1. Introducción a la Arquitectura Cliente-Servidor

La arquitectura cliente-servidor describe cómo un cliente solicita servicios a un servidor mediante protocolos como HTTP o HTTPS. El cliente muestra la interfaz al usuario, mientras que el servidor procesa peticiones, ejecuta lógica y accede a bases de datos.

2. Desarrollo del Lado del Cliente (Frontend)

El desarrollo del lado del cliente (frontend) incluye HTML para estructura, CSS para diseño y JavaScript para interactividad. Frameworks como React, Vue o Angular permiten crear interfaces dinámicas. El frontend se enfoca en experiencia de usuario, diseño responsivo y comunicación con APIs.

3. Programación del Lado del Servidor (Backend)

La programación del lado del servidor (backend) gestiona la lógica interna de una aplicación. Usa lenguajes como PHP, Node.js, Python o Java. El backend procesa datos, maneja autenticación, se conecta a bases de datos y genera contenido dinámico o APIs.

4. Manejo de Sesiones y Autenticación

El manejo de sesiones permite almacenar información temporal del usuario, como su estado autenticado. Las cookies guardan datos en el navegador. La autenticación verifica identidad (ej. usuario/contraseña, OAuth, JWT), mientras que la autorización define permisos.

5. Gestión de Bases de Datos en el Lado del Servidor

La gestión de bases de datos incluye sistemas SQL (MySQL, PostgreSQL) y NoSQL (MongoDB, Redis). Desde el servidor, se accede mediante SQL directo o usando ORMs. Las bases de datos permiten almacenar y consultar información persistente.

6. APIs y Servicios Web

Las APIs permiten comunicación entre aplicaciones. Las APIs REST usan métodos HTTP como GET, POST, PUT y DELETE y normalmente devuelven JSON. Existen también SOAP, WebSockets y GraphQL. La seguridad se maneja mediante tokens, OAuth o API keys.

Mapa Conceptual: Arquitectura Cliente-Servidor

Arquitectura Cliente-Servidor:

- Cliente: interfaz, peticiones, navegador.

- Servidor: procesa lógica, accede a BD, responde.

- Comunicación: HTTP/HTTPS, TCP/IP, SSL/TLS.

- Ciclo: solicitud → proceso → respuesta.

Mapa Conceptual: Desarrollo del Lado del Cliente (Frontend)

Frontend:

- HTML: estructura.

- CSS: diseño responsivo.

- JavaScript: interacción.

- Frameworks: React, Vue, Angular.

- Herramientas: Tailwind, Webpack, Git.

Mapa Conceptual: Backend

Backend:

- Lenguajes: PHP, Node.js, Python.

- Funciones: lógica, bases de datos, autenticación.

- Frameworks: Laravel, Express, Django.

- Respuestas: HTML, JSON, archivos.

Mapa Conceptual: Sesiones y Autenticación

Sesiones y Autenticación:

- Sesiones: datos temporales.

- Cookies: datos en navegador.

- Autenticación: identidad (JWT, OAuth).

- Autorización: permisos.

- Seguridad: HTTPS, hashing, CSRF, XSS.

Mapa Conceptual: Bases de Datos

Bases de Datos:

- SQL: MySQL, PostgreSQL.

- NoSQL: MongoDB.

- Acceso: SQL, ORM.

- Diseño: claves, normalización.

- CRUD, Joins, vistas.

Mapa Conceptual: APIs y Servicios Web

APIs:

- Tipos: REST, SOAP, WebSockets.

- Formatos: JSON, XML.

- Seguridad: API Keys, JWT, OAuth.

- Herramientas: Axios, Postman, Swagger.