

unidad 1

EJERCICIO 1



20 de septiembre de 2024

Adrián Peña carnero

Contenido

[Ejercicio1 2](#_Toc177749994)

[Ejercicio2 2](#_Toc177749995)

[Ejercicio 3 4](#_Toc177749996)

[Ejercicio 4 6](#_Toc177749997)

[Ejercicio 5 7](#_Toc177749998)

[Ejercicio 6 8](#_Toc177749999)

UD1. Actividades

# Ejercicio1

UD1E1. Revisar la primera web y vuelve a crearla con lo aprendido en lenguaje de marcas de primero. Compara el maquetado primigenio y el maquetado actual.

(Esta adjuntado en la carpeta)

# Ejercicio2

UD1E2. Familiarízate con la web Can I use.

* Comparar navegadores (puntuaciones).

Interfaz de usuario gráfica

Descripción generada automáticamente

* Revisa las diferencias entre Chrome y Firefox ¿qué elementos importantes echas de menos?

Gráfico

Descripción generada automáticamente

**CSS Reflections, Cross-Fade Function, y FileSystem API**:

No están soportados en Firefox, mientras que Chrome sí los soporta

**CSS box-decoration-break:**

Soporte parcial en Chrome y soporte completo en Firefox.

**Speech Recognition API:**

No está soportada en Firefox, pero Chrome tiene soporte parcial.

* Algunas etiquetas tienen prefijos -moz-, -webkit-, ¿qué son?, razona si tiene sentido su empleo incluyendo algún ejemplo.

|  |  |
| --- | --- |
| -moz- | Para Firefox y los demás navegadores de Gecko. |
| -webkit- | Para los navegadores WebKit: Chrome, Safari,etc. |

Los prefijos -moz-, -webkit-, y otros son prefijos específicos del motor de renderizado utilizados en CSS para que dependa de que navegador utilicen, interpreten una cosa u otra.

**Prefijo que utiliza cada navegador:**

* -webkit-: Chrome, Safari
* -moz-: Firefox
* -ms-: Internet Explorer

Texto

Descripción generada automáticamente con confianza media

Tiene sentido ya que lo que pretenden conseguir es que el navegador entienda lo que estamos probando.

# Ejercicio 3

UD1E3. Busca información referente al uso de distintas tecnologías:

* Cuota de mercado por navegadores.

|  |  |  |  |
| --- | --- | --- | --- |
| Navegador | Mundial | España | Cantabria (Estimado) |
| Google Chrome | 65% | 61% | 60% |
| Safari | 18% | 15% | 14% |
| Microsoft Edge | 5% | 7% | 6% |
| Mozilla Firefox | 3% | 5% | 5% |
| Opera | 2% | 2% | 2% |
| Otros | 7% | 10% | 13% |

* Cuota de servidores web.

|  |  |  |  |
| --- | --- | --- | --- |
| Servidor Web | Mundial | España | Cantabria (Estimado) |
| Apache | 30% | 28% | 27% |
| Nginx | 27% | 30% | 31% |
| Microsoft IIS | 5% | 6% | 5% |
| LiteSpeed | 3% | 2% | 2% |
| Otros | 35% | 34% | 35% |

* Cuota de frameworks frontends.

|  |  |  |  |
| --- | --- | --- | --- |
| Framework | Mundial | España | Cantabria (Estimado) |
| React | 40% | 45% | 43% |
| Vue.js | 18% | 20% | 19% |
| Angular | 12% | 10% | 12% |
| Svetle | 4% | 5% | 4% |
| Ember.js | 2% | 3% | 2% |
| Otros | 24% | 17% | 20% |

Navegadores: A nivel mundial, Chrome ha mantenido su dominio. En España y Cantabria, se observa una situación similar, aunque Edge y Firefox tienen una cuota levemente mayor en estas regiones.

Servidores web: Nginx ha ganado terreno, especialmente en España, y Apache ha disminuido su uso comparado con años anteriores. En Cantabria, los patrones son similares.

Frameworks front-end: React sigue siendo el líder en todo el mundo, con una cuota ligeramente mayor en España y Cantabria, mientras que Vue.js ha crecido en popularidad en el ámbito autonómico.

**Fuentes de la información añadida**

**Navegadores:**

StatCounter Global Stats

NetMarketShare

**Servidores web:**

W3Techs

Netcraft

**Frameworks:**

State of JS

GitHub Trends

BuiltWith

Localiza datos a nivel mundial, nacional y autonómico. Incluyendo su evolución en los últimos años.

# Ejercicio 4

UD1E4. Haz una comparativa de distintos frameworks de frontend. React, Vue.js y Angular. Qué tipo de aplicaciones construyen, quien los mantiene, razona cual se ajusta más al desarrollo de aplicaciones de gestión.

He almacenado y hecho en formato de resumen una comparativa más esquemática, así podré elegir el mejor framework que se ajuste a mis necesidades.

|  |  |  |  |
| --- | --- | --- | --- |
| Característica | React | Vue.js | Angular |
| Desarrollador | Meta (Facebook) | Comunidad Open Source | Google |
| Tipo | Librería para UI | Framework progresivo | Framework completo |
| Lenguaje | JavaScript, JSX | JavaScript, HTML templates | TypeScript |
| Arquitectura | Vista (View) | Vista (View) + flexible | Modelo-Vista-Controlador (MVC) |
| Curva de aprendizaje | Media | Baja | Alta |
| Rendimiento | Alto (gracias al Virtual DOM) | Alto (Virtual DOM) | Medio (debido a su complejidad) |
| Popularidad | Muy alta | Alta | Alta |
| Escalabilidad | Buena con librerías adicionales | Buena para apps pequeñas/medianas | Muy buena para apps grandes y complejas |
| Mejor uso | SPA, interfaces interactivas | Dashboards, PWAs, proyectos rápidos | Aplicaciones empresariales a gran escala |
| Ventajas | Flexibilidad, gran ecosistema | |  | | --- | | Fácil de aprender, modular |  |  | | --- | |  | | Estructura robusta, herramientas integradas |
| Desventajas | Necesita librerías adicionales | Menor adopción corporativa | Curva de aprendizaje empinada |

**OPCION RECOMENDADA**

Si es para una empresa de aplicaciones de gestión empresarial que necesita una estructura solida y con capacidad por si aumenta sin duda elegiría Angular.

Para proyectos pequeños o medianos por su simplicidad y rapidez de implementación.

# Ejercicio 5

UD1E5. ¿Cuál es la relación entre Chrome v8, node y electronjs? . Identifica aplicaciones populares que empleen electronjs.

La relación entre Chrome v8, node y electronjs es:

**V8** es el motor de JavaScript que permite ejecutar código JavaScript de manera eficiente. Se usa en Chrome y también en otros entornos como Node.js.

**Node.js**, que está basado en V8, permite ejecutar JavaScript fuera del navegador, en servidores o sistemas locales.

**Electron.js** combina V8 para crear la interfaz de usuario y Node.js para acceder a funciones del sistema operativo. Esto permite crear aplicaciones de escritorio con tecnologías web que funcionan en múltiples plataformas.

Aplicaciones Populares que utilicen electronjs:

* Slack / [Gestión del trabajo mediante IA y herramientas de productividad | Slack](https://slack.com/intl/es-es/)
* Visual Studio Code / [Visual Studio Code - Code Editing. Redefined](https://code.visualstudio.com/)
* Discord / [Discord - Group Chat That’s All Fun & Games](https://discord.com/)
* Trello / [Gestiona los proyectos de tu equipo desde cualquier lugar | Trello](https://trello.com/es)
* Spotify / [Reproductor web de Spotify: música para todo el mundo](https://open.spotify.com/intl-es)

# Ejercicio 6

UD1E6. Localiza las páginas de documentación de MDN y W3C. Familiarízate con su estructura localizando las guías de aprendizaje y los buscadores de etiquetas. ¿Qué página consideras que cuenta con la documentación más actualizada y completa?

Si pichas en este link, vas directo a la pagina web de MDN a el apartado de guías:

[Aprende desarrollo web | MDN (mozilla.org)](https://developer.mozilla.org/es/docs/Learn)

Si pinchas en este link, vas directo a la pagina web de W3Schools donde entras directo a la guía de JavaScript:

[JavaScript Tutorial (w3schools.com)](https://www.w3schools.com/js/default.asp)

Considero después de localizar y familiarizarme con las webs, que MDN es una página mas completa, con menos errores y que aporta una documentación más completa.