

TAREA ACADÉMICA

1. Elaborar una monografía del tema asignado siguiendo la estructura establecida en la parte inferior.
 - a. Html Básico
 - b. Html Avanzado GRUPO1
 - c. CSS Básico GRUPO 2
 - d. CSS Avanzado GRUPO 3
 - e. CSS con Bootstrap GRUPO 4
 - f. CSS con Tailwind GRUPO 4
 - g. JavaScript Básico GRUPO 5
 - h. JavaScript Avanzado GRUPO 6
 - i. TypeScript
2. Exposición de los temas.
 - a. 10 min de teoría
 - b. 10 min de ejemplos
 - c. 10 min de preguntas
3. Estructura de la monografía.

1. Portada

- Título de la monografía
- Nombre del autor o autores
- Institución
- Fecha de presentación

2. Índice o tabla de contenido

- Listado organizado de las secciones y subsecciones con números de página correspondientes.

3. Introducción

- Presentación del tema.
- Objetivo general y específicos del trabajo.
- Justificación o importancia del tema.
- Metodología utilizada (opcional).

4. Desarrollo o cuerpo principal

- **10 secciones temáticas:** Aquí se organiza toda la información de la última versión de manera lógica y progresiva datos respaldados por fuentes confiables.
- Ejemplos de código por capítulo o secciones en detalle.

5. Conclusión

- Resumen de los puntos más importantes.



- Reflexión que cambios se producirá en el futuro y que hay por investigar.

6. Referencias bibliográficas

- Listado de las fuentes consultadas, siguiendo un formato de citación IEEE.

4. Temas detallados.

REQUISITO: Temas de HTML básico:

1. Introducción al HTML y estructura básica de un documento.
2. Etiquetas principales (<html>, <head>, <body>).
3. Formateo de texto: <p>, <h1> a <h6>, , ,
, <hr>.
4. Listas: , , .
5. Imágenes: y atributos principales como src y alt.
6. Enlaces: <a> y sus atributos como href y target.
7. Tablas: <table>, <tr>, <td>, <th>, y atributos relacionados.
8. Formularios básicos: <form>, <input>, <label>.
9. Comentarios en HTML (<!-- -->).
10. Estructura semántica básica: <header>, <footer>, <main>, <section>, <article>.

Temas de HTML avanzado:

1. Elementos y atributos globales (data-*, id, class, style, title).
2. Uso de template y slot para contenido dinámico.
3. Elementos multimedia: y con sus atributos.
4. Integración de SVG y Canvas para gráficos.
5. Formularios avanzados: validación y tipos de entrada (, , etc.).
6. API de HTML5: Drag and Drop, GeoLocation, Web Storage (LocalStorage y SessionStorage).
7. Microdatos y marcado semántico avanzado.
8. Integración con JavaScript: eventos y manipulación del DOM.
9. Técnicas de accesibilidad y etiquetado (aria-*).
10. Buenas prácticas de SEO en HTML.

CSS Básico

1. Selectores simples y compuestos y Especificidad
2. Propiedades de Texto y Fuentes
3. Modelo de Caja (Box Model)
4. Colores y Fondos
5. Unidades de Medida (px, %, em, rem, vh, vw)
6. Posicionamiento (static, relative, absolute, fixed, sticky)
7. Flexbox (Diseño flexible)
8. Grid Layout (Diseño de cuadrícula)
9. Pseudo-clases y Pseudo-elementos



10. Transiciones y Animaciones básicas

CSS Avanzado

1. Variables CSS y Custom Properties
2. Funciones CSS (calc(), clamp(), min(), max())
3. Grid Layout avanzado (Áreas de grid, auto-fit, auto-fill)
4. Flexbox avanzado (Alineaciones, distribución dinámica)
5. Diseño responsivo (Media Queries y Mobile-first)
6. Scroll Snap y Técnicas de Scroll
7. Clipping y Masking con CSS
8. Propiedades y Contenedores CSS (Container Queries)
9. Filtros y Efectos Visuales avanzados
10. Técnicas de Performance y Optimización en CSS

Temas: Bootstrap

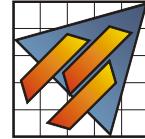
1. Sistema de Grid (Diseño responsivo con columnas)
2. Componentes predefinidos (Botones, tarjetas, modales, alertas)
3. Tipografía y estilos de texto personalizables
4. Formularios y validaciones integradas
5. Clases de utilidad para espaciado, alineación y visibilidad
6. Personalización con variables CSS y SASS
7. Animaciones y transiciones integradas
8. Modificadores de color y estilos temáticos
9. Diseño responsivo avanzado con media queries
10. Integración con JavaScript (Tooltips, Dropdowns, Carousels)

Tailwind CSS

1. Concepto de clases utilitarias (Atomic CSS)
2. Diseño responsivo con sm, md, lg, xl
3. Personalización de colores y estilos con configuración tailwind.config.js
4. Espaciado, tipografía y bordes con clases rápidas (m-4, text-lg, border)
5. Flexbox y Grid usando clases (flex, grid, gap)
6. Dark Mode y variantes de estado (hover, focus, active)
7. Optimización de rendimiento con PurgeCSS
8. Animaciones y efectos (transition, transform, shadow)
9. Componentización con @apply y combinación de estilos
10. Integración con frameworks como React, Vue y Next.js

Temas de JavaScript Básico

1. Introducción a JavaScript y su ejecución en navegadores
2. Variables y tipos de datos (String, Number, Boolean, Arrays, Objects)
3. Operadores (aritméticos, lógicos, comparación)
4. Condicionales (if, else, switch)



-
-
5. Bucles (for, while, do while)
 6. Funciones (declaración, expresiones, parámetros, retorno)
 7. Manipulación del **DOM** (seleccionar elementos, modificar contenido, eventos)
 8. Eventos y manejadores (addEventListener, onclick, etc.)
 9. JSON y almacenamiento local (localStorage, sessionStorage)
 10. Depuración y uso de la consola (console.log, debugger)

Temas de JavaScript Avanzado

1. **Canvas** y gráficos (getContext, dibujar formas, animaciones)
2. Programación orientada a objetos (clases, this, prototype)
3. Manipulación avanzada del DOM (creación de elementos dinámicos, atributos, estilos)
4. Promesas y async/await (manejo de asincronía)
5. Fetch API y AJAX (consumo de APIs)
6. WebSockets (comunicación en tiempo real)
7. Closures y scope (contexto de ejecución)
8. Gestión de errores (try, catch, throw)
9. Web Workers y optimización de rendimiento

Temas de TypeScript

1. **Introducción a TypeScript:** Ventajas sobre JavaScript, instalación y configuración.
2. **Tipos de Datos:** string, number, boolean, array, tuple, enum, any, void, never.
3. **Interfaces y Tipos Personalizados:** Definición de estructuras de datos con interface y type.
4. **Clases y Programación Orientada a Objetos:** Herencia, modificadores de acceso, abstract, static.
5. **Funciones Tipadas:** Tipado de parámetros y retornos, sobrecarga de funciones.
6. **Genéricos:** Creación de componentes reutilizables con T y extends.
7. **Manipulación del DOM con TypeScript:** Uso de document.querySelector, eventos y tipado de elementos.
8. **Módulos y Espaciado de Nombres:** Importación/exportación de módulos y organización de código.
9. **Manejo de errores y try/catch:** Uso de excepciones, Error y validaciones en TypeScript.
10. **Integración con Frameworks:** Uso de TypeScript con React, Angular, Vue.js.