

TAREA LMSGI 02

MIGUEL SÁNCHEZ MARTÍNEZ

Lenguajes de marcas y sistemas de gestión de la información

ASIR 24/25

Contenido

[Ejercicio01: 3](#_Toc185826459)

[HTML: 3](#_Toc185826460)

[CSS: 4](#_Toc185826461)

# Ejercicio01:

## HTML:

En la etiqueta <head>, lo mas destacable es un script que nos permite, mediante CDN, acceder a la biblioteca de iconos de la web fontawesome.com para poder de manera sencilla mostrar los que necesitemos en el sitio.

Usamos etiqueta header para delimitar la cabecera de la pagina, en ella ubicaremos el logo de la pagina con información muy relevante. En este caso, por diseño, ubicamos una etiqueta aside que contiene el menú de navegación en la parte inferior del header, dentro del mismo se muestra una lista donde cada elemento será un elemento del menú, que albergara un enlace a distintas paginas de la web, de momento todos redirigen a la pagina principal.

Texto

Descripción generada automáticamente

En el body de la pagina, ubicamos 1 seccion que contiene el titulo del plato, una lista con los ingredientes, una imagen descriptiva y una lista con los pasos a seguir, todo ello ubicado en contenedores div para organizarlo a modo de tarjetas.

Captura de pantalla de computadora

Descripción generada automáticamente

Por ultimo tenemos un footer donde mostramos los derechos de autor y el logo de Cidead.

Captura de pantalla de computadora

Descripción generada automáticamente

## CSS:

He marcado el CSS por clase, por lo que el identificador sea un punto. La primera etiqueta, por costumbre suele ser la que afecta a toda la pagina, es este caso lo mas relevante es la propiedad *box-sizing: border-box* la cual indica que no sume los anchos de padding y borde al total de la caja.

Pasamos a la siguiente etiqueta, voy a nombrar solo las propiedades relevantes, la etiqueta title da estilos al titulo de la pagina, lo mas relevante de esta etiqueta son las propiedades border y border radius que nos marcan el borde y la propiedad box-shadow que añade una pequeña sombra al cotenedor.

En la etiqueta menú, marcamos le diseño del menú, marcamos display: flex y lo disponemos en row para que se muestre en horizontal, marcando justyfy-content: space-arounrd para que ocupe todo el ancho. Al tratarse de una lista, marcamos la propiedad list-style: none para que no se muestre ningún identificador de elemento de lista ( ).

En el body indicamos el tipo de letra, color y el color del fondo.

Captura de pantalla de computadora

Descripción generada automáticamente

Pasamos a la zona principal de la pagina, donde disponemos el contenido en tarjetas, así que daremos estilo a cada una de ellas,

Lo mas relevante de esta zona son las etiquetas grid-template-colums, donde le indicamos a display-grid: column la cantidad de columnas que queremos, marcamos también un max-width para limitar el ancho de la tarjeta.

Captura de pantalla de computadora

Descripción generada automáticamente

Por ultimo damos estilos al footer de la pagina, la intención es que se muestre en todas las paginas del sitio.

Captura de pantalla de computadora

Descripción generada automáticamente

# Ejercicio02:

Para este ejercicio he decidido, puesto que dispogo de un fichero .json con toda mi información, hacer uso del mismo y que extraiga los datos del mismo de manera dinámica con javascript, así pues cualquier añadido de información se agrega en formato adecuado al fichero y se mostrara en la web de manera dinámica, con los estilos adecuados.

## HTML:

En el <head> como parte relevante he añadido una etiqueta <script> con el cdn de fontawesome para obtener los iconos de la tabla de contacto.

En el <header> de la pagina hemos situado un <div> que contiene la cabecera del sitio

Ya dentro del <body> hemos situado una etiqueta <aside> que contendrá el menú de navegación en la parte izquierda del sitio y siempre esta fijo en esa posición, los elementos del menú nos dirigen a las distintas secciones del sitio.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Pasamos seguidamente a las distintas secciones del portfolio, en esta parte es donde he hecho uso de javascript para acceder dinámicamente al fichero Json y crear una parte de la pagina en base a la información del fichero.

En cada etiqueta <section> disponemos de un id, el cual le indica al menú de navegación que parte del sitio es para poder navegar hasta ella.

A continuación tenemos una etiqueta <h1> que nos mostrara el titulo de la sección y acto seguido una etiqueta <div> con un id, esta es la que contendrá el contenido creado dinámicamente a raíz del fichero json, mediante código javascript.

Captura de pantalla de computadora

Descripción generada automáticamente

Seuidamente pasamos a la ultima sección de la web, donde mostramos los datos de contacto y redes sociales mediante una tabla, siguiendo la misma estructura que el resto de la pagina pero sin contenido dinámico. Tenemos una etiqueta <table> que nos mostrara la estructura de una tabla seguidamente de unas etiquetas <tr> que nos indican las filas de la tabla, en primer lugar una etiqueta <th> que nos indica la cabecera de la tabla y a continuación una etiqueta <td> con los datos de la misma. La ultima etiqueta <td> continene un <div> con la finalidad de “dividir” en 3 la celda y mostrar 3 iconos que harán de link a distintas redes sociales, los iconos se muestran gracias al CDN de fontawesome descrito en el <head> de la pagina, mediante la clase mostramos el icono.

Para finalizar tenemos la etiqueta <script> que accede al fichero .js con el código javascritp, se coloca al final del body para que acceda cuando el DOM este completamente cargado, para así poder ubicar bien el código HTML generado.

Captura de pantalla de computadora

Descripción generada automáticamente

Vamos a describir el fichero script.js, el cual se encarga de crear dinámicamente contenido HTML e incrustarlo en la web junto con los datos del fichero .Json.

La línea 1 indica que este evento se dispara cuando el documento HTML ha sido completamente cargado y analizado, sin esperar a que las hojas de estilo e imágenes terminen de cargarse. Esto asegura que el script se ejecute solo después de que el DOM esté listo.

Seguidamente la función “fetch” accede a los datos del fichero .json y los asigna a una variable.

La línea 6 accede al elemento con el id “aboutInfo” y carga los datos solicitados del .json.

En las siguientes 3 secciones (Estudios, Experiencia y Proyectos) se sigue la misma estructura, accedemos al contenedor con el id descrito, crea un elemento <article> y le asigna una clase “project” la cual le dará unos estilos generales para las 3 secciones, acto seguido incrusta dentro del <article> el código html descrito con las clases para darle estilos.

Por ultimo accedemos al método “appendChild” para incrustar el código html.

En la parte final se muestra un código para que en caso de error, nos lo muestre por consola.