



UNIVERSITAT DE  
BARCELONA

Facultad de Matemáticas e Informática  
Grado en Ingeniería Informática

# Computación Orientada a la Web

**Entrega Sección 1**

**Alumno:** William D. Sotto

# Memoria de la Práctica – Sección 1

## 1. Introducción

El objetivo de esta práctica es **crear una página web estática** para un sistema de reservas de hoteles y lo haré en dos apartados distintos:

- **Apartado 1:** Emplear únicamente **HTML y CSS** (sin Bootstrap).
- **Apartado 2:** Adaptar la misma página para usar **Bootstrap** y lograr un diseño responsivo.

He seguido las indicaciones de los **PDF** proporcionados en clase:

- **“01\_5\_HTML\_CSS\_repaso.pdf”:** Viene la estructura fundamental de un documento HTML y la organización con CSS externo (separación contenido / presentación).
- **“01\_6\_BOOTSTRAP.pdf”:** Explica cómo integrar la librería Bootstrap y su sistema de rejillas, componentes, clases para formularios, etc.
- **“PHP Form.pdf”:** Muestra cómo funcionan los formularios y el envío de datos (GET/POST), que me servirán para crear los pequeños formularios de búsqueda de la web.

## 2. Apartado 1: Versión HTML + CSS (sin Bootstrap)

Los ficheros principales que he entregado son:

1. **noBootstrap/index.html**
2. **noBootstrap/styles.css**

### 2.1 Estructura y Organización de **index.html**

Tal como se recomienda en **“01\_5\_HTML\_CSS\_repaso.pdf”**, he creado un documento HTML con:

- **Encabezado** (`<header>`) fijo, donde muestro el título “Reservas de Hoteles”.
- **Cuerpo principal** (contenedor `<div class="container">`) con distintos bloques que representan:
  - Un **formulario de inicio de sesión** (Sign In), con campos `<input type="text" ...>` y `<input type="password" ...>`.

- Un **formulario** para buscar en Google y Wikipedia. Esto está basado en la explicación de formularios GET en "PHP Form.pdf". Cuando se envía, se dirige a <https://www.google.com/search> (o <https://es.wikipedia.org/wiki/Especial:Buscar>), pasándole el contenido del input con `name="q"` (o `name="search"`).
- **Tarjetas (cards)** que describen varios hoteles: "Hotel Poseidon Resort", "Hesperia Murcia Centro", "Hotel Barceló Valencia", "Hotel Riu Plaza España", y "Hotel Nelva". Cada tarjeta contiene:
  - Un `<h2>` con el nombre del hotel.
  - Una imagen (`<img>`) con su respectivo `alt`.
  - Texto descriptivo, como zona, estrellas o precio.

Las secciones siguen la idea de **estructurar** el contenido en HTML descrita en la teoría (ej., secciones con `<div>` para layout, uso de encabezados `<h1>`, `<h2>` ...).

## 2.2 Hojas de Estilo en **styles.css**

En "01\_5\_HTML\_CSS\_repaso.pdf" se enfatiza la separación de contenido y estilo. Por eso, tocaba crear un archivo CSS externo para personalizar, así que añadí algunas como:

**Posicionamiento fijo** del `<header>`:

```
header {
  background: #007bff;    color: white;
  padding-left: 20px;    text-align: left;
  position: fixed;       top: 0;
  width: 100%;
}
```

**Contenedor principal:**

```
.container {
  padding-top: 100px;    padding-bottom: 50px;
  display: flex;         justify-content: center;
  margin: 20px;         flex-wrap: wrap;
}
```

- Con `padding-top: 100px` se resuelve el espacio bajo el header fijo.

**Tarjetas (.card)** para cada hotel, con `box-shadow` y `border-radius` (inspirado en la teoría sobre diseño y layout de CSS):

```
.card {
  background: white;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);... }
```

**Formularios** (con `.search-container`) para los inputs de búsqueda, donde se definen anchos, padding y bordes.

Estas clases las apliqué intentando seguir los **ejemplos de layout** del PDF de HTML/CSS, aprovechando `display: flex;`, `flex-wrap`, y ajustando márgenes y rellenos en la UI.

---

## 3. Apartado 2: Versión con Bootstrap

Para el segundo apartado, guardé los archivos como:

1. **Bootstrap/bootstrap.html**
2. **Bootstrap/bootstrap\_styles.css** (para retoques personalizados)

### 3.1 Uso de Bootstrap en `bootstrap.html`

Según “01\_6\_BOOTSTRAP.pdf”, debía integrar en mi `<head>` el **CSS de Bootstrap**, y obtuve las siguientes ideas: Por ejemplo:

**Barra de navegación** con `.navbar`, `.bg-primary`, `.navbar-dark`:

```
<nav class="navbar navbar-expand-lg bg-primary navbar-dark">
  <div class="container">
    <a class="navbar-brand fw-bold" href="#">Reservas de Hoteles</a>
  </div>
</nav>
```

**Sistema de rejilla** (`.row`, `.col-md-6`, `.col-md-4`, etc.) para organizar el contenido en columnas responsivas:

```
<div class="container-fluid bg-primary">
  <div class="row">
    <div class="col-md-6 d-none d-md-block bg-hotel">
      ...
    </div>
    <div class="col-md-6 d-flex align-items-center justify-content-center">
      ...
    </div>...
</div>
```

- Aquí creé dos columnas: la izquierda con un fondo de imagen, la derecha con el formulario de login, así el Login no quedaba tan suelto como el primer index.html
- **Clases de formularios** (`.form-group`, `.form-control`, `.btn btn-primary`), sustituyendo la maquetación manual por las utilidades de Bootstrap.

### Ejemplo real de **bootstrap.html**:

```
<h2 class="text-center">Buscar en la Web</h2>
<form class="row g-2" action="https://www.google.com/search" method="GET">
  <div class="col-md-8">
    <input type="text" class="form-control" name="q" placeholder="Buscar en Google...">
  </div> <div class="col-md-4">
    <button type="submit" class="btn btn-primary w-100">Google</button>
  </div> </form>
...
```

- Se puede observar que rehusé **la lógica de formularios** del primer apartado (buscar en Google/Wikipedia), pero ahora con **clases** de Bootstrap (`.form-control`, `.col-md-8`, `.col-md-4`, etc.) y un layout más responsivo, como se explica en "01\_6\_BOOTSTRA.pdf".

## 3.2 Personalizaciones en **bootstrap\_styles.css**

Aunque Bootstrap tiene la mayoría de los estilos, incluí un CSS adicional para refinar detalles que no pude encontrar. Por ejemplo:

```
.navbar {
  position: fixed;      top: 0;
  width: 100%;         z-index: 100;
}
.main-container { padding-top: 50px; }
.bg-hotel { background: url('.') no-repeat center center fixed;
  background-size: cover;                      min-height: 100vh;
}
```

Al final entonces, **combino** clases definidas ya por Bootstrap con modificaciones propias para lograr un diseño a mi medida y agrado.

## 4. Formularios y Métodos GET/POST

En ambos apartados (HTML/CSS y Bootstrap) incluí ejemplos de formularios para **buscar en Google** y **Wikipedia**. Esto se relaciona con "**PHP Form.pdf**", donde se explica que:

- **method="GET"**: Los datos se envían en la propia URL (como `?q=Hotel` al hacer la búsqueda).
- **action="URL"**: Indica a dónde enviar la petición.

Fue más simple de lo que esperaba, y en mi código se puede ver como:

```
<form action="https://www.google.com/search" method="GET">
  <input type="text" name="q" placeholder="Buscar en Google...">
  <button type="submit">Google</button>
</form>
```

Lo cual ejemplifica la parte teórica sobre el uso de formularios GET para búsquedas.

## 5. Conclusión

Esta práctica demuestra cómo:

1. **Versión sin Bootstrap** ([index.html](#), [styles.css](#)):
  - Seguí la buena práctica de separar contenido (HTML) y estilo (CSS).
  - Pude seguir "01\_5\_HTML\_CSS\_repaso.pdf" para crear un layout sencillo usando [position: fixed](#), [display: flex](#), [box-shadow](#) en tarjetas, etc.
  - Y añadí métodos GET (aunque sea solo para búsquedas web, no para procesar en un servidor propio).
2. **Versión con Bootstrap** ([bootstrap.html](#), [bootstrap\\_styles.css](#)):
  - Incorporé la librería recomendada en "01\_6\_BOOTSTRAP.pdf" y usé clases de rejilla ([row](#), [col-](#), [container](#), etc.), componentes de formulario ([form-control](#), [.btn](#)), y la barra de navegación ([.navbar](#)).
  - La web es responsiva automáticamente gracias a la rejilla de Bootstrap.
  - Mantuve la idea de los mismos formularios e información de hoteles, pero con estilos y componentes listos de Bootstrap.

### Pequeña anécdota:

Cuando empecé, maquetar la página HTML y CSS me parecía tedioso. Era un montón de ajustar márgenes, layouts y probar! Con Bootstrap, en cambio, sentí que la web casi se ordenaba sola: sólo agregaba las clases ([row](#), [col-md-6](#), etc.) y todo se veía bonito y responsivo. Al final, comprobé la importancia de dominar tanto el lado "manual" (por si hay que personalizar mucho) como la rapidez de usar un framework.