

**TALLER DE ARQUITECTURAS DE LAS APLICACIONES WEB**

**BOHORQUEZ PRIETO JUAN DAVID**

**ANGARITA VEGA KEVIN JULIÁN**

**UNIVERSIDAD DE PAMPLONA**

**FACULTAD DE INGENIERÍAS Y ARQUITECTURA**

**PROGRAMA INGENIERÍA DE SISTEMAS**

**BASE DE DATOS I - AR**

**VILLA ROSARIO**

**2025**

## 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?

RTA La Arquitectura de las Aplicaciones Web es el esquema o diseño estructural que define cómo se organizan e interactúan entre sí los distintos componentes de una aplicación basada en la web. Esta arquitectura especifica la relación y la comunicación entre el cliente (interfaz de usuario) y el servidor, así como entre los diversos módulos y subsistemas que la componen, con el fin de garantizar un funcionamiento eficiente, escalabilidad, mantenimiento y una buena experiencia de usuario. Incluye la estructura tanto del lado del cliente (front-end) como del lado del servidor (back-end) y puede involucrar componentes como servidores web, bases de datos, pasarelas API y sistemas de almacenamiento.

La arquitectura de aplicaciones web determina cómo la aplicación procesa las solicitudes del usuario, cómo se distribuye la carga entre servidores y la manera en que se envía la respuesta al usuario final, a menudo utilizando técnicas como cachés, equilibradores de carga y contenedores del lado del servidor. Esta arquitectura es fundamental para el rendimiento y la personalización de la aplicación web, y una elección incorrecta puede afectar negativamente el negocio o la experiencia del usuario. Específicamente, se puede dividir en capas tales como la capa de presentación (interfaz), la capa de aplicación (lógica de negocio) y la capa de almacenamiento persistente (bases de datos o servicios en la nube)

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

RTA La Arquitectura de las Aplicaciones Web es importante porque establece la base para el funcionamiento eficiente, escalable y mantenible de una aplicación. Una buena arquitectura permite que la aplicación se adapte fácilmente a las necesidades cambiantes del negocio, ofreciendo flexibilidad para crecer y ajustarse a nuevos requerimientos sin generar grandes costos o retrabajos. Además, facilita un desarrollo organizado y modular, lo que optimiza el trabajo en equipo y acelera la creación y mejora de funcionalidades. También mejora la gestión del código, haciendo que los cambios, actualizaciones o migraciones sean más sencillos y menos riesgosos. En resumen, la arquitectura adecuada asegura un mejor rendimiento, seguridad, mantenimiento, experiencia del usuario y sostenibilidad a largo plazo de la aplicación web.

#### **Beneficios clave de una buena arquitectura**

- Adaptabilidad a cambios del negocio.
- Desarrollo modular y organizado.
- Mejor mantenimiento y gestión del código.

- Escalabilidad y flexibilidad.
- Optimización del rendimiento y seguridad.

### 3. ¿Cuáles son los Componentes de la Aplicación Web?

RTA La Arquitectura de las Aplicaciones Web es un diseño estructural compuesto por varios componentes clave que interactúan para ofrecer funcionalidad, escalabilidad y rendimiento. A continuación, una descripción completa de sus componentes principales organizados en capas:

#### **Capa de Presentación (Cliente - Front-end)**

Esta capa incluye todo lo que el usuario final ve y con lo que interactúa directamente.

- Interfaz de Usuario (UI): Contiene elementos visuales como páginas, formularios, menús y botones. Se diseña para ser responsable y compatible con distintos dispositivos.
- Programación del lado del cliente: Se utilizan lenguajes como JavaScript para añadir interactividad y dinamismo sin recargar la página completa. Los frameworks comunes incluyen React, Angular y Vue.js, que facilitan la creación de interfaces complejas y reactivas.

#### **Capa de Negocio o Lógica (Servidor - Back-end)**

Es la capa que procesa las solicitudes recibidas del cliente y ejecuta la lógica del negocio.

- Lógica de la aplicación: Gestiona las reglas del negocio, controla qué información se procesa y define cómo se atienden las solicitudes. Se implementa con lenguajes como Node.js, Python, Ruby o Java, con frameworks como Express.js o Django.
- Servidor de aplicaciones: Es la plataforma que ejecuta la lógica empresarial, conecta con bases de datos, y maneja sesiones y autenticación.
- Servicios API: Facilitan la interacción entre diferentes sistemas o componentes mediante protocolos estándar, exponiendo funcionalidades de manera controlada.

#### **Capa de Datos y Almacenamiento**

Esta capa se encarga de almacenar y gestionar la persistencia de datos.

- ❖ Base de datos: Puede ser relacional (SQL) o NoSQL, donde se guardan los datos de la aplicación. La base de datos se conecta al servidor para operaciones de lectura y escritura.

- ❖ Sistemas de caché: Mejoran el rendimiento al guardar temporalmente los datos solicitados frecuentemente.
- ❖ Almacenamiento en la nube: Opcional para archivos, logs y otros datos que requieren disponibilidad distribuida y escalabilidad.

### **Componentes de Infraestructura y Redes**

- ✓ Estos componentes hacen posible que la aplicación funcione correctamente en Internet.
- ✓ Servidor web: Maneja y responde las solicitudes HTTP del cliente.
- ✓ Balanceador de carga (Load Balancer): Distribuye el tráfico entre varios servidores para evitar sobrecargas y mejorar la disponibilidad.
- ✓ CDN (Content Delivery Network): Distribuye contenido estático como imágenes y archivos para reducir la latencia y mejorar la velocidad de carga.
- ✓ Middleware y seguridad: Incluye capas de autenticación, autorización, cifrado y validaciones para proteger los datos y la comunicación.

### **Patrones y modelos usados**

- Arquitectura en capas: separación clara entre presentación, negocio y datos.
- Modelo-Vista-Controlador (MVC): estructura común para organizar componentes.
- Microservicios: divide la aplicación en servicios pequeños e independientes para facilitar la escalabilidad y mantenimiento.
- Arquitectura sin servidor (Serverless): basada en funciones ejecutadas en la nube sin gestión directa de servidores.

#### **4. Mencione las Mejores Prácticas para el desarrollo web RTA**

##### **Diseño web adaptable**

El diseño web responsivo es una práctica recomendada crucial para el desarrollo web de pymes, emprendedores, profesionales del marketing, dueños de negocios, agencias, directores ejecutivos, directores de marketing y cualquier persona involucrada en el marketing digital. Es un enfoque que garantiza que su sitio web se adapte y se visualice a la perfección en todos los dispositivos y tamaños de pantalla, desde grandes monitores de escritorio hasta pequeñas pantallas de smartphones. En el mundo actual, priorizado por los dispositivos móviles, un sitio web responsivo ya no es un lujo: es una necesidad. Este enfoque utiliza una combinación de cuadrículas flexibles, imágenes y consultas de medios

CSS para crear una experiencia de usuario fluida y consistente, independientemente de cómo acceda su público a su sitio. Por eso, merece un lugar destacado en esta lista de prácticas recomendadas para el desarrollo web.

### **Optimización del rendimiento del sitio web**

Optimizar el rendimiento de un sitio web es crucial para crear una experiencia de usuario positiva y alcanzar el éxito online. Se centra en mejorar la velocidad y la eficiencia de tu sitio web, garantizando que las páginas carguen rápidamente sin importar el dispositivo o la conexión a internet del usuario. Un sitio web lento puede generar visitantes frustrados, altas tasas de rebote (que abandonan el sitio rápidamente) y, en última instancia, la pérdida de oportunidades de negocio. En el acelerado mundo digital actual, los usuarios esperan una gratificación instantánea; si tu sitio no se la ofrece, rápidamente encontrarán uno que sí. Esta práctica merece un lugar en cualquier lista de "mejores prácticas de desarrollo web", ya que impacta directamente en la satisfacción del usuario, las tasas de conversión y el posicionamiento en buscadores.

### **Implementación de la accesibilidad (a11y)**

La accesibilidad (a menudo abreviada como a11y) en el desarrollo web garantiza que su sitio sea usable por todos, incluidas las personas con discapacidad. Esto significa diseñar y construir su sitio web para que las personas con discapacidades visuales, auditivas, motoras o cognitivas puedan percibir, comprender, navegar e interactuar eficazmente con su contenido. Implementar la accesibilidad no se trata solo de "ser amable", sino de crear una mejor experiencia para todos los usuarios y abrir su negocio a un público más amplio.

### **Search Engine Optimization (SEO)**

La optimización para motores de búsqueda (SEO) consiste en optimizar tu sitio web para aumentar su visibilidad en buscadores como Google. Al optimizar tu contenido y configuración técnica, atraes más tráfico orgánico (gratuito) de personas que buscan información relacionada con tu negocio. Esto implica comprender qué busca tu público objetivo (investigación de palabras clave) y adaptar el contenido de tu sitio web para satisfacer esas necesidades. En esencia, el SEO consiste en hacer que tu sitio web sea fácilmente detectable tanto por los motores de búsqueda como por los usuarios.

### **Implementación de seguridad**

La implementación de la seguridad es un aspecto crucial de las mejores prácticas de desarrollo web. Abarca todas las medidas y protocolos implementados para proteger su sitio web de una amplia gama de ciberamenazas. Estas amenazas incluyen accesos no autorizados, filtraciones de datos, infecciones de malware y otras vulnerabilidades. Para pymes, emprendedores, profesionales del marketing, propietarios de negocios, agencias, directores ejecutivos, directores de marketing y responsables de marketing, una seguridad robusta no es solo un detalle técnico, sino un imperativo empresarial. Descuidarla puede tener consecuencias devastadoras, dañar su reputación, minar la confianza de los clientes e incurrir en importantes pérdidas financieras.

### **Arquitectura modular y basada en componentes**

La arquitectura modular y basada en componentes es una práctica recomendada crucial para el desarrollo web, que impacta significativamente la eficiencia, la mantenibilidad y la escalabilidad de tu proyecto. Es un enfoque moderno que merece un lugar en esta lista porque agiliza el proceso de desarrollo y prepara tu sitio web para el éxito a largo plazo. En lugar de crear páginas web como entidades monolíticas, esta metodología divide la interfaz de usuario en módulos reutilizables e independientes, o "componentes". Piensa en ello como construir con LEGO: tienes piezas individuales (componentes) que puedes combinar de diversas maneras para crear diferentes estructuras (páginas).

### **Control de versiones e implementación de CI/CD**

El control de versiones y la Integración/Despliegue Continuo (CI/CD) son prácticas recomendadas cruciales para el desarrollo web, especialmente para empresas en crecimiento, emprendedores, profesionales del marketing y agencias que buscan soluciones web escalables y fáciles de mantener. Estas prácticas optimizan el proceso de desarrollo, mejoran la calidad del código y aceleran la entrega, lo que las hace esenciales para cualquier proyecto web moderno.

### **Supervisión y análisis del rendimiento**

La monitorización y el análisis del rendimiento son cruciales para cualquier sitio web que busque ofrecer una experiencia de usuario excepcional y alcanzar sus objetivos comerciales. Esta práctica recomendada implica implementar herramientas y procesos para monitorizar diversos aspectos del rendimiento, el comportamiento del usuario y las tasas de conversión de su sitio web. Al adoptar un enfoque basado en datos, obtendrá información valiosa para identificar cuellos de botella, optimizar la experiencia del usuario y tomar decisiones informadas sobre las futuras prioridades de desarrollo, lo que lo convierte en un elemento fundamental de las prácticas recomendadas para el desarrollo web.

5. Escriba y explique brevemente los Tipos de Arquitectura de Aplicaciones Web, sus ventajas y desventajas

RTA

### **TIPOS DE ARQUITECTURA DE APLICACIONES**

#### **Arquitectura de Una Sola Página**

La arquitectura de una aplicación de una sola página (SPA) es tan sencilla como su nombre: toda la aplicación se basa en una sola página. Una vez que el usuario accede a tu aplicación, no necesita navegar a ninguna otra página web. La aplicación se hace lo suficientemente dinámica como para obtener y renderizar pantallas que satisfagan los requisitos de los usuarios mientras navegan por la propia aplicación.

Las SPA son estupendas cuando se trata de proporcionar una experiencia rápida y sin fisuras a los usuarios finales o consumidores. Sin embargo, carecen del toque de un sitio web tradicional, y pueden ser difíciles de optimizar para el SEO.

### **Ventajas de la Arquitectura SPA**

Algunos de los pros de la arquitectura SPA son:

Puedes construir aplicaciones web altamente interactivas.

Las SPA son fáciles de escalar.

Optimizar las SPA para el rendimiento no requiere mucho esfuerzo.

### **Desventajas de la Arquitectura SPA**

Algunos de los inconvenientes de la arquitectura SPA son:

Las SPA limitan la flexibilidad con los hipervínculos y el SEO.

El renderizado inicial suele ser lento.

La navegación por la aplicación puede ser poco intuitiva.

### **Arquitectura de Aplicaciones Web Progresivas**

La arquitectura de la Aplicación Web Progresiva (PWA) se basa en la Arquitectura de Página Única proporcionando capacidades offline para tu aplicación web. Tecnologías como Capacitor e Ionic se utilizan para crear PWA que pueden proporcionar a los usuarios una experiencia uniforme en todas las plataformas.

Al igual que las SPA, las PWA son fluidas y sin fisuras. Con la capacidad añadida de instalarse en los dispositivos de los usuarios (a través de service workers), tus usuarios obtienen una experiencia más uniforme con tu aplicación.

Al mismo tiempo, puede ser difícil optimizar estas aplicaciones para el SEO, y las actualizaciones de las aplicaciones instaladas pueden ser difíciles de impulsar.

### **Ventajas de la Arquitectura PWA**

Hay muchas ventajas de la arquitectura PWA, entre ellas:

Las aplicaciones se ejecutan con mucha fluidez y ofrecen compatibilidad entre plataformas.

La escalabilidad es sencilla.

Los desarrolladores pueden acceder al acceso sin conexión y a las API nativas de los dispositivos, como los trabajadores en segundo plano y las notificaciones push.

### **Desventajas de la Arquitectura PWA**

Algunos de los contras de la arquitectura PWA pueden ser:

Hay un soporte limitado para la gestión de enlaces y el SEO.

Enviar actualizaciones a las PWA sin conexión es más complejo que con las aplicaciones nativas.

La compatibilidad de las PWA con los navegadores web y sistemas operativos es limitada.

### **Arquitectura de Renderizado del Lado del Servidor**

En la renderización del lado del servidor (SSR), las páginas web del frontend se renderizan en un servidor backend después de ser solicitadas por el usuario. Esto ayuda a reducir la carga en el dispositivo del cliente, ya que recibe una página web estática de HTML, CSS y JS.

Las aplicaciones SSR son muy populares entre los blogs y los sitios web de comercio electrónico. Esto se debe a que hacen que la gestión de enlaces y el SEO sean bastante sencillos. Además, la primera representación de las aplicaciones SSR es bastante rápida, ya que el cliente no tiene que procesar ningún código JS para representar las pantallas.

### **Ventajas de la Arquitectura SSR**

Estas aplicaciones son estupendas para los sitios web con mucho SEO. La carga de la primera página es casi instantánea en la mayoría de los casos. Puedes combinarla con un servicio de caché para mejorar aún más el rendimiento de tu aplicación.

### **Desventajas de la Arquitectura SSR**

Algunos de los inconvenientes de utilizar la arquitectura SSR son:

No se recomienda para páginas web complejas o pesadas, ya que el servidor puede tardar en generar completamente la página, lo que provoca un retraso en la primera carga. Se recomienda sobre todo para aplicaciones que no se centran mucho en la interfaz de usuario y sólo buscan una mayor escalabilidad o seguridad.

### **Arquitectura de Aplicaciones Prerrenderizadas**

La arquitectura de aplicaciones pre-renderizadas también se conoce como arquitectura de generación de sitios estáticos. En esta arquitectura, las páginas web del frontend de la aplicación se generan previamente y se almacenan como archivos HTML, CSS y JS planos en el servidor. Una vez que el usuario solicita una página, ésta se obtiene directamente y se le muestra. Esto hace que la aplicación web sea muy rápida, con tiempos de carga mínimos de cualquier tipo. Sin embargo, esta arquitectura aumenta el tiempo de construcción de la aplicación, ya que las páginas web se renderizan durante el proceso de construcción.

Las aplicaciones web prerrenderizadas son estupendas para cuando quieres generar contenido estático, como blogs o detalles de productos que no cambian a menudo. También puedes utilizar plantillas para simplificar el diseño de tu página web. Sin embargo, es casi imposible crear aplicaciones web dinámicas con esta arquitectura. Si quieres construir una página de búsqueda que tome la consulta en su ruta estás en el lugar equivocado.

Dado que cada posible ruta de la aplicación se pre-renderiza durante el proceso de construcción, es imposible tener rutas dinámicas como las anteriores, ya que hay infinitas



posibilidades que no pueden pre-renderizarse durante la construcción (y tampoco tiene sentido hacerlo).

### **Ventajas de la Arquitectura Prerrenderizada**

Algunas de las principales ventajas de la arquitectura de aplicaciones prerrenderizadas son:

Las páginas web se generan en HTML, CSS y JS puros, por lo que su rendimiento es similar al de las aplicaciones construidas con vanilla JS.

Si conoces todas las rutas posibles de tu aplicación, el SEO se vuelve superfácil.

### **Desventajas de la Arquitectura Prerrenderizada**

Como cualquier modelo de arquitectura, la pre-renderizada tiene sus inconvenientes:

El contenido dinámico no se puede servir con estas aplicaciones.

Hacer cualquier cambio en la aplicación web significa reconstruir y desplegar completamente la aplicación desde cero.

### **Arquitectura de Aplicaciones Isomórficas**

Las aplicaciones isomórficas son aquellas que son una mezcla de aplicaciones renderizadas en el lado del servidor y SPA. Esto significa que dichas aplicaciones se renderizan primero en el servidor como una aplicación normal renderizada del lado del servidor. Una vez que las recibe el cliente, la app se hidrata y adjunta el DOM virtual para un procesamiento más rápido y eficiente por parte del cliente. Esto convierte esencialmente la aplicación en una aplicación de una sola página.

Isomorphic reúne lo mejor de ambos mundos. El procesamiento y la interfaz de usuario del cliente son superrápidos, gracias a la SPA. También obtienes una renderizada inicial rápida y un soporte completo de SEO y enlaces, gracias al renderizado del lado del servidor.

### **Ventajas de la Arquitectura Isomórfica**

Éstas son algunas de las ventajas de utilizar una arquitectura de aplicación isomórfica:

Las apps isomórficas tienen una renderización inicial súper rápida y soporte completo para el SEO.

Estas aplicaciones también tienen un buen rendimiento en el cliente, ya que se convierten en una SPA después de la carga.

### **Desventajas de la Arquitectura Isomórfica**

Algunos de los contras de la arquitectura de aplicación isomórfica pueden ser:

La puesta en marcha de una aplicación de este tipo requiere de un conocimiento profundo. Las opciones del stack tecnológico son limitadas cuando se trata de diseñar una aplicación isomórfica. Sólo puedes elegir entre un puñado de librerías y frameworks (en su mayoría) basados en JS.

### **Arquitectura Orientada al Servicio**

La arquitectura orientada a servicios es una de las alternativas más populares a la forma tradicional de construir aplicaciones en forma de monolito. En esta arquitectura, las aplicaciones web se dividen en servicios que representan una unidad funcional de negocio cada uno. Estos servicios están débilmente acoplados e interactúan entre sí mediante el paso de mensajes.

La arquitectura orientada a servicios añade estabilidad y escalabilidad al stack tecnológico de tu aplicación. Sin embargo, el tamaño de los servicios en la SOA no está claramente definidos y suelen estar vinculados a los componentes de negocio, no a los componentes técnicos; de ahí que el mantenimiento pueda ser a veces un problema.

### **Ventajas de la Arquitectura Orientada a Servicios**

Las principales ventajas de la arquitectura orientada a servicios son:

Esta arquitectura ayuda a construir aplicaciones altamente escalables y fiables. Los componentes son reutilizables y se comparten para mejorar los esfuerzos de desarrollo y mantenimiento.

### **Desventajas de la Arquitectura Orientada a Servicios**

Aquí tienes una lista de los posibles inconvenientes de utilizar la arquitectura orientada a servicios:

Las aplicaciones SOA aún no son 100% flexibles, ya que el tamaño y el alcance de cada servicio no son fijos. Puede haber servicios del tamaño de aplicaciones empresariales que pueden ser difíciles de mantener.

La compartición de componentes introduce dependencias entre los servicios.

### **Arquitectura de Microservicios**

La arquitectura de microservicios fue diseñada para resolver los problemas de la arquitectura orientada a servicios. Los microservicios son componentes aún más modulares que encajan entre sí para construir una aplicación web. Sin embargo, los microservicios se centran en mantener cada componente pequeño y con un contexto limitado. Contexto delimitado significa esencialmente que cada microservicio tiene su código y sus datos acoplados con mínimas dependencias de otros microservicios.

La arquitectura de microservicios es probablemente la mejor arquitectura para construir aplicaciones que pretenden escalar algún día a miles y millones de usuarios. Cada componente es resistente, escalable y fácil de mantener. Sin embargo, el mantenimiento del ciclo de vida de DevOps para una aplicación basada en microservicios requiere esfuerzos adicionales, por lo que podría no ser adecuado para casos de uso más pequeños.

## **Ventajas de la Arquitectura de Microservicios**

Algunas de las ventajas de la arquitectura de microservicios son:

Los componentes de las aplicaciones son muy modulares, independientes y pueden reutilizarse en mayor medida que los de la arquitectura orientada a servicios. Cada componente puede escalarse de forma independiente para satisfacer el tráfico variable de usuarios. Las aplicaciones basadas en microservicios son altamente tolerantes a los fallos.

## **Desventajas de la Arquitectura de Microservicios**

Los inconvenientes de la arquitectura de microservicios son:

Para los proyectos más pequeños, la arquitectura de microservicios puede requerir demasiado esfuerzo de mantenimiento.

## **Arquitectura Sin Servidor**

La arquitectura sin servidor es otra de las novedades en el mundo de las arquitecturas de aplicaciones web. Esta arquitectura se centra en desglosar tu aplicación en términos de las funciones que debe realizar. A continuación, estas funciones se alojan en plataformas FaaS (Function-as-a-Service) como funciones que se invocan a medida que llegan las solicitudes.

A diferencia de la mayoría de las otras arquitecturas de esta lista, las aplicaciones construidas con la arquitectura sin servidor no permanecen en funcionamiento todo el tiempo. Se comportan como lo harían las funciones — esperan a ser llamadas, y al ser llamadas, ejecutan el proceso definido y devuelven un resultado. Debido a esta naturaleza, reducen los costes de mantenimiento y son altamente escalables sin mucho esfuerzo. Sin embargo, es difícil llevar a cabo tareas de larga duración utilizando estos componentes.

## **Ventajas de la Arquitectura Sin Servidor**

Estas son las principales ventajas de la arquitectura sin servidor:

Las aplicaciones sin servidor son muy fácilmente escalables. Incluso pueden adaptarse al tráfico entrante en tiempo real para reducir la carga de tu infraestructura. Estas aplicaciones pueden utilizar el modelo de precios de pago por uso de las plataformas sin servidor para reducir los costes de infraestructura. Las aplicaciones sin servidor son bastante fáciles de construir y desplegar, ya que todo lo que tienes que hacer es escribir una función y alojarla en una plataforma como funciones Firebase, AWS Lambda, etc.

## **Desventajas de la Arquitectura sin Servidor**

Las desventajas de la arquitectura sin servidor son:

Las tareas de larga duración pueden ser costosas en una arquitectura de este tipo. Cuando una función recibe una petición después de mucho tiempo, se conoce como arranque en frío. Los arranques en frío son lentos y pueden proporcionar una mala experiencia a tu usuario final.

6. Escriba Ejemplos de Tecnologías aplicada a:

- a. Frontend: React, ...
- b. Backend: ASP.NET ...
- c. Base de Datos: SQL Server, ...
- d. Almacenamiento de Imágenes: Amazon S3  
RTA

| Áreas                      | Ejemplos de Tecnologías                             | Explicación breve   |
|----------------------------|---|---|
| Interfaz                   | React, Angular, Vue.js, Svelte                      | Tecnologías que gestionan la interfaz y experiencia del usuario; React es popular por su enfoque en componentes reutilizables y rendimiento con DOM virtual. Vista es sencilla y flexible, Angular es robusto para grandes aplicaciones, Svelte optimiza el rendimiento con compilación estática. |
| Backend                    | ASP.NET, Node.js (Express.js), Django, Laravel      | Frameworks para lógica de negocio, servidores y gestión de bases de datos; ASP.NET es potente para aplicaciones empresariales, Node.js es versátil y rápido, Django es seguro y con arquitectura clara, Laravel popular en PHP para web robusta.  |
| Base de datos              | SQL Server, MySQL, PostgreSQL, MongoDB              | Sistemas para almacenar datos; SQL Server es un RDBMS robusto de Microsoft, MySQL y PostgreSQL son bases relacionales populares, MongoDB es NoSQL para datos no estructurados y flexibles.  |
| Almacenamiento de imágenes | Amazon S3, Google Cloud Storage, Azure Blob Storage | Servicios en la nube para guardar y distribuir archivos multimedia de forma escalable, segura y con alta disponibilidad. Amazon S3 es uno de los más usados por su  |

|  |  |  |
|--|--|--|
|  |  | confiabilidad y fácil integración con otros AWS. |
|--|--|--|

7. ¿Que es un protocolo de comunicación?

RTA Un protocolo de comunicación es un conjunto de reglas, normas y procedimientos que permiten que dos o más dispositivos o sistemas se comuniquen entre sí para intercambiar información de manera ordenada y eficiente. Estas reglas definen cómo se envían y reciben los datos, cómo se estructuran los mensajes, cómo se detectan y corrigen errores, y cómo se sincronizan las comunicaciones para garantizar que la información llegue correctamente al destino.

En esencia, un protocolo de comunicación asegura que los dispositivos involucrados en una comunicación hablen un mismo "idioma" y sigan un acuerdo común, facilitando así un intercambio de datos confiable y coherente. Puede estar implementado tanto en hardware como en software, y se aplica en redes de computadoras, telecomunicaciones y cualquier sistema que requiera transferencia de datos.

Por ejemplo, el conocido protocolo TCP/IP es el que permite el funcionamiento de Internet, fragmentando los datos en paquetes que viajan por diferentes rutas hasta llegar a su destino, donde se reensamblan adecuadamente. Otro ejemplo simple es definir las reglas para que una paloma mensajera entregue un mensaje siguiendo ciertas pautas para asegurarse de que el mensaje llegue y se comprenda.

8. Explique los protocolos http, https

RTA El protocolo HTTP (Protocolo de transferencia de hipertexto) es un protocolo de comunicación de capa de aplicación que sigue un esquema de solicitud-respuesta entre un cliente (como un navegador web) y un servidor. Este protocolo permite la transferencia de información en forma de páginas web y otros recursos a través de la red. Las solicitudes HTTP incluyen un método (como GET para obtener datos, POST para enviar datos, etc.) y una URI que identifica el recurso deseado. HTTP es un protocolo sin estado, lo que significa que cada solicitud es independiente y no guarda información entre ellas, aunque se pueden usar cookies para mantener el contexto de la sesión. Es un protocolo basado en texto, lo que facilita la depuración y la comprensión de las comunicaciones.

Por otro lado, HTTPS (HTTP Secure) es una versión segura de HTTP que utiliza cifrado para proteger la información transmitida entre el cliente y el servidor. HTTPS emplea tecnologías de seguridad como SSL/TLS para cifrar los datos, lo que impide que terceros puedan interceptar o modificar la información durante la transmisión. Para funcionar, HTTPS requiere que el servidor tenga un certificado SSL/TLS emitido por una autoridad confiable. Durante la conexión, el navegador verifica el certificado, establece una clave de sesión cifrada y luego intercambia datos de forma segura usando esa clave. HTTPS es fundamental para proteger la privacidad y seguridad en sitios web, especialmente donde se maneja información sensible como datos personales o financieros.

9. 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

RTA Un hosting o alojamiento web es un servicio que permite almacenar los archivos y datos de un sitio web en servidores conectados a internet, haciendo que el sitio sea accesible para los usuarios en línea. Existen varios tipos de hosting, cada uno adecuado para diferentes necesidades y tamaños de proyecto.

**Tipos principales de hosting:**

- Hosting compartido: Varios sitios web comparten los recursos del mismo servidor. Es económico y fácil de usar, ideal para sitios pequeños o en etapa inicial. Sin embargo, el rendimiento puede verse afectado en horas de alta demanda por parte de otros usuarios.
- Hosting VPS (Servidor Privado Virtual): Un servidor físico está particionado en varios servidores virtuales independientes, cada uno con recursos dedicados. Ofrece mayor control, personalización y mejor rendimiento que el compartido.
- Hosting dedicado: El servidor completo está reservado para un solo cliente. Proporciona máximo rendimiento, control y seguridad, indicado para sitios con alto tráfico y necesidades avanzadas.
- Hosting en la nube (Cloud Hosting): Usa una red de servidores virtuales en la nube, lo que permite escalabilidad dinámica y alta disponibilidad. Es ideal para sitios con tráfico variable o que necesitan alta confiabilidad.

| Proveedor                     | Tipo de alojamiento             | Características principales                                  | Mejor uso   |
|-------------------------------|---------------------------------|--|---|
| Bluehost                      | Compartido, VPS, Dedicado       | Fácil integración WordPress, soporte 24/7                    | Pequeños y medianos sitios, blogs, negocios           |
| HostGator                     | Compartido, VPS, Dedicado, Nube | Panel intuitivo, ancho de banda ilimitada, garantías uptime  | Emprendimientos y sitios en crecimiento               |
| SiteGround                    | Compartido, Nube, Dedicado      | Alto rendimiento, seguridad, soporte experto.                | Negocios digitales y comercio electrónico             |
| Servicios web de Amazon (AWS) | Alojamiento en la nube          | Altísima escalabilidad, múltiples servicios cloud integrados | Proyectos grandes, aplicaciones con tráfico variable. |

Nosotros elegimos un Hosting VPS por su equilibrio entre control, rendimiento y escalabilidad. En proveedores, SiteGround es muchas veces una opción sólida por

su soporte y estabilidad, mientras que para proyectos muy grandes o con demandas variables, AWS es preferible por su flexibilidad en la nube incluso si tiene curva de aprendizaje mayor.

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

RTA Un servidor de dominio, conocido también como servidor DNS (Domain Name Server), es un sistema que traduce nombres de dominio legibles para las personas en direcciones IP numéricas que los computadores utilizan para identificar servidores en Internet. Así, cuando se escribe un dominio en un navegador, el servidor DNS busca la dirección IP correspondiente y dirige al usuario al sitio web correcto. Sin esta traducción, se tendría que recordar la dirección IP numérica de cada página, lo que sería complejo.

The screenshot shows the homepage of 'Colombia Hosting'. The header includes contact information for Bogotá, Medellín, Cali, Barranquilla, and other cities, along with a navigation menu with links like Hosting, Dominios, Reseller, VPS, Más servicios, Nosotros, Soporte, and Clientes. The main banner features the text 'Hosting en Colombia Con la mejor y más avanzada Tecnología incluida' and lists various services and technologies like cPanel, Inmunify360, MagicSpam, CloudLinux OS+, LITESPEED, Affinity, MAILCHANNELS, JetBackup, and Inmunify email. Below the banner, there's a section titled 'Algunas razones para elegirnos' with icons for backups, security, and performance. The bottom section is titled 'Planes Hosting en Colombia' and includes a brief description of their hosting services.

The screenshot shows the domain registration page of 'Colombia Hosting'. The header is similar to the homepage. The main section is titled 'Registro de dominios en Colombia' and includes a subtext '¡Compruebe que el dominio deseado esté disponible!'. There is a search bar with the placeholder text 'Compruebe la disponibilidad del dominio...' and a 'BUSCAR' button. Below the search bar, there's a note: 'Por favor ingrese el nombre del dominio que desee verificar sin "www." por ejemplo: dominio.com (dominio.TLD)'. At the bottom, there are logos for various domain extensions: .com, .store, .co, .net, .site, .xyz, .org, .online, and .info.

The screenshot shows the 'Registre su dominio en Colombia' section of the website. It features the heading 'Fácil y rápido' and a paragraph explaining the benefits of having a domain. Below the text, there are three columns of benefits, each with a right-pointing arrow icon. The first column mentions that visitors can easily find the website by entering a simple URL. The second column mentions that it promotes the company's image on the internet. The third column mentions that it facilitates client loyalty and differentiation from competitors. At the bottom left, there is a small notification box that says 'Esteban (Bogotá, Colombia) Se ha unido a Latinoamérica Hosting hace 6 minutos Nivel de satisfacción: Clic aquí >'. On the right side, there are two circular icons: a green one with a speech bubble and a blue one with a speech bubble.

}

.com .store .co .net .site .xyz .org .online .info

## Registre su dominio en Colombia

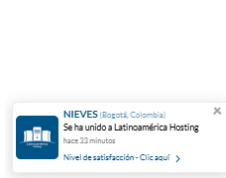
### Fácil y rápido

Contar con un nombre de dominio propio para la web de su proyecto o negocio es altamente beneficioso, ya que le otorgará a su página web una **identidad clara en la red**. Si se encuentra en proceso de **comprar un dominio en Colombia**, es crucial que conozca las ventajas de elegir el dominio adecuado:

- » Los visitantes pueden volver fácilmente a su **sitio web** con solo introducir una dirección URL **sencilla de recordar**.
- » Promueve la imagen de su **marca o empresa** en Internet y le aporta un **valor extra**.
- » Permite un **mayor crecimiento** de su negocio o proyecto, gracias al incremento de tráfico desde los buscadores.
- » Facilita la **fidelización de sus clientes** y la diferenciación con sus competidores del sector.

## ¡Triunfe en Internet con Latinoamérica Hosting!

### Escoja el dominio ideal



#### Fácil de recordar

- ☒ Para adquirir un **nombre de dominio en Colombia**, el primer paso consiste en elegir cuidadosamente un nombre que sea preciso, relevante para su proyecto y fácilmente memorizable.

#### Conozca su público

- ☒ Elija un nombre de dominio que resulte relevante para su **público objetivo**, incorporando palabras clave que estén directamente relacionadas con su **producto o servicio**.



## Tabla de dominios

### Los más económicos del mercado

| Extensión | Registro<br>* IVA incluido | Renovación<br>* IVA incluido | Transferencia<br>* IVA incluido |
|-----------|----------------------------|------------------------------|---------------------------------|
| .com      | \$59.500 anual             | \$59.500 anual               | \$59.500 anual                  |
| .net      | \$58.000 anual             | \$59.500 anual               | \$58.000 anual                  |
| .org      | \$61.500 anual             | \$63.000 anual               | \$61.500 anual                  |
| .co       | \$145.000 anual            | \$152.000 anual              | \$145.000 anual                 |
| .com.co   | \$54.500 anual             | \$54.500 anual               | \$54.500 anual                  |
| .net.co   | \$54.500 anual             | \$54.500 anual               | \$54.500 anual                  |
| .pe       | \$90.000 anual             | \$110.000 anual              | \$110.000 anual                 |
| .com.pe   | \$90.000 anual             | \$110.000 anual              | \$110.000 anual                 |
| .mx       | \$135.000 anual            | \$135.000 anual              | \$135.000 anual                 |
| .com.mx   | \$45.000 anual             | \$80.000 anual               | \$45.000 anual                  |
| .biz      | \$91.500 anual             | \$112.000 anual              | \$91.500 anual                  |
| .info     | \$25.000 anual             | \$154.000 anual              | \$143.000 anual                 |
| .us       | \$46.500 anual             | \$57.500 anual               | \$46.500 anual                  |
| .me       | \$44.000 anual             | \$105.000 anual              | \$100.000 anual                 |
| .shop     | \$12.000 anual             | \$238.000 anual              | \$200.000 anual                 |
| .lol      | \$14.000 anual             | \$160.000 anual              | \$135.000 anual                 |