# Informe de la Aplicación

Hotel- Booking

**Grupo 15**

Guillermo Garrido Contreras

Manuel Revuelta Reyes

Andrea García Castillo

## Índice

Contenido

[Informe de la Aplicación 1](#_Toc168501732)

[Índice 2](#_Toc168501733)

[Introducción 3](#_Toc168501734)

[Objetivos del proyecto 4](#_Toc168501735)

[Esquema funcional 5](#_Toc168501736)

[Modelo Entidad-Relación 6](#_Toc168501737)

[Diagrama de Casos de uso 7](#_Toc168501738)

[Especificaciones Casos de uso 8](#_Toc168501739)

[Especificaciones de Diseño 20](#_Toc168501740)

[Diagramas de Navegación 20](#_Toc168501741)

[Casos de Prueba 20](#_Toc168501742)

[Otros aspectos 20](#_Toc168501743)

[Credenciales 20](#_Toc168501744)

## Introducción

El presente documento constituye el Plan del Proyecto “Hotel Booking”, que será un portal de reserva de habitaciones en hoteles.

Este sistema permitirá la reserva de habitaciones por parte de los clientes los cuales podrán hacer una o varias reservas en el mismo o en diferentes hoteles, así como un usuario administrador que dará soporte a la página para crear nuevos hoteles o darlos de baja en caso de que sea necesario.

A continuación, se detalla la organización del proyecto, la metodología de trabajo a seguir y demás aspectos relevantes.

## Objetivos del proyecto

|  |  |
| --- | --- |
| OBJ-001 | Gestión de clientes |
| Versión | 01 |
| Autores | Grupo 15 |
| Fuente | Reunión con los stakeholders |
| Descripción | El sistema deberá permitir el registro de usuarios, así como la gestión de sus propios datos personales y reservas |
| Importancia | Alta |
| Estado | Aprobado |
| Comentarios | -- |

|  |  |
| --- | --- |
| OBJ-002 | Gestión de hoteles |
| Versión | 01 |
| Autores | Grupo 15 |
| Fuente | Reunión con los stakeholders |
| Descripción | El Sistema gestionará por medio de un usuario admin la gestión de los hoteles y controlará cuales están disponibles para reservas |
| Importancia | Alta |
| Estado | Aprobado |
| Comentarios | -- |

|  |  |
| --- | --- |
| OBJ-003 | Gestión de habitaciones |
| Versión | 01 |
| Autores | Grupo 15 |
| Fuente | Reunión con los stakeholders |
| Descripción | El Sistema gestionará por medio de un usuario admin la gestión de las habitaciones y controlará cuales están disponibles para reservas |
| Importancia | Alta |
| Estado | Aprobado |
| Comentarios | -- |

|  |  |
| --- | --- |
| OBJ-004 | Gestión de reservas |
| Versión | 01 |
| Autores | Grupo 15 |
| Fuente | Reunión con los stakeholders |
| Descripción | El Sistema gestionará por medio de un usuario admin la gestión de las reservas |
| Importancia | Alta |
| Estado | Aprobado |
| Comentarios | -- |

## Esquema funcional

Arquitectura del sistema

El sistema se verá desplegado según las especificaciones del diagrama que se muestra a continuación

Diagrama

Descripción generada automáticamente

Figura 1: Diagrama de despliegue de la aplicación.

El sistema se compone de las siguientes partes: un servidor con la aplicación web, un equipo cliente que solicita información al servidor y un servicio de hibernate.

El equipo del cliente solo necesitará un navegador para interactuar con el servidor. Nuestra base de datos se ha implementado mediante Hibernate.

El componente JRE nos permite ejecutar aplicaciones Java.

CSS ha sido creado por nosotros para que la aplicación tenga una configuración única y exclusiva.

Se usará la arquitectura de tres capas: Presentación, Negocio y Datos.

A continuación, se muestra un desglose de las tecnologías usadas en ellas:

Diagrama

Descripción generada automáticamente

Figura 2: Diagrama de componentes de arquitectura software.

## Modelo Entidad-Relación

## Diagrama de Casos de uso

Diagrama, Escala de tiempo

Descripción generada automáticamente

## Especificaciones Casos de uso

CU-001. Login.

|  |  |
| --- | --- |
| CU-001 | Login |
| Descripción | Un usuario se loga en el sistema |
| Actores | Usuario registrado |
| Precondición | Debe estar registrado ya en el sistema |
| Postcondición | El usuario queda logado y se guarda en sesión |
| Puntos de extensión | |
| Dependencias | |
| Flujo normal | 1. El Usuario entra en el sitio web y selecciona la opción de login.  2. El sistema pide que se introduzca un nombre de usuario y su contraseña.  3. El Usuario introduce los datos solicitados.  4. El Sistema reconoce al Usuario y dependiendo del rol del usuario le redirige a una vista u otra.  5. El Usuario accede al Sistema. |
| Flujo alternativo | 4.- El nombre de usuario introducido no es válido |
| 3.- El usuario deja algún campo vacío | |

CU-002. Logout.

|  |  |
| --- | --- |
| CU-002 | Logout |
| Descripción | Un usuario cierra sesión en el sistema |
| Actores | Usuario registrado y logado |
| Precondición | Debe estar logado ya en el sistema |
| Postcondición | El sistema elimina la sesión y la cierra |
| Puntos de extensión | |
| Dependencias | |
| Flujo normal | 1. El Usuario navega por el sitio web y selecciona la opción de logout.  2. El sistema finaliza la sesión del usuario |
| Flujo alternativo |  |

CU-003. Alta usuario.

|  |  |
| --- | --- |
| CU-003 | Alta usuario. |
| Descripción | Un usuario quiere registrarse en el sistema |
| Actores | Usuario no registrado en el sistema previamente |
| Precondición |  |
| Postcondición | El sistema registra el nuevo usuario en la base de datos |
| Puntos de extensión | |
| Dependencias | |
| Flujo normal | 1. El usuario accede al formulario de registro  2. El usuario introduce los datos requeridos  3. El sistema valida los datos introducidos  4. El sistema registra el usuario y le da acceso al sistema dependiendo del rol |
| Flujo alternativo | 3-. El usuario introduce datos inválidos para el sistema |

CU-004. Listar Hotel.

|  |  |
| --- | --- |
| CU-004 | Listar Hotel. |
| Descripción | El sistema muestra el listado de hoteles disponibles |
| Actores | Usuario logado en el sistema |
| Precondición | Debe haber iniciado sesión previamente |
| Postcondición | El sistema muestra los datos de los hoteles |
| Puntos de extensión | |
| Dependencias | |
| Flujo normal | 1. El sistema le muestra los hoteles |
| Flujo alternativo | 1-. No hay hoteles registrados |

CU-005. Crear Hotel.

|  |  |
| --- | --- |
| CU-005 | Crear Hotel. |
| Descripción | El usuario administrador registra un nuevo hotel en el sistema. |
| Actores | Usuario logado en el sistema con rol de administrador |
| Precondición | Debe haber iniciado sesión previamente y ser administrador |
| Postcondición | El sistema registra el nuevo hotel y lo lista. |
| Puntos de extensión | |
| Dependencias | |
| Flujo normal | 1. El usuario administrador pulsa el botón “Crear hotel” 2. El sistema le lleva a un formulario de alta con los datos necesarios para dar de alta el hotel 3. El usuario introduce los datos 4. El sistema valida que los datos 5. El sistema añade el nuevo hotel al sistema |
| Flujo alternativo | 4. El usuario no rellena todos los campos  4. El usuario introduce valores no permitidos |

CU-006. Modificar Hotel.

|  |  |
| --- | --- |
| CU-006 | Modificar Hotel. |
| Descripción | El usuario administrador desea modificar los datos de un hotel |
| Actores | Usuario logado en el sistema con rol de administrador |
| Precondición | Debe haber iniciado sesión previamente y ser administrador |
| Postcondición | El sistema actualiza los datos del Hotel |
| Puntos de extensión | |
| Dependencias | |
| Flujo normal | 1. El usuario administrador pulsa el botón “Modificar Hotel” 2. El sistema le muestra los datos del hotel que desea modificar 3. El usuario modifica los datos 4. El sistema valida que los datos 5. El sistema actualiza los datos del hotel |
| Flujo alternativo | 4. El usuario no rellena todos los campos  4. El usuario introduce valores no permitidos |

CU-007. Eliminar Hotel.

|  |  |
| --- | --- |
| CU-007 | Eliminar Hotel. |
| Descripción | El usuario administrador desea eliminar un hotel |
| Actores | Usuario logado en el sistema con rol de administrador |
| Precondición | Debe haber iniciado sesión previamente y ser administrador |
| Postcondición | El sistema elimina el hotel de la base de datos |
| Puntos de extensión | |
| Dependencias | |
| Flujo normal | 1. El usuario administrador pulsa el botón “Eliminar Hotel” 2. El sistema elimina el hotel y sus datos relacionados 3. El sistema actualiza los datos de la base de datos |
| Flujo alternativo |  |

CU-008. Listar Habitación.

|  |  |
| --- | --- |
| CU-008 | Listar Habitación. |
| Descripción | El sistema muestra un listado de habitaciones asociadas a un hotel |
| Actores | Usuario logado en el sistema |
| Precondición | Debe haber iniciado sesión previamente |
| Postcondición | El sistema muestra un listado de las habitaciones asociadas a un hotel |
| Puntos de extensión | |
| Dependencias | |
| Flujo normal | 1. El usuario pulsa el botón listar habitación de un hotel 2. El sistema muestra listado de habitaciones |
| Flujo alternativo | 1. El hotel no tiene habitaciones para mostrar |

CU-009. Crear Habitación.

|  |  |
| --- | --- |
| CU-009 | Crear Habitación |
| Descripción | El usuario administrador registra una nueva habitación en el sistema |
| Actores | Usuario logado en el sistema con rol de administrador |
| Precondición | Debe haber iniciado sesión previamente y ser administrador |
| Postcondición | El sistema registra una nueva habitación sobre el hotel seleccionado en la base de datos |
| Puntos de extensión | |
| Dependencias | |
| Flujo normal | 1. El usuario administrador pulsa el botón “Crear habitación” 2. El sistema le lleva a un formulario de alta con los datos necesarios para dar de alta la habitación 3. El usuario introduce los datos 4. El sistema valida que los datos 5. El sistema añade una nueva habitación en la base de datos |
| Flujo alternativo | 4. El usuario no rellena todos los campos  4. El usuario introduce valores no permitidos |

CU-010. Modificar Habitación.

|  |  |
| --- | --- |
| CU-010 | Modificar Habitación. |
| Descripción | El usuario administrador desea modificar los datos de la habitación |
| Actores | Usuario logado en el sistema con rol de administrador |
| Precondición | Debe haber iniciado sesión previamente y ser administrador |
| Postcondición | El sistema actualiza los datos de la habitación |
| Puntos de extensión | |
| Dependencias | |
| Flujo normal | 1. El usuario administrador pulsa el botón “Modificar Habitación” 2. El sistema le muestra los datos de la habitación a modificar. 3. El usuario modifica los datos 4. El sistema valida que los datos 5. El sistema actualiza los datos de la habitación. |
| Flujo alternativo | 4. El usuario no rellena todos los campos  4. El usuario introduce valores no permitidos |

CU-011. Eliminar Habitación.

|  |  |
| --- | --- |
| CU-011 | Eliminar Habitación. |
| Descripción | El usuario administrador desea eliminar una habitación |
| Actores | Usuario logado en el sistema con rol de administrador |
| Precondición | Debe haber iniciado sesión previamente y ser administrador |
| Postcondición | El sistema elimina la habitación de la base de datos |
| Puntos de extensión | |
| Dependencias | |
| Flujo normal | 1. El usuario administrador pulsa el botón “Eliminar Habitación” 2. El sistema elimina la habitación y sus datos relacionados 3. El sistema actualiza los datos de la base de datos |
| Flujo alternativo |  |

CU-012. Listar Reservas.

|  |  |
| --- | --- |
| CU-012 | Listar Reservas. |
| Descripción | El sistema muestra un listado de reservas asociadas al usuario logado |
| Actores | Usuario logado en el sistema |
| Precondición | Debe haber iniciado sesión previamente |
| Postcondición | El sistema muestra un listado de las reservas del usuario |
| Puntos de extensión | |
| Dependencias | |
| Flujo normal | 1. El usuario accede a su cuenta 2. El sistema muestra listado de sus reservas |
| Flujo alternativo | 1. El usuario no tiene reservas hechas |

CU-013. Crear Reservas.

|  |  |
| --- | --- |
| CU-013 | Crear Reserva |
| Descripción | El usuario registra una nueva reserva en el sistema |
| Actores | Usuario logado en el sistema |
| Precondición | Debe haber iniciado sesión |
| Postcondición | El sistema registra una nueva reserva sobre la habitación seleccionada en la base de datos |
| Puntos de extensión | |
| Dependencias | |
| Flujo normal | 1. El usuario pulsa el botón “Crear reserva” 2. El sistema le lleva a un formulario de alta con los datos necesarios para dar de alta la reserva 3. El usuario introduce los datos 4. El sistema valida que los datos 5. El sistema añade una nueva reserva en la base de datos |
| Flujo alternativo | 4. El usuario no rellena todos los campos  4. El usuario introduce valores no permitidos |

CU-014. Modificar Reservas.

|  |  |
| --- | --- |
| CU-014 | Modificar Reserva. |
| Descripción | El usuario desea modificar los datos de la reserva |
| Actores | Usuario logado en el sistema |
| Precondición | Debe haber iniciado sesión |
| Postcondición | El sistema actualiza los datos de la reserva |
| Puntos de extensión | |
| Dependencias | |
| Flujo normal | 1. El usuario pulsa el botón “Modificar Reserva” 2. El sistema le muestra los datos de la reserva a modificar. 3. El usuario modifica los datos 4. El sistema valida que los datos 5. El sistema actualiza los datos de la reserva. |
| Flujo alternativo | 4. El usuario no rellena todos los campos  4. El usuario introduce valores no permitidos |

CU-015. Eliminar Reservas.

|  |  |
| --- | --- |
| CU-015 | Eliminar Reserva. |
| Descripción | El usuario desea eliminar una reserva |
| Actores | Usuario logado en el sistema |
| Precondición | Debe haber iniciado sesión |
| Postcondición | El sistema elimina la reserva de la base de datos |
| Puntos de extensión | |
| Dependencias | |
| Flujo normal | 1. El usuario administrador pulsa el botón “Eliminar Reserva” 2. El sistema elimina la reserva y sus datos relacionados 3. El sistema actualiza los datos de la base de datos |
| Flujo alternativo |  |

CU-016. Listar Rol.

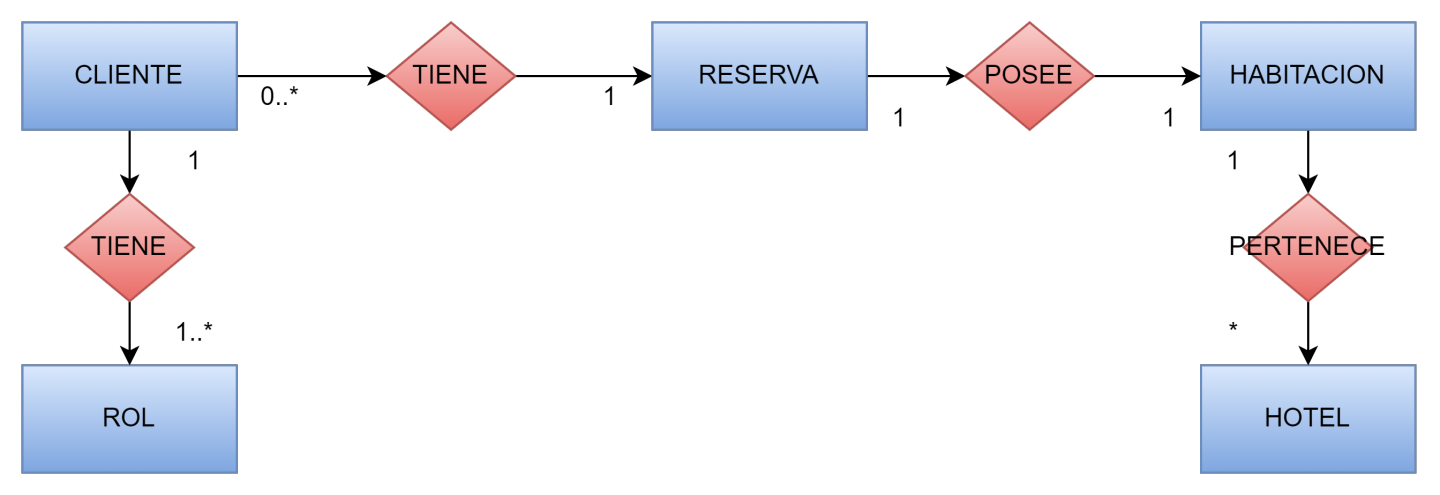
|  |  |
| --- | --- |
| CU-016 | Listar Roles. |
| Descripción | El sistema muestra un listado de roles a la hora de hacer un registro en el sistema |
| Actores | Usuario no logado en el sistema |
| Precondición |  |
| Postcondición | El sistema muestra un listado de los roles disponibles |
| Puntos de extensión | |
| Dependencias | |
| Flujo normal | 1. El usuario crea una cuenta 2. El sistema muestra listado de roles 3. El usuario elige el rol y registra sus datos en el sistema 4. El sistema registra al nuevo usuario y lo redirige a la página de login |
| Flujo alternativo |  |

## Especificaciones de Diseño

En la siguiente imagen se detalla la estructura de la base de datos, así como la relación entre las tablas

Diagrama, Tabla

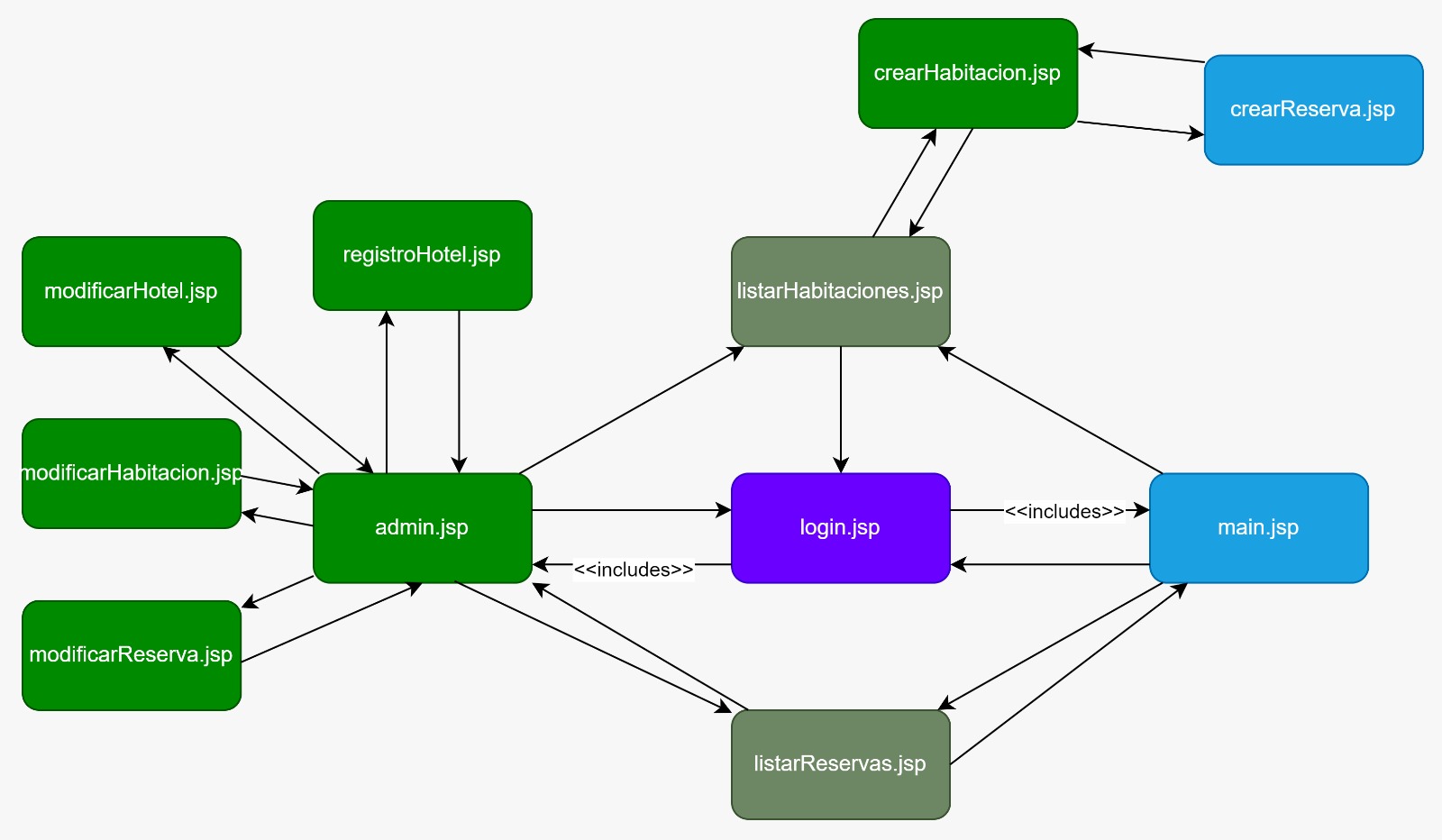
Descripción generada automáticamente



## 

## Diagramas de Navegación

Mediante este diagrama que se muestra se ilustra las distintas formas de navegación que se pueden cursar en el sistema.



## Casos de Prueba

## Otros aspectos

Hemos usado Struts2 en nuestro proyecto. Las principales ventajas de Struts2 son:

Arquitecto MVC: Una de las razones principales para elegir Struts2 es su arquitectura Modelo-Vista-Controlador (MVC). Esto nos permitirá separar claramente la lógica de negocio, la presentación y el control de flujo en nuestro proyecto. Esta organización nos facilitará el desarrollo, el mantenimiento y la escalabilidad de la aplicación.

Amplia adopción y comunidad activa: Struts2 es un framework ampliamente adoptado y ha sido utilizado en muchos proyectos a lo largo del tiempo. Esto significa que contamos con una gran cantidad de recursos disponibles, por ejemplo, documentación, tutoriales, ejemplos y foros de ayuda. Además, la comunidad de Struts2 es muy activa y brinda soporte y soluciones a problemas comunes que podríamos encontrar a lo largo del proyecto.

Integración con tecnologías existentes: Struts tiene una gran capacidad para integrarse con las tecnologías existentes en nuestro proyecto. Es compatible con una variedad de tecnologías y estándares, como JavaServer Pages (JSP), Servlet,JavaBeans, JSON y AJAX.

Manejo de formularios y validación: Struts2 nos ofrece un conjunto de etiquetas y mecanismos que simplificarán el manejo de formularios y la validación de datos. Esto ahorrará tiempo y esfuerzo al facilitar tareas comunes, como recopilar datos del usuario, validar la entrada y gestionar errores de forma eficiente.

Seguridad: La seguridad es una preocupación fundamental y Struts2 brinda funcionalidades integradas para proteger nuestra aplicación web contra amenazas comunes, como inyección de código, ataques XSS y manipulación de parámetros. El framework en sí implementa medidas de seguridad, como la validación de entradas y la prevención de ataques de inyección, lo que nos da confianza en la protección de nuestra aplicación.

## Credenciales

|  |  |  |
| --- | --- | --- |
| Usuario | Contraseña | Rol |
| Pepe | 123 | Cliente(2) |
| Paco | 123 | Administrador (1) |