

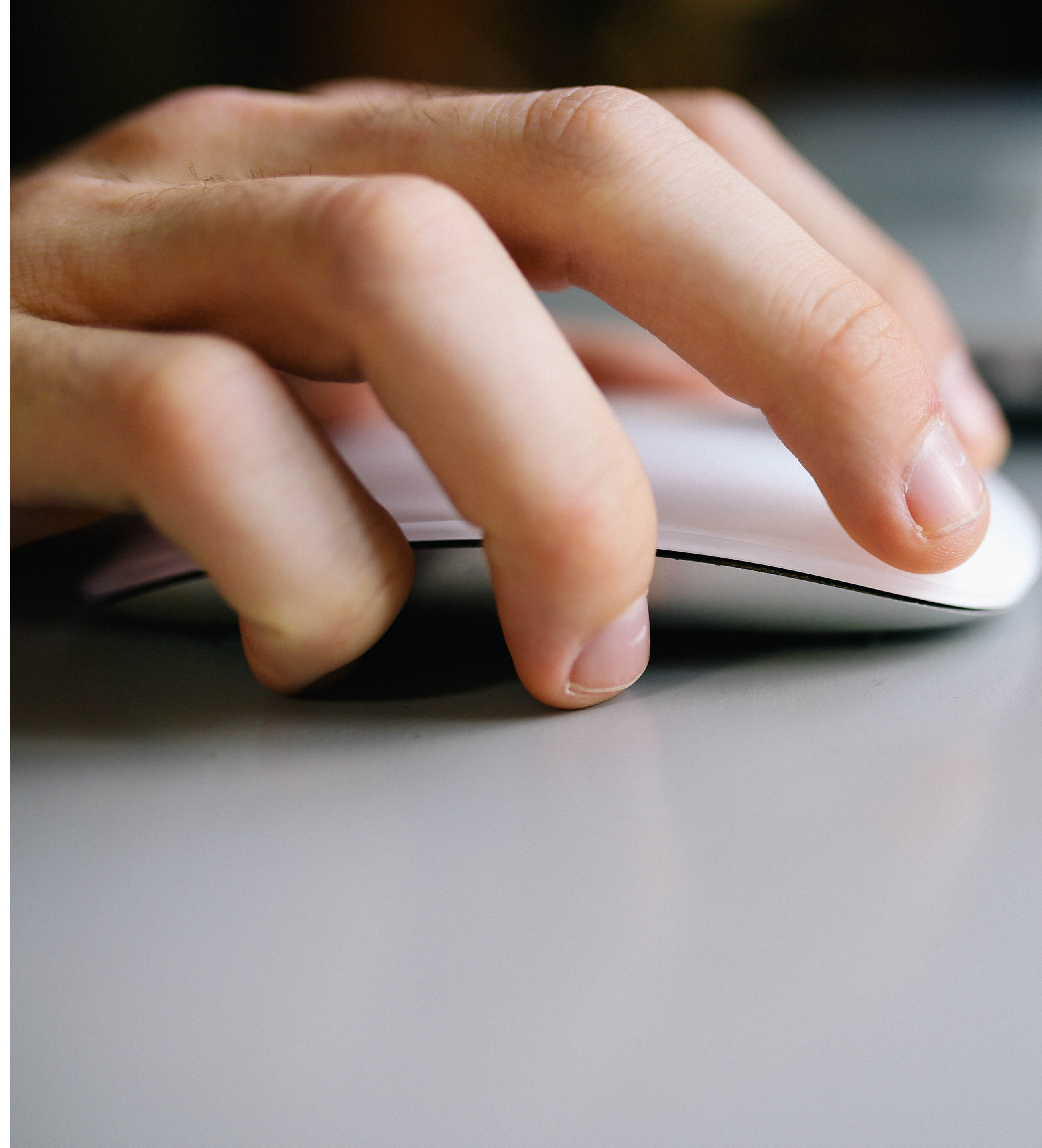
Desarrollo web



Desarrollo web front-end —

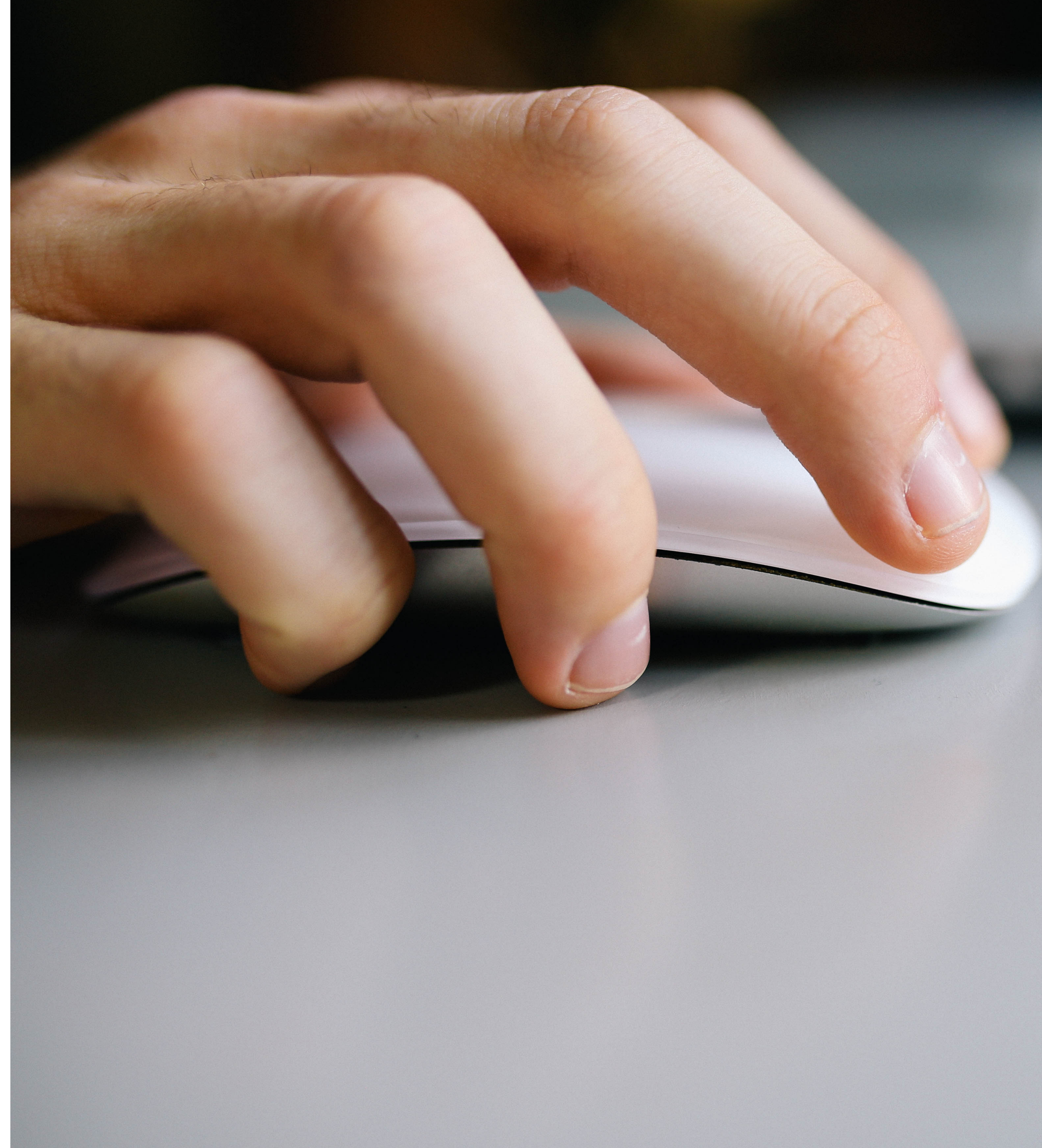
Dirigido a aquellos que quieren aprender a diseñar y construir sitios web desde cero.

Las únicas herramientas son una computadora con un navegador web y un editor de texto.



Agenda —

- Introducción
- HTML
- CSS
- JavaScript
- Librerías JavaScript
- Herramientas de depuración y pruebas

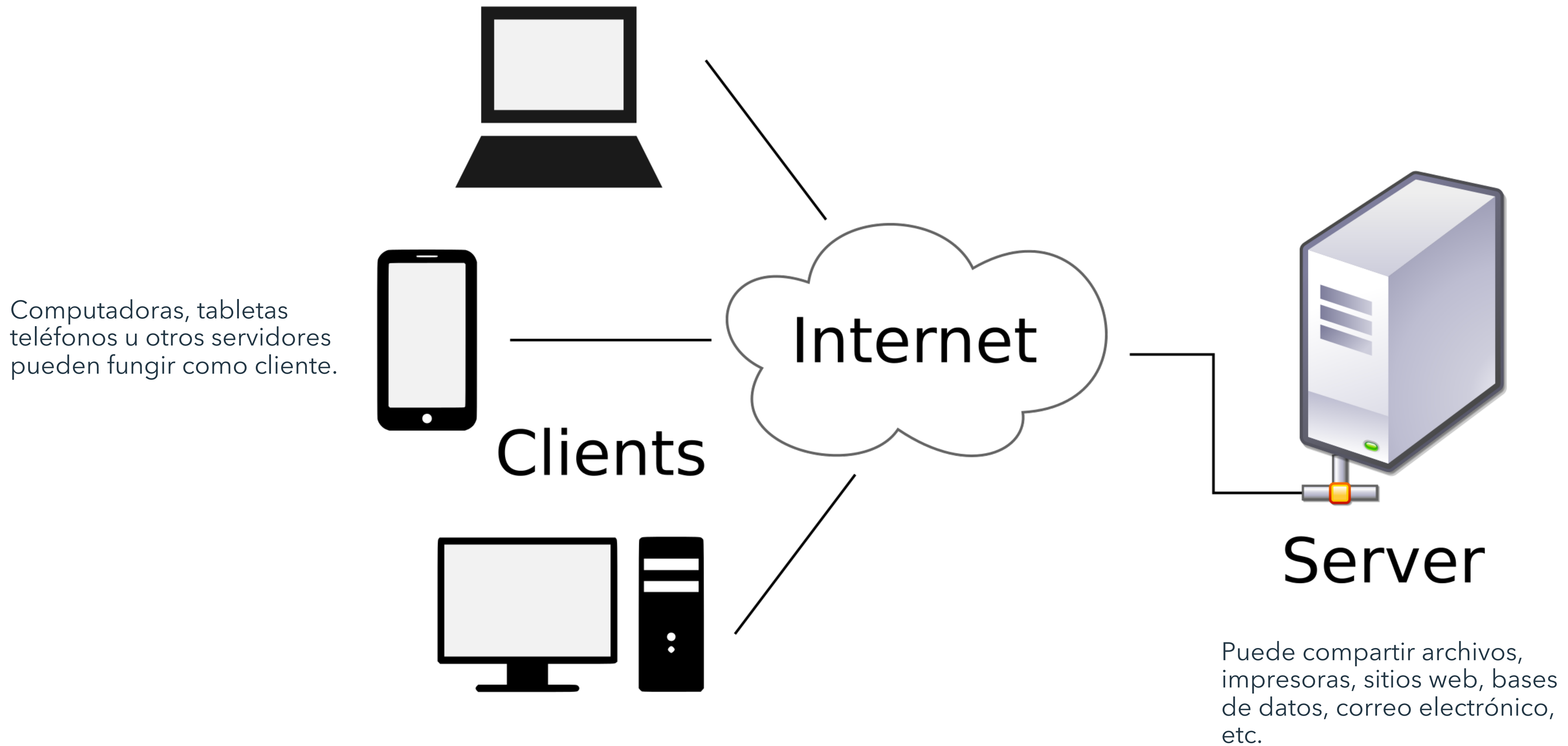


Introducción



Arquitectura cliente - servidor —

Es un **modelo de diseño de software** en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes.



Cómo accedemos a la web —

NAVEGADOR WEB

Las personas acceden a los sitios web utilizando un software llamado navegador web.

SERVIDOR WEB

Cuando navegas en internet, las páginas web se solicitan a una computadora especial, el servidor web.

CLIENTES

Accedemos a la web en una gama cada vez mayor de dispositivos que incluyen computadoras, tabletas y teléfonos móviles.

Cómo se crea un sitio web —

LO QUE VES

Al visitar un sitio web, el navegador web recibe **HTML** y **CSS** para crear lo que ves; **JavaScript**, para añadir mejoras en la interfaz de usuario; además de **audio** y **vídeo**.

CÓMO SE CREA

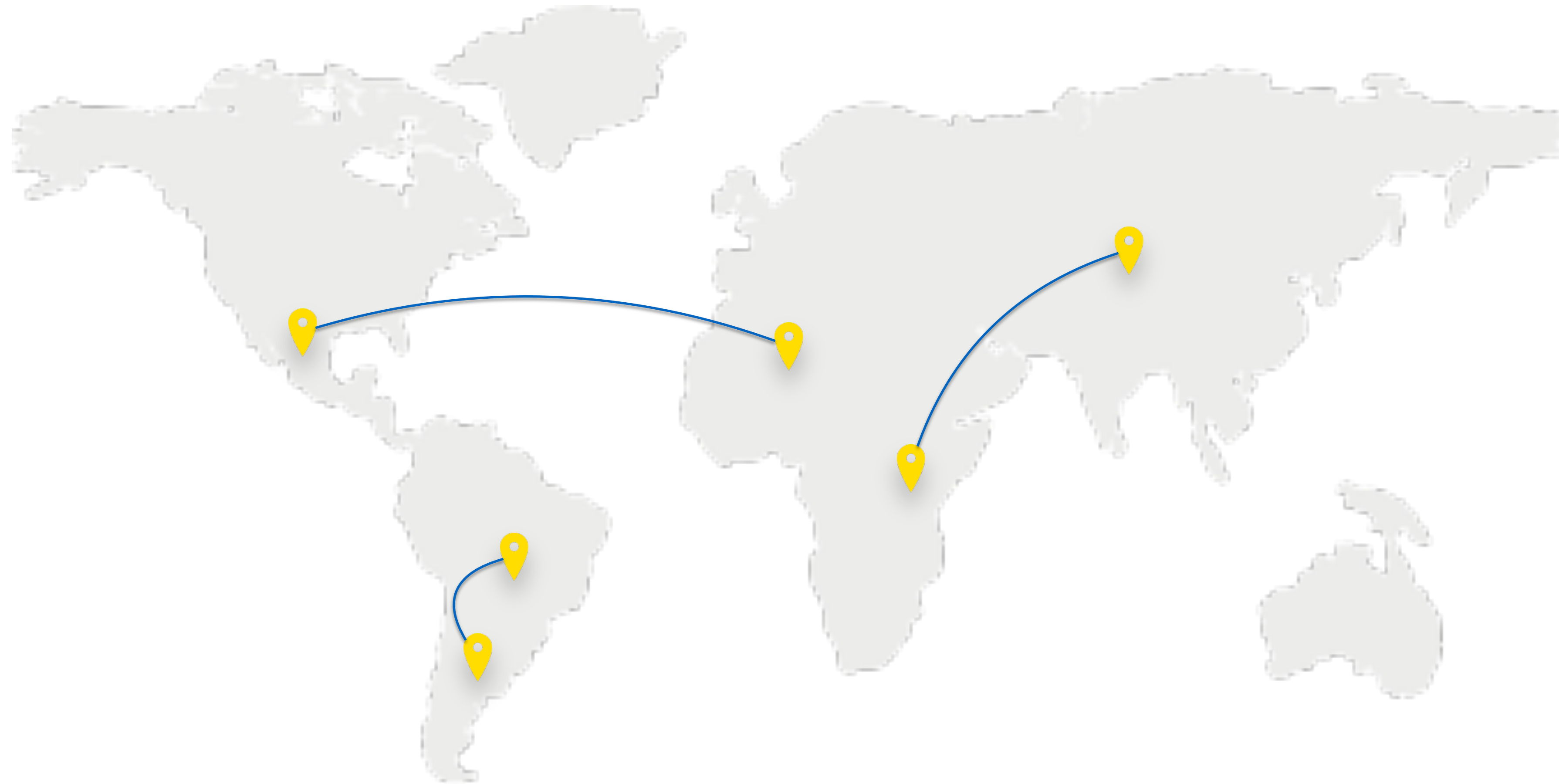
Todos los sitios web usan HTML y CSS, pero los sitios web de redes sociales y las plataformas de comercio electrónico agregan más tecnologías como **PHP** o **Java**.

DISEÑO DE SOFTWARE

Es necesario razonar el propósito del sitio web para definir los **elementos de software** que lo conforman y las **relaciones** entre ellos.

CÓMO FUNCIONA —

1. Escribes el **nombre de dominio**.
2. El navegador web contacta una red de servidores (**DNS**) para obtener la IP asociada.
3. La **IP** es entregada al navegador web para contactar al servidor web que la aloja.
4. El servidor web regresa la **página web** solicitada.



Programación por capas —

Es una arquitectura cliente - servidor en la que las funciones de presentación, procesamiento de aplicaciones y administración de datos están físicamente separadas. El uso más extendido de esta arquitectura es la arquitectura de tres niveles **Modelo - Vista - Controlador** (MVC).

Arquitectura MVC —

CAPA DE PRESENTACIÓN

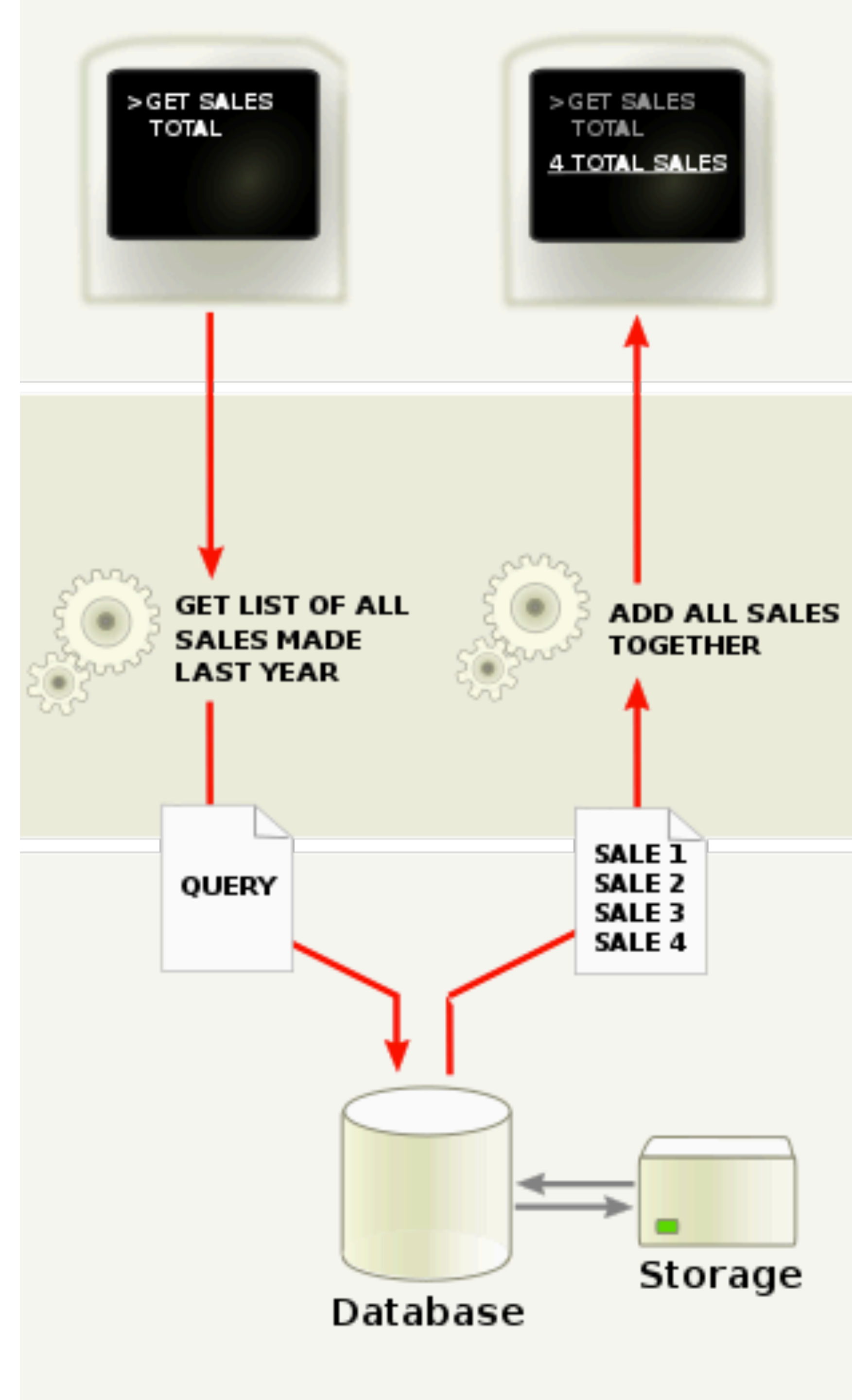
Proporciona una representación del modelo.

CAPA LÓGICA

Cambia el estado del modelo y coordina la aplicación según la petición del usuario.

CAPA DE DATOS

Mantiene la información, estado y lógica de la aplicación.



Front-end y back-end —

El **front-end** de una aplicación (capa de presentación) es lo que un humano ve y con lo que éste puede interactuar, encargado de traducir la interacción del usuario a las especificaciones del back-end.

El **back-end** de una aplicación (capas lógica y de datos) trabaja tras bambalinas haciendo útil el front-end. Recibe y otorga respuestas a las peticiones del usuario.

¿Qué se espera de un desarrollador? —

FRONT-END

Representar mediante HTML y CSS el diseño de una página web.

Mejorar la experiencia de usuario con JavaScript.

Seguir estándares W3C.

BACK-END

Coordinación de una página web con lenguajes de programación de servidor y bases de datos.

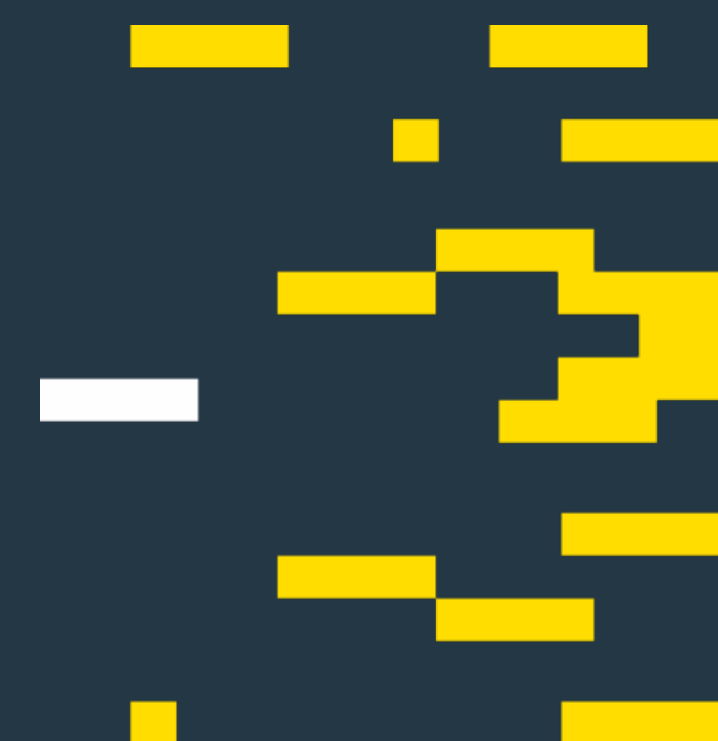
FULL STACK

Conocimiento entre las dos facetas previas.

Capacidad de asumir tareas de diferente índole.

Hacer la vida más fácil para el usuario.

HTML



Estructura

Para cualquier tipo de documento, la estructura es muy importante para ayudar a los lectores a entender el mensaje.

Como en un periódico, las páginas web suelen tener un título, algún texto, y posiblemente imágenes, audio o vídeo.

Estructura HTML

```
<html>
  <body>
    <h1>Título</h1>
    <h2>Subtítulo</h2>
    <p>
      Lorem ipsum...
    </p>
  </body>
</html>
```

Título

Subtítulo

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Etiquetas

APERTURA

`<p>`

Los caracteres entre paréntesis indican el propósito de la etiqueta.

CIERRE

`</p>`

Se caracteriza por una diagonal.

Una elemento HTML comprende la etiqueta de apertura y de cierre.

ATRIBUTOS

`<p lang = "es">`

Proveen información adicional.

Siempre en la etiqueta de apertura.

Se componen de nombre y valor.

Etiquetas relevantes —

- `<html>`
- `<head>`
- `<title>`
- `<body>`



Texto —

Cuando se crea una página web, se utilizan etiquetas (marcas) adicionales para agregar contenido a las páginas.

Para facilitar la lectura del código, se suelen agregar saltos de línea y espacios adicionales. Sin embargo, el navegador web suele hacer un colapso del espacio en blanco.

Marcado

ESTRUCTURAL

Son los elementos que se puede usar para describir tanto los títulos como los párrafos

SEMÁNTICO

Proporciona información adicional; como cuando se pone énfasis en una oración, en una cita, etc.

Etiquetas relevantes —

- `<h1>` - `<h6>`
- `<p>`
- `
`
- `<hr />`



Listas —

ORDENADAS

``

Listas donde cada elemento de la lista está numerado.

DESORDENADAS

``

Listas que utilizan una viñeta en lugar de un número.

DEFINICIÓN

`<dl>`

Se componen de un conjunto de términos junto con sus definiciones.

Etiquetas relevantes —

- ``
- `<dt>`
- `<dd>`



Enlaces

Los enlaces son una característica importante de la web porque permiten pasar de una página web a otra, lo que permite la idea de navegar.

Nos permiten navegar a otra página web del mismo sitio u otro, incluso a otra parte de la misma página. Pueden ser configurados para abrir el enlace en otra pestaña, o abrir un gestor de correo electrónico y escribirle a alguien.

Escribiendo enlaces —

ETIQUETA

`<a>`

Los enlaces son creados utilizando la etiqueta **<a>**.

CONTENIDO

`<a>Clic aquí`

El usuario podrá hacer clic en el contenido de la etiqueta.

ATRIBUTO HREF

``

Se especifica el destino del enlace utilizando el atributo **href**.

Uniform Resource Locator (URL) —

ABSOLUTA

Empieza con el nombre de dominio para el sitio, y puede continuar con la ruta hacia una página específica.

RELATIVA

Son utilizadas para enlazar a una página del mismo sitio. No requieren el nombre de dominio para hacer el enlace.

URL Relativa —

MISMA CARPETA

Sólo utilizar el nombre del recurso.

SUBCARPETA

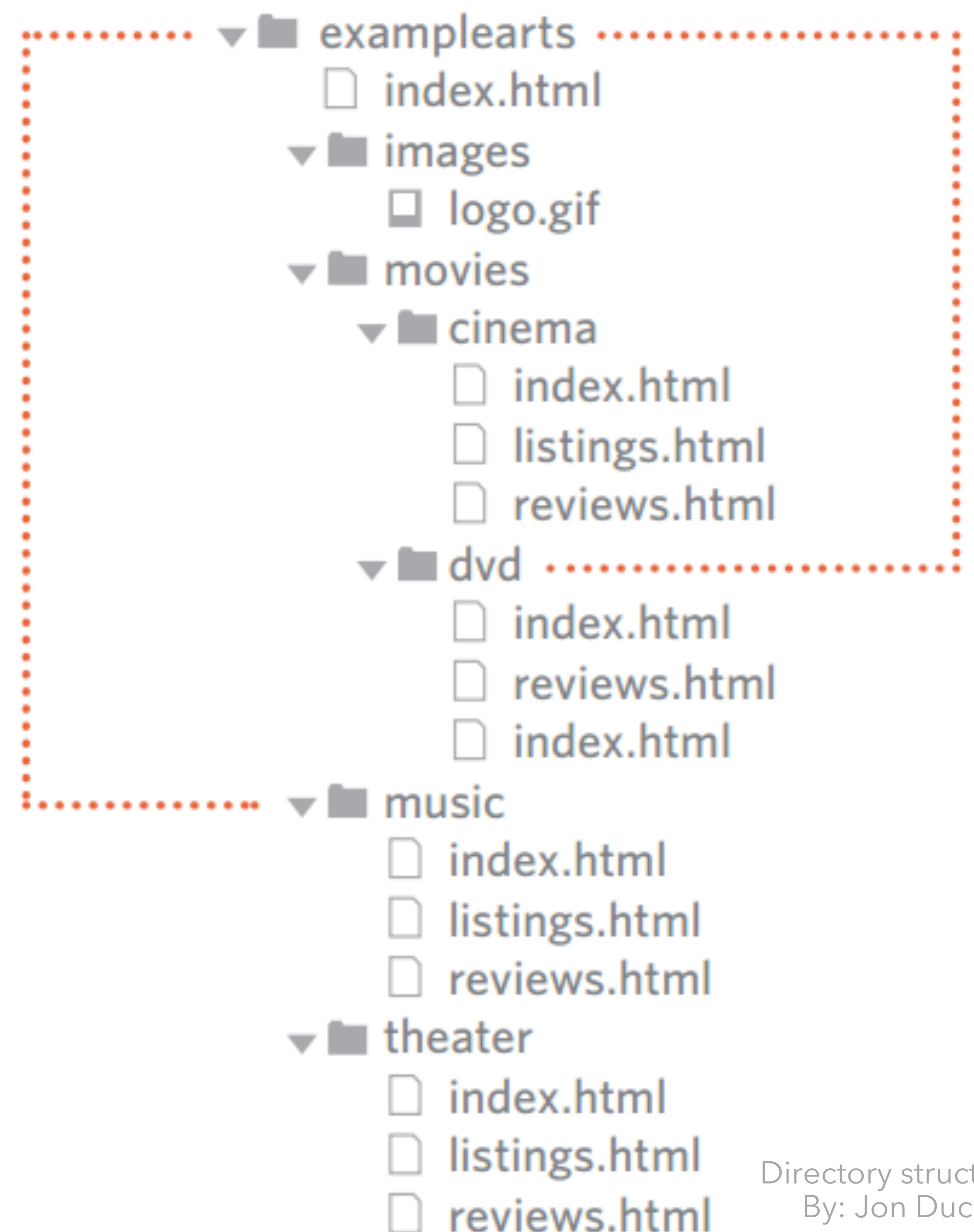
Utilizar el nombre de la carpeta, seguida por una diagonal y, finalmente, el nombre del archivo.

CARPETA CONTENEDORA

Utilizar `../` para indicar la carpeta superior de la actual, después el nombre del archivo.

CARPETA PRINCIPAL

Utilizar `/` para apuntar a raíz.



Funciones relevantes —

- href="mailto:ex@3ct.mx"
- target=" "
- href="#top"
- href="http://3ct.mx#top"



Imágenes —

Hay muchas razones por las que una imagen deba ser añadida a una página web: un logo, una fotografía, un diagrama o una gráfica.

Tres reglas para imágenes:

- Formato adecuado
(JPEG, + colores; GIF, - colores; PNG, transparencias; SVG, logotipos)
- Correcto tamaño
- Correcta resolución (72 ppi máximo)

Añadiendo imágenes —

ETIQUETA

``

Para añadir una imagen se requiere de la etiqueta ****.

CONTENIDO

``

El atributo **src** le dice al navegador web dónde encontrar la imagen. Generalmente, en el propio sitio.

Tablas

Algunos tipos de información requieren ser presentados en una tabla. Por ejemplo, los reportes que pueden presentar una gran cantidad de datos complejos.

Etiquetas relevantes —

- table
- tr
- th
- td



Formularios

Tradicionalmente, se comprende como un formulario un documento que contiene espacios para proporcionar información.

HTML ofrece este concepto de formulario para recolectar información de los usuarios del sitio.

Protocolo HTTP —

HTTP

El protocolo **Hypertext Transfer Protocol** (HTTP) permite las transferencias de información en la World Wide Web (WWW).

MÉTODO GET

- Datos anexados a la URL.
- Predeterminado.
- Hasta 2 000 caracteres.

MÉTODO POST

- Datos anexados a la solicitud HTTP.
- No hay límites de tamaño.
- Permite carga de archivos.

Métodos de petición —

/test/demo_form.asp?name1=value1&name2=value2

POST /test/demo_form.asp HTTP/1.1

Host: w3schools.com

name1=value1&name2=value2

Estructura de formularios —

DÓNDE

```
<form action="server.do">
```

El atributo **action** se utiliza para definir a dónde enviar la información.

CÓMO

```
<form method="get">
```

Mediante el atributo **method** se indica cómo se enviará la información.

Cómo funciona —

INFORMACIÓN

username="John"

La información viaja en parejas **clave - valor**.

CLAVE

<input name="username" />

Cada elemento de formulario debe tener al atributo **name** que representa la clave.

VALOR

El valor será otorgado por el usuario, o por valores definidos por el desarrollador (atributo **value**).

Controles de texto —

TEXTO

```
<input type="text" />
```

Se usa para una sola línea de texto, como direcciones de correo electrónico y nombres.

CONTRASEÑA

```
<input type="password" />
```

Recolecta información del usuario, pero no la muestra.

ÁREA DE TEXTO

```
<textarea>
```

Para grandes áreas de texto multilínea.

Controles de selección —

BOTONES DE RADIO

```
<input type="radio" />
```

Para cuando un usuario debe seleccionar una de muchas opciones.

CASILLAS DE VERIFICACIÓN

```
<input type="checkbox" />
```

Para cuando un usuario debe seleccionar y deseleccionar una o más opciones.

LISTAS DESPLEGABLES

```
<select>
```

Para cuando un usuario debe seleccionar una de muchas opciones de una lista.

Otros controles

BOTÓN DE ENVIAR

```
<input type="submit">
```

Para cuando se quiere enviar la información recolectada al servidor.

ARCHIVOS

```
<input type="file" />
```

Permite a los usuarios enviar archivos al servidor.

CSS



Selectores CSS —

Cascading Style Sheets (CSS) tiene como propósito crear páginas web más atractivas controlando el diseño de éstas.

CSS permite crear reglas para especificar cómo el contenido de un elemento debe ser mostrado.

Reglas de estilo —

SELECTOR

p { ... }

El **selector** indica a qué elemento aplica la regla.

REGLAS

p { color: "red" }

La **regla** indica cómo el elemento referido debe ser mostrado.

PROPIEDADES

color: "red"

La **propiedad** indica qué aspecto de los elementos se van a cambiar.

El **valor** es la configuración usada para la propiedad.

CSS Externo —

ETIQUETA

`<link />`

La etiqueta **<link>** puede ser usada dentro de un documento HTML para importar un archivo CSS.

ATRIBUTO *rel*

`rel="stylesheet"`

El atributo **rel** especifica la relación entre la página HTML y el archivo enlazado.

ATRIBUTO *href*

`href="..."`

href especifica la ruta al archivo CSS.

CSS Interno —

ETIQUETA

`<style>`

También se puede especificar reglas CSS colocándolas dentro de la etiqueta **<style>**.

RECOMENDACIÓN

Mantener el código CSS en el mismo archivo HTML es una mala práctica.

Se recomienda utilizar un archivo CSS externo.

Selectores CSS —

UNIVERSAL

* { ... }

Aplica a **todos** los elementos en el documento.

TIPO

p { ... }

Aplica a todos los **elementos** de la etiqueta indicada.

CLASE

.title { ... }

Aplica a cualquier elemento cuyo atributo **class** tiene un valor que coincide con el especificado.

Selectores CSS —

IDENTIFICADOR

#name { ... }

Aplica a un elemento cuyo atributo **id** coincide con el especificado.

DESCENDIENTE

div a { ... }

Aplica a un elemento que es **descendiente** de otro.

MULTISELECTOR

h1, .h2, #h3 { ... }

Aplica a todos los elementos listados.

Por qué Hojas de Estilo en Cascada —

ÚLTIMA REGLA

Si dos selectores son idénticos, el último de los dos prevalecerá.

ESPECIFICIDAD

Si un selector es más específico que otro, éste tendrá mayor jerarquía.

IMPORTANCIA

Añadir **!important** al final de una regla indicará que ésta es más importante que otras.

HERENCIA

Algunas propiedades aplicadas en elementos padres serán heredadas por sus hijos y otras no.

Se puede inducir este comportamiento con **inherit**.

Color

color

color = "red"

La propiedad **color** permite especificar el color de texto de un elemento.

background-color

background-color = "red"

Puede indicar el color de fondo del elemento mediante **background-color**.

Color

VALORES RGB

`rgb(100, 100, 90)`

Expresa el color en términos de qué tanto rojo, verde y azul.

El rango aceptable es 0 - 255.

CÓDIGO HEXA

`#EE3E80`

Es un código de seis dígitos que representan colores RGB cada dos dígitos.

El rango es 0 - 9 y A - F.

NOMBRES

`red`

Aproximadamente 150 colores diferentes de los cuales elegir.

Texto —

FAMILIA

font-family

Especifica las posibles fuentes de un texto.

Si el navegador no soporta primera, elegirá una segunda, etc.

PESO

font-weight

Permite añadir énfasis a un texto.

Valores como **light** y **bold** son válidos.

ESTILO

font-style

Especifica el estilo de un texto.

Valores como **normal** e **italic** son válidos.

Texto —

TRANSFORMACIÓN

text-transform

Se usa para cambiar a mayúsculas o minúsculas un texto.

Valores como **uppercase**, **lowercase** y **capitalize** son válidos.

DECORACIÓN

text-decoration

Permite añadir un estilo adicional al texto.

Valores como **none**, **underline** y **line-through** son válidos.

ALTURA

line-height

Indica el espacio vertical entre las líneas de texto. Éste se especifica en unidades como **pixeles**, **porcentajes** o **ems**.

Texto —

ALINEACIÓN

`text-align`

Permite controlar la alineación horizontal de un texto.

Valores como **left**, **right**, **center** y **justify** son válidos.

ALINEACIÓN

`vertical-align`

Permite controlar la alineación vertical de un texto.

Valores como **bottom**, **middle** y **top** son válidos.

Selectores relevantes —

- :first-letter
- :first-line
- :link
- :visited
- :hover
- :active
- :focus



Cajas

ANCHO

`width`

La propiedad **width** permite especificar el ancho de un elemento.

Éste se especifica en unidades como **pixeles**, **porcentajes** o **ems**.

ALTO

`height`

La propiedad **height** permite especificar el ancho de un elemento.

Cajas

BORDE

border

Cada caja tiene un borde, incluso si éste no se ve.

Éste se especifica en unidades como **pixeles**, **porcentajes** o **ems**.

MARGEN

margin

Se refiere al espacio ente un elemento y otro a partir del borde.

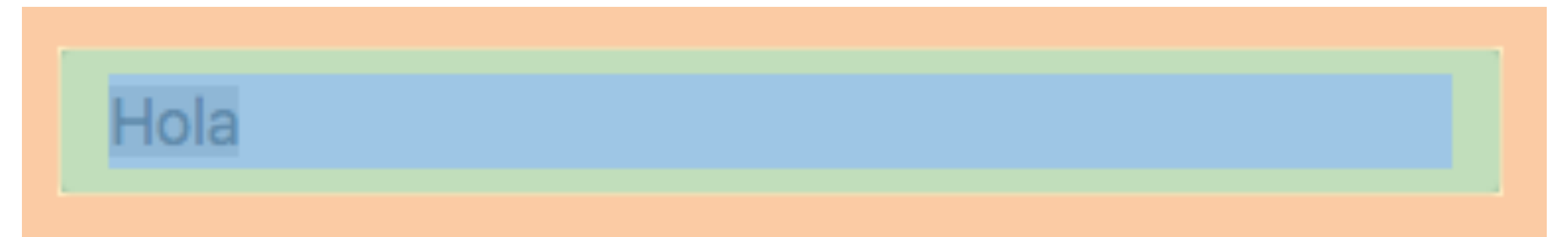
MARGEN INTERNO

padding

Se refiere al espacio entre el contenido de la caja y el borde.

Cajas

- content
- background-color
- padding
- border
- margin



Visibilidad —

DESPLEGAR

display

Controla el comportamiento de la caja.

Valores como **inline** y **block** son válidos.

ALINEACIÓN

vertical-align

Permite controlar la alineación vertical de un texto.

Valores como **bottom**, **middle** y **top** son válidos.

Visibilidad —

EN LÍNEA

`display: inline`

Permite que un elemento no tenga margen y comparta la línea donde se encuentra con otros.

EN BLOQUE

`display: block`

Obliga a que el margen de un elemento crezca tanto como el elemento padre. **Impide** a que más un elemento se muestre en la misma línea.

NINGUNA

`display: none`

El elemento no se mostrará.

Propiedades relevantes —

- border-width
- border-style
- border-color
- box-shadow
- Border-radius

