

1. SELECCIÓN DE ARQUITECTURAS

Y HERRAMIENTAS DE PROGRAMACIÓN

ACTIVIDADES FINALES

Actividades de comprobación

1.1. ¿Cuál es el objetivo último del desarrollo de software?

- a) Crear empleo.
- b) Resolver una necesidad.
- c) Reducir costes.
- d) Promover la innovación.

1.2. ¿Cuál de las siguientes no es una fase del proceso de desarrollo de software?

- a) Definición de requisitos.
- b) Envío del presupuesto.
- c) Diseño de la arquitectura.
- d) Despliegue y mantenimiento.

1.3. ¿Qué fase del ciclo de vida de un software consume más tiempo y recursos?

- a) Planificación.
- b) Implementación.
- c) Pruebas.
- d) Depende del tipo de aplicación.

1.4. ¿En qué enfoque de desarrollo de aplicaciones puedes programar a golpe de ratón?

- a) Desarrollo de aplicaciones personalizadas.
- b) Desarrollo rápido de aplicaciones.
- c) Desarrollo de aplicaciones low-code.

d) Desarrollo de aplicaciones móviles.

1.5. ¿De qué depende en gran medida la elección de un lenguaje de programación para el desarrollo de una aplicación web?

- a) Del presupuesto del proyecto.
- b) De la cualificación del equipo de desarrollo.
- c) De la cultura de la empresa.

d) De las tareas que debe completar la aplicación y las plataformas con las que trabaja.

1.6. ¿Cuál es el propósito de un desarrollador front-end?

- a) Vincular la presentación con la base de datos.
- b) Dar forma a la lógica de negocio de la aplicación.
- c) Optimizar el sistema de transmisión de datos.

d) Programar las partes visibles para el usuario.

1.7. ¿Cuál de las siguientes no es una ventaja de utilizar una arquitectura de tres niveles?

- a) Los niveles no pueden actualizarse sin afectar a los demás niveles.
- b) Cada nivel se ejecuta en su propia infraestructura.
- c) Cada nivel puede ser desarrollado de forma independiente.
- d) Un nivel se puede escalar sin afectar a los demás niveles.

1.8. ¿En qué nivel de la three-tier architecture encaja el desarrollo del lado cliente?

- a) Nivel de datos.
- b) Nivel de presentación.
- c) Nivel de aplicación.
- d) El lado cliente no tiene sentido en esta arquitectura.

1.9. ¿Qué niveles de la arquitectura de tres niveles conforman el back-end?

- a) Presentación y datos.
- b) Aplicación y datos.
- c) Presentación y aplicación.

d) Aplicación.

1.10. ¿Cuáles son las tres tecnologías que son imprescindibles dominar para el desarrollo web en entorno cliente?

a) SASS, HTML y CSS.

b) JavaScript, AngularJS y React.

c) HTML, CSS y JavaScript.

d) jQuery, vue.js y Git.

1.11. ¿Qué característica no pertenece a JavaScript?

a) Es necesaria una conexión a internet para que funcione.

b) Es fácil de implementar.

c) Ayuda a corregir diseños de CSS.

d) Aporta dinamismo a una web.

1.12. Señala una alternativa a JavaScript

d)Python

1.14. Una de las características que se han estudiado sobre JavaScript es que aportan dinamismo sin tener que esperar a que el servidor reaccione y muestre otra página. Pero, ¿por qué es esto tan importante? ¿Solo por no hacer esperar al usuario? Reflexiona y escribe una lista con todas las ventajas adicionales que esto supone en tu opinión.

El dinamismo en JavaScript es importante porque mejora la experiencia de usuario y permite que el contenido cambie sin necesidad de refrescar la página.

1.15. Busca en los sitios web de referencia de estadísticas más conocidos sobre tecnología qué porcentaje de aplicaciones web se crean con JavaScript en la actualidad.

El 61 %de todos los desarrolladores del mundo utiliza JavaScript para crear páginas web.

1.16. Busca e instala en tu navegador preferido un plugin o extensión que te permita simular cómo se vería cualquier web en cualquier tamaño de pantalla de los dispositivos más usados en la actualidad.

Elegí la más conocida que sería Responsive Viewer.

1.17. Instala en tu navegador los cuatro navegadores más usados del momento, ejecuta en cada uno de ellos la misma tarea y observa en los monitores de recursos de tu sistema operativo el consumo de cada navegador. ¿Cuál de ellos ha consumido menos recursos? Repite el ejercicio seleccionando diferentes tareas: cargando una web sencilla, una web de alto tráfico, un juego, un video de alta calidad...

Los cuatro navegadores más usados en el mundo son **Google Chrome, Safari, Mozilla Firefox y Microsoft Edge**.

- **Firefox** suele consumir menos RAM en sitios simples.
- **Chrome** y **Edge** usan más memoria por pestaña debido al aislamiento de procesos.
- **Safari** (en Mac) es el más eficiente energéticamente.

1.18. Activa y desactiva JavaScript en los cuatro navegadores más utilizados y comprueba en cada caso que se ha realizado correctamente.

Chrome / Edge:

- Ir a `chrome://settings/content/javascript` o `edge://settings/content/javascript`.
- Activar o desactivar "Permitir JavaScript".
- Comprobar con <https://www.whatismybrowser.com/detect/is-javascript-enabled>.

Firefox:

- En la barra escribe `about:config`.
- Busca `javascript.enabled` → doble clic (`true` = activo / `false` = inactivo).
- Comprobar en el mismo enlace.

Safari:

- Menú **Safari > Preferencias > Seguridad**.
- (Des)marca "Permitir JavaScript".
- Comprobar con el enlace anterior.

Actividades de aplicación

1.13. De todas las tecnologías del lado cliente que se han citado a lo largo de la unidad, selecciona cuatro de ellas para cada uno de estos desarrollos. Razona tu respuesta en cada

caso:

a) Un sitio web estéticamente muy llamativo cuyo único propósito es recabar direcciones de correo electrónico a través de un formulario.

HTML, CSS, JavaScript y jQuery serían los más óptimos ya que con ayuda de jQuery podemos manipular elementos de una página web o hacer animaciones, sin necesidad de escribir JavaScript puro.

b) Un sitio web de alto tráfico dedicado a la votación de las mejores noticias del día, con comentarios, reacciones y discusiones en línea.

HTML, CSS, JavaScript y React , que este último es una librería más flexible que se enfoca en crear componentes de interfaz de usuario. Con esta combinación podríamos crear una aplicación bastante completa y sólida.

c) Un sitio web estático cuyo único objetivo es informar de las condiciones de acceso a una beca.

HTML, CSS, JavaScript y Angular , aparte de los tres pilares del front-end Angular es una solución todo en uno (un *framework*) para crear aplicaciones grandes y complejas , ya que más que estética , para una beca se necesita funcionalidad.