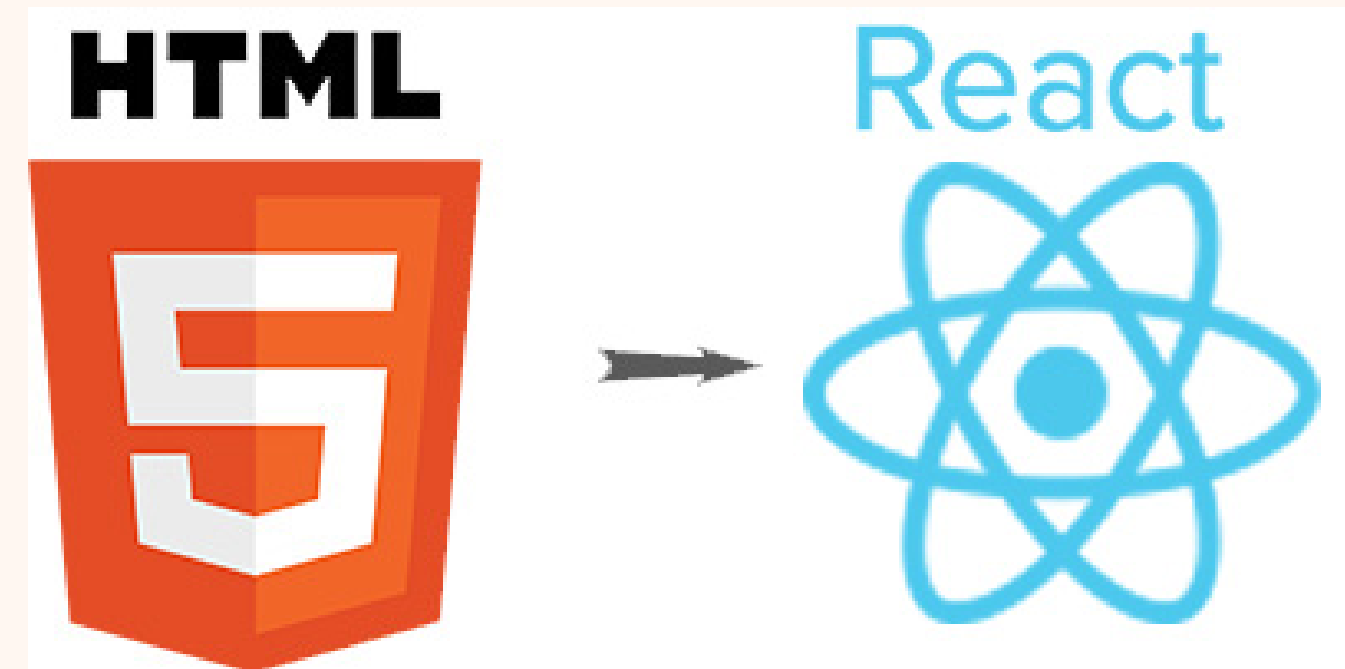
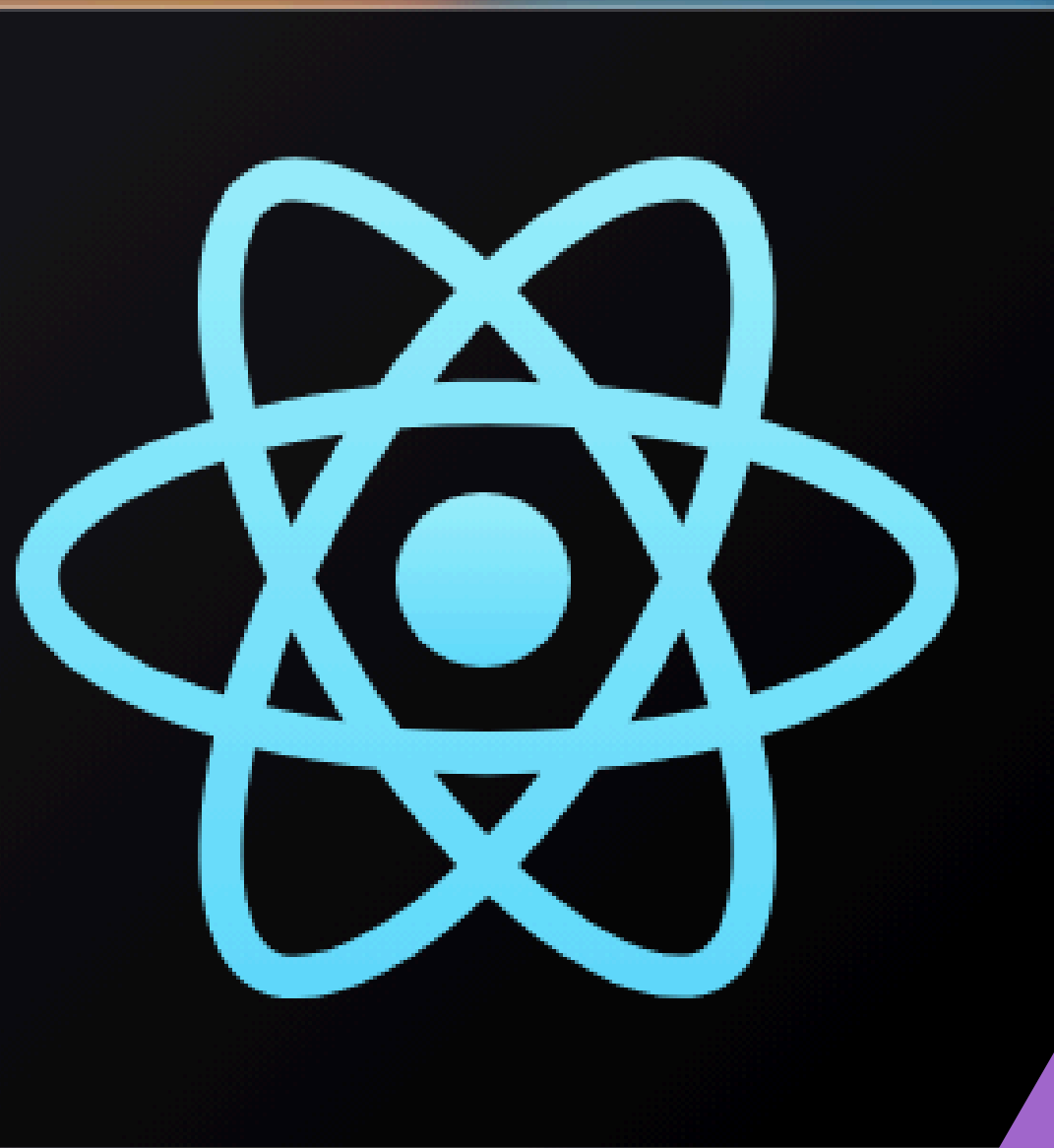


PROGRAMACION WEB



Docente: Josue Israel Vasquez Martínez

1.2 Arquitectura de las aplicaciones web

1 **Modelo Cliente-Servidor:**

- Cliente: Navegador web que solicita recursos (HTML, CSS, JavaScript).
- Servidor: Procesa las solicitudes y envía respuestas (backend).

2 **Arquitectura de tres capas**

- Capa de presentación: Interfaz de usuario (frontend).
- Capa de lógica de negocio: Reglas y procesos (backend).
- Capa de datos: Almacenamiento y gestión de información (bases de datos).

3 **Arquitecturas modernas**

- SPA (Single Page Application): Carga una sola página y actualiza dinámicamente.
- Microservicios: Divide la aplicación en servicios independientes.



1.3 TECNOLOGÍAS PARA EL DESARROLLO DE APLICACIONES WEB

Las tecnologías se dividen en dos áreas principales: frontend y backend.

1. Frontend:

- Lenguajes: HTML, CSS, JavaScript.
- Frameworks: React, Angular, Vue.js, Svelte.
- Herramientas: Webpack, Babel, Sass.

2. Backend:

- Lenguajes: Python, PHP, Java, Ruby, Node.js.
- Frameworks: Django, Laravel, Spring, Express.js.
- Bases de datos: MySQL, PostgreSQL.
- Bases de datos No Relacionales: MongoDB, GraphQL

3. Otras tecnologías:

- APIs RESTful: Comunicación entre frontend y backend.
- WebSockets: Comunicación en tiempo real.
- DevOps: Docker, Kubernetes, CI/CD.



ACTIVIDAD

Investigar y comparar:

- Dos frameworks de frontend
- Dos de backend



1.4 PLANIFICACIÓN DE APLICACIONES WEB

La planificación es crucial para el éxito de un proyecto web. Incluye:

1. Análisis de requisitos:
 - Identificar necesidades del cliente y usuarios finales.
 - Definir funcionalidades y objetivos.
2. Diseño:
 - Crear wireframes y prototipos.
 - Definir la arquitectura y tecnologías a utilizar.
3. Desarrollo:
 - Implementar frontend y backend.
 - Integrar bases de datos y APIs.
4. Pruebas:
 - Realizar pruebas unitarias, de integración y de usabilidad.
5. Despliegue y mantenimiento:
 - Publicar la aplicación en un servidor.
 - Monitorear y realizar actualizaciones.



1.4 EJEMPLO DE PLANIFICACIÓN PARA CREAR UN SITIO WEB

Proyecto: Crear un sitio web para una tienda online de ropa.

1. Análisis de Requisitos

En esta fase, se definen los objetivos del sitio web y las necesidades del cliente y los usuarios.

Preguntas Clave:

- ¿Cuál es el propósito del sitio web?
 - Vender ropa en línea y mostrar catálogos de productos.
- ¿Quiénes son los usuarios principales?
 - Hombres y mujeres de 18 a 45 años interesados en moda.
- ¿Qué funcionalidades son esenciales?
 - Catálogo de productos, carrito de compras, pasarela de pago, registro de usuarios, blog de moda.
- ¿Qué tecnologías se usarán?
 - Frontend: HTML, CSS, JavaScript (React).
 - Backend: Node.js con Express.
 - Base de datos: MongoDB.

Entregables:

- Documento de requisitos con:
 - Objetivos del sitio web.
 - Funcionalidades prioritarias.
 - Público objetivo.

1.4 EJEMPLO DE PLANIFICACIÓN PARA CREAR UN SITIO WEB

2. Diseño

En esta fase, se crea la estructura y el diseño visual del sitio web.

Actividades:

1. Wireframes:

- Dibujar esquemas básicos de las páginas principales (inicio, catálogo, carrito, blog).
- Herramientas: Figma, Adobe XD o papel y lápiz.

2. Diseño Visual:

- Crear el diseño de la interfaz (colores, tipografía, imágenes).
- Ejemplo: Diseño moderno con colores pastel y tipografía limpia.

3. Arquitectura de la Información:

- Organizar el contenido del sitio web (menús, categorías, enlaces).
- Ejemplo:
 - Inicio
 - Catálogo (Hombres, Mujeres, Niños)
 - Blog
 - Contacto

Entregables:

- Wireframes y diseños visuales aprobados.
- Estructura de navegación del sitio web.

1.4 EJEMPLO DE PLANIFICACIÓN PARA CREAR UN SITIO WEB

3. Desarrollo

En esta fase, se construye el sitio web siguiendo los diseños y requisitos.

Actividades:

1. Frontend:

- Crear la interfaz de usuario usando HTML, CSS y JavaScript.
- Implementar interactividad con React (por ejemplo, filtros de búsqueda en el catálogo).

2. Backend:

- Configurar el servidor con Node.js y Express.
- Crear APIs para manejar solicitudes (por ejemplo, obtener productos, procesar pagos).

3. Base de Datos:

- Diseñar la base de datos en MongoDB.
- Crear colecciones para productos, usuarios y pedidos.

4. Integración:

- Conectar el frontend con el backend usando APIs.
- Ejemplo: El frontend solicita la lista de productos al backend, que consulta la base de datos y devuelve la información.

Entregables:

- Sitio web funcional con todas las páginas y funcionalidades implementadas.

1.4 EJEMPLO DE PLANIFICACIÓN PARA CREAR UN SITIO WEB

4. Pruebas

En esta fase, se verifica que el sitio web funcione correctamente y esté libre de errores.

Actividades:

1. Pruebas de Funcionalidad:

- Verificar que todas las funcionalidades trabajen correctamente (por ejemplo, agregar productos al carrito, procesar pagos).

2. Pruebas de Usabilidad:

- Asegurarse de que el sitio sea fácil de usar y navegar.
- Ejemplo: Probar con usuarios reales y recoger feedback.

3. Pruebas de Rendimiento:

- Verificar que el sitio cargue rápidamente y maneje múltiples usuarios.
- Herramientas: Google PageSpeed Insights, Lighthouse.

4. Pruebas de Seguridad:

- Asegurarse de que los datos de los usuarios estén protegidos.
- Ejemplo: Implementar HTTPS y validar entradas de usuario.

Entregables:

- Reporte de pruebas con errores corregidos.
- Sitio web listo para su lanzamiento.

1.4 EJEMPLO DE PLANIFICACIÓN PARA CREAR UN SITIO WEB

5. Despliegue y Mantenimiento

En esta fase, el sitio web se publica y se asegura su correcto funcionamiento a largo plazo.

Actividades:

1. Despliegue:

- Publicar el sitio web en un servidor (por ejemplo, usando AWS, Google Cloud o un hosting compartido).
- Configurar el dominio (por ejemplo, www.tiendaropa.com).

2. Mantenimiento:

- Monitorear el sitio web para detectar errores o caídas.
- Actualizar el contenido (por ejemplo, agregar nuevos productos o publicar entradas en el blog).
- Realizar copias de seguridad periódicas.

Entregables:

- Sitio web en vivo y accesible para los usuarios.
- Plan de mantenimiento continuo.

RESUMEN DEL PLAN DE PROYECTO SITIO WEB PARA UNA TIENDA ONLINE DE ROPA.

Fase	Actividades	Entregables
Análisis de Requisitos	Definir objetivos, funcionalidades y tecnologías.	Documento de requisitos.
Diseño	Crear wireframes, diseño visual y arquitectura de la información.	Wireframes y diseños aprobados.
Desarrollo	Implementar frontend, backend, base de datos e integración.	Sitio web funcional.
Pruebas	Realizar pruebas de funcionalidad, usabilidad, rendimiento y seguridad.	Reporte de pruebas y errores corregidos.
Despliegue y Mantenimiento	Publicar el sitio web y monitorear su funcionamiento.	Sitio web en vivo y plan de mantenimiento.

CONSEJOS PARA UNA BUENA PLANIFICACIÓN

1. Comunica con el cliente: Asegúrate de entender bien sus necesidades y expectativas.
2. Establece plazos realistas: Divide el proyecto en etapas y asigna tiempos para cada una.
3. Uso de una Metodología:
4. Usa herramientas de gestión:
5. Prueba constantemente: No dejes las pruebas para el final; hazlas en cada fase del desarrollo.

HTML

HTML (HyperText Markup Language) es un lenguaje compuesto por un grupo de etiquetas definidas con un nombre rodeado de paréntesis angulares.

```
index.html">Home</a></li>  
ome-events.html">Home Events</li>  
ulti-col-menu.html">Multiple C  
s-children"> <a href="#" class=  
a href="tall-button-header.htm  
a href="image-logo.html">Image  
lass="active"><a href="tall-l  
as-children"> <a href="#"  
<a href="variable-width  
<a href="testimonial-s  
<a href="featured-work  
<a href="equal-column-  
<a href="video-slider  
<a href="mini-bootstr
```

Modelo de cajas

