

# **Universidad Autónoma de Baja California**

## **Facultad de Ciencias Químicas e Ingeniería**

Plan de Ingeniero en Software y Tecnologías Emergentes



### **Practica 2:**

Tecnologías Esenciales para el Desarrollo Web  
y Configuración de servidores

### **Materia:**

Aplicaciones Web (371)

### **Docente:**

Itzel Barriba Cazares

### **Participante(es):**

Luis Eduardo Galindo Amaya (1274895)

11 de sep de 2024

## Sumario

Instrucciones.....	3
Lenguajes de backend.....	4
Python.....	4
Java.....	5
PHP.....	6
Ruby.....	6
Node.js (JavaScript).....	7
Lenguajes para administración de servidores.....	8
Bash.....	8
Json.....	8
Python.....	8
Xml.....	8
Perl.....	8
Comandos comunes para administraci3n.....	9
Lenguajes de frontend.....	10
HTML (HyperText Markup Language).....	10
Javascript.....	10
CSS.....	11
Protocolos de Seguridad en Servidores.....	12
HTTP (Hypertext Transfer Protocol).....	12
FTP (File Transfer Protocol).....	12
SMTP (Simple Mail Transfer Protocol).....	12
DNS (Domain Name System).....	12
MySQL (Sistema de gesti3n de bases de datos).....	12

# Instrucciones

## 1. Investigación de lenguajes de programación para backend:

- Investiga 5 lenguajes usados en desarrollo backend.
- Puntos a investigar:
  - Sintaxis básica y paradigmas que soportan.
  - Capacidades.
  - Propósito principal en el backend.

## 2. Investigación de lenguajes para administración de servidores:

- Investiga 3 lenguajes o entornos comunes para configuración y gestión de servidores (ej. Bash, Python, Perl).
- Puntos a investigar:
  - Casos de uso en la gestión de servidores.
  - Comandos o scripts comunes.
  - Ventajas y desventajas en sistemas operativos (Linux, Windows).

## 3. Investigación de lenguajes para frontend:

- Puntos a investigar:
  - Función de cada lenguaje en la interfaz de usuario.
  - Ejemplos de código para una página básica, estilo e interacción (botón que muestra mensaje).
  - Limitaciones y desafíos.

## 4. Protocolos de seguridad en servidores:

- Investiga 3 protocolos de seguridad (ej. HTTPS, SSH, SSL/TLS).
- Puntos a investigar:
  - Funcionamiento y relevancia para la seguridad.
  - Situaciones comunes de uso.
  - Comparación de características principales (cifrado, facilidad de implementación).

## 5. Servicios, puertos y protocolos en servidores:

- Investiga 5 servicios comunes en servidores (ej. HTTP, FTP, SMTP, DNS, MySQL).

## Lenguajes de backend

### Python

#### Sintaxis básica y paradigmas

Python es un lenguaje de alto nivel, de sintaxis sencilla y legible. Soporta paradigmas de programación orientada a objetos, funcional e imperativa.

#### Capacidades

Python tiene una gran variedad de bibliotecas y módulos nativos que permiten realizar desde operaciones matemáticas hasta la construcción de servidores web, acceso a bases de datos y manejo de archivos. **De manera nativa python ya incluye herramientas para trabajar la web:**

- **http.server** Biblioteca estándar que permite crear un servidor HTTP básico de manera sencilla.
- **socket** Librería estándar para manejar conexiones de red y crear servidores web desde cero.
- **BaseHTTPRequestHandler** Clase de la biblioteca estándar que permite manejar peticiones HTTP, como GET y POST, de manera manual.

#### Propósito en el backend

Python es comúnmente utilizado para el desarrollo rápido de aplicaciones web, gracias a su facilidad para integrar servidores HTTP, manejar datos y comunicarse con bases de datos. Algunos frameworks comunes para trabajar aplicaciones web con python son:

- **Django** es un framework web de alto nivel en Python que fomenta el desarrollo rápido y un diseño limpio y pragmático. Creado por desarrolladores experimentados, se encarga de muchos aspectos del desarrollo web, permitiendo que te concentres en escribir tu aplicación
- **FastAPI** es un framework web moderno y rápido (de alto rendimiento) para construir APIs con Python, basado en las indicaciones de tipos estándar de Python.
- **Flask** es un framework minimalista escrito en Python que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código.

## Java

### Sintaxis básica y paradigmas

Java es un lenguaje orientado a objetos, tipado estáticamente y con una sintaxis derivada de C++.

### Capacidades

Java es robusto y multiplataforma, se utiliza para aplicaciones de alta escala, con soporte para concurrencia, seguridad, y acceso a redes.

### Propósito en el backend

Java es muy utilizado en grandes empresas para construir aplicaciones escalables y seguras. Es común en aplicaciones empresariales y sistemas distribuidos.

- **Spring** es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java.

## PHP

### Sintaxis básica y paradigmas

PHP es un lenguaje de scripting interpretado, su sintaxis se basa en C y Perl. Soporta programación imperativa y orientada a objetos.

### Capacidades

PHP es especialmente diseñado para el desarrollo web, con soporte nativo para interacción con bases de datos, manejo de formularios y generación dinámica de contenido HTML.

### Propósito en el backend

PHP se utiliza ampliamente en el desarrollo de aplicaciones web debido a su simplicidad y su integración nativa con servidores web como Apache.

- **Laravel** es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti".

## Ruby

### Sintaxis básica y paradigmas

Ruby es un lenguaje interpretado, orientado a objetos, con una sintaxis simple y legible.

### Capacidades

Ruby destaca por su facilidad de uso, flexibilidad y capacidad para crear aplicaciones web rápidamente.

### Propósito en el backend

Utilizado principalmente con el framework Ruby on Rails para construir aplicaciones web rápidas y eficientes, pero el lenguaje puede ser utilizado sin frameworks para tareas del backend.

- **Ruby on Rails**, también conocido como RoR o Rails, es un framework de aplicaciones web de código abierto del lado del servidor escrito en el lenguaje de programación Ruby, siguiendo el paradigma del patrón Modelo Vista Controlador.

## Node.js (JavaScript)

### Sintaxis básica y paradigmas

Node.js permite el uso de JavaScript del lado del servidor, siguiendo el paradigma de programación asíncrona, orientada a eventos y sin bloqueo (non-blocking).

### Capacidades

Manejo de múltiples conexiones simultáneamente, gracias a su modelo basado en eventos. Integra fácilmente APIs, bases de datos y procesamiento en tiempo real.

### Propósito en el backend

Node.js es ideal para aplicaciones web en tiempo real, servicios API, y sistemas de microservicios debido a su capacidad para manejar grandes volúmenes de solicitudes concurrentes.

### Lenguaje del Lado del Cliente y del Servidor

JavaScript se ejecuta en el navegador del usuario, permitiendo crear páginas web interactivas. Todo, desde la manipulación del DOM hasta la validación de formularios, se realiza con JavaScript.

- **Express.js** es el framework más popular y ligero para Node.js. Proporciona una capa mínima para crear aplicaciones web y APIs, ofreciendo mucha flexibilidad y control.

## Lenguajes para administración de servidores

Lenguaje	Uso
<b>Bash</b>	Utilizado principalmente para administración de sistemas y automatización de tareas en sistemas operativos Unix/Linux. Se emplea para ejecutar comandos en la terminal.
<b>Json</b>	Formato de intercambio de datos basado en texto utilizado para enviar datos estructurados entre servidores y clientes web.
<b>Python</b>	Lenguaje de propósito general utilizado en desarrollo web (backend), ciencia de datos, inteligencia artificial, automatización, y administración de sistemas.
<b>Xml</b>	Formato utilizado para la representación y almacenamiento de datos estructurados en documentos. Es común en la configuración de aplicaciones, intercambio de datos entre sistemas y servicios web (SOAP).
<b>Perl</b>	Utilizado principalmente para procesamiento de texto, scripting de sistemas, administración de servidores y automatización. Es famoso por sus poderosas capacidades de manipulación de cadenas y expresiones regulares.



## Comandos comunes para administración

Categoría	Linux	Windows	Descripción
Gestión de usuarios	adduser, useradd, passwd	net user, Set-ADUser	Crear y administrar usuarios en el servidor
Gestión de procesos	ps, top, kill, systemctl	tasklist, taskkill, Get-Process, Stop-Process	Ver, monitorizar y gestionar procesos en el servidor
Gestión de red	ifconfig, ping, netstat, iptables	ipconfig, ping, Test-Connection, netstat	Ver la configuración de red, comprobar conectividad, gestionar puertos y firewall
Gestión de archivos	ls, cp, mv, rm, df, du	dir, copy, move, del, Get-Item	Listar, mover, copiar, y eliminar archivos y directorios
Instalación de paquetes	apt, yum, dnf, dpkg, rpm	Install-Package, Get-WindowsFeature	Instalar y actualizar paquetes y características
Copia de seguridad	rsync, scp, tar	robocopy, xcopy, Backup-SqlDatabase	Copia de archivos entre servidores, sincronización de directorios y backups
Seguridad y firewall	ufw, fail2ban, ssh, htop	Windows Firewall, Get-NetFirewallRule	Configurar firewall y prevenir ataques, acceder remotamente
Supervisión del sistema	free, uptime, dmesg, vmstat	Get-EventLog, Get-Process, Get-Service	Monitorizar memoria, tiempo de actividad, mensajes del sistema
Automatización de scripts	bash, sh	PowerShell	Crear scripts para automatizar tareas repetitivas

## Lenguajes de frontend

### HTML (HyperText Markup Language)

#### Sintaxis básica y paradigmas

HTML usa etiquetas (tags) que definen la estructura del contenido. Cada etiqueta comienza con `<etiqueta>` y termina con `</etiqueta>`. Algunas etiquetas son autosuficientes (`<img>`, `<br>`).

#### Capacidades

Permite organizar el contenido en títulos, párrafos, listas, tablas, etc. Define enlaces para navegar entre páginas, y puede incluir formularios para interactuar con el usuario. Puede incrustar imágenes, videos, audios, y otros objetos multimedia.

#### Limitaciones

No controla el diseño (CSS lo hace) ni el comportamiento (JavaScript lo maneja).

#### Propósito en el frontend

HTML es el esqueleto de las páginas web. Define la estructura básica del contenido que el navegador renderizará para el usuario. Todo el contenido visible y su organización se basa en HTML

## Javascript

#### Sintaxis básica y paradigmas

JavaScript usa una sintaxis basada en variables, funciones, y estructuras de control de flujo. Soporta paradigmas como la programación orientada a objetos, funcional e imperativa.

#### Capacidades

Permite manipular el DOM, manejar eventos de usuario, realizar solicitudes asíncronas al servidor y almacenar datos localmente en el navegador.

#### Limitaciones

Puede ser afectado por diferencias entre navegadores y es susceptible a manipulaciones malintencionadas. Además, su ejecución intensiva puede afectar el rendimiento.

**Propósito en el frontend**

Añadir interactividad y dinamismo, actualizando contenido sin recargar la página, manejando eventos y controlando la lógica del frontend.

**CSS****Sintaxis básica y paradigmas**

CSS utiliza selectores y reglas para aplicar estilos a elementos HTML. Es un lenguaje declarativo que define el aspecto visual del contenido.

**Capacidades**

Define colores, tamaños, espaciado, posiciones y permite crear diseños adaptativos con Flexbox y Grid, además de animaciones y transiciones.

**Limitaciones**

No tiene lógica ni control de flujo y depende del navegador para su correcta interpretación, lo que puede causar problemas de compatibilidad.

**Propósito en el frontend**

Controlar la apariencia visual de la página, definiendo el estilo y diseño de los elementos para mejorar la experiencia del usuario.

## Protocolos de Seguridad en Servidores

### HTTP (Hypertext Transfer Protocol)

Es el protocolo utilizado para la transferencia de páginas web y otros recursos a través de la web. Los servidores HTTP como Apache o Nginx permiten la entrega de contenido web a los navegadores de los usuarios.

**Puerto:** 80 (HTTP) y 443 (HTTPS para conexiones seguras).

### FTP (File Transfer Protocol)

Permite la transferencia de archivos entre un servidor y un cliente. Es comúnmente utilizado para cargar y descargar archivos en servidores de manera eficiente.

**Puerto:** 21 (control) y 20 (transferencia de datos).

### SMTP (Simple Mail Transfer Protocol)

Es utilizado para enviar correos electrónicos entre servidores de correo y desde un cliente a un servidor de correo. Trabaja junto con otros protocolos como IMAP o POP3 para la entrega y recepción de correos.

**Puerto:** 25 (predeterminado), 465 (SMTP seguro con SSL/TLS).

### DNS (Domain Name System)

Traducir nombres de dominio legibles por humanos (como [www.ejemplo.com](http://www.ejemplo.com)) a direcciones IP que las máquinas pueden entender. Es un servicio esencial para la navegación por Internet.

**Puerto:** 53

### MySQL (Sistema de gestión de bases de datos)

Provee un sistema de base de datos relacional para almacenar y gestionar datos. MySQL se utiliza comúnmente en aplicaciones web para almacenar datos estructurados, como usuarios, publicaciones o productos.

**Puerto:** 3306

## Fuentes

- Ahmed, K. (n.d.). *The 5 Best Backend Development Languages to Master (2024)*. roadmap.sh. <https://roadmap.sh/backend/languages>
- Schooner, N. (2022, November 10). The 8 top back-end Programming languages for 2023. *Boot.dev Blog*. <https://blog.boot.dev/backend/best-backend-programming-languages/>
- Hedau, S. (2024, August 20). Best Backend Languages for Web Development | The 2024 List. *Softspace Solutions*. <https://softspacesolutions.com/blog/best-backend-languages-for-web-development/>
- Esemé, S. (2023, April 24). *Mastering backend*. Mastering Backend. <https://masteringbackend.com/posts/top-5-backend-programming-languages/>
- Stamp, M. (2024, July 22). *Top 100 comandos Linux que debes conocer*. Guías Para Sitios Web, Tips & Conocimiento. <https://www.dreamhost.com/blog/es/comandos-linux-que-debes-conocer/>
- Petrov, V. (2023, February 16). *Trucos y comandos esenciales para administradores de servidores Linux*. Veselin. <https://veselin.es/trucos-y-comandos-esenciales-para-administradores-de-servidores-linux/>
- Corbalán, A. S. (2022, April 8). Los 50 mejores Comandos Linux del Shell Bash que debes conocer - Antonio Sánchez Corbalán. *Antonio Sánchez Corbalán*. <https://sanchezcorbalan.es/mejores-comandos-linux-bash/>