

Creación del frontend de la web
The Poké-Booking Service

Informe de la Práctica 1

Computació Orientada al Web
Simone Balocco

Trabajo realizado por
Marcos Plaza Gonzalez

Marzo 2021

Tabla de contenidos

Introducción	3
Apartado 1. Implementación con HTML y CSS	4
Apartado 2. Implementación con HTML y CSS junto a Bootstrap y Fontawesome	6
Conclusiones	8

Introducción

"**The Poké-Booking Service** es una plataforma con la que entrenadores Pokémon de todas las edades pueden reservar hotel en cualquier rincón del vasto mundo Pokémon. Ya sea por placer o porque te diriges a tu próximo desafío de gimnasio Pokémon, esta web es para ti."

Este es de alguna forma el eslogan promocional de mi web. Dado que la temática de la web es libre, he elegido la franquicia *Pokémon* para crear mi proyecto. En este mundo ficticio, encontramos diferentes ciudades y pueblos (cada uno con sus atractivos únicos) localizados en diferentes regiones con diferentes climas y criaturas.

Por otro lado, esta es una web de reserva de hoteles, por lo tanto la estructura y funcionalidades de la web debe cumplir unos requisitos mínimos. Como referencia para la página principal, me he fijado en la página de *Booking* que es la que suelo utilizar cuando viajo.

El objetivo más general de este proyecto es trasladar las páginas web de reserva de hoteles al mundo de los *Pokémon*.

Aun así, el objetivo principal concreto de esta práctica es aprender a armar el frontend de una web (en este caso una sola página) desde 0 a partir de las tecnologías *HTML*, *CSS* y *Bootstrap* vistas en la teoría de clase.

En este informe se explica la estructura de la página principal de *The Poké-Booking Service*, es decir, los diferentes elementos que la componen, que a su vez deben ser acordes con la temática propuesta. Se divide en dos partes; apartado 1 y apartado 2. La documentación de la programación de la vista es la que se encuentra el *Campus Virtual* de la asignatura (*01_5_HTML_CSS_repasso_S*, *01_6_BOOTSTRAP*). También he buscado en internet información sobre los mismos elementos que aparecen en los pdfs.

Apartado 1. Implementación con *HTML* y *CSS*

En este primer apartado se describe la página principal de *The Poké-Booking Service*, pero empleando únicamente lenguaje *HTML* junto las hojas de estilo correspondientes, *CSS*. Los elementos del layout de página web se dividen en 3 bloques principales; cabecera o header, contenido principal (que a su vez se dividirá en dos bloques) y footer. En la teoría hemos visto cómo implementar este tipo de estructura sencilla para distribuir los bloques de contenido usando *HTML* junto *CSS*, estableciendo valores para los distintos atributos para *<div>*.

1.a) Cabecera

En esta sección se dispone el título o logo de la empresa, así como los botones necesarios para; llegar a la página de ayuda, hacer el registro e inicio de sesión del usuario y un enlace a mi página de *GitHub* (a modo de firma de autor). De momento, es una cabecera simple. No se descarta implementar funcionalidades extra en entregas siguientes.

1.b) Contenido principal

En este apartado es donde se concentran las funcionalidades principales de la web.

En primer lugar encontramos la caja de búsqueda. Es un elemento útil y compacto que contiene un formulario que reúne los tres inputs esenciales (cabe decir que son de tipos diferentes; text, date y numbers) para que el usuario encuentre el hotel perfecto para él; ciudad o localización del hotel, fecha de entrada y de salida así como el número de personas que se van a alojar.

A continuación he dividido el contenido en dos columnas. La **columna** que se encuentra **a la izquierda** la he reservado para complementar el objetivo de buscar un hotel para reservar. La primera caja de esta columna está hecha para mostrar la lista de regiones que cuentan con hoteles para reservar. El objetivo de cada una de las entradas es llevar al usuario a otra página donde se muestre la información general de la región; atractivos turísticos, ciudades más importantes, que clase de Pokémon podemos encontrar, que gimnasios Pokémon están disponibles, etcétera. A continuación vemos otra sección más abajo de esta donde se mantiene un ranking de los hoteles mejor valorados y puntuados, para fomentar la competencia entre hoteles y proporcionar a los usuarios lo mejor de lo mejor de nuestra base de datos. Por último ofrecemos un filtro para buscar por clase de Pokémon. Ahora mismo es un formulario simple para buscar tu Pokémon favorito en Google. No se descarta añadir en un futuro una sección de filtros un poco más compacta donde tengamos para filtrar según diferentes criterios.

En la **columna** más grande, **situada a la derecha** tenemos los hoteles en tendencia. Aquellos hoteles más visitados y mejor puntuados (y que paguen a la empresa por darles publicidad adicional) tendrán una mayor visibilidad. Esta columna no está pensada para emular un *infinite scroll*, sino que paginaremos todos aquellos hoteles en tendencias en diferentes pestañas. De momento tenemos un par de hoteles en tendencias. Como apunte se ha intentado que las imágenes no sobresalgan del *<div>* que las contiene para hacerlas lo más "responsive" posible (en *CSS* le damos el *width* que creamos y el *height* lo ponemos a auto).

1.c) Footer

En esta sección incluiremos información adicional de la página web, contactos de teléfono y mail, así como enlaces útiles para usuarios y arrendatarios de alojamiento.

El resultado de la estructura descrita es el siguiente.

Titulo/logo

The Poké-Booking Service

Book hotels around the Poké-globe

Grupa botones

Need help?

Sign Up

Sign In

About me

Explore the Pokémon world! Find deals on hotels, homes, and much more...

From cozy country homes to funky city apartments

City name?

Date? →

How many People?

Caja de búsqueda

So many regions to visit...

- [Kanto](#)
- [Johto](#)
- [Hoenn](#)
- [Sinnoh](#)
- [Unova](#)
- [Kalos](#)
- [Alola](#)
- [Galar](#)

Look what's new today

Bellchime Trail Hotel on Ecruteak City

[View location on map](#)



Brief description

Tower belonging to an old Japanese temple where the legendary pokémon Ho-oh is usually worshiped. It has amazing views at all times of the year, as well as facilities of the highest luxury. It is ideal for couples as the legend says that those who manage to see the seven-colored bird will enjoy eternal bliss.

* The facilities are made of a special prepol wood. However, many of the furniture is made of oak wood, therefore please avoid that the fire-type pokémon roam freely.

Room type	Sleeps	Price for 1 night
Luxury room (Book here)	1 queen bed (2 people) (More information)	170000P
Basic room (Book here)	2 twin beds (2 people) (More information)	300000P
Onsen room (Book here)	2 twin beds (2 people) (More information)	100000P

Estival Avenue Hostel on Lumiose City

[View location on map](#)



Brief description

Hostel located in one of the four main avenues of this city. It has refined luxury facilities ideal for all types of guests. Food is undoubtedly one of the strong points of this place, as it has the best trio of chefs in the world, Cilian, Chili and Kress.

Room type	Sleeps	Price for 1 night
Luxury room (Book here)	1 queen bed (2 people) (More information)	170000P
Basic room (Book here)	2 twin beds (2 people) (More information)	300000P

Hey trainer, what is your favourite Pokémon?

Complementos



Hoteles en tendencia

Footer

Marcos Plaza González. Computació Orientada al web. Práctica 1, apartat 1.

Figura 1. Estructura y secciones de la página.

Cabecera

Contenido principal

Footer

5

Apartado 2. Implementación con *HTML* y *CSS* junto a *Bootstrap* y *Fontawesome*

En este segundo y último apartado se describe la página principal de *The Poké-Booking Service*, pero esta vez empleando la librería *Bootstrap*. Tal y como ha sucedido antes, los elementos del layout de página web se dividen en 3 bloques principales; cabecera o header, contenido principal y footer. Cabe decir que la librería *Bootstrap* implementa el diseño responsive. Este tipo de diseño hace que independientemente del tamaño de la pantalla, los elementos de la web van a adaptarse a las dimensiones necesarias de manera automática. Además *Bootstrap* trae consigo el sistema de parrilla el cual facilita mucho la distribución de los elementos de manera ordenada y limpia, entre otros elementos que ayudan al programador a seguir un orden. A continuación citaré los cambios establecidos en comparación al anterior apartado. Cabe decir que el estilo de algunos elementos de *Bootstrap* se ha manipulado a través de style directamente en el *HTML*, y aunque no es una buena práctica de programación, en el momento de la implementación me ha servido para probar más fácilmente como funcionan los atributos de cada clase.

2.a) Cabecera

En esta sección se ha empleado la clase *navbar* que ayuda a disponer los elementos de una barra de navegación tradicional. Además del logo y el subtítulo, he añadido un *button-toolbar* donde se alojan de manera horizontal y contigua cada uno de los botones implementados anteriormente. Los botones se generan a partir de la clase *btn* y fácilmente se pueden manipular sus atributos estéticos accediendo a ellos directamente. Como detalle adicional se ha implementado un *alert* para advertir sobre los peligros de la pandemia del *Pokérus* un virus que afecta a los *Pokémon*.

2.b) Contenido principal

El propósito de este contenido principal es exactamente el mismo que en el anterior apartado. La caja de búsqueda, es un formulario de la clase *form-inline* que dispone todos sus elementos de manera horizontal, al mismo tiempo que se emplean *form-group* con el atributo *mb-2* (para dejar márgenes equivalentes) para cada *input* junto a su *label* correspondiente. Uno de los puntos fuertes de *Bootsrap* es el sistema de parrilla que se puede emplear dentro de un elemento de la clase *container* y *container-fluid*. En mi caso utilizo *container-fluid* porque me deja manipular los márgenes y *paddings* de manera más directa. Debemos tener en cuenta que estos containers se pueden dividir un máximo de 12 columnas por cada fila que pongamos. Por tanto, se disponen los elementos anteriores del cuerpo principal de la página, utilizando *row* para crear una fila al mismo tiempo que usamos *col* para dividir esta fila en las dos columnas que queremos. Para cada columna se le puede asignar un tamaño. En mi caso, para la columna izquierda he querido que utilice 4 de las 12 disponibles, mientras que para la derecha he usado las 8 restantes. De esta forma he podido emular el resultado anterior de forma más fácil y ordenada.

Por otro lado, se han añadido más elementos decorativos como un paginador para la columna donde se muestran los hoteles en tendencias, badges para representar la nota (sobre 10) que tiene cada hotel, además de una puntuación adicional de estrellas (de 0 a 5). Las tablas se presentan con un resultado mucho más estético utilizando la clase *table* y *table-striped*. Las imágenes son responsive directamente empleando la clase *img-fluid*.

2.c) Footer

En este apartado no ha habido cambios.

A continuación veremos la comparativa entre los dos apartados. Tanto la vista como en el navegador como en un dispositivo móvil.

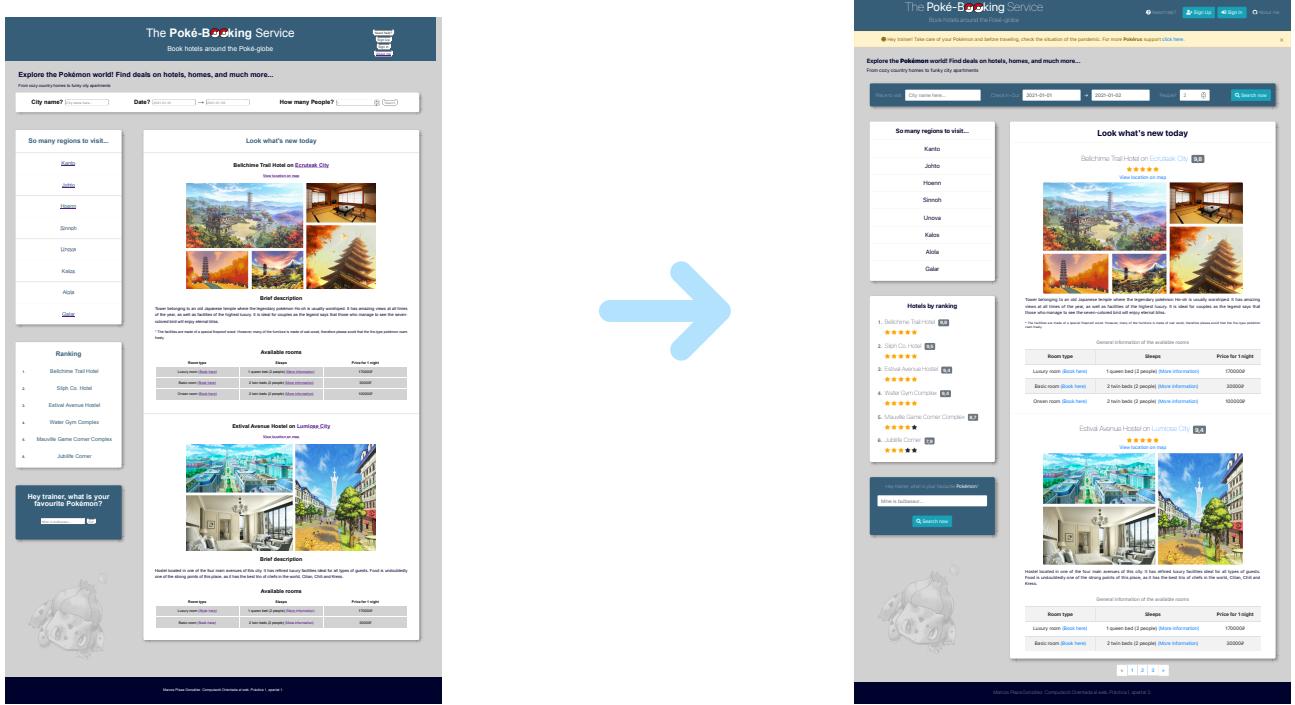


Figura 2. Comparativa de resultados en Safari de escritorio.

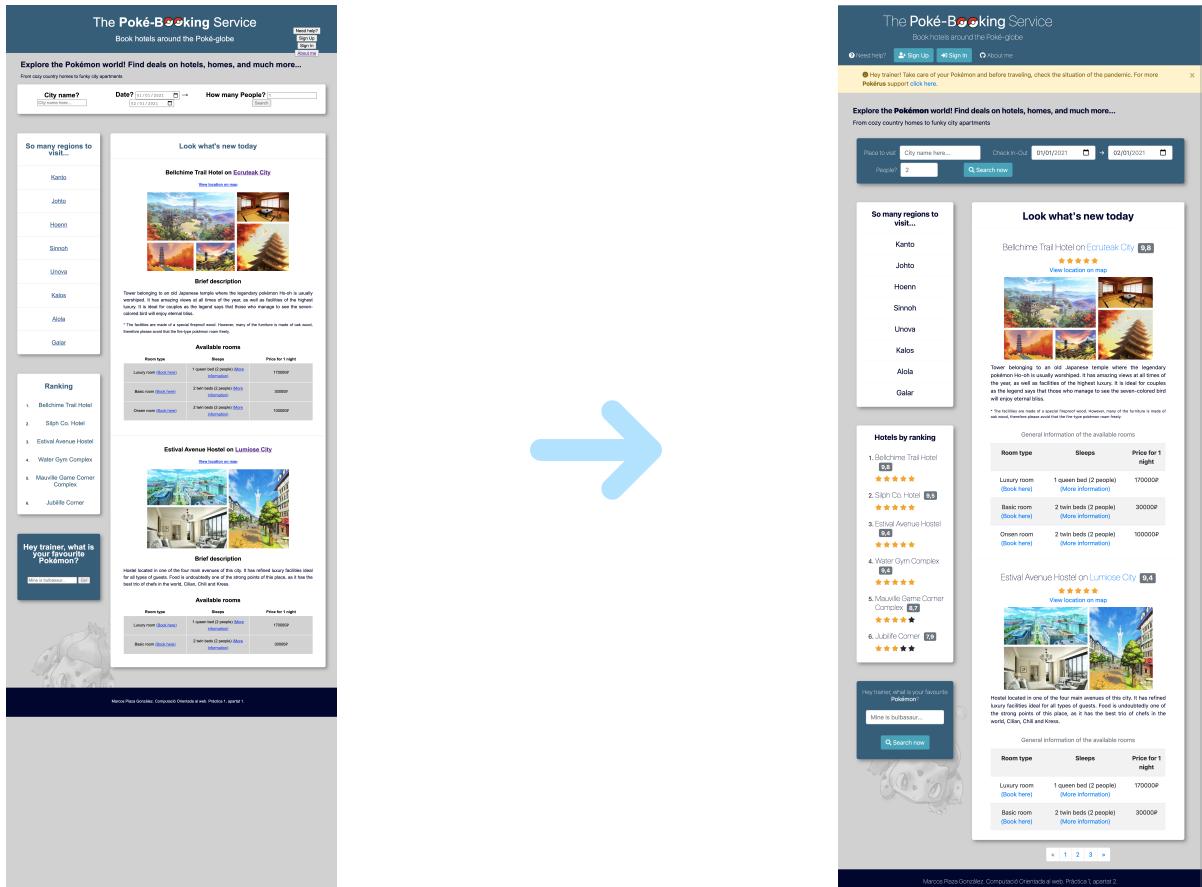


Figura 3. Comparativa de resultados en Chrome de móvil.

Conclusiones

Tal y como hemos visto, el objetivo para esta práctica se ha logrado cumplir. A partir de los elementos expuestos en la teoría se ha construido una web con un layout más o menos meditado y coherente acorde la temática propuesta. Los resultados se han probado en distintos navegadores, así como en distintas pantallas y la visualización es correcta.

Hemos visto que en el apartado 1 tenemos una página más o menos responsive aunque no está implementada con *Bootstrap*. Por otro lado las líneas que tenemos en la hoja de estilo o CSS son muchas en contraste a lo que tenemos en el apartado 2. Por esta razón, pienso que *Bootstrap* es una herramienta muy útil para el desarrollo de frontend, ya que ayuda al programador a implementar menos cosas y a utilizar las herramientas que trae consigo la librería, en mucho menos tiempo y en mi opinión con mejores resultados, tanto estéticos como técnicos (ya que utiliza funciones de *Javascript* de forma complementaria a las hojas de estilo).

Personalmente, pienso que la clave para el desarrollo de esta práctica ha sido establecer un orden y por supuesto tener clara la jerarquía para los elementos que se pretenden usar en el *HTML* usando de manera complementaria las hojas de estilo en cascada o CSS. Aprender *Bootstrap* es otra de las claves, ya que como he dicho, trae consigo muchas funciones útiles que ayudan a ahorrar tiempo y recursos.