



# Documentación Entrega 2

Programación en Internet

Jesús Castaño Tato Y Fernando Parra Salomón



# ÍNDICE

Introducción.....	2
Registro y acceso de usuarios.....	2
Gestión de alojamientos.....	2
Gestión de habitaciones.....	5
Gestión de reservas .....	6
Valoración.....	8
Alojamientos favoritos .....	10
Buscar alojamiento .....	12
Completo/disponible .....	14

## Introducción

El proyecto tratado a continuación se trata de una recreación de la conocida página para efectuar viajes Booking. Esta recreación cuenta con funcionalidades como buscar alojamientos por distintos parámetros, gestionar el conjunto de propiedades y habitaciones de estas, efectuar reservas de habitaciones, entre otras.

## Registro y acceso de usuarios

Para la realización de esta funcionalidad se han creado tres servlets, el LoginServlet, el RegistroServlet y el LogoutServlet.

**LoginServlet:** Este servlet maneja el proceso de inicio de sesión de los usuarios. Verifica las credenciales proporcionadas por el usuario con las almacenadas en la base de datos. Si las credenciales son correctas, se inicia una sesión para el usuario y se redirige a la página de búsqueda de alojamientos. Si las credenciales son incorrectas, se muestra un mensaje de error y se vuelve a mostrar el formulario de inicio de sesión.

En este servlet también se introduce un mapa en la sesión para manejar los datos del carrito para efectuar reservas.

**RegistroServlet:** Este servlet se encarga del proceso de registro de nuevos usuarios. En el método doGet, elimina cualquier usuario existente en la sesión y redirige al formulario de registro. En el método doPost, recibe los datos del formulario de registro, valida la información del usuario y, si es válida, añade el usuario a la base de datos. Luego, inicia una sesión para el nuevo usuario, crea un mapa para los datos del carrito (al igual que en el Login) y redirige a la página de búsqueda de alojamientos. Si los datos del usuario no son válidos, muestra mensajes de error en la vista de registro.

**LogoutServlet:** Este servlet se encarga de eliminar al usuario de la sesión y de redirigir de nuevo a la página de inicio de sesión.

## Gestión de alojamientos

Para esta funcionalidad se han utilizado varios servlets, se ha modificado la base de datos y se han añadido clases en consecuencia. Para acceder a esta funcionalidad, hay que iniciar sesión, una vez iniciada, en la pantalla principal se pulsa el botón con el nombre de usuario y a continuación en mis alojamientos.

Booking.com

Carrito

Chema

1

2

Mi perfil

Mis alojamientos

Mis reservas

Mis favoritos

Logout

Encuentra tu próxima estancia

Busca ofertas en hoteles, casas y mucho más...

¿A dónde vas?

Ofertas

Promociones, descuentos y ofertas especiales para ti

Disfruta de tus vacaciones más largas

Busca alojamientos que ofrezcan estancias largas, hay muchas con tarifa mensual reducida

Elige tu estancia

Vuela al destino de tus sueños

Inspírate, compara y reserva vuelos con más flexibilidad

Buscar vuelos

Busca por tipo de alojamiento

Hoteles

Apartamentos

Resorts

Villas

Una vez dentro se encuentra la funcionalidad de **Nuevo Alojamiento**, **Editar** alojamiento y **Eliminar** alojamiento.

Booking.com

Carrito

Chema

Mis alojamientos

Nuevo alojamiento

1

Caceres

NH Oquendo

Plaza San Juan  
655655655  
a 0.2km del centro

2

3

Editar

Eliminar

Gestionar habitaciones

Caceres

Prueba para Detalles

Plaza de España  
123456789  
a 3.0km del centro

Editar

Eliminar

Gestionar habitaciones

Malaga

Antes de explicar el funcionamiento se van a indicar las nuevas tablas de la base de datos, así como sus clases relacionadas. La primera tabla se trata de la de services, esta como indica su nombre gestiona los distintos servicios que puede tener una propiedad.



▼	services	
	id	INTEGER
	name	TEXT
	icon	TEXT

Cuenta con un ID único para cada servicio, el nombre de este y otro campo para meter el icono que va a utilizar en el jsp.

La segunda tabla es la de propertiesServices, similar en su estructura a la de propertiesCategories.



▼	propertiesServices	
	idp	INTEGER
	ids	INTEGER

Esta cuenta con un idp que hace referencia a la propiedad que sea y un ids que referencia a un servicio que esta ofrece, habrá tantas entradas con la misma propiedad como servicios ofrezca.

Se han creado sus respectivas clases con getters y setters y en el DAO se han creado algunos métodos entre los que destacan alguno para obtener el nombre de los servicios que tiene una propiedad, otro que obtiene todos los servicios de la base de datos...

Con respecto a los servlets cabe destacar que tanto el Nuevo alojamiento como el Editar, utilizan el mismo jsp, pero en cada código se le pasa por la request un String para cambiar el título de la página.

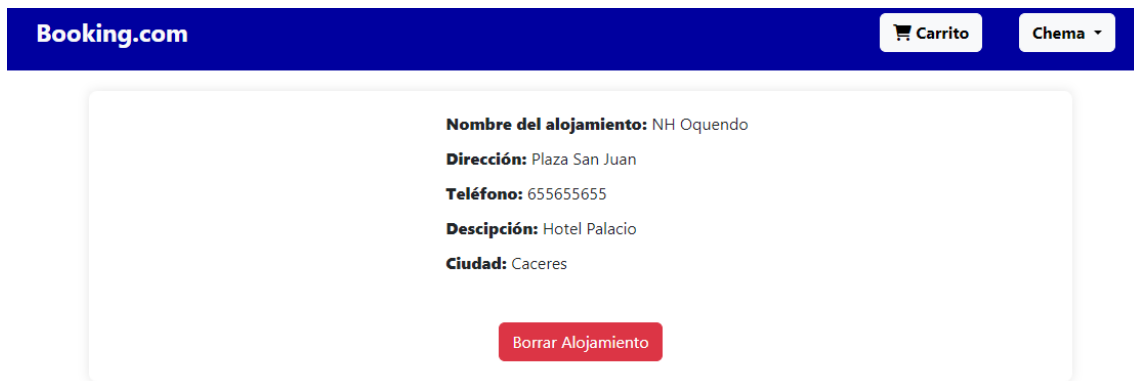
**CrearPropiedadServlet:** En primer lugar, se comprueba si hay alguna propiedad en la sesión en cuyo caso la elimina, a continuación, obtiene la lista de todos los servicios que existen y la envía a la request y por último ya redirige a la página.

Una vez introducida y validada la información del alojamiento que se quiere crear, se le asignan a una nueva propiedad los valores introducidos por el usuario y se añade a la tabla de propiedades, además se añadirán tantas filas a la tabla de propertiesServices como servicios haya indicado el usuario que tiene la propiedad.

**EditPropertyServlet:** Este servlet muestra los mismos datos que el anterior, pero con los campos ya precargados recuperándolos de la base de datos.

El usuario puede modificar los distintos campos y una vez se pulsa en guardar y se validaran, estos se actualizan en la base de datos. En el caso de los servicios se añaden aquellos nuevos y se eliminan aquellos que estaban y que ahora se han deseleccionado.

**DeletePropertyServlet:** Este servlet muestra una página de resumen con alguna de las características de la propiedad.



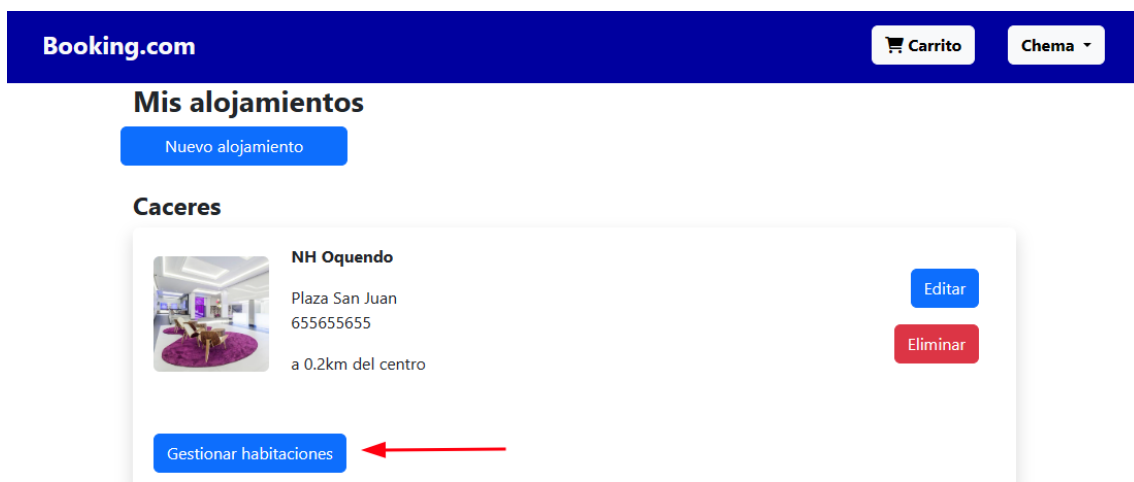
Y al pulsar en borrar se eliminan todos los datos relacionados con esa propiedad.

## Gestión de habitaciones

Para cumplir con los requisitos que se piden, se necesitó la creación de varios servlets. Como puntualización, el jsp utilizado tanto por **CreateAccommodationServlet** como por **EditAccommodationServlet** es el mismo, lo único que varía es el título que se muestra, el cual se le pasa como CheckType.

**ManageAccommodationsServlet:** Se comprueba si el id de usuario propietario del alojamiento al que pertenece la habitación es el mismo que el usuario que está en la sesión, si es así se obtienen todas las habitaciones asociadas con ese alojamiento y se muestran.

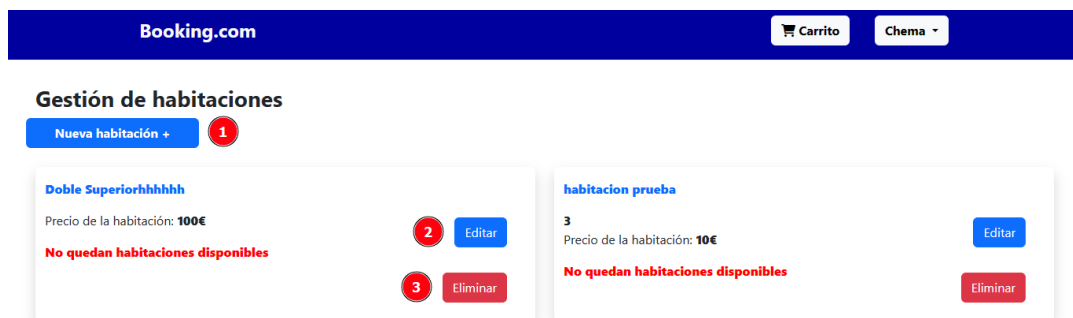
En la imagen siguiente se puede ver donde actúa ManageAccommodationsServlet.



**CreateAccommodationServlet:** En primer lugar, se comprueba si la propiedad de la que se quieren crear habitaciones existe, y que el id del usuario asociado a la propiedad es el mismo que el usuario de la sesión. En caso afirmativo se redirige a una nueva página donde rellenar los datos que se quieran poner de la habitación y se añaden los datos de la nueva habitación a la tabla “*accommodations*” de la base de datos.

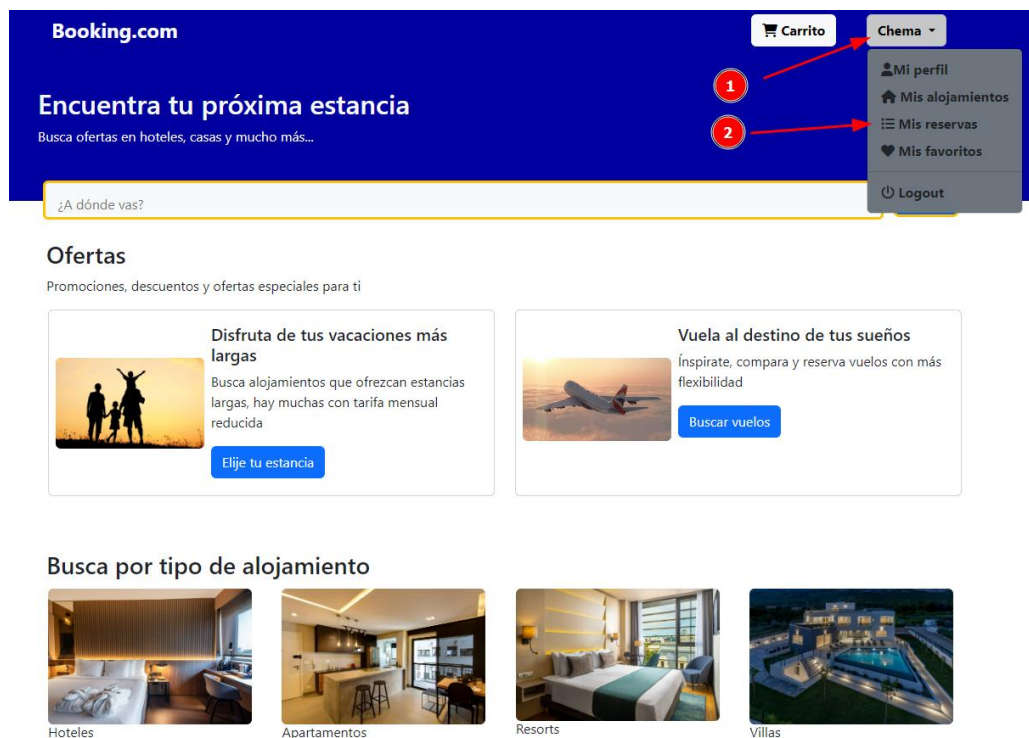
**EditAccommodationServlet:** Mismas comprobaciones al inicio. Redirige al archivo .jsp compartido con CreateAccommodationServlet. Durante la edición de los datos pertinentes, se actualiza la entrada correspondiente en la tabla "accommodations" con el identificador de la propiedad asociada. Además, se lleva a cabo una comprobación para determinar si el número de habitaciones es cero; en tal caso, se marca el alojamiento como "no disponible". Por otro lado, si el número de habitaciones es mayor o igual a uno y el alojamiento estaba previamente marcado como "no disponible", se actualiza su estado a "disponible".

**DeleteAccommodationServlet:** Se hacen las mismas comprobaciones que en el CreateAccommodationServlet y se redirige a una página donde se muestran todos los datos de la habitación a borrar junto con el botón definitivo para borrar la habitación, para así evitar que el borrado se produzca de manera accidental.



## Gestión de reservas

Esta funcionalidad cuenta con varias partes, en primer lugar, una lista de reservas que actúa como histórico, para acceder a ella se despliega el botón con el nombre del usuario.



## Reservas y viajes

## Caceres



## NH Oquendo

Plaza San Juan

Hotel Palacio

Precio: 100€

## Caceres



## NH Oquendo

Plaza San Juan

Hotel Palacio

Precio: 100€

## Caceres



## Atrio Resort

Plaza de San Mateo

Hotel 5 estrellas de lujo

Precio: 100€

De esta parte se encarga el **BookingsServlet**, este recupera de la base de datos todas las reservas que ha efectuado el usuario, para ello se accede a la tabla bookings, a raíz de cada una de estas reservas se busca en la tabla de bookingsAccommodations y una vez se tiene una de las habitaciones de la reserva (ya que solo se pueden reservar cada vez habitaciones de una misma propiedad), se puede buscar la propiedad a la que pertenece para recuperar el nombre.

Las reservas se llevan a cabo en la página de detalles de un alojamiento en específico.

Tipo de alojamiento	Precio de la habitación	Seleccionar habitaciones
<b>Doble Superior Premium Plus</b> Doble para uso compartido <b>Quedan 12 habitaciones en nuestra web</b> ✓ Wifi ✓ Spa ✓ Balcón	€ 100 Incluye impuestos y cargos	<input type="text" value="0"/>
<b>Suite</b> Suite de lujo <b>Quedan 23 habitaciones en nuestra web</b> ✓ Wifi ✓ Spa ✓ Balcón	€ 200 Incluye impuestos y cargos	<input type="text" value="0"/>
<button>Reservar</button>		

De esta parte se encarga el doPost del **PropertyDetailsServlet**. Lo que hace es guardar en un mapa la habitación junto con el número que ha decidido reservar, siempre que este sea mayor de 1.

Las reservar no se efectúan directamente al pulsar en este botón de reservar, si no que se ha incluido una página intermedia que hace el trabajo de carrito, de esto se encarga el **ShoppingCartServlet**.



**NH Oquendo**  
Plaza San Juan  
**4.5** Fabuloso - 4559 comentarios

**Los datos de tu reserva**  
Has seleccionado:  

Doble Superior

1 Habitaciones

Suite

2 Habitaciones

**Desglose del precio**  

**Total****€ 500**  
Incluye impuestos y cargos

**Información sobre el precio**  

Incluye € 50.0 de impuestos y cargos

10% de IVA // € 50.0

Reservar

Este carrito sirve para ver a modo de resumen las habitaciones que se han escogido así como su cantidad y precio total, también permite eliminar un conjunto de habitaciones si al final no se quiere reservar estas. Al pulsar el botón de reservar se efectúan las correspondientes entradas en las tablas de bookings, bookingsAccommodations, así como de reducir el número disponible de habitaciones para aquellas habitaciones que se hayan reservado. Relacionado con esto anterior, si se diera el caso de que se han reservado las últimas habitaciones de una propiedad, esta se pone a no disponible.

Esta funcionalidad del carrito no forma parte del ejercicio base necesario para llevar a cabo las reservas, ya que estas se podrían hacer directamente desde el **PropertyDetailsServlet** al pulsar en el botón de reservar, es por esto que lo consideramos un **requisito extra**.

## Valoración

Las distintas propiedades tienen una sección en la que se pueden ver comentarios y notas que ha puesto la gente.

Aquí se pueden dar dos situaciones, por un lado, puede que solo aparezcan opiniones de otras personas, esto es porque el dueño de una propiedad no tiene la posibilidad de puntuarse a sí mismo, este control se lleva a cabo en el jsp de **DetalleAlojamiento** y lo

que hace si el id de usuario de la sesión coincide con el de la propiedad, entonces no muestra la sección de comentar, solo los comentarios de otras personas.

**Opiniones de otros usuarios**

Usuario: Toño

Calificación: 4

Comentario: Buen sitio

Si, por el contrario, no se es el dueño de la propiedad, se muestra la zona para comentar.

Deja tu opinión

Calificación

5

Comentario

Excelente también

Modificar review

**Opiniones de otros usuarios**

Con respecto a esta sección se pueden dar de nuevo tres casos, el primero es que esté vacío debido a que el usuario en cuestión nunca ha hecho una review a la propiedad, el segundo es que ya haya hecho una, entonces aparecerá rellena y por último se puede querer eliminar la review.

De todo esto se encarga el **ReviewServlet** y el doGet del **PropertyDetailsServlet**.

El primero es el que se encarga de la funcionalidad, es decir, de añadir una review a la base de datos, de actualizarla o de eliminarla. También se encarga de modificar la media del establecimiento en función de las reviews.

Con respecto a esta media cabe destacar que las estrellas de la propiedad varían de forma dinámica en función de la nota que tenga esa propiedad.



**NH Oquendo**

★★★★☆

Caceres a 0.2km del centro

**Hotel Palacio**  
Permite mascotas  
**Disponible al menos una habitación**

**Fabuloso 4.5**  
655655655  
[Ver disponibilidad >](#)



**Hotel Don Manuel**

★★★★☆

Caceres a 0.5km del centro

**Hotel urbano spa**  
No permite mascotas  
**No disponible en el momento**

**Fabuloso 4.0**  
123123123  
[Ver disponibilidad >](#)

El segundo es el que se encarga de recuperar las reviews de una determinada propiedad y rellenarlas en la página de detalles de esta. Para diferenciar la review del usuario de la sesión de las otras reviews de esa propiedad, se carga la lista de reviews y se va iterando hasta que el idu (usuario que ha hecho la review) coincide con el id del usuario, entonces esa review se mete en una variable aparte que se manda por la request, se elimina de la lista de reviews de la propiedad y esta lista se envía como otro atributo de la request.




## Alojamientos favoritos

Para controlar los alojamientos favoritos de un usuario se ha creado un servlet llamado **FavoritesPropertiesServlet**.

**FavoritesPropertiesServlet:** Este servlet se distingue de otros por su dualidad en las funciones *doGet* y *doPost*. En el *doGet*, recorre la lista de alojamientos favoritos del usuario, y se redirige a una página (*AlojamientosUsuario.jsp*) donde se muestra esta lista con la información correspondiente de cada alojamiento.


Por otro lado, en la función *doPost* se hacen las operaciones necesarias para la página *AlojamientosUsuario.jsp* (cuando es invocada desde **FavoritesPropertiesServlet**), así como la operación que modifica la base de datos que se llama desde “*Alojamientos.jsp*” al hacer click al corazón de algún alojamiento.

Si el post se llama desde “*Alojamientos.jsp*” comprueba si el alojamiento seleccionado se encuentra como favorito dentro de la tabla **userFavoritesProperties** y se comprueba el estado del corazón (si se pasa marcado o desmarcado), si el corazón se marca el alojamiento es añadido a la tabla con su idu e idp correspondientes, si se desmarca el alojamiento se elimina de la tabla de favoritos, en ambos casos se vuelve a redirigir a la página “*Alojamientos.jsp*” manteniendo los filtros elegidos desde la misma y con la información ya actualizada de alojamientos favoritos.

✓	 userFavoritesProperties	CREATE TABLE "userFavoritesP
	 idu	INTEGER "idu" INTEGER NOT NULL
	 idp	INTEGER "idp" INTEGER NOT NULL

**AlojamientosUsuario.jsp** lo comparten varios servlets, se cambia lo mostrado según el valor del CheckType.


**Booking.com**

 Carrito

Chema ▾

### Mis alojamientos favoritos

#### Caceres




**Hotel Don Manuel**  
Plaza Concordia  
123123123  
a 0.5km del centro

Ver alojamiento

Eliminar de favoritos

#### Caceres



**NH Oquendo**  
Plaza San Juan  
655655655  
a 0.2km del centro

Ver alojamiento

Eliminar de favoritos

**Alojamientos.jsp** se invoca a la función doPost del servlet al hacer click en el ícono del corazón para agregar el alojamiento a la lista de favoritos. Una vez añadido vuelve a esta página con el alojamiento añadido/eliminado de favoritos. Además, se conservan los filtros aplicados previamente en la página, tales como ordenar por valoración, todos, con disponibilidad o sin disponibilidad.

Buscar

☒ Todos

☐ Con disponibilidad

☐ Sin disponibilidad

## Ordenar por valoración

[Ver disponibilidad >](#)[Ver disponibilidad >](#)[Ver disponibilidad >](#)

**AlojamientosServlet:** Este servlet se mostrar los alojamientos encontrados que cumplan con las coincidencias del campo “destino” que engloba localidad, descripción y título de los alojamientos existentes en la base de datos. En base al destino, y otros filtros, como por ejemplo el de encontrar alojamientos según su disponibilidad (por defecto muestra todos, estén o no disponibles), muestra la información de los alojamientos.



## Completo/disponible

Si un alojamiento se queda sin habitaciones disponibles, éste indicará que no acepta más reservas y que por lo tanto no se encuentra disponible. Por defecto se buscarán todos los alojamientos, pero se le da al usuario la opción de poder filtrar los alojamientos según su disponibilidad.

La disponibilidad de un alojamiento viene determinada por las habitaciones disponibles, si en el alojamiento al menos una de sus habitaciones está disponible se mostrará como disponible. Por el contrario, si el alojamiento no tiene ninguna habitación disponible para reservar, se mostrará como no disponible.

Para ello se implementa esta función en **AlojamientosServlet**.

**AlojamientosServlet:** Como se ha comentado en el punto anterior, este servlet facilita varias funcionalidades, incluida la capacidad de filtrar los alojamientos según su disponibilidad. Recibe un parámetro de “*Alojamientos.jsp*” que le permite manejar la lista de alojamientos a mostrar según la opción de filtrado seleccionada.

Para lograr esto, el servlet distingue entre tres casos principales: “*Todos*”, “*Con disponibilidad*” y “*Sin disponibilidad*”. Dependiendo de la opción elegida por el usuario, el servlet devuelve una lista apropiada de alojamientos utilizando las diferentes funciones en el DAO de alojamientos.