María Luz Fernández Gismero.

ANEXO UD2. Manual de Boostrap

1. Fundamentos imprescindibles de HTML/CSS para entender Bootstrap

1.1 ¿Qué es una clase CSS y para qué sirve?

Una **clase** es un selector CSS que comienza por . y agrupa reglas de estilo **reutilizables**. Se aplica en el atributo class de una etiqueta HTML para **dar estilo** o **comportamiento visual**.

Cómo se implementa

- Defines (si quieres) tus propias clases en un CSS propio.
- Usas las **clases ya definidas por Bootstrap** para casi todo (espaciado, colores, tipografía, layout…).

Ejemplo

/* custom.css */

.destacado { color: #0d6efd; font-weight: 600; }

Texto con tu clase personalizada

- .destacado es la **clase**; color y font-weight son las reglas que se aplican al elemento con class="destacado".

1.2 Usar múltiples clases a la vez

Puedes combinar varias clases separadas por espacios en el atributo class.

Cómo se implementa

- Escribes class="clase1 clase2 clase3".
- En Bootstrap es muy típico encadenar utilidades: class="text-center fw-bold mb-3".

Ejemplo

Centrado, negrita y con margen inferior

- text-center centra el texto.
- fw-bold pone negrita.
- mb-3 añade margen inferior.

1.3 Cascada, especificidad y orden de carga

Si dos reglas chocan, gana:

1. La de **mayor especificidad** (por ejemplo, #id > .clase).
2. A igual especificidad, la **cargada más tarde**.

Cómo se implementa

- Carga **primero** el CSS de Bootstrap y **después** el tuyo (custom.css) para poder sobrescribir estilos con la misma especificidad.

Ejemplo

<link rel="stylesheet" href="/css/bootstrap.min.css">

<link rel="stylesheet" href="/css/custom.css"> <!-- tu CSS al final -->

- Tu CSS se aplica después, así que puede ganar en la cascada si la especificidad es igual.

1.4 Variables CSS y herencia

Bootstrap usa **variables CSS** (custom properties) para colores, espaciado, etc. Puedes ajustar tu tema sin recompilar Sass modificando variables en :root o por componente.

Cómo se implementa

- Sobrescribe variables: --bs-primary, --bs-body-font-family, etc.

Ejemplo

<style>

:root {

--bs-primary: #6f42c1; /* morado como color primario */

--bs-font-sans-serif: system-ui, -apple-system, "Segoe UI", Roboto, "Helvetica Neue", Arial, sans-serif;

}

```
</style>
```

- Cambias el color **primario** global y la **fuente** base sin tocar los archivos de Bootstrap.

2. ¿Qué es Bootstrap? (contexto e historia)

2.1 Qué es / Para qué sirve

Un **framework CSS/JS** que te da una base sólida para construir sitios **responsive** con una rejilla de 12 columnas, componentes listos y utilidades para diseñar sin escribir CSS desde cero.

2.2 Cómo se implementa (visión general)

- Incluye los archivos de **CSS** y **JS** de Bootstrap.
- Estructura con container → row → col.
- Usa **utilidades** (m-, p-, d-, text-, bg-, etc.) y **componentes** (btn, card, navbar, ...).

2.3 Ejemplo mínimo ya funcional

```
<!doctype html>

<html lang="es">

 <head>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <title>Inicio rápido</title>

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/css/bootstrap.min.css" rel="stylesheet">

 </head>

 <body>

  <div class="container py-4">

   <h1 class="display-6">¡Hola, Bootstrap!</h1>

   <p class="lead">Tu primera página responsive.</p>

   <button class="btn btn-primary">Botón</button>

  </div>
```

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/js/bootstrap.bundle.min.js">
</script>

</body>

</html>

- meta viewport activa el diseño adaptable.
- CSS aplica estilos globales; el bundle JS añade interacciones (dropdown, modal, offcanvas...).

3. Instalación y puesta en marcha

3.1 Qué es / Para qué sirve

Necesitas cargar los archivos del framework para usar su rejilla, utilidades y componentes.

3.2 Cómo se implementa

CDN (rápido y sencillo)

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/js/bootstrap.bundle.min.js">
</script>

Local (archivos descargados)

<link rel="stylesheet" href="/vendor/bootstrap/css/bootstrap.min.css">

<script src="/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

npm (proyectos con bundler)

npm install bootstrap@5.3.8

// main.js

import 'bootstrap/dist/css/bootstrap.min.css'

import 'bootstrap/dist/js/bootstrap.bundle.min.js'

3.3 Ejemplo de estructura de proyecto

project/

├── index.html

├── css/

```
│   ├── bootstrap.min.css
│   └── custom.css
└── js/
    ├── bootstrap.bundle.min.js
    └── app.js
```

- Estructura limpia para trabajar localmente o con bundler.

4. Sistema responsive (Grid)

4.1 Qué es / Para qué sirve

Layout basado en **12 columnas** y **breakpoints** (sm, md, lg, xl, xxl) para adaptar el diseño a distintos anchos de pantalla.

4.2 Cómo se implementa

- Contenedores: .container, .container-fluid, .container-md, etc.
- Fila: .row.
- Columnas: .col, .col-1..12, .col-md-6, etc.
- Espacios (gutters): g-*, gx-*, gy-*.
- Orden/offset: .order-*, .offset-*.

4.3 Ejemplo A — Columnas por breakpoint

```
<div class="container my-4">
  <div class="row g-3">
    <div class="col-12 col-md-6 col-lg-4">A</div>
    <div class="col-12 col-md-6 col-lg-4">B</div>
    <div class="col-12 col-lg-4">C</div>
  </div>
</div>
```

- Móvil: col-12 apila.
- md: A/B ocupan 50%; C sigue 100%.
- lg: A/B/C ocupan 1/3.
- g-3 añade separación uniforme.

4.4 Ejemplo B — Offset, auto, orden

```
<div class="container my-4">
  <div class="row g-2 align-items-center">
    <div class="col-4 offset-4">Centrada con offset</div>
    <div class="col-auto order-2 order-md-1">Auto según contenido</div>
    <div class="col order-1 order-md-2">Resto del ancho</div>
  </div>
</div>
```

- offset-4 desplaza 4 columnas.
- col-auto se ajusta al contenido.
- order-* altera el orden por tamaño.

4.5 Ejemplo C — Alineación vertical y horizontal

```
<div class="container my-4">
  <div class="row align-items-center" style="min-height: 120px;">
    <div class="col-4">Bloque 1</div>
    <div class="col-4 text-center">Bloque 2</div>
    <div class="col-4 d-flex justify-content-end">Bloque 3</div>
  </div>
</div>
```

- .align-items-center centra verticalmente toda la fila.
- .text-center centra texto del bloque 2.
- .d-flex .justify-content-end empuja el bloque 3 a la derecha.

4.6 Ejemplo D — Anidado y gutters independientes

```
<div class="container my-4">
  <div class="row g-4">
    <div class="col-lg-8">
```

```
    <div class="row gy-2">
      <div class="col-6">A1</div>
      <div class="col-6">A2</div>
      <div class="col-12">A3 (anidado)</div>
    </div>
  </div>
  <div class="col-lg-4">B</div>
 </div>
</div>
```

- Grid **anidado**: una fila .row dentro de una columna.
- gy-2 controla el gutter vertical interno sin afectar al externo.

5. Utilidades (Utilities)

5.1 Espaciado (márgenes y padding)

Qué es / Para qué sirve: ajustar rápidamente separación.

Cómo se implementa: m|p + t|b|s|e|x|y + -0..5 (ej.: mt-3, px-4, mx-auto).

Ejemplo

```
<div class="p-4 mb-3 bg-light">Bloque con padding y margen</div>
```

- p-4 padding interno; mb-3 margen inferior; bg-light fondo claro.

5.2 Display y visibilidad

Cómo se implementa: .d-none|block|inline|inline-block|flex|grid + responsive (.d-md-flex).

Ejemplo

```
<p class="d-none d-md-block">Visible desde pantallas medianas</p>
```

- Oculto en móvil, visible en md+.

5.3 Flexbox helpers

Cómo se implementa: .d-flex, .justify-content-*, .align-items-*, .gap-*.

Ejemplo

```
<div class="d-flex justify-content-between align-items-center p-3 bg-light">
  <span>Izquierda</span><span>Centro</span><span>Derecha</span>
</div>
```

- Distribución horizontal con espacio entre extremos y alineado vertical central.

5.4 Tipografía

Cómo se implementa: .fs-1..6, .fw-bold, .fst-italic, .lh-*, .text-start|center|end (+ responsive text-md-start).

Ejemplo

```
<h2 class="fs-3 fw-semibold mb-2">Título</h2>
<p class="lh-lg text-muted">Cuerpo de texto con interlineado largo.</p>
```

- fs-3 tamaño de fuente; fw-semibold grosor; lh-lg interlineado amplio.

5.5 Colores y fondos (sistema de color)

Qué es / Para qué sirve: paleta temática (primary, secondary, success, danger, warning, info, light, dark, body, muted).

Cómo se implementa: .text-*, .bg-*, .border-*, variables --bs-*.

Ejemplo

```
<p class="text-primary">Texto primario</p>
<div class="p-3 mb-2 bg-warning text-dark">Aviso</div>
<div class="p-3 mb-2 bg-dark text-white">Bloque oscuro</div>
```

- Combinación de color de fondo y texto legible.

5.6 Bordes, sombras y tamaño

Cómo se implementa: .border, .border-0, .rounded, .rounded-pill, .shadow, .w-25|50|75|100, .h-*, .img-fluid.

Ejemplo

```
<img class="img-fluid rounded shadow" src="https://picsum.photos/800/400" alt="Responsive">
```

- Imagen se adapta al contenedor; esquinas redondeadas y sombra ligera.

5.7 Ratio para iframes y vídeos

Cómo se implementa: .ratio, .ratio-16x9, .ratio-4x3.

Ejemplo

```
<div class="ratio ratio-16x9">
  <iframe src="https://www.youtube.com/embed/dQw4w9WgXcQ" title="Video" allowfullscreen></iframe>
</div>
```

- Mantiene proporción responsive del contenido embebido.

6. Componentes esenciales

6.1 Botones

Qué es / Para qué sirve: acciones con jerarquía visual.

Cómo se implementa: .btn + variante (.btn-primary, .btn-outline-*, tamaños btn-sm|lg).

Ejemplo

```
<button class="btn btn-primary">Aceptar</button>
<button class="btn btn-outline-danger">Cancelar</button>
<button class="btn btn-success btn-lg">Grande</button>
```

- Base .btn; colores y tamaños comunican prioridad.

6.2 Tarjetas (Cards)

Cómo se implementa: .card, .card-body, .card-title, .card-text, .card-img-top.

Ejemplo

```
<div class="card h-100 shadow-sm" style="max-width:22rem;">
  <img src="https://picsum.photos/600/400?1" class="card-img-top" alt="...">
  <div class="card-body">
    <h5 class="card-title">Tarjeta</h5>
    <p class="card-text">Texto de ejemplo.</p>
    <a href="#" class="btn btn-success">Acción</a>
  </div>
</div>
```

- Estructura semántica clara y visual lista.

6.3 Navbar (colapsable y offcanvas)

Cómo se implementa: .navbar, .navbar-expand-*, .navbar-toggler, .collapse u offcanvas.

Ejemplo (collapse)

```
<nav class="navbar navbar-expand-lg bg-body-tertiary">
  <div class="container">
    <a class="navbar-brand" href="#">Marca</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#menu">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="menu">
      <ul class="navbar-nav ms-auto">
        <li class="nav-item"><a class="nav-link active" href="#">Inicio</a></li>
        <li class="nav-item"><a class="nav-link" href="#">Servicios</a></li>
      </ul>
    </div>
  </div>
```

```
</nav>
```

- En móvil se pliega; desde lg se expande.

Ejemplo (offcanvas)

```
<nav class="navbar bg-body-tertiary">
  <div class="container">
    <a class="navbar-brand" href="#">Marca</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="offcanvas" data-bs-target="#navOffcanvas">
      <span class="navbar-toggler-icon"></span>
    </button>
  </div>
</nav>
<div class="offcanvas offcanvas-end" tabindex="-1" id="navOffcanvas">
  <div class="offcanvas-header">
    <h5 class="offcanvas-title">Menú</h5>
    <button type="button" class="btn-close" data-bs-dismiss="offcanvas"></button>
  </div>
  <div class="offcanvas-body">
    <ul class="navbar-nav ms-auto">
      <li class="nav-item"><a class="nav-link active" href="#">Inicio</a></li>
      <li class="nav-item"><a class="nav-link" href="#">Servicios</a></li>
    </ul>
  </div>
</div>
```

- Menú lateral; se abre/cierra con atributos data-bs-*.

6.4 Formularios + validación visual

Cómo se implementa: .form-label, .form-control, .form-select, .input-group, .form-check, .is-valid/invalid.

Ejemplo

```
<form class="container py-4" novalidate>
  <div class="row g-3">
    <div class="col-md-6">
      <label class="form-label" for="email">Email</label>
      <input class="form-control is-invalid" type="email" id="email" required>
      <div class="invalid-feedback">Introduce un email válido.</div>
    </div>
    <div class="col-md-6">
      <label class="form-label" for="pass">Contraseña</label>
      <input class="form-control is-valid" type="password" id="pass" minlength="6">
      <div class="valid-feedback">Se ve bien.</div>
    </div>
    <div class="col-12">
      <button class="btn btn-primary">Enviar</button>
    </div>
  </div>
</form>
```

- Estados de validación comunican feedback inmediato.

6.5 Tablas responsive

Cómo se implementa: .table, .table-striped, .table-hover, envoltorio .table-responsive.

Ejemplo

```
<div class="table-responsive">
```

```
 <table class="table table-striped table-hover align-middle">
   <thead>
     <tr><th>#</th><th>Nombre</th><th>Estado</th></tr>
   </thead>
   <tbody>
     <tr><td>1</td><td>Ana</td><td><span class="badge text-bg-success">Activa</span></td></tr>
     <tr><td>2</td><td>Luis</td><td><span class="badge text-bg-secondary">Pendiente</span></td></tr>
   </tbody>
 </table>
</div>
```

- El contenedor responsive añade scroll horizontal solo en pantallas estrechas.

6.6 Alertas y modales

Cómo se implementa: .alert alert-* y .modal con data-bs-toggle="modal".

Ejemplo (modal)

```
<button class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#m1">Abrir modal</button>

<div class="modal fade" id="m1" tabindex="-1">
 <div class="modal-dialog">
   <div class="modal-content">
     <div class="modal-header">
       <h5 class="modal-title">Título</h5>
       <button class="btn-close" data-bs-dismiss="modal"></button>
     </div>
     <div class="modal-body">Contenido del modal...</div>
     <div class="modal-footer">
```

```
      <button class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</button>
      <button class="btn btn-primary">Guardar</button>
    </div>
  </div>
 </div>
</div>
```

- El bundle JS gestiona apertura/cierre y accesibilidad.

7. Temas, colores y modo oscuro

7.1 Sistema de colores y variables

Qué es / Para qué sirve: personalizar la **identidad visual** con variables.

Cómo se implementa

- Cambia --bs-primary (y otras) en :root.
- Usa .text-*, .bg-*, .border-* para aplicar colores.

Ejemplo

```
<style>
 :root { --bs-primary: #ff4d4f; }
</style>
<button class="btn btn-primary">Primario personalizado</button>
```

- El botón adopta el nuevo color primario.

7.2 Modo oscuro con data-bs-theme

Cómo se implementa: establece data-bs-theme="dark" en <html> o en un contenedor.

Ejemplo

```
<html lang="es" data-bs-theme="dark">
 <!-- resto del documento -->
</html>
```

- Bootstrap ajusta variables para un esquema de color oscuro.

8. Accesibilidad y buenas prácticas

8.1 Accesibilidad

- Usa etiquetas semánticas (<nav>, <main>, <header>).
- Proporciona alt en imágenes.
- Añade atributos aria-* en elementos interactivos si corresponde.

8.2 Buenas prácticas

- **Mobile-first**: empieza con clases sin sufijo y añade -md-, -lg-.
- Prioriza **utilidades** antes que CSS propio.
- Coloca tu custom.css **después** del de Bootstrap.

9. Casos completos

9.1 Landing con hero (navbar + grid + ratio)

Cómo se implementa: navbar colapsable, sección hero 50/50, CTA y vídeo responsive.

Ejemplo

```
<header>

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

    <div class="container">

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

      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#menu">

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

      </button>

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

        <ul class="navbar-nav ms-auto">

          <li class="nav-item"><a class="nav-link active" href="#">Inicio</a></li>

          <li class="nav-item"><a class="nav-link" href="#">Servicios</a></li>

          <li class="nav-item"><a class="nav-link" href="#">Contacto</a></li>
```

```
      </ul>
     </div>
    </div>
   </nav>
   <section class="container py-5">
    <div class="row align-items-center g-4">
     <div class="col-12 col-lg-6">
      <h1 class="display-5 mb-3">Título llamativo</h1>
      <p class="lead">Subtítulo con propuesta de valor.</p>
      <div class="d-flex gap-2">
       <a class="btn btn-primary btn-lg" href="#">Comenzar</a>
       <a class="btn btn-outline-secondary btn-lg" href="#">Saber más</a>
      </div>
     </div>
     <div class="col-12 col-lg-6">
      <div class="ratio ratio-16x9">
       <iframe src="https://www.youtube.com/embed/dQw4w9WgXcQ" title="Video"
allowfullscreen></iframe>
      </div>
     </div>
    </div>
   </section>
</header>
```

- Diseño responsive, botones de acción y vídeo que mantiene proporción.

9.2 Mosaico de tarjetas (catálogo/noticias)

Cómo se implementa: grid con row/col y tarjetas .card.

Ejemplo

```
<section class="container py-5">
  <h2 class="mb-4">Noticias</h2>
  <div class="row g-4">
    <!-- Repite este patrón de artículo -->
    <article class="col-12 col-sm-6 col-lg-4">
      <div class="card h-100 shadow-sm">
        <img src="https://picsum.photos/600/400?1" class="card-img-top" alt="...">
        <div class="card-body">
          <h3 class="card-title h5">Tarjeta 1</h3>
          <p class="card-text">Texto de ejemplo para la tarjeta.</p>
          <a href="#" class="btn btn-primary">Leer más</a>
        </div>
      </div>
    </article>
    <!-- ... más artículos ... -->
  </div>
</section>
```

- Columnas 1/2/3 según breakpoint; alturas igualadas con .h-100.

9.3 Página de formulario con validación

Cómo se implementa: grid + controles de formulario + feedback visual.

Ejemplo

```
<main class="container py-4">
  <h2 class="mb-3">Contacto</h2>
  <form novalidate class="row g-3">
```

```
    <div class="col-md-6">
      <label class="form-label" for="nombre">Nombre</label>
      <input class="form-control" type="text" id="nombre" required>
      <div class="invalid-feedback">Campo obligatorio.</div>
    </div>
    <div class="col-md-6">
      <label class="form-label" for="email2">Email</label>
      <input class="form-control" type="email" id="email2" required>
      <div class="invalid-feedback">Introduce un email válido.</div>
    </div>
    <div class="col-12">
      <label class="form-label" for="mensaje2">Mensaje</label>
      <textarea class="form-control" id="mensaje2" rows="4"></textarea>
    </div>
    <div class="col-12">
      <button class="btn btn-primary">Enviar</button>
    </div>
  </form>
</main>
```

- Estructura clara, controles accesibles, feedback preparado (aplícalo con JS al validar).

10. Checklist final para tus proyectos

- Cargar CSS/JS de Bootstrap (CDN o local o npm).
- Usar container → row → col para el layout.
- Definir la jerarquía con **tipografía** (fs-*, fw-*, lh-*).
- Aplicar **colores** y contraste adecuados (text-*, bg-*).
- Ajustar **espaciado** con utilidades m-/p-.
- Hacer **componentes** (botones, navbar, tarjetas, formularios…).
- Probar en **móvil primero** y en todos los breakpoints.

- Revisar **accesibilidad** (alt, aria, foco visible).
- Personalizar variables --bs-* si necesitas branding.