**TALLER ARQUITECTURA DE APLICACIONES WEB**

La arquitectura de aplicaciones web es fundamental para diseñar sistemas eficientes y escalables. Permite que los distintos componentes de una aplicación interactúen de manera coherente.

1. ¿Qué es la Arquitectura de las Aplicaciones Web?

La **arquitectura de aplicaciones web** es el esqueleto o la estructura fundamental sobre la cual se diseña, construye y despliega una aplicación web. Define cómo los diferentes componentes del software, como la interfaz de usuario (frontend), la lógica del negocio (backend) y la base de datos, se organizan e interactúan entre sí. En esencia, es el plano que guía el desarrollo y asegura que la aplicación sea funcional, escalable y mantenible.

1. ¿Por qué es importante la Arquitectura de las Aplicaciones Web?

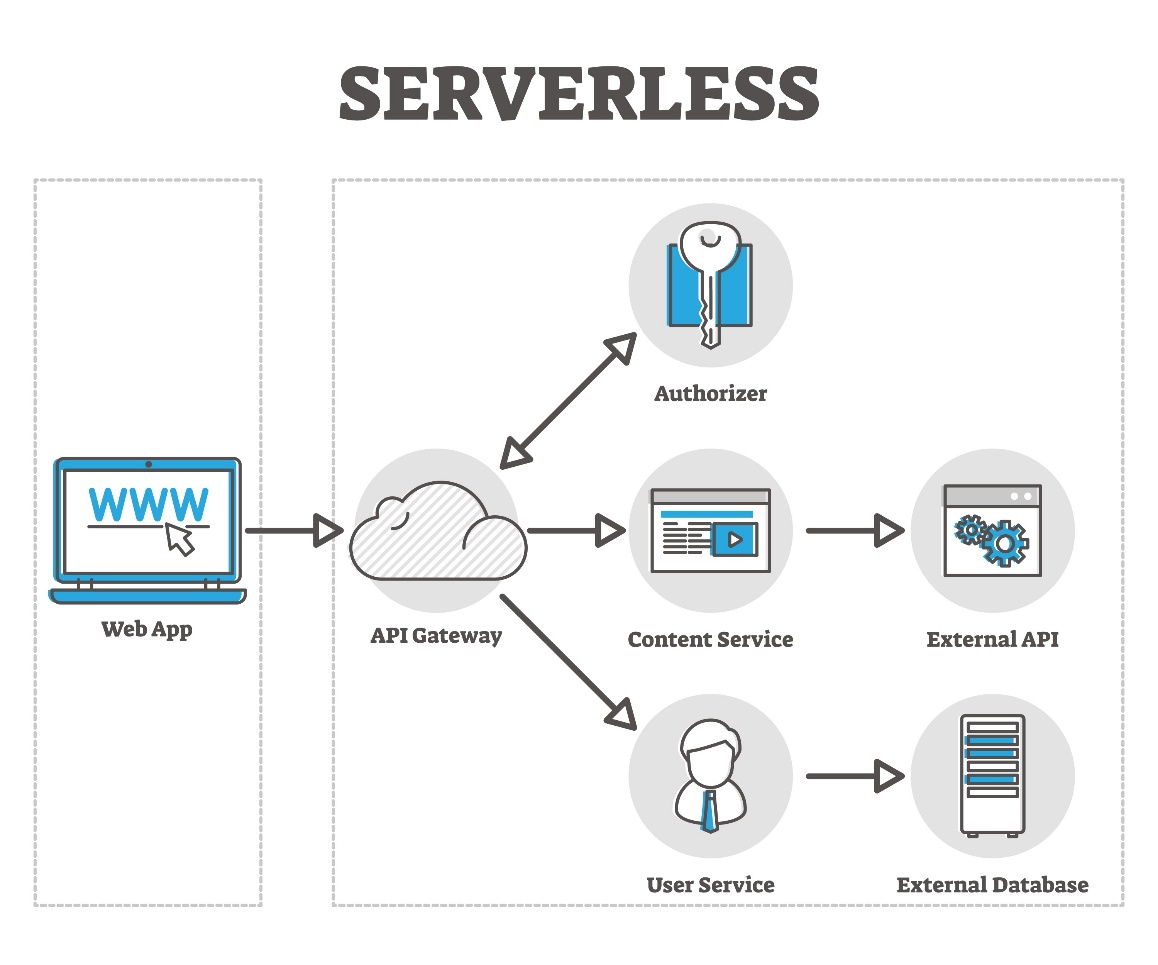
Una arquitectura bien diseñada es crucial por varias razones:

* **Escalabilidad:** Permite que la aplicación crezca y maneje un mayor número de usuarios y datos sin degradar su rendimiento.
* **Rendimiento:** Asegura que la aplicación sea rápida y responda eficientemente a las peticiones del usuario.
* **Mantenibilidad:** Facilita la corrección de errores, la actualización de componentes y la adición de nuevas funcionalidades sin tener que reescribir todo el sistema.
* **Seguridad:** Una buena arquitectura establece defensas robustas contra amenazas y vulnerabilidades comunes.
* **Organización del equipo:** Permite que diferentes equipos de desarrolladores trabajen en distintas partes de la aplicación de manera simultánea y coordinada.

1. ¿Cuáles son los Componentes de la Aplicación Web?

Generalmente, una aplicación web moderna se compone de dos partes principales:

* **Componentes del lado del cliente (Frontend):** Es todo lo que el usuario ve e interactúa en su navegador. Se encarga de la presentación y la experiencia de usuario (UI/UX).
* **Componentes del lado del servidor (Backend):** Es el "cerebro" de la aplicación. Incluye:
  + **Servidor Web:** Gestiona las peticiones de los clientes.
  + **Lógica de la Aplicación:** Procesa los datos, aplica las reglas del negocio y toma decisiones.
  + **Base de Datos:** Almacena y recupera toda la información de la aplicación de manera persistente.



1. Mencione las Mejores Prácticas para el desarrollo web

 **Diseño Responsivo (Responsive Design):** Asegurar que la aplicación se vea y funcione bien en cualquier dispositivo (móviles, tabletas, ordenadores).

 **Optimización del Rendimiento:** Minimizar los tiempos de carga, optimizar imágenes y reducir las peticiones al servidor.

 **Seguridad Primero (Security First):** Implementar medidas de seguridad desde el inicio del desarrollo para proteger los datos de los usuarios.

 **Código Limpio y Modular:** Escribir código que sea fácil de leer, entender y mantener.

 **Pruebas Continuas (Testing):** Realizar pruebas automatizadas y manuales para asegurar que la aplicación funcione correctamente y sin errores.

 **Control de Versiones:** Utilizar sistemas como Git para gestionar los cambios en el código y facilitar la colaboración.

1. Escriba y esplique brevemente los Tipos de Arquitectura de Aplicaciones Web, sus ventajas y desventajas

Existen varios modelos o patrones arquitectónicos para estructurar aplicaciones web. La elección de uno u otro depende de factores como la complejidad del proyecto, los requisitos de escalabilidad, el tamaño del equipo de desarrollo y el presupuesto. A continuación, se describen los tipos más comunes:

**Arquitectura Monolítica**

* **Explicación:** Este es el enfoque tradicional donde todos los componentes de la aplicación (la interfaz de usuario, la lógica de negocio y el acceso a datos) se desarrollan y despliegan como una única unidad indivisible. Es como una casa de un solo bloque donde todo está bajo el mismo techo.
* **Ventajas:** Su principal ventaja es la simplicidad en las fases iniciales del desarrollo y el despliegue. Al ser una sola base de código, es más fácil de probar y depurar al principio.
* **Desventajas:** A medida que la aplicación crece, se vuelve muy difícil de escalar, mantener y actualizar. Un pequeño cambio requiere volver a desplegar toda la aplicación, y un error en cualquier módulo puede hacer que todo el sistema falle.

**Arquitectura de Microservicios**

* **Explicación:** En este modelo, la aplicación se descompone en un conjunto de servicios pequeños e independientes que se comunican entre sí. Cada servicio se enfoca en una única función de negocio (como la gestión de usuarios, el procesamiento de pagos o el catálogo de productos) y puede ser desarrollado, desplegado y escalado de forma autónoma.
* **Ventajas:** Ofrece una alta escalabilidad, ya que solo se escalan los servicios que lo necesitan. Aporta flexibilidad tecnológica, permitiendo usar diferentes tecnologías para cada servicio, y facilita el mantenimiento y las actualizaciones continuas sin afectar al resto de la aplicación.
* **Desventajas:** La principal desventaja es su complejidad. La gestión de múltiples servicios, la comunicación entre ellos y el despliegue distribuido requieren un mayor esfuerzo y herramientas más sofisticadas.

**Aplicación de Página Única (Single-Page Application - SPA)**

* **Explicación:** Es una arquitectura donde la mayor parte de la lógica de la aplicación se ejecuta en el navegador del cliente. La aplicación carga una única página HTML y, a medida que el usuario interactúa, actualiza el contenido de forma dinámica solicitando solo los datos necesarios al servidor, sin necesidad de recargar la página completa.
* **Ventajas:** Proporciona una experiencia de usuario muy fluida y rápida, similar a la de una aplicación de escritorio o móvil. Reduce la carga en el servidor, ya que este solo se encarga de enviar datos en bruto.
* **Desventajas:** El tiempo de carga inicial puede ser más largo, ya que debe cargarse todo el framework de la aplicación. Además, puede presentar desafíos para la optimización en motores de búsqueda (SEO) si no se implementan técnicas específicas.

**Arquitectura sin Servidor (Serverless)**

* **Explicación:** A pesar de su nombre, sí existen servidores, pero el desarrollador no tiene que gestionarlos. En este modelo, se utiliza un proveedor de servicios en la nube (como AWS, Google Cloud o Azure) que se encarga de toda la infraestructura. El desarrollador solo escribe y sube el código de la lógica de negocio en forma de funciones que se ejecutan en respuesta a eventos específicos.
* **Ventajas:** La principal ventaja es la reducción de costos, ya que se paga únicamente por el tiempo de ejecución del código, no por un servidor que esté siempre encendido. Ofrece escalabilidad automática y libera a los desarrolladores de la gestión de la infraestructura.
* **Desventajas:** Existe una fuerte dependencia del proveedor de la nube (vendor lock-in). También puede haber limitaciones en el tiempo de ejecución de las funciones y puede ser más complejo depurar y monitorear el sistema en comparación con arquitecturas tradicionales.

1. Escriba Ejemplos de Tecnologías aplicada a:

**a. Frontend:** React, Angular, Vue.js, Svelte, HTML5, CSS3, JavaScript.

**b. Backend:** ASP.NET, Node.js (Express), Python (Django, Flask), Java (Spring), PHP (Laravel).

**c. Base de Datos:** SQL Server, MySQL, PostgreSQL, MongoDB (NoSQL), Oracle Database.

**d. Almacenamiento de Imágenes:** Amazon S3, Google Cloud Storage, Azure Blob Storage, Cloudinary.

1. Que es un protocolo de comunicación

Un **protocolo de comunicación** es un conjunto de reglas y estándares que definen cómo se establece, mantiene y termina la comunicación entre dos o más dispositivos en una red. Especifican el formato, el tiempo, la secuencia y el control de errores de los mensajes intercambiados.

1. Explique los protocolos http, https

 **HTTP (Protocolo de Transferencia de Hipertexto):** Es el protocolo fundamental de la World Wide Web. Se utiliza para transmitir datos, como páginas web, entre un navegador (cliente) y un servidor web. Sin embargo, los datos enviados a través de HTTP no están encriptados, lo que los hace vulnerables a ser interceptados.

 **HTTPS (Protocolo de Transferencia de Hipertexto Seguro):** Es la versión segura de HTTP. Utiliza una capa adicional de seguridad llamada SSL/TLS (Capa de Sockets Seguros/Seguridad de la Capa de Transporte) para cifrar la comunicación entre el cliente y el servidor. Esto asegura que la información, como contraseñas o datos de tarjetas de crédito, se transmita de forma confidencial y protegida contra espionaje.

1. Que es hosting, investigue los tipos de hosting y haga una tabla comparativa de mínimo 4 proveedores de este servicio, elija uno apropiado para su proyecto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caracteristica | Hostinger | Bluehost | SiteGround | GoDaddy |
| Precio inicial | Muy bajo 3$/mes | Bajo 4$/Mes | Medio 5$/Mes | Bajo a Medio |
| Rendimiento | Medio | Bueno, recomendado en wordpress | Excelente, rápido y fiable | Bueno y con rendimiento consistente |
| Facilidad de uso | Sencillo, panel de control propio (hpanel) | Facil (cPanel estándar) | Facil, panel propio (Site Tools) | Facil con panel de control intuitivo |
| Soporte Tecnico | Chat en vivo 24/7 | Chat y teléfono 24/7 | Chat, teléfono, tickets 24/7 | Telefono y chat 24/7 |
| Ideal para: | Principiantes, blogs, sitios pequeños | Sitios de WordPress | Negocios, tiendas online | Pequeños negocios, compras de dominios |

1. Que es un servidor de Dominio, escojan un dominio para su proyecto e investiga si está disponible, agregue capturas para comprobar su investigación

Un **servidor de dominio** (o Servidor de Nombres de Dominio - DNS) es como la agenda de contactos de Internet. Su función principal es traducir los nombres de dominio que las personas pueden leer (como www.google.com) a las direcciones IP numéricas que las computadoras usan para identificarse entre sí en la red (como 142.250.217.14). Sin el DNS, tendríamos que memorizar largas secuencias de números para visitar cualquier sitio web.

**Investigación de Dominio para un Proyecto (Ejemplo)**

Supongamos que nuestro proyecto es una plataforma para conectar a artesanos locales y la hemos llamado **"ArteConecta"**.

* **Dominio escogido:** arteconecta.com

A continuación, se muestra una investigación de su disponibilidad utilizando un registrador de dominios como GoDaddy.

**Comprobación de Disponibilidad:**

Al buscar arteconecta.com, el sistema nos informa si está disponible o si ya ha sido registrado por otra persona.

* **Captura de pantalla de la investigación (Simulada):**

**Resultado:** La investigación muestra que el dominio arteconecta.com ya está registrado.

* **Captura de pantalla de Alternativas (Simulada):**

**Resultado:** El sistema sugiere alternativas como arteconecta.net, arteconecta.org, o arteconecta.co. Para un proyecto enfocado en Colombia, arteconecta.co sería una excelente alternativa y está disponible.