

	<p style="text-align: center;">UNIVERSIDAD DON BOSCO</p> <p style="text-align: center;">TÉCNICO EN INGENIERÍA EN COMPUTACIÓN (PILET)</p>
<p style="text-align: center;">CICLO I - 2026</p>	<p style="text-align: center;">Investigación Aplicada I: Server Side</p> <p style="text-align: center;">Materia: Desarrollo de Aplic. Web con Soft. Interpretado en el Servidor DSS404 (<i>Teoría</i>)</p>

Estudiantes

Nombre	Código
Lorena Giselle Serrano Majano	SM253027
Anthony Alexander Monterrosa Rivera	MR252990
Carlos Eduardo López Saca	LS242665
Edwin Gabriel Medrano Yanes	MY252997
Elmer Eduardo Rivas Avilés	RA252958

Docente: Ing. Kevin Jiménez

Fecha: Febrero 23, 2026.

Grupo: 01T

Índice

I. Programación del Lado del Servidor (Server-Side).....	3
Definición y concepto fundamental.....	3
Diferencia entre Client-Side y Server-Side.....	3
¿Dónde y cómo se ejecuta el código server-side?.....	4
Rol del servidor en aplicaciones web.....	5
II. Cuadro Comparativo de Lenguajes Server-Side.....	6
III. Referencias bibliográficas.....	8

I. Programación del Lado del Servidor (Server-Side)

Definición y concepto fundamental

La programación del lado del servidor es una tecnología que se encarga de procesar las solicitudes realizadas por los usuarios a través de la interpretación de scripts dentro del servidor web. Su función principal es generar de manera dinámica páginas HTML como respuesta a dichas solicitudes. Todo lo que ocurre dentro del servidor se conoce como procesamiento del lado del servidor o server-side processing. Cuando una aplicación necesita interactuar con el servidor, por ejemplo, para guardar o recuperar información, realiza una petición desde el lado del cliente. Durante este proceso, el servidor ejecuta el código correspondiente para atender la solicitud y enviar la respuesta adecuada.

(WordPress, 2015)

Diferencia entre Client-Side y Server-Side

En el desarrollo web, los términos lado del cliente y lado del servidor se utilizan para indicar el lugar donde se ejecuta el código de una aplicación. El lado del cliente se refiere a todo lo que ocurre en el navegador del usuario, incluyendo la interfaz visual, las interacciones y los efectos que puede ver directamente. En cambio, el lado del servidor corresponde al código que se ejecuta en el servidor web, encargado de procesar solicitudes, generar contenido dinámico, gestionar bases de datos y enviar respuestas al usuario. Aunque estos conceptos suelen asociarse con frontend y backend, no son exactamente lo mismo, ya que describen específicamente la ubicación donde se ejecuta el código dentro de la arquitectura de una aplicación web.

Figura 1

Transmisión de datos



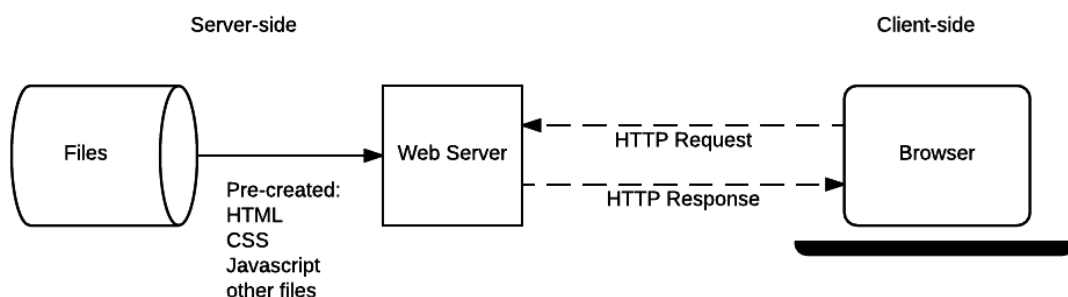
Nota: Transmisión de datos entre lado-cliente y lado-servidor. Fuente: (*¿Qué Significa Lado Del Cliente Y Lado Del Servidor? | Lado Del Cliente Vs. Lado Del Servidor*, n.d.)

¿Dónde y cómo se ejecuta el código server-side?

En un sitio web dinámico, la mayor parte del código necesario para su funcionamiento se ejecuta en el servidor, proceso que se conoce como programación del lado del servidor o *back-end scripting*. Cuando un usuario realiza una solicitud que requiere contenido dinámico, esta es enviada al servidor, donde el sistema interpreta la petición, consulta la base de datos si es necesario, procesa la información y la combina con plantillas HTML para generar una respuesta. Finalmente, el servidor envía al navegador del usuario el contenido ya procesado. Este código puede desarrollarse en diversos lenguajes de programación como PHP, Python, Ruby, C#, o NodeJS, entre otros.

Figura 2

Imagen ilustrativa sobre el funcionamiento de Server-Side



Nota: Imagen ilustrativa sobre el funcionamiento de Server-Side. Fuente: (MDN Web Docs, n.d.)

Rol del servidor en aplicaciones web

Un servidor web es un programa informático encargado de procesar solicitudes provenientes de los clientes, estableciendo conexiones que pueden ser síncronas o asíncronas. Su función principal es generar o proporcionar una respuesta que posteriormente será interpretada y mostrada por el navegador del usuario. Para la transmisión de estos datos se utilizan protocolos de comunicación, siendo el más común el protocolo HTTP, el cual pertenece a la capa de aplicación del modelo OSI. En el contexto de las aplicaciones web, el servidor cumple un papel fundamental al gestionar las peticiones, procesar la información necesaria y enviar el contenido adecuado al cliente. (*Servidor Web*, n.d.)

II. Cuadro Comparativo de Lenguajes Server-Side

Tabla 1

Cuadro comparativo entre las tecnologías “server-side”.

Tecnología	Manejo de HTTP	Tipado	Conexión a Bases de Datos	Rendimiento	Referencia
PHP	Modelo de ejecución síncrono. Cada petición HTTP inicia y termina el proceso del script, liberando la memoria al finalizar..	Dinámico, pero permite tipado gradual. Se pueden declarar tipos de datos en los parámetros y retornos de las funciones.	Utiliza la interfaz PDO para la conectividad universal. El ORM Eloquent es el estándar para manejar bases de datos como objetos.	Medio. Es muy rápido para aplicaciones web tradicionales, especialmente tras la introducción del motor JIT en PHP 8.	(<i>PHP Manual</i> , 2026)
Node.js	Modelo asíncrono basado en un bucle de eventos (<i>Event Loop</i>). Procesa múltiples peticiones en un solo hilo sin bloquear la ejecución.	Dinámico y débil de forma nativa. Se utiliza TypeScript para implementar tipado estático y evitar errores en el desarrollo.	Se comunica mediante Promesas y Async/Await. Los ORMs más usados son Prisma y Sequelize, que facilitan consultas asíncronas.	Muy alto en tareas de entrada/salida de datos (APIs, Chats), pero eficiente bajo en operaciones que saturan el procesador (CPU).	(Node.js Contributors, n.d.)
Python	Utiliza interfaces WSGI (síncrono) o ASGI (asíncrono). FastAPI permite manejar peticiones concurrentes de forma nativa.	Dinámico y fuerte. No requiere declarar el tipo de variable, pero no permite mezclar tipos incompatibles (ej. texto y número) sin conversión.	Basada principalmente en Django ORM o SQLAlchemy. Permite gestionar la base de datos mediante clases de Python de forma muy legible.	Moderado. Es más lento que los lenguajes compilados, pero compensa con una alta velocidad de escritura de código.	(Python Software Foundation, n.d.)
Java	Basado en Servlets y multihilo. Asigna un hilo de ejecución específico para cada petición del usuario, lo que le da mucha estabilidad.	Estático y fuerte. Todos los tipos de datos deben definirse al programar, lo que garantiza que el código sea muy seguro y predecible.	Utiliza el estándar JPA (Java Persistence API) a través de Hibernate. Es excelente para manejar relaciones complejas entre datos.	Muy alto. La máquina virtual (JVM) optimiza el código constantemente mientras la aplicación está corriendo.	(Oracle Corporation, n.d.)
Ruby (Ruby on Rails)	Modelo basado en la interfaz Rack. Es principalmente síncrono y se apoya	Dinámico y fuerte. Prioriza la legibilidad del código (parece lenguaje natural),	Utiliza Active Record. Es el ORM más famoso por su sencillez,	Es menos eficiente en el uso de recursos que Java o .NET, pero muy	(Ruby Documentation Team, n.d.)

	en servidores como Puma para manejar la concurrencia.	aunque requiere pruebas exhaustivas al no tener tipado estático.	permitiendo manipular tablas de bases de datos como si fueran listas de objetos.	eficaz para prototipos y aplicaciones de rápido crecimiento.	
ASP.NET (C#)	Utiliza un modelo de hilos de ejecución vinculados al sistema operativo, lo que lo hace fuerte pero más pesado y menos flexible ante cargas masivas.	Utiliza C# o VB.NET. Es muy rígido, lo que facilita el mantenimiento de aplicaciones empresariales gigantescas que han durado décadas	Se basa en ADO.NET y Entity Framework (versión 6 o inferior). Su integración está optimizada casi exclusivamente para SQL Server y entornos locales.	Es potente para aplicaciones corporativas, pero su arquitectura "monolítica" (todas las librerías cargan a la vez) lo hace más lento al arrancar y consumir recursos.	(<i>ASP.NET Framework Overview</i> , n.d.)
.NET Core	Utiliza el servidor Kestrel. Está diseñado para ser asíncrono, ligero y optimizado para arquitecturas de microservicios modernas.	Estático y fuerte. Ofrece un control riguroso sobre los datos y permite el uso de LINQ para consultar colecciones y bases de datos.	Su estándar es Entity Framework Core, un ORM de alto rendimiento que integra las consultas directamente en el lenguaje C#.	Excelente. Se sitúa entre los frameworks más rápidos del mundo en pruebas de carga y procesamiento masivo de datos.	(<i>ASP.NET Core Fundamentals Overview</i> , 2025)

III. Referencias bibliográficas

- ASP.NET Core fundamentals overview*. (2025, December 16). Microsoft Learn. Retrieved February 21, 2026, from <https://learn.microsoft.com/en-us/aspnet/core/fundamentals/?view=aspnetcore-10.0&>
- ASP.NET Framework Overview*. (n.d.). Microsoft Learn: Build skills that open doors in your career. Retrieved February 21, 2026, from <https://learn.microsoft.com/>
- MDN Web Docs. (n.d.). *Introducción al lado servidor*. Retrieved Febrero 21, 2026, from https://developer.mozilla.org/es/docs/Learn_web_development/Extensions/Server-side/First_steps/Introduction
- Node.js Contributors. (n.d.). *Data Model & Database Support*. Retrieved February 21, 2026, from <https://docs.python.org/>
- Oracle Corporation. (n.d.). *Java Platform, Standard Edition Documentation*. Retrieved February 21, 2026, from <https://docs.oracle.com/en/java/javase/>
- PHP Manual*. (2026, February 5). PHP. Retrieved February 21, 2026, from <https://www.php.net/manual/>
- Python Software Foundation. (n.d.). *Data Model & Database Support*. Retrieved February 21, 2026, from <https://docs.python.org/>
- ¿Qué significa lado del cliente y lado del servidor? | Lado del cliente vs. Lado del servidor*. (n.d.). Cloudflare. Retrieved February 21, 2026, from <https://www.cloudflare.com/es-es/learning/serverless/glossary/client-side-vs-server-side/>
- Ruby Documentation Team. (n.d.). *Object-Oriented Fundamentals*. Retrieved February 21, 2026, from <https://docs.ruby-lang.org/en/>
- Servidor web*. (n.d.). Wikipedia. Retrieved February 21, 2026, from https://es.wikipedia.org/wiki/Servidor_web
- WordPress. (2015, Marzo 16). *Introducción al Lenguaje*. <https://programacion1class.wordpress.com/5-1-introduccion-al-lenguaje/>