles administrativos. |
| MYSQL | Base de datos relacional | <ul style="list-style-type: none">• Muy usado en la industria.• Seguridad y soporte transaccional ACID.• Compatible con múltiples lenguajes de programación.• Soporta grandes volúmenes de datos. | <ul style="list-style-type: none">• Menos flexible con datos no estructurados.• Puede requerir optimización para consultas complejas. | Aplicaciones con estructuras de datos definidas, sistemas bancarios, ecommerce. |