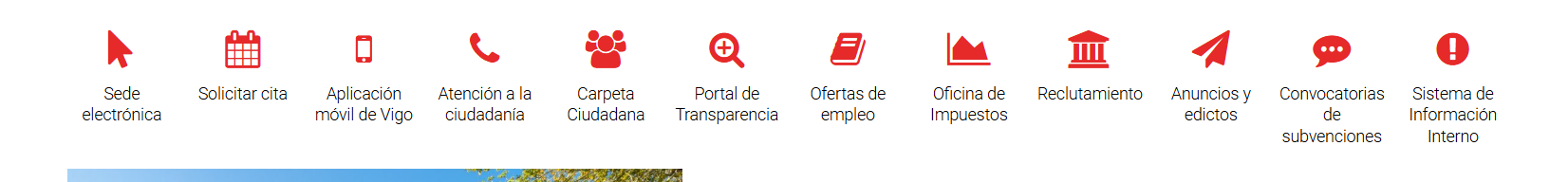
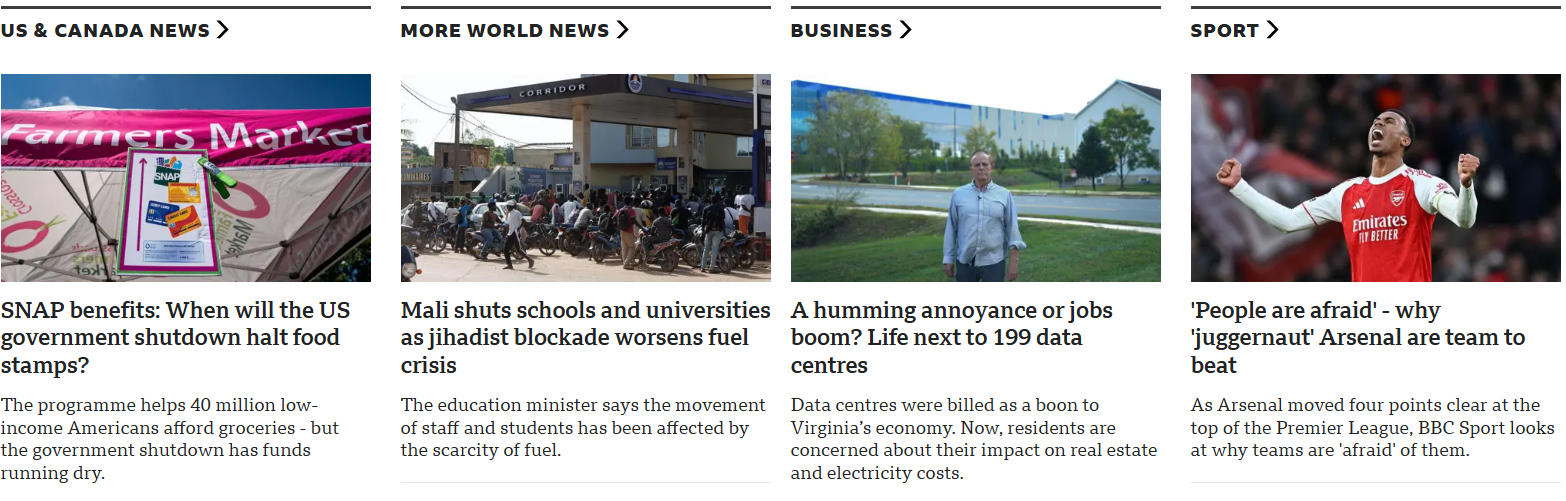
# Fases de diseñar desde 0

## Fase 1:

1. Replicar un sitio hecho
2. Modificar las funciones
3. Creamos un patrón









**Colores a utilizar:**



**Tipografía:**

Monserrat:

<style>

@import url('https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100..900;1,100..900&display=swap');

</style>

Lato:

<style>

@import url('https://fonts.googleapis.com/css2?family=Lato:ital,wght@0,100;0,300;0,400;0,700;0,900;1,100;1,300;1,400;1,700;1,900&family=Montserrat:ital,wght@0,100..900;1,100..900&display=swap');

</style>

**Barra navegación con dropdown:**

<nav class="navbar navbar-expand-lg bg-body-tertiary">

<div class="container-fluid">

<a class="navbar-brand" href="#">Navbar scroll</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarScroll" aria-controls="navbarScroll" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarScroll">

<ul class="navbar-nav me-auto my-2 my-lg-0 navbar-nav-scroll" style="--bs-scroll-height: 100px;">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="#">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">

Link

</a>

<ul class="dropdown-menu">

<li><a class="dropdown-item" href="#">Action</a></li>

<li><a class="dropdown-item" href="#">Another action</a></li>

<li><hr class="dropdown-divider"></li>

<li><a class="dropdown-item" href="#">Something else here</a></li>

</ul>

</li>

<li class="nav-item">

<a class="nav-link disabled" aria-disabled="true">Link</a>

</li>

</ul>

<form class="d-flex" role="search">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search"/>

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

**Tarjetas:**

<div class="card" style="width: 18rem;">

<img src="..." class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Some quick example text to build on the card title and make up the bulk of the card’s content.</p>

<a href="#" class="btn btn-primary">Go somewhere</a>

</div>

</div>

**Footer:**

<div class="card text-center">

<div class="card-header">

Featured

</div>

<div class="card-body">

<h5 class="card-title">Special title treatment</h5>

<p class="card-text">With supporting text below as a natural lead-in to additional content.</p>

<a href="#" class="btn btn-primary">Go somewhere</a>

</div>

<div class="card-footer text-body-secondary">

2 days ago

</div>

</div>

## Fase 2:

**Paso 1 Definir ¿Qué?** (5 minutos)

* ¿Qué hace la app en una frase?
* ¿Qué ve el usuario cuando la abre?
* ¿Qué acciones puede hacer el usuario?

**Paso 2 Identifica los datos** (10 minutos)

* ¿Qué información necesito guardar?
* ¿Qué estructura tienen estos datos? (array, objetos, array de objetos)

Ejemplo: Tenemos una app de gastos

const gastos =[

{id: 1,

descripción: “café”,

precio: 3.50,

fecha: “2025-10-27”

},

{id: 2,

descripción: “transporte”,

precio: 120.50,

fecha: “2025-10-27”

},

{id: 3,

descripción: “ropa”,

precio: 80,

fecha: “2025-10-27”

}];

**Paso 3 HTML mínimo** (10 minutos)

Crear estructura básica sin estilos

<div id=”app”>

<input id=”input”>

<button id=”btn”> Añadir gasto</button>

<ul id=”list”></ul>

</div>

**Paso 4 Las tres funciones de la app** (30 minutos)

Toda app tienen estas funciones básicas:

1: Obtener los elementos del html (DOM)

const input = document.getElementById(‘input’);

const btn= document.getElementById(‘btn’);

const list= document.getElementById(‘list’);

2. Función para renderizar/mostrar datos

function render(data)={

list.innerHTML=””;

data.forEach(item=>{//Crear y añadir elementos

});

}

3. Función para manejar las acciones del usuario

btn.addEvenListener(‘click’,()=>{mostrar(misDatos)

//lógica de los datos

//actualiza la lista

});

**Paso 5 Iteración** (recorrido)

Hacer las funciones separadas y correctas (una a una), no hacer juntas.

## 

## Fase 3

**Nivel 1**

Ejercicios para manejar las funciones

* Contador de click con botones
* Generador de colores aleatorios (cambiar el fondo)
* Conversor de unidades (euro a dolar o medidas de peso)

**Nivel 2**

Trabajar con listas

* Lista de compras(añadir, eliminar, marcar)
* Gestor de notas de alumnos
* Calculadora con funciones avanzadas

**Nivel 3**

Lógica más compleja:

* Quiz interactivo
* Cronómetro o cuenta atrás
* Filtro de películas/música

**Nivel 4**

APIs y asincronía

* App de clima
* Buscador de música
* Dashboard de criptomonedas

Hacer aplicaciones SIMPLES que funcione lo básico