



# **FACULTAD DE INGENIERÍA**

CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

## **SISTEMA DE ADMINISTRACIÓN HOTELERA**

Informe académico

**Autor(es):**

Diego Tasaico  
Neftali Zapata  
Olenka Lazo  
Mauricio Villanueva

**Curso:**

Modelamiento y Análisis de Software

**Docente:**

Jorge Alfredo Guevara Jimenez

LIMA – PERÚ

2020-2

## ÍNDICE

<b>CAPÍTULO 1. INTRODUCCIÓN .....</b>	<b>1</b>
1. Identificación del problema.....	1
2. Planteamiento de la solución.....	1
3. Cómo implementar la solución .....	1
4. Justificación y limitaciones de la investigación.....	1
5. Ventajas de la solución.....	2
6. Desventajas de la solución .....	2
7. Objetivo general .....	2
8. Objetivos específicos .....	2
<b>CAPÍTULO 2. MARCO TEÓRICO .....</b>	<b>2</b>
9. Marco teórico.....	2
10. Marco metodológico .....	4
<b>CAPÍTULO 3. DESARROLLO DE LA INVESTIGACIÓN.....</b>	<b>5</b>
11. Implementación de la solución planteada .....	5
<b>CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>134</b>
12. Conclusiones.....	134
13. Recomendaciones .....	135
14. Referencias Bibliográficas .....	136
15. Anexos .....	136

## ÍNDICE DE TABLAS

<b>Tabla 1: Proceso de Negocio .....</b>	<b>5</b>
<b>Tabla 2: Trazabilidad de Requisitos .....</b>	<b>13</b>
<b>Tabla 3: Especificación de Casos de uso - Versión 1 - 1 .....</b>	<b>52</b>
<b>Tabla 4: Especificación de Casos de uso - Versión 1 - 2.....</b>	<b>54</b>
<b>Tabla 5: Especificación de Casos de uso - Versión 1 - 3.....</b>	<b>56</b>
<b>Tabla 6: Especificación de Casos de uso - Versión 1 - 4.....</b>	<b>58</b>
<b>Tabla 7: Especificación de Casos de uso - Versión 2 - 1.....</b>	<b>60</b>
<b>Tabla 8: Especificación de Casos de uso - Versión 2 - 2.....</b>	<b>62</b>
<b>Tabla 9: Plan de pruebas Incremental - Versión 1 .....</b>	<b>65</b>
<b>Tabla 10: Plan de pruebas Incremental - Versión 2.....</b>	<b>66</b>
<b>Tabla 11: Hallazgos e Incidencias - Versión 1 .....</b>	<b>118</b>
<b>Tabla 12: Hallazgos e Incidencias - Versión 2 .....</b>	<b>119</b>
<b>Tabla 13: Modelo FURPS - 1 .....</b>	<b>126</b>
<b>Tabla 14: Modelo FURPS - 2 .....</b>	<b>133</b>

## ÍNDICE DE FIGURAS

<b>Ilustración 1: Proceso de Negocio.....</b>	<b>5</b>
<b>Ilustración 2: Proceso de Software.....</b>	<b>6</b>
<b>Ilustración 3: Diagrama de Casos de uso .....</b>	<b>7</b>
<b>Ilustración 4: Prototipo - Versión 1 - 1.....</b>	<b>7</b>
<b>Ilustración 5: Prototipo - Versión 1 - 2.....</b>	<b>8</b>
<b>Ilustración 6: Prototipo - Versión 1 - 3.....</b>	<b>8</b>
<b>Ilustración 7: Prototipo - Versión 1 - 4.....</b>	<b>9</b>
<b>Ilustración 8: Prototipo - Versión 2 - 1.....</b>	<b>9</b>
<b>Ilustración 9: Prototipo - Versión 2 - 2.....</b>	<b>10</b>
<b>Ilustración 10: Diagrama de actividades con particiones o calles.....</b>	<b>11</b>
<b>Ilustración 11: Requisitos no funcionales - Diego Tasaico - 1.....</b>	<b>14</b>
<b>Ilustración 12: Requisitos no funcionales - Diego Tasaico - 2.....</b>	<b>14</b>
<b>Ilustración 13: Requisitos no funcionales - Diego Tasaico - 3.....</b>	<b>15</b>
<b>Ilustración 14: Requisitos no funcionales - Diego Tasaico - 4.....</b>	<b>15</b>
<b>Ilustración 15: Requisitos no funcionales - Mauricio Villanueva - 1.....</b>	<b>16</b>
<b>Ilustración 16: Requisitos no funcionales - Mauricio Villanueva - 2.....</b>	<b>16</b>
<b>Ilustración 17: Requisitos no funcionales - Mauricio Villanueva - 3.....</b>	<b>17</b>
<b>Ilustración 18: Requisitos no funcionales - Abihail Zapata - 1.....</b>	<b>17</b>
<b>Ilustración 19: Requisitos no funcionales - Abihail Zapata - 2.....</b>	<b>18</b>
<b>Ilustración 20: Requisitos no funcionales - Abihail Zapata - 3.....</b>	<b>18</b>
<b>Ilustración 21: Requisitos no funcionales - Abihail Zapata - 4.....</b>	<b>19</b>
<b>Ilustración 22: Requisitos no funcionales - Olenka Lazo - 1 .....</b>	<b>19</b>
<b>Ilustración 23: Requisitos no funcionales - Olenka Lazo - 2 .....</b>	<b>20</b>
<b>Ilustración 24: Requisitos no funcionales - Olenka Lazo - 3 .....</b>	<b>20</b>
<b>Ilustración 25: Requisitos no funcionales - Olenka Lazo - 4 .....</b>	<b>21</b>
<b>Ilustración 26: Repositorio GitHub .....</b>	<b>21</b>

Ilustración 27: Repositorio GitHub - Informes - 1 .....	22
Ilustración 28: Repositorio GitHub - Informes - 2 .....	22
Ilustración 29: Repositorio GitHub - Base de Datos - 1 .....	23
Ilustración 30: Repositorio GitHub - Base de Datos - 2 .....	23
Ilustración 31: Repositorio GitHub - Base de Datos - 3 .....	24
Ilustración 32: Repositorio GitHub - Software - 1 .....	24
Ilustración 33: Repositorio GitHub - Software - 2 .....	25
Ilustración 34: Repositorio GitHub - Software - 3 .....	25
Ilustración 35: Repositorio GitHub - Software - 4 .....	26
Ilustración 36: Repositorio GitHub - Software - 5 .....	26
Ilustración 37: Repositorio GitHub - Software - 6 .....	27
Ilustración 38: Repositorio GitHub - Software - 7 .....	27
Ilustración 39: Diagrama de Casos de uso Relacionado - General .....	39
Ilustración 40: Diagrama de Casos de uso Relacionado - Versión 1 .....	40
Ilustración 41: Diagrama de Casos de uso Relacionado - Versión 2 .....	41
Ilustración 42: Código de programación del software - 1 .....	42
Ilustración 43: Código de programación del software - 2 .....	42
Ilustración 44: Código de programación del software - 3 .....	43
Ilustración 45: Código de programación del software - 4 .....	43
Ilustración 46: Código de programación del software - 5 .....	44
Ilustración 47: Código de programación del software - 6 .....	44
Ilustración 48: Código de programación del software - 7 .....	45
Ilustración 49: Código de programación del software - 8 .....	45
Ilustración 50: Código de programación del software - 9 .....	46
Ilustración 51: Código de programación del software - 10 .....	46
Ilustración 52: Código de programación del software - 11 .....	47
Ilustración 53: Código de programación del software - 12 .....	47
Ilustración 54: Código de programación del software - 13 .....	48

Ilustración 55: Código de programación del software - 14.....	48
Ilustración 56: Código de programación del software - 15.....	49
Ilustración 57: Código de programación del software - 16.....	49
Ilustración 58: Código de programación del software - 17.....	50
Ilustración 59: Código de programación del software - 18.....	50
Ilustración 60: Registro de Cliente - Datos de entrada normal - Versión 1 .....	67
Ilustración 61: Registro de Cliente - Resultado esperado normal - Versión 1 .....	67
Ilustración 62: Registro de Cliente - Datos de entrada anómalo - Versión 1 .....	68
Ilustración 63: Registro de Cliente - Resultado esperado anómalo - Versión 1 .....	68
Ilustración 64: Eliminar Cliente - Datos de entrada normal - Versión 1.....	69
Ilustración 65: Eliminar Cliente - Resultado esperado normal - Versión 1.....	69
Ilustración 66: Eliminar Cliente - Datos de entrada anómalo - Versión 1 .....	70
Ilustración 67: Eliminar Cliente - Resultado esperado anómalo - Versión 1.....	70
Ilustración 68: Modificar Cliente - Datos de entrada normal - Versión 1 .....	71
Ilustración 69: Modificar Cliente - Resultado esperado normal - Versión 1 .....	71
Ilustración 70: Modificar Cliente - Datos de entrada anómalo - Versión 1 .....	72
Ilustración 71: Modificar Cliente - Resultado esperado anómalo - Versión 1 .....	72
Ilustración 72: Consultar Cliente - Datos de entrada normal - Versión 1 .....	73
Ilustración 73: Consultar Cliente - Resultado esperado normal - Versión 1 .....	73
Ilustración 74: Consultar Cliente - Datos de entrada anómalo - Versión 1 .....	74
Ilustración 75: Consultar Cliente - Resultado esperado anómalo - Versión 1 .....	74
Ilustración 76: Registrar Trabajador - Datos de entrada normal - Versión 1.....	75
Ilustración 77: Registrar Trabajador - Resultado esperado normal - Versión 1 .....	75
Ilustración 78: Registrar Trabajador - Datos de entrada anómalo - Versión 1.....	76
Ilustración 79: Registrar Trabajador - Resultado esperado anómalo - Versión 1.....	76
Ilustración 80: Eliminar Trabajador - Datos de entrada normal - Versión 1 .....	77
Ilustración 81: Eliminar Trabajador - Resultado esperado normal - Versión 1 .....	77
Ilustración 82: Eliminar Trabajador - Datos de entrada anómalo - Versión 1 .....	78

Ilustración 83: Eliminar Trabajador - Resultado esperado anómalo - Versión 1 .....	78
Ilustración 84: Modificar Trabajador - Datos de entrada normal - Versión 1 .....	79
Ilustración 85: Modificar Trabajador - Resultado esperado normal - Versión 1 .....	79
Ilustración 86: Modificar Trabajador - Datos de entrada anómalo - Versión 1.....	80
Ilustración 87: Modificar Trabajador - Resultado esperado anómalo - Versión 1 .....	80
Ilustración 88: Consultar Trabajador - Datos de entrada normal - Versión 1.....	81
Ilustración 89: Consultar Trabajador - Resultado esperado normal - Versión 1 .....	81
Ilustración 90: Consultar Trabajador - Datos de entrada anómalo - Versión 1.....	82
Ilustración 91: Consultar Trabajador - Resultado esperado anómalo - Versión 1 .....	82
Ilustración 92: Validar Trabajador - Datos de entrada normal - Versión 1 .....	83
Ilustración 93: Validar Trabajador - Resultado esperado normal - Versión 1 .....	83
Ilustración 94: Validar Trabajador - Datos de entrada anómalo - Versión 1.....	84
Ilustración 95: Validar Trabajador - Resultado esperado anómalo - Versión 1 .....	84
Ilustración 96: Registro de entrada de cliente - Datos de entrada normal - Versión 1.....	85
Ilustración 97: Registro de entrada de cliente - Resultado esperado normal - Versión 1 .....	85
Ilustración 98: Registro de entrada de cliente - Datos de entrada anómalo - Versión 1.....	86
Ilustración 99: Registro de entrada de cliente - Resultado esperado anómalo - Versión 1....	86
Ilustración 100: Eliminar la entrada de cliente - Datos de entrada normal - Versión 1 .....	87
Ilustración 101: Eliminar la entrada de cliente - Resultado esperado normal - Versión 1 .....	87
Ilustración 102: Eliminar la entrada de cliente - Datos de entrada anómalo - Versión 1 .....	88
Ilustración 103: Eliminar la entrada de cliente - Resultado esperado anómalo - Versión 1 ...	88
Ilustración 104: Modificar la entrada de cliente - Datos de entrada normal - Versión 1 .....	89
Ilustración 105: Modificar la entrada de cliente - Resultado esperado normal - Versión 1 ....	89
Ilustración 106: Modificar la entrada de cliente - Datos de entrada anómalo - Versión 1.....	90
Ilustración 107: Modificar la entrada de cliente - Resultado esperado anómalo - Versión 1..	90
Ilustración 108: Consultar la entrada de cliente - Datos de entrada normal - Versión 1.....	91
Ilustración 109: Consultar la entrada de cliente - Resultado esperado normal - Versión 1....	91
Ilustración 110: Consultar la entrada de cliente - Datos de entrada anómalo - Versión 1 .....	92

Ilustración 111: Consultar la entrada de cliente - Resultado esperado anómalo - Versión 1	92
Ilustración 112: Registro de salida de cliente - Datos de entrada normal - Versión 1 .....	93
Ilustración 113: Registro de salida de cliente - Resultado esperado normal - Versión 1 .....	93
Ilustración 114: Registro de salida de cliente - Datos de entrada anómalo - Versión 1.....	94
Ilustración 115: Registro de salida de cliente - Resultado esperado anómalo - Versión 1.....	94
Ilustración 116: Eliminar la salida de cliente - Datos de entrada normal - Versión 1 .....	95
Ilustración 117: Eliminar de salida de cliente - Resultado esperado normal - Versión 1.....	95
Ilustración 118: Eliminar de salida de cliente - Datos de entrada anómalo - Versión 1 .....	96
Ilustración 119: Eliminar de salida de cliente - Resultado esperado anómalo - Versión 1....	96
Ilustración 120: Modificar de salida de cliente - Datos de entrada normal - Versión 1 .....	97
Ilustración 121: Modificar de salida de cliente - Resultado esperado normal - Versión 1 .....	97
Ilustración 122: Modificar de salida de cliente - Datos de entrada anómalo - Versión 1 .....	98
Ilustración 123: Modificar de salida de cliente - Resultado esperado anómalo - Versión 1 ...	98
Ilustración 124: Consultar la salida de cliente - Datos de entrada normal - Versión 1 .....	99
Ilustración 125: Consultar la salida de cliente - Resultado esperado normal - Versión 1 .....	99
Ilustración 126: Consultar la salida de cliente - Datos de entrada anómalo - Versión 1.....	100
Ilustración 127: Consultar la salida de cliente - Resultado esperado anómalo - Versión 1..	100
Ilustración 128: Registrar Habitaciones - Datos de entrada normal - Versión 2.....	101
Ilustración 129: Registrar Habitaciones - Resultado esperado normal - Versión 2.....	101
Ilustración 130: Registrar Habitaciones - Datos de entrada anómalo - Versión 2 .....	102
Ilustración 131: Registrar Habitaciones - Resultado esperado anómalo - Versión 2 .....	102
Ilustración 132: Eliminar Habitaciones - Datos de entrada normal - Versión 2 .....	103
Ilustración 133: Eliminar Habitaciones - Resultado esperado normal - Versión 2 .....	103
Ilustración 134: Eliminar Habitaciones - Datos de entrada anómalo - Versión 2.....	104
Ilustración 135: Eliminar Habitaciones - Resultado esperado anómalo - Versión 2 .....	104
Ilustración 136: Modificar Habitaciones - Datos de entrada normal - Versión 2.....	105
Ilustración 137: Modificar Habitaciones - Resultado esperado normal - Versión 2 .....	105
Ilustración 138: Modificar Habitaciones - Datos de entrada anómalo - Versión 2.....	106

Ilustración 139: Modificar Habitaciones - Resultado esperado anómalo - Versión 2.....	106
Ilustración 140: Consultar Habitaciones - Datos de entrada normal - Versión 2.....	107
Ilustración 141: Consultar Habitaciones - Resultado esperado normal - Versión 2.....	107
Ilustración 142: Consultar Habitaciones - Datos de entrada anómalo - Versión 2.....	108
Ilustración 143: Consultar Habitaciones - Resultado esperado anómalo - Versión 2.....	108
Ilustración 144: Registrar Reserva - Datos de entrada normal - Versión 2 .....	109
Ilustración 145: Registrar Reserva - Resultado esperado normal - Versión 2 .....	109
Ilustración 146: Registrar Reserva - Datos de entrada anómalo - Versión 2 .....	110
Ilustración 147: Registrar Reserva - Resultado esperado anómalo - Versión 2 .....	110
Ilustración 148: Eliminar Reserva - Datos de entrada normal - Versión 2.....	111
Ilustración 149: Eliminar Reserva - Resultado esperado normal - Versión 2.....	111
Ilustración 150: Eliminar Reserva - Datos de entrada anómalo - Versión 2.....	112
Ilustración 151: Eliminar Reserva - Resultado esperado anómalo - Versión 2.....	112
Ilustración 152: Modificar Reserva - Datos de entrada normal - Versión 2.....	113
Ilustración 153: Modificar Reserva - Resultado esperado normal - Versión 2.....	113
Ilustración 154: Modificar Reserva - Datos de entrada anómalo - Versión 2 .....	114
Ilustración 155: Modificar Reserva - Resultado esperado anómalo - Versión 2 .....	114
Ilustración 156: Consultar Reserva - Datos de entrada normal - Versión 2 .....	115
Ilustración 157: Consultar Reserva - Resultado esperado normal - Versión 2 .....	115
Ilustración 158: Consultar Reserva - Datos de entrada anómalo - Versión 2 .....	116
Ilustración 159: Consultar Reserva - Resultado esperado anómalo - Versión 2 .....	116
Ilustración 160: Videoconferencia con representante de la empresa .....	136
Ilustración 161: Integrante - Diego Tasaico .....	137
Ilustración 162: Integrante - Mauricio Villanueva .....	137
Ilustración 163: Integrante - Olenka Lazo.....	138
Ilustración 164: Integrante - Abihail Zapata .....	138
Ilustración 165: Acta de compromiso.....	139
Ilustración 166: Funcionamiento del Software - Acceso al sistema.....	140

<b>Ilustración 167: Funcionamiento del Software - Menú Principal.....</b>	<b>140</b>
<b>Ilustración 168: Funcionamiento del Software - Vista de los clientes .....</b>	<b>141</b>
<b>Ilustración 169: Funcionamiento del Software - Vista de los trabajadores.....</b>	<b>141</b>
<b>Ilustración 170: Funcionamiento del Software - Vista de los movimientos .....</b>	<b>141</b>
<b>Ilustración 171: Funcionamiento del Software - Vista de las reservas.....</b>	<b>141</b>
<b>Ilustración 172: Funcionamiento del Software - Vista de las habitaciones.....</b>	<b>141</b>
<b>Ilustración 173: Funcionamiento del Software - Detección de errores .....</b>	<b>141</b>
<b>Ilustración 174: Funcionamiento del Software - Conexión a Base de Datos .....</b>	<b>141</b>
<b>Ilustración 175: Funcionamiento del Software - Tiempo de respuesta .....</b>	<b>141</b>
<b>Ilustración 176: Funcionamiento del Software - Espacio mínimo de memoria.....</b>	<b>141</b>

## CAPÍTULO 1. INTRODUCCIÓN

### 1. Identificación del problema

Actualmente el problema que presenta el hotel surge en el proceso de alquiler de habitaciones, ya que no cuenta con un sistema de administración y todo se realiza de forma manual, como el cobro por el servicio, el registro de habitaciones, la verificación de habitaciones disponibles y verificación de la cantidad de clientes que se hospedan durante un intervalo de tiempo. Este problema, si bien cumple con su objetivo que es la atención al cliente, demanda mucho tiempo, esfuerzo y malinterpretación de la información con respecto a los datos del cliente y su reserva.

### 2. Planteamiento de la solución

Para poder solucionar este problema lo que el hotel necesita es un sistema a través del cual se pueda buscar hospedaje, gestar la reserva de éste y realizar su administración, para ello se empezará con un modelo o prototipo que debe tener estas funcionalidades básicas.

### 3. Cómo implementar la solución

Se va a crear una solución de software que será usada por el usuario de una manera local para mejorar la productividad de la empresa y sus trabajadores. Para esto tomamos en cuenta las necesidades del cliente, el proceso que se lleva durante el registro y el producto final esperado del sistema desarrollado.

### 4. Justificación y limitaciones de la investigación

Limitaciones:

- Tiempo: Tendremos todo este ciclo académico para desarrollar sistema.
- Espacio: Todo el proyecto será realizado mediante sistemas digitales.
- Recursos: Tendremos que tomar en cuenta las limitaciones para la adquisición de los aparatos electrónicos.

Justificación:

- Se justifica en base a los avances tecnológicos y en su implementación en las grandes, medianas y pequeñas empresas.
- Se justifica en base a la eficacia y productividad de los trabajadores.
- Se justifica en base a la experiencia y servicio brindado a los clientes.

## 5. Ventajas de la solución

Como ventaja principal tenemos la eficacia que se generará en el registro de nuevos clientes, se podrá visualizar desde un primer momento un cambio significativo en el tiempo requerido para ello.

## 6. Desventajas de la solución

Como desventaja principal tenemos la adquisición del material necesario para la implementación del sistema, otra de las desventajas sería el gasto en capacitaciones para el personal administrativo del lugar.

## 7. Objetivo general

Desarrollar un sistema de administración hotelera, que pueda cumplir las necesidades de la empresa, contando con la posibilidad de hacer las reservas de las habitaciones y realizar el cobro respectivo al cliente.

## 8. Objetivos específicos

- Implementar un mantenimiento de clientes, donde se podrá registrar, consultar, modificar y eliminar a los clientes, incluyendo la entrada y salida del cliente.
- Implementar el mantenimiento de habitaciones y reservas.
- Permitir al administrador o recepcionista poder consultar sobre las habitaciones que fueron reservadas.

## CAPÍTULO 2. MARCO TEÓRICO

### 9. Marco teórico

- Modelo de negocio: Alonso, Martínez y Segovia (2005) señalan que el modelo de negocio es necesario para poder entender los procesos de negocio de la empresa y así poder tener una descripción detallada de los requisitos del sistema.
- Proceso de software: “Un proceso de software es una secuencia de actividades que conducen a la elaboración de un producto de software.” (Sommerville, 2011, p. 9).
- Prototipos: “permiten a los usuarios ver que tan bien el sistema apoya a su trabajo.” (Sommerville, 2011, p. 45).
- Casos de uso: “un caso de uso identifica a los actores implicados en una interacción” (Sommerville, 2011, p. 107).
- Diagrama de casos de uso: “representa todas las interacciones que se describirán en los requerimientos del sistema.” (Sommerville, 2011, p. 107).

- Requerimientos: Alonso, Martínez y Segovia (2005) señalan que los requerimientos se reconocen a partir de las especificaciones del problema que va a mencionar el usuario.
- Diagramas de actividad: “Los diagramas de actividad intentan mostrar las actividades que incluyen un proceso de sistema, así como el flujo de control de una actividad a otra.” (Sommerville, 2011, p. 123).
- Proceso unificado: Alonso, Martínez y Segovia (2005) señalan que el proceso unificado modela el sistema como un conjunto de bloques interconectados y se implementan mediante componentes.
- GitHub: Es una plataforma de desarrollo inspirada en tu forma de trabajar. Desde el código abierto hasta el negocio, puede alojar y revisar código, administrar proyectos y crear software junto con 50 millones de desarrolladores. (Git Hub, s.f.)
- Lucid chart: Ofrece una interfaz intuitiva y cientos de plantillas que permiten que tus equipos comuniquen sistemas, procesos e ideas complejos con gráficos. Eleva tu perspectiva en todas las áreas de tu negocio para impulsar ventas, simplificar la gestión del personal, mapear tu infraestructura y más. (Lucidchart, s.f.)
- NetBeans: Es una herramienta para desarrollar rápida y fácilmente aplicaciones de escritorio, móviles y web con Java, JavaScript, HTML5, PHP, C/C++ y más. (NetBeans, s.f.)
- WampServer: Es un ambiente de desarrollo web para Windows. Te permite crear aplicaciones con Apache2, PHP y la base de datos MySQL. (WampServer, s.f.)
- MySQL WorkBench: Es una herramienta visual unificada para arquitectos de base de datos, desarrolladores y DBAs. Provee herramientas de modelamiento de datos, desarrollo SQL, herramientas de administración de servidores, administración de usuarios, respaldos y mucho más. (MySQL WorkBench, s.f.)
- MySQL: Es una base de datos de código abierto. Esta 100% desarrollada, gestionada y sostenida por el equipo de MySQL. (MySQL, s.f.)
- Modelo de FURPS: El modelo FURPS+ establece cinco características como factores de calidad que son los que le dan nombre: funcionalidad, usabilidad, confiabilidad, prestación y soporte. (Modelo de Calidad Furps, 2018)
- Plan de pruebas incremental: El plan de pruebas de performance detalla lo que se debe ejecutar en el tiempo que hemos asignado para la prueba. De tal forma que se puedan responder las preguntas sobre el rendimiento del sistema, después de haber ejecutado la menor cantidad de ejecuciones posibles. (Toledo, s.f.)

## 10. Marco metodológico

- Paso 1: Formar el equipo de trabajo.
- Paso 2: Identificar una empresa.
- Paso 3: Entrevistar al administrador del negocio.
- Paso 4: Modelar y especificar el proceso de negocio.
- Paso 5: Hacer el modelo de proceso de software.
- Paso 6: Hacer el modelo de casos de uso.
- Paso 7: Creación de prototipos.
- Paso 8: Hacer el diagrama de actividades con particiones.
- Paso 9: Avance de informe
- Paso 10: Realizar la captura de los requisitos
- Paso 11: Validar los requisitos
- Paso 12: Realizar la trazabilidad de los requisitos
- Paso 13: Instalar los requisitos no funcionales
- Paso 14: Realizar el diagrama de casos de uso relacionado
- Paso 15: Especificar los casos de uso
- Paso 16: Realizar plan de pruebas incremental versión 1
- Paso 17: Realizar plan de pruebas incremental versión 2
- Paso 18: Ejecutar plan de pruebas versión 1
- Paso 19: Ejecutar plan de pruebas versión 2
- Paso 20: Especificar los hallazgos encontrados versión 1
- Paso 21: Especificar los hallazgos encontrados versión 2
- Paso 22: Realizar el modelo de FURPS+

## CAPÍTULO 3. DESARROLLO DE LA INVESTIGACIÓN

### 11. Implementación de la solución planteada

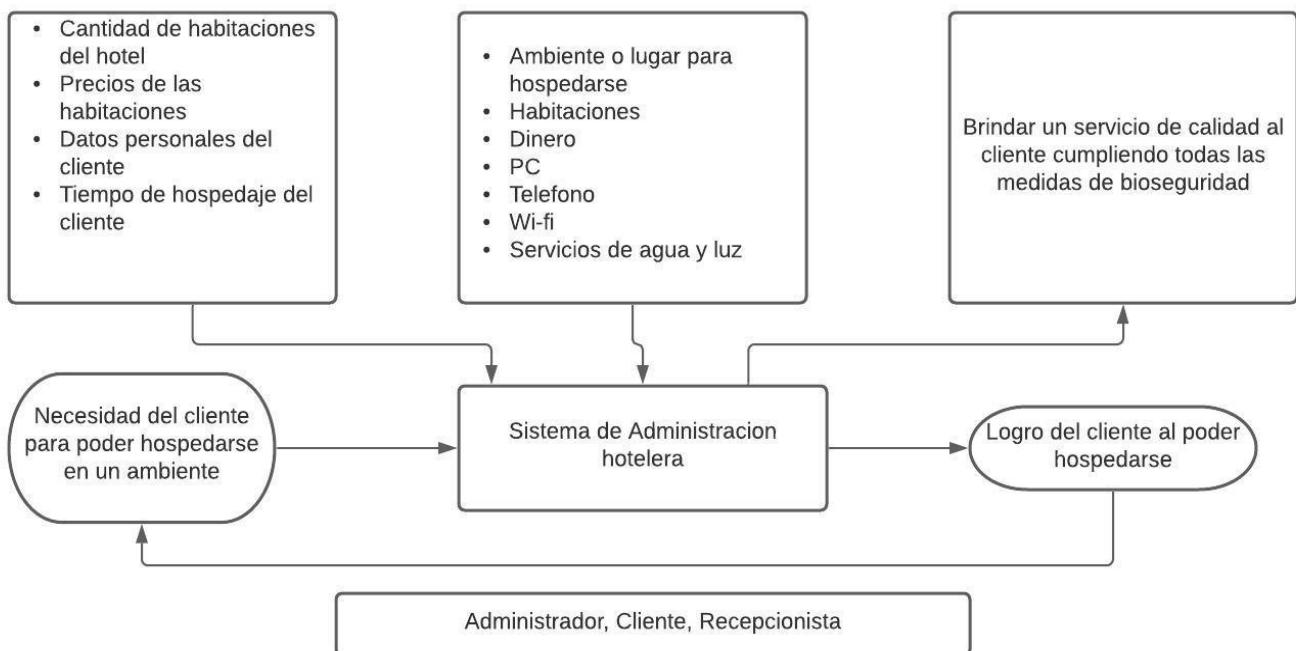


Ilustración 1: Proceso de Negocio

Procesos	Entradas	Actividades	Salidas	Valor para el cliente
Sistema de Administración hotelera	Necesidad del cliente para poder hospedarse en un ambiente	<ul style="list-style-type: none"> <li>Registro de cliente.</li> <li>Consulta de precio de habitación.</li> <li>Selección de habitación para el cliente.</li> <li>Verificar que la habitación se encuentre vacía y limpia.</li> <li>Realización del pago del cliente al hotel.</li> <li>Registrar salida del cliente del hotel.</li> </ul>	Registro de salida del cliente	Brindar un servicio de calidad al cliente cumpliendo todas las medidas de bioseguridad

Tabla 1: Proceso de Negocio

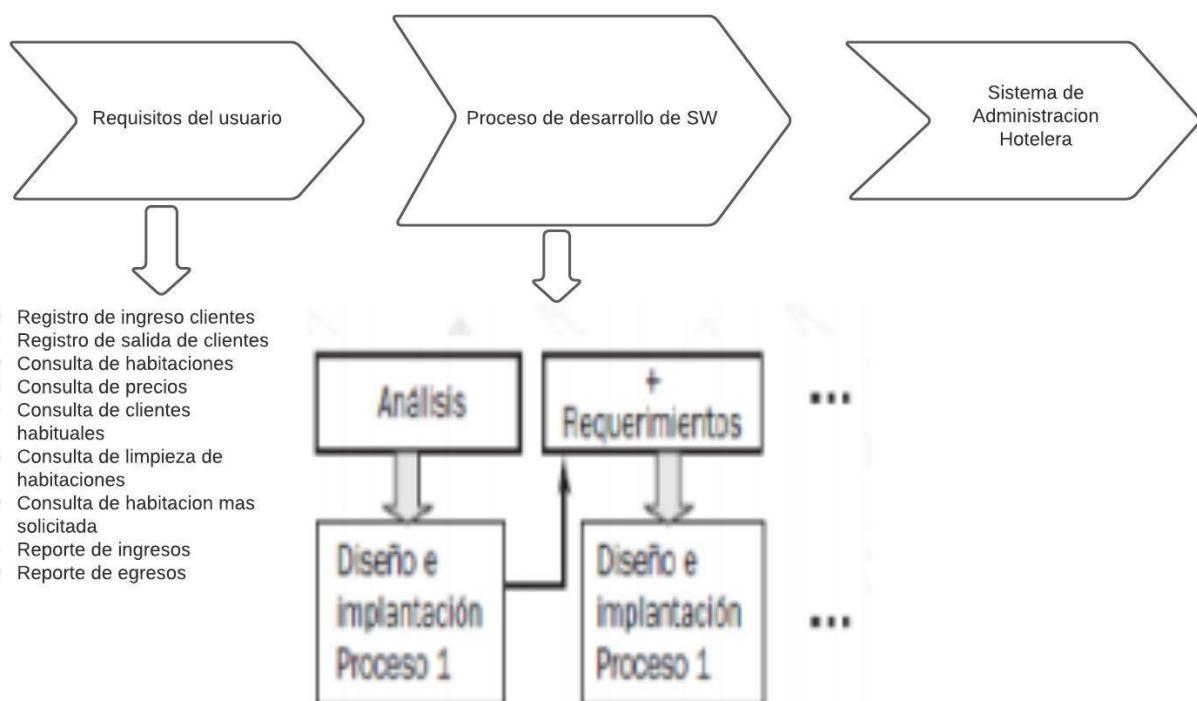


Ilustración 2: Proceso de Software

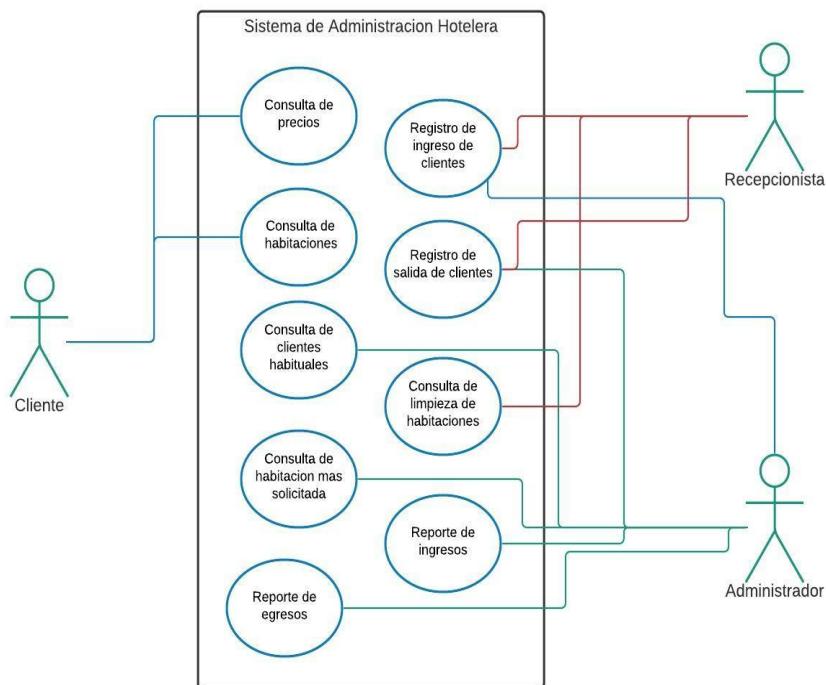


Ilustración 3: Diagrama de Casos de uso

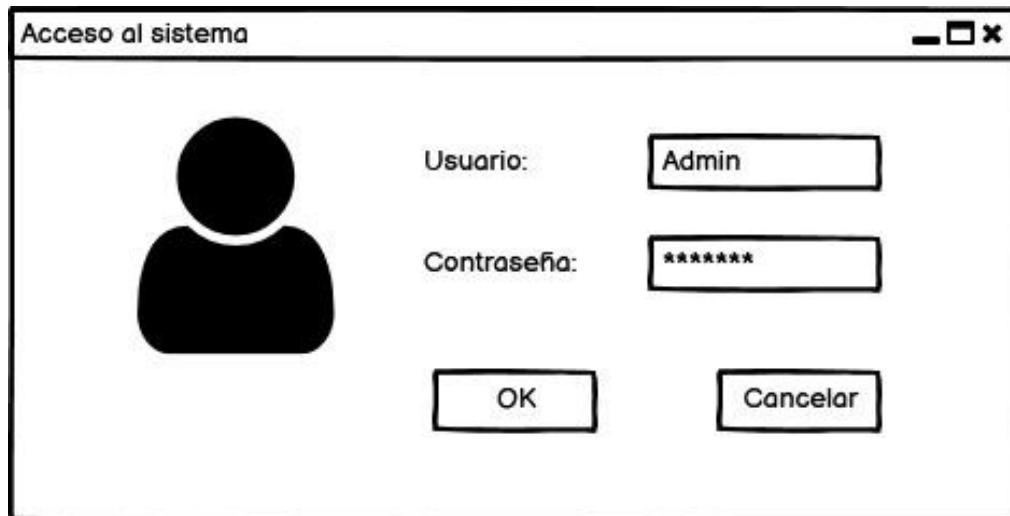


Ilustración 4: Prototipo - Versión 1 - 1

### Clientes

Registro de clientes

Código:	<input type="text"/>
Nombres:	<input type="text"/>
Apellido Paterno:	<input type="text"/>
Apellido Materno:	<input type="text"/>
Tipo de Documento:	<input type="text"/>
Número de Documento:	<input type="text"/>
Celular:	<input type="text"/>
Email:	<input type="text"/>

Opciones

Detalles

Código	Nombres	Apellido Paterno	Apellido Materno	Tipo de Documento	Nº Documento	Celular	Email
1	Mario	Aguirre	Pereyra	DNI	48189768	991642315	mario@correo.com
2	Dominga	Oropeza	Lopez	DNI	32382559	992317456	dominga@correo.co
3	Jorge	Molina	Pereyra	DNI	46328748	912118543	jorge@correo.com
4	Pablo	Nuñez	Luna	DNI	72834178	995434515	pablo@correo.com

Ilustración 5: Prototipo - Versión 1 - 2

### Trabajador

Registro de trabajadores

Código:	<input type="text"/>
Nombres:	<input type="text"/>
Apellido Paterno:	<input type="text"/>
Apellido Materno:	<input type="text"/>
Tipo de Documento:	<input type="text"/>
Número de Documento:	<input type="text"/>
Celular:	<input type="text"/>
Email:	<input type="text"/>
Sueldo:	<input type="text"/>
Acceso:	<input type="text"/>
Login:	<input type="text"/>
Password:	<input type="text"/>
Estado:	<input type="text"/>

Opciones

Detalles

Código	Nombres	Apellido Paterno	Apellido Materno	Tipo de Documento	Nº Documento	Celular	Email	Sueldo	Acceso	Login	Password	Estado
1	Pedro	Azabache	Perez	DNI	48189768	991642315	pedro@correo.com	1178	Recepcionista	pedroazab	azab481	1
2	Julio	Lopez	Martinez	DNI	32382559	992317456	julio@correo.com	2500	Administrador	juliolop	lop323	1
3	Mauricio	Quispe	Leon	DNI	46328748	912118543	mauricio@correo.co	1178	Recepcionista	mouquip	qui463	1
4	Hector	Sanchez	Luna	DNI	72834178	995434515	hector@correo.com	1178	Recepcionista	hectzan	san728	1

Ilustración 6: Prototipo - Versión 1 - 3

### Movimientos

Registro de Movimientos

Código:	<input type="text"/>
Nombres Cliente:	<input type="text"/>
Apellido Paterno:	<input type="text"/>
Apellido Materno:	<input type="text"/>
Número de habitación:	<input type="text"/>
Fecha hora Ingreso:	<input type="text"/>
Fecha hora Salida:	<input type="text"/>

Opciones

<input type="button" value="Actualizar"/>	<input type="button" value="Limpiar Campos"/>	<input type="button" value="Editar"/>	<input type="button" value="Marcar entrada"/>	<input type="button" value="Marcar salida"/>	<input type="button" value="Eliminar"/>	<input type="button" value="Listar"/>
---	---	---------------------------------------	---	--	---	---------------------------------------

Detalles

Código	Nombre	Apellido Paterno	Apellido Materno	Nro de habitación	Fecha hora Ingreso	Fecha hora Salida
1	Mario	Aguirre	Pereyra	204	2020-11-10 17:39:01	2020-11-11 20:39:01
2	Dominga	Oropeza	Lopez	102	2020-11-11 21:39:01	2020-11-12 20:39:01
3	Jorge	Molina	Pereyra	203	2020-11-11 13:00:01	2020-11-13 14:39:01
4	Pablo	Nuñez	Luna	301	2020-11-13 11:39:01	2020-11-14 20:23:01

Ilustración 7: Prototipo - Versión 1 - 4

### Habitaciones

Registro de Habitaciones

Código:	<input type="text"/>
Número de habitación:	<input type="text"/>
Piso:	<input type="text"/>
Precio:	<input type="text"/>
Tipo de habitación:	<input type="text"/>
Estado:	<input type="text"/>

Opciones

Buscar por número:	<input type="text"/>	<input type="button" value="Buscar"/>	<input type="button" value="Actualizar"/>	<input type="button" value="Eliminar"/>	<input type="button" value="Listar"/>
--------------------	----------------------	---------------------------------------	---	---	---------------------------------------

Lista de Habitaciones

Código	Número	Piso	Precio	Tipo de habitación	Estado
1	204	2	100	Individual	Limpia
2	102	1	230	Matrimonial	Limpia
3	203	2	150	Doble	Limpia
4	301	3	75	Individual	Limpia

Ilustración 8: Prototipo - Versión 2 - 1

Reserva																																																																					
Registro de reserva		Opciones																																																																			
Código de Reserva:	<input type="text"/>	Actualizar		Editor		Eliminar		Listar																																																													
Código de Cliente:	<input type="text"/>							Salir																																																													
Código de Trabajador:	<input type="text"/>																																																																				
Código de Habitación:	<input type="text"/>																																																																				
Tipo de Reserva:	<input type="text"/>																																																																				
Fecha de Reserva:	<input type="text"/> / / 	Actualizar		Editor		Eliminar		Listar																																																													
Fecha de ingreso:	<input type="text"/> / / 							Salir																																																													
Fecha de salida:	<input type="text"/> / / 																																																																				
Costo :	<input type="text"/>																																																																				
Estado:	<input type="text"/>																																																																				
<input type="button" value="Registrar"/> <input type="button" value="Limpiar Campos"/> <input type="button" value="Nuevo"/>		Detalles <table border="1"> <thead> <tr> <th>Código reserva</th> <th>Código cliente</th> <th>Código trabajador</th> <th>Código habitación</th> <th>Tipo de reserva</th> <th>Fecha de reserva</th> <th>Fecha de ingreso</th> <th>Fecha de salida</th> <th>Costo</th> <th>Estado</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>Reserva</td> <td>30/09/2020</td> <td>03/10/2020</td> <td>04/10/2020</td> <td>100</td> <td>1</td> </tr> <tr> <td>2</td> <td>2</td> <td>2</td> <td>2</td> <td>Reserva</td> <td>30/09/2020</td> <td>03/10/2020</td> <td>05/10/2020</td> <td>250</td> <td>1</td> </tr> <tr> <td>3</td> <td>3</td> <td>3</td> <td>3</td> <td>Reserva</td> <td>30/09/2020</td> <td>04/10/2020</td> <td>05/10/2020</td> <td>100</td> <td>1</td> </tr> <tr> <td>4</td> <td>4</td> <td>4</td> <td>4</td> <td>Reserva</td> <td>30/09/2020</td> <td>05/10/2020</td> <td>06/10/2020</td> <td>100</td> <td>1</td> </tr> <tr> <td>5</td> <td>5</td> <td>5</td> <td>5</td> <td>Alquiler</td> <td>30/09/2020</td> <td>05/10/2020</td> <td>07/10/2020</td> <td>250</td> <td>1</td> </tr> </tbody> </table>								Código reserva	Código cliente	Código trabajador	Código habitación	Tipo de reserva	Fecha de reserva	Fecha de ingreso	Fecha de salida	Costo	Estado	1	1	1	1	Reserva	30/09/2020	03/10/2020	04/10/2020	100	1	2	2	2	2	Reserva	30/09/2020	03/10/2020	05/10/2020	250	1	3	3	3	3	Reserva	30/09/2020	04/10/2020	05/10/2020	100	1	4	4	4	4	Reserva	30/09/2020	05/10/2020	06/10/2020	100	1	5	5	5	5	Alquiler	30/09/2020	05/10/2020	07/10/2020	250	1
Código reserva	Código cliente	Código trabajador	Código habitación	Tipo de reserva	Fecha de reserva	Fecha de ingreso	Fecha de salida	Costo	Estado																																																												
1	1	1	1	Reserva	30/09/2020	03/10/2020	04/10/2020	100	1																																																												
2	2	2	2	Reserva	30/09/2020	03/10/2020	05/10/2020	250	1																																																												
3	3	3	3	Reserva	30/09/2020	04/10/2020	05/10/2020	100	1																																																												
4	4	4	4	Reserva	30/09/2020	05/10/2020	06/10/2020	100	1																																																												
5	5	5	5	Alquiler	30/09/2020	05/10/2020	07/10/2020	250	1																																																												

Ilustración 9: Prototipo - Versión 2 - 2

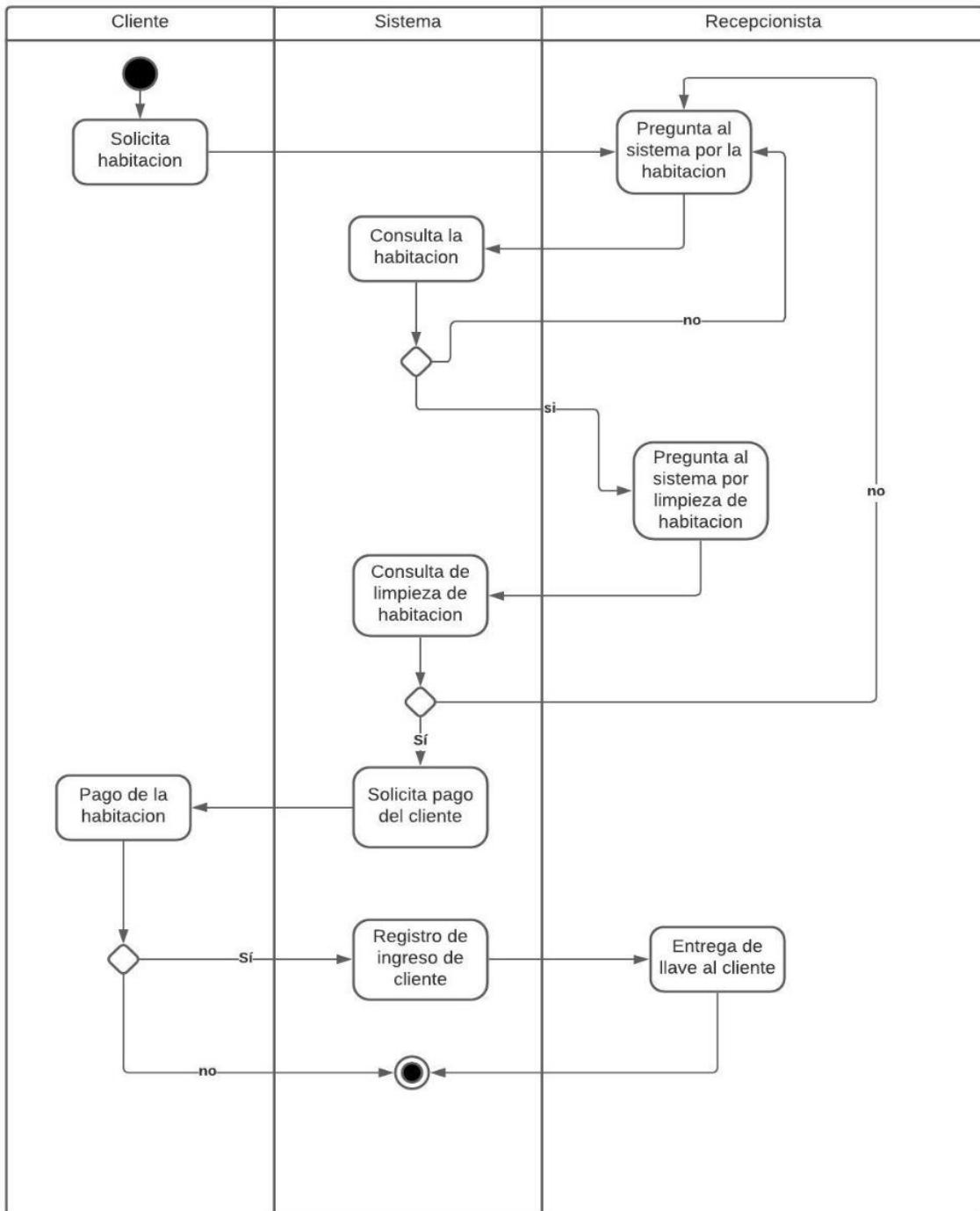


Ilustración 10: Diagrama de actividades con particiones o calles

Tipo de Requisito	ID	Requisito	Casos de Uso	Prototipo	Versión	Instalado
Funcional	1	Permite el Registro de trabajadores	Registro de Trabajador	Si	1	/
Funcional	2	Permite hacer un listado de los trabajadores existentes	Consulta de Trabajadores	Si	1	/
Funcional	3	Permite Modificar a los trabajadores registrados	Modificar al Trabajador	Si	1	/
Funcional	4	Permite Eliminar trabajadores registrados	Eliminar Trabajador	Si	1	/
Funcional	5	Verificar el usuario y acceder al sistema	Validar trabajador	Si	1	/
Funcional	6	Permite el Registro de clientes	Registro de Clientes	Si	1	/
Funcional	7	Permite hacer un listado de los clientes existentes	Consulta de Clientes	Si	1	/
Funcional	8	Permite Modificar a los clientes registrados	Modificar al Cliente	Si	1	/
Funcional	9	Permite Eliminar clientes registrados	Eliminar Cliente	Si	1	/
Funcional	10	Permite registrar la fecha de entrada del cliente.	Registro de entrada del cliente	Si	1	/
Funcional	11	Permite registrar la fecha de salida del cliente	Registro de salida del cliente	Si	1	/
Funcional	12	Permite mostrar la fecha de entrada del cliente	Consulta de entrada de los clientes	Si	1	/
Funcional	13	Permite mostrar la fecha de salida del cliente	Consulta de la salida de los clientes	Si	1	/
Funcional	14	Permite modificar la fecha de entrada del cliente	Modificar entrada del cliente	Si	1	/

Funcional	15	Permite modificar la fecha de salida del cliente	Modificar salida del cliente	Si	1	/
Funcional	16	Permite eliminar la fecha de entrada del cliente	Eliminar la entrada del cliente	Si	1	/
Funcional	17	Permite eliminar la fecha de salida del cliente	Eliminar la salida del cliente	Si	1	/
Funcional	18	Permite registrar habitaciones	Registro de habitaciones	Si	2	/
Funcional	19	Saber el estado de las habitaciones	Consulta de habitaciones	Si	2	/
Funcional	20	Permite modificar habitaciones	Modificar habitaciones	Si	2	/
Funcional	21	Permite eliminar habitaciones	Eliminar habitaciones	Si	2	/
Funcional	22	Permite registrar reservas	Registro de reservas	Si	2	/
Funcional	23	Saber el estado de las reservas	Consulta de reservas	Si	2	/
Funcional	24	Permite modificar reservas	Modificar reservas	Si	2	/
Funcional	25	Permite eliminar reservas	Eliminar reservas	Si	2	/
Funcional	26	Saber los clientes habituales	Consulta de clientes habituales	No	3	/
No funcional	27	NetBeans	/	No	/	Si
No funcional	28	MySQL o WampServer	/	No	/	Si
No funcional	29	MySQL WorkBench	/	No	/	Si

Tabla 2: Trazabilidad de Requisitos

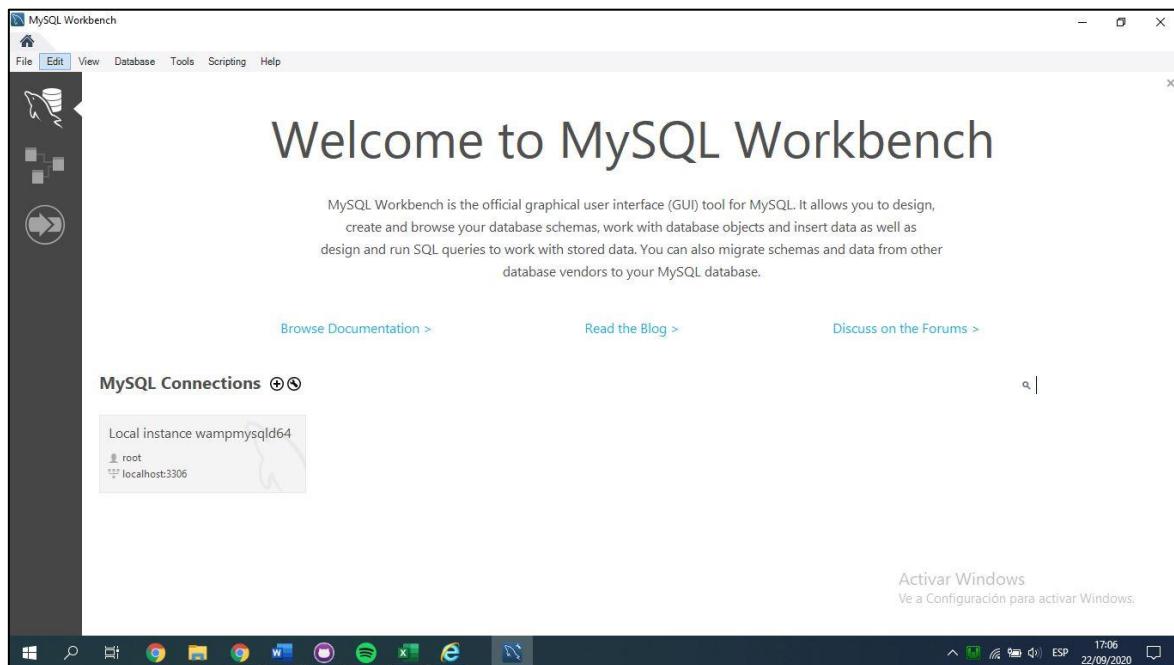


Ilustración 11: Requisitos no funcionales - Diego Tasaico - 1

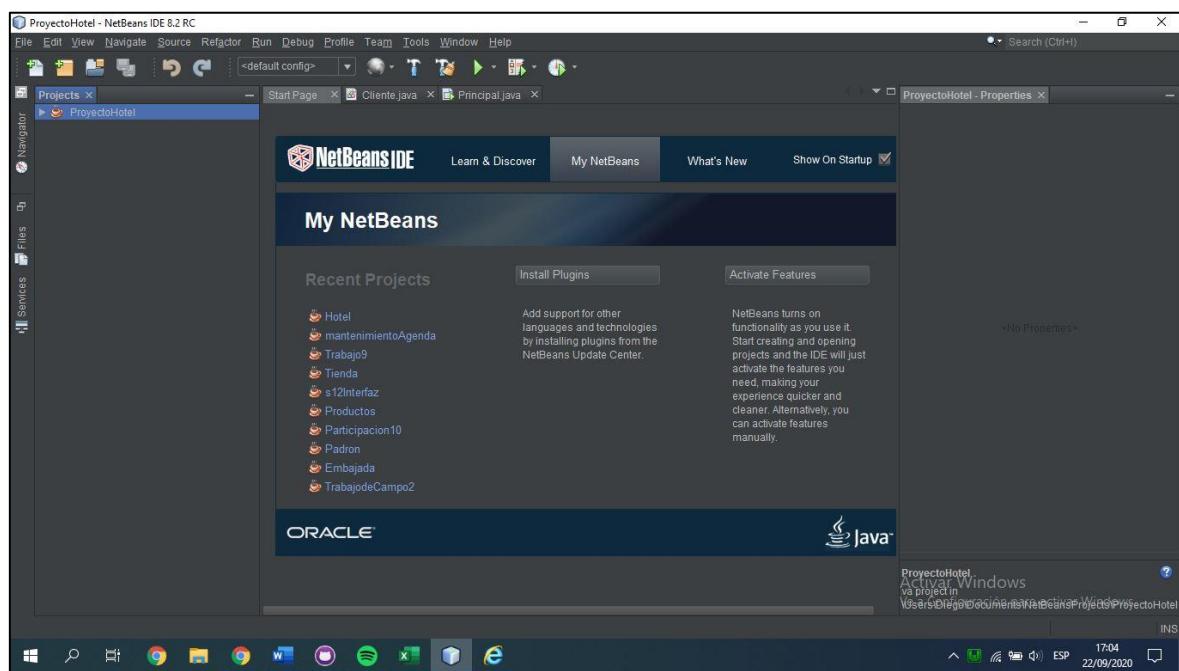


Ilustración 12: Requisitos no funcionales - Diego Tasaico - 2

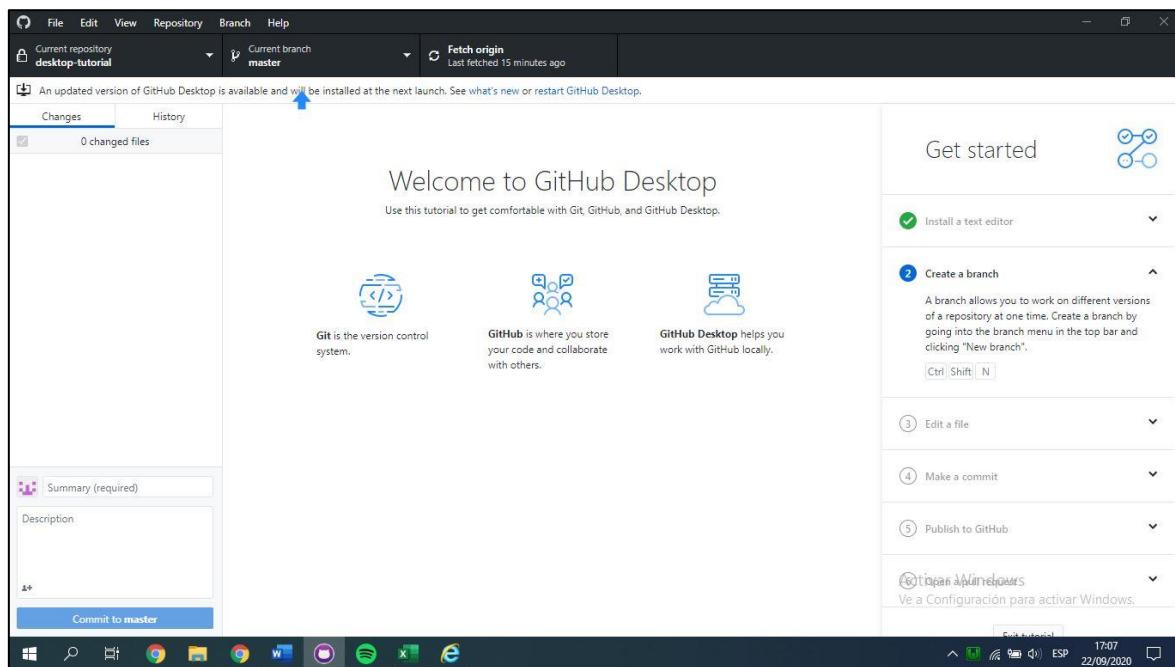


Ilustración 13: Requisitos no funcionales - Diego Tasaico - 3

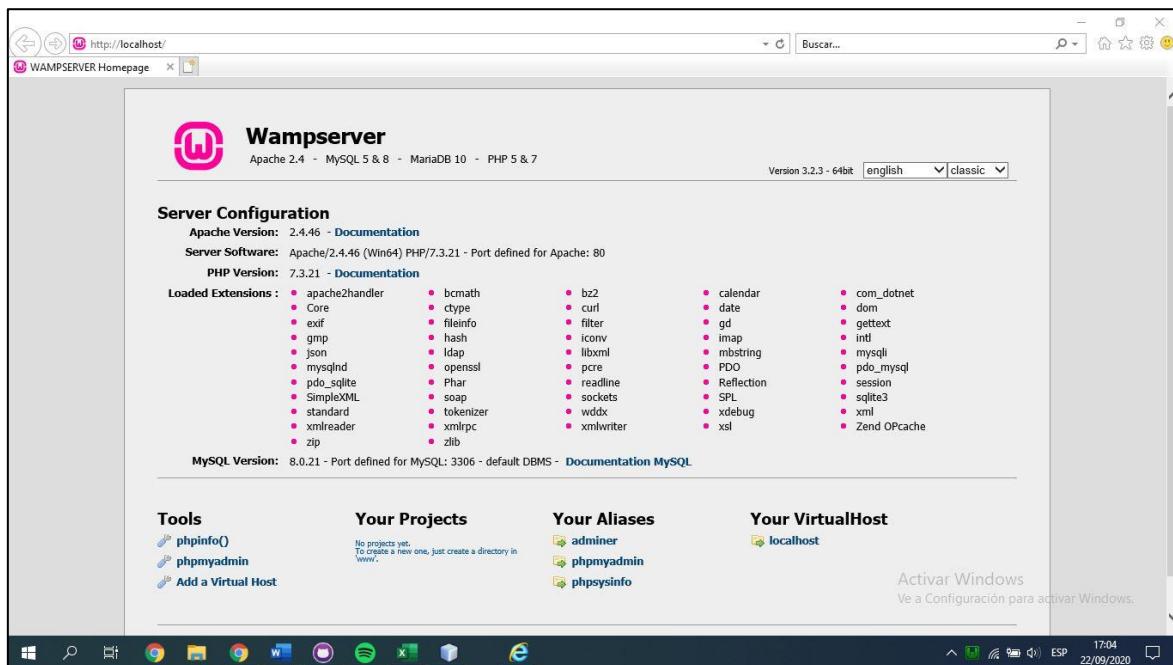


Ilustración 14: Requisitos no funcionales - Diego Tasaico - 4

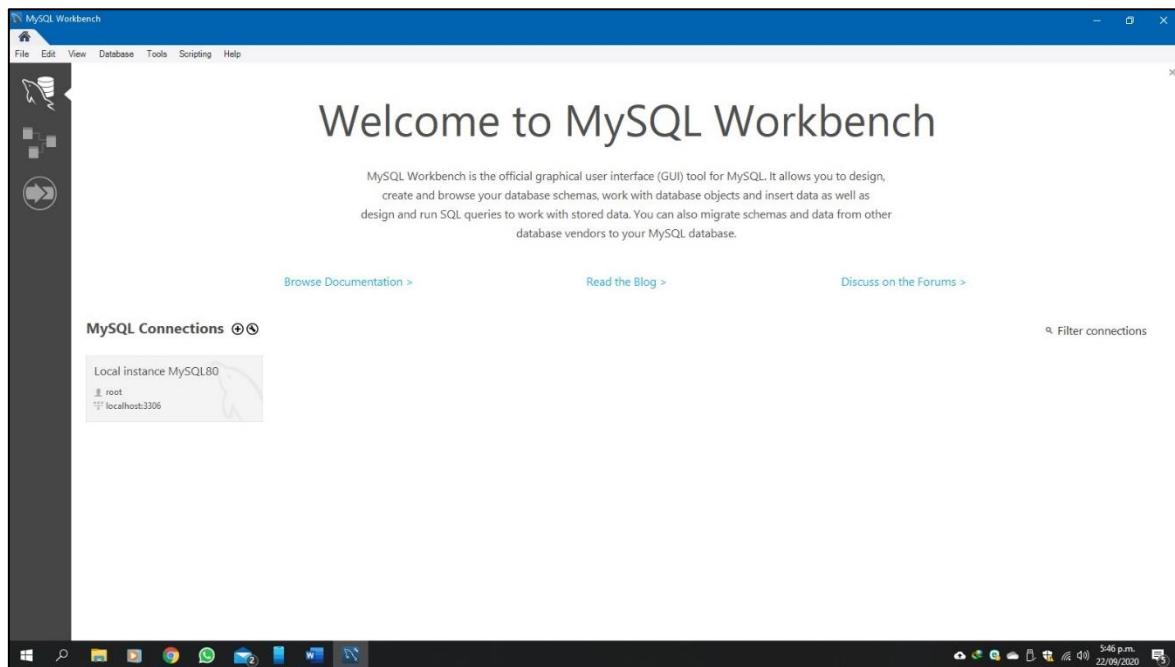


Ilustración 15: Requisitos no funcionales - Mauricio Villanueva - 1

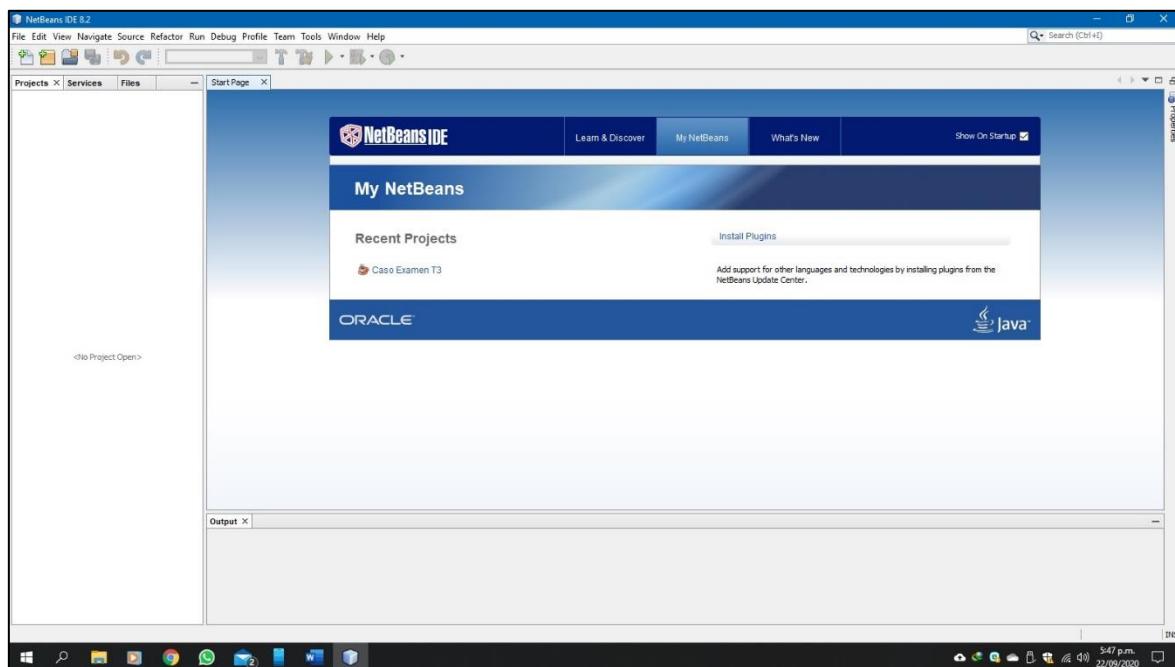


Ilustración 16: Requisitos no funcionales - Mauricio Villanueva - 2

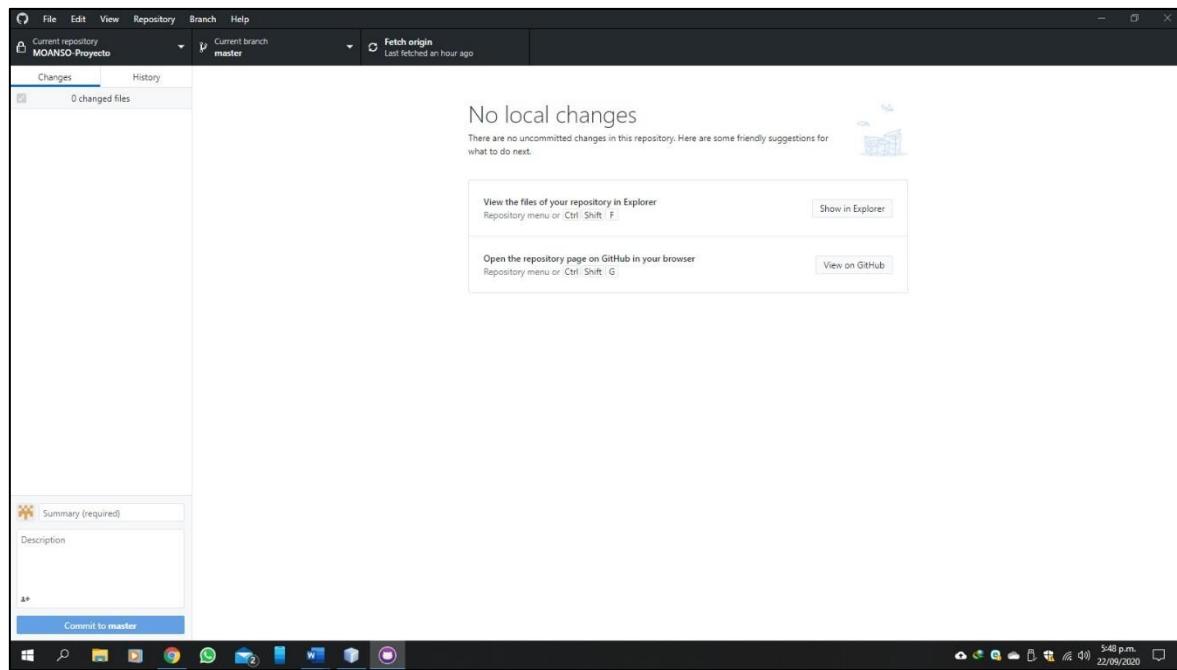


Ilustración 17: Requisitos no funcionales - Mauricio Villanueva - 3

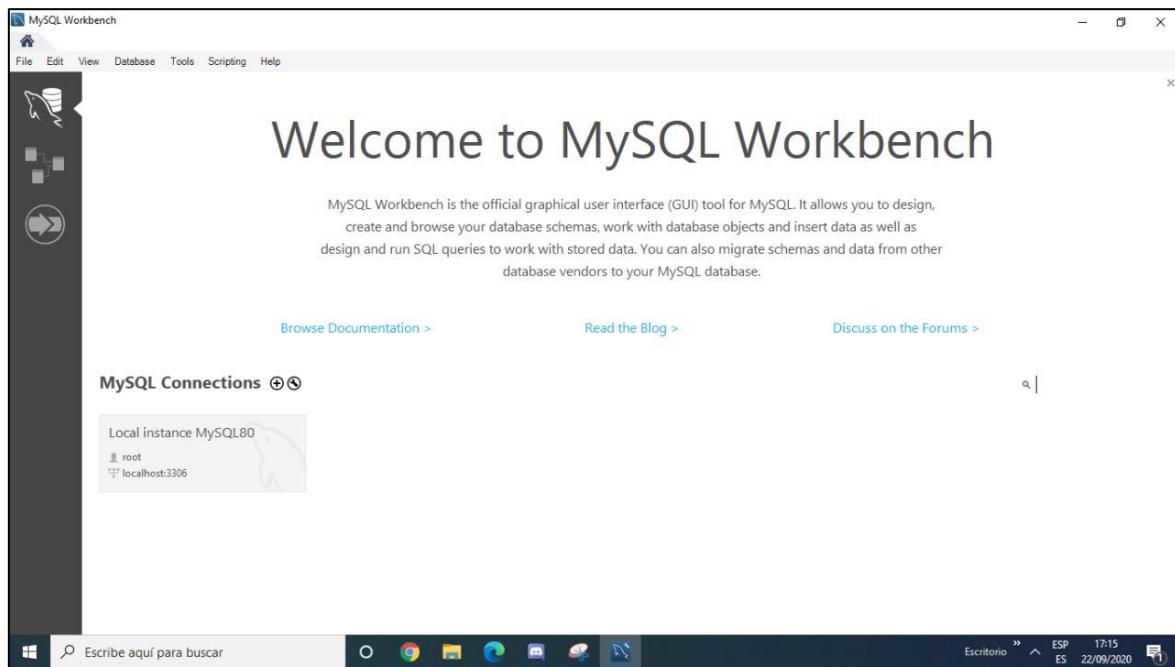


Ilustración 18: Requisitos no funcionales - Abihail Zapata - 1

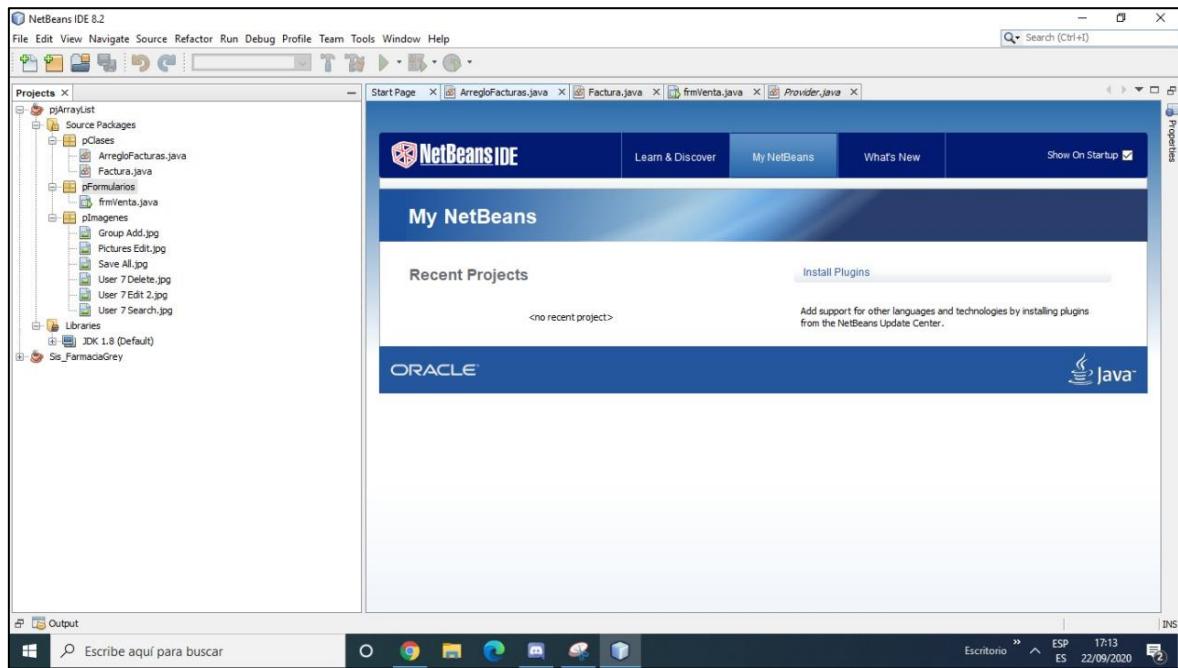


Ilustración 19: Requisitos no funcionales - Abihail Zapata - 2

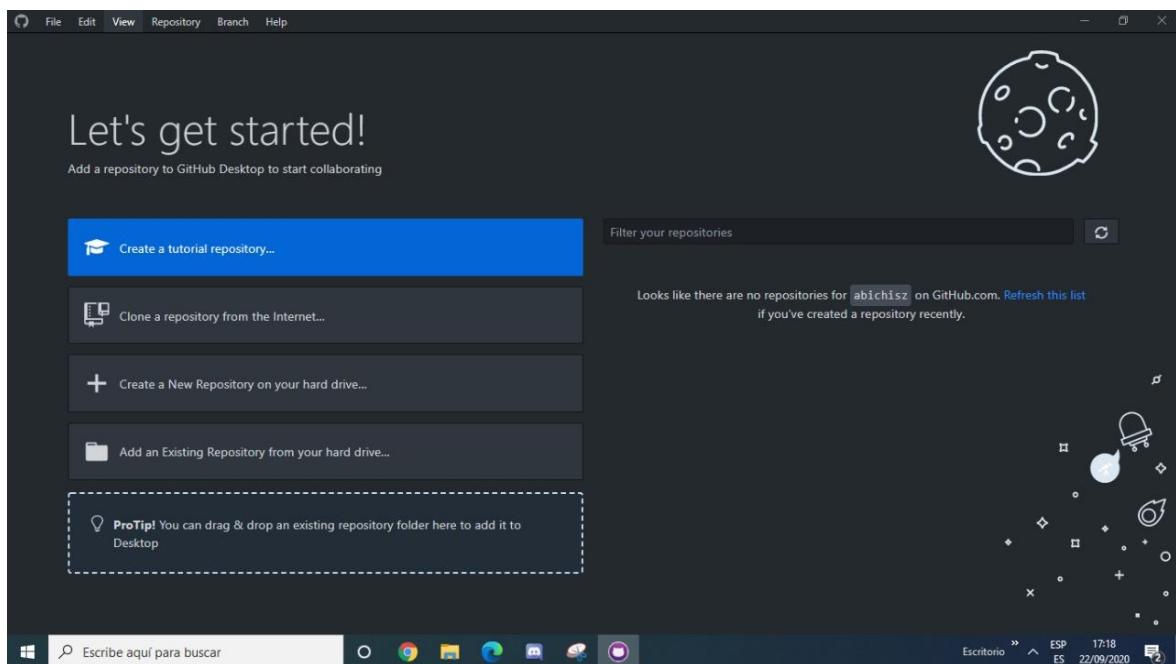


Ilustración 20: Requisitos no funcionales - Abihail Zapata - 3

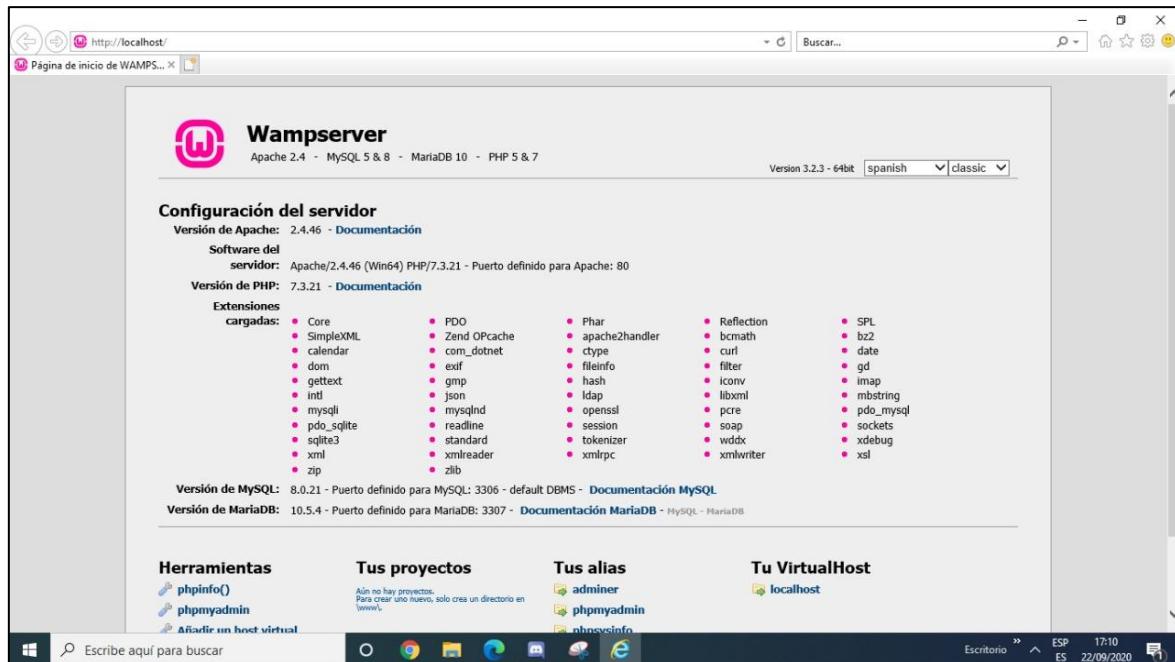


Ilustración 21: Requisitos no funcionales - Abihail Zapata - 4

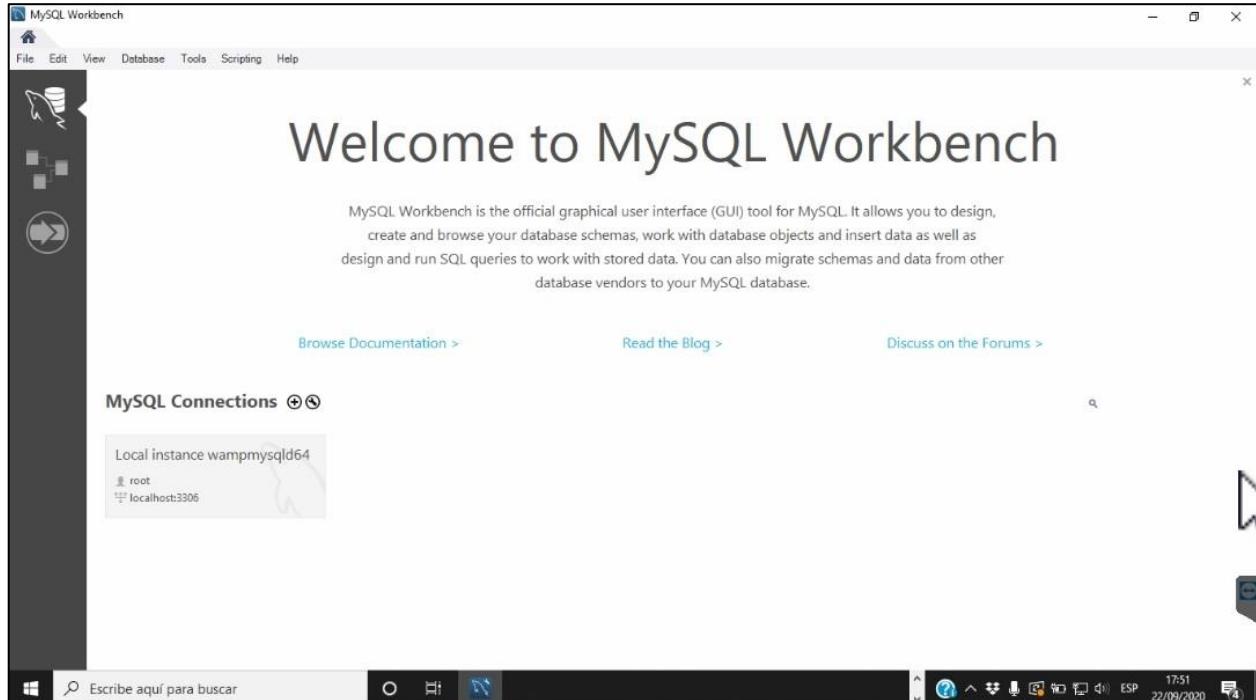


Ilustración 22: Requisitos no funcionales - Olenka Lazo - 1

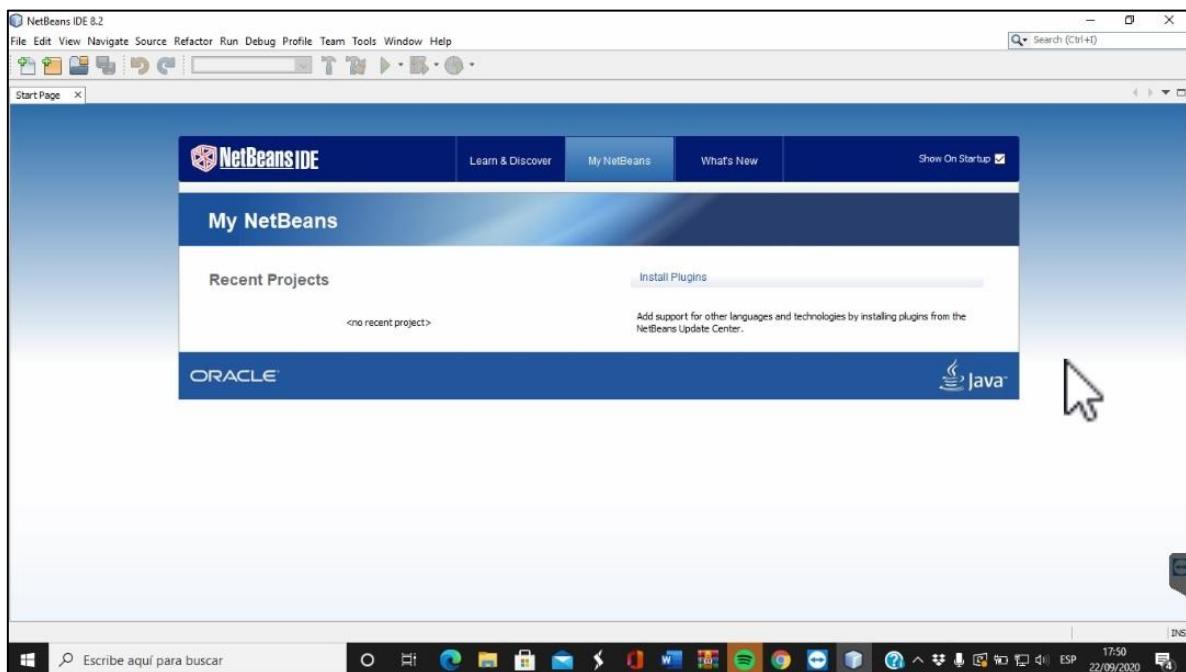


Ilustración 23: Requisitos no funcionales - Olenka Lazo - 2

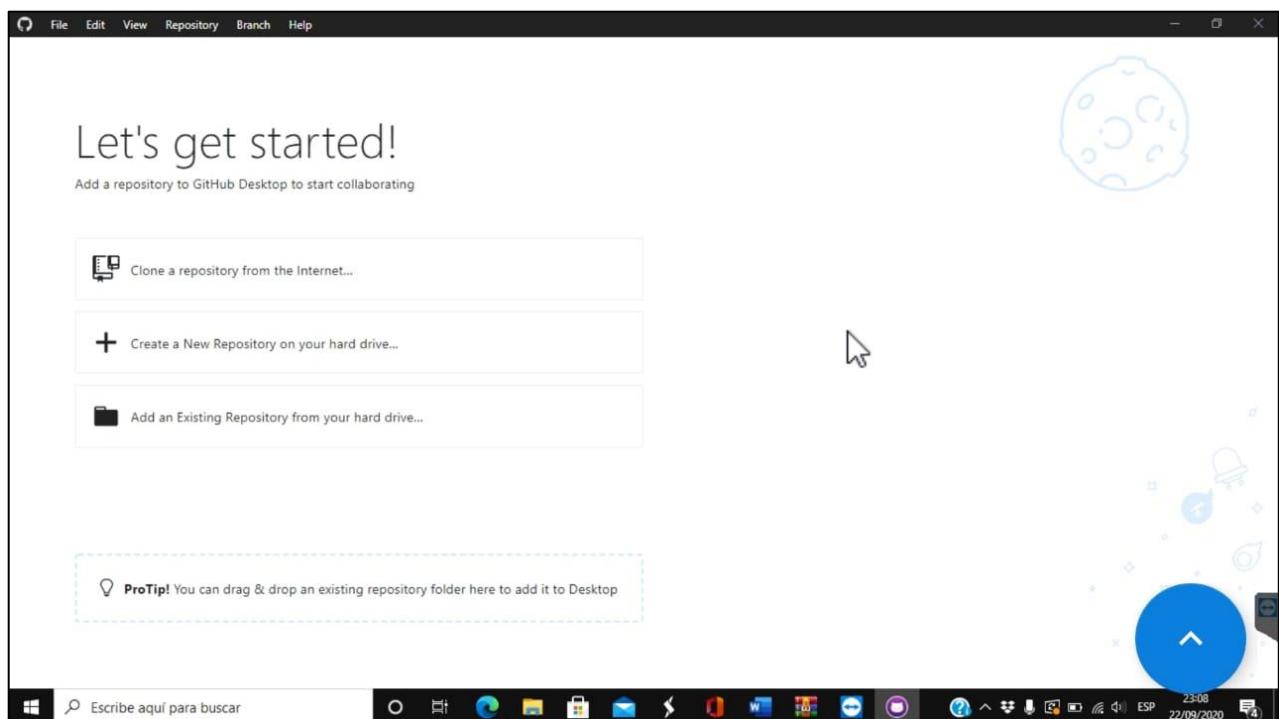


Ilustración 24: Requisitos no funcionales - Olenka Lazo - 3

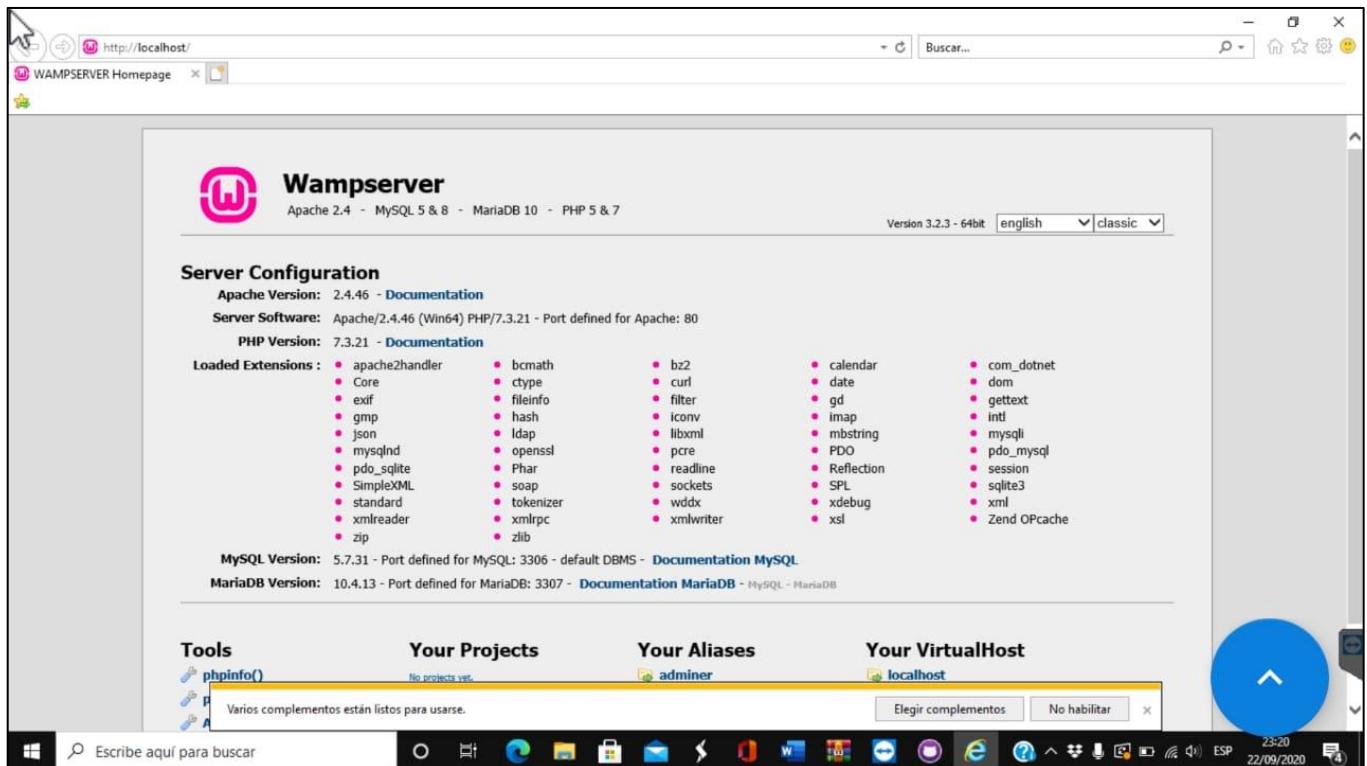


Ilustración 25: Requisitos no funcionales - Olenka Lazo - 4

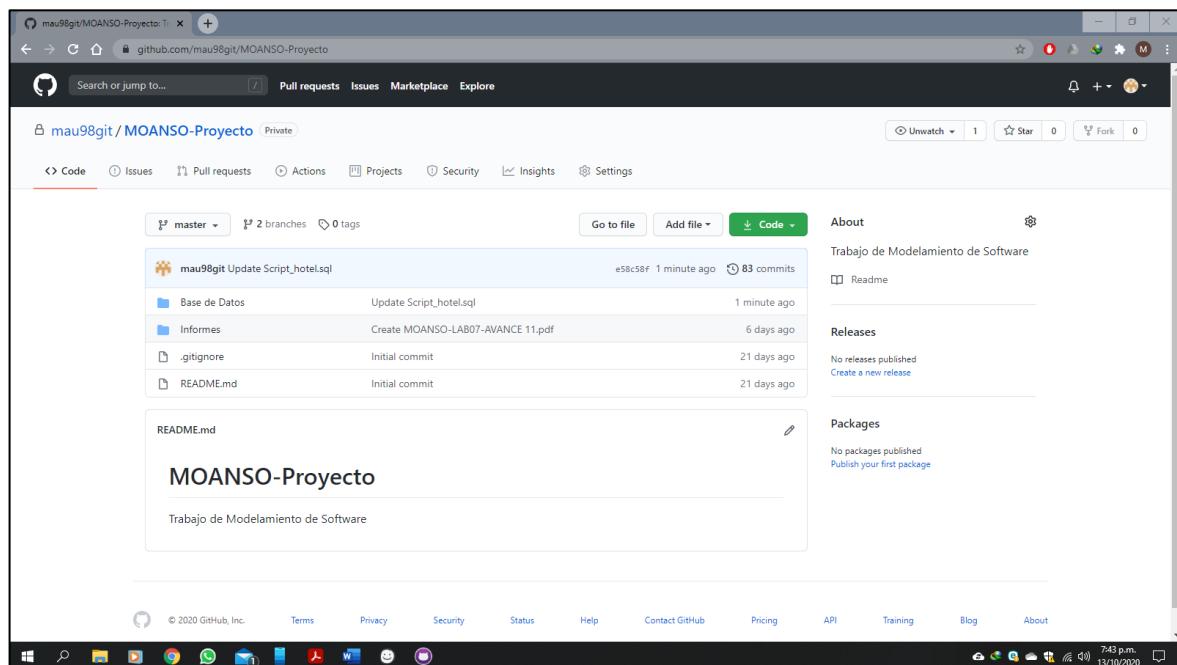


Ilustración 26: Repositorio GitHub

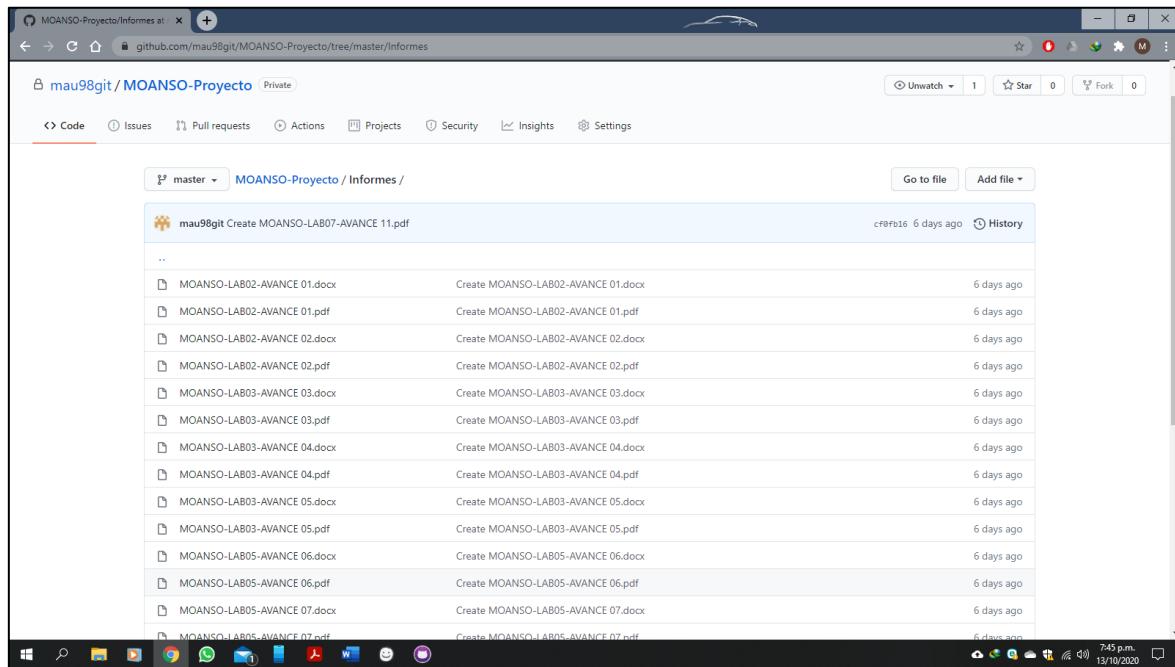


Ilustración 27: Repositorio GitHub - Informes - 1

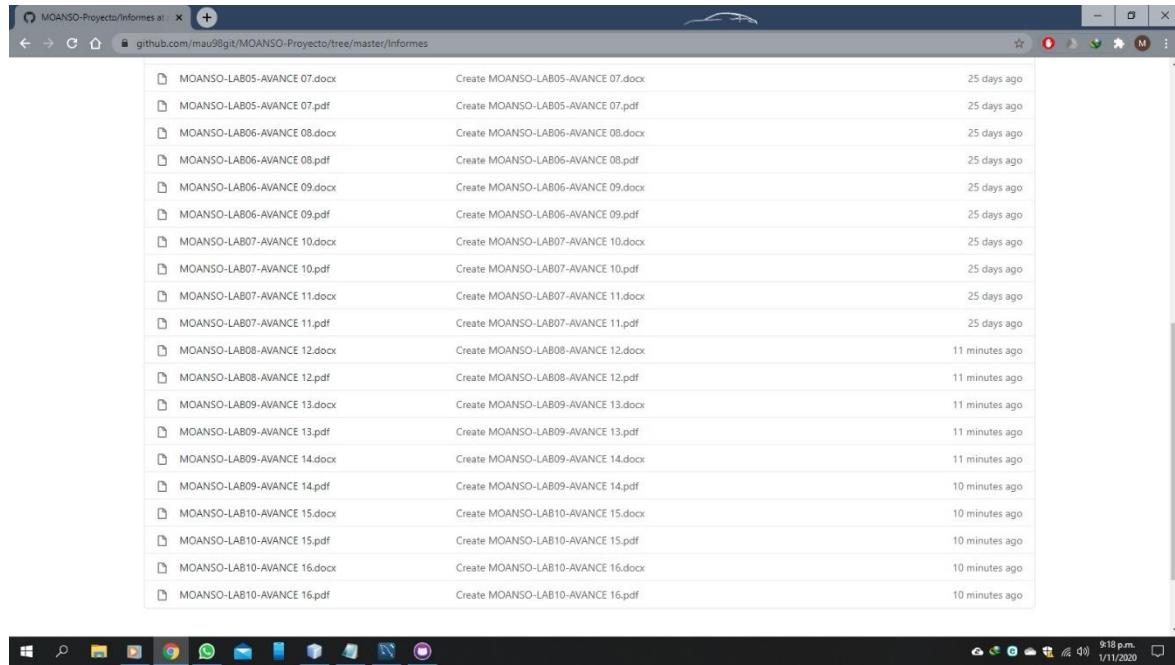


Ilustración 28: Repositorio GitHub - Informes - 2

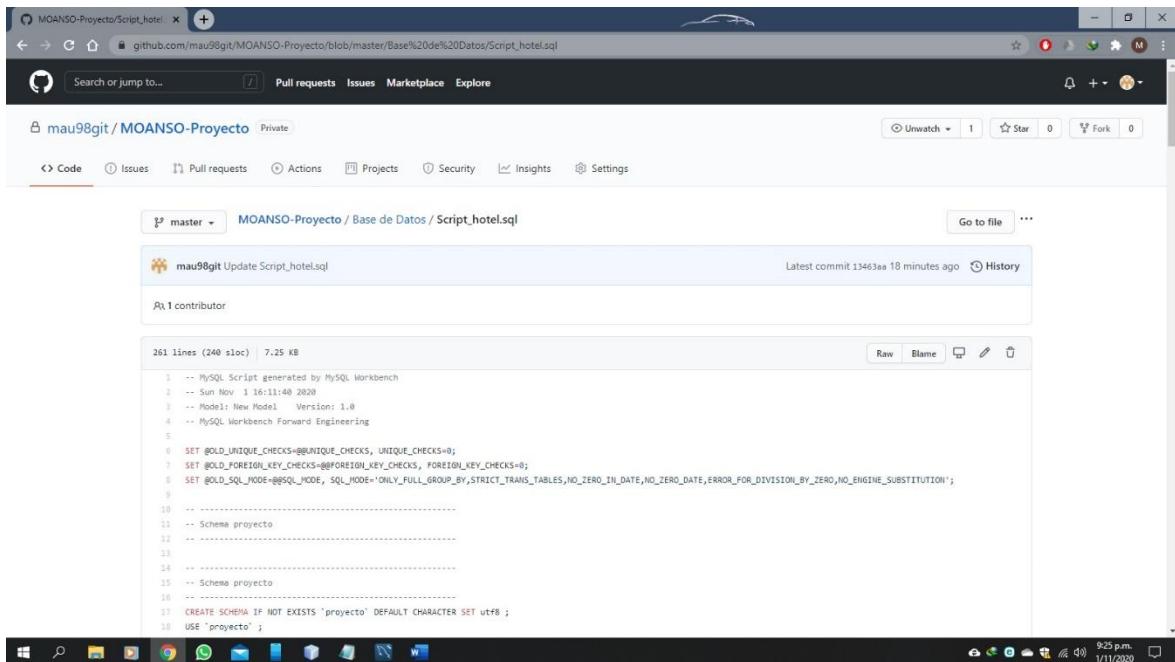


Ilustración 29: Repositorio GitHub - Base de Datos - 1

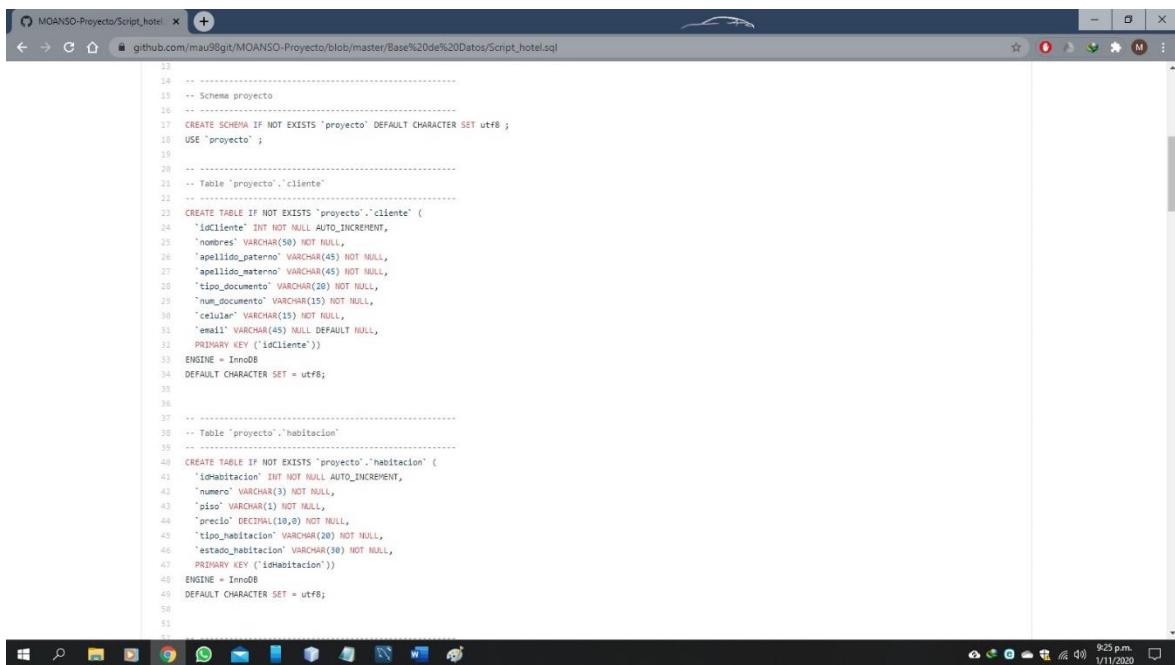
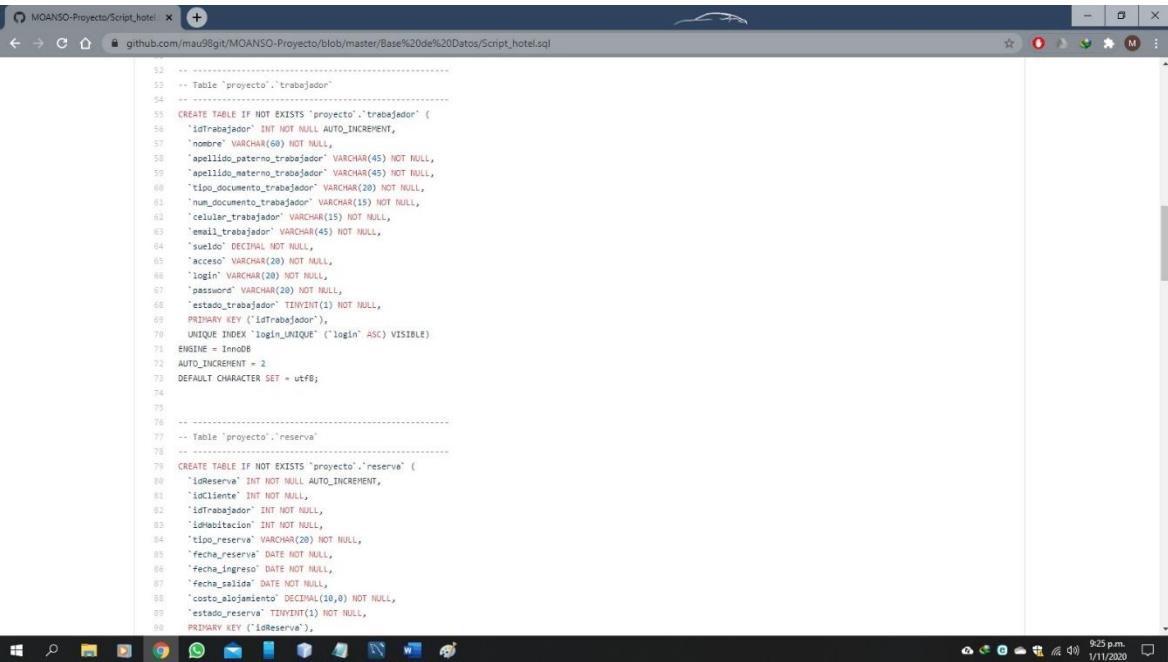


Ilustración 30: Repositorio GitHub - Base de Datos - 2



```

32 -- -----
33 -- Table 'proyecto'.'trabajador'
34 --
35 CREATE TABLE IF NOT EXISTS `proyecto`.'trabajador' (
36     `idtrabajador` INT NOT NULL AUTO_INCREMENT,
37     `nombre` VARCHAR(60) NOT NULL,
38     `apellido_paterno_trabajador` VARCHAR(45) NOT NULL,
39     `apellido_materno_trabajador` VARCHAR(45) NOT NULL,
40     `tipo_documento_trabajador` VARCHAR(20) NOT NULL,
41     `num_documento_trabajador` VARCHAR(15) NOT NULL,
42     `celular_trabajador` VARCHAR(15) NOT NULL,
43     `email_trabajador` VARCHAR(45) NOT NULL,
44     `sueldo` DECIMAL NOT NULL,
45     `acceso` VARCHAR(20) NOT NULL,
46     `login` VARCHAR(20) NOT NULL,
47     `password` VARCHAR(20) NOT NULL,
48     `estado_trabajador` TINYINT(1) NOT NULL,
49     PRIMARY KEY (`idtrabajador`),
50     UNIQUE INDEX `login_UNIQUE` (`login` ASC) VISIBLE)
51 ENGINE = InnoDB
52 AUTO_INCREMENT = 2
53 DEFAULT CHARACTER SET = utf8;
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77 -- Table 'proyecto'.'reserva'
78 --
79 CREATE TABLE IF NOT EXISTS `proyecto`.'reserva' (
80     `idReserva` INT NOT NULL AUTO_INCREMENT,
81     `idcliente` INT NOT NULL,
82     `idtrabajador` INT NOT NULL,
83     `idhabitacion` INT NOT NULL,
84     `tipo_reserva` VARCHAR(20) NOT NULL,
85     `fecha_reserva` DATE NOT NULL,
86     `fecha_ingreso` DATE NOT NULL,
87     `fecha_salida` DATE NOT NULL,
88     `costo_alquiler` DECIMAL(10,0) NOT NULL,
89     `estado_reserva` TINYINT(1) NOT NULL,
90     PRIMARY KEY (`idReserva`),
91 );

```

Ilustración 31: Repositorio GitHub - Base de Datos - 3

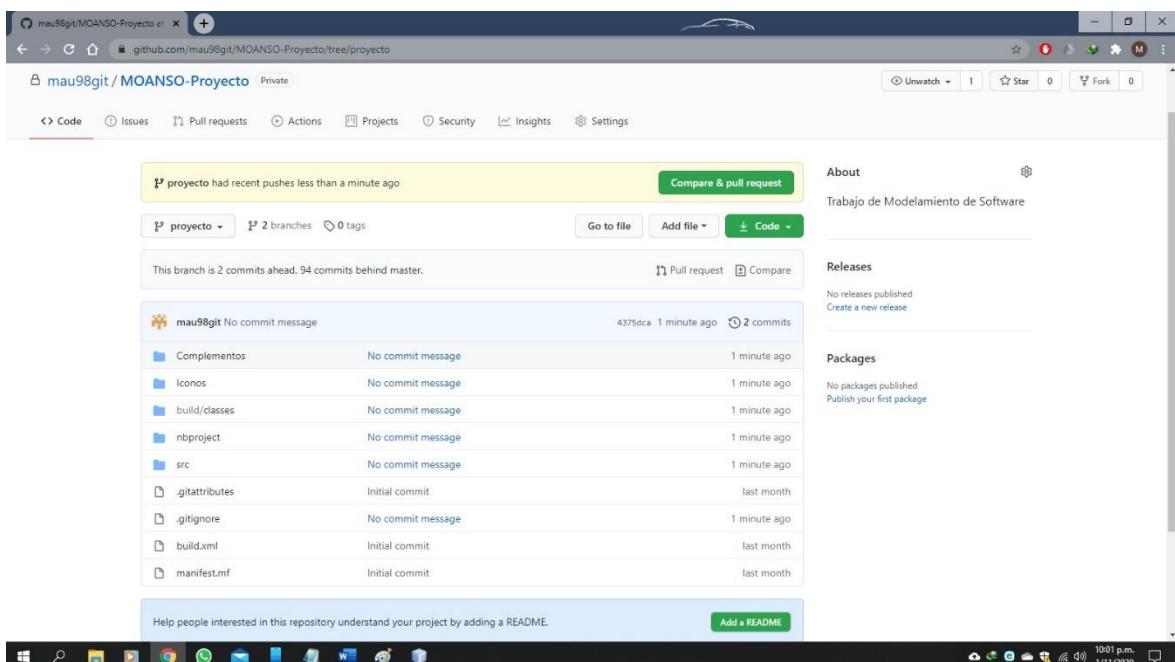


Ilustración 32: Repositorio GitHub - Software - 1

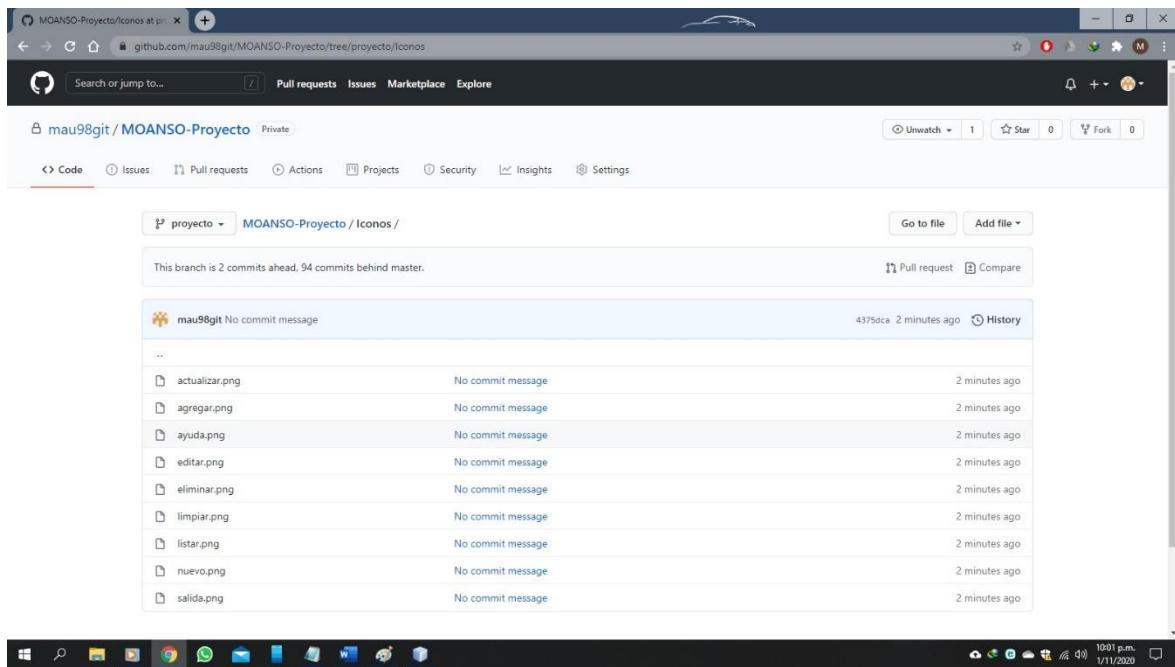


Ilustración 33: Repositorio GitHub - Software - 2

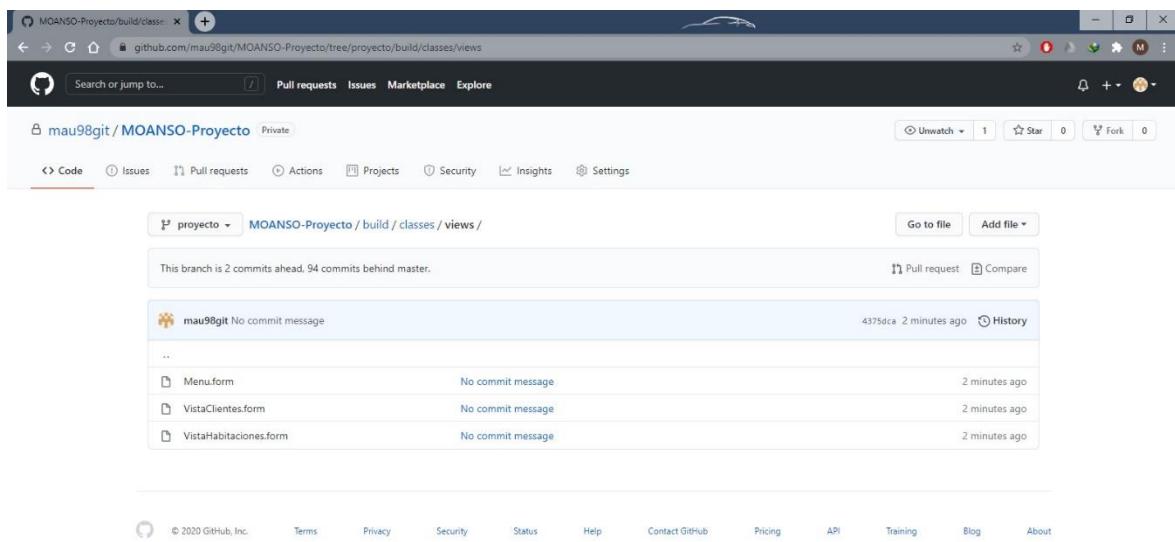


Ilustración 34: Repositorio GitHub - Software - 3

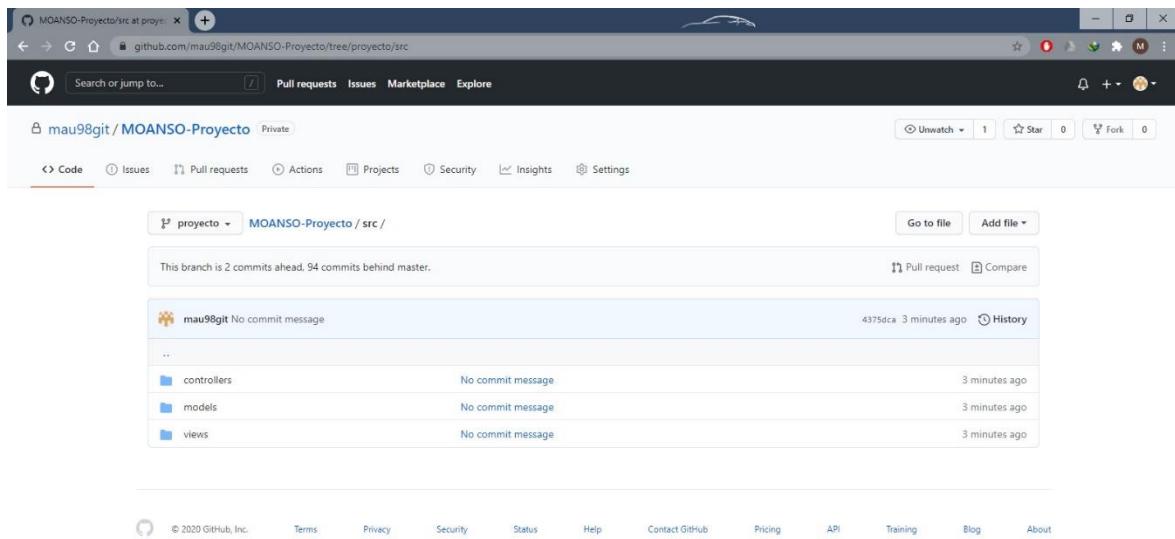


Ilustración 35: Repositorio GitHub - Software - 4

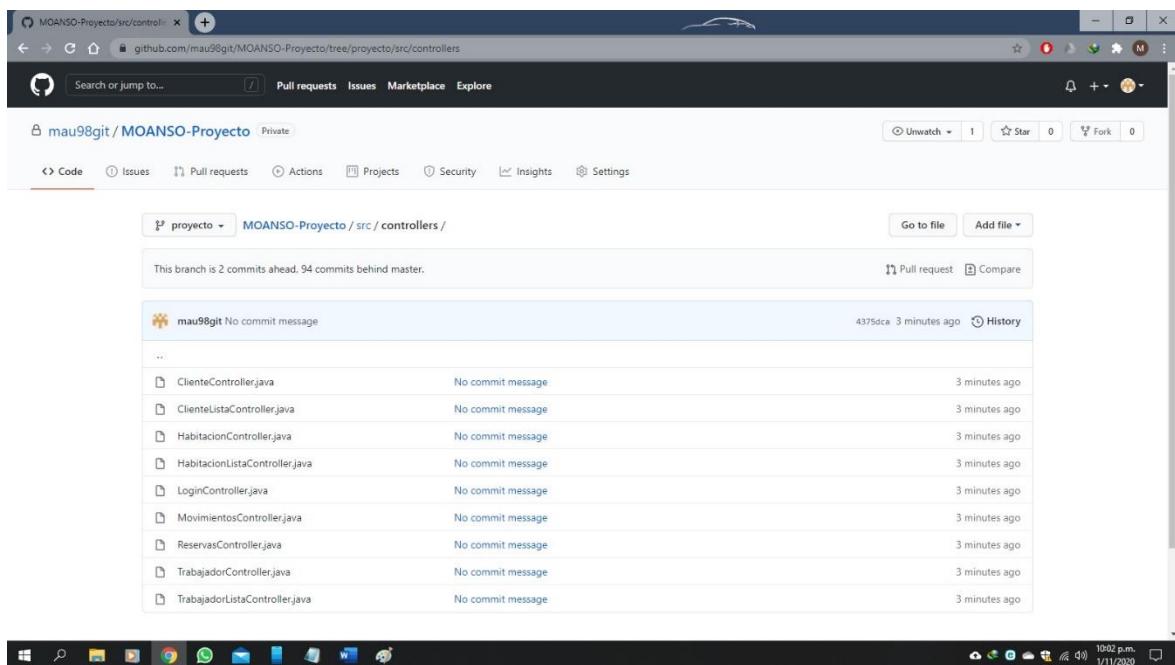


Ilustración 36: Repositorio GitHub - Software - 5

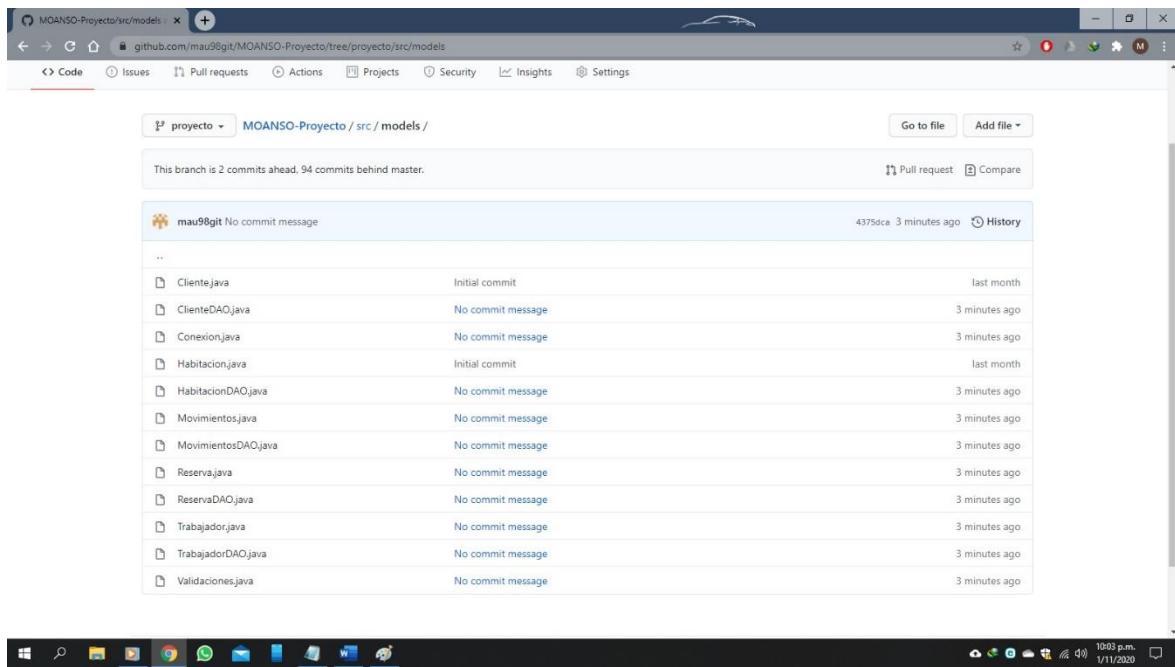


Ilustración 37: Repositorio GitHub - Software - 6

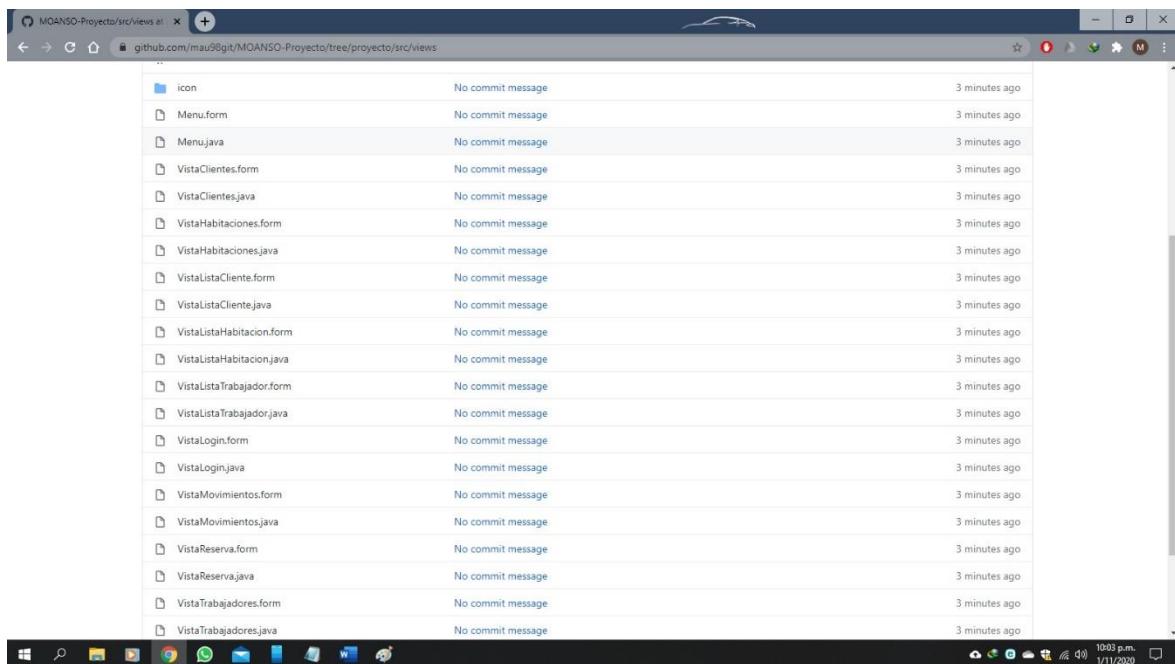


Ilustración 38: Repositorio GitHub - Software - 7

## Script de MySQL

```
-- MySQL Script generated by MySQL Workbench
-- Sun Nov 1 16:11:40 2020
-- Model: New Model Version: 1.0
-- MySQL Workbench Forward Engineering
```

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
-- Schema proyecto
```

```
-- Schema proyecto
```

```
CREATE SCHEMA IF NOT EXISTS `proyecto` DEFAULT CHARACTER SET utf8 ;
USE `proyecto` ;
```

```
-- Table `proyecto`.`cliente`
```

```
-----  
CREATE TABLE IF NOT EXISTS `proyecto`.`cliente` (  
    `idCliente` INT NOT NULL AUTO_INCREMENT,  
    `nombres` VARCHAR(50) NOT NULL,  
    `apellido_paterno` VARCHAR(45) NOT NULL,  
    `apellido_materno` VARCHAR(45) NOT NULL,  
    `tipo_documento` VARCHAR(20) NOT NULL,  
    `num_documento` VARCHAR(15) NOT NULL,  
    `celular` VARCHAR(15) NOT NULL,  
    `email` VARCHAR(45) NULL DEFAULT NULL,  
    PRIMARY KEY (`idCliente`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

```
-- Table `proyecto`.`habitacion`  
-----
```

```
CREATE TABLE IF NOT EXISTS `proyecto`.`habitacion` (  
    `idHabitacion` INT NOT NULL AUTO_INCREMENT,  
    `numero` VARCHAR(3) NOT NULL,  
    `piso` VARCHAR(1) NOT NULL,  
    `precio` DECIMAL(10,0) NOT NULL,  
    `tipo_habitacion` VARCHAR(20) NOT NULL,  
    `estado_habitacion` VARCHAR(30) NOT NULL,  
    PRIMARY KEY (`idHabitacion`))
```

ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;

-- -----

-- Table `proyecto`.`trabajador`

-- -----

```
CREATE TABLE IF NOT EXISTS `proyecto`.`trabajador` (
  `idTrabajador` INT NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(60) NOT NULL,
  `apellido_paterno_trabajador` VARCHAR(45) NOT NULL,
  `apellido_materno_trabajador` VARCHAR(45) NOT NULL,
  `tipo_documento_trabajador` VARCHAR(20) NOT NULL,
  `num_documento_trabajador` VARCHAR(15) NOT NULL,
  `celular_trabajador` VARCHAR(15) NOT NULL,
  `email_trabajador` VARCHAR(45) NOT NULL,
  `sueldo` DECIMAL NOT NULL,
  `acceso` VARCHAR(20) NOT NULL,
  `login` VARCHAR(20) NOT NULL,
  `password` VARCHAR(20) NOT NULL,
  `estado_trabajador` TINYINT(1) NOT NULL,
  PRIMARY KEY (`idTrabajador`),
  UNIQUE INDEX `login_UNIQUE` (`login` ASC) VISIBLE)
```

ENGINE = InnoDB  
AUTO\_INCREMENT = 2  
DEFAULT CHARACTER SET = utf8;

-- Table `proyecto`.`reserva`

CREATE TABLE IF NOT EXISTS `proyecto`.`reserva` (

  `idReserva` INT NOT NULL AUTO\_INCREMENT,

  `idCliente` INT NOT NULL,

  `idTrabajador` INT NOT NULL,

  `idHabitacion` INT NOT NULL,

  `tipo\_reserva` VARCHAR(20) NOT NULL,

  `fecha\_reserva` DATE NOT NULL,

  `fecha\_ingreso` DATE NOT NULL,

  `fecha\_salida` DATE NOT NULL,

  `costo\_alojamiento` DECIMAL(10,0) NOT NULL,

  `estado\_reserva` TINYINT(1) NOT NULL,

  PRIMARY KEY (`idReserva`),

  INDEX `fk\_reserva\_habitacion\_idx` (`idHabitacion` ASC) VISIBLE,

  INDEX `fk\_reserva\_cliente\_idx` (`idCliente` ASC) VISIBLE,

  INDEX `fk\_reserva\_trabajador\_idx` (`idTrabajador` ASC) VISIBLE,

  CONSTRAINT `fk\_reserva\_cliente`

    FOREIGN KEY (`idCliente`)

      REFERENCES `proyecto`.`cliente` (`idCliente`),

  CONSTRAINT `fk\_reserva\_habitacion`

    FOREIGN KEY (`idHabitacion`)

      REFERENCES `proyecto`.`habitacion` (`idHabitacion`),

```
CONSTRAINT `fk_reserva_trabajador`  
FOREIGN KEY (`idTrabajador`)  
REFERENCES `proyecto`.`trabajador` (`idTrabajador`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

```
-- -----  
-- Table `proyecto`.` pago`
```

```
-----  
CREATE TABLE IF NOT EXISTS `proyecto`.` pago` (  
`idPago` INT NOT NULL AUTO_INCREMENT,  
`idReserva` INT NOT NULL,  
`tipo_comprobante` VARCHAR(20) NOT NULL,  
`num_comprobante` VARCHAR(20) NOT NULL,  
`igv` DECIMAL(10,0) NOT NULL,  
`total_pago` DECIMAL(10,0) NOT NULL,  
`fecha_emision` DATE NOT NULL,  
`fecha_pago` DATE NOT NULL,  
PRIMARY KEY (`idPago`),  
INDEX `fk_pago_reserva_idx` (`idReserva` ASC) VISIBLE,  
CONSTRAINT `fk_pago_reserva`  
FOREIGN KEY (`idReserva`)  
REFERENCES `proyecto`.` reserva` (`idReserva`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

```
-- Table `proyecto`.`movimiento`
```

```
CREATE TABLE IF NOT EXISTS `proyecto`.`movimiento` (
```

```
 `idMovimiento` INT NOT NULL AUTO_INCREMENT,
```

```
 `nombreCliente` VARCHAR(45) NULL,
```

```
 `apellido_p_Cliente` VARCHAR(45) NULL,
```

```
 `apellido_m_Cliente` VARCHAR(45) NULL,
```

```
 `habitacionCliente` VARCHAR(45) NULL,
```

```
 `fecha_hora_ingreso` TIMESTAMP NULL,
```

```
 `fecha_hora_salida` TIMESTAMP NULL,
```

```
 `estado_movimiento` TINYINT(1) NULL,
```

```
 PRIMARY KEY (`idMovimiento`))
```

```
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;
```

```
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
```

```
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

```
CREATE TRIGGER mov_trig AFTER INSERT ON cliente FOR EACH ROW
INSERT INTO movimiento (idMovimiento, nombreCliente, apellido_p_Cliente,
apellido_m_Cliente, habitacionCliente, fecha_hora_ingreso, fecha_hora_salida,
estado_movimiento)
```

```
VALUES (null, NEW.nombres, NEW.apellido_paterno, NEW.apellido_materno,  
null, null,null,1);
```

```
INSERT INTO `proyecto`.`trabajador`(`idTrabajador`,  
`nombre`,  
`apellido_paterno_trabajador`,  
`apellido_materno_trabajador`,  
`tipo_documento_trabajador`,  
`num_documento_trabajador`,  
`celular_trabajador`,  
`email_trabajador`,  
`sueldo`,  
`acceso`,  
`login`,  
`password`,  
`estado_trabajador`)
```

VALUES

```
(null,  
"Diego",  
"Tasaico",  
"Freundt",  
"DNI",  
"73194743",  
"966107790",  
"tasaicod4@gmail.com",  
1500,
```

"Administrador",

"userdiego",

"1234",

1);

INSERT INTO `proyecto`.`trabajador`(`idTrabajador`,

`nombre`,

`apellido\_paterno\_trabajador`,

`apellido\_materno\_trabajador`,

`tipo\_documento\_trabajador`,

`num\_documento\_trabajador`,

`celular\_trabajador`,

`email\_trabajador`,

`sueldo`,

`acceso`,

`login`,

`password`,

`estado\_trabajador`)

VALUES

(null,

"Mauricio",

"Villanueva",

"Yaranga",

"DNI",

"73191743",

"966106790",

"mauvill@gmail.com",

1500,

"Administrador",

"usermau",

"1234",

1);

INSERT INTO `proyecto`.`trabajador`(`idTrabajador`,

`nombre`,

`apellido\_paterno\_trabajador`,

`apellido\_materno\_trabajador`,

`tipo\_documento\_trabajador`,

`num\_documento\_trabajador`,

`celular\_trabajador`,

`email\_trabajador`,

`sueldo`,

`acceso`,

`login`,

`password`,

`estado\_trabajador`)

VALUES

(null,

"Olenka",

"Lazo",

"Pinchi",

"DNI",

"73194793",  
"966109790",  
"ole4@gmail.com",  
1500,  
"Administrador",  
"userole",  
"1234",  
1);

```
INSERT INTO `proyecto`.`trabajador`(`idTrabajador`,  
`nombre`,  
`apellido_paterno_trabajador`,  
`apellido_materno_trabajador`,  
`tipo_documento_trabajador`,  
`num_documento_trabajador`,  
`celular_trabajador`,  
`email_trabajador`,  
`sueldo`,  
`acceso`,  
`login`,  
`password`,  
`estado_trabajador`)  
VALUES  
(null,  
"Abihail",  
"Zapata",
```

"Zamora",  
"DNI",  
"73194143",  
"916107790",  
"abih@gmail.com",  
1500,  
"Administrador",  
"userabi",  
"1234",  
1);

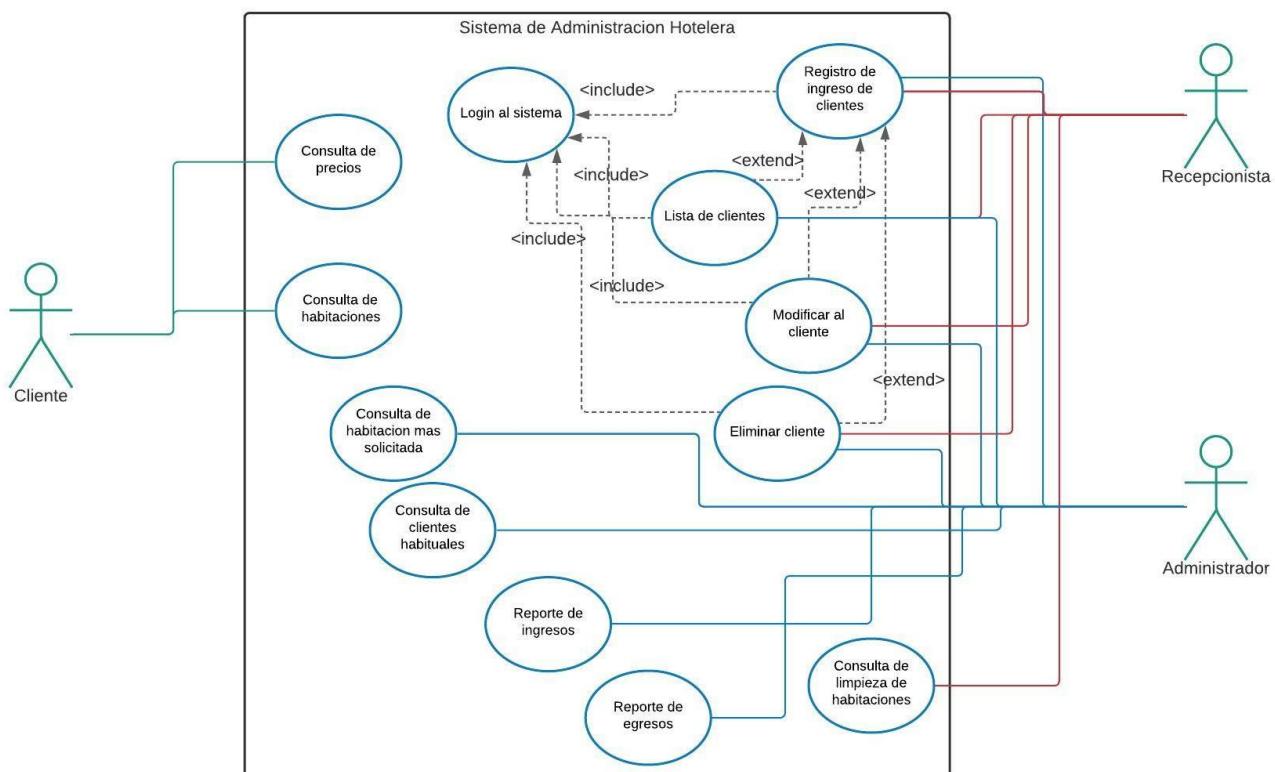


Ilustración 39: Diagrama de Casos de uso Relacionado - General

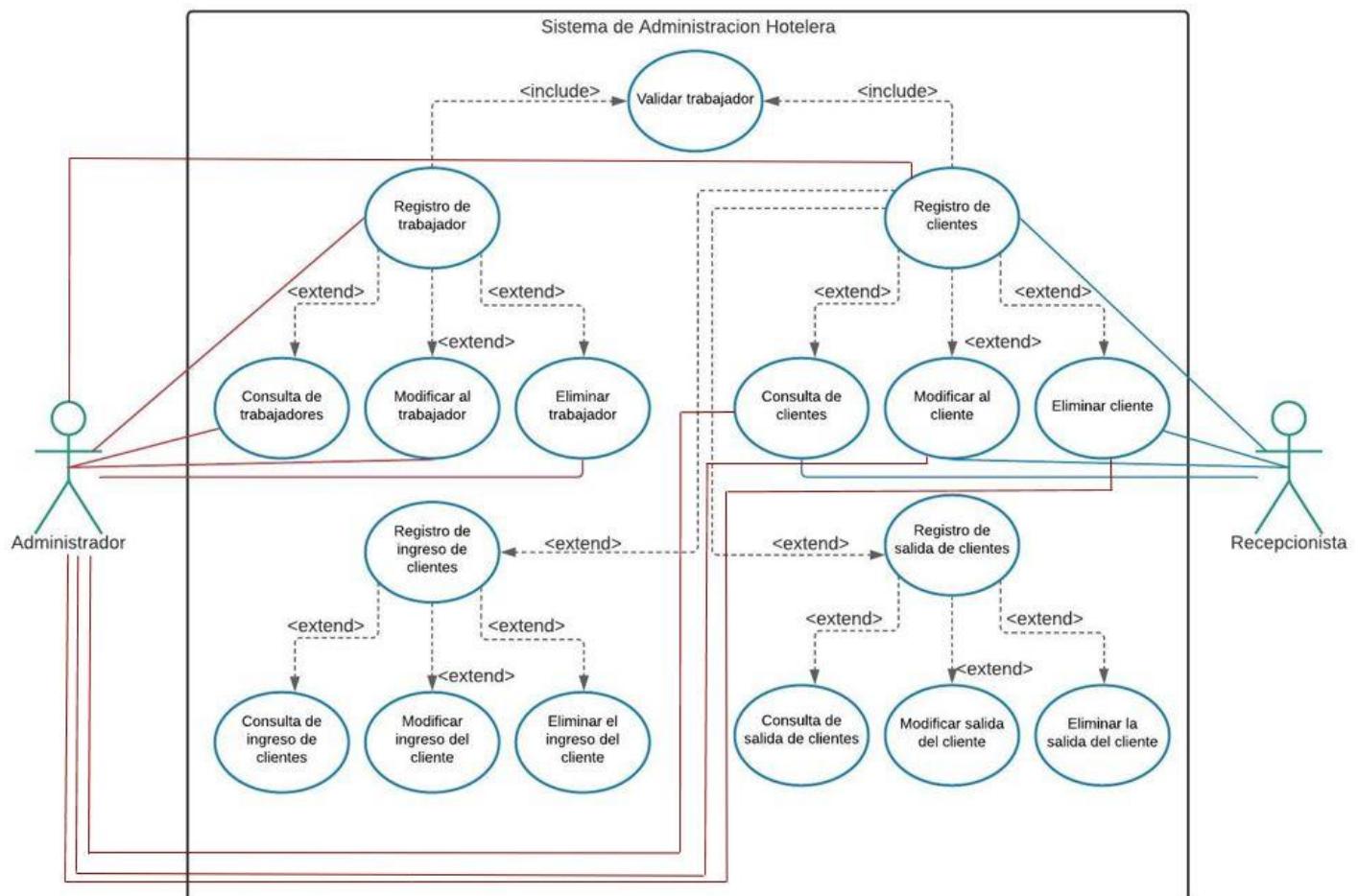


Ilustración 40: Diagrama de Casos de uso Relacionado - Versión 1

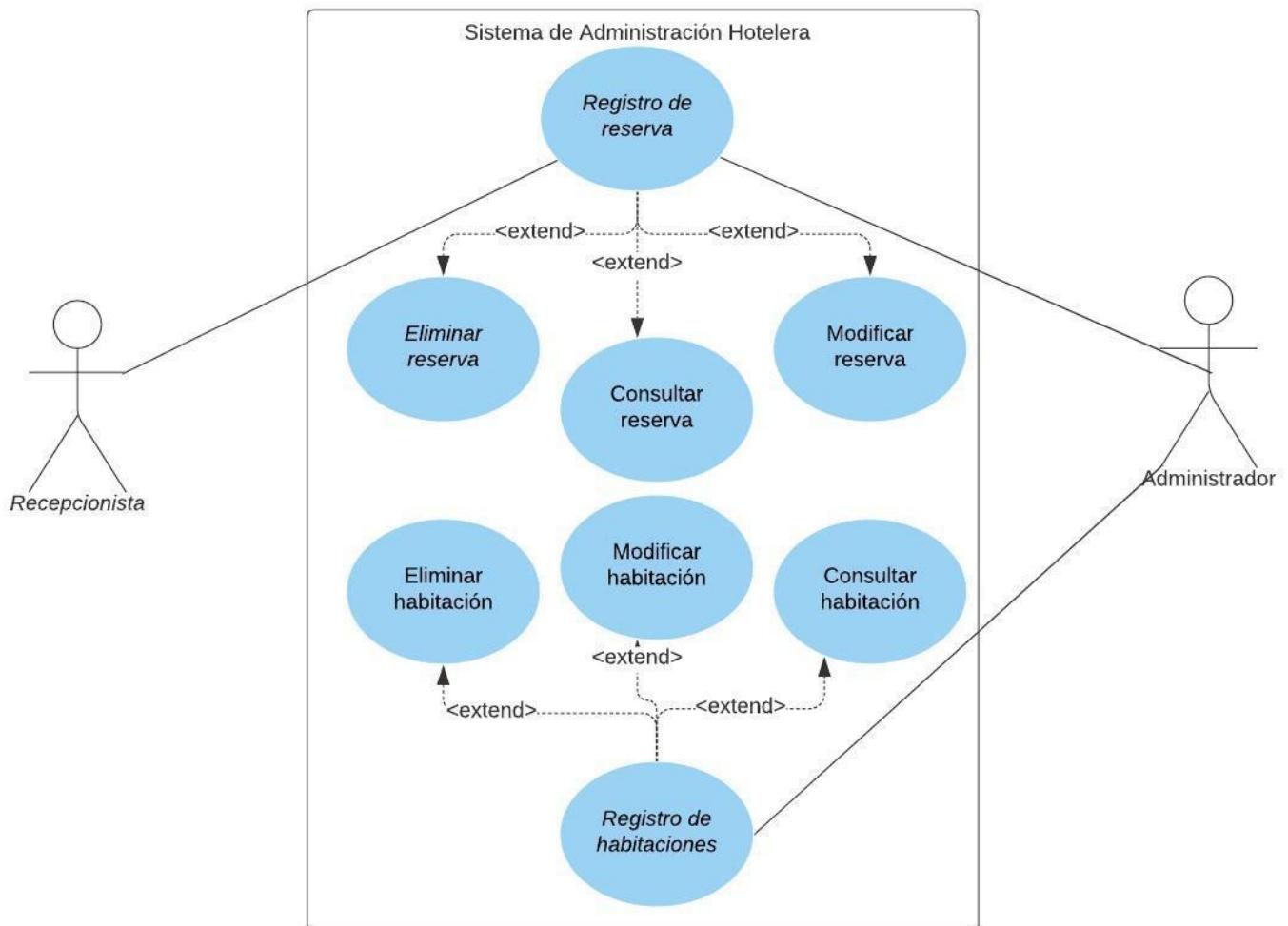
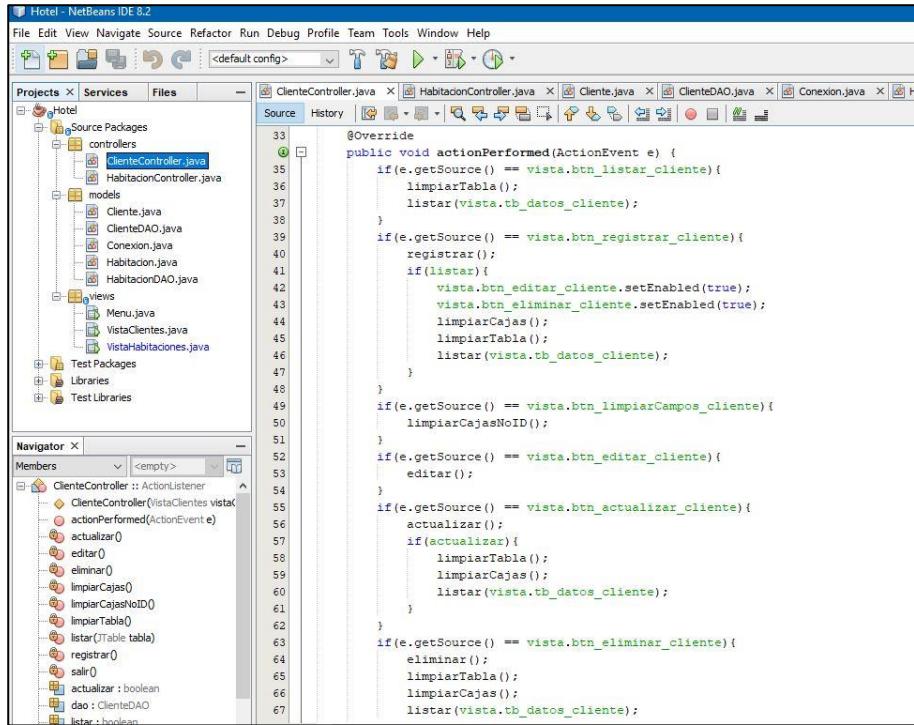


Ilustración 41: Diagrama de Casos de uso Relacionado - Versión 2

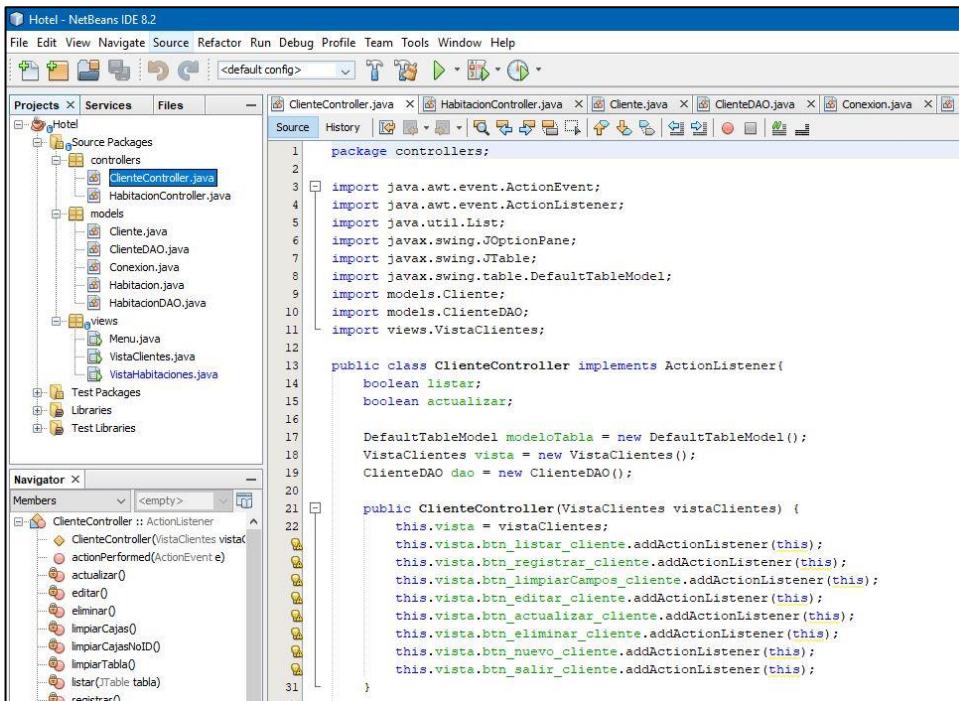


```

@Override
public void actionPerformed(ActionEvent e) {
    if(e.getSource() == vista.btn_listar_cliente){
        limpiarTabla();
        listar(vista.tb_datos_cliente);
    }
    if(e.getSource() == vista.btn_registrar_cliente){
        registrar();
        if(listar){
            vista.btn_editar_cliente.setEnabled(true);
            vista.btn_eliminar_cliente.setEnabled(true);
            limpiarCajas();
            limpiarTabla();
            listar(vista.tb_datos_cliente);
        }
    }
    if(e.getSource() == vista.btn_limpiarCampos_cliente){
        limpiarCajasNoID();
    }
    if(e.getSource() == vista.btn_editar_cliente){
        editar();
    }
    if(e.getSource() == vista.btn_actualizar_cliente){
        actualizar();
        if(actualizar){
            limpiarTabla();
            limpiarCajas();
            listar(vista.tb_datos_cliente);
        }
    }
    if(e.getSource() == vista.btn_eliminar_cliente){
        eliminar();
        limpiarTabla();
        limpiarCajas();
        listar(vista.tb_datos_cliente);
    }
}

```

Ilustración 42: Código de programación del software - 1



```

package controllers;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.List;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import models.Cliente;
import models.ClienteDAO;
import views.VistaClientes;

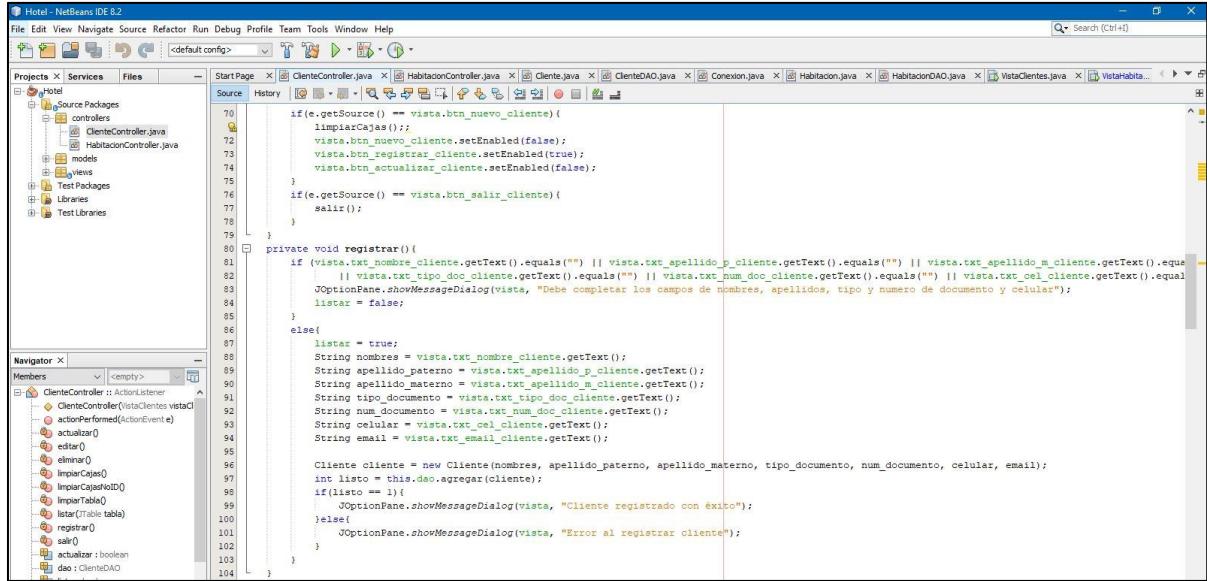
public class ClienteController implements ActionListener{
    boolean listar;
    boolean actualizar;

    DefaultTableModel modeloTabla = new DefaultTableModel();
    VistaClientes vista = new VistaClientes();
    ClienteDAO dao = new ClienteDAO();

    public ClienteController(VistaClientes vistaClientes) {
        this.vista = vistaClientes;
        this.vista.btn_listar_cliente.addActionListener(this);
        this.vista.btn_registrar_cliente.addActionListener(this);
        this.vista.btn_limpiarCampos_cliente.addActionListener(this);
        this.vista.btn_editar_cliente.addActionListener(this);
        this.vista.btn_actualizar_cliente.addActionListener(this);
        this.vista.btn_eliminar_cliente.addActionListener(this);
        this.vista.btn_nuevo_cliente.addActionListener(this);
        this.vista.btn_salir_cliente.addActionListener(this);
    }
}

```

Ilustración 43: Código de programación del software - 2



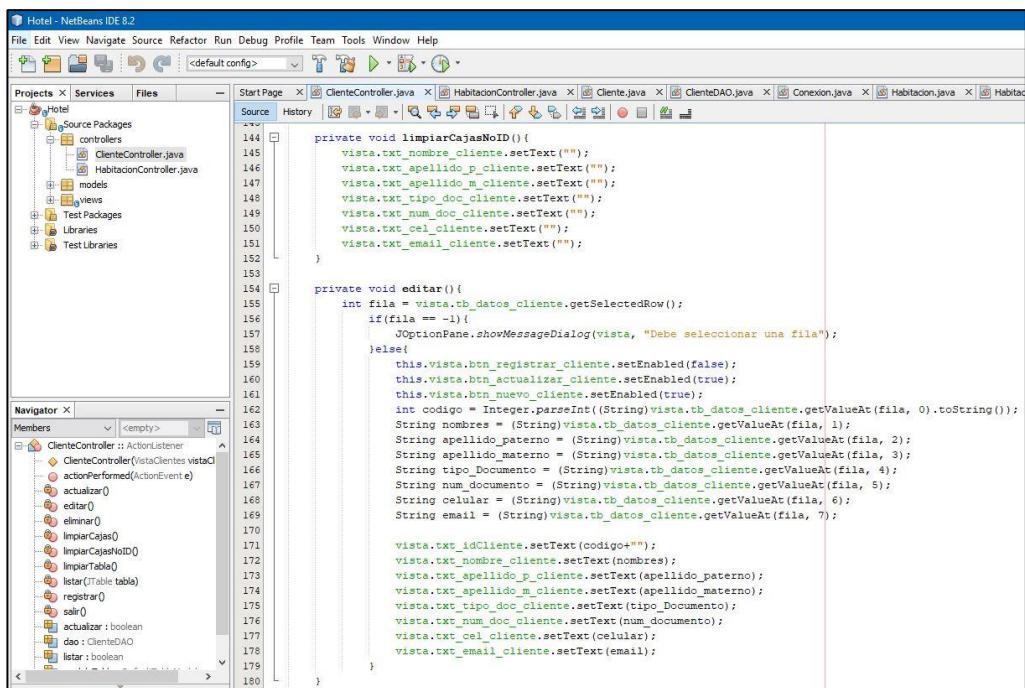
```

private void registrar() {
    if (vista.txt_nombre_cliente.getText().equals("") || vista.txt_apellido_p_cliente.getText().equals("") || vista.txt_apellido_m_cliente.getText().equals("") || vista.txt_tipo_doc_cliente.getText().equals("") || vista.txt_num_doc_cliente.getText().equals("") || vista.txt_cel_cliente.getText().equals("") || vista.txt_email_cliente.getText().equals("")) {
        JOptionPane.showMessageDialog(vista, "Debe completar los campos de nombres, apellidos, tipo y numero de documento y celular");
        lista = false;
    } else {
        lista = true;
        String nombres = vista.txt_nombre_cliente.getText();
        String apellido_paterno = vista.txt_apellido_p_cliente.getText();
        String apellido_materno = vista.txt_apellido_m_cliente.getText();
        String tipo_documento = vista.txt_tipo_doc_cliente.getText();
        String num_documento = vista.txt_num_doc_cliente.getText();
        String celular = vista.txt_cel_cliente.getText();
        String email = vista.txt_email_cliente.getText();

        Cliente cliente = new Cliente(nombres, apellido_paterno, apellido_materno, tipo_documento, num_documento, celular, email);
        int listo = this.dao.agregar(cliente);
        if(listo == 1){
            JOptionPane.showMessageDialog(vista, "Cliente registrado con éxito");
        } else{
            JOptionPane.showMessageDialog(vista, "Error al registrar cliente");
        }
    }
}

```

Ilustración 44: Código de programación del software - 3



```

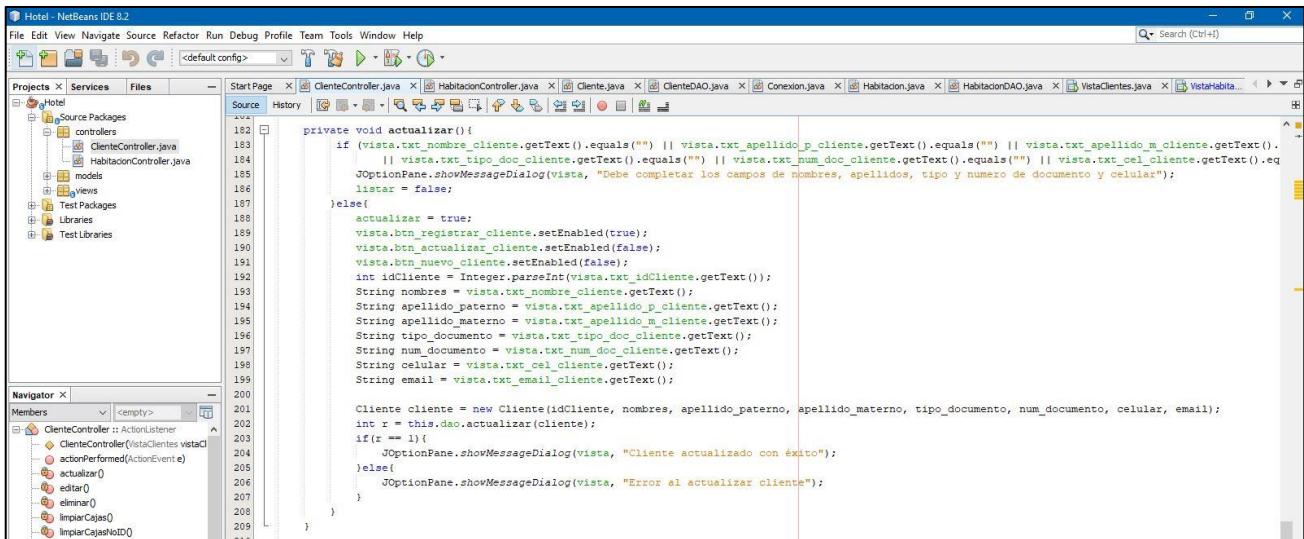
private void limpiarCajasNoID() {
    vista.txt_nombre_cliente.setText("");
    vista.txt_apellido_p_cliente.setText("");
    vista.txt_apellido_m_cliente.setText("");
    vista.txt_tipo_doc_cliente.setText("");
    vista.txt_num_doc_cliente.setText("");
    vista.txt_cel_cliente.setText("");
    vista.txt_email_cliente.setText("");
}

private void editar() {
    int fila = vista.tb_datos_cliente.getSelectedRow();
    if(fila == -1){
        JOptionPane.showMessageDialog(vista, "Debe seleccionar una fila");
    } else{
        this.vista.btn_registrar_cliente.setEnabled(false);
        this.vista.btn_actualizar_cliente.setEnabled(true);
        this.vista.btn_nuevo_cliente.setEnabled(true);
        int codigo = Integer.parseInt((String)vista.tb_datos_cliente.getValueAt(fila, 0).toString());
        String nombres = (String)vista.tb_datos_cliente.getValueAt(fila, 1);
        String apellido_paterno = (String)vista.tb_datos_cliente.getValueAt(fila, 2);
        String apellido_materno = (String)vista.tb_datos_cliente.getValueAt(fila, 3);
        String tipo_Documento = (String)vista.tb_datos_cliente.getValueAt(fila, 4);
        String num_documentos = (String)vista.tb_datos_cliente.getValueAt(fila, 5);
        String celular = (String)vista.tb_datos_cliente.getValueAt(fila, 6);
        String email = (String)vista.tb_datos_cliente.getValueAt(fila, 7);

        vista.txt_idCliente.setText(codigo+"");
        vista.txt_nombre_cliente.setText(nombres);
        vista.txt_apellido_p_cliente.setText(apellido_paterno);
        vista.txt_apellido_m_cliente.setText(apellido_materno);
        vista.txt_tipo_doc_cliente.setText(tipo_Documento);
        vista.txt_num_doc_cliente.setText(num_documentos);
        vista.txt_cel_cliente.setText(celular);
        vista.txt_email_cliente.setText(email);
    }
}

```

Ilustración 45: Código de programación del software - 4



```

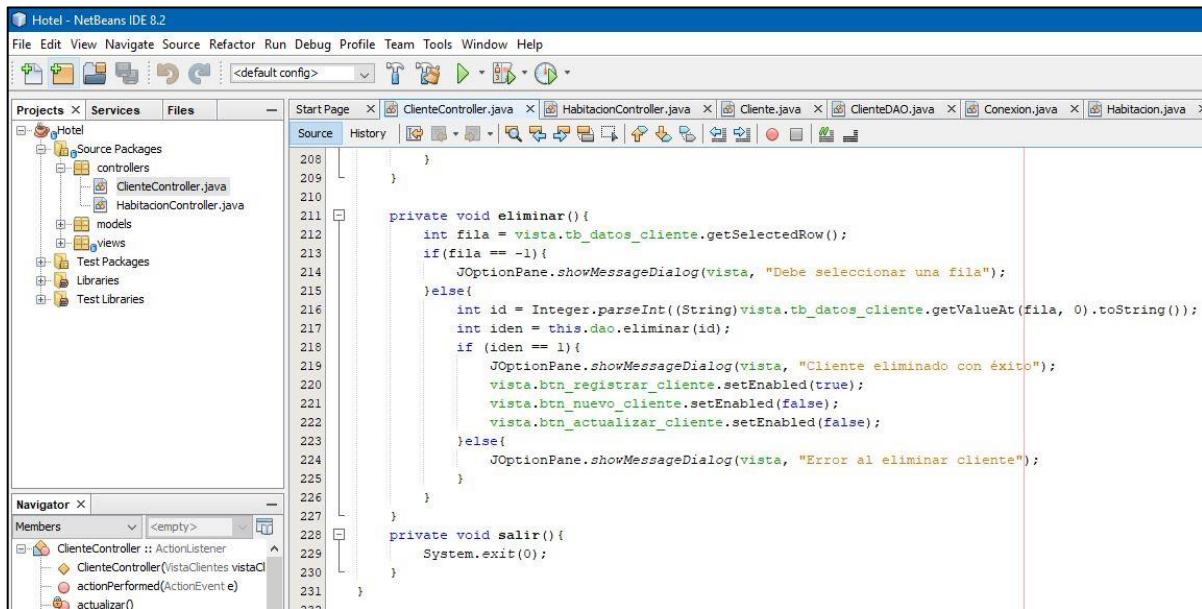
private void actualizar() {
    if (vista.txt_nombre_cliente.getText().equals("") || vista.txt_apellido_p_cliente.getText().equals("") || vista.txt_apellido_m_cliente.getText().equals("") || vista.txt_num_doc_cliente.getText().equals("") || vista.txt_cel_cliente.getText().equals("")){
        JOptionPane.showMessageDialog(vista, "Debe completar los campos de nombres, apellidos, tipo y numero de documento y celular");
        listar = false;
    }else{
        actualizar = true;
        vista.btn_registrar_cliente.setEnabled(true);
        vista.btn_actualizar_cliente.setEnabled(false);
        vista.btn_nuevo_cliente.setEnabled(false);
        int idCliente = Integer.parseInt(vista.txt_idCliente.getText());
        String nombres = vista.txt_nombre_cliente.getText();
        String apellido_paterno = vista.txt_apellido_p_cliente.getText();
        String apellido_materno = vista.txt_apellido_m_cliente.getText();
        String tipo_documento = vista.txt_tipo_doc_cliente.getText();
        String num_documento = vista.txt_num_doc_cliente.getText();
        String celular = vista.txt_cel_cliente.getText();
        String email = vista.txt_email_cliente.getText();

        Cliente cliente = new Cliente(idCliente, nombres, apellido_paterno, apellido_materno, tipo_documento, num_documento, celular, email);

        int r = this.dao.actualizar(cliente);
        if(r == 1){
            JOptionPane.showMessageDialog(vista, "Cliente actualizado con éxito");
        }else{
            JOptionPane.showMessageDialog(vista, "Error al actualizar cliente");
        }
    }
}

```

Ilustración 46: Código de programación del software - 5



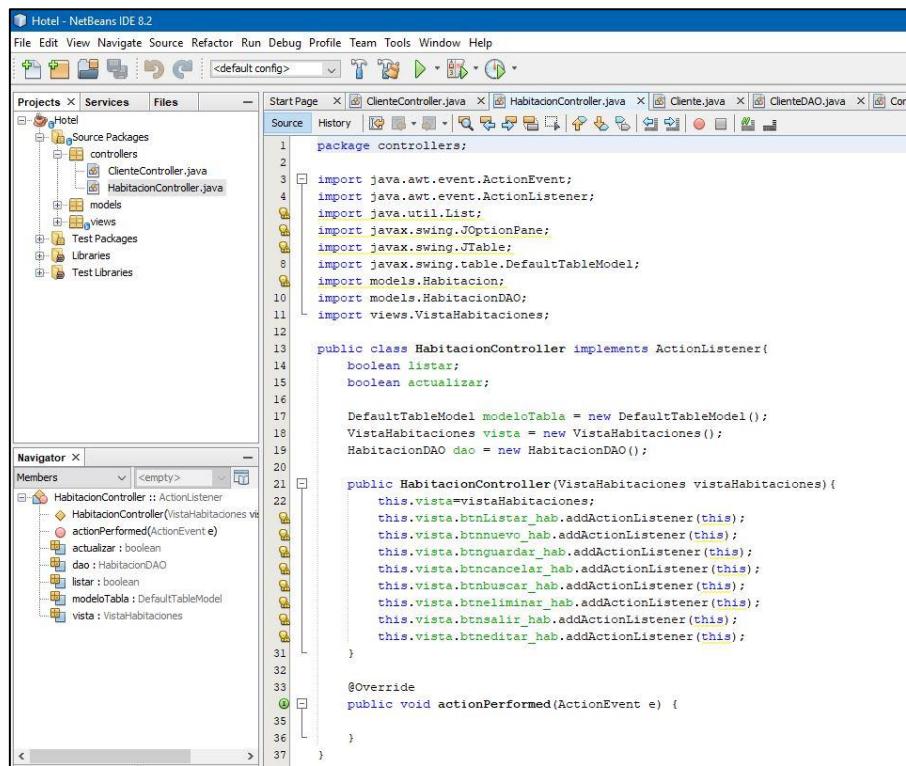
```

private void eliminar(){
    int fila = vista.tb_datos_cliente.getSelectedRow();
    if(fila == -1){
        JOptionPane.showMessageDialog(vista, "Debe seleccionar una fila");
    }else{
        int id = Integer.parseInt((String)vista.tb_datos_cliente.getValueAt(fila, 0).toString());
        int iden = this.dao.eliminar(id);
        if (iden == 1){
            JOptionPane.showMessageDialog(vista, "Cliente eliminado con éxito");
            vista.btn_registrar_cliente.setEnabled(true);
            vista.btn_nuevo_cliente.setEnabled(false);
            vista.btn_actualizar_cliente.setEnabled(false);
        }else{
            JOptionPane.showMessageDialog(vista, "Error al eliminar cliente");
        }
    }
}

private void salir(){
    System.exit(0);
}

```

Ilustración 47: Código de programación del software - 6

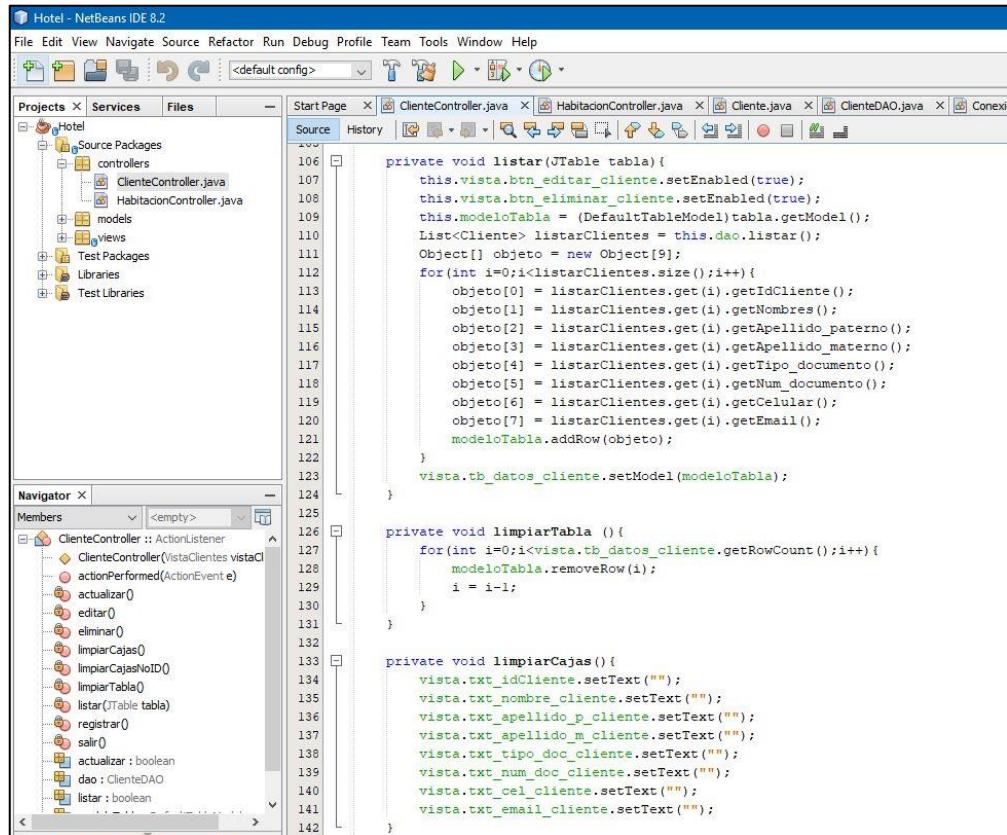


```

Hotel - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Services Files
Start Page ClienteController.java HabitacionController.java Cliente.java ClienteDAO.java
Source History
1 package controllers;
2
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import java.util.List;
6 import javax.swing.JOptionPane;
7 import javax.swing.JTable;
8 import javax.swing.table.DefaultTableModel;
9 import models.Habitacion;
10 import models.HabitacionDAO;
11 import views.VistaHabitaciones;
12
13 public class HabitacionController implements ActionListener{
14     boolean listar;
15     boolean actualizar;
16
17     DefaultTableModel modeloTabla = new DefaultTableModel();
18     VistaHabitaciones vista = new VistaHabitaciones();
19     HabitacionDAO dao = new HabitacionDAO();
20
21     public HabitacionController(VistaHabitaciones vistaHabitaciones){
22         this.vista=vistaHabitaciones;
23         this.vista.btnlistar_hab.addActionListener(this);
24         this.vista.btnnuevo_hab.addActionListener(this);
25         this.vista.btnguardar_hab.addActionListener(this);
26         this.vista.btncancelar_hab.addActionListener(this);
27         this.vista.btnbuscar_hab.addActionListener(this);
28         this.vista.btneliminar_hab.addActionListener(this);
29         this.vista.btnsalir_hab.addActionListener(this);
30         this.vista.btneditar_hab.addActionListener(this);
31     }
32
33     @Override
34     public void actionPerformed(ActionEvent e) {
35
36     }
37 }

```

Ilustración 48: Código de programación del software - 7

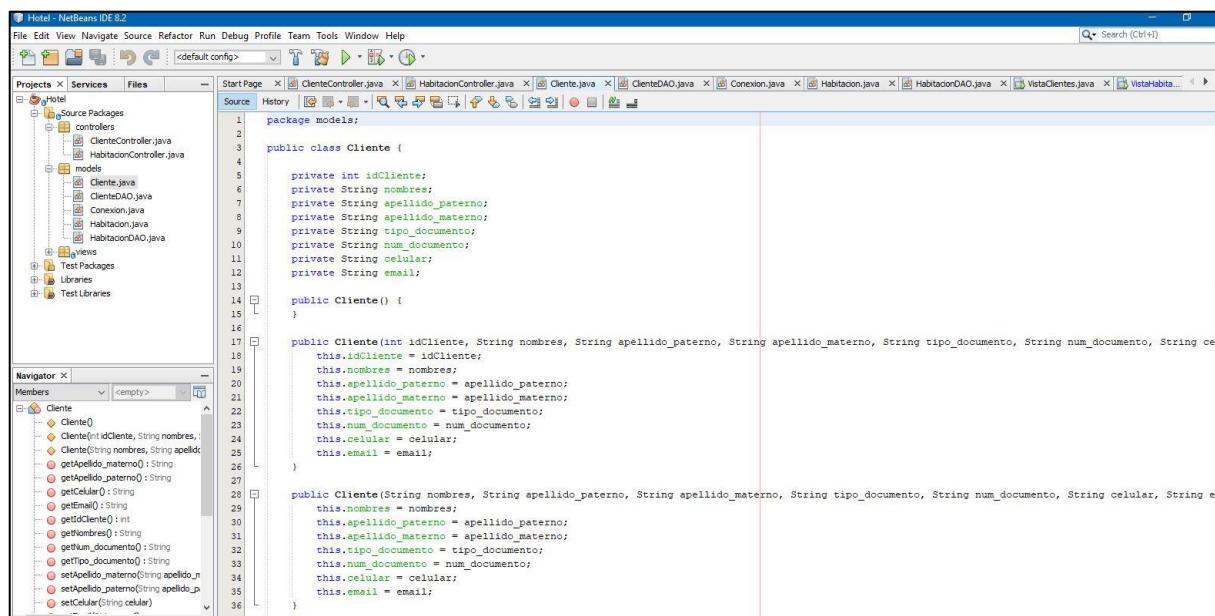


```

Hotel - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Services Files
Start Page ClienteController.java HabitacionController.java Cliente.java ClienteDAO.java
Source History
106     private void listar(JTable tabla) {
107         this.vista.btn_editar_cliente.setEnabled(true);
108         this.vista.btn_eliminar_cliente.setEnabled(true);
109         this.modeloTabla = (DefaultTableModel)tabla.getModel();
110         List<Cliente> listarClientes = this.dao.listar();
111         Object[] objeto = new Object[9];
112         for(int i=0;i<listarClientes.size();i++){
113             objeto[0] = listarClientes.get(i).getIdCliente();
114             objeto[1] = listarClientes.get(i).getNombre();
115             objeto[2] = listarClientes.get(i).getApellido_paterno();
116             objeto[3] = listarClientes.get(i).getApellido_materno();
117             objeto[4] = listarClientes.get(i).getTipo_documento();
118             objeto[5] = listarClientes.get(i).getNum_documento();
119             objeto[6] = listarClientes.get(i).getCelular();
120             objeto[7] = listarClientes.get(i).getEmail();
121             modeloTabla.addRow(objeto);
122         }
123         vista.tb_datos_cliente.setModel(modeloTabla);
124     }
125
126     private void limpiarTabla () {
127         for(int i=0;i<vista.tb_datos_cliente.getRowCount();i++){
128             modeloTabla.removeRow(i);
129             i = i-1;
130         }
131     }
132
133     private void limpiarCajas(){
134         vista.txt_idCliente.setText("");
135         vista.txt_nombre_cliente.setText("");
136         vista.txt_apellido_p_cliente.setText("");
137         vista.txt_apellido_m_cliente.setText("");
138         vista.txt_tipo_doc_cliente.setText("");
139         vista.txt_num_doc_cliente.setText("");
140         vista.txt_cel_cliente.setText("");
141         vista.txt_email_cliente.setText("");
142     }

```

Ilustración 49: Código de programación del software - 8



```

package models;

public class Cliente {

    private int idCliente;
    private String nombres;
    private String apellido_paterno;
    private String apellido_materno;
    private String tipo_documento;
    private String num_documento;
    private String celular;
    private String email;

    public Cliente() {
    }

    public Cliente(int idCliente, String nombres, String apellido_paterno, String apellido_materno, String tipo_documento, String num_documento, String celular, String email) {
        this.idCliente = idCliente;
        this.nombres = nombres;
        this.apellido_paterno = apellido_paterno;
        this.apellido_materno = apellido_materno;
        this.tipo_documento = tipo_documento;
        this.num_documento = num_documento;
        this.celular = celular;
        this.email = email;
    }

    public Cliente(String nombres, String apellido_paterno, String apellido_materno, String tipo_documento, String num_documento, String celular, String email) {
        this.nombres = nombres;
        this.apellido_paterno = apellido_paterno;
        this.apellido_materno = apellido_materno;
        this.tipo_documento = tipo_documento;
        this.num_documento = num_documento;
        this.celular = celular;
        this.email = email;
    }

    public int getIdCliente() {
        return idCliente;
    }

    public void setIdCliente(int idCliente) {
        this.idCliente = idCliente;
    }

    public String getNombres() {
        return nombres;
    }

    public void setNombres(String nombres) {
        this.nombres = nombres;
    }

    public String getApellido_paterno() {
        return apellido_paterno;
    }

    public void setApellido_paterno(String apellido_paterno) {
        this.apellido_paterno = apellido_paterno;
    }

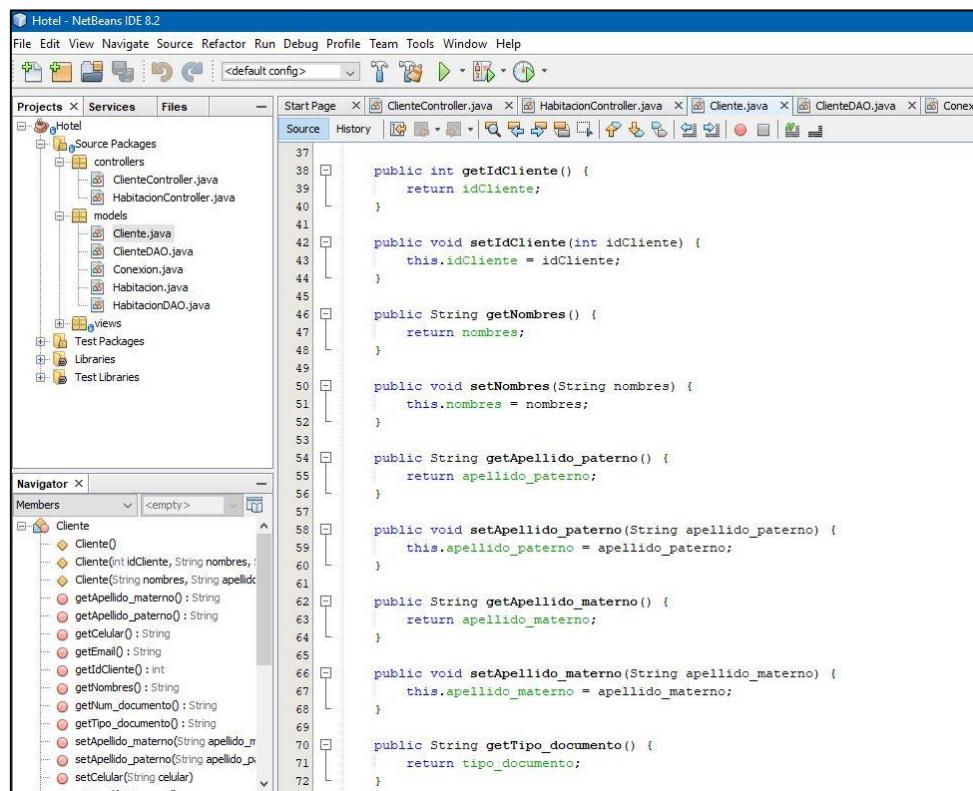
    public String getApellido_materno() {
        return apellido_materno;
    }

    public void setApellido_materno(String apellido_materno) {
        this.apellido_materno = apellido_materno;
    }

    public String getTipo_documento() {
        return tipo_documento;
    }
}

```

Ilustración 50: Código de programación del software - 9



```

public int getIdCliente() {
    return idCliente;
}

public void setIdCliente(int idCliente) {
    this.idCliente = idCliente;
}

public String getNombres() {
    return nombres;
}

public void setNombres(String nombres) {
    this.nombres = nombres;
}

public String getApellido_paterno() {
    return apellido_paterno;
}

public void setApellido_paterno(String apellido_paterno) {
    this.apellido_paterno = apellido_paterno;
}

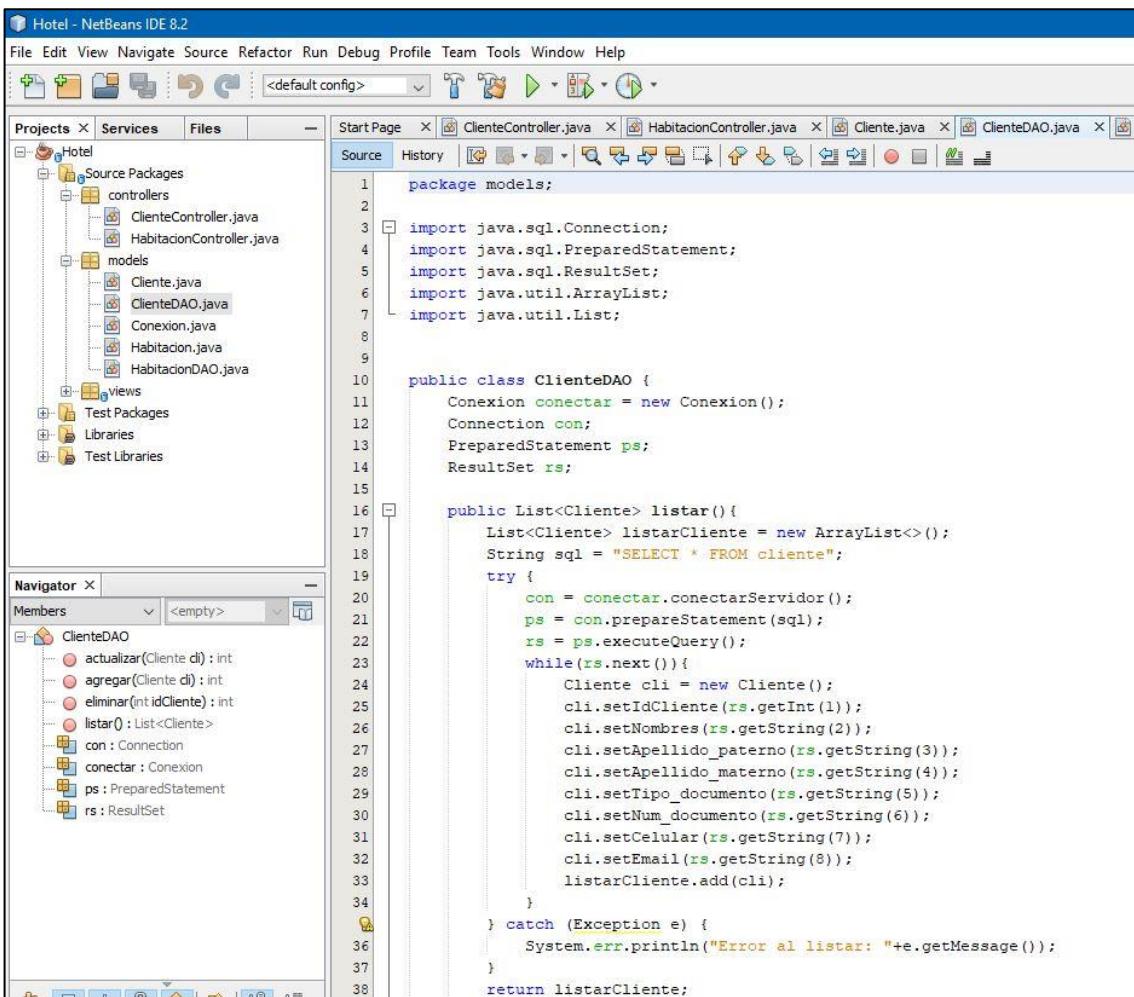
public String getApellido_materno() {
    return apellido_materno;
}

public void setApellido_materno(String apellido_materno) {
    this.apellido_materno = apellido_materno;
}

public String getTipo_documento() {
    return tipo_documento;
}

```

Ilustración 51: Código de programación del software - 10



```

package models;

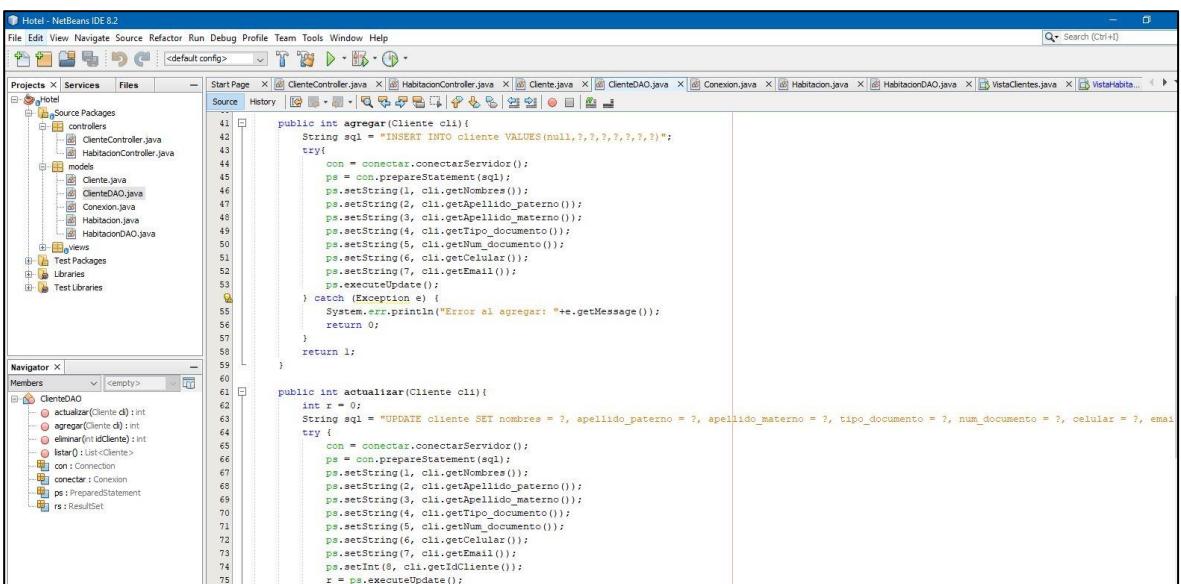
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

public class ClienteDAO {
    Conexion conectar = new Conexion();
    Connection con;
    PreparedStatement ps;
    ResultSet rs;

    public List<Cliente> listar(){
        List<Cliente> listarCliente = new ArrayList<>();
        String sql = "SELECT * FROM cliente";
        try {
            con = conectar.conectarServidor();
            ps = con.prepareStatement(sql);
            rs = ps.executeQuery();
            while(rs.next()){
                Cliente cli = new Cliente();
                cli.setIdCliente(rs.getInt(1));
                cli.setNombres(rs.getString(2));
                cli.setApellido_paterno(rs.getString(3));
                cli.setApellido_materno(rs.getString(4));
                cli.setTipo_documento(rs.getString(5));
                cli.setNum_documento(rs.getString(6));
                cli.setCelular(rs.getString(7));
                cli.setEmail(rs.getString(8));
                listarCliente.add(cli);
            }
        } catch (Exception e) {
            System.err.println("Error al listar: "+e.getMessage());
        }
        return listarCliente;
    }
}

```

Ilustración 52: Código de programación del software - 11



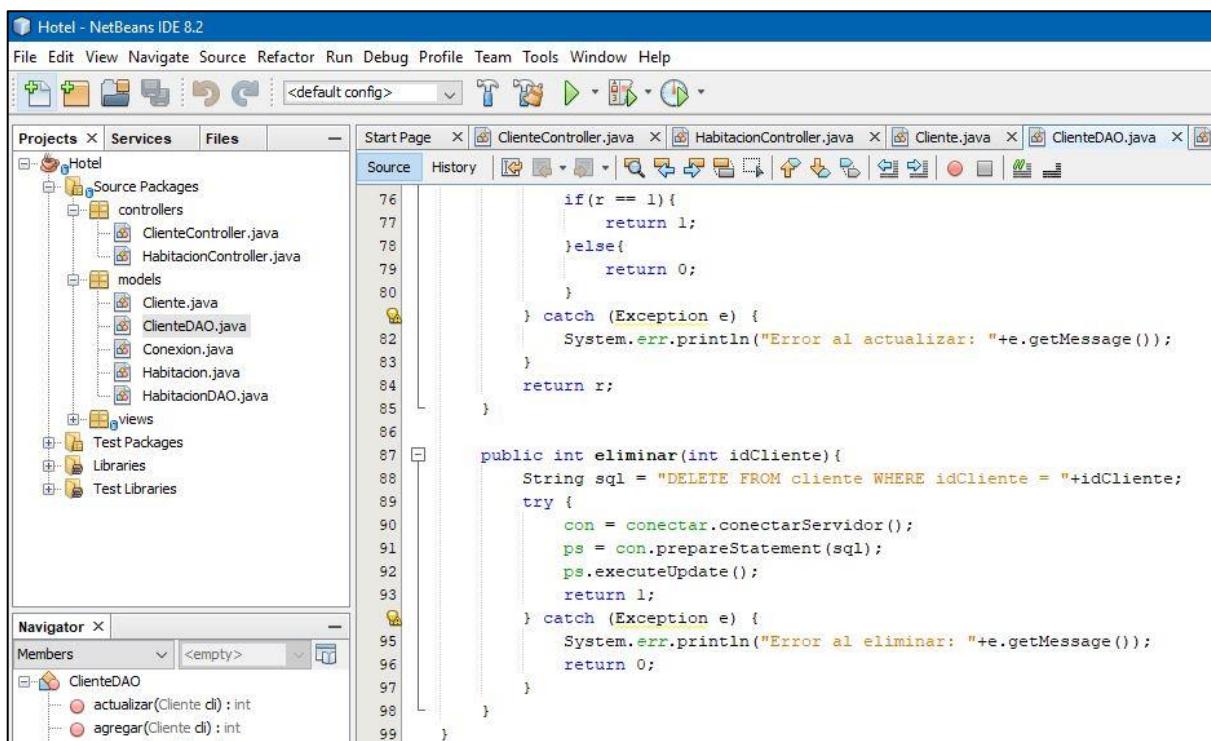
```

public int agregar(Cliente cli){
    String sql = "INSERT INTO cliente VALUES(null, ?, ?, ?, ?, ?, ?)";
    try{
        con = conectar.conectarServidor();
        ps = con.prepareStatement(sql);
        ps.setString(1, cli.getNombres());
        ps.setString(2, cli.getApellido_paterno());
        ps.setString(3, cli.getApellido_materno());
        ps.setString(4, cli.getTipo_documento());
        ps.setString(5, cli.getNum_documento());
        ps.setString(6, cli.getCelular());
        ps.setString(7, cli.getEmail());
        ps.executeUpdate();
    } catch (Exception e) {
        System.err.println("Error al agregar: "+e.getMessage());
        return 0;
    }
    return 1;
}

public int actualizar(Cliente cli){
    int r = 0;
    String sql = "UPDATE cliente SET nombres = ?, apellido_paterno = ?, apellido_materno = ?, tipo_documento = ?, num_documento = ?, celular = ?, email = ? WHERE idcliente = ?";
    try{
        con = conectar.conectarServidor();
        ps = con.prepareStatement(sql);
        ps.setString(1, cli.getNombres());
        ps.setString(2, cli.getApellido_paterno());
        ps.setString(3, cli.getApellido_materno());
        ps.setString(4, cli.getTipo_documento());
        ps.setString(5, cli.getNum_documento());
        ps.setString(6, cli.getCelular());
        ps.setString(7, cli.getEmail());
        ps.setInt(8, cli.getIdCliente());
        r = ps.executeUpdate();
    } catch (Exception e) {
        System.err.println("Error al actualizar: "+e.getMessage());
    }
    return r;
}

```

Ilustración 53: Código de programación del software - 12



```

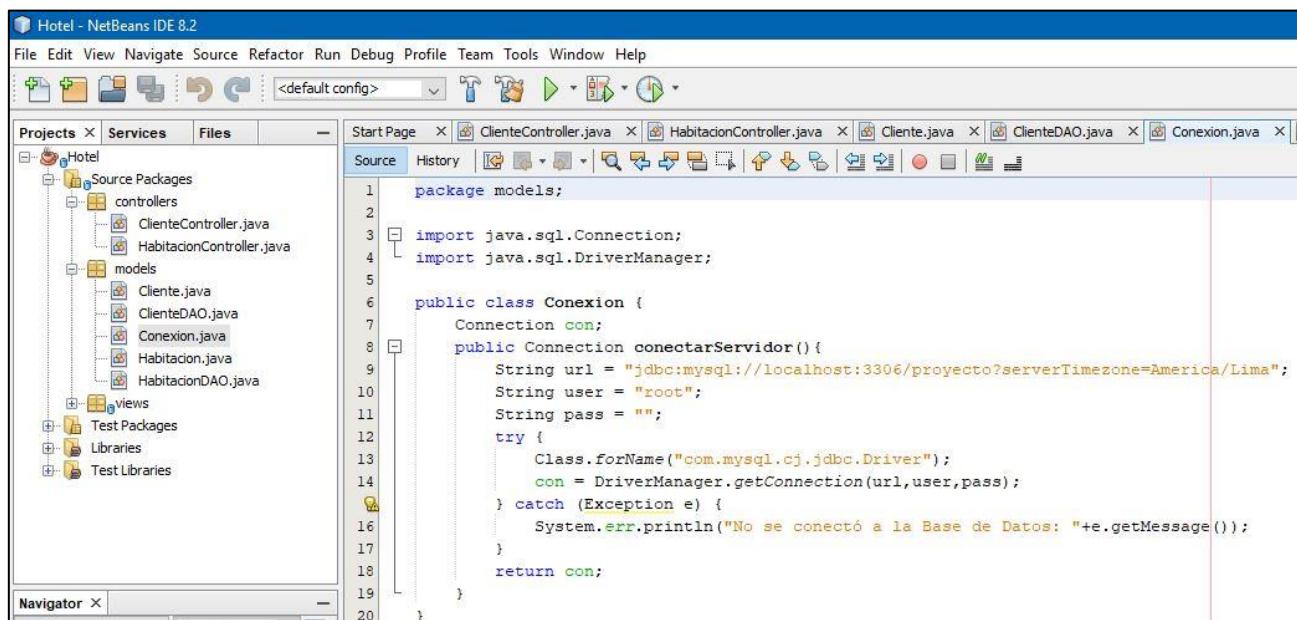
Hotel - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects X Services Files
Source Packages
  controllers
    ClienteController.java
    HabitacionController.java
  models
    Cliente.java
    ClienteDAO.java
    Conexion.java
    Habitacion.java
    HabitacionDAO.java
    views
Test Packages
Libraries
Test Libraries

Start Page X ClienteController.java X HabitacionController.java X Cliente.java X ClienteDAO.java X
Source History <default config> T T D G P S
76 if(r == 1){
77     return 1;
78 }else{
79     return 0;
}
80 } catch (Exception e) {
81     System.err.println("Error al actualizar: "+e.getMessage());
82 }
83
84 return r;
}

public int eliminar(int idCliente){
String sql = "DELETE FROM cliente WHERE idCliente = "+idCliente;
try {
85 con = conectar.conectarServidor();
86 ps = con.prepareStatement(sql);
87 ps.executeUpdate();
88 return 1;
} catch (Exception e) {
89     System.err.println("Error al eliminar: "+e.getMessage());
90 }
91
92 return 0;
}
93
94
95
96
97
98
99
}

```

Ilustración 54: Código de programación del software - 13



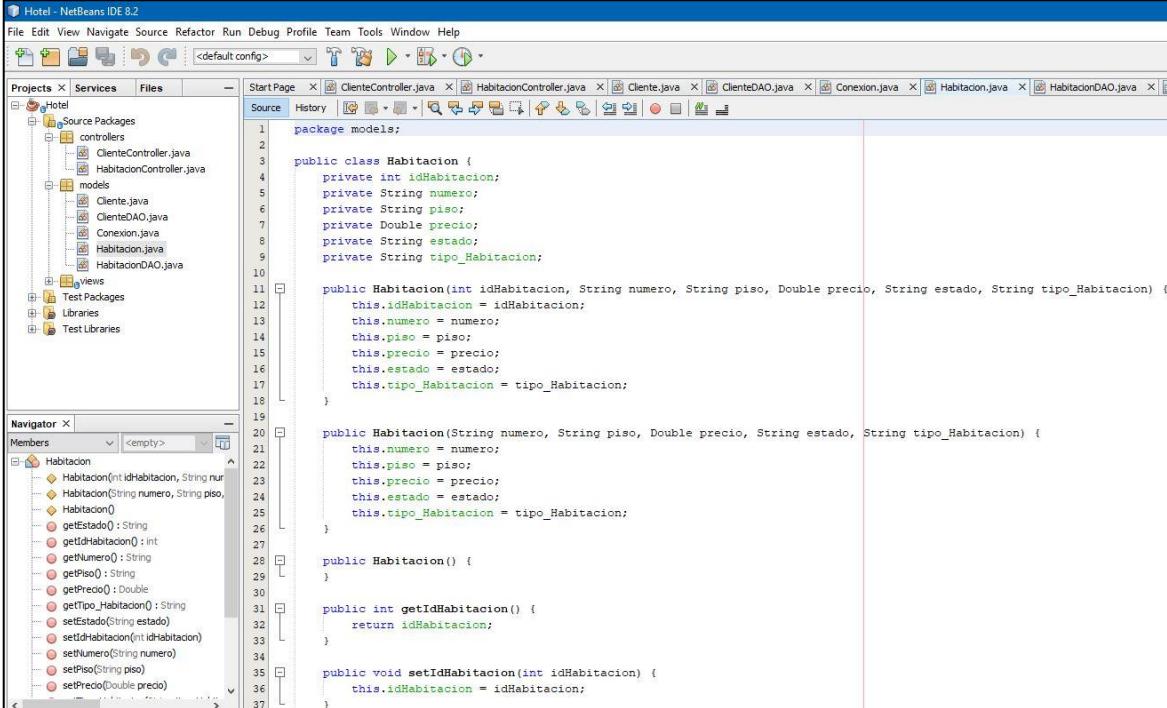
```

Hotel - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects X Services Files
Source Packages
  controllers
    ClienteController.java
    HabitacionController.java
  models
    Cliente.java
    ClienteDAO.java
    Conexion.java
    Habitacion.java
    HabitacionDAO.java
    views
Test Packages
Libraries
Test Libraries

Start Page X ClienteController.java X HabitacionController.java X Cliente.java X ClienteDAO.java X Conexion.java X
Source History <default config> T T D G P S
1 package models;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5
6 public class Conexion {
7     Connection con;
8     public Connection conectarServidor(){
9         String url = "jdbc:mysql://localhost:3306/proyecto?serverTimezone=America/Lima";
10        String user = "root";
11        String pass = "";
12        try {
13            Class.forName("com.mysql.cj.jdbc.Driver");
14            con = DriverManager.getConnection(url,user,pass);
15        } catch (Exception e) {
16            System.err.println("No se conectó a la Base de Datos: "+e.getMessage());
17        }
18        return con;
19    }
20
}

```

Ilustración 55: Código de programación del software - 14



```

package models;

public class Habitacion {
    private int idHabitacion;
    private String numero;
    private String piso;
    private Double precio;
    private String estado;
    private String tipo_Habitacion;

    public Habitacion(int idHabitacion, String numero, String piso, Double precio, String estado, String tipo_Habitacion) {
        this.idHabitacion = idHabitacion;
        this.numero = numero;
        this.piso = piso;
        this.precio = precio;
        this.estado = estado;
        this.tipo_Habitacion = tipo_Habitacion;
    }

    public Habitacion(String numero, String piso, Double precio, String estado, String tipo_Habitacion) {
        this.numero = numero;
        this.piso = piso;
        this.precio = precio;
        this.estado = estado;
        this.tipo_Habitacion = tipo_Habitacion;
    }

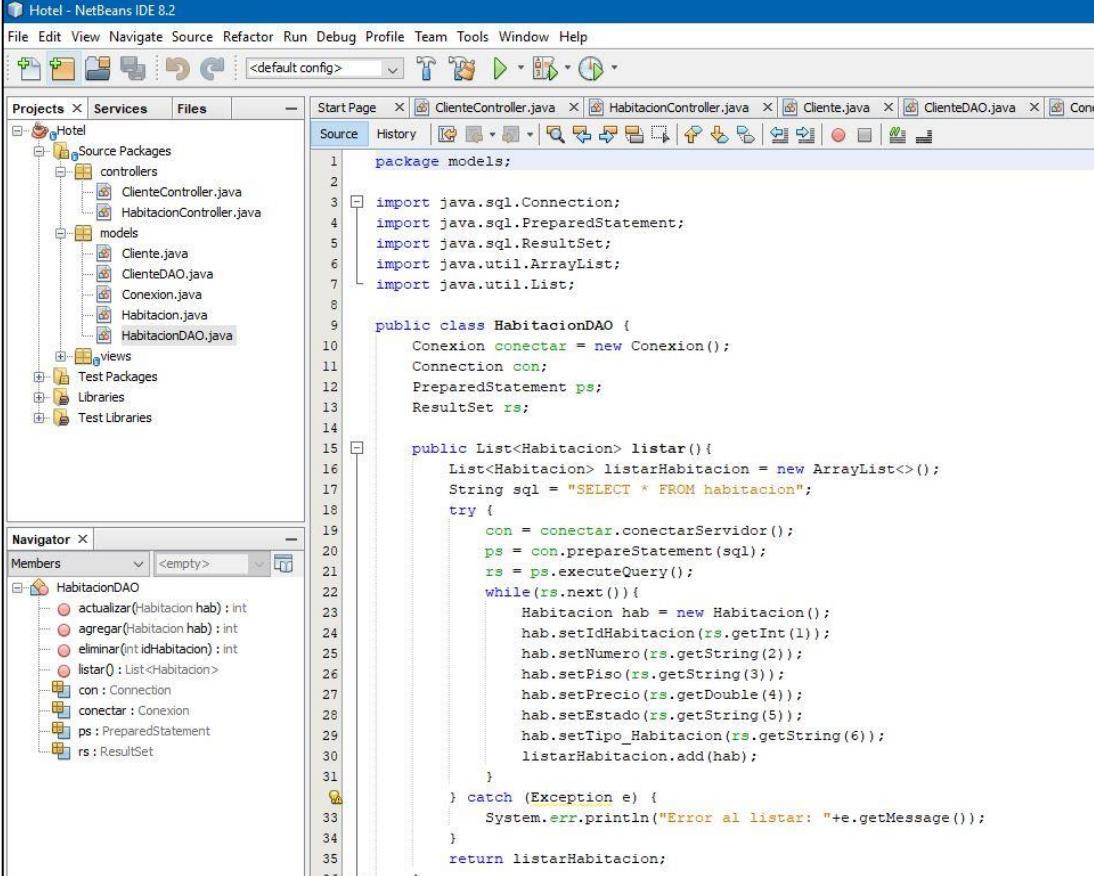
    public Habitacion() {
    }

    public int getIdHabitacion() {
        return idHabitacion;
    }

    public void setIdHabitacion(int idHabitacion) {
        this.idHabitacion = idHabitacion;
    }
}

```

Ilustración 56: Código de programación del software - 15



```

package models;

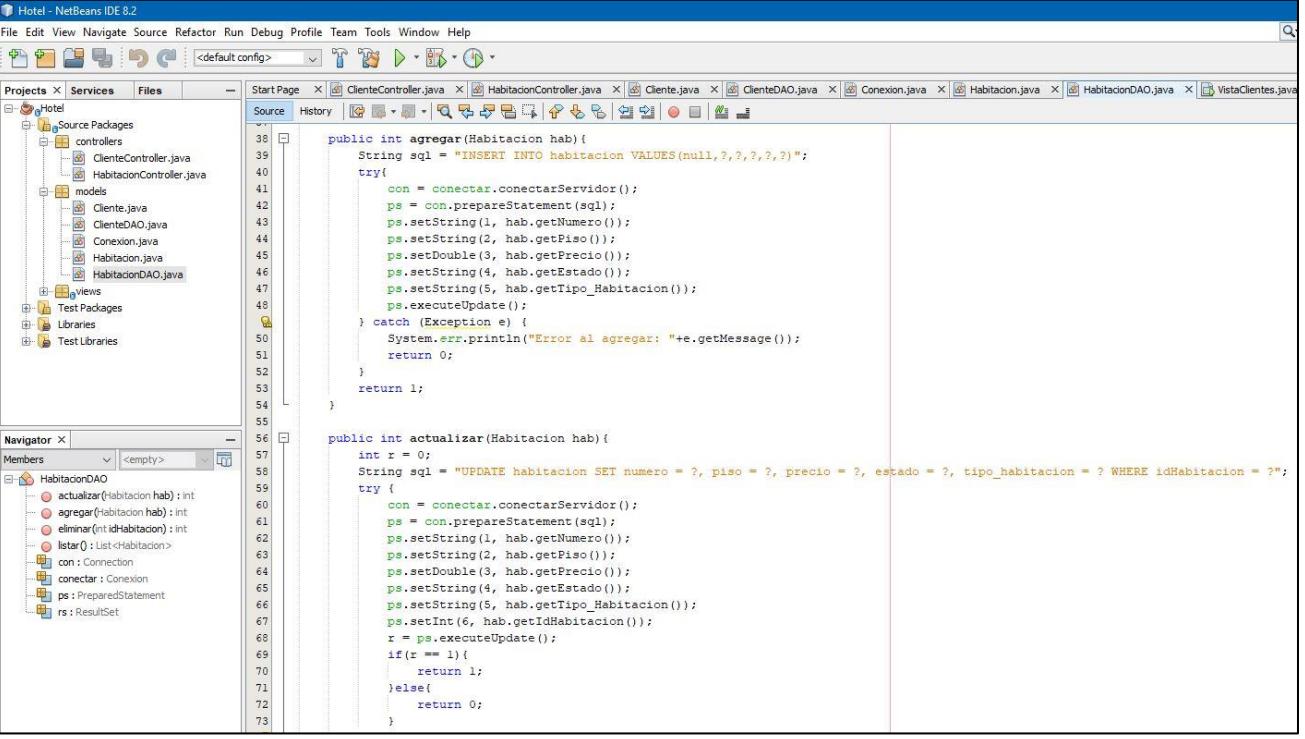
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

public class HabitacionDAO {
    Conexion conectar = new Conexion();
    Connection con;
    PreparedStatement ps;
    ResultSet rs;

    public List<Habitacion> listar(){
        List<Habitacion> listarHabitacion = new ArrayList<>();
        String sql = "SELECT * FROM habitacion";
        try {
            con = conectar.conectarServidor();
            ps = con.prepareStatement(sql);
            rs = ps.executeQuery();
            while(rs.next()){
                Habitacion hab = new Habitacion();
                hab.setIdHabitacion(rs.getInt(1));
                hab.setNumero(rs.getString(2));
                hab.setPiso(rs.getString(3));
                hab.setPrecio(rs.getDouble(4));
                hab.setEstado(rs.getString(5));
                hab.setTipo_Habitacion(rs.getString(6));
                listarHabitacion.add(hab);
            }
        } catch (Exception e) {
            System.out.println("Error al listar: "+e.getMessage());
        }
        return listarHabitacion;
    }
}

```

Ilustración 57: Código de programación del software - 16



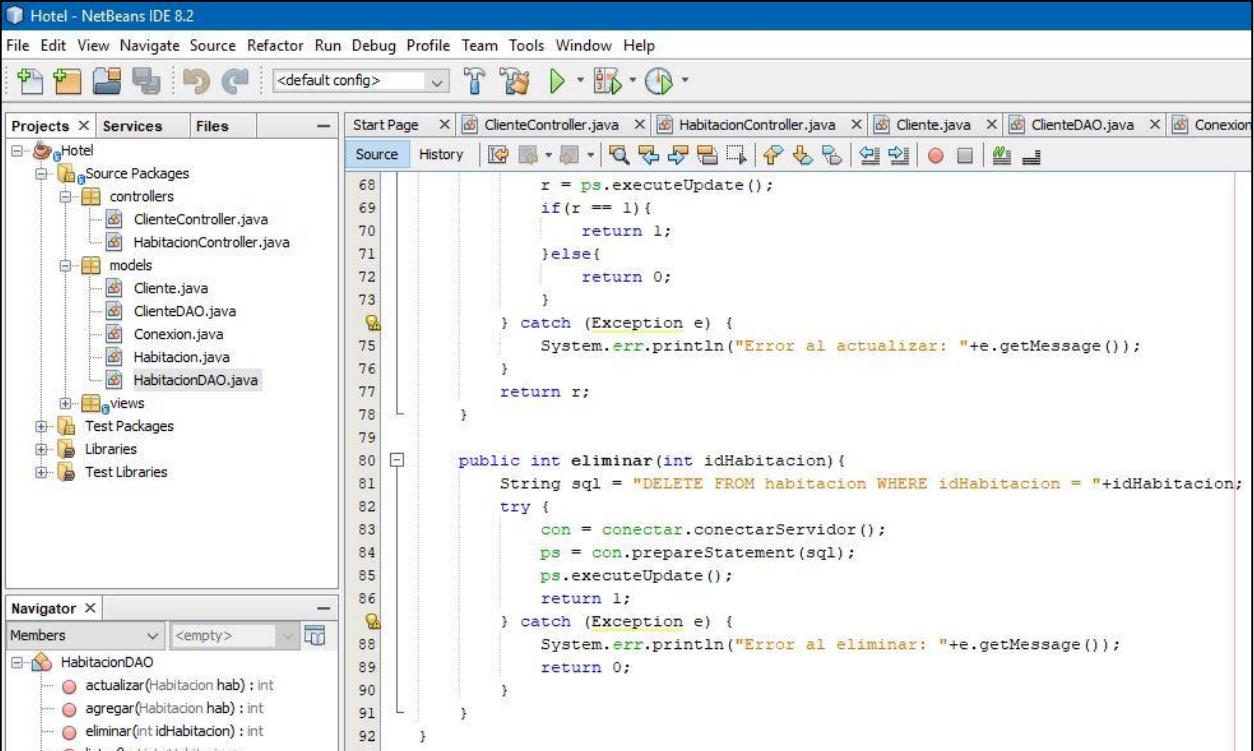
```

public int agregar(Habitacion hab){
    String sql = "INSERT INTO habitacion VALUES(null, ?, ?, ?, ?, ?)";
    try{
        con = conectar.conectarServidor();
        ps = con.prepareStatement(sql);
        ps.setString(1, hab.getIdHabitacion());
        ps.setString(2, hab.getPiso());
        ps.setDouble(3, hab.getPrecio());
        ps.setString(4, hab.getEstado());
        ps.setString(5, hab.getTipo_Habitacion());
        ps.executeUpdate();
    } catch (Exception e) {
        System.err.println("Error al agregar: "+e.getMessage());
        return 0;
    }
    return 1;
}

public int actualizar(Habitacion hab){
    int r = 0;
    String sql = "UPDATE habitacion SET numero = ?, piso = ?, precio = ?, estado = ?, tipo_habitacion = ? WHERE idHabitacion = ?";
    try {
        con = conectar.conectarServidor();
        ps = con.prepareStatement(sql);
        ps.setString(1, hab.getIdHabitacion());
        ps.setString(2, hab.getPiso());
        ps.setDouble(3, hab.getPrecio());
        ps.setString(4, hab.getEstado());
        ps.setString(5, hab.getTipo_Habitacion());
        ps.setInt(6, hab.getIdHabitacion());
        r = ps.executeUpdate();
        if(r == 1){
            return 1;
        }else{
            return 0;
        }
    } catch (Exception e) {
        System.err.println("Error al actualizar: "+e.getMessage());
    }
    return r;
}

```

Ilustración 58: Código de programación del software - 17



```

r = ps.executeUpdate();
if(r == 1){
    return 1;
}else{
    return 0;
}

} catch (Exception e) {
    System.err.println("Error al actualizar: "+e.getMessage());
}
return r;
}

public int eliminar(int idHabitacion){
    String sql = "DELETE FROM habitacion WHERE idHabitacion = "+idHabitacion;
    try {
        con = conectar.conectarServidor();
        ps = con.prepareStatement(sql);
        ps.executeUpdate();
        return 1;
    } catch (Exception e) {
        System.err.println("Error al eliminar: "+e.getMessage());
        return 0;
    }
}

```

Ilustración 59: Código de programación del software - 18

ID	ITEM	DESCRIPCIÓN
1	Nombre Corto	Registro de Trabajador
2	Actores	Administrador
3	Objetivo	Ingresar un nuevo trabajador a nuestro sistema.
4	Disparador	El administrador desea registrar un nuevo trabajador en el sistema
5	Pre condiciones	Se debe validar el trabajador mediante el Login
6	Post condiciones	El trabajador será registrado exitosamente.
7	Escenario básico	<p>Escenario 1: Validar trabajador.</p> <ul style="list-style-type: none"> <li>• Administrador ingresa su usuario y su contraseña</li> <li>• El sistema valida el usuario y contraseña</li> </ul> <p>Escenario 2: Consulta de trabajadores</p> <ul style="list-style-type: none"> <li>• Administrador solicita la consulta de todos los trabajadores.</li> <li>• El sistema le muestra los datos de todos los trabajadores.</li> </ul> <p>Escenario 3: Modificar al trabajador</p> <ul style="list-style-type: none"> <li>• Administrador solicita modificar los datos del trabajador</li> <li>• El sistema le permite modificar los datos del trabajador y se realizan los cambios respectivos</li> </ul> <p>Escenario 4: Eliminar al trabajador</p> <ul style="list-style-type: none"> <li>• Administrador solicita eliminar un trabajador</li> <li>• El sistema le permite eliminar al trabajador.</li> </ul> <ul style="list-style-type: none"> <li>➤ El administrador registra todos los datos del trabajador</li> <li>➤ El sistema valida todos los datos del trabajador</li> <li>➤ El sistema registra el nombre del trabajador</li> <li>➤ El sistema registra el apellido paterno del trabajador</li> <li>➤ El sistema registra el apellido materno del trabajador</li> <li>➤ El sistema registra el tipo de documento del trabajador</li> <li>➤ El sistema registra el número de documento del trabajador</li> <li>➤ El sistema registra el celular del trabajador</li> <li>➤ El sistema registra el email del trabajador</li> <li>➤ El sistema registra el sueldo del trabajador</li> </ul>

		<ul style="list-style-type: none"> <li>➤ El sistema registra el acceso del trabajador</li> <li>➤ El sistema registra el login del trabajador</li> <li>➤ El sistema registra el password del trabajador</li> <li>➤ El sistema registra el estado del trabajador</li> </ul>
8	Escenario Alternativo	<ul style="list-style-type: none"> <li>➤ Si no se registra el trabajador, enviar mensaje de ayuda.</li> <li>➤ Si faltan datos del trabajador, enviar mensaje de ayuda.</li> <li>➤ Si el usuario no es válido, enviar mensaje de ayuda.</li> <li>➤ Si la contraseña no es válida, enviar mensaje de ayuda.</li> <li>➤ Si se consulta datos de un trabajador no seleccionado, enviar mensaje de ayuda.</li> <li>➤ Si se modifica datos de un trabajador no seleccionado, enviar mensaje de ayuda.</li> <li>➤ Si se elimina un trabajador no seleccionado, enviar mensaje de ayuda.</li> </ul>
9	Prioridad	Versión 1

Tabla 3: Especificación de Casos de uso - Versión 1 - 1

ID	ITEM	DESCRIPCIÓN
1	Nombre Corto	Registro de Clientes
2	Actores	Administrador y Recepcionista
3	Objetivo	Ingresar un nuevo cliente a nuestro sistema.
4	Disparador	El administrador o recepcionista desea registrar un nuevo cliente en el sistema
5	Pre condiciones	Se debe validar el trabajador mediante el Login
6	Post condiciones	El cliente será registrado exitosamente.
7	Escenario básico	<p>Escenario 1: Validar trabajador.</p> <ul style="list-style-type: none"> <li>• Administrador o recepcionista ingresa su usuario y su contraseña</li> <li>• El sistema valida el usuario y contraseña</li> </ul> <p>Escenario 2: Consulta de clientes</p> <ul style="list-style-type: none"> <li>• Administrador o recepcionista solicita la consulta de todos los clientes.</li> <li>• El sistema le muestra los datos de todos los clientes.</li> </ul> <p>Escenario 3: Modificar al cliente</p> <ul style="list-style-type: none"> <li>• Administrador o recepcionista solicita modificar los datos del cliente</li> <li>• El sistema le permite modificar los datos del cliente y se realizan los cambios respectivos</li> </ul> <p>Escenario 4: Eliminar al cliente</p> <ul style="list-style-type: none"> <li>• Administrador o recepcionista solicita eliminar un cliente</li> <li>• El sistema le permite eliminar al cliente.</li> </ul> <ul style="list-style-type: none"> <li>➤ El administrador o recepcionista registra todos los datos del cliente</li> <li>➤ El sistema valida todos los datos del cliente</li> <li>➤ El sistema registra el nombre del cliente</li> <li>➤ El sistema registra el apellido paterno del cliente</li> <li>➤ El sistema registra el apellido materno del cliente</li> </ul>

		<ul style="list-style-type: none"> <li>➤ El sistema registra el tipo de documento del cliente</li> <li>➤ El sistema registra el número de documento del cliente</li> <li>➤ El sistema registra el celular del cliente</li> <li>➤ El sistema registra el email del cliente</li> </ul>
8	Escenario Alternativo	<ul style="list-style-type: none"> <li>➤ Si no se registra el cliente, enviar mensaje de ayuda.</li> <li>➤ Si faltan datos del cliente, enviar mensaje de ayuda.</li> <li>➤ Si el usuario no es válido, enviar mensaje de ayuda.</li> <li>➤ Si la contraseña no es válida, enviar mensaje de ayuda.</li> <li>➤ Si se consulta datos de un cliente no seleccionado, enviar mensaje de ayuda.</li> <li>➤ Si se modifica datos de un cliente no seleccionado, enviar mensaje de ayuda.</li> <li>➤ Si se elimina un cliente no seleccionado, enviar mensaje de ayuda.</li> </ul>
9	Prioridad	Versión 1

Tabla 4: Especificación de Casos de uso - Versión 1 - 2

ID	ITEM	DESCRIPCIÓN
1	Nombre Corto	Registro de ingreso de cliente
2	Actores	Administrador
3	Objetivo	Ingresar el ingreso de cliente a nuestro sistema.
4	Disparador	El administrador desea registrar el ingreso del cliente en el sistema
5	Pre condiciones	Se debe validar el trabajador mediante el Login
6	Post condiciones	El ingreso de cliente será registrado exitosamente.
7	Escenario básico	<p>Escenario 1: Validar trabajador.</p> <ul style="list-style-type: none"> <li>• Administrador ingresa su usuario y su contraseña</li> <li>• El sistema valida el usuario y contraseña</li> </ul> <p>Escenario 2: Consulta de ingreso de clientes</p> <ul style="list-style-type: none"> <li>• Administrador solicita la consulta de ingreso de todos los clientes.</li> <li>• El sistema le muestra el ingreso de todos los clientes.</li> </ul> <p>Escenario 3: Modificar el ingreso de cliente</p> <ul style="list-style-type: none"> <li>• Administrador solicita modificar el ingreso del cliente</li> <li>• El sistema le permite modificar el ingreso del cliente y se realizan los cambios respectivos</li> </ul> <p>Escenario 4: Eliminar el ingreso del cliente</p> <ul style="list-style-type: none"> <li>• Administrador solicita eliminar el ingreso de cliente</li> <li>• El sistema le permite eliminar el ingreso de cliente.</li> </ul> <p>➤ El administrador registra el ingreso del cliente      ➤ El sistema valida los datos de ingreso del cliente      ➤ El sistema registra el ingreso del cliente</p>
8	Escenario Alternativo	<p>➤ Si no se registra el ingreso del cliente, enviar mensaje de ayuda.</p> <p>➤ Si faltan datos en el ingreso de cliente, enviar mensaje de ayuda.</p> <p>➤ Si el usuario no es válido, enviar mensaje de ayuda.</p> <p>➤ Si la contraseña no es válida, enviar mensaje de ayuda.</p>

		<ul style="list-style-type: none"><li>➤ Si se consulta el ingreso del cliente no seleccionado, enviar mensaje de ayuda.</li><li>➤ Si se modifica el ingreso del cliente no seleccionado, enviar mensaje de ayuda.</li><li>➤ Si se elimina el ingreso del cliente no seleccionado, enviar mensaje de ayuda.</li></ul>
9	Prioridad	Versión 1

Tabla 5: Especificación de Casos de uso - Versión 1 - 3

ID	ITEM	DESCRIPCIÓN
1	Nombre Corto	Registro de salida de cliente
2	Actores	Administrador
3	Objetivo	Ingresar la salida de cliente a nuestro sistema.
4	Disparador	El administrador desea registrar la salida del cliente en el sistema
5	Pre condiciones	Se debe validar el trabajador mediante el Login
6	Post condiciones	La salida de cliente será registrada exitosamente.
7	Escenario básico	<p>Escenario 1: Validar trabajador.</p> <ul style="list-style-type: none"> <li>• Administrador ingresa su usuario y su contraseña</li> <li>• El sistema valida el usuario y contraseña</li> </ul> <p>Escenario 2: Consulta de salida de clientes</p> <ul style="list-style-type: none"> <li>• Administrador solicita la consulta de salida de todos los clientes.</li> <li>• El sistema le muestra la salida de todos los clientes.</li> </ul> <p>Escenario 3: Modificar la salida de cliente</p> <ul style="list-style-type: none"> <li>• Administrador solicita modificar la salida del cliente</li> <li>• El sistema le permite modificar la salida del cliente y se realizan los cambios respectivos</li> </ul> <p>Escenario 4: Eliminar la salida del cliente</p> <ul style="list-style-type: none"> <li>• Administrador solicita eliminar la salida de cliente</li> <li>• El sistema le permite eliminar la salida de cliente.</li> </ul> <p>➤ El administrador registra la salida del cliente</p> <p>➤ El sistema valida los datos de salida del cliente</p> <p>➤ El sistema registra la salida del cliente</p>
8	Escenario Alternativo	<p>➤ Si no se registra la salida del cliente, enviar mensaje de ayuda.</p> <p>➤ Si faltan datos en la salida de cliente, enviar mensaje de ayuda.</p> <p>➤ Si el usuario no es válido, enviar mensaje de ayuda.</p> <p>➤ Si la contraseña no es válida, enviar mensaje de ayuda.</p>

		<ul style="list-style-type: none"><li>➤ Si se consulta la salida del cliente no seleccionado, enviar mensaje de ayuda.</li><li>➤ Si se modifica la salida del cliente no seleccionado, enviar mensaje de ayuda.</li><li>➤ Si se elimina la salida del cliente no seleccionado, enviar mensaje de ayuda.</li></ul>
9	Prioridad	Versión 1

Tabla 6: Especificación de Casos de uso - Versión 1 - 4

ID	ITEM	DESCRIPCIÓN
1	Nombre Corto	Registro de Habitaciones
2	Actores	Administrador
3	Objetivo	Ingresar una habitación a nuestro sistema.
4	Disparador	El administrador desea registrar una habitación en el sistema
5	Pre condiciones	Se debe validar el trabajador mediante el Login
6	Post condiciones	La habitación será registrada exitosamente.
7	Escenario básico	<p>Escenario 1: Validar trabajador.</p> <ul style="list-style-type: none"> <li>• Administrador ingresa su usuario y su contraseña</li> <li>• El sistema valida el usuario y contraseña</li> </ul> <p>Escenario 2: Consulta de habitaciones</p> <ul style="list-style-type: none"> <li>• Administrador solicita la consulta de todas las habitaciones.</li> <li>• El sistema le muestra los datos de todas las habitaciones.</li> </ul> <p>Escenario 3: Modificar habitación</p> <ul style="list-style-type: none"> <li>• Administrador solicita modificar los datos de la habitación</li> <li>• El sistema le permite modificar los datos de la habitación y se realizan los cambios respectivos</li> </ul> <p>Escenario 4: Eliminar habitación</p> <ul style="list-style-type: none"> <li>• Administrador solicita eliminar una habitación</li> <li>• El sistema le permite eliminar la habitación</li> </ul> <ul style="list-style-type: none"> <li>➤ El administrador registra todos los datos de la habitación</li> <li>➤ El sistema valida todos los datos de la habitación</li> <li>➤ El sistema registra el número de la habitación</li> <li>➤ El sistema registra el piso de la habitación</li> <li>➤ El sistema registra el precio de la habitación</li> <li>➤ El sistema registra el tipo de habitación</li> <li>➤ El sistema registra el estado de la habitación</li> </ul>
8	Escenario Alternativo	<ul style="list-style-type: none"> <li>➤ Si no se registra la habitación, enviar mensaje de ayuda.</li> <li>➤ Si faltan datos de la habitación, enviar mensaje de ayuda.</li> </ul>

		<ul style="list-style-type: none"> <li>➤ Si el usuario no es válido, enviar mensaje de ayuda.</li> <li>➤ Si la contraseña no es válida, enviar mensaje de ayuda.</li> <li>➤ Si se consulta datos de una habitación no seleccionada, enviar mensaje de ayuda.</li> <li>➤ Si se modifica datos de una habitación no seleccionada, enviar mensaje de ayuda.</li> <li>➤ Si se elimina una habitación no seleccionada, enviar mensaje de ayuda.</li> </ul>
9	Prioridad	Versión 2

Tabla 7: Especificación de Casos de uso - Versión 2 - 1

ID	ITEM	DESCRIPCIÓN
1	Nombre Corto	Registro de Reserva
2	Actores	Administrador y Recepcionista
3	Objetivo	Ingresar nueva reserva a nuestro sistema
4	Disparador	El administrador o recepcionista desea registrar una reserva en el sistema
5	Pre condiciones	<ul style="list-style-type: none"> <li>• Validar trabajador mediante el Login</li> <li>• Validar registro del cliente</li> </ul>
6	Post condiciones	La reserva será registrada exitosamente.
7	Escenario básico	<p>Escenario 1: Validar trabajador.</p> <ul style="list-style-type: none"> <li>• Administrador o recepcionista ingresa su usuario y su contraseña</li> <li>• El sistema valida el usuario y contraseña</li> </ul> <p>Escenario 2: Consulta de reservas</p> <ul style="list-style-type: none"> <li>• Administrador o recepcionista solicita la consulta de todas las reservas.</li> <li>• El sistema le muestra los datos de todas las reservas.</li> </ul> <p>Escenario 3: Modificar reserva</p> <ul style="list-style-type: none"> <li>• Administrador o recepcionista solicita modificar reserva del cliente.</li> <li>• El sistema le permite modificar los datos de la reserva y se realizan los cambios respectivos</li> </ul> <p>Escenario 4: Eliminar reserva</p> <ul style="list-style-type: none"> <li>• Administrador o recepcionista solicita eliminar una reserva.</li> <li>• El sistema le permite eliminar la reserva.</li> </ul> <p>➤ El administrador o recepcionista registra todos los datos de la reserva</p> <p>➤ El sistema valida todos los datos de la reserva</p> <p>➤ El sistema registra el tipo de reserva</p> <p>➤ El sistema registra la fecha de reserva</p>

		<ul style="list-style-type: none"> <li>➤ El sistema registra la fecha de ingreso de la reserva</li> <li>➤ El sistema registra la fecha de salida de la reserva</li> <li>➤ El sistema registra el costo de la reserva</li> <li>➤ El sistema registra el estado de la reserva</li> </ul>
8	Escenario Alternativo	<ul style="list-style-type: none"> <li>➤ Si no se registra la reserva, enviar mensaje de ayuda.</li> <li>➤ Si faltan datos de la reserva, enviar mensaje de ayuda.</li> <li>➤ Si el usuario no es válido, enviar mensaje de ayuda.</li> <li>➤ Si la contraseña no es válida, enviar mensaje de ayuda.</li> <li>➤ Si se consulta datos de una reserva no seleccionada, enviar mensaje de ayuda.</li> <li>➤ Si se modifica datos de una reserva no seleccionada, enviar mensaje de ayuda.</li> <li>➤ Si se elimina una reserva no seleccionada, enviar mensaje de ayuda.</li> </ul>
9	Prioridad	Versión 2

Tabla 8: Especificación de Casos de uso - Versión 2 - 2

Caso de uso	Datos de Entrada normal	Resultado Esperado normal	Datos de entrada anómalo	Resultado esperado anómalo
Registro cliente	Nombres Apellido Paterno Apellido Materno Tipo de Documento Número de Documento Celular Email	Cliente registrado con éxito	-Nombre en espacios -Apellido en espacios -Tipo de documento incorrecto -Número de documento inexistente -Faltan dígitos -Email ingresado incorrectamente	Debe completar todos los campos
Eliminar cliente	Seleccionar cliente de la tabla	Cliente eliminado con éxito	No selecciona cliente de la tabla	Debe seleccionar una fila
Modificar cliente	Nombres Apellido Paterno Apellido Materno Tipo de Documento Número de Documento Celular Email	Cliente actualizado con éxito	-Nombre en espacios -Apellido en espacios -Tipo de documento incorrecto -Número de documento inexistente -Falta dígitos -Email ingresado incorrectamente	Debe completar todos los campos
Consultar cliente	Si existen clientes en la base de datos	Cliente listado con éxito	No existen clientes en la base de datos	No hay clientes para listar
Registrar trabajador	Nombres Apellido Paterno Apellido Materno Tipo de Documento Número de Documento Celular Email Sueldo Acceso Login	Trabajador registrado con éxito	-Nombre en espacios -Apellido en espacios -Tipo de documento incorrecto -Número de documento inexistente -Falta dígitos	Debe completar todos los campos

	Password Estado		-Login inexistente -Password incorrecto	
Eliminar trabajador	Seleccionar al trabajador de la tabla	Trabajador eliminado con éxito	No selecciona al trabajador de la tabla	Debe seleccionar una fila
Modificar trabajador	Nombres Apellido Paterno Apellido Materno Tipo de Documento Número de Documento Celular Email Sueldo Acceso Login Password Estado	Trabajador actualizado con éxito	-Nombre en espacios -Apellido en espacios -Tipo de documento incorrecto -Número de documento inexistente -Falta dígitos -Login inexistente -Password incorrecto	Debe completar todos los campos
Consultar trabajador	Si existen trabajadores en la base de datos	Trabajador listado con éxito	No existen trabajadores en la base de datos	No hay trabajadores para listar
Validar trabajador	Usuario Contraseña	Trabajador validado con éxito	-Usuario inexistente -Contraseña incorrecta	Usuario o contraseña incorrecto
Registro Entrada Cliente	la de Seleccionar la entrada de cliente de la tabla	Entrada registrada con éxito	No selecciona la entrada de cliente de la tabla	Debe seleccionar una fila
Eliminar Entrada Cliente	la de Seleccionar la entrada de cliente de la tabla	Movimiento eliminado con éxito	No selecciona la entrada de cliente de la tabla	Debe seleccionar una fila
Modificar Entrada Cliente	la de Seleccionar la entrada de cliente de la tabla	Entrada actualizada con éxito	No selecciona la entrada de cliente de la tabla	Debe seleccionar una fila
Consultar Entrada Cliente	la de Si existen movimientos en la base de datos	Consulta realizada con éxito	No existen movimientos en la base de datos	No hay movimientos para listar
Registro Salida Cliente	de de Seleccionar la entrada de cliente de la tabla	Salida registrada con éxito	No selecciona la entrada de cliente de la tabla	Debe seleccionar una fila
Eliminar Salida Cliente	la de Seleccionar la entrada de cliente de la tabla	Movimiento eliminado con éxito	No selecciona la entrada de cliente de la tabla	Debe seleccionar una fila

Modificar Salida de Cliente	Seleccionar la entrada de cliente de la tabla	Salida actualizada con éxito	No selecciona la entrada de cliente de la tabla	Debe seleccionar una fila
Consultar Salida de Cliente	Si existen movimientos en la base de datos	Consulta realizada con éxito	No existen movimientos en la base de datos	No hay movimientos para listar

Tabla 9: Plan de pruebas Incremental - Versión 1

Caso de uso	Datos de Entrada normal	Resultado Esperado normal	Datos de entrada anómalo	Resultado esperado anómalo
Registro de reserva	Tipo de reserva Fecha de reserva Fecha de ingreso Fecha de salida Costo Estado	Reserva realizada con éxito	-Tipo de reserva incorrecta -Fecha de reserva inválida -Fecha de ingreso y salida inválida -Estado inválido	Debe completar todos los campos.
Modificar reserva	Tipo de reserva Fecha de reserva Fecha de ingreso Fecha de salida Costo Estado	Reserva modificada con éxito	-Tipo de reserva incorrecta -Fecha de reserva inválida -Fecha de ingreso y salida inválida -Estado inválido	Debe completar todos los campos.
Eliminar reserva	Seleccionar reserva de la tabla	Reserva eliminada con éxito	No selecciona reserva de la tabla	Debe seleccionar una fila
Consultar reserva	Si existen reservas en la base de datos	Consulta realizada con éxito	No existen reservas en la base de datos	No hay reservas para listar
Registro de habitación	Número Piso Descripción Características Precio diario Estado Tipo de habitación	Habitación registrada con éxito	-Número de habitación inválida -Piso inválido -Exceso de límites de caracteres -Estado inválido -Tipo de habitación inválida	Debe completar todos los campos.
Modificar habitación	Número Piso	Habitación modificada con éxito	-Número de habitación inválida -Piso inválido	Debe completar

	Descripción Características Precio diario Estado Tipo de habitación		-Exceso de límites de caracteres -Estado inválido -Tipo de habitación inválida	todos los campos.
Eliminar habitación	Seleccionar habitación de la tabla	Habitación eliminada con éxito	No selecciona habitación de la tabla	Debe seleccionar una fila
Consultar habitación	Si existen habitaciones en la base de datos	Consulta realizada con éxito	No existen habitaciones en la base de datos	No hay habitaciones para listar

Tabla 10: Plan de pruebas Incremental - Versión 2

## Ejecución del Plan de Pruebas:

### Caso de uso Registro de Cliente:

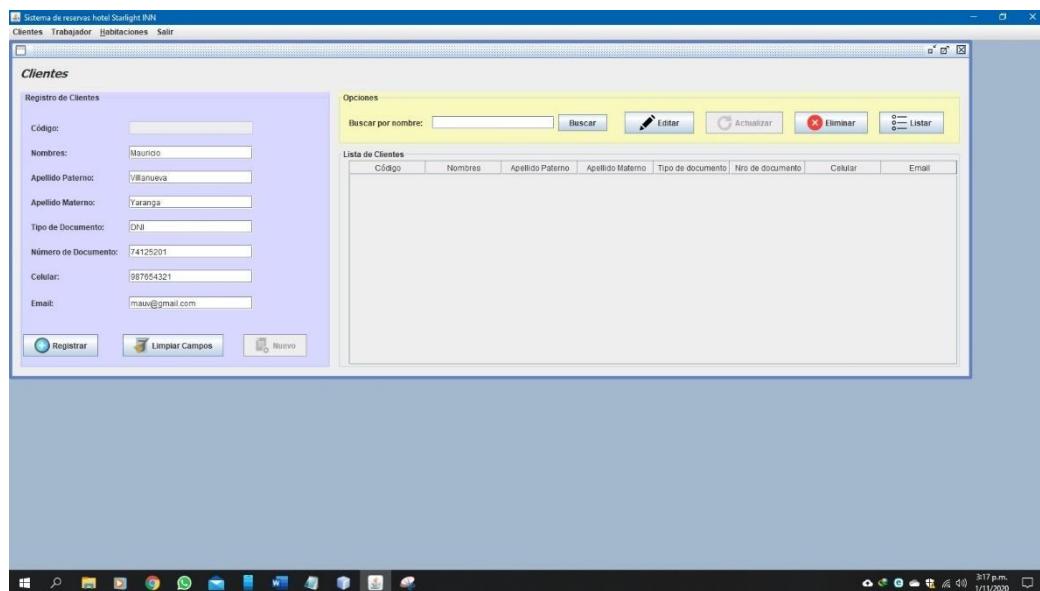


Ilustración 60: Registro de Cliente - Datos de entrada normal - Versión 1

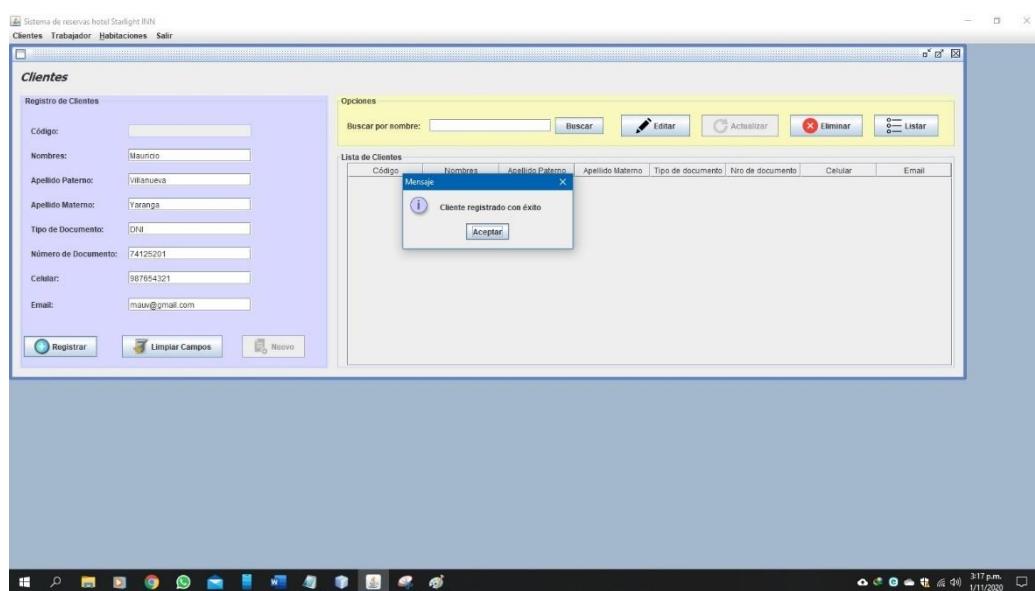


Ilustración 61: Registro de Cliente - Resultado esperado normal - Versión 1

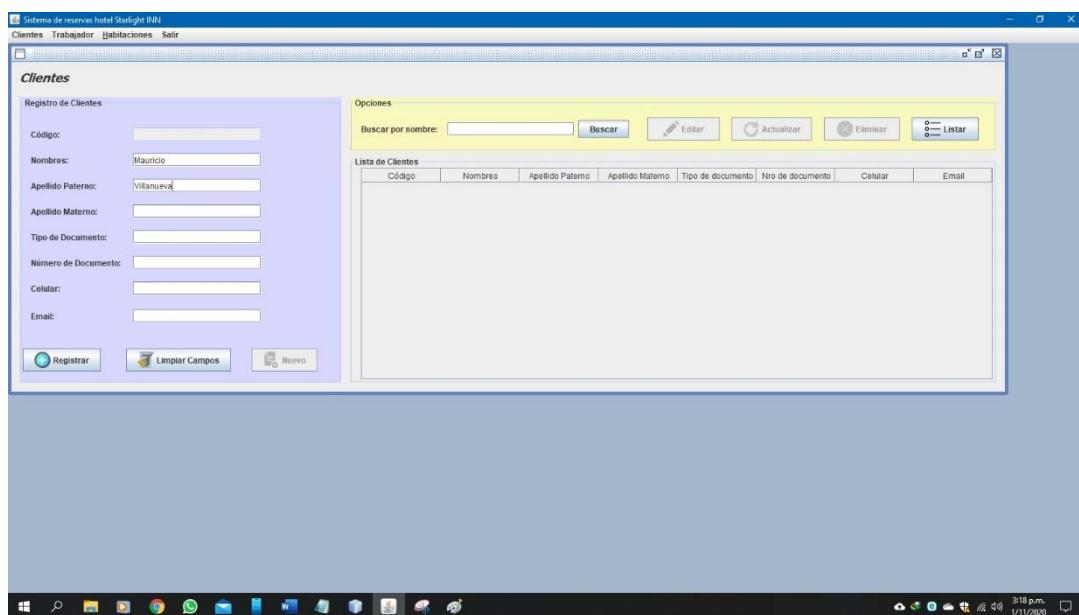


Ilustración 62: Registro de Cliente - Datos de entrada anómalo - Versión 1

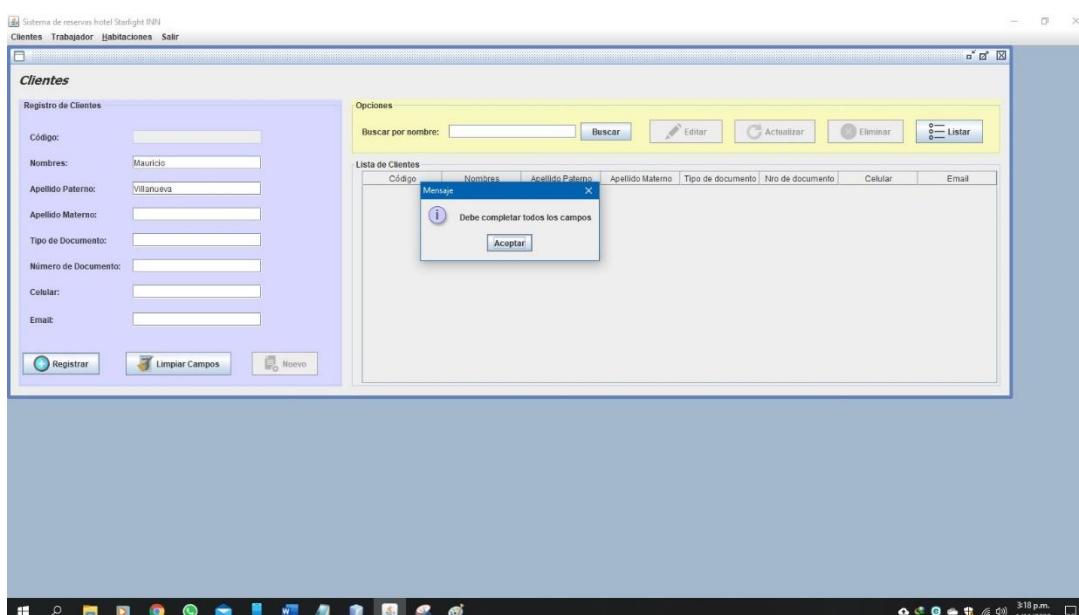


Ilustración 63: Registro de Cliente - Resultado esperado anómalo - Versión 1

## Caso de uso Eliminar Cliente:

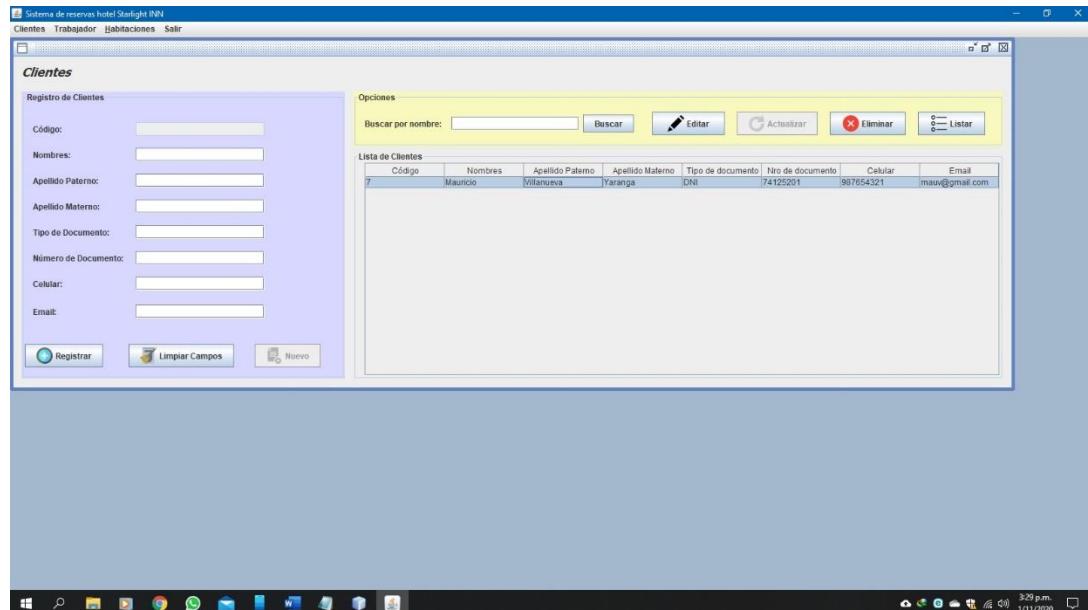


Ilustración 64: Eliminar Cliente - Datos de entrada normal - Versión 1

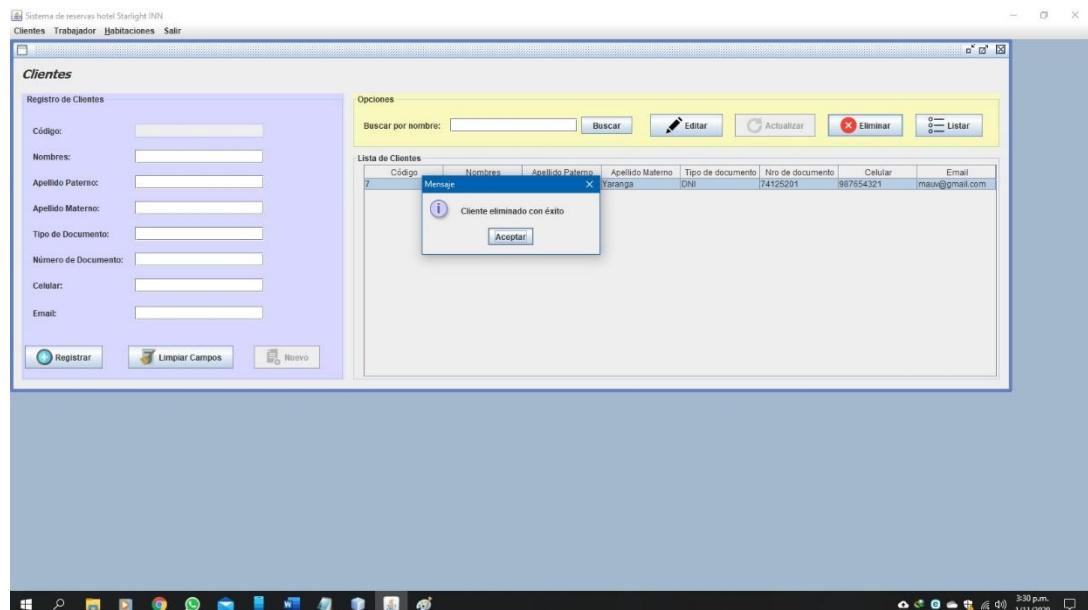


Ilustración 65: Eliminar Cliente - Resultado esperado normal - Versión 1

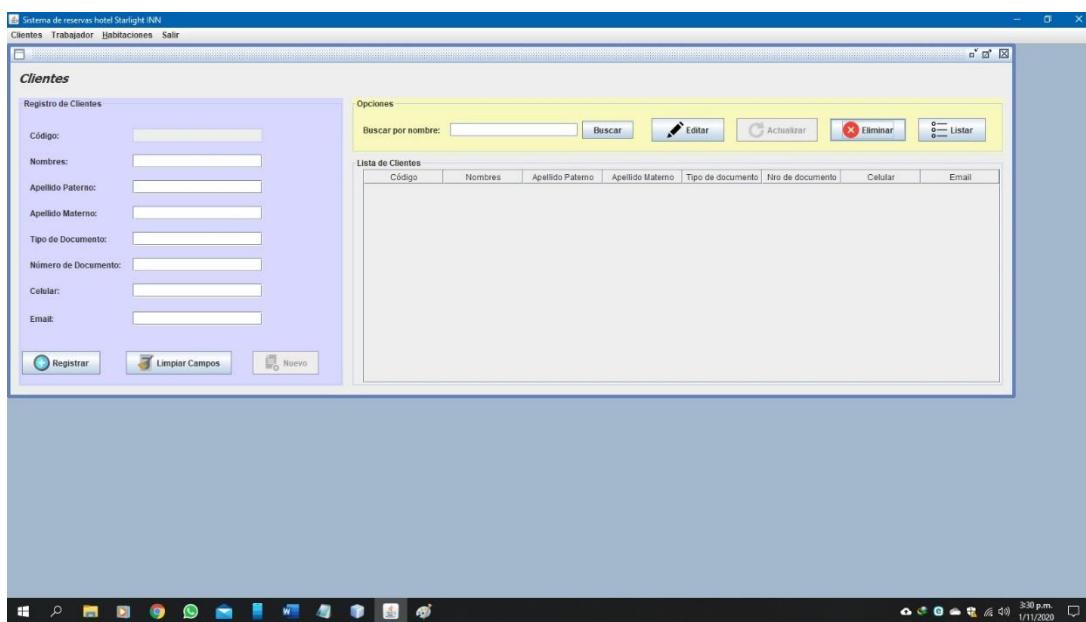


Ilustración 66: Eliminar Cliente - Datos de entrada anómalo - Versión 1

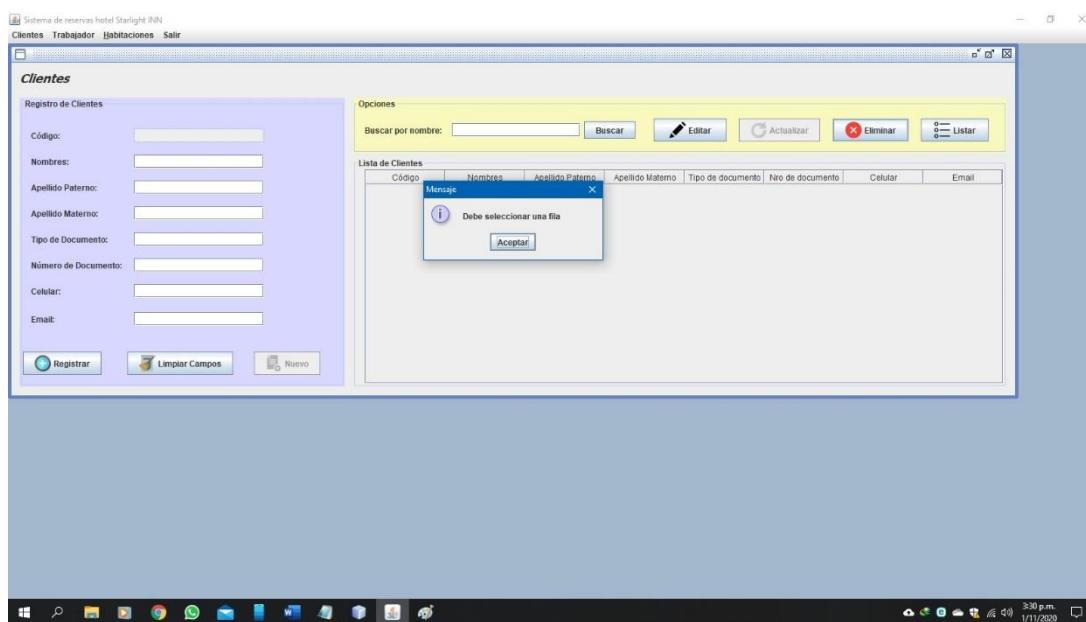


Ilustración 67: Eliminar Cliente - Resultado esperado anómalo - Versión 1

## Caso de uso Modificar Cliente:

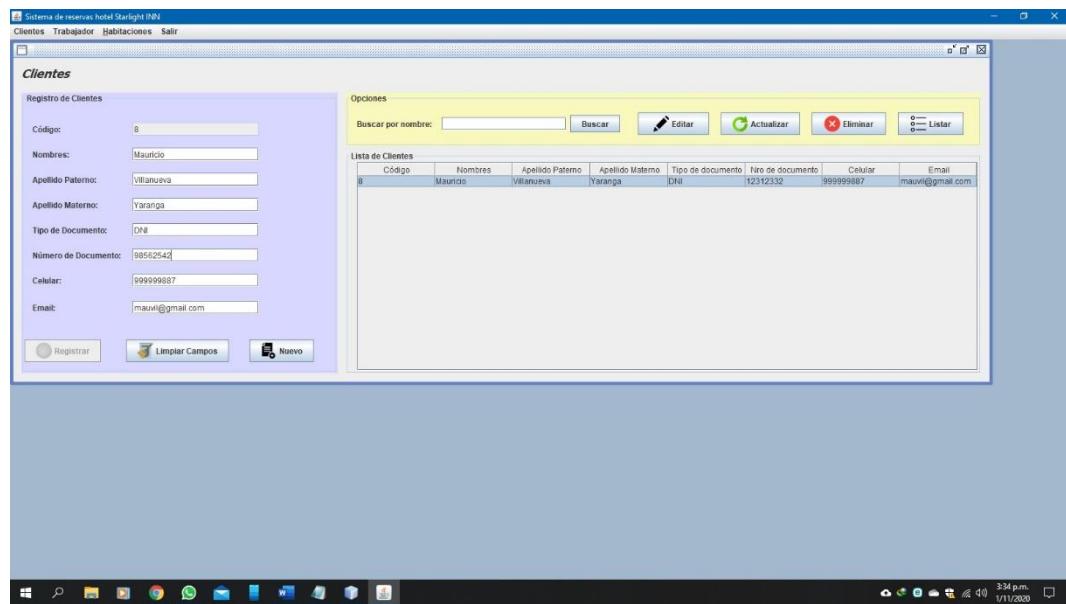


Ilustración 68: Modificar Cliente - Datos de entrada normal - Versión 1

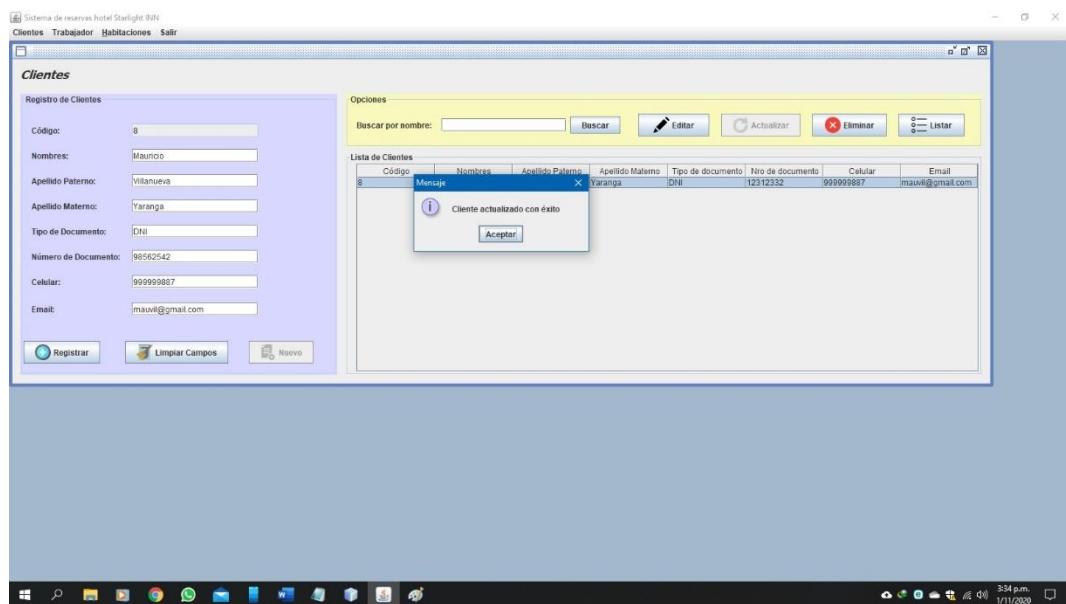


Ilustración 69: Modificar Cliente - Resultado esperado normal - Versión 1

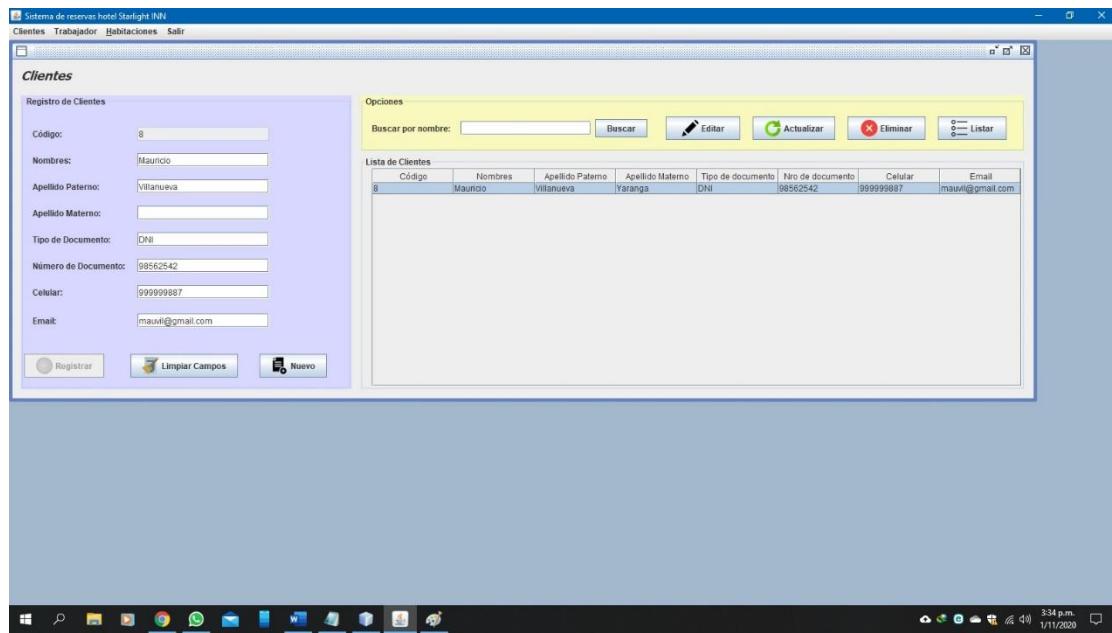


Ilustración 70: Modificar Cliente - Datos de entrada anómalo - Versión 1

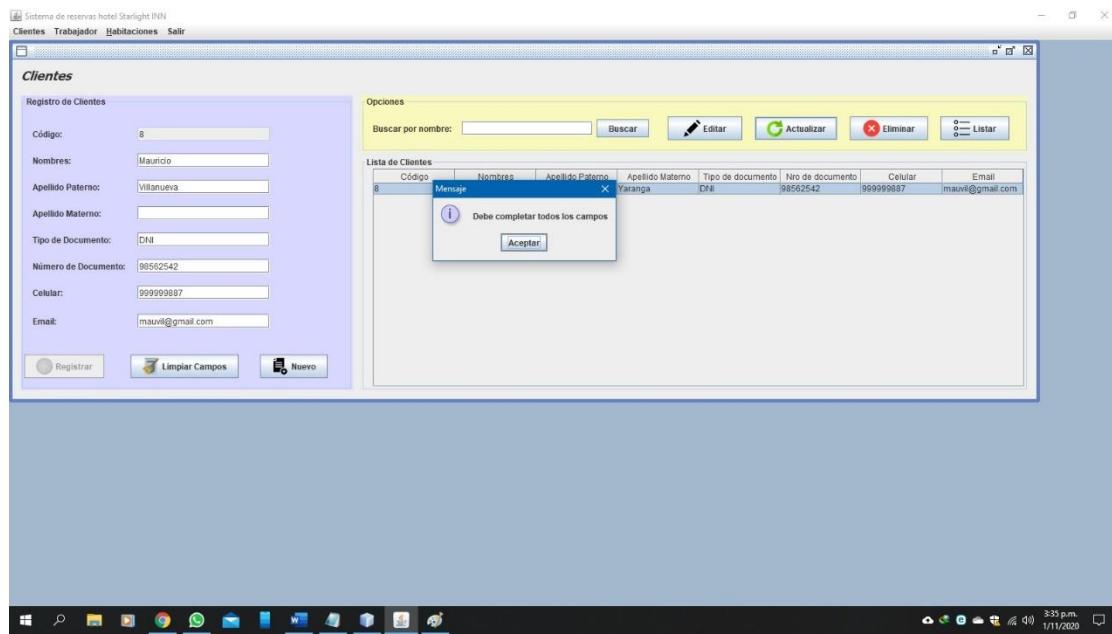


Ilustración 71: Modificar Cliente - Resultado esperado anómalo - Versión 1

## Caso de uso Consultar Cliente:

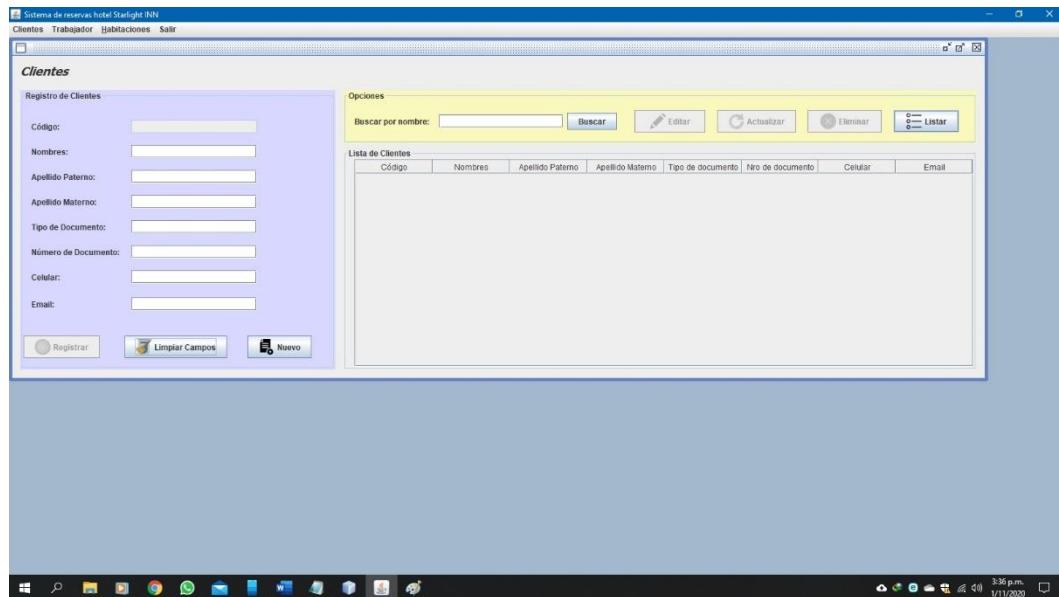


Ilustración 72: Consultar Cliente - Datos de entrada normal - Versión 1

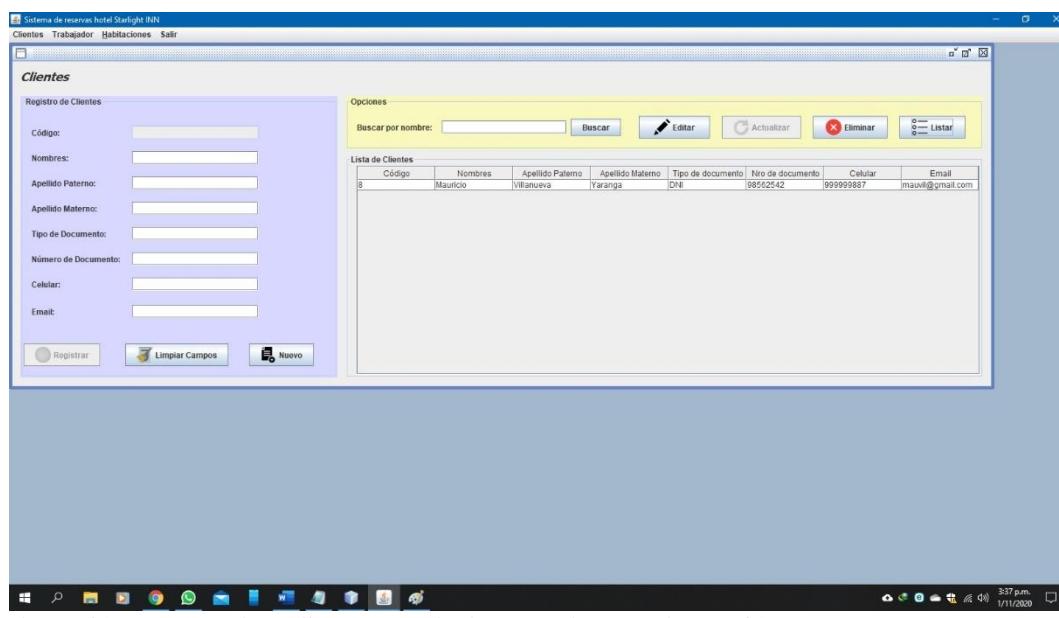


Ilustración 73: Consultar Cliente - Resultado esperado normal - Versión 1

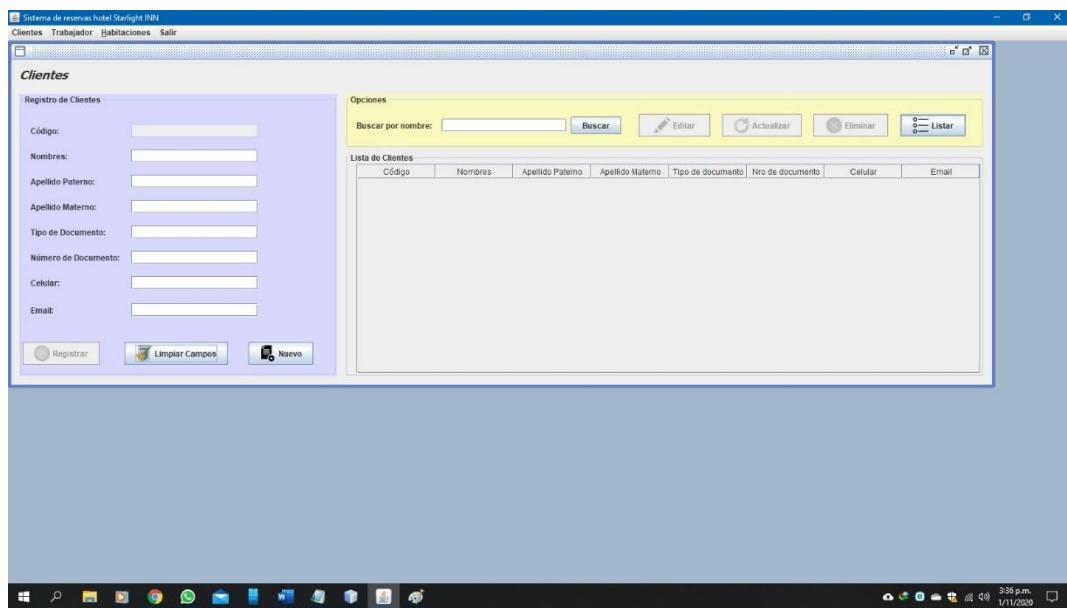


Ilustración 74: Consultar Cliente - Datos de entrada anómalo - Versión 1

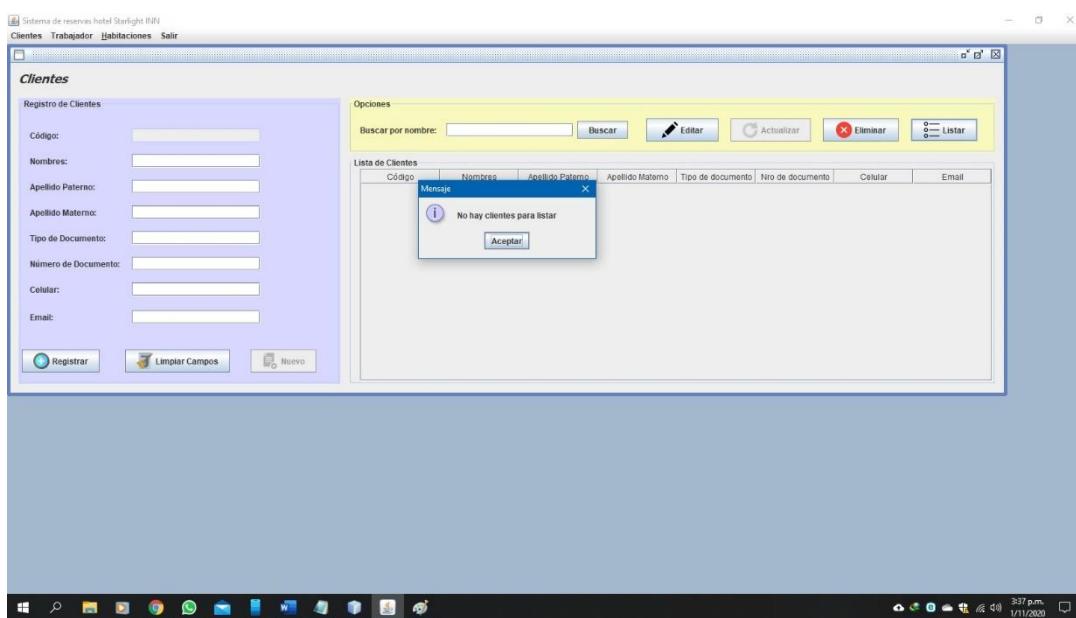


Ilustración 75: Consultar Cliente - Resultado esperado anómalo - Versión 1

## Caso de uso Registrar Trabajador:

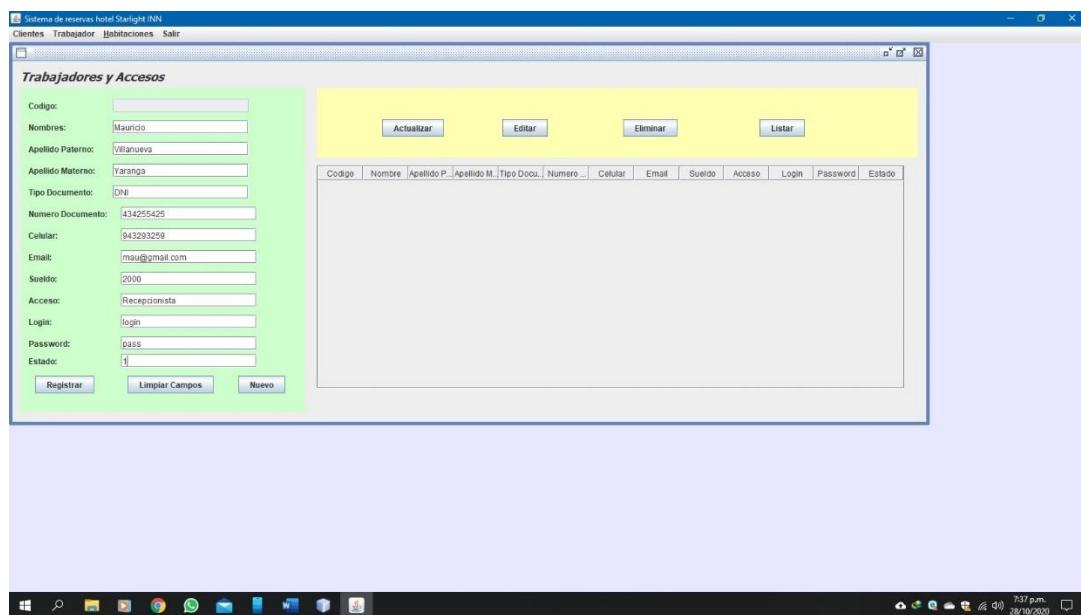


Ilustración 76: Registrar Trabajador - Datos de entrada normal - Versión 1

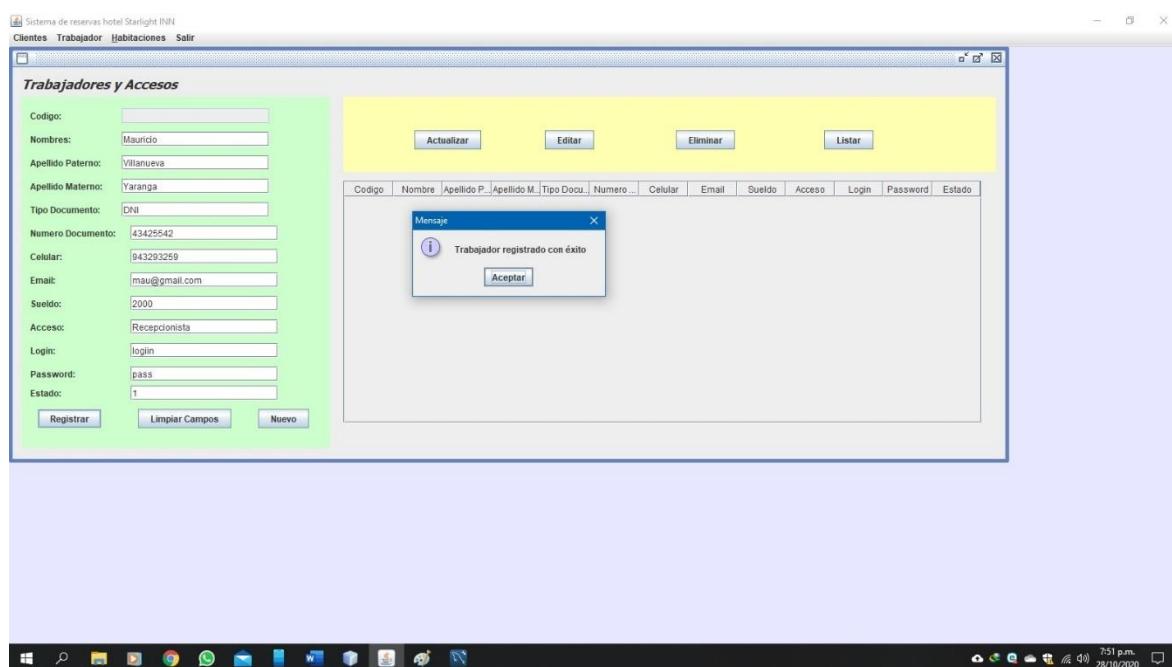


Ilustración 77: Registrar Trabajador - Resultado esperado normal - Versión 1

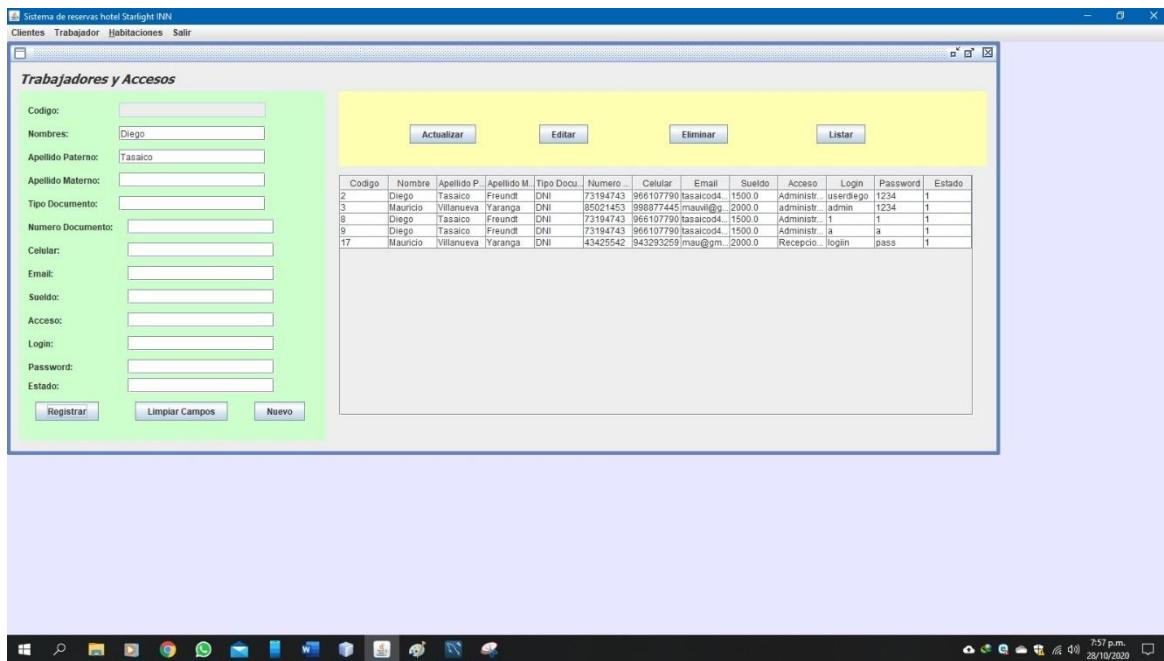


Ilustración 78: Registrar Trabajador - Datos de entrada anómalo - Versión 1

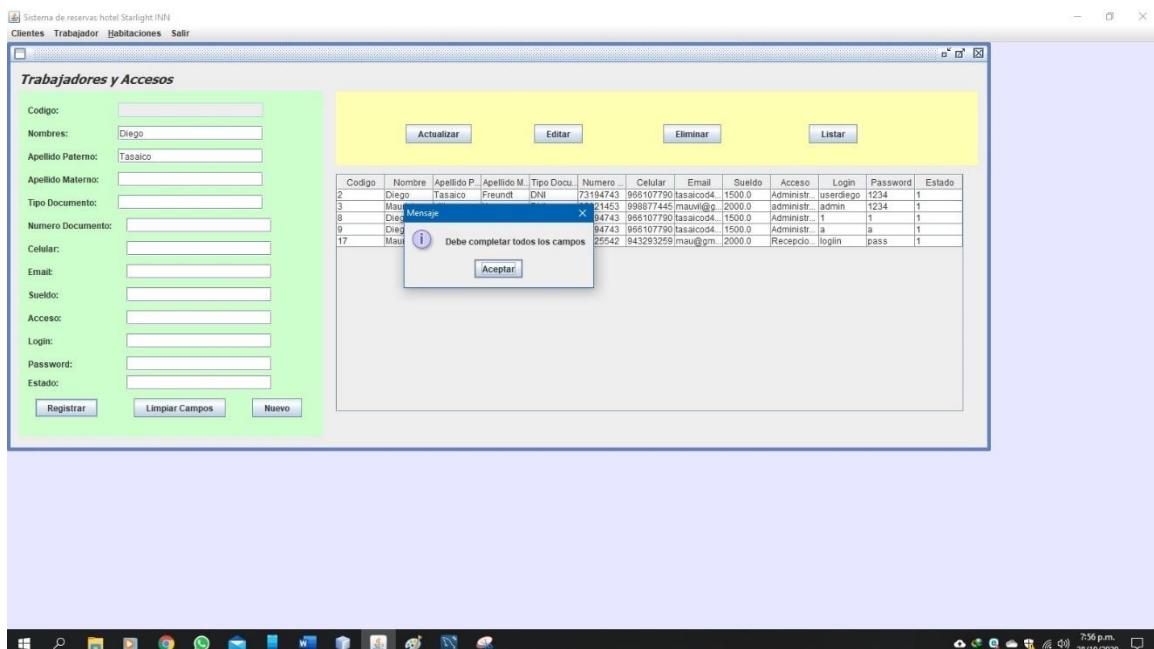


Ilustración 79: Registrar Trabajador - Resultado esperado anómalo - Versión 1

## Caso de uso Eliminar Trabajador:

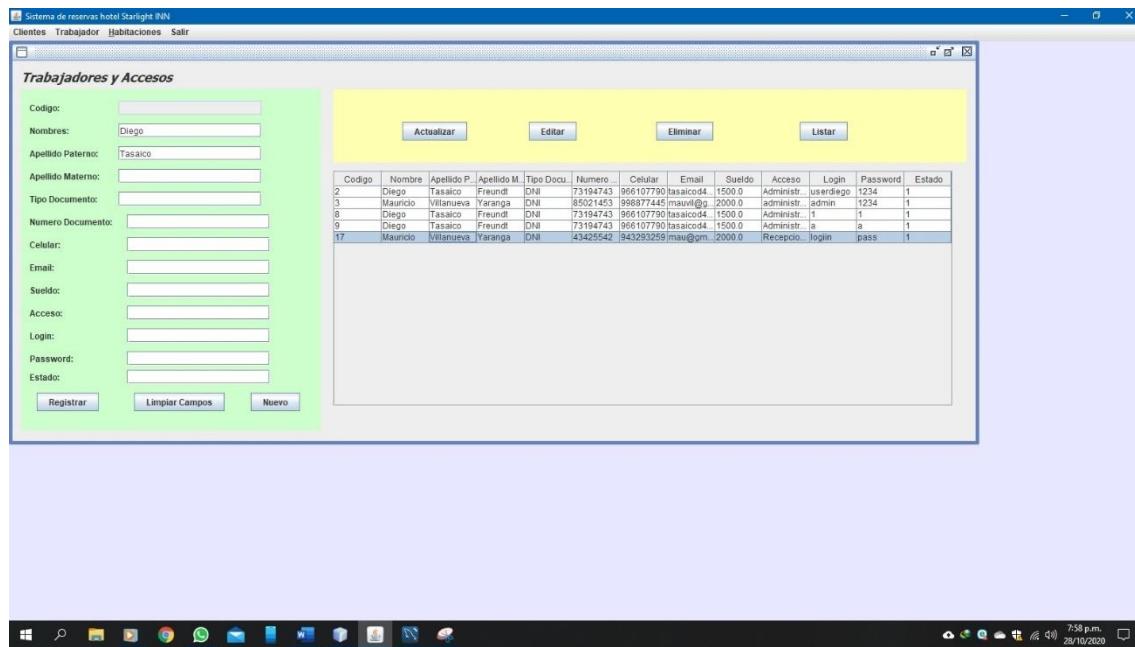


Ilustración 80: Eliminar Trabajador - Datos de entrada normal - Versión 1

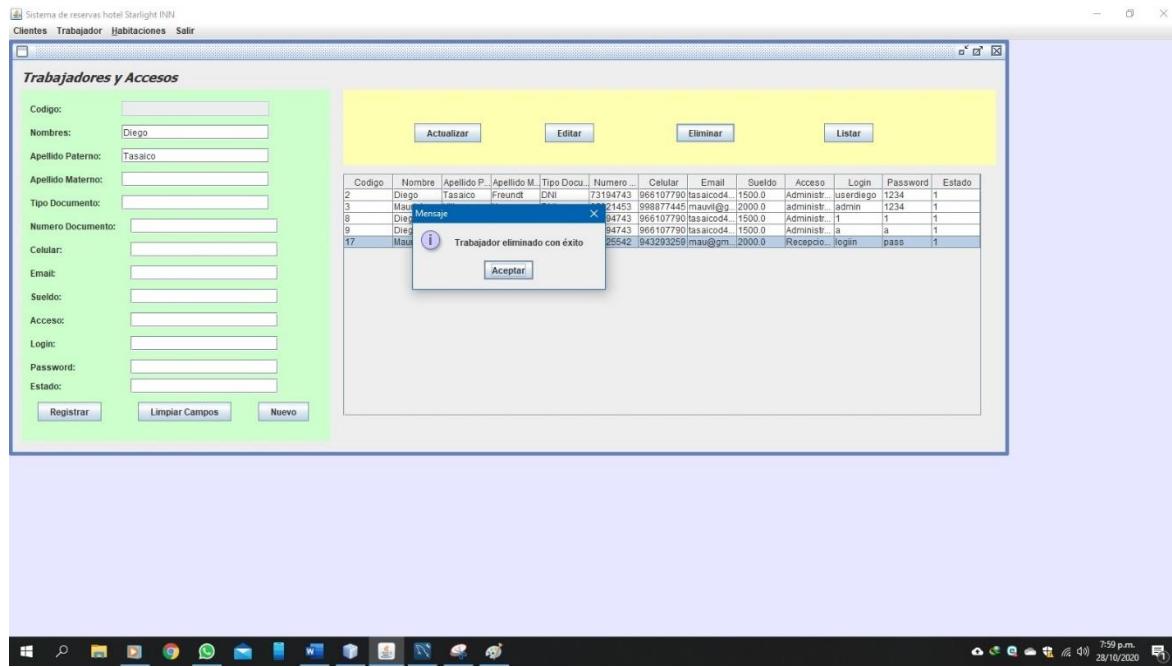


Ilustración 81: Eliminar Trabajador - Resultado esperado normal - Versión 1

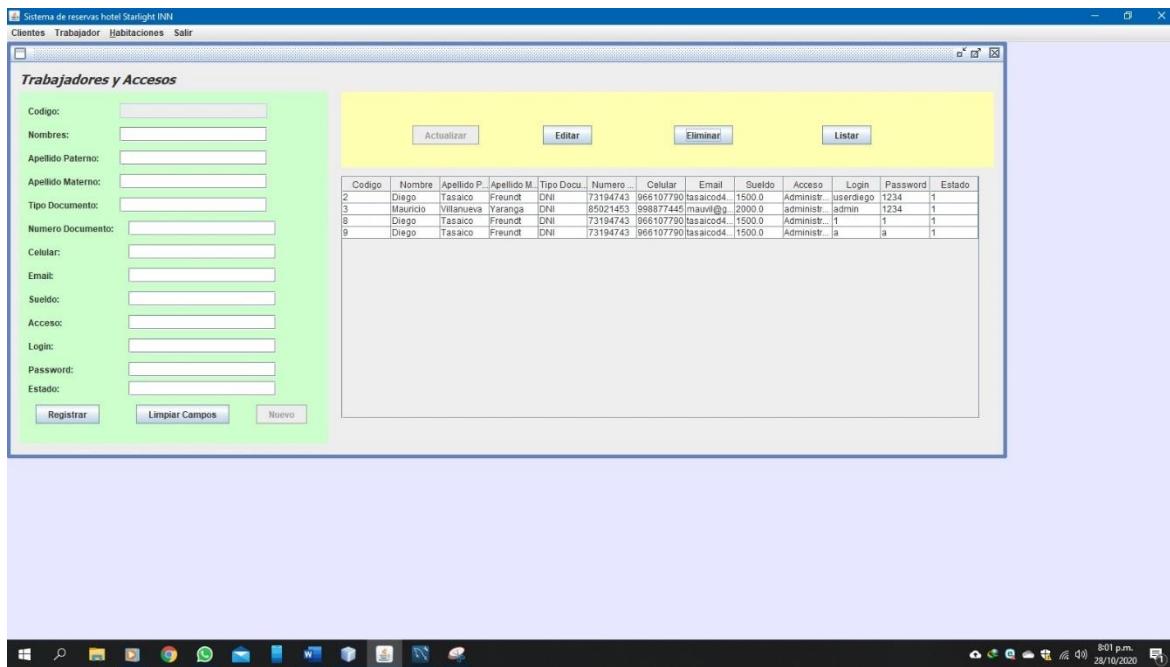


Ilustración 82: Eliminar Trabajador - Datos de entrada anómalo - Versión 1

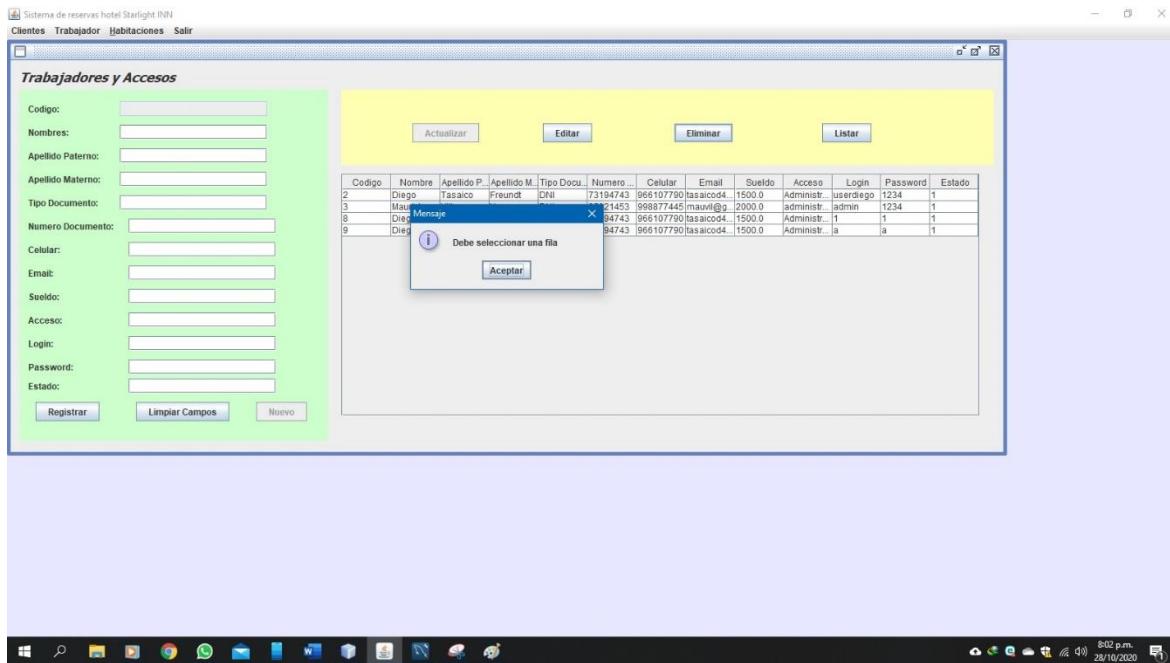


Ilustración 83: Eliminar Trabajador - Resultado esperado anómalo - Versión 1

## Caso de uso Modificar Trabajador:

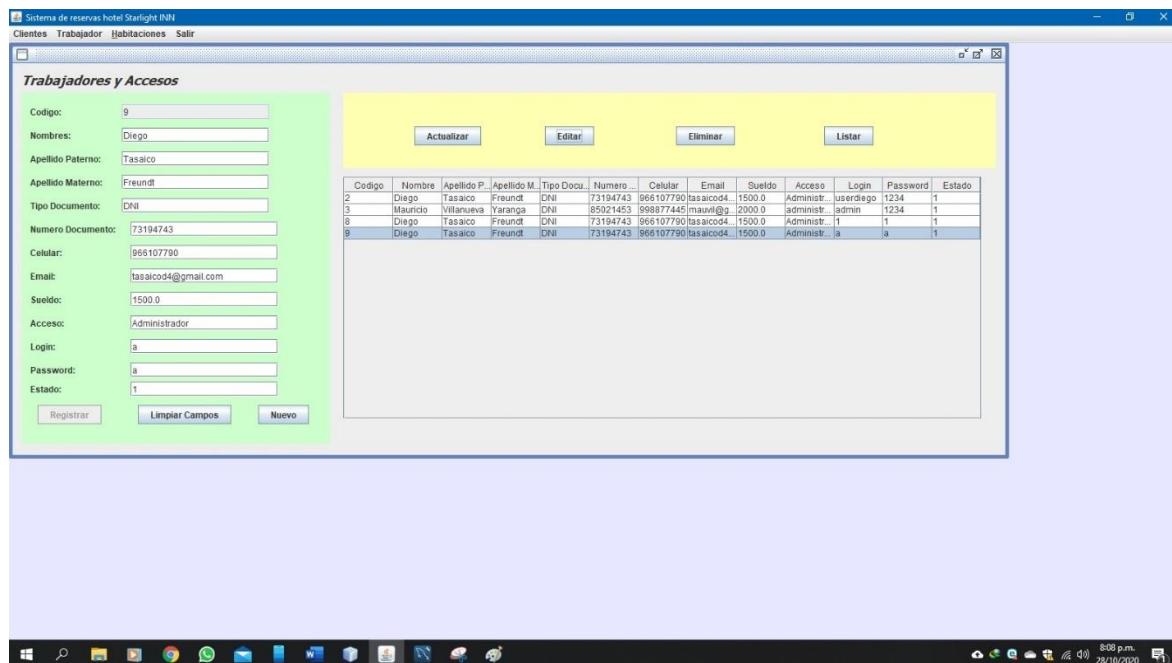


Ilustración 84: Modificar Trabajador - Datos de entrada normal - Versión 1

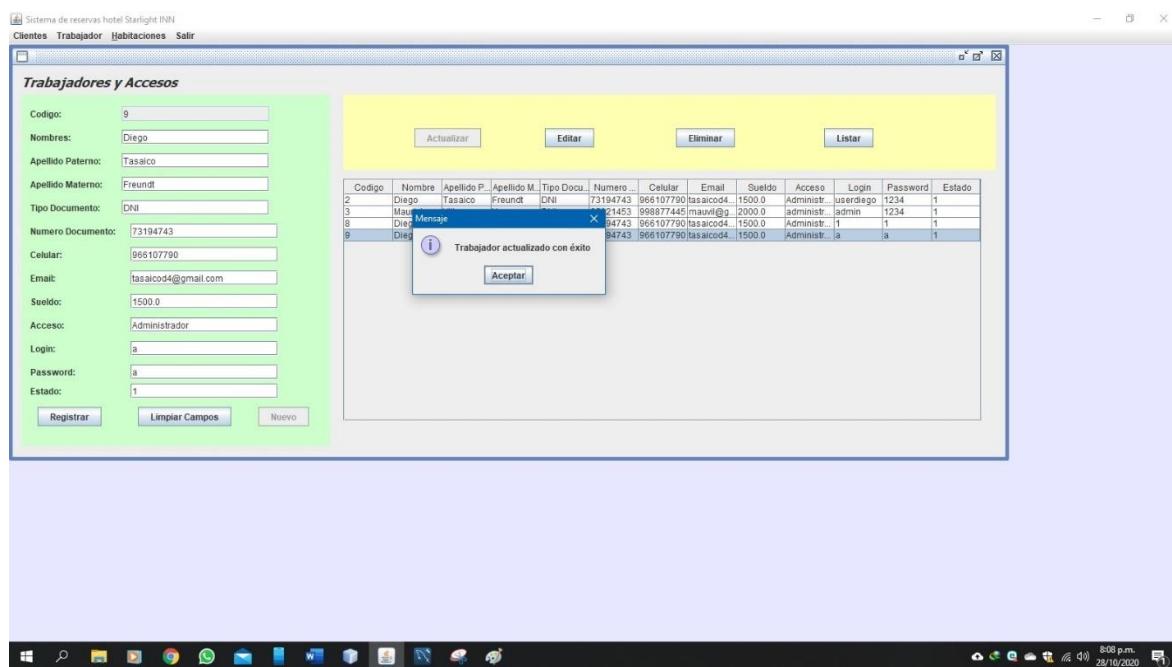


Ilustración 85: Modificar Trabajador - Resultado esperado normal - Versión 1

Sistema de reservas hotel Starlight INN

Ciudad Trabajador Habitaciones Salir

### Trabajadores y Accesos

Código:	9
Nombre:	Diego
Apellido Paterno:	Tasaico
Apellido Materno:	
Tipo Documento:	DNI
Número Documento:	73194743
Celular:	966107790
Email:	tasaico4@gmail.com
Sueldo:	1500.0
Acceso:	Administrador
Login:	a
Password:	a
Estado:	1

Código	Nombre	Apellido P.	Apellido M.	Tipo Docu.	Número	Celular	Email	Sueldo	Acceso	Login	Password	Estado
2	Diego	Tasaico	Freundt	DNI	73194743	966107790	tasaicod4	1500.0	Administrador	userdiego	1234	1
3	Mauricio	Villanueva	Frangia	DNI	96621453	998877445	maxv@zg	2000.0	Administrador	admin	1234	1
8	Diego	Tasaico	Freundt	DNI	73194743	966107790	tasaicod4	1500.0	Administrador	1	1	1
9	Diego	Tasaico	Freundt	DNI	73194743	966107790	tasaicod4	1500.0	Administrador	a	a	1

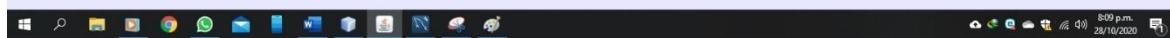


Ilustración 86: Modificar Trabajador - Datos de entrada anómalo - Versión 1

Sistema de reservas hotel Starlight INN

Ciudad Trabajador Habitaciones Salir

### Trabajadores y Accesos

Código:	9
Nombre:	Diego
Apellido Paterno:	Tasaico
Apellido Materno:	
Tipo Documento:	DNI
Número Documento:	73194743
Celular:	966107790
Email:	tasaico4@gmail.com
Sueldo:	1500.0
Acceso:	Administrador
Login:	a
Password:	a
Estado:	1

Código	Nombre	Apellido P.	Apellido M.	Tipo Docu.	Número	Celular	Email	Sueldo	Acceso	Login	Password	Estado
2	Diego	Tasaico	Freundt	DNI	73194743	966107790	tasaicod4	1500.0	Administrador	userdiego	1234	1
3	Mauricio	Villanueva	Frangia	DNI	96621453	998877445	maxv@zg	2000.0	Administrador	admin	1234	1
8	Diego	Tasaico	Freundt	DNI	73194743	966107790	tasaicod4	1500.0	Administrador	1	1	1
9	Diego	Tasaico	Freundt	DNI	73194743	966107790	tasaicod4	1500.0	Administrador	a	a	1

**Mensaje**

Debe completar todos los campos

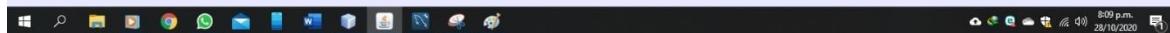


Ilustración 87: Modificar Trabajador - Resultado esperado anómalo - Versión 1

## Caso de uso Consultar Trabajador:

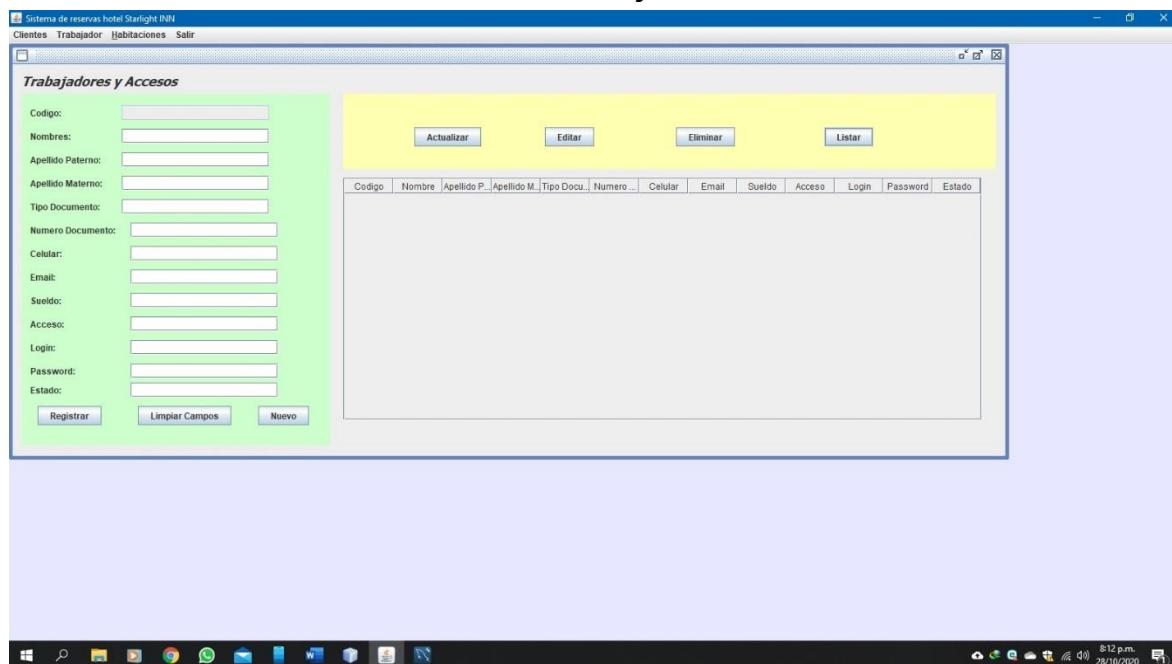


Ilustración 88: Consultar Trabajador - Datos de entrada normal - Versión 1

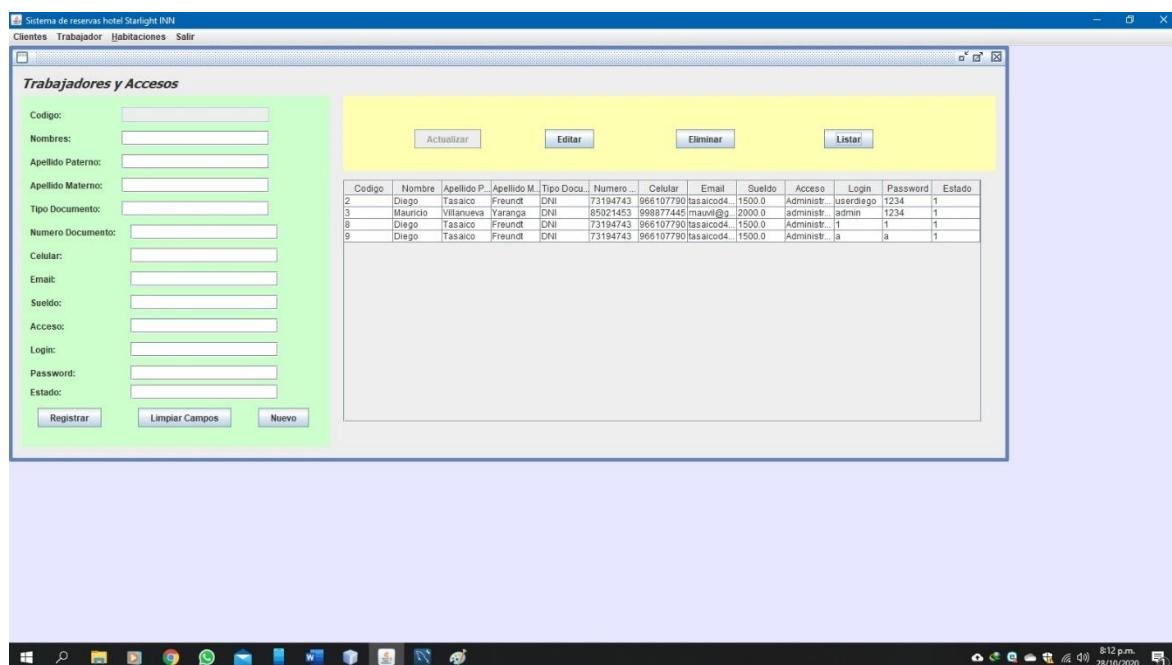


Ilustración 89: Consultar Trabajador - Resultado esperado normal - Versión 1

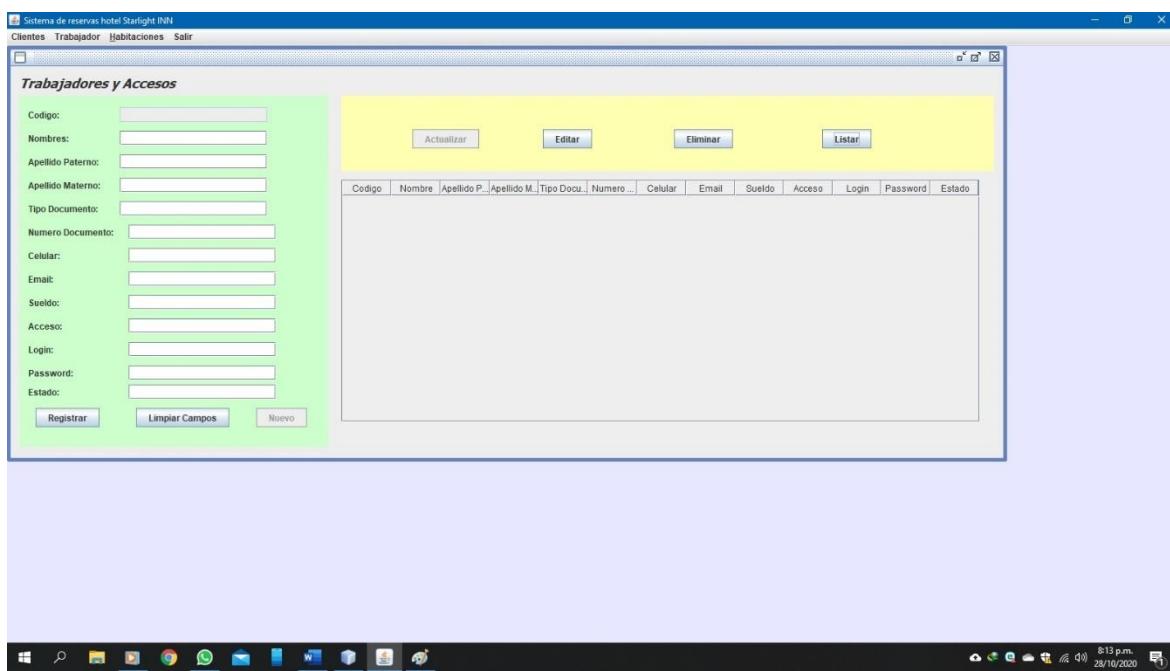


Ilustración 90: Consultar Trabajador - Datos de entrada anómalo - Versión 1

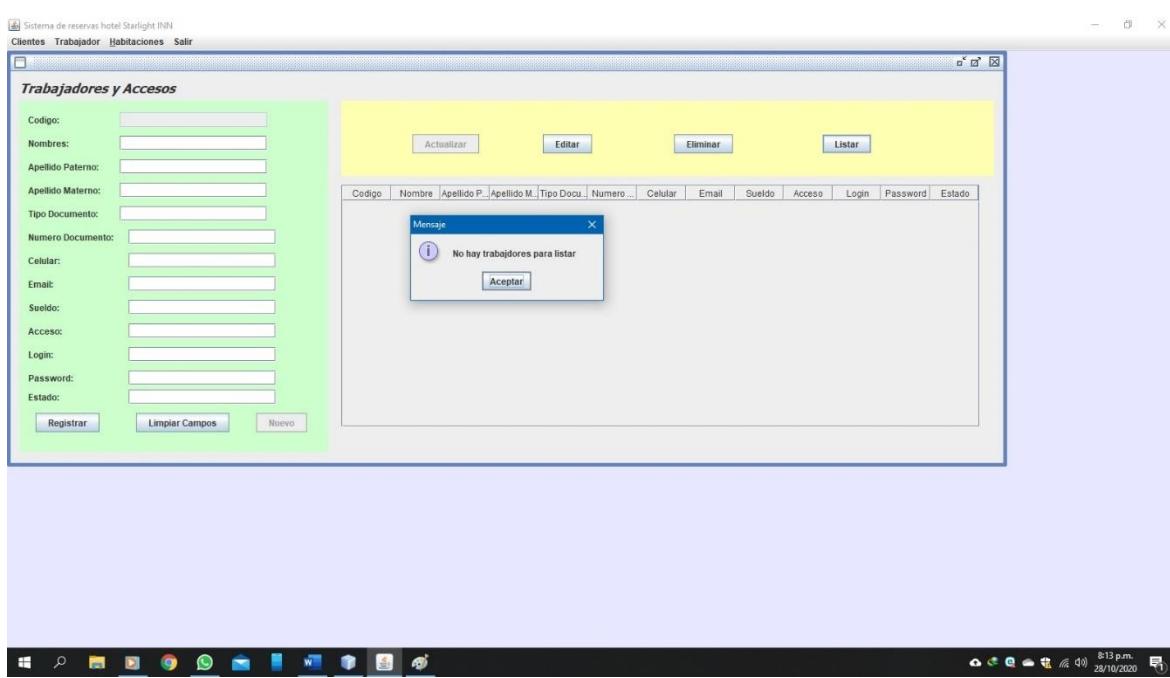


Ilustración 91: Consultar Trabajador - Resultado esperado anómalo - Versión 1

## Caso de uso Validar Trabajador:

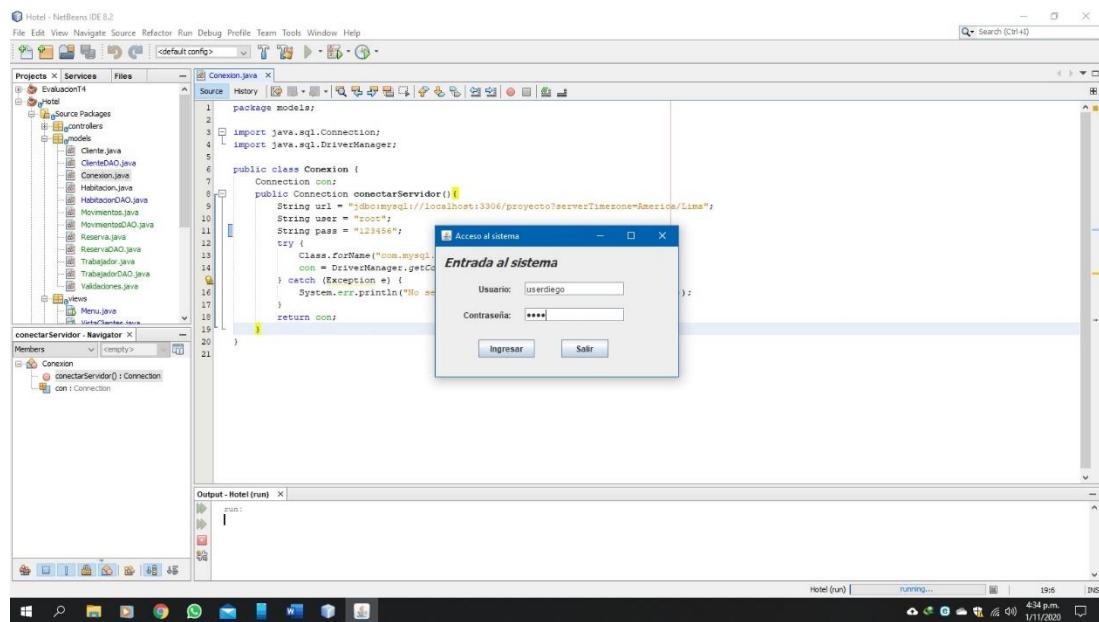


Ilustración 92: Validar Trabajador - Datos de entrada normal - Versión 1

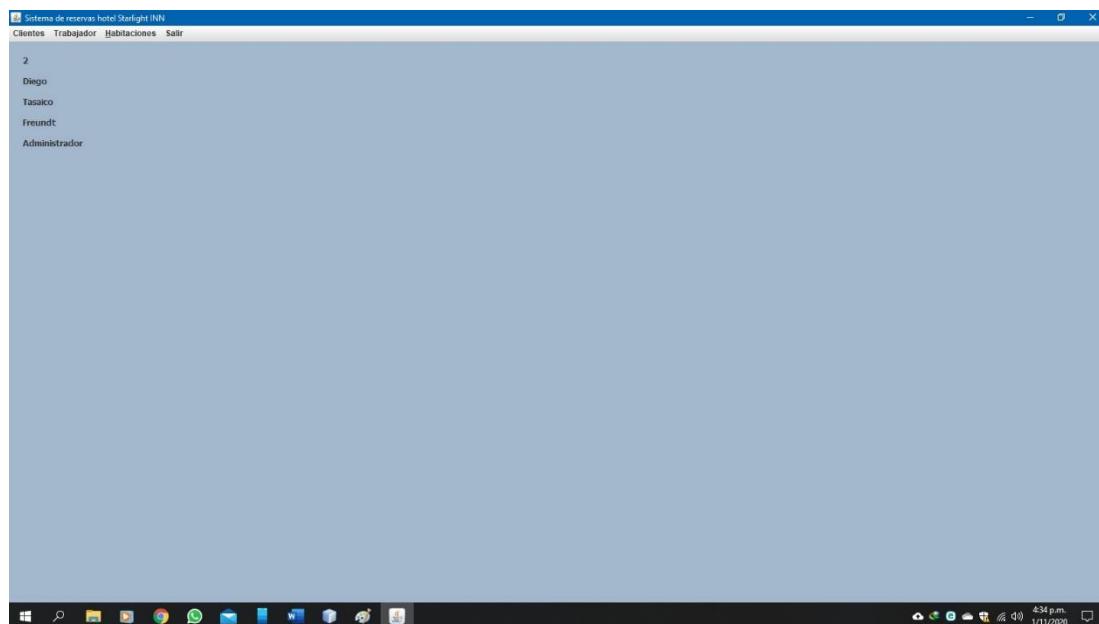


Ilustración 93: Validar Trabajador - Resultado esperado normal - Versión 1

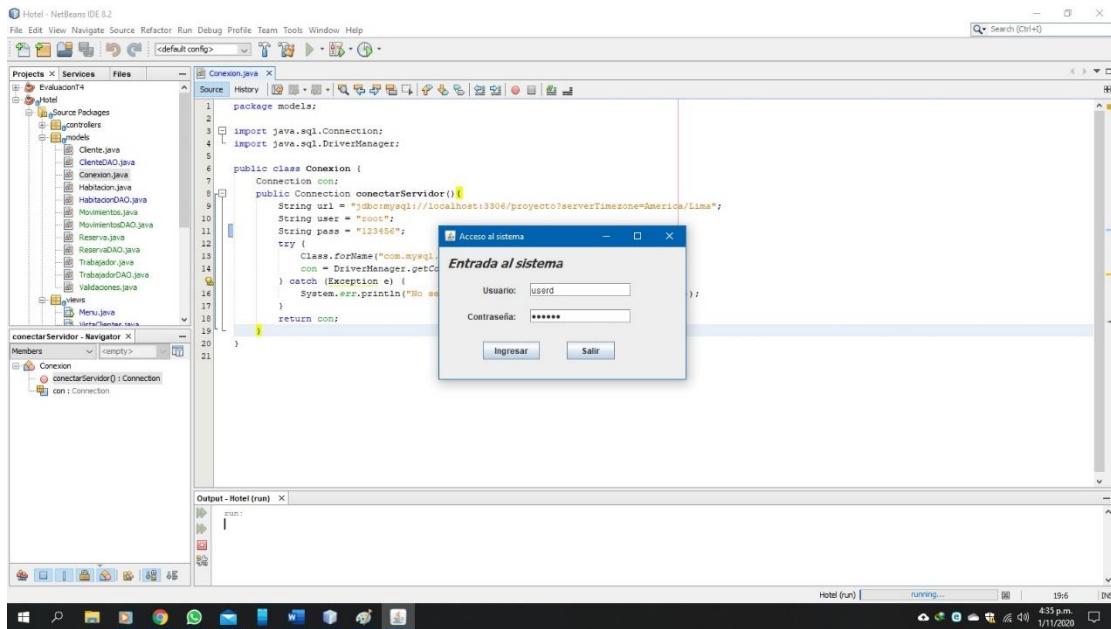


Ilustración 94: Validar Trabajador - Datos de entrada anómalo - Versión 1

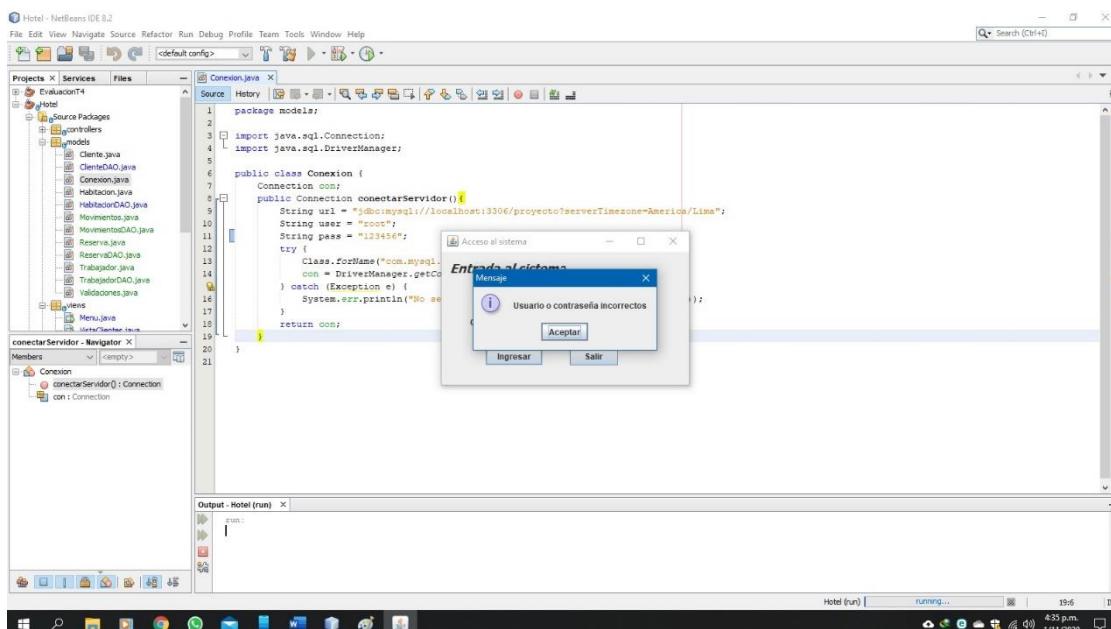


Ilustración 95: Validar Trabajador - Resultado esperado anómalo - Versión 1

## Caso de uso Registro de Entrada Cliente:

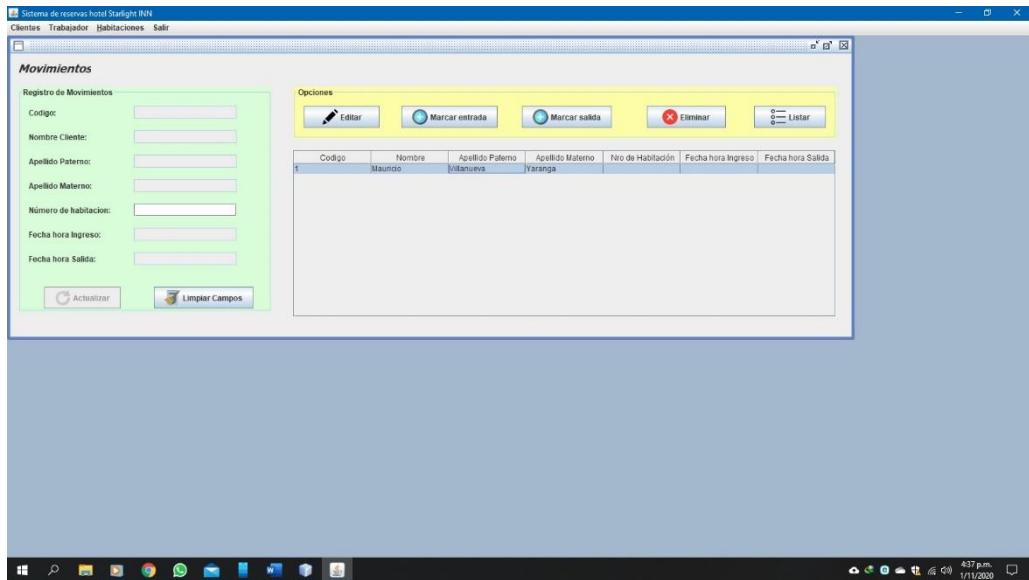


Ilustración 96: Registro de entrada de cliente - Datos de entrada normal - Versión 1

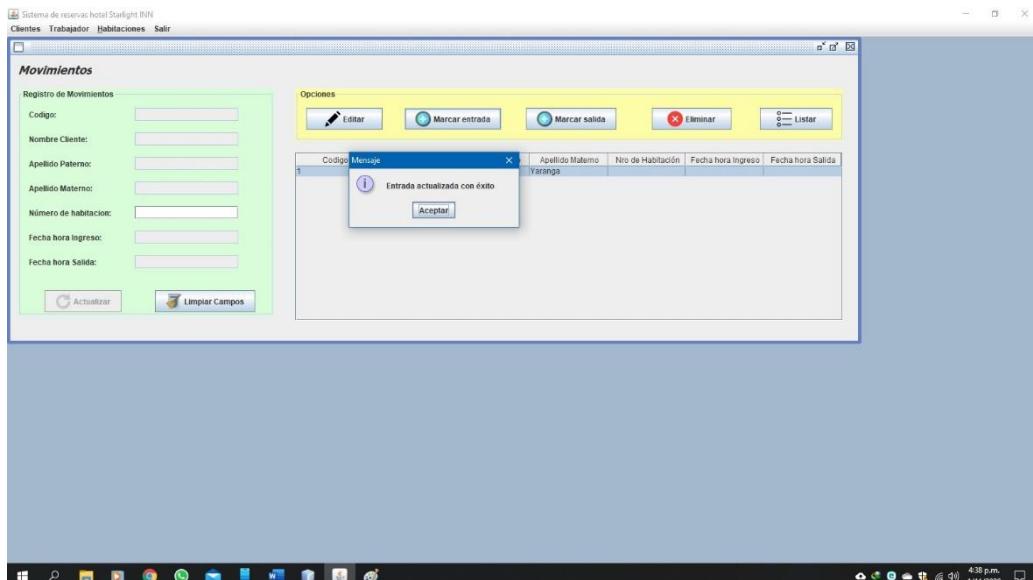


Ilustración 97: Registro de entrada de cliente - Resultado esperado normal - Versión 1

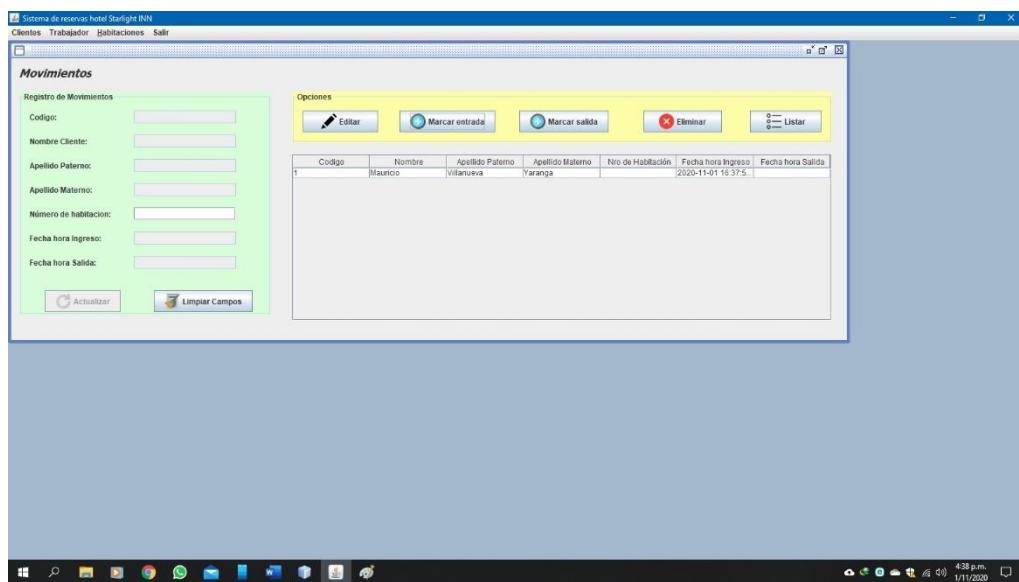


Ilustración 98: Registro de entrada de cliente - Datos de entrada anómalo - Versión 1

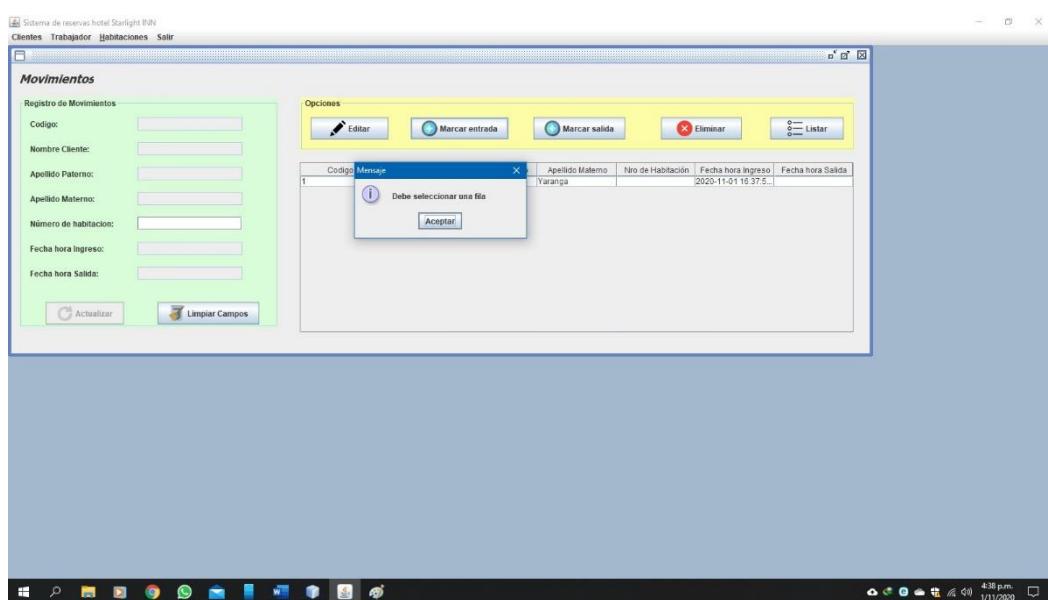


Ilustración 99: Registro de entrada de cliente - Resultado esperado anómalo - Versión 1

## Caso de uso Eliminar la Entrada Cliente:

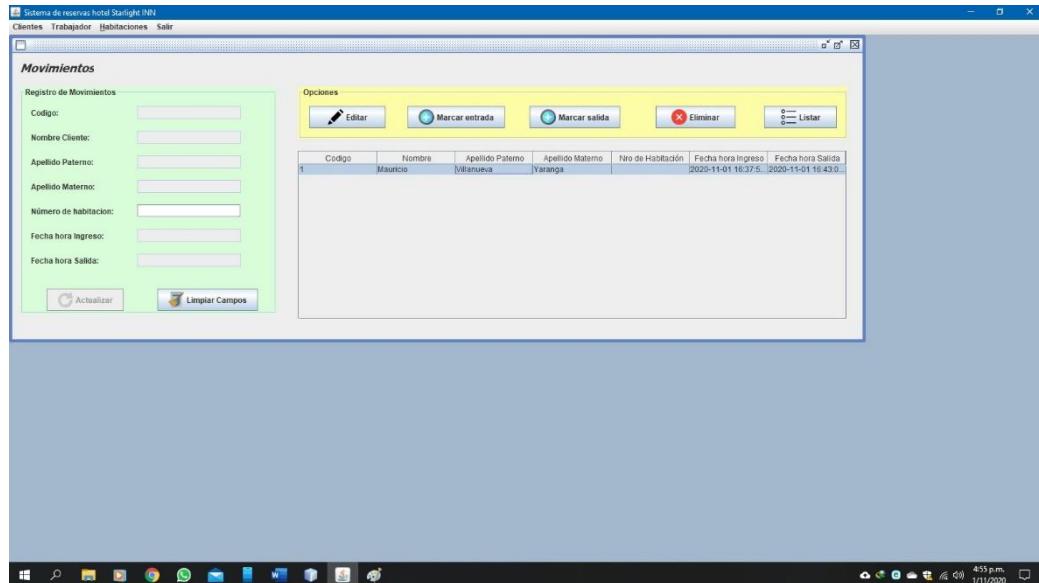


Ilustración 100: Eliminar la entrada de cliente - Datos de entrada normal - Versión 1

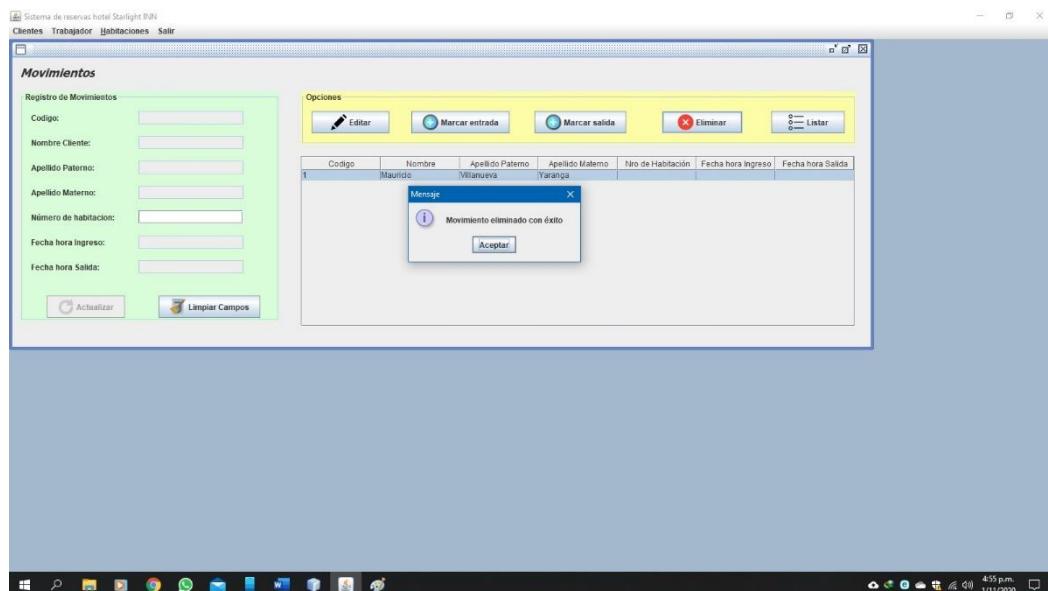


Ilustración 101: Eliminar la entrada de cliente - Resultado esperado normal - Versión 1

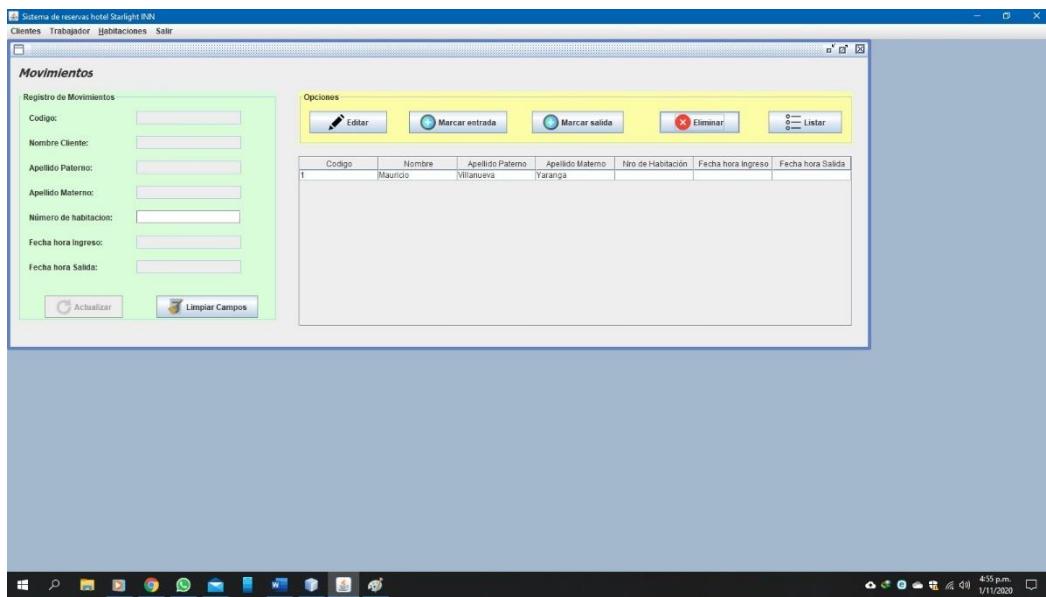


Ilustración 102: Eliminar la entrada de cliente - Datos de entrada anómalo - Versión 1

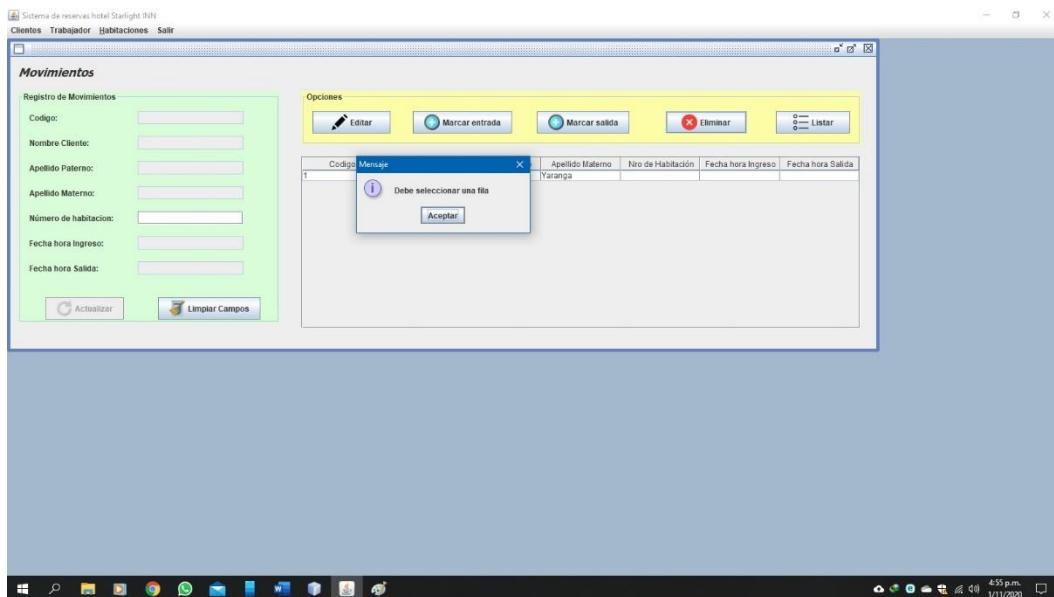


Ilustración 103: Eliminar la entrada de cliente - Resultado esperado anómalo - Versión 1

## Caso de uso Modificar la Entrada Cliente:

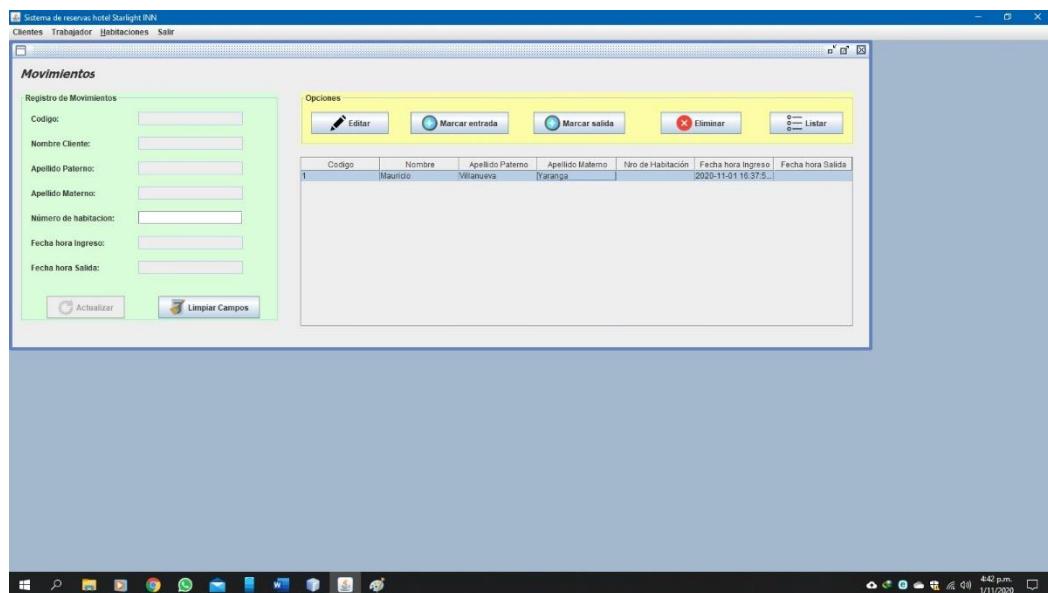


Ilustración 104: Modificar la entrada de cliente - Datos de entrada normal - Versión 1

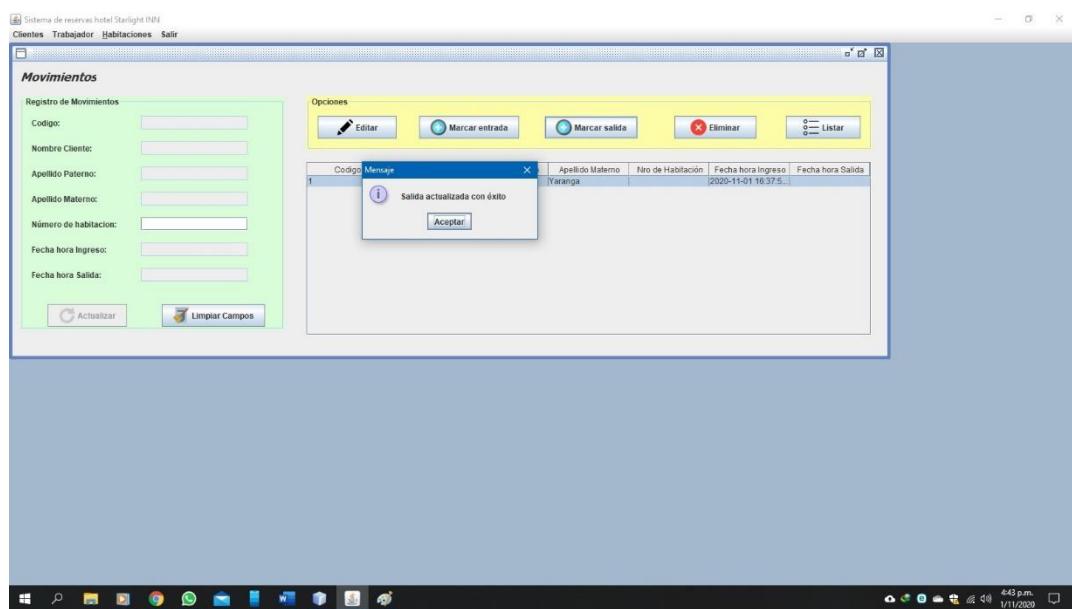


Ilustración 105: Modificar la entrada de cliente - Resultado esperado normal - Versión 1

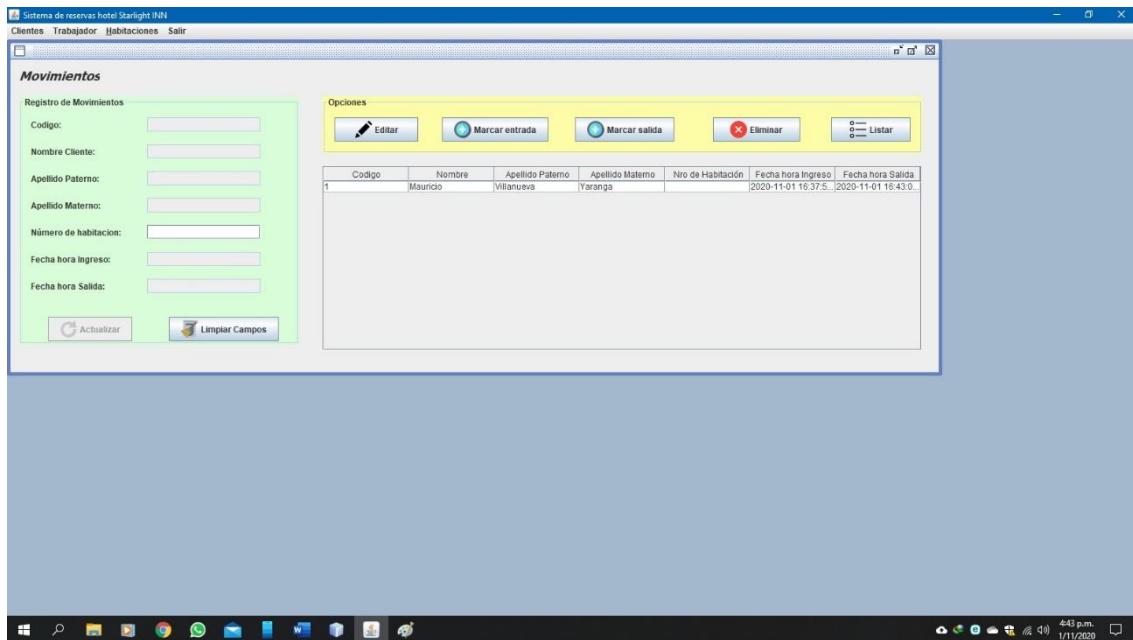


Ilustración 106: Modificar la entrada de cliente - Datos de entrada anómalo - Versión 1

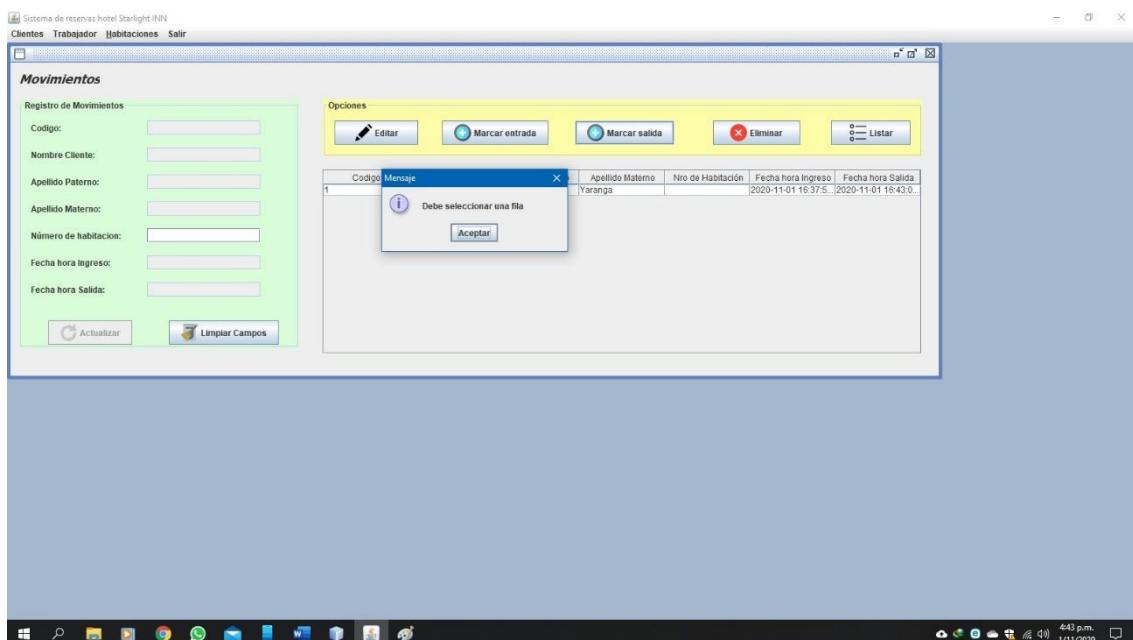


Ilustración 107: Modificar la entrada de cliente - Resultado esperado anómalo - Versión 1

## Caso de uso Consultar la Entrada Cliente:

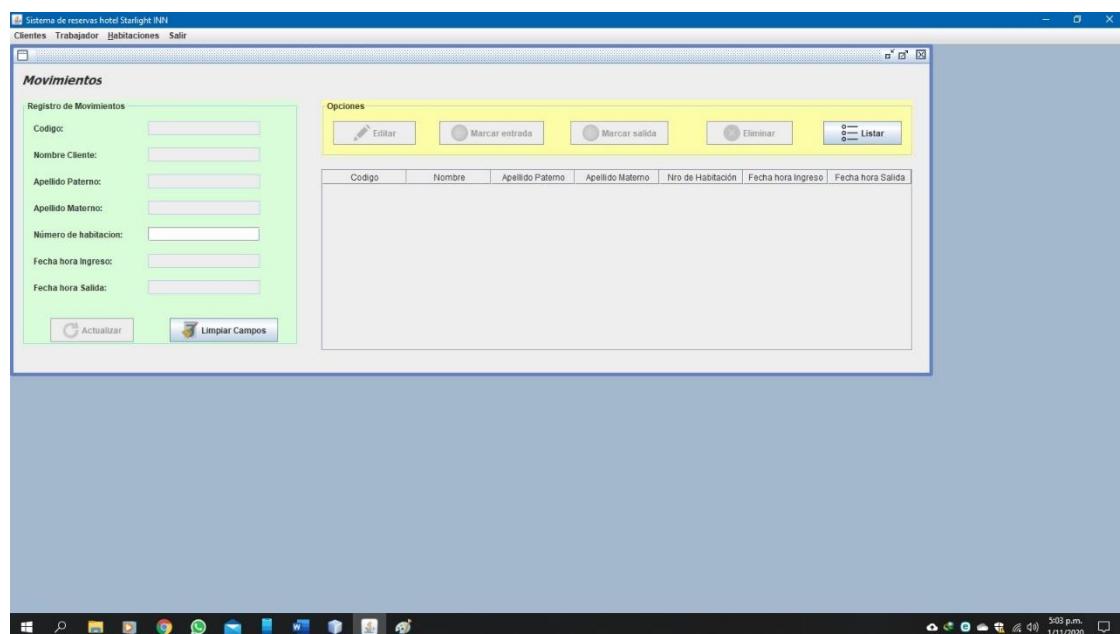


Ilustración 108: Consultar la entrada de cliente - Datos de entrada normal - Versión 1

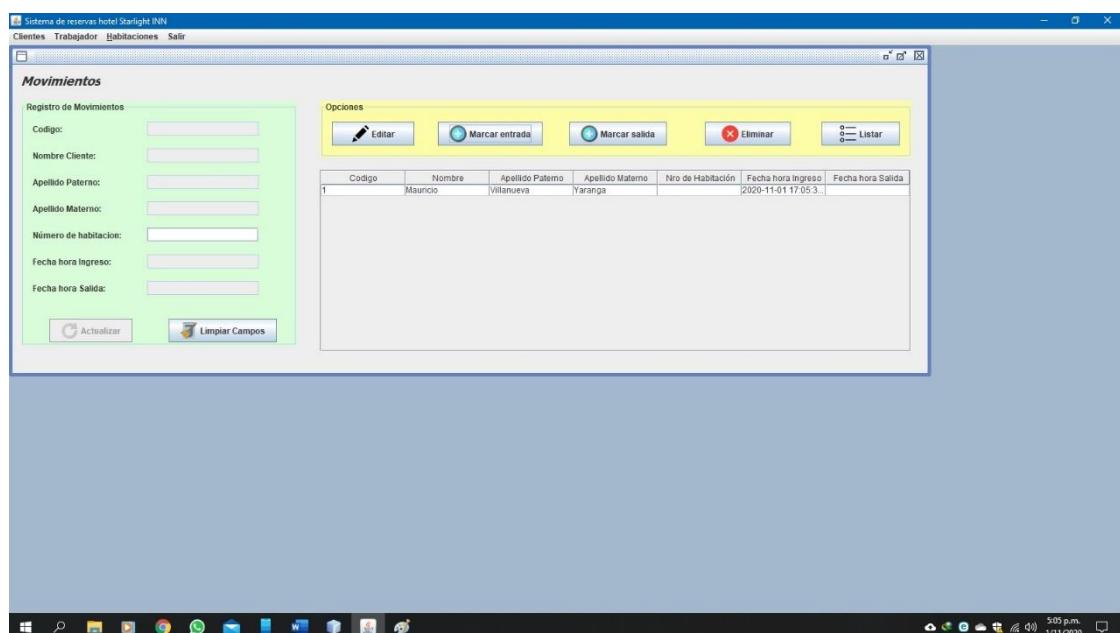


Ilustración 109: Consultar la entrada de cliente - Resultado esperado normal - Versión 1

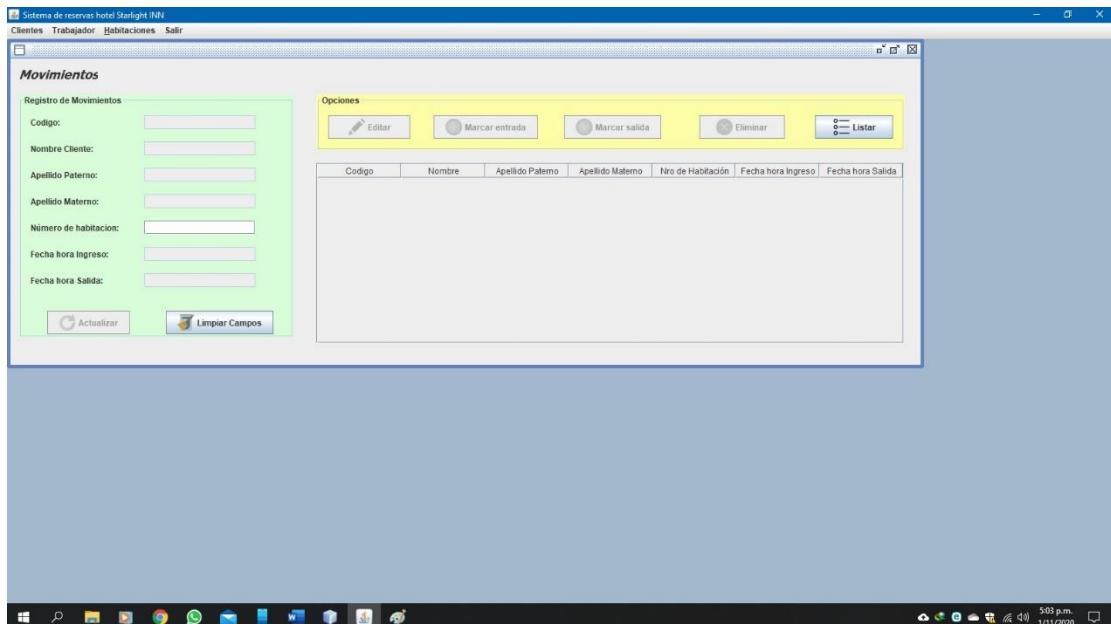


Ilustración 110: Consultar la entrada de cliente - Datos de entrada anómalo - Versión 1

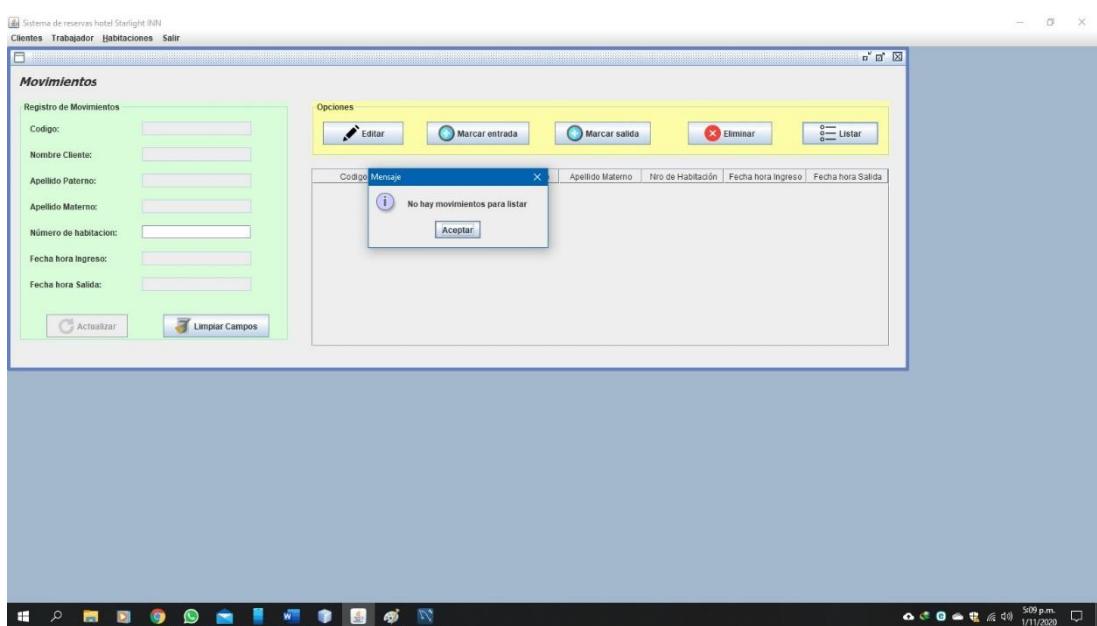


Ilustración 111: Consultar la entrada de cliente - Resultado esperado anómalo - Versión 1

## Caso de uso Registro de Salida Cliente:

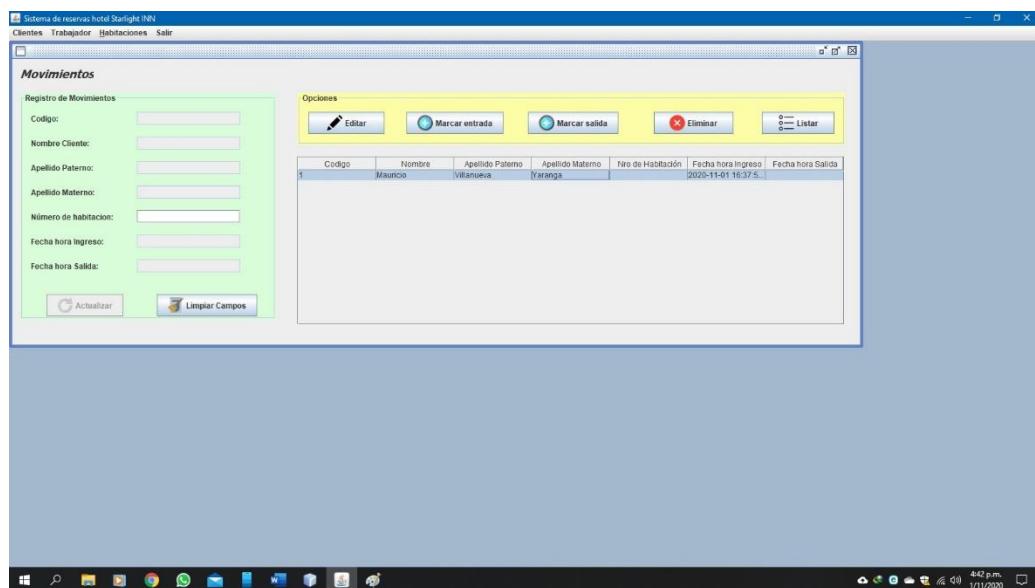


Ilustración 112: Registro de salida de cliente - Datos de entrada normal - Versión 1

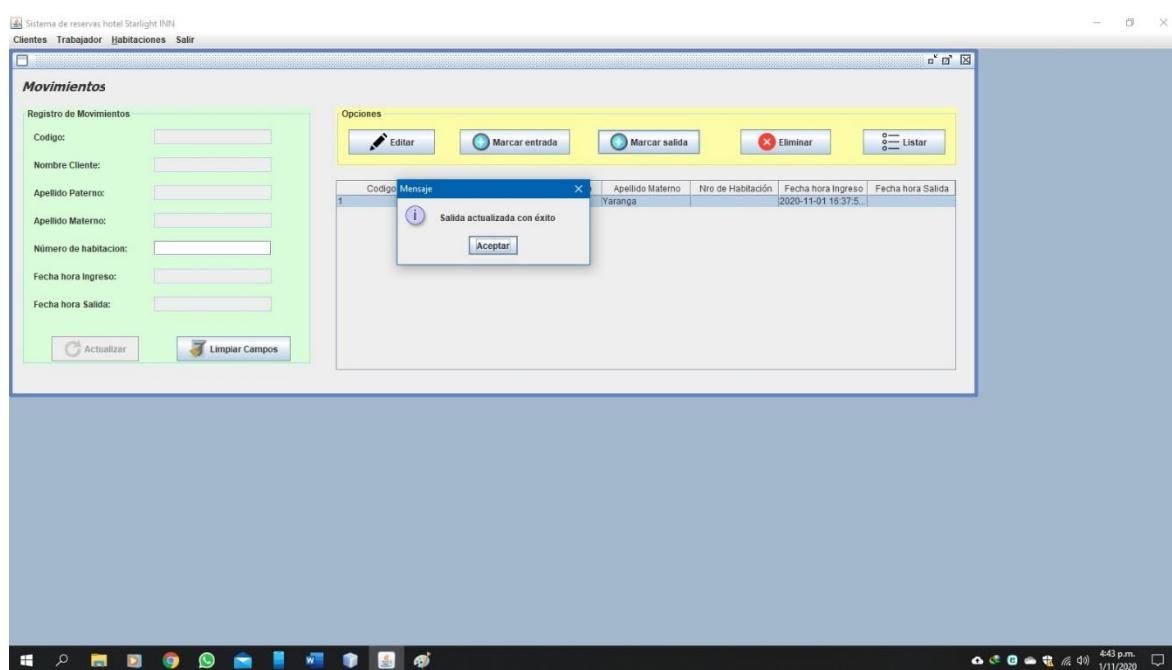


Ilustración 113: Registro de salida de cliente - Resultado esperado normal - Versión 1

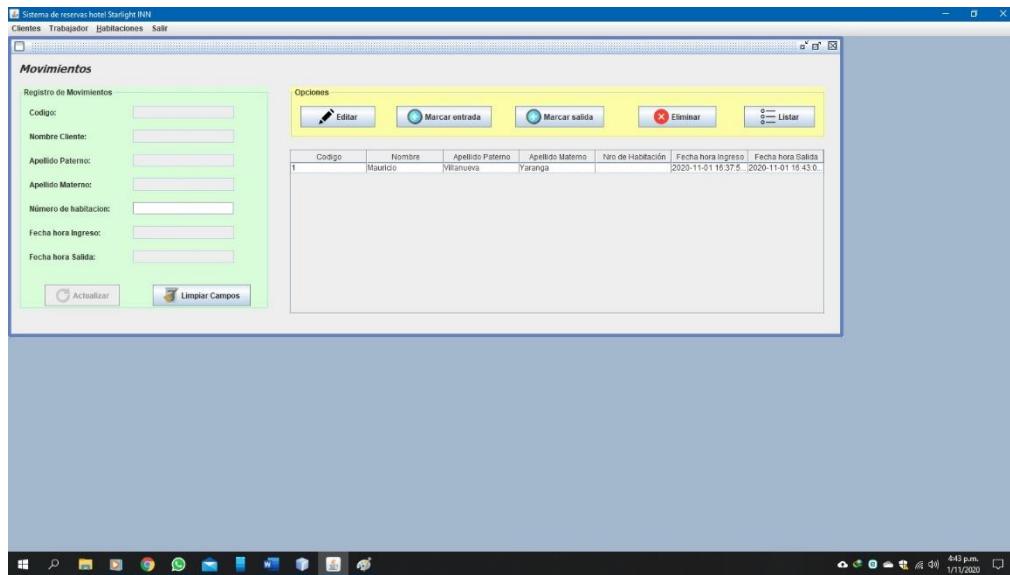


Ilustración 114: Registro de salida de cliente - Datos de entrada anómalo - Versión 1

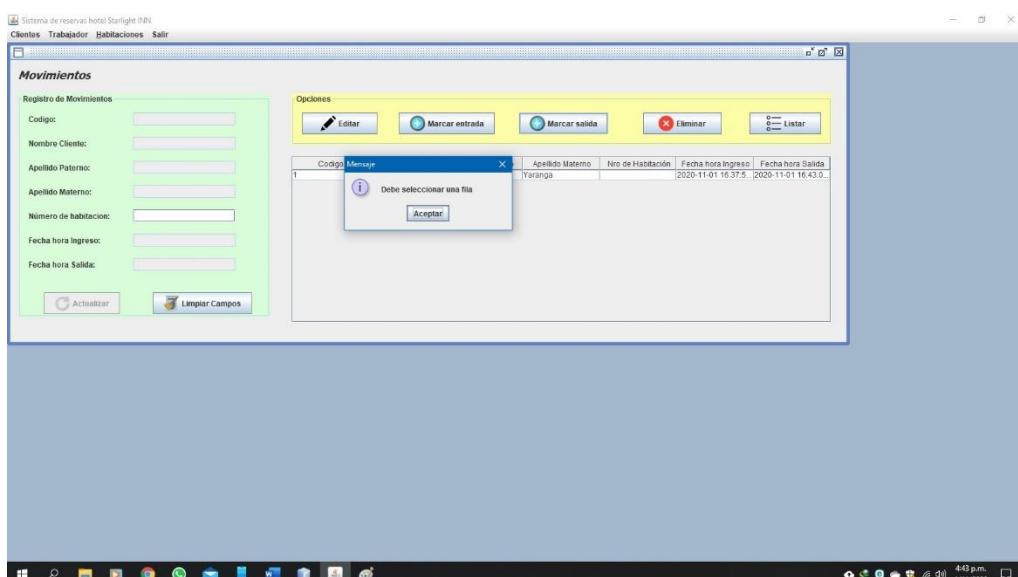


Ilustración 115: Registro de salida de cliente - Resultado esperado anómalo - Versión 1

## Caso de uso Eliminar la Salida Cliente:

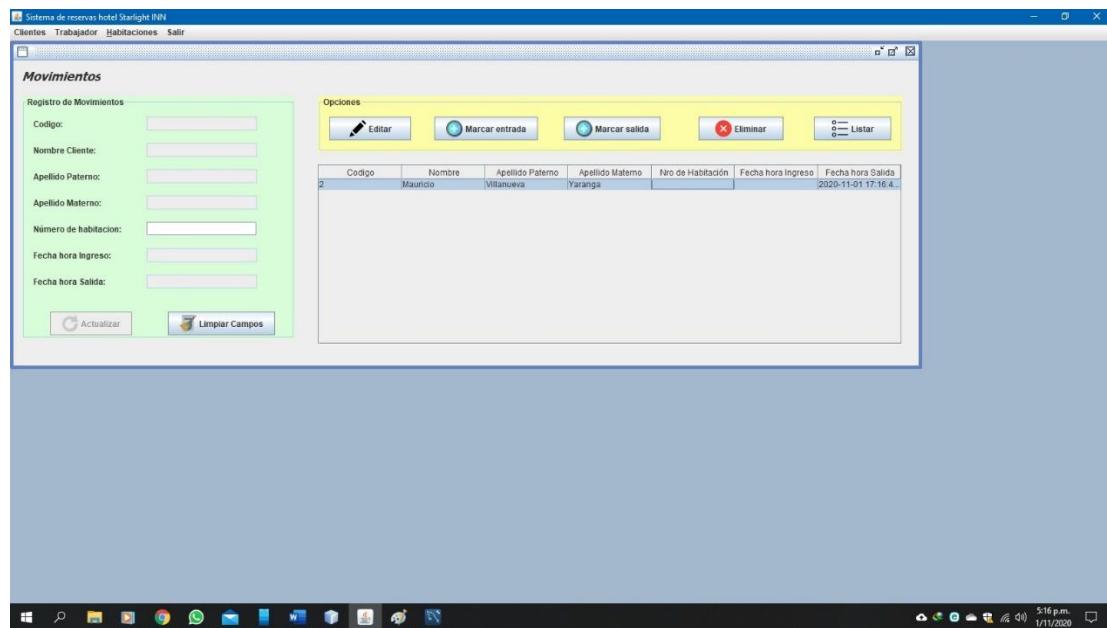


Ilustración 116: Eliminar la salida de cliente - Datos de entrada normal - Versión 1

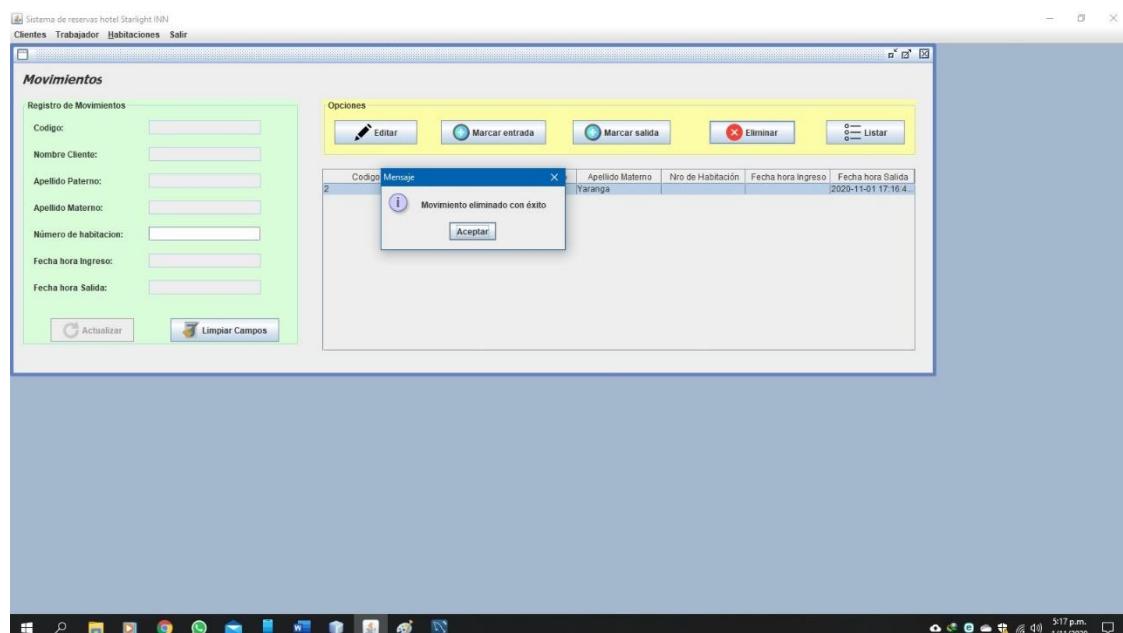


Ilustración 117: Eliminar de salida de cliente - Resultado esperado normal - Versión 1

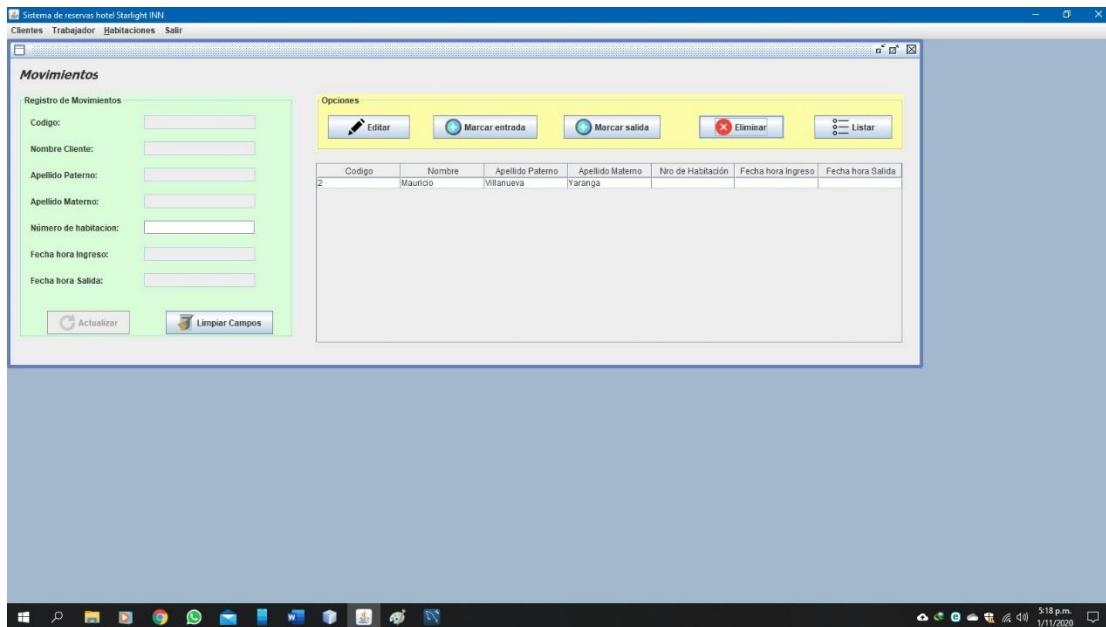


Ilustración 118: Eliminar de salida de cliente - Datos de entrada anómalo - Versión 1

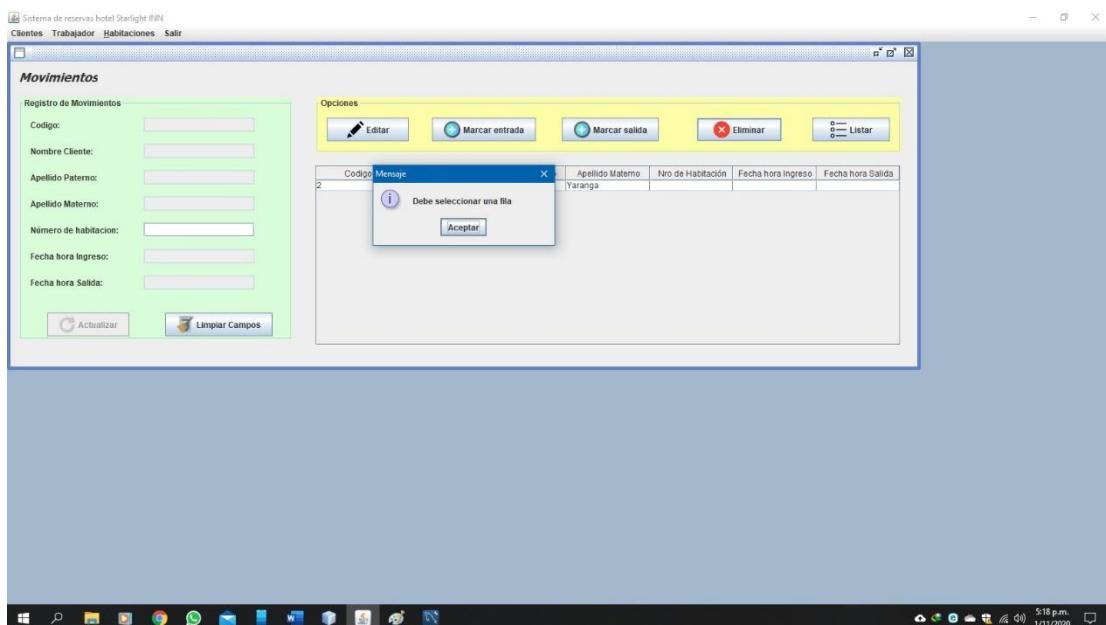


Ilustración 119: Eliminar de salida de cliente - Resultado esperado anómalo - Versión 1

## Caso de uso Modificar Salida de Cliente:

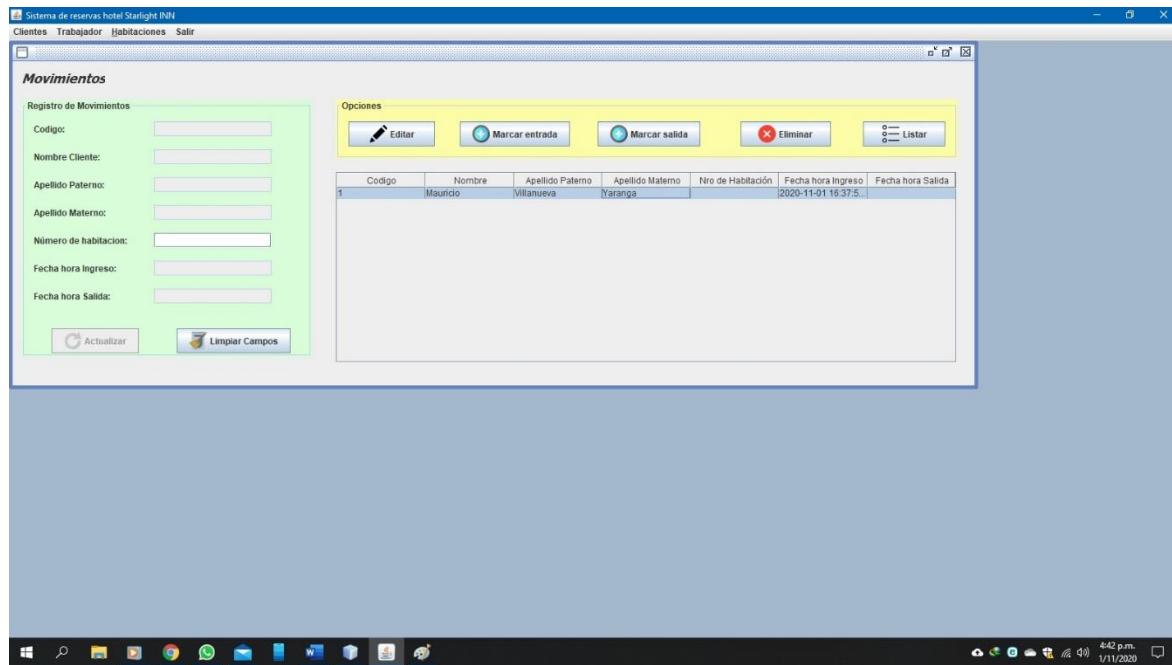


Ilustración 120: Modificar de salida de cliente - Datos de entrada normal - Versión 1

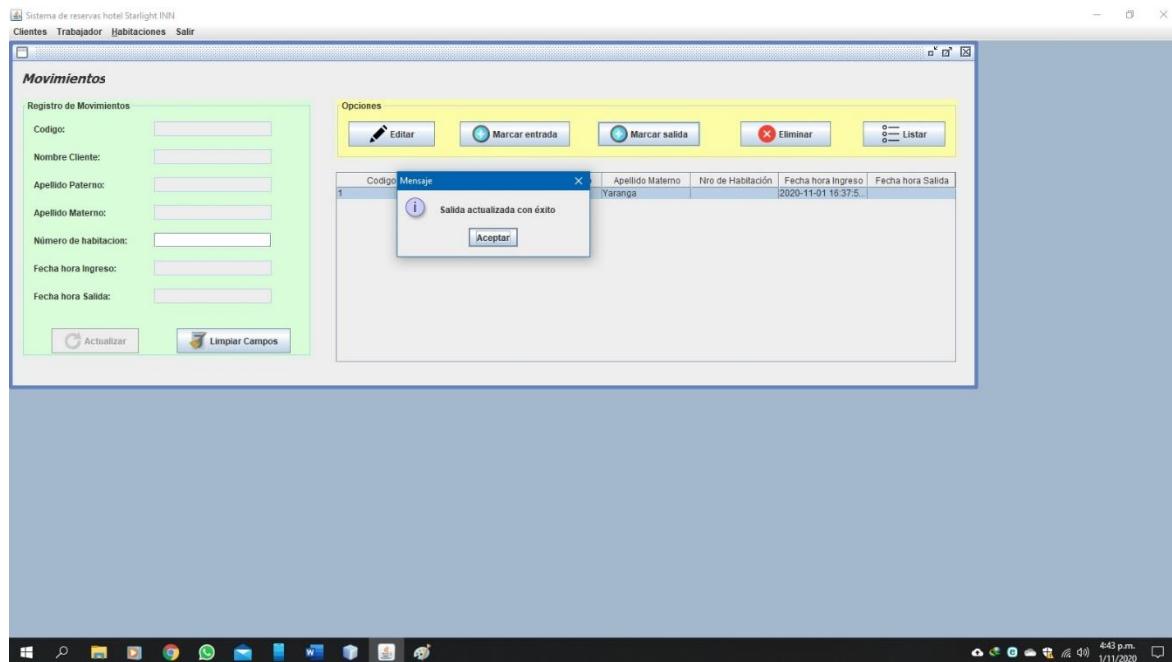


Ilustración 121: Modificar de salida de cliente - Resultado esperado normal - Versión 1

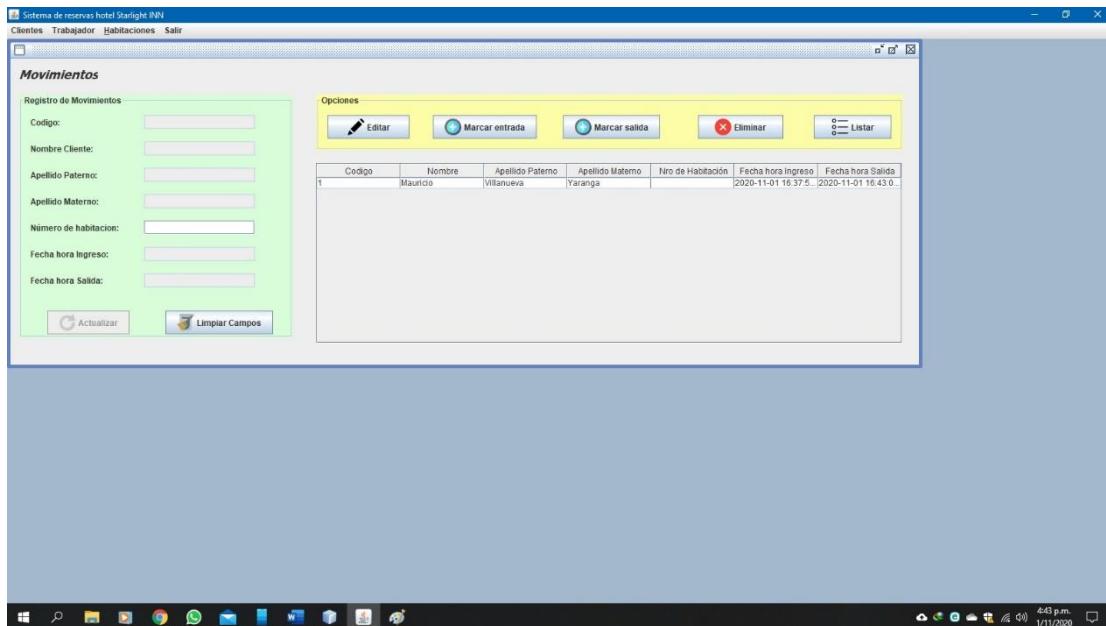


Ilustración 122: Modificar de salida de cliente - Datos de entrada anómalo - Versión 1

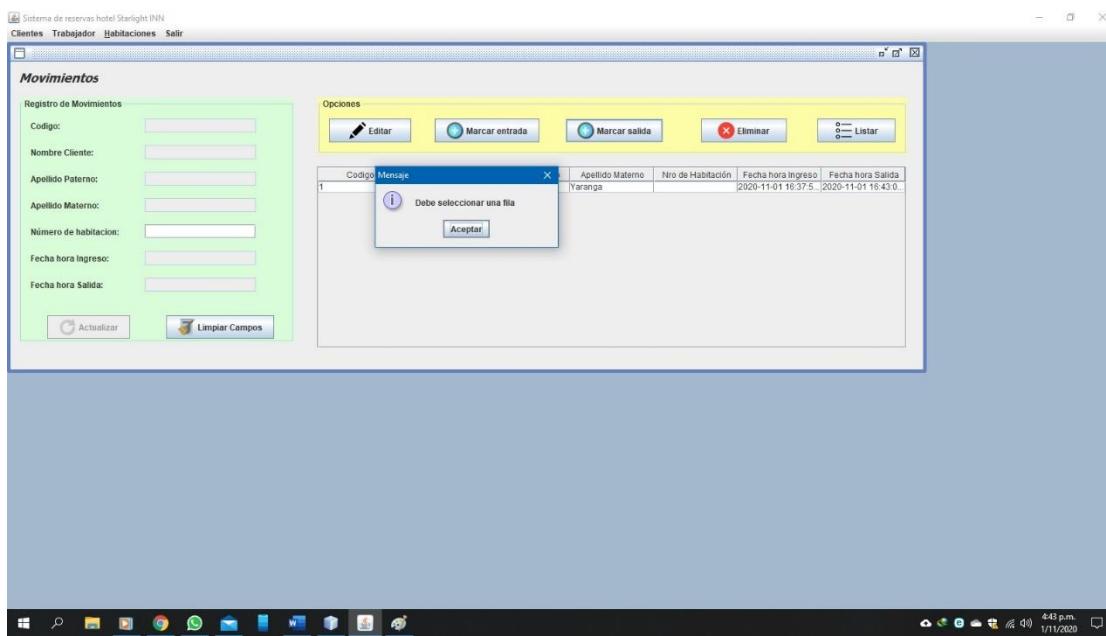


Ilustración 123: Modificar de salida de cliente - Resultado esperado anómalo - Versión 1

## Caso de uso Consultar la Salida Cliente:

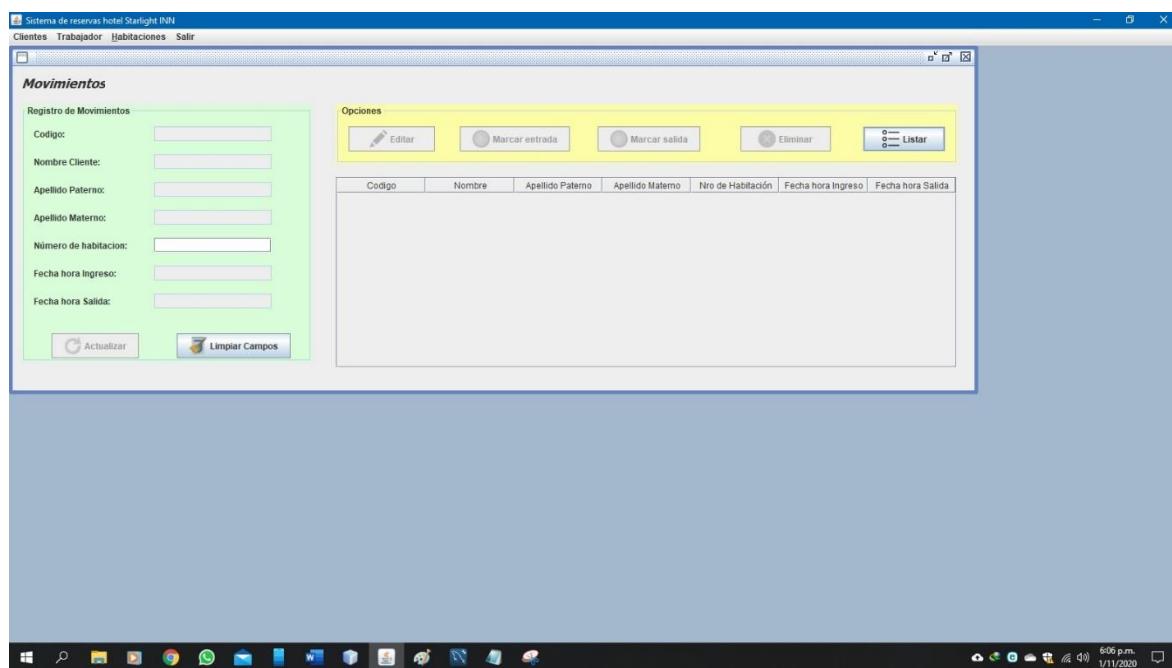


Ilustración 124: Consultar la salida de cliente - Datos de entrada normal - Versión 1

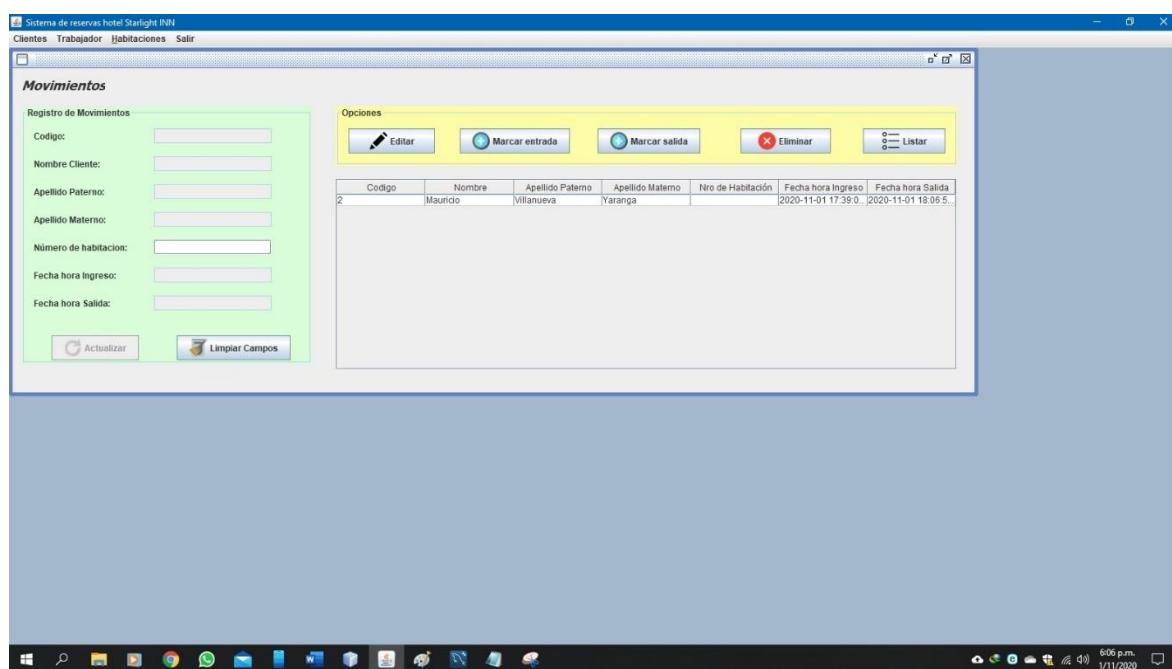


Ilustración 125: Consultar la salida de cliente - Resultado esperado normal - Versión 1

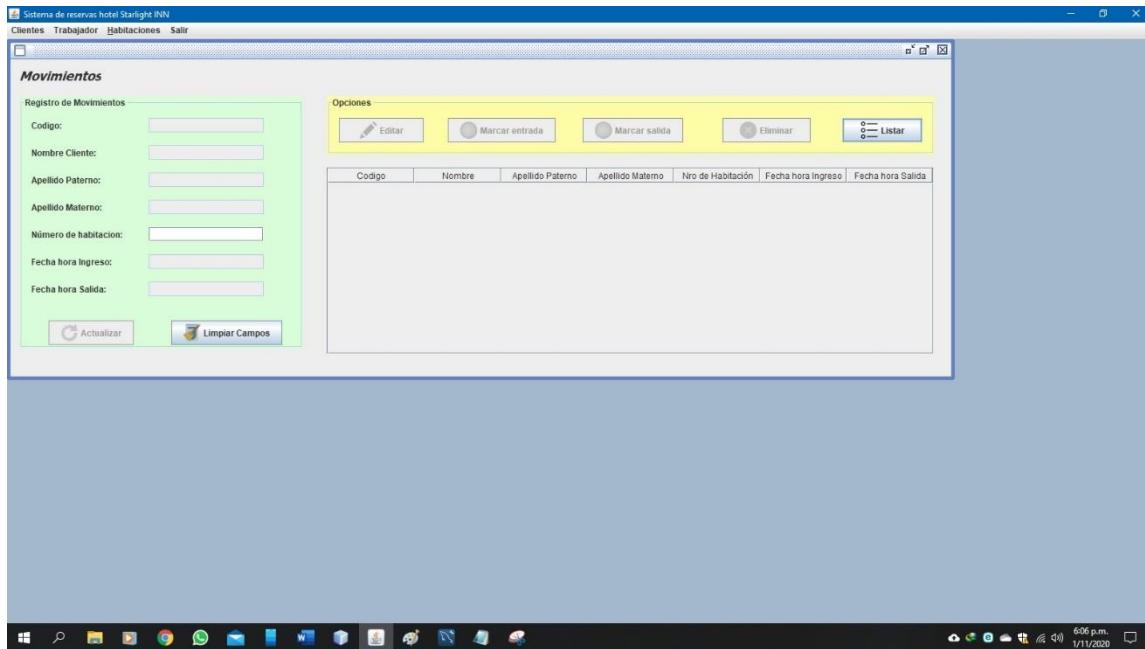


Ilustración 126: Consultar la salida de cliente - Datos de entrada anómalo - Versión 1

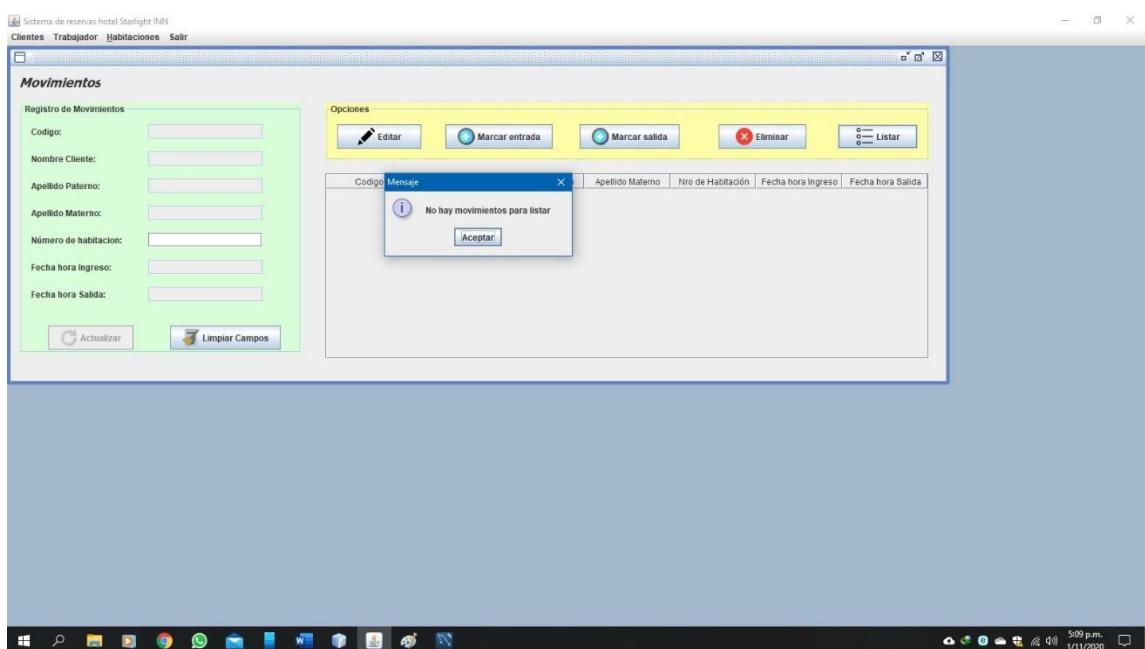


Ilustración 127: Consultar la salida de cliente - Resultado esperado anómalo - Versión 1

## Caso de uso Registrar Habitaciones:

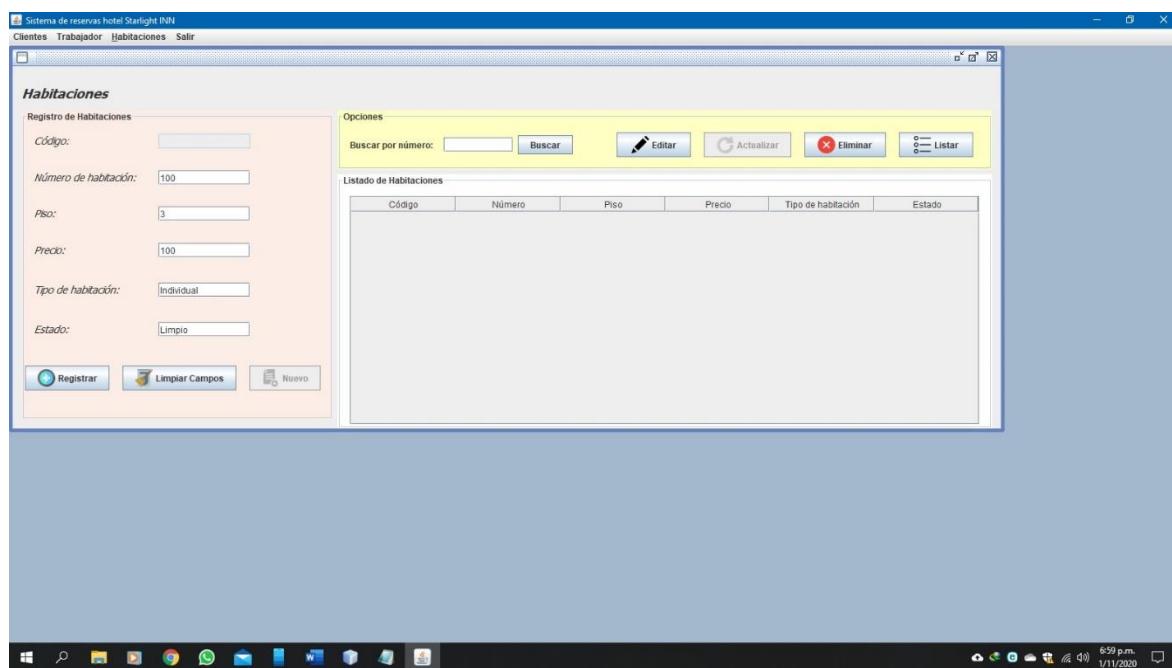


Ilustración 128: Registrar Habitaciones - Datos de entrada normal - Versión 2

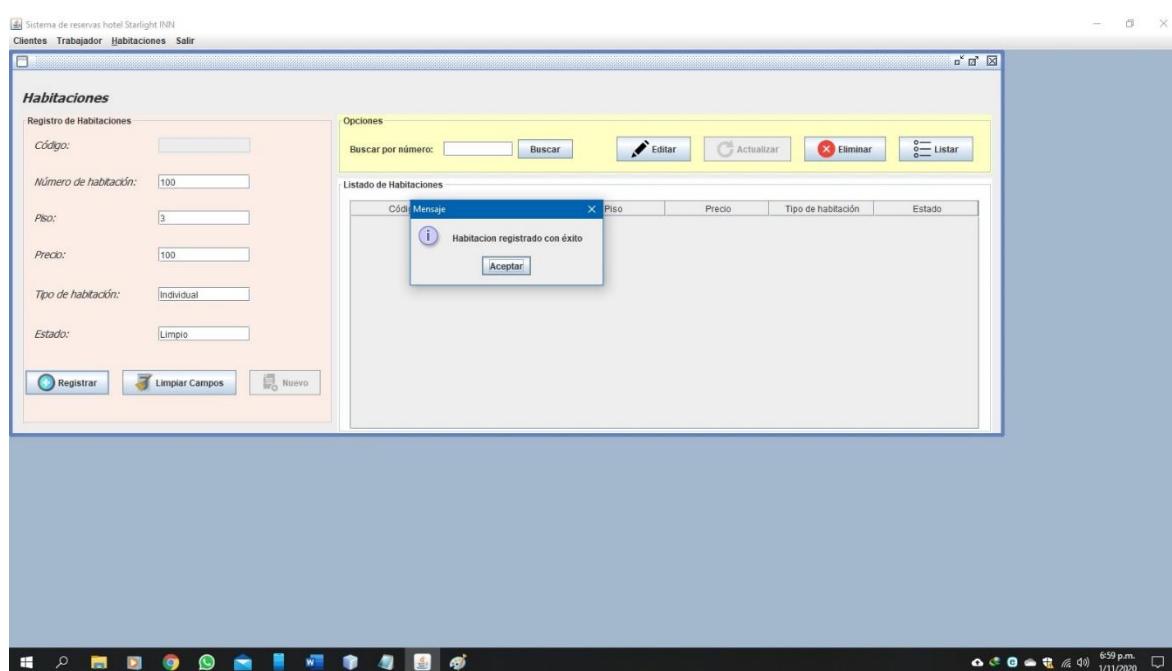


Ilustración 129: Registrar Habitaciones - Resultado esperado normal - Versión 2

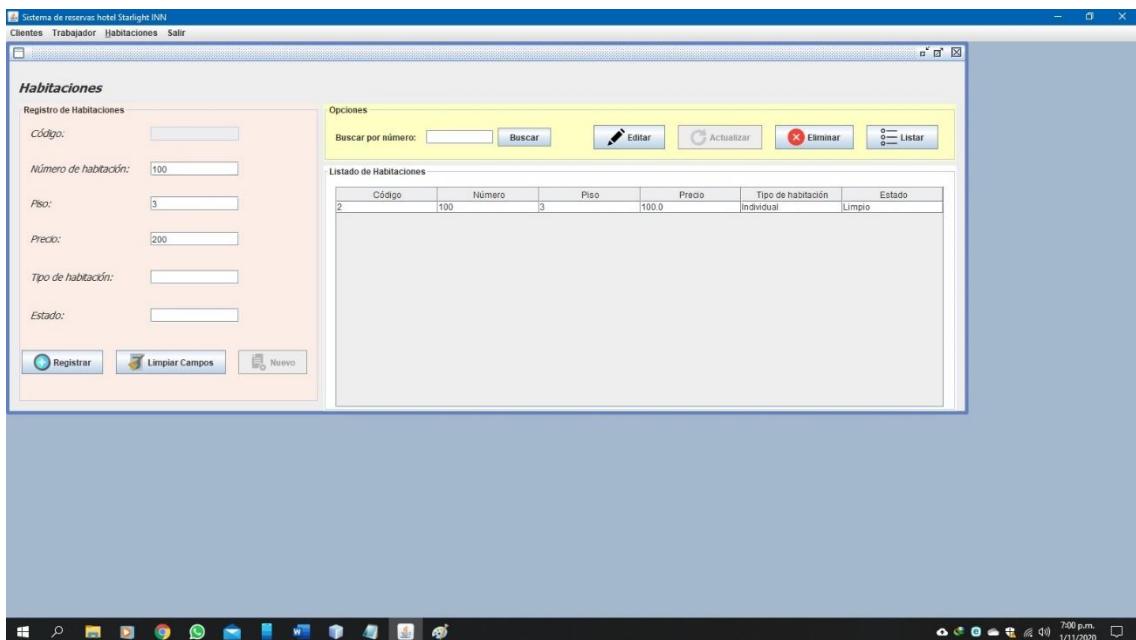


Ilustración 130: Registrar Habitaciones - Datos de entrada anómalo - Versión 2

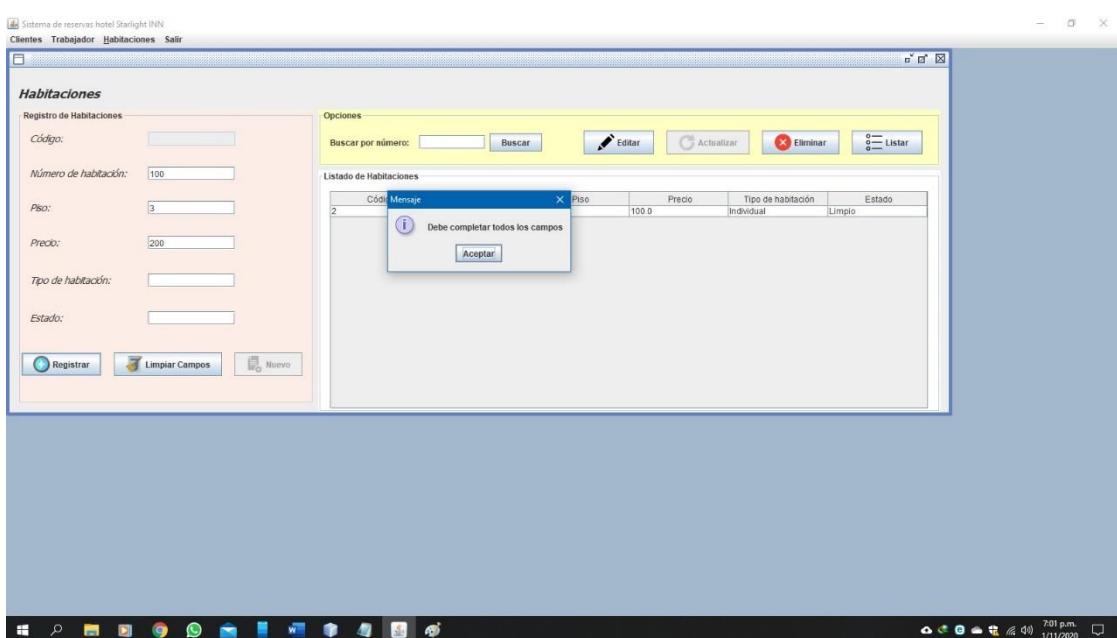


Ilustración 131: Registrar Habitaciones - Resultado esperado anómalo - Versión 2

## Caso de uso Eliminar Habitaciones:

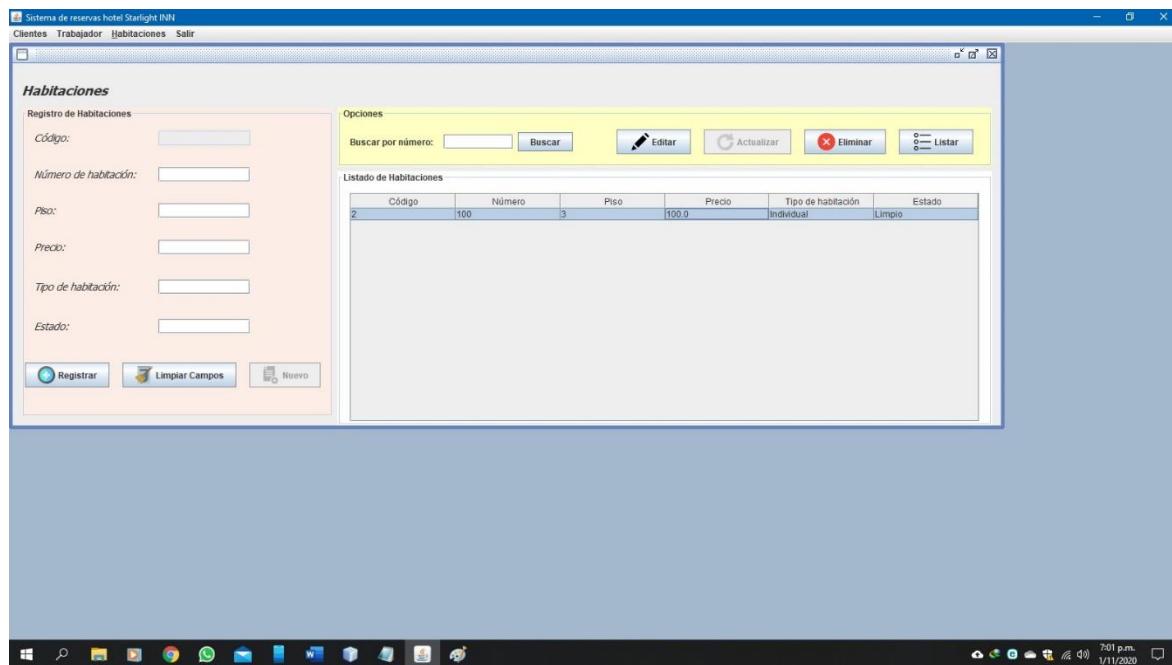


Ilustración 132: Eliminar Habitaciones - Datos de entrada normal - Versión 2

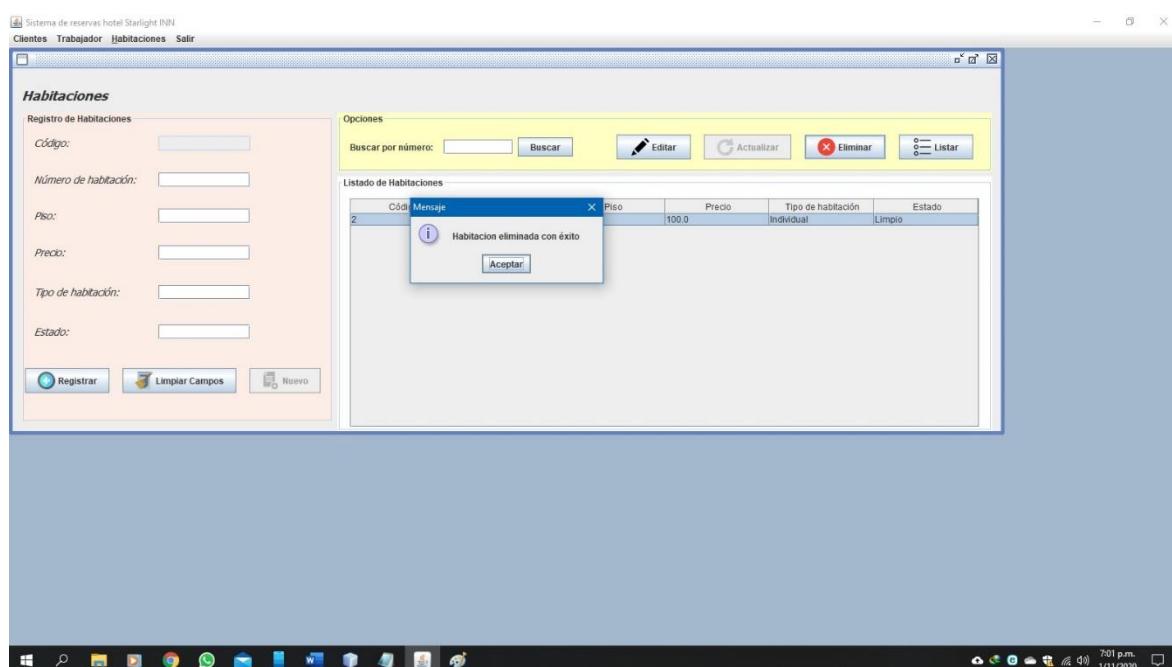


Ilustración 133: Eliminar Habitaciones - Resultado esperado normal - Versión 2

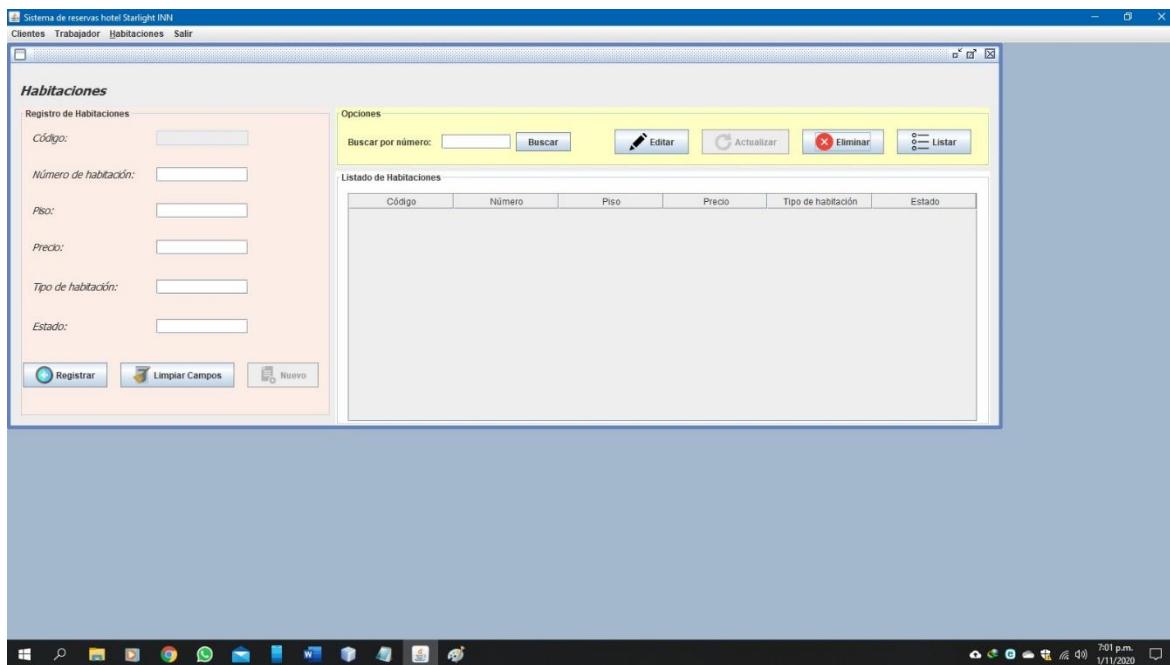


Ilustración 134: Eliminar Habitaciones - Datos de entrada anómalo - Versión 2

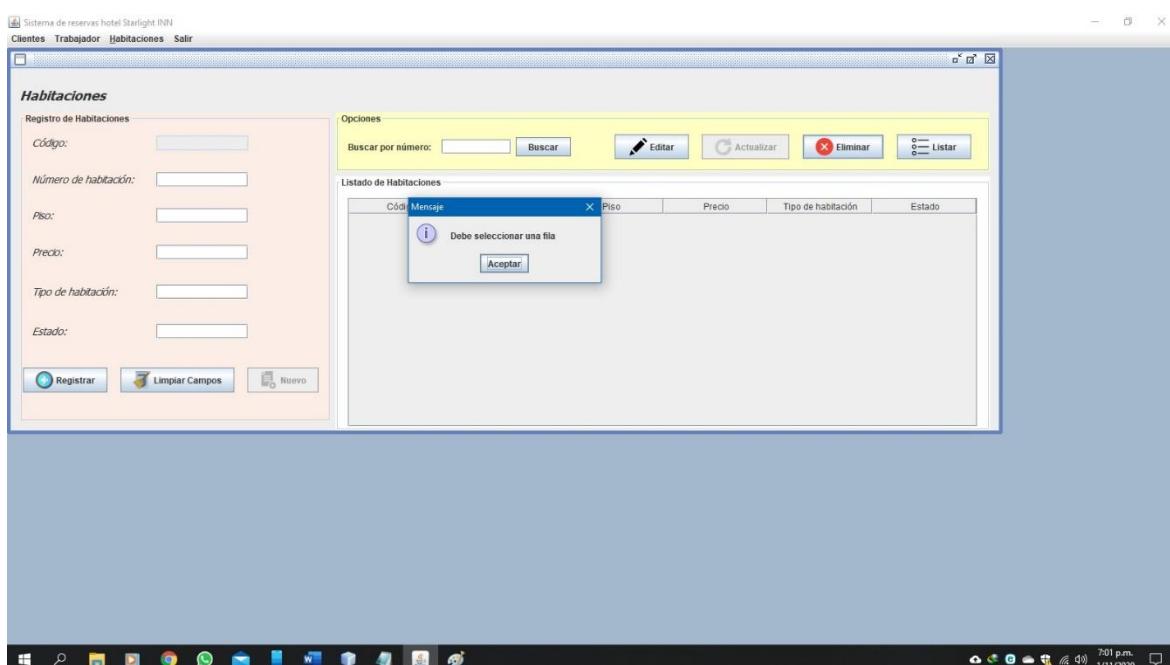


Ilustración 135: Eliminar Habitaciones - Resultado esperado anómalo - Versión 2

## Caso de uso Modificar Habitaciones:

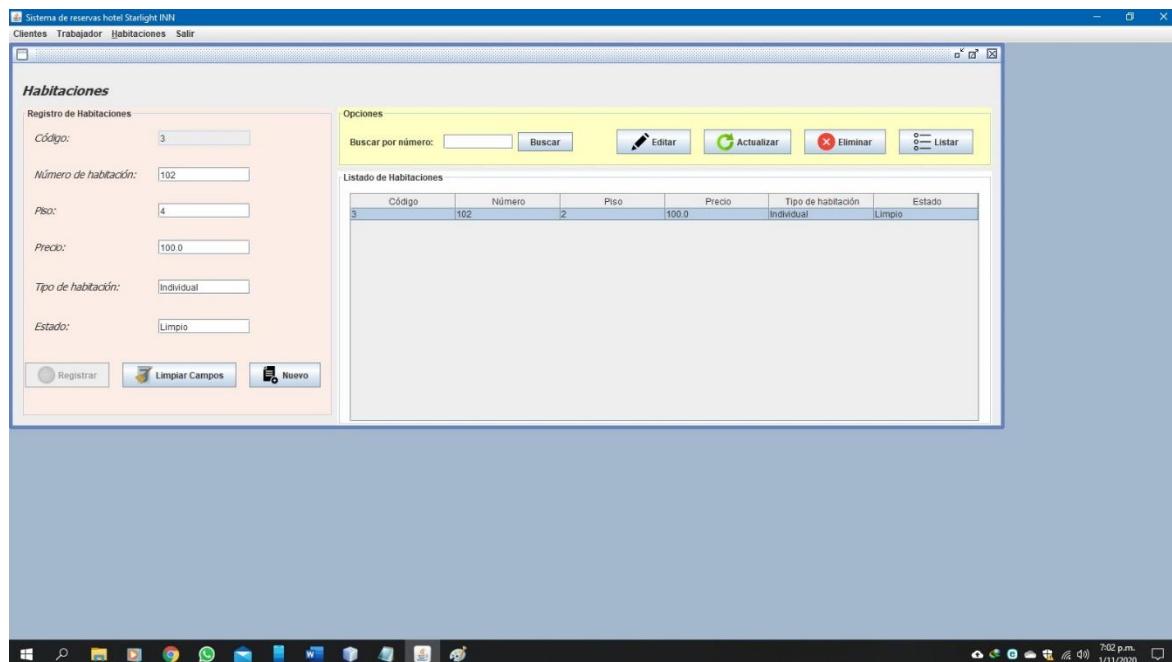


Ilustración 136: Modificar Habitaciones - Datos de entrada normal - Versión 2

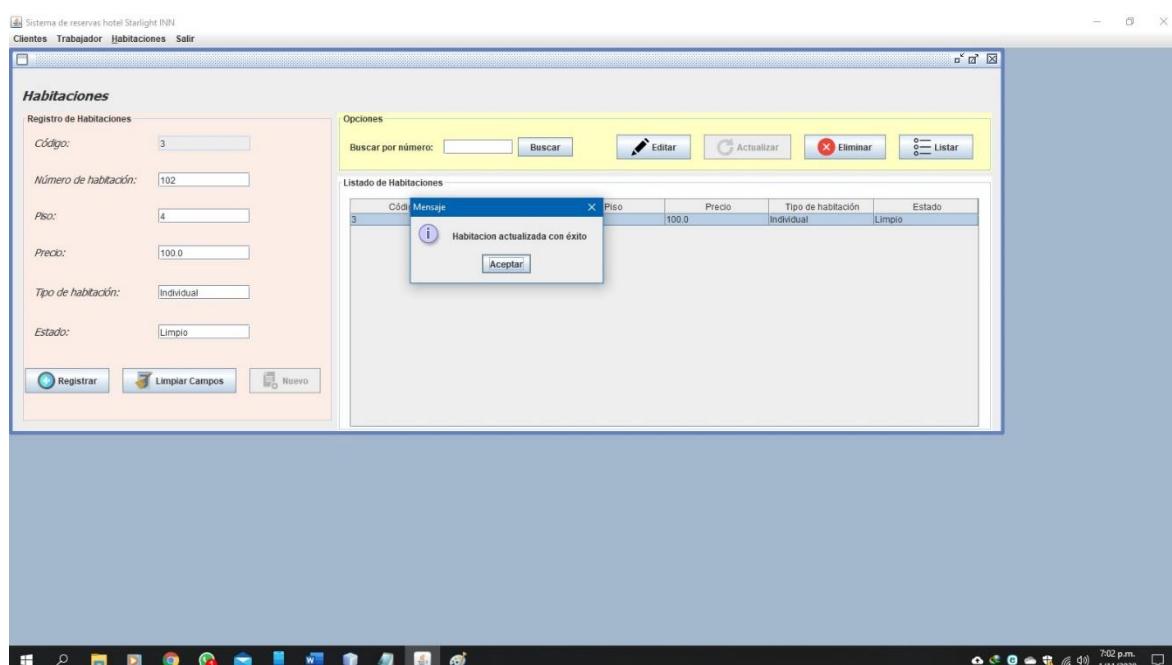


Ilustración 137: Modificar Habitaciones - Resultado esperado normal - Versión 2

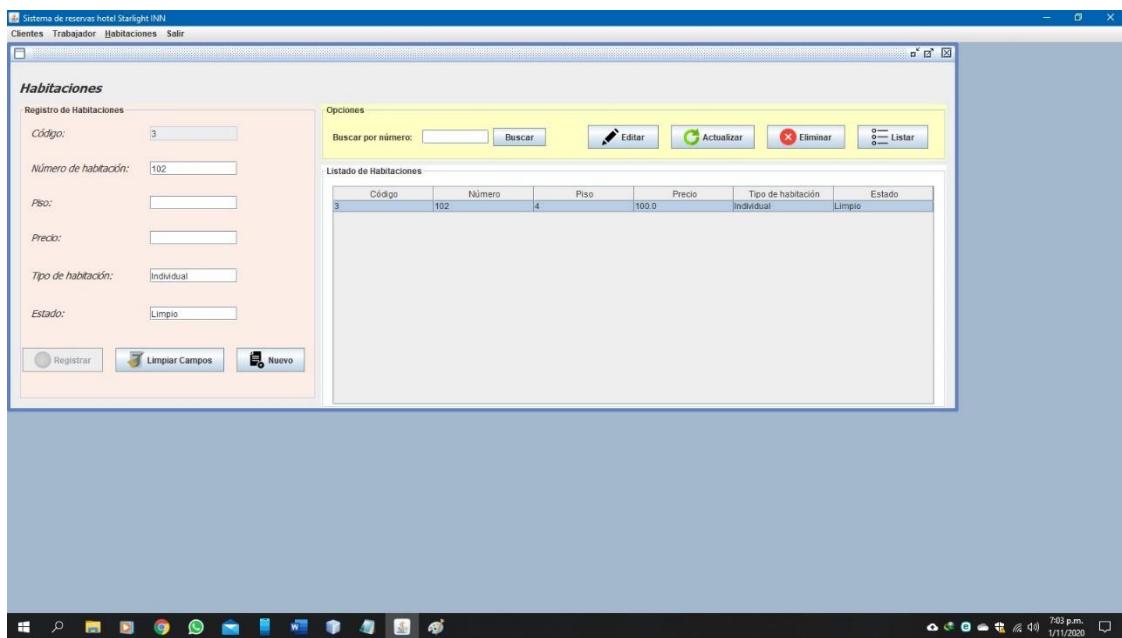


Ilustración 138: Modificar Habitaciones - Datos de entrada anómalo - Versión 2

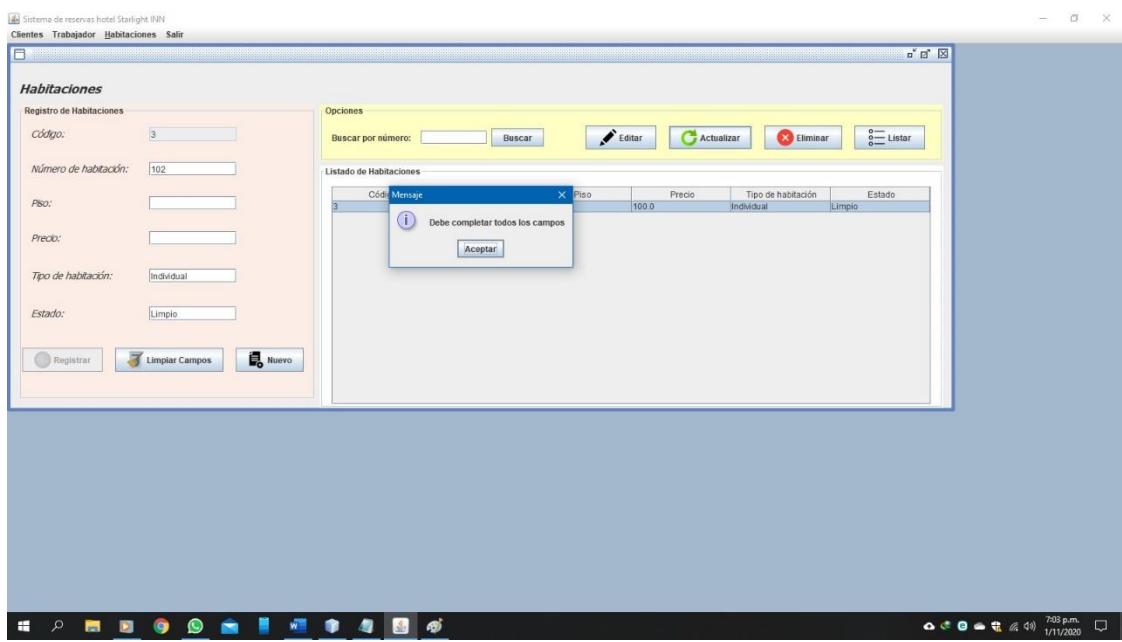


Ilustración 139: Modificar Habitaciones - Resultado esperado anómalo - Versión 2

## Caso de uso Consultar Habitaciones:

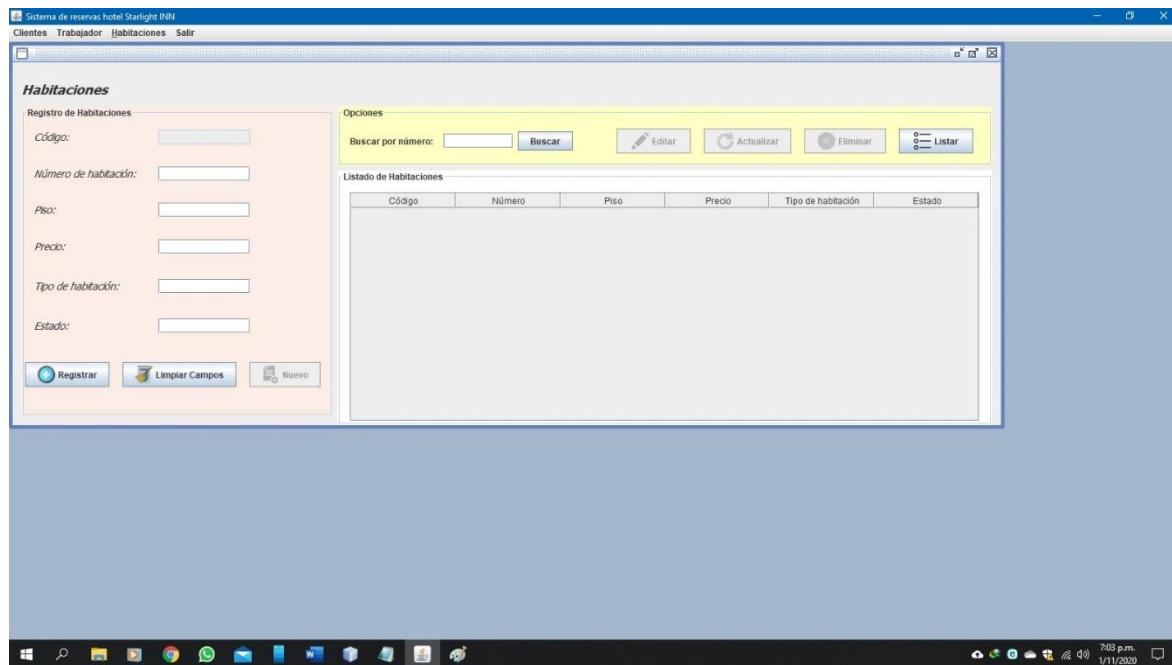


Ilustración 140: Consultar Habitaciones - Datos de entrada normal - Versión 2

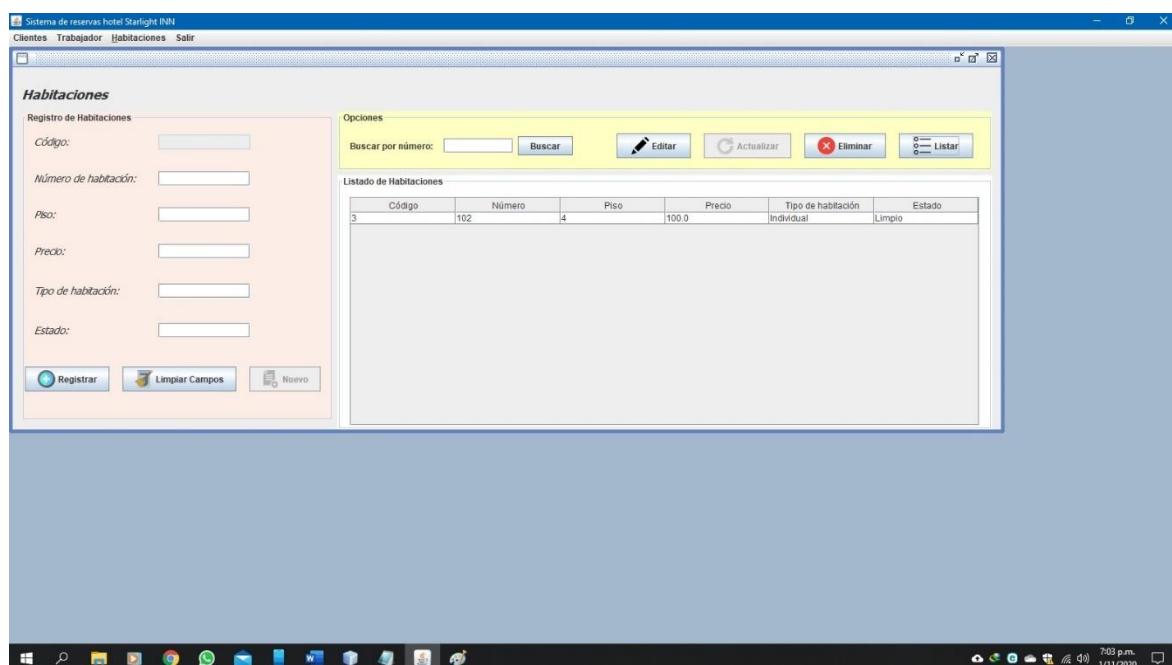


Ilustración 141: Consultar Habitaciones - Resultado esperado normal - Versión 2

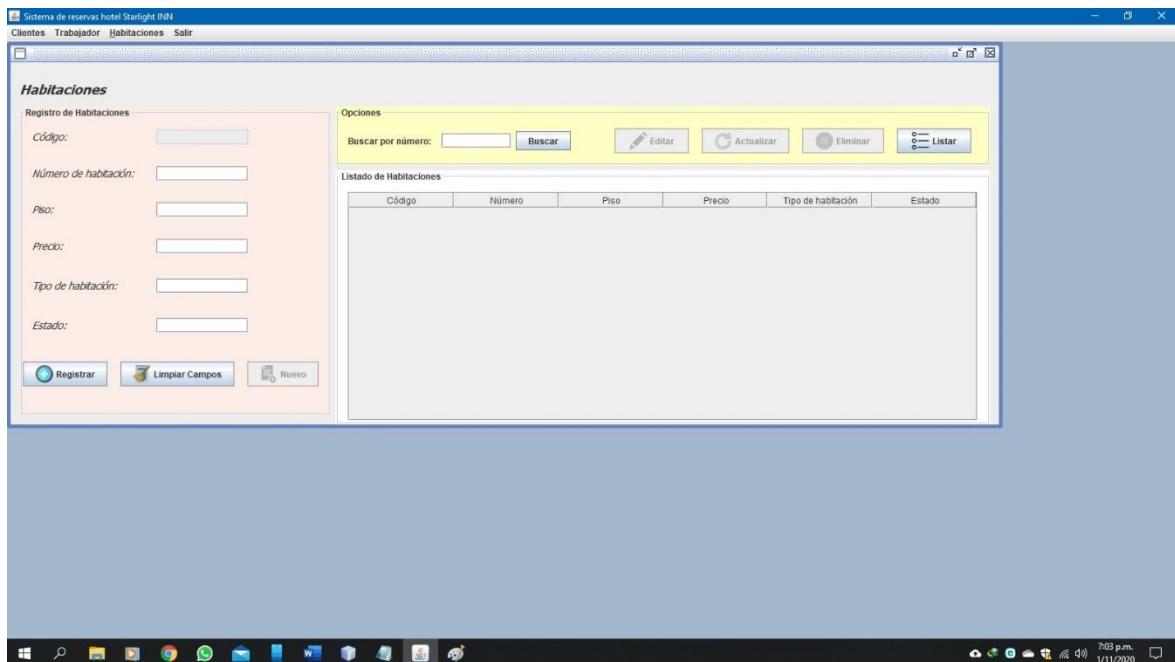


Ilustración 142: Consultar Habitaciones - Datos de entrada anómalo - Versión 2

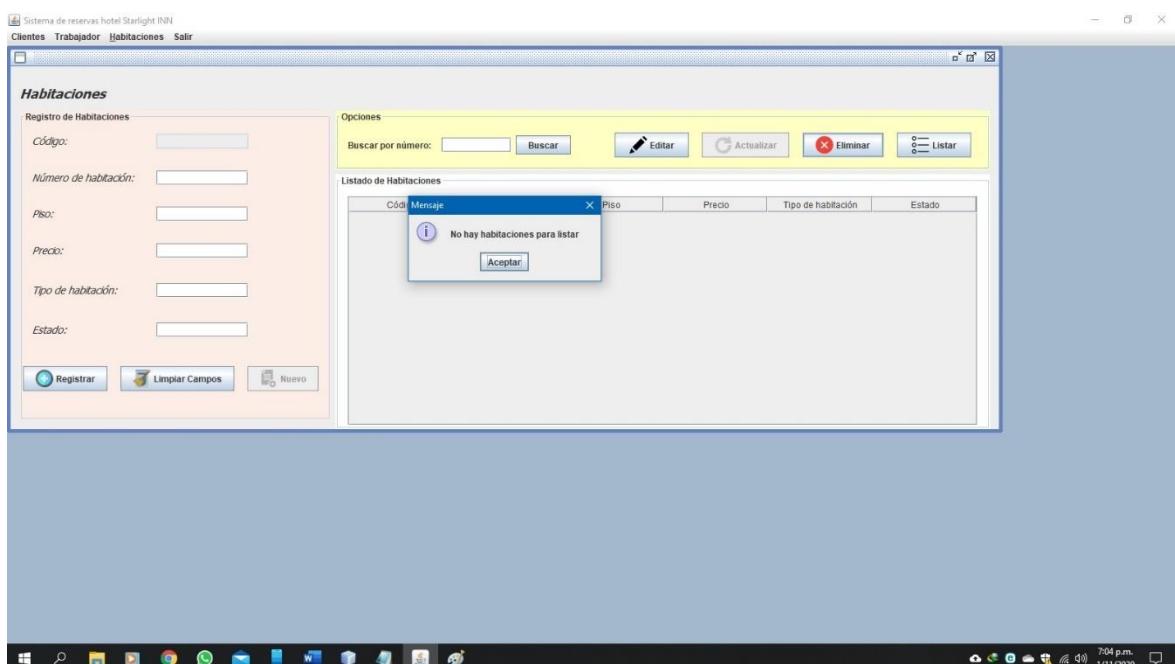


Ilustración 143: Consultar Habitaciones - Resultado esperado anómalo - Versión 2

## Caso de uso Registrar Reserva:

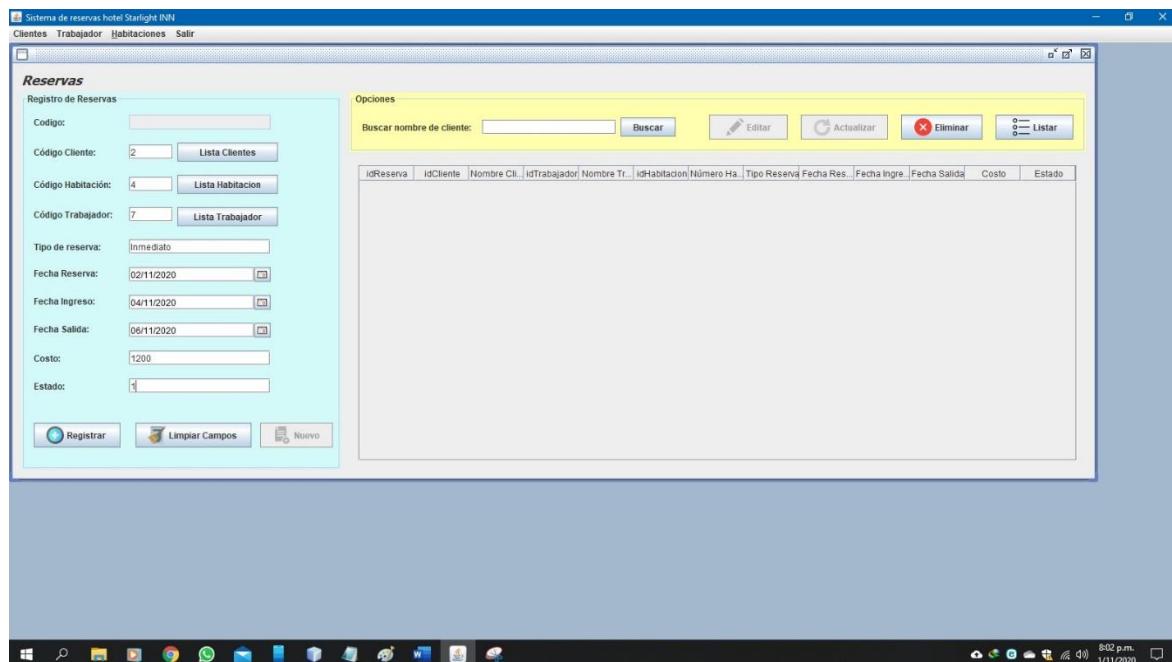


Ilustración 144: Registrar Reserva - Datos de entrada normal - Versión 2

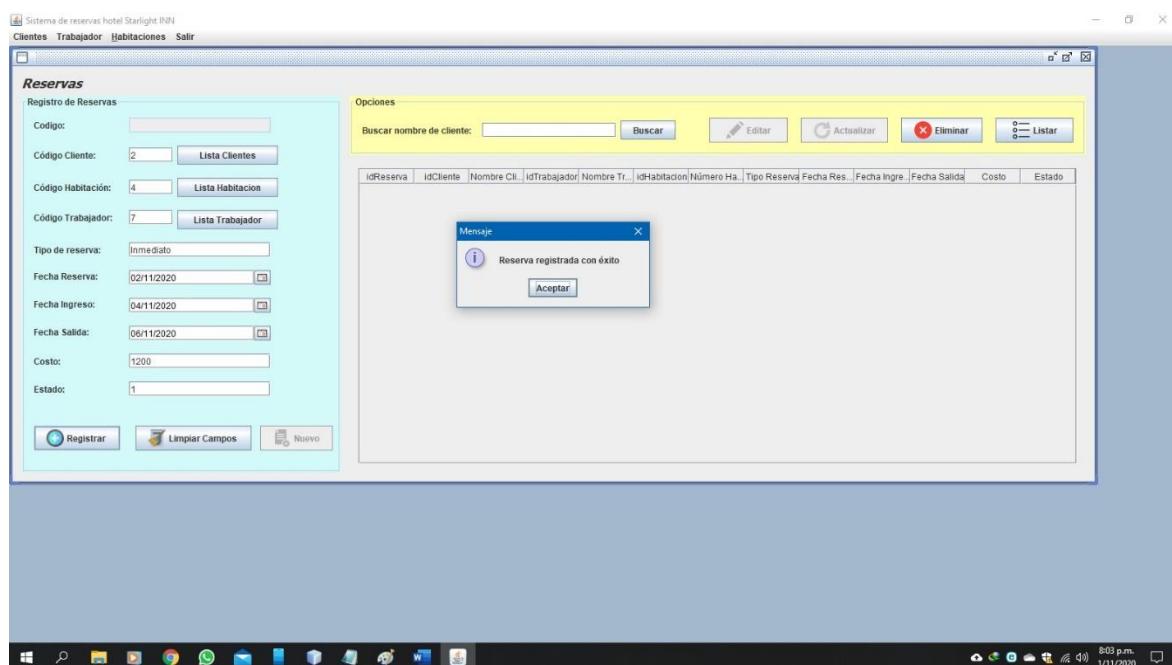


Ilustración 145: Registrar Reserva - Resultado esperado normal - Versión 2

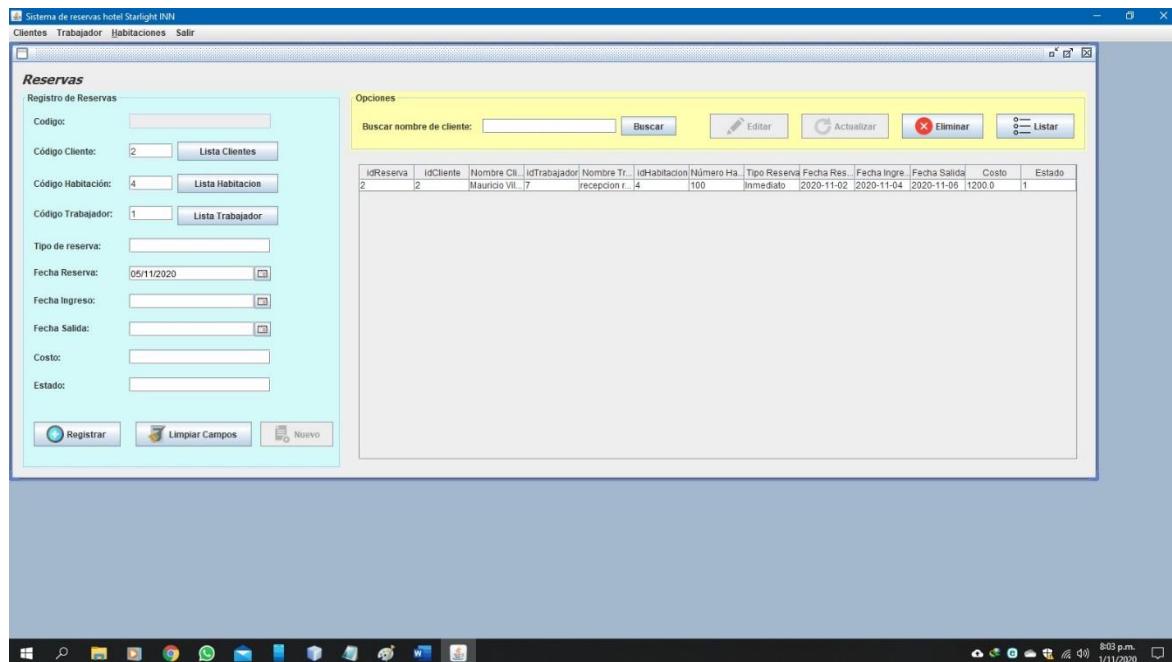


Ilustración 146: Registrar Reserva - Datos de entrada anómalo - Versión 2

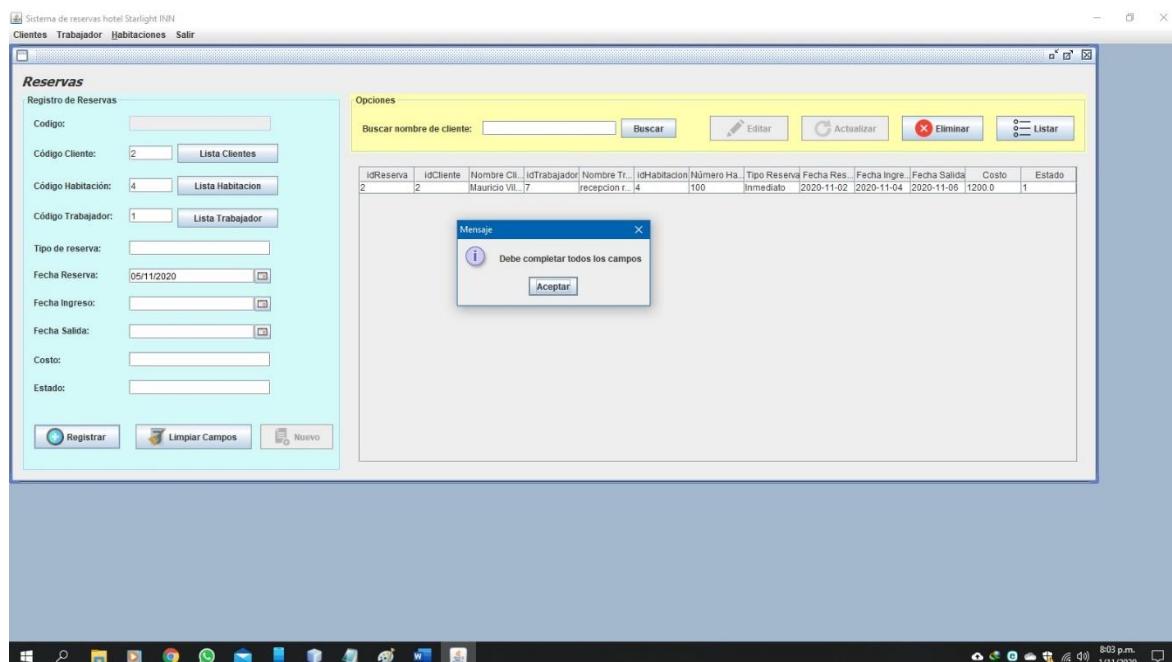


Ilustración 147: Registrar Reserva - Resultado esperado anómalo - Versión 2

## Caso de uso Eliminar Reserva:

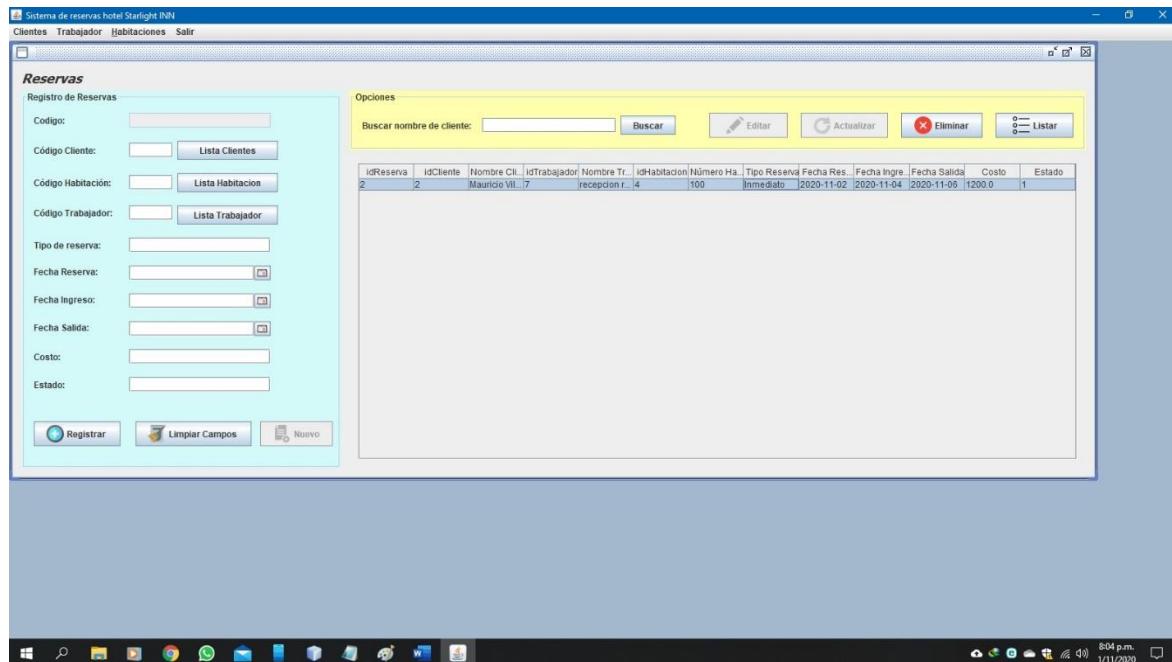


Ilustración 148: Eliminar Reserva - Datos de entrada normal - Versión 2

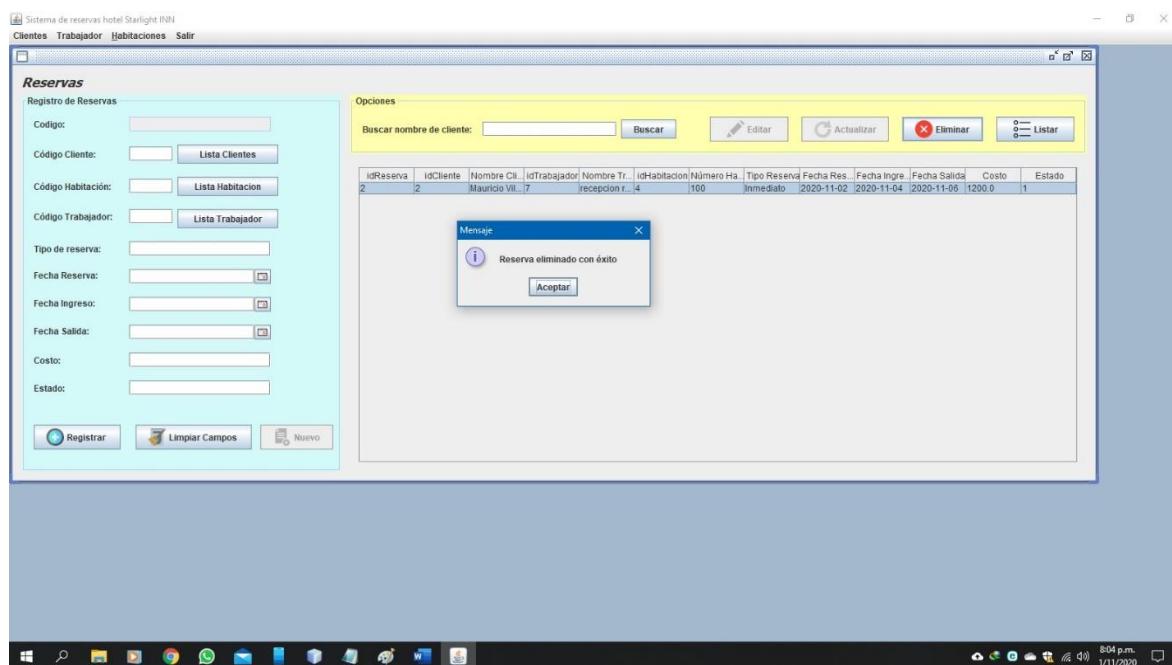


Ilustración 149: Eliminar Reserva - Resultado esperado normal - Versión 2

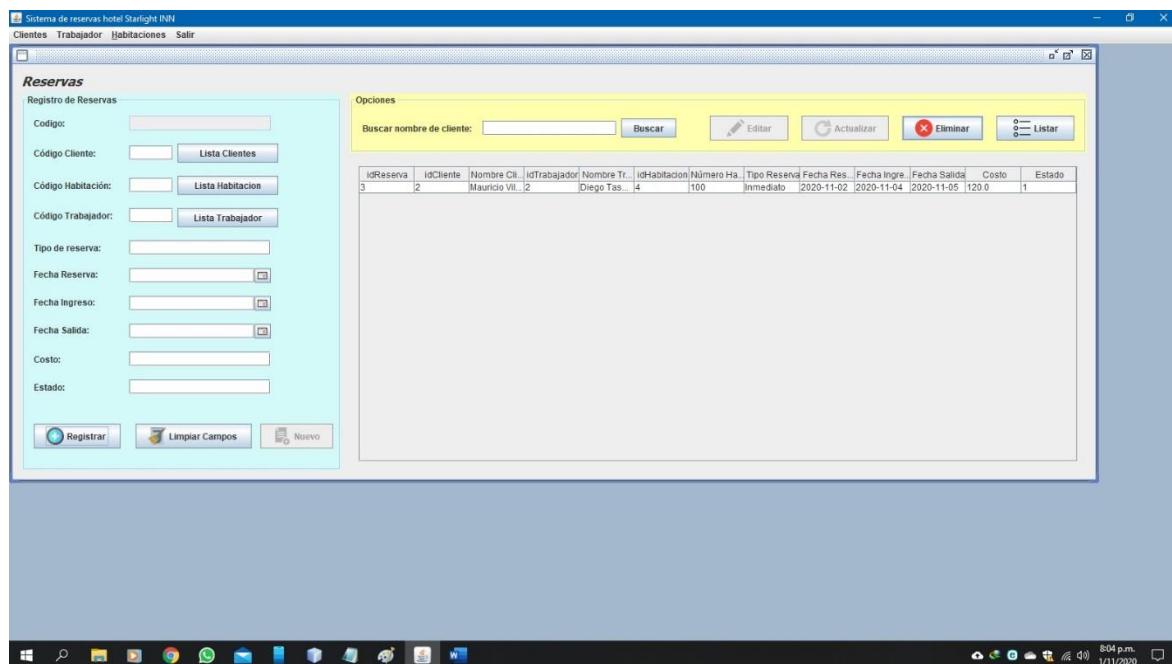


Ilustración 150: Eliminar Reserva - Datos de entrada anómalo - Versión 2

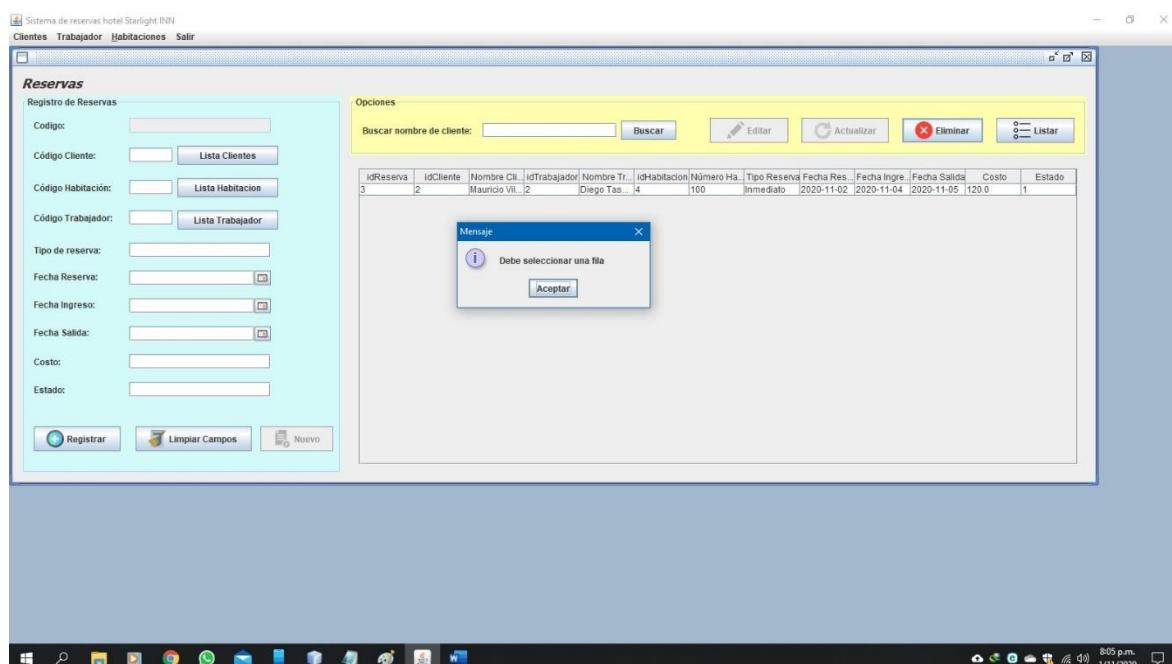


Ilustración 151: Eliminar Reserva - Resultado esperado anómalo - Versión 2

## Caso de uso Modificar Reserva:

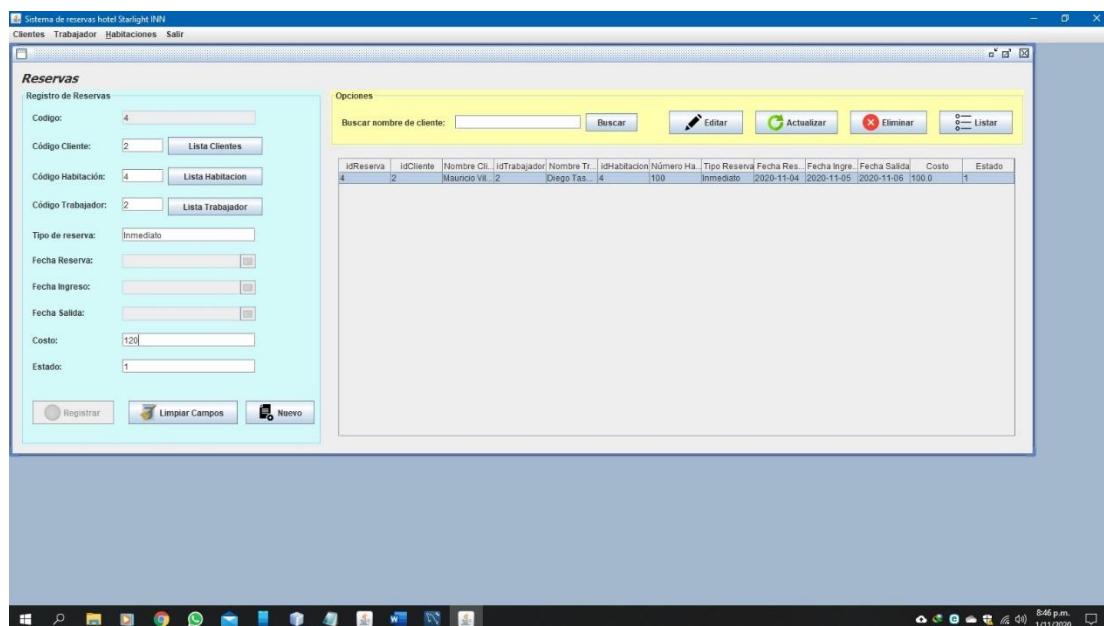


Ilustración 152: Modificar Reserva - Datos de entrada normal - Versión 2

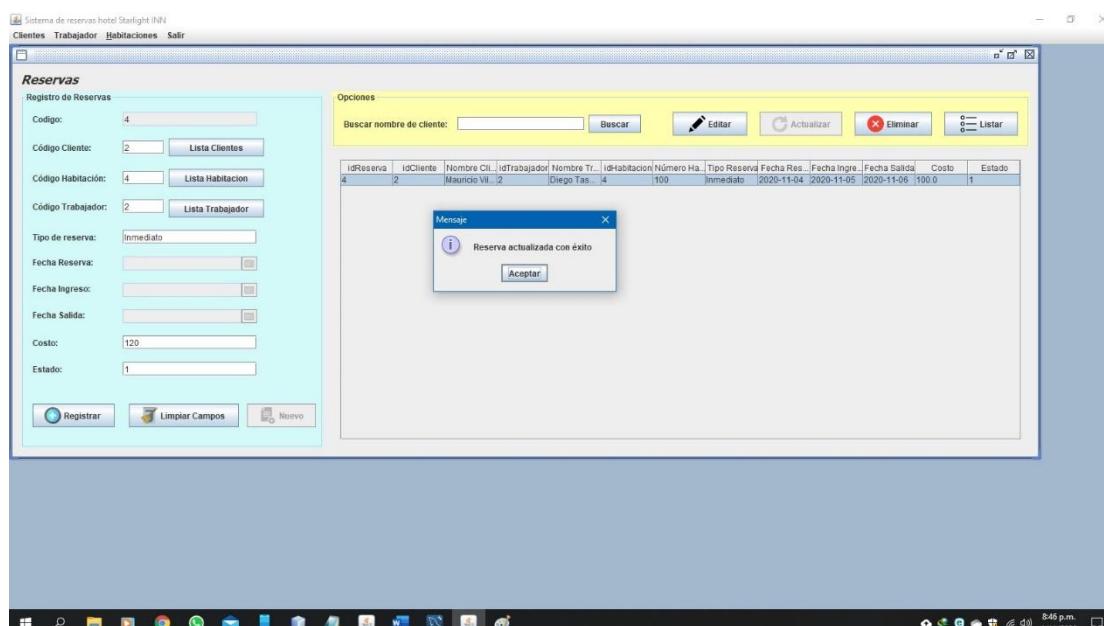


Ilustración 153: Modificar Reserva - Resultado esperado normal - Versión 2

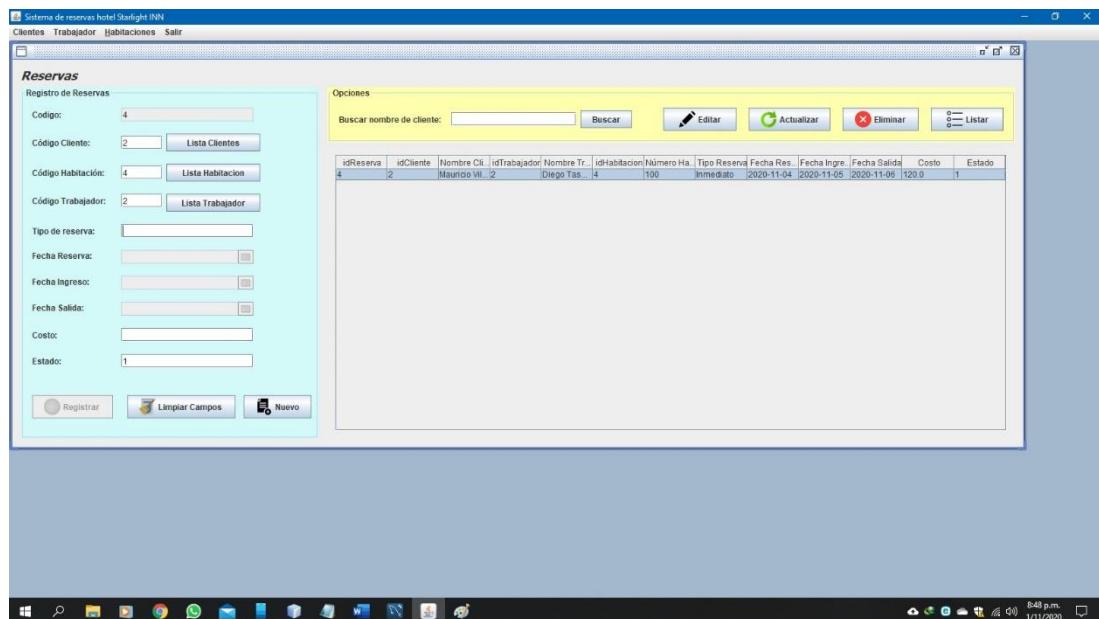


Ilustración 154: Modificar Reserva - Datos de entrada anómalo - Versión 2

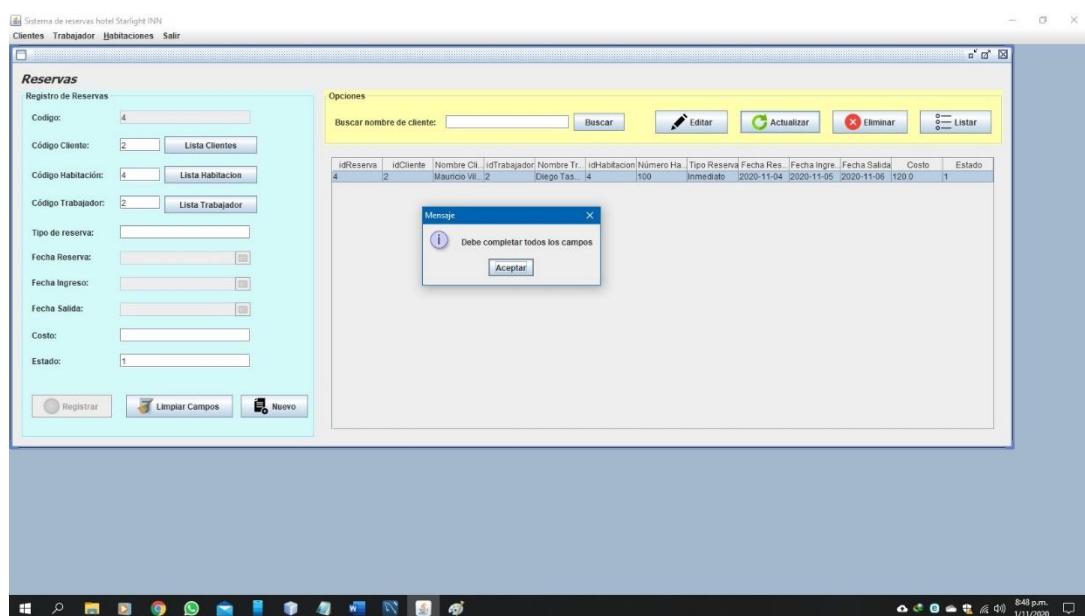


Ilustración 155: Modificar Reserva - Resultado esperado anómalo - Versión 2

## Caso de uso Consultar Reserva:

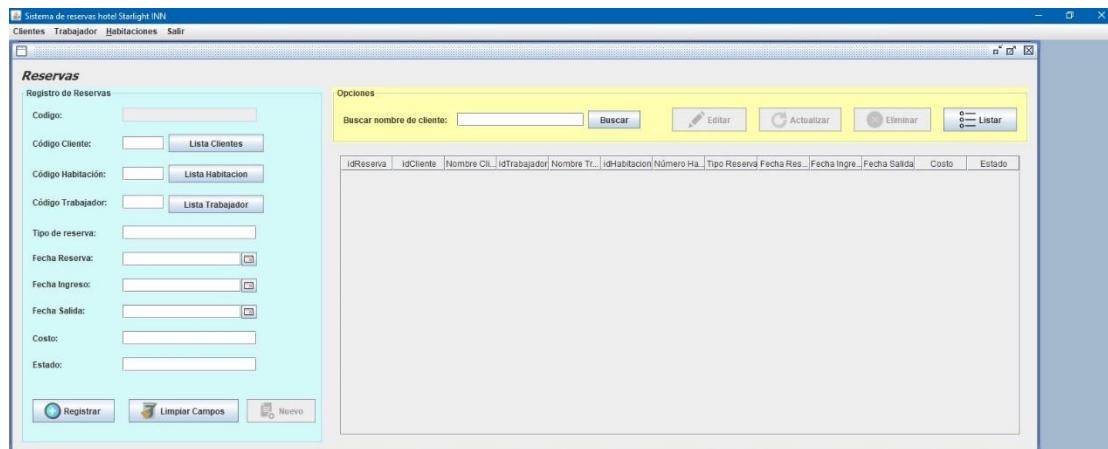


Ilustración 156: Consultar Reserva - Datos de entrada normal - Versión 2

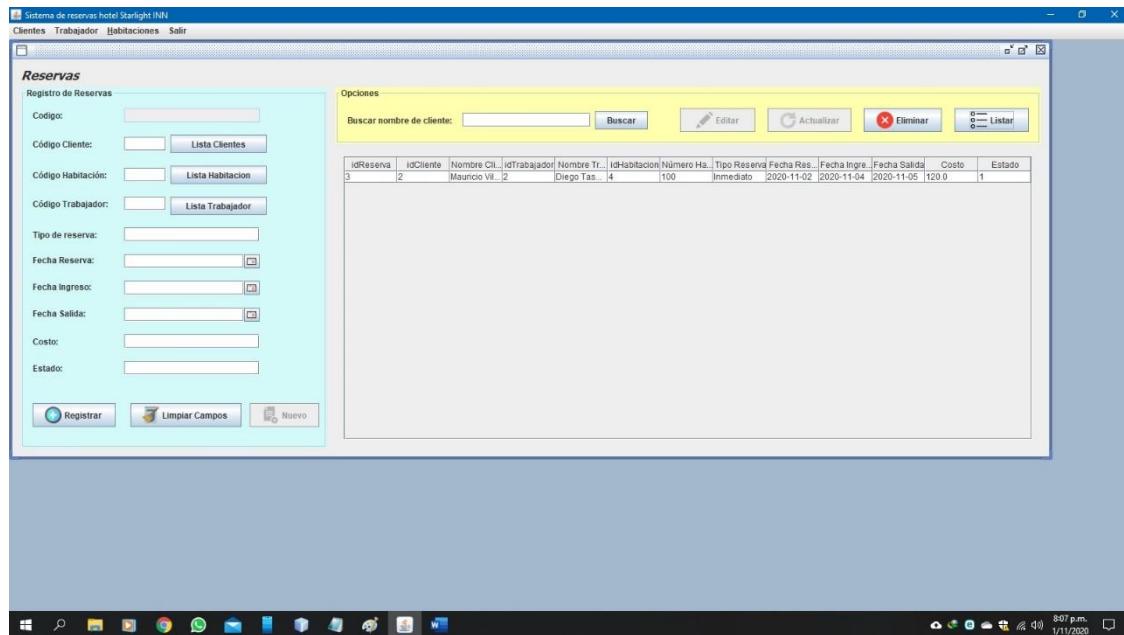


Ilustración 157: Consultar Reserva - Resultado esperado normal - Versión 2

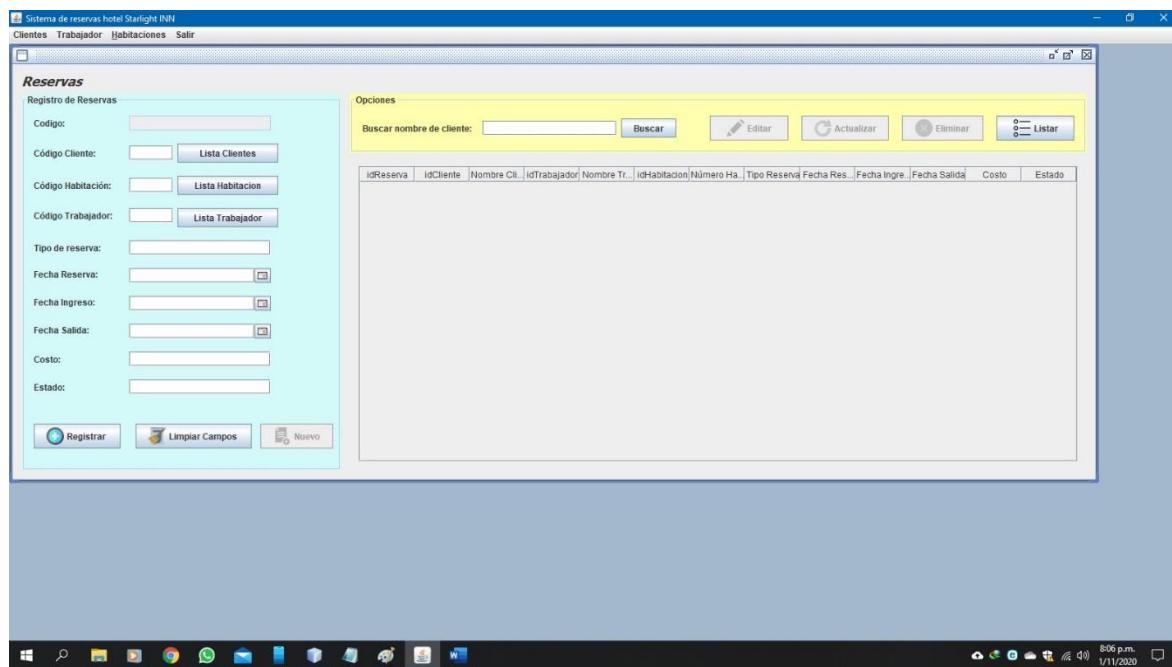


Ilustración 158: Consultar Reserva - Datos de entrada anómalo - Versión 2

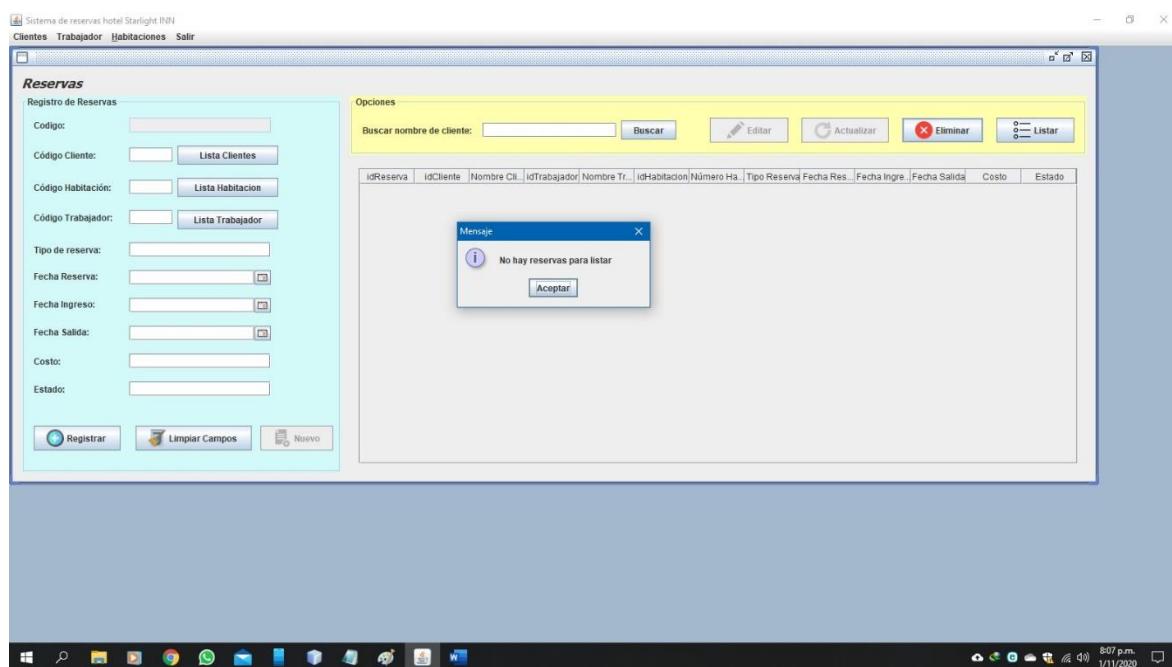


Ilustración 159: Consultar Reserva - Resultado esperado anómalo - Versión 2

## Hallazgos e Incidencias

VERSIÓN 1				
ENTIDAD	NRO	HALLAZGOS E INCIDENCIAS	RESPONSABLE	FECHA
Cliente	1	Tipo Documento acepta números cuando sólo debe aceptar texto.	Diego Tasaico	20/10/2020
	2	Número de Documento acepta letras cuando sólo debe aceptar números.	Diego Tasaico	20/10/2020
	3	Celular acepta letras cuando sólo debe aceptar números.	Mauricio Villanueva	20/10/2020
	4	Email se registra en la base de datos aun cuando no se ingresa el @.	Olenka Lazo	20/10/2020
	5	Celular acepta números negativos cuando no debería hacerlo	Abihail Zapata	20/10/2020
Trabajador	6	Cuando se ingresa letras en el campo Sueldo, no muestra mensaje de ayuda.	Olenka Lazo	20/10/2020
	7	Cuando se ingresa otro tipo de dato en Estado, no muestra mensaje de ayuda.	Abihail Zapata	24/10/2020
	8	Tipo de Documento acepta números cuando sólo debe aceptar texto.	Abihail Zapata	25/10/2020
	9	Número de Documento acepta letras cuando sólo debe aceptar números	Mauricio Villanueva	25/10/2020
	10	Celular acepta letras cuando sólo debe aceptar números	Diego Tasaico	27/10/2020
	11	Email se registra en la base de datos aun cuando no se ingresa el @	Olenka Lazo	28/10/2020
	12	Login duplicado si acepta el sistema cuando debería ser único.	Abihail Zapata	28/10/2020
	13	Password se registra con un sólo valor, cuando debería haber un mínimo de 4 caracteres	Mauricio Villanueva	28/10/2020
	14	Sueldo permite ingresar números negativos cuando no debería ser posible	Abihail Zapata	29/10/2020
	15	En el acceso permite ingresar cualquier tipo de dato, cuando sólo debería aceptar Administrador o Recepción	Diego Tasaico	29/10/2020
	16	Celular acepta números negativos cuando no debería hacerlo	Diego Tasaico	30/10/2020



Movimientos	17	Número de habitación permite ingresar letras cuando sólo debería aceptar números	Mauricio Villanueva	01/11/2020
	18	Número de habitación permite ingresar números negativos cuando sólo debería aceptar positivos	Olenka Lazo	01/11/2020
Login	19	Permite ingresar más de 20 caracteres en el usuario cuando debería existir un límite	Abihail Zapata	01/11/2020
	20	Permite ingresar más de 20 caracteres en la contraseña cuando debería existir un límite	Diego Tasaico	01/11/2020

Tabla 11: Hallazgos e Incidencias - Versión 1

VERSIÓN 2				
ENTIDAD	NRO	HALLAZGOS E INCIDENCIAS	RESPONSABLE	FECHA
Habitaciones	1	Permite colocar letras en Número de habitación cuando sólo debería aceptar números	Diego Tasaico	30/10/2020
	2	Permite colocar números negativos en Número de habitación y eso no debería ser posible.	Diego Tasaico	30/10/2020
	3	Permite ingresar un Número de habitación de menos de 2 dígitos y mayor a 4 dígitos cuando sólo debería ser de 3 dígitos exactamente	Mauricio Villanueva	30/10/2020
	4	Permite ingresar letras en Piso cuando sólo se debe ingresar números	Olenka Lazo	31/10/2020
	5	Permite ingresar números negativos en Piso cuando sólo debe aceptar números positivos	Abihail Zapata	31/10/2020
	6	Permite ingresar más de 2 dígitos en Piso cuando sólo debería aceptar números de 1 dígito.	Diego Tasaico	31/10/2020
	7	No permite ingresar números decimales en el Precio cuando sí debería aceptarlo	Abihail Zapata	31/10/2020
	8	Se pueden colocar letras en el Precio cuando sólo debe aceptar números	Olenka Lazo	31/10/2020
	9	Permite ingresar números negativos en el Precio cuando sólo debe aceptar números positivos	Olenka Lazo	01/11/2020

	10	Permite ingresar números en Tipo de habitación, cuando sólo debería aceptar letras	Mauricio Villanueva	01/11/2020
	11	Permite ingresar valores incorrectos en el Estado cuando sólo deben permitir valores específicos, se requiere una validación	Diego Tasaico	01/11/2020
Reservas	12	Permite ingresar letras en el Código Cliente cuando sólo debe aceptar números	Olenka Lazo	20/10/2020
	13	Permite ingresar letras en el Código Habitación cuando sólo debe aceptar números	Abihail Zapata	24/10/2020
	14	Permite ingresar letras en el Código Trabajador cuando sólo debe aceptar números	Abihail Zapata	25/10/2020
	15	Permite ingresar números en Tipo de reserva cuando sólo debe aceptar letras	Mauricio Villanueva	25/10/2020
	16	Permite ingresar Fecha Reserva incorrecta, se requiere una validación	Diego Tasaico	27/10/2020
	17	Permite ingresar Fecha Ingreso incorrecta, se requiere una validación	Olenka Lazo	28/10/2020
	18	Permite ingresar Fecha Salida incorrecta, se requiere una validación	Abihail Zapata	28/10/2020
	19	Permite ingresar letras en el Costo cuando sólo debe aceptar números	Mauricio Villanueva	28/10/2020
	20	Permite ingresar números negativos en el Costo cuando sólo debe aceptar números positivos	Abihail Zapata	29/10/2020
	21	No permite ingresar números decimales en el Costo cuando sí debería.	Diego Tasaico	29/10/2020
	22	Permite ingresar valores excesivamente altos en el Costo, debe existir un límite de dígitos.	Diego Tasaico	30/10/2020
	23	Permite ingresar valores incorrectos al Estado, se necesita verificación	Mauricio Villanueva	01/11/2020
	24	Permite ingresar números en la Búsqueda por nombre de cliente, se necesita verificación	Olenka Lazo	01/11/2020

Tabla 12: Hallazgos e Incidencias - Versión 2

**MODELO FURPS+ APLICADO AL PROYECTO AL 27/10/2020**

FACTOR	NRO	PREGUNTA	ARTEFACTO	SI CUMPLE	NO CUMPLE	ACCIÓN CORRECTIVA	RESPONSABLE	FECHA SOLUCIÓN	EVIDENCIA	Puntaje obtenido PO	Puntaje deseado PD	Grado Apego = (PO/PD) * 100%
Funcionalidad versión o incremento 1	1	¿El software ejecuta el caso de uso Registro de trabajador?	Software funcionando	X					Sección ejecución del plan de pruebas	60	85	70.59%
	2	¿El software ejecuta el caso de uso Modificar trabajador?	Software funcionando	X					Sección ejecución del plan de pruebas			
	3	¿El software ejecuta el caso de uso Eliminar trabajador?	Software funcionando	X					Sección ejecución del plan de pruebas			
	4	¿El software ejecuta el caso de uso Consulta de trabajadores?	Software funcionando	X					Sección ejecución del plan de pruebas			
	5	¿El software ejecuta el caso de uso Validar trabajador?	Software funcionando	X					Sección ejecución del plan de pruebas			
	6	¿El software ejecuta el caso de uso Registro de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			

7	¿El software ejecuta el caso de uso Modificar cliente?	Software funcionando	X					Sección ejecución del plan de pruebas	
8	¿El software ejecuta el caso de uso Eliminar cliente?	Software funcionando	X					Sección ejecución del plan de pruebas	
9	¿El software ejecuta el caso de uso Consulta de clientes?	Software funcionando	X					Sección ejecución del plan de pruebas	
10	¿El software ejecuta el caso de uso Registro de Entrada de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas	
11	¿El software ejecuta el caso de uso Eliminar la Entrada de cliente?	Software funcionando		X	Incluir caso de uso	Abihail Zapata	27/10/2020	Sección ejecución del plan de pruebas	
12	¿El software ejecuta el caso de uso Modificar la Entrada de cliente?	Software funcionando		X	Incluir caso de uso	Diego Tasaico	27/10/2020	Sección ejecución del plan de pruebas	
13	¿El software ejecuta el caso de uso Consultar la Entrada de cliente?	Software funcionando		X	Incluir caso de uso	Mauricio Villanueva	27/10/2020	Sección ejecución del plan de pruebas	

14	¿El software ejecuta el caso de uso Registro de Salida de cliente?	Software funcionando	X						Sección ejecución del plan de pruebas		
15	¿El software ejecuta el caso de uso Eliminar la Salida de cliente?	Software funcionando		X	Incluir caso de uso	Olenka Lazo	27/10/2020	Sección ejecución del plan de pruebas			
16	¿El software ejecuta el caso de uso Modificar Salida de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			
17	¿El software ejecuta el caso de uso Consultar la Salida de cliente?	Software funcionando		X	Incluir caso de uso	Mauricio Villanueva	27/10/2020	Sección ejecución del plan de pruebas			
Funcionalidad versión o incremento 2	18	¿El software ejecuta el caso de uso Registro de reserva?	Código fuente		X	Incluir caso de uso	Diego Tasaico	28/10/2020	Sección ejecución del plan de pruebas	0	40
	19	¿El software ejecuta el caso de uso Modificar reserva?	Código fuente		X	Incluir caso de uso	Diego Tasaico	28/10/2020	Sección ejecución del plan de pruebas		
	20	¿El software ejecuta el caso de uso Eliminar reserva?	Código fuente		X	Incluir caso de uso	Mauricio Villanueva	28/10/2020	Sección ejecución del plan de pruebas		
	21	¿El software ejecuta el caso de uso Consultar reserva?	Código fuente		X	Incluir caso de uso	Olenka Lazo	29/10/2020	Sección ejecución del plan de pruebas		

22	¿El software ejecuta el caso de uso Registro de habitación?	Código fuente		X	Incluir caso de uso	Olenka Lazo	29/10/2020	Sección ejecución del plan de pruebas		
23	¿El software ejecuta el caso de uso Modificar habitación?	Código fuente		X	Incluir caso de uso	Mauricio Villanueva	30/10/2020	Sección ejecución del plan de pruebas		
24	¿El software ejecuta el caso de uso Eliminar habitación?	Código fuente		X	Incluir caso de uso	Abihail Zapata	30/10/2020	Sección ejecución del plan de pruebas		
25	¿El software ejecuta el caso de uso Consultar habitación?	Código fuente		X	Incluir caso de uso	Abihail Zapata	30/10/2020	Sección ejecución del plan de pruebas		
Usabilidad	21	¿Se muestran las imágenes de la carpeta de software y de informe en el GitHub?	GitHub	X				Sección repositorio GitHub	20	25
	22	¿Si los matices de color de texto y de fondo son cómodos para el usuario?	Software funcionando	X				Sección funcionamiento del software		
	23	¿Es de fácil uso para el usuario?	Software funcionando	X				Sección funcionamiento del software		

Confiabilidad	24	¿Permite una fácil salida del software?	Software funcionando		X	Corregir la salida del sistema	Mauricio Villanueva	31/10/2020	Sección funcionamiento de software		
	25	¿Se adapta al tipo de usuario?	Software funcionando	X					Sección funcionamiento del software		
	26	¿Si el software controla el acceso no disponible de la Base de datos?	Software funcionando		X	Controlar el acceso a Base de Datos si no se conecta.	Mauricio Villanueva	31/10/2020	Sección funcionamiento de software	5	25
	27	¿Es de uso seguro para los usuarios?	Software funcionando		X	Corregir seguridad del sistema	Mauricio Villanueva	31/10/2020	Sección funcionamiento de software		
	28	¿Es resistente ante fallos?	Software funcionando		X	Corregir la resistencia de fallos del sistema	Diego Tasaico	31/10/2020	Sección funcionamiento de software		
Rendimiento	29	¿Todas las partes del sistema se encuentran conectadas entre sí?	Software funcionando	X					Sección funcionamiento del software		
	30	¿Tiene un comportamiento adecuado ante situaciones inusuales?	Software funcionando		X	Corregir la respuesta ante situaciones inusuales	Neftali Zapata	31/10/2020	Sección funcionamiento de software		
Rendimiento	31	¿Si el tiempo de respuesta es menor a 7 segundos?	Software funcionando	X					Sección funcionamiento del software	15	25

32	¿Si el software ocupa espacio de memoria mínima?	Administrador de tareas del SO		X	Corrección de espacio de memoria	Neftali Zapata	31/10/2020	Sección de funcionamiento de software			
33	¿Disponible las 24 horas del día?	Software funcionando	X					Sección funcionamiento del software			
34	¿Soluciona errores?	Software funcionando		X	Mejorar la solución de errores	Olenka Lazo	31/10/2]20	Sección funcionamiento de software			
35	¿Detecta errores?	Software funcionando	X					Sección funcionamiento del software			
Soporte	36	¿Si el software cuenta con un procedimiento de compilación e instalación?	Procedimiento		X	Completar procedimiento	Olenka Lazo	31/10/2020	Sección procedimientos	0	25
	37	¿Cuenta con un resguardo de datos?	Software funcionando		X	Crear la opción de resguardo de datos	Diego Tasaico	31/10/2020	Sección funcionamiento de software		
	38	¿Puede ejecutarse ante alguna condición extrema?	Software funcionando		X	Corregir al ejecutar	Olenka Lazo	31/10/2020	Sección funcionamiento de software		
	39	¿Detecta errores en el sistema de manera rápida?	Software funcionando		X	Corregir el tiempo de solución	Neftali Zapata	31/10/2020	Sección funcionamiento de software		

	40	¿Cuenta con una opción de ayuda al usuario?	Software funcionando		X	Implementar opción de ayuda	Diego Tasaico	31/10/2020	Sección funcionamiento de software			
PLUS (+)	41	¿Se ha completado el marco teórico con citas y referencias?	Informe		X	Completar el marco teórico con citas y referencias	Olenka Zapata	01/11/2020	Sección marco teórico del informe	5	20	25%
	42	¿Se ha completado el plan de pruebas de la versión 1 y 2?	Informe	X					Sección plan de pruebas incremental			
	43	¿Se ha completado la ejecución de pruebas de la versión 1 y 2?	Informe		X	Falta completar plan de pruebas en versión 2	Mauricio Villanueva	01/11/2020	Sección ejecución de plan de pruebas			
	44	¿Se ha completado las conclusiones y recomendaciones de la versión 2?	Informe		X	Completar conclusiones y recomendaciones de la versión 2	Abihail Zapata	01/11/2020	Sección conclusiones y recomendaciones			

Tabla 13: Modelo FURPS - 1

**MODELO FURPS+ APLICADO AL PROYECTO AL 01/11/2020**

FACTOR	NRO	PREGUNTA	ARTEFACTO	SI CUMPLE	NO CUMPLE	ACCIÓN CORRECTIVA	RESPONSABLE	FECHA SOLUCIÓN	EVIDENCIA	Puntaje obtenido PO	Puntaje deseado PD	Grado Apego = (PO/PD) * 100%
Funcionalidad versión o incremento 1	1	¿El software ejecuta el caso de uso Registro de trabajador?	Software funcionando	X					Sección ejecución del plan de pruebas	85	85	100%
	2	¿El software ejecuta el caso de uso Modificar trabajador?	Software funcionando	X					Sección ejecución del plan de pruebas			
	3	¿El software ejecuta el caso de uso Eliminar trabajador?	Software funcionando	X					Sección ejecución del plan de pruebas			
	4	¿El software ejecuta el caso de uso Consulta de trabajadores?	Software funcionando	X					Sección ejecución del plan de pruebas			
	5	¿El software ejecuta el caso de uso Validar trabajador?	Software funcionando	X					Sección ejecución del plan de pruebas			
	6	¿El software ejecuta el caso de uso Registro de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			

7	¿El software ejecuta el caso de uso Modificar cliente?	Software funcionando	X					Sección ejecución del plan de pruebas	
8	¿El software ejecuta el caso de uso Eliminar cliente?	Software funcionando	X					Sección ejecución del plan de pruebas	
9	¿El software ejecuta el caso de uso Consulta de clientes?	Software funcionando	X					Sección ejecución del plan de pruebas	
10	¿El software ejecuta el caso de uso Registro de Entrada de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas	
11	¿El software ejecuta el caso de uso Eliminar la Entrada de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas	
12	¿El software ejecuta el caso de uso Modificar la Entrada de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas	
13	¿El software ejecuta el caso de uso Consultar la Entrada de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas	

14	¿El software ejecuta el caso de uso Registro de Salida de cliente?	Software funcionando	X						Sección ejecución del plan de pruebas		
15	¿El software ejecuta el caso de uso Eliminar la Salida de cliente?	Software funcionando	X						Sección ejecución del plan de pruebas		
16	¿El software ejecuta el caso de uso Modificar Salida de cliente?	Software funcionando	X						Sección ejecución del plan de pruebas		
17	¿El software ejecuta el caso de uso Consultar la Salida de cliente?	Software funcionando	X						Sección ejecución del plan de pruebas		
Funcionalidad versión o incremento 2	18	¿El software ejecuta el caso de uso Registro de reserva?	Código fuente	X					Sección ejecución del plan de pruebas	40	40 100%
	19	¿El software ejecuta el caso de uso Modificar reserva?	Código fuente	X					Sección ejecución del plan de pruebas		
	20	¿El software ejecuta el caso de uso Eliminar reserva?	Código fuente	X					Sección ejecución del plan de pruebas		
	21	¿El software ejecuta el caso de uso Consultar reserva?	Código fuente	X					Sección ejecución del plan de pruebas		

22	¿El software ejecuta el caso de uso Registro de habitación?	Código fuente	X						Sección ejecución del plan de pruebas		
23	¿El software ejecuta el caso de uso Modificar habitación?	Código fuente	X						Sección ejecución del plan de pruebas		
24	¿El software ejecuta el caso de uso Eliminar habitación?	Código fuente	X						Sección ejecución del plan de pruebas		
25	¿El software ejecuta el caso de uso Consultar habitación?	Código fuente	X						Sección ejecución del plan de pruebas		
Usabilidad	21	¿Se muestra las imágenes de la carpeta de software y de informe en el GitHub?	GitHub	X					Sección repositorio GitHub	25	25
	22	¿Si los matices de color de texto y de fondo son cómodos para el usuario?	Software funcionando	X					Sección funcionamiento del software		
	23	¿Es de fácil uso para el usuario?	Software funcionando	X					Sección funcionamiento del software		

	24	¿Permite una fácil salida del software?	Software funcionando	X					Sección funcionamiento de software			
	25	¿Se adapta al tipo de usuario?	Software funcionando	X					Sección funcionamiento del software			
Confiabilidad	26	¿Si el software controla el acceso no disponible de la Base de datos?	Software funcionando	X					Sección funcionamiento de software	25	25	100%
	27	¿Es de uso seguro para los usuarios?	Software funcionando	X					Sección funcionamiento de software			
	28	¿Es resistente ante fallos?	Software funcionando	X					Sección funcionamiento de software			
	29	¿Todas las partes del sistema se encuentran conectadas entre sí?	Software funcionando	X					Sección funcionamiento del software			
	30	¿Tiene un comportamiento adecuado ante situaciones inusuales?	Software funcionando	X					Sección funcionamiento de software			
Rendimiento	31	¿Si el tiempo de respuesta es menor a 7 segundos?	Software funcionando	X					Sección funcionamiento del software	25	25	100%

32	¿Si el software ocupa espacio de memoria mínima?	Administrador de tareas del SO	X						Sección de funcionamiento de software		
33	¿Disponible las 24 horas del día?	Software funcionando	X						Sección funcionamiento del software		
34	¿Soluciona errores?	Software funcionando	X						Sección funcionamiento de software		
35	¿Detecta errores?	Software funcionando	X						Sección funcionamiento del software		
Soporte	36	¿Si el software cuenta con un procedimiento de compilación e instalación?	Software funcionando	X					Sección funcionamiento del software	25	25
	37	¿Cuenta con un resguardo de datos?	Software funcionando	X					Sección funcionamiento de software		
	38	¿Puede ejecutarse ante alguna condición extrema?	Software funcionando	X					Sección funcionamiento de software		
	39	¿Detecta errores en el sistema de manera rápida?	Software funcionando	X					Sección funcionamiento de software		

	40	¿Cuenta con una opción de ayuda al usuario?	Software funcionando	X					Sección funcionamiento de software			
PLUS (+)	41	¿Se ha completado el marco teórico con citas y referencias?	Informe	X					Sección marco teórico del informe	20	20	100%
	42	¿Se ha completado el plan de pruebas de la versión 1 y 2?	Informe	X					Sección plan de pruebas incremental			
	43	¿Se ha completado la ejecución de pruebas de la versión 1 y 2?	Informe	X					Sección ejecución de plan de pruebas			
	44	¿Se ha completado las conclusiones y recomendaciones de la versión 2?	Informe	X					Sección conclusiones y recomendaciones			

Tabla 14: Modelo FURPS - 2

## CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES

### 12. Conclusiones

- Se presentó a la empresa una solicitud para obtener el permiso respectivo, mencionándole el objetivo del estudio. Una vez aceptado el documento se coordinó con el administrador, para la recolección de datos con el fin de obtener las facilidades pertinentes para llevar a cabo el estudio.
- La recopilación de información permitió comprender mejor el sistema, que después fue representado a través de los diagramas utilizados. De dicho análisis, del cual se pudo extraer los requerimientos del sistema que sirven de base para el desarrollo de este.
- Se desarrolló el sistema utilizando NetBeans IDE 8.2 como lenguaje de programación y SQL server para la gestión de la base de datos, logrando la automatización eficiente del sistema.
- Se utilizó adicionalmente, las herramientas informáticas: Lucidchart, MySQL WorkBench y Balsamiq que nos permitió realizar los diagramas y prototipos, logrando así un mejor desarrollo y conceptualización básica que requiere el sistema.
- Por último, se cumplió con el objetivo del sistema, permitiendo así la implementación de sus respectivas opciones entre ellas mantenimiento de clientes, donde se podrá registrar, consultar, modificar y eliminar a los clientes, incluyendo la entrada y salida del cliente.
- Esta versión del sistema permite la administración de los procesos de registro de habitaciones y reservas, además del control de hospedaje de los huéspedes durante su estadía, hasta la salida de las instalaciones del hotel.
- El diseño de la base de datos asegura la integridad y confidencialidad de los datos, además de la centralización de la información. Asimismo, el diseño de interfaz permite una visualización previa de las pantallas de los nuevos registros que se van a utilizar en el sistema.
- El sistema desarrollado se ajusta a las diversas formas existentes para controlar el hospedaje y reservación, esto gracias a los parámetros con los que cuenta el sistema.
- Los nuevos casos de uso permitieron adaptar de mejor manera el desarrollo del sistema con una serie de requerimientos los cuales fueron creciendo mediante se iba avanzando en la elaboración del sistema.

- Para el funcionamiento del sistema se llevó a cabo la ejecución de planes de prueba, los cuales tuvo como finalidad poder encontrar las debilidades del sistema y a su vez poder reforzarlos.

### 13. Recomendaciones

- Se recomienda continuar la realización de los próximos prototipos en la aplicación Balsamiq ya que nos permite comprender de manera visual el proyecto y explorar más opciones.
- Para seguir mejorando el sistema de administración hotelera se ha recomendado seguir solicitando información a la empresa para así culminar los objetivos proyectados.
- Al ser el presente proyecto, un sistema desarrollado a base del mantenimiento de registro de clientes y trabajadores, se recomienda la implementación del mantenimiento de las habitaciones y reservas para un mejor manejo.
- Se recomienda usar el mensaje de ayuda con el objetivo de resolver algún inconveniente que se presente.
- Y finalmente, se hará un énfasis a la revisión constante de las funciones y procesos, debido a que existirán cambios en cada una de las funcionalidades a fin de así poder mejorar el desempeño del sistema.
- Se recomienda efectuar los respaldos del sistema periódicamente con el fin de disponer de un medio para recuperación de datos inscritos por última vez, en caso de que haya ocurrido un problema o perdida.
- Es recomendable seguir investigando de manera continua para así ir actualizando la información del sistema.
- La optimización de procesos y mejora continua es indispensable para reducir fallos en las áreas operativas del sistema.
- Es recomendable observar la calidad del sistema, si se realiza un uso correcto y el grado de satisfacción para asegurar el resultado que tiene este en el rendimiento.
- Por último, para seguir mejorando su productividad es recomendable implementar un sistema de reporte para la versión 3.

## 14. Referencias Bibliográficas

- Alonso, F., Martínez, L., & Segovia , J. (2005). *Introducción a la ingeniería de Software*. Zaragoza : Delta Publicaciones.
- Git Hub*. (s.f.). Obtenido de <https://github.com/>
- Lucidchart*. (s.f.). Obtenido de <https://www.lucidchart.com/pages/es>
- Modelo de Calidad Furps. (2018). *FANDOM*.
- MySQL*. (s.f.). Obtenido de <https://www.mysql.com/>
- MySQL WorkBench*. (s.f.). Obtenido de <https://www.mysql.com/products/workbench/>
- NetBeans*. (s.f.). Obtenido de <https://netbeans.org/>
- Sommerville, I. (2011). *Ingeniería de software* (9a. ed.). Pearson Educación.
- Toledo, F. (s.f.). Como armas un plan de pruebas performance? *Abstracta*.
- WampServer*. (s.f.). Obtenido de <https://www.wampserver.com/en/>

## 15. Anexos



Ilustración 160: Videoconferencia con representante de la empresa

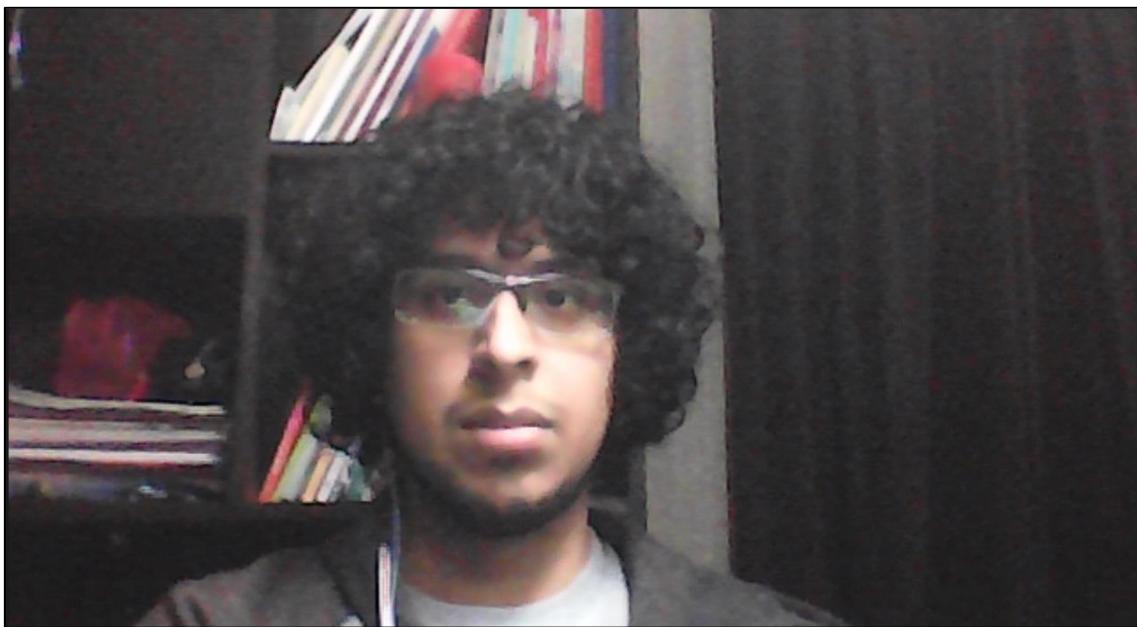


Ilustración 161: Integrante - Diego Tasaico



Ilustración 162: Integrante - Mauricio Villanueva



Ilustración 163: Integrante - Olenka Lazo



Ilustración 164: Integrante - Abihail Zapata



Ilustración 165: Acta de compromiso

## Funcionamiento del Software:

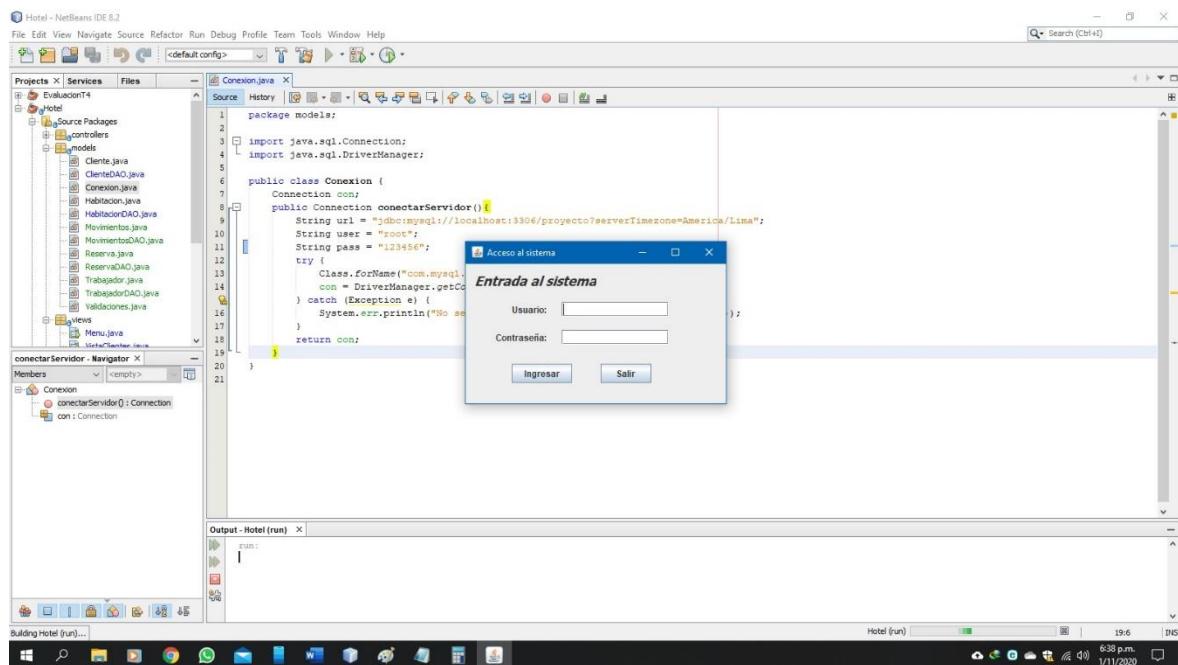


Ilustración 166: Funcionamiento del Software - Acceso al sistema

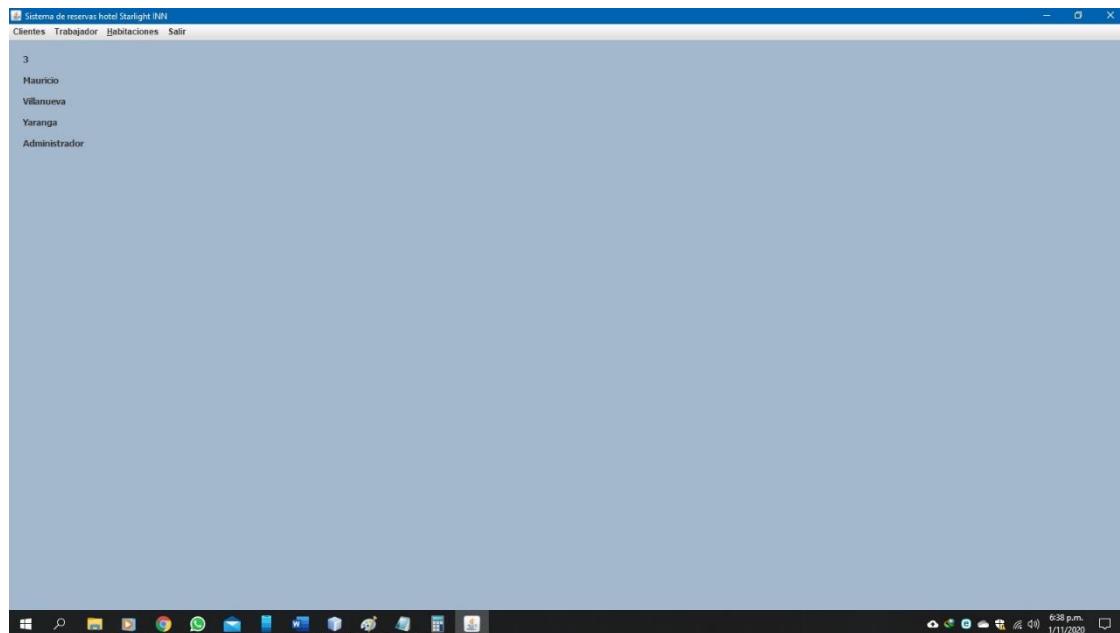


Ilustración 167: Funcionamiento del Software - Menú Principal

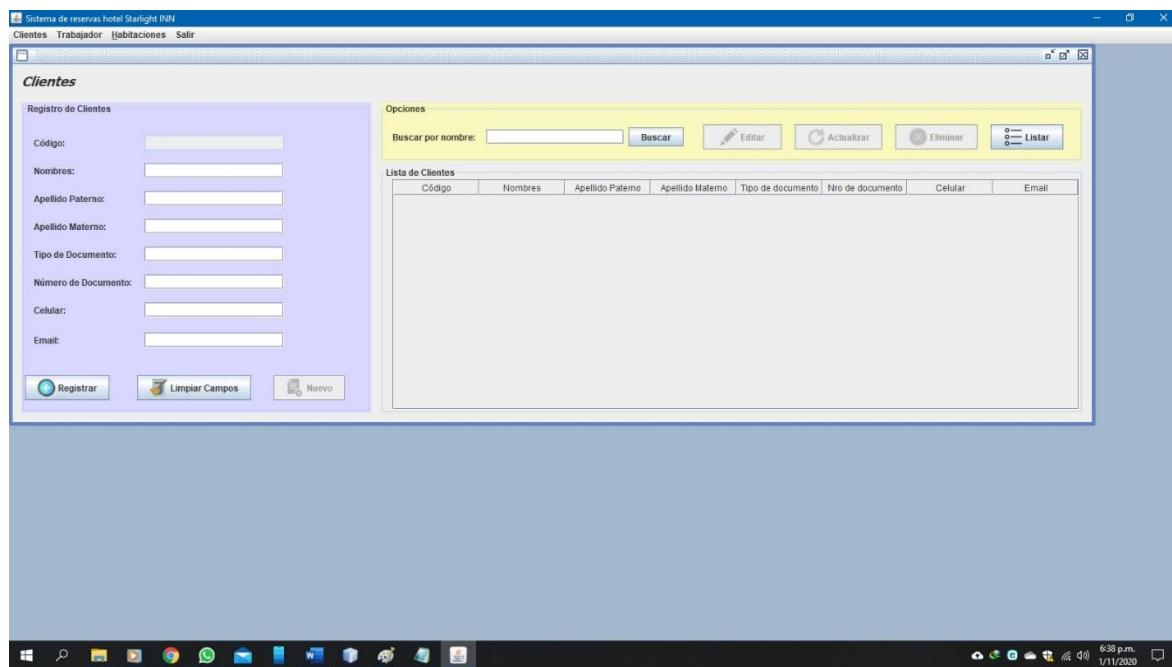


Ilustración 168: Funcionamiento del Software - Vista de los clientes

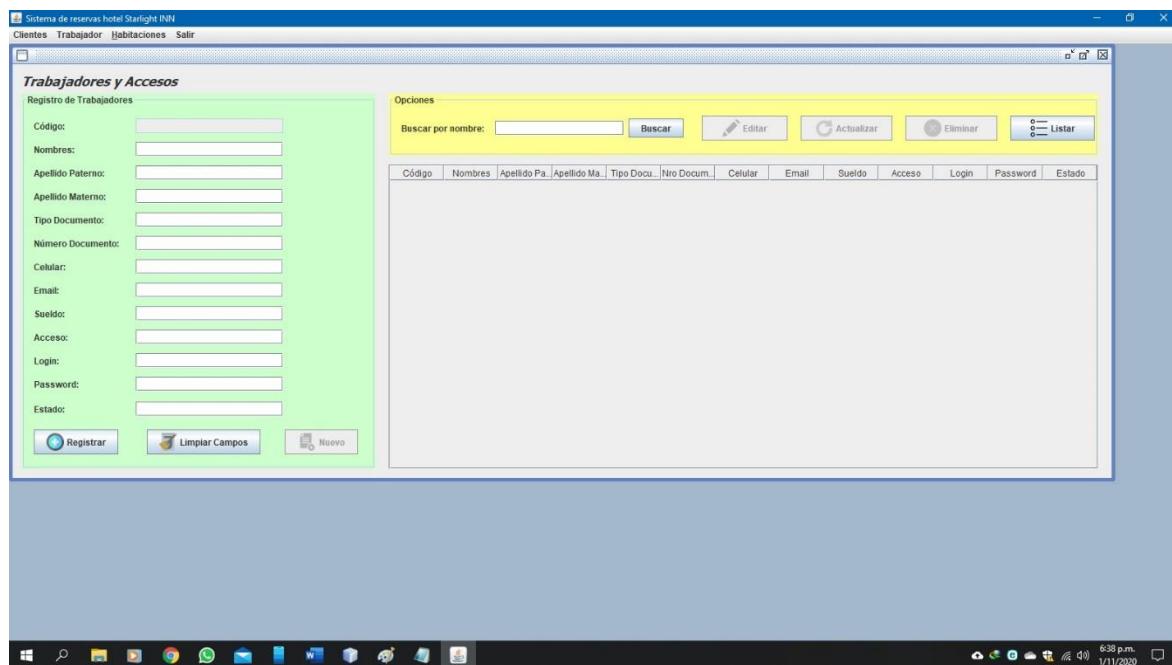


Ilustración 169: Funcionamiento del Software - Vista de los trabajadores

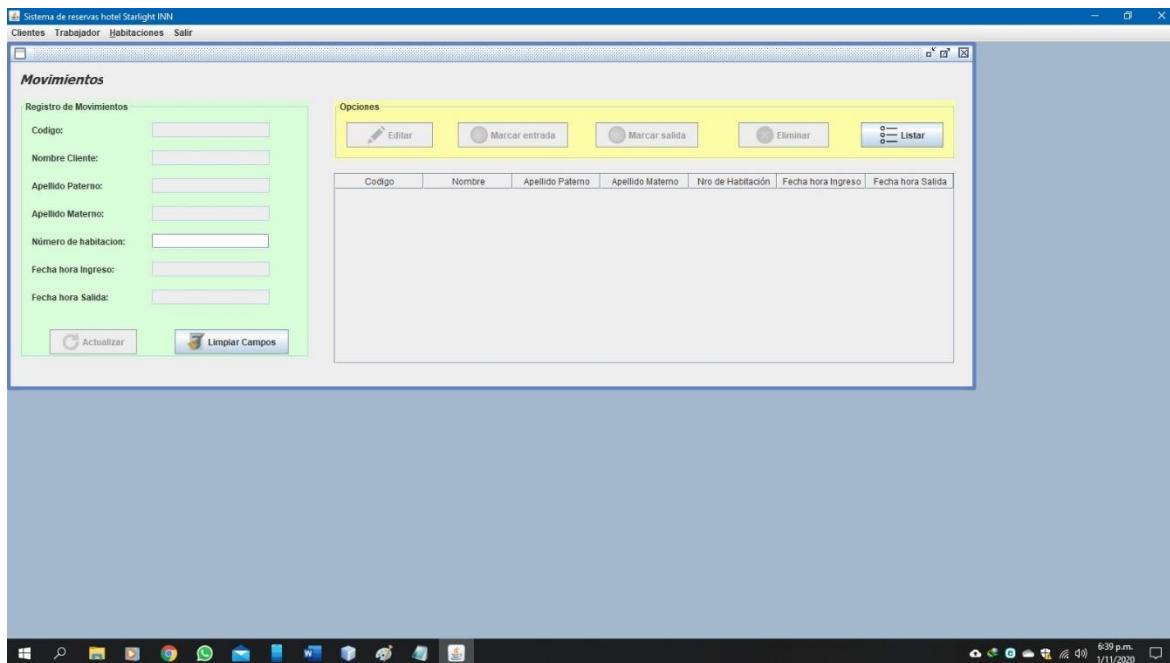


Ilustración 170: Funcionamiento del Software - Vista de los movimientos

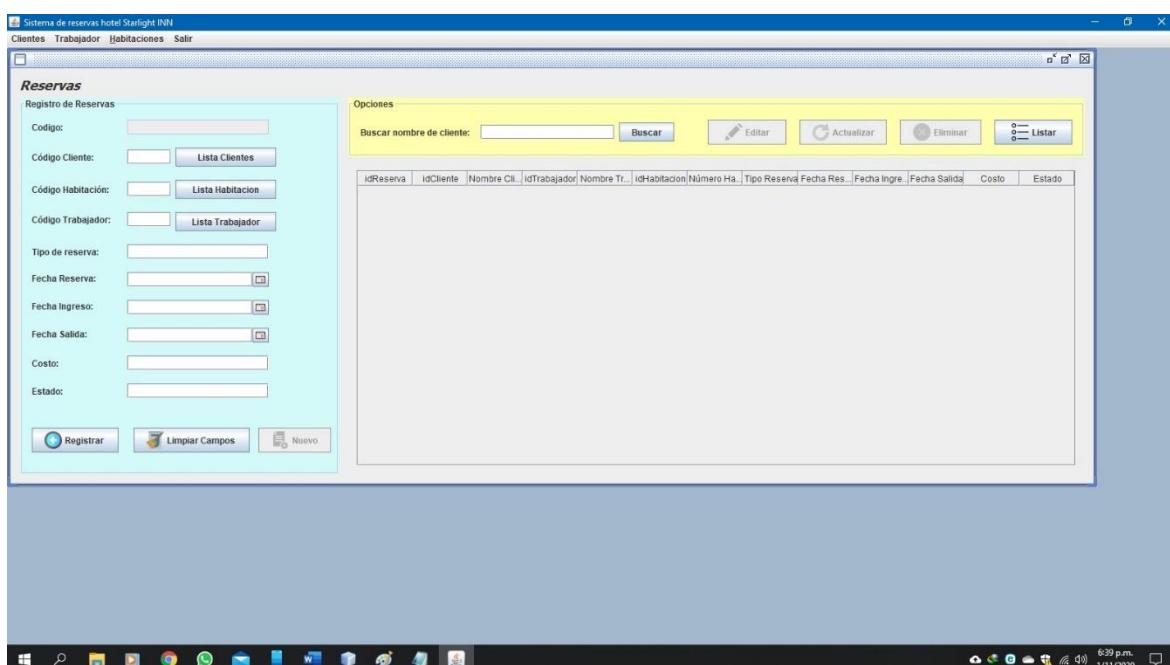


Ilustración 171: Funcionamiento del Software - Vista de las reservas

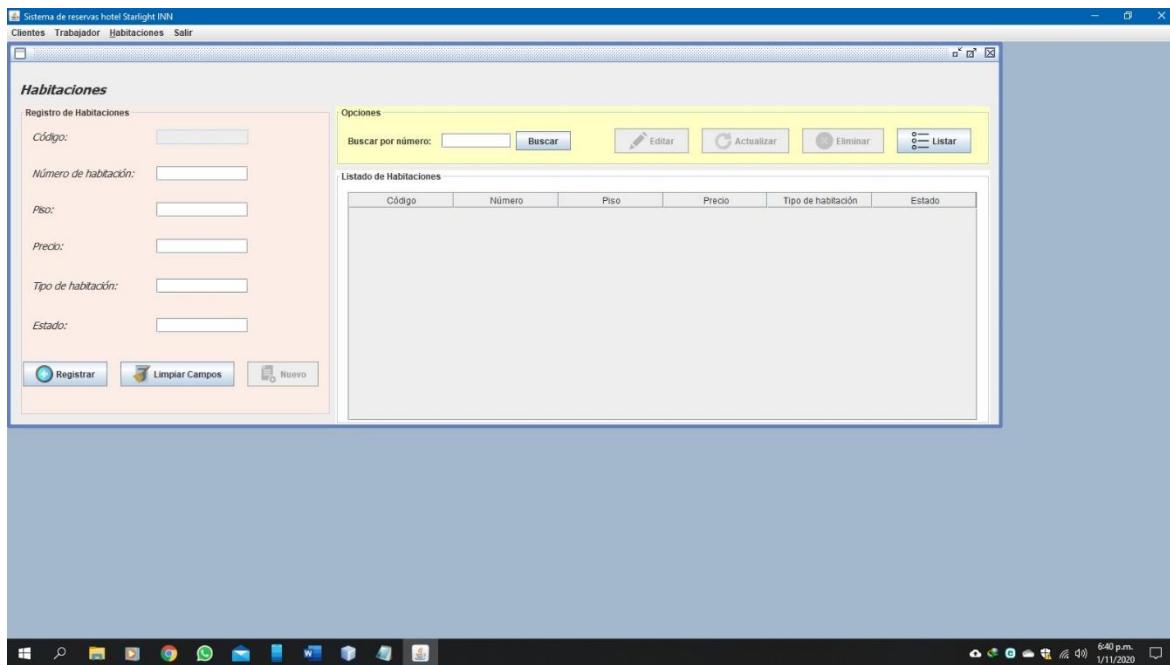


Ilustración 172: Funcionamiento del Software - Vista de las habitaciones

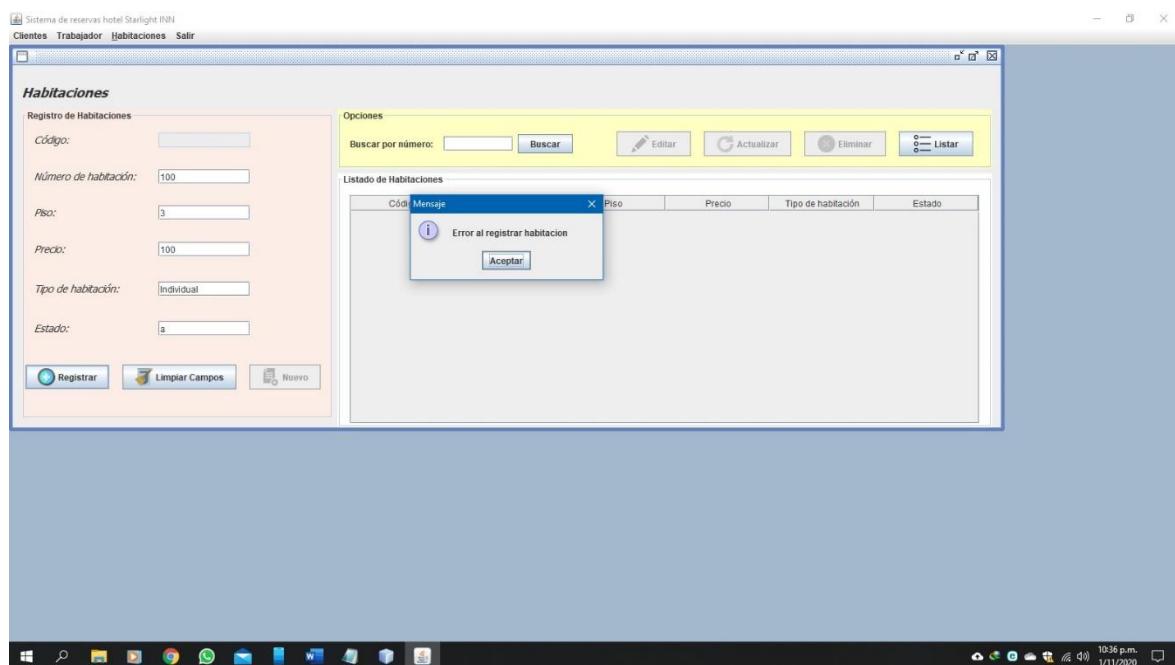


Ilustración 173: Funcionamiento del Software - Detección de errores

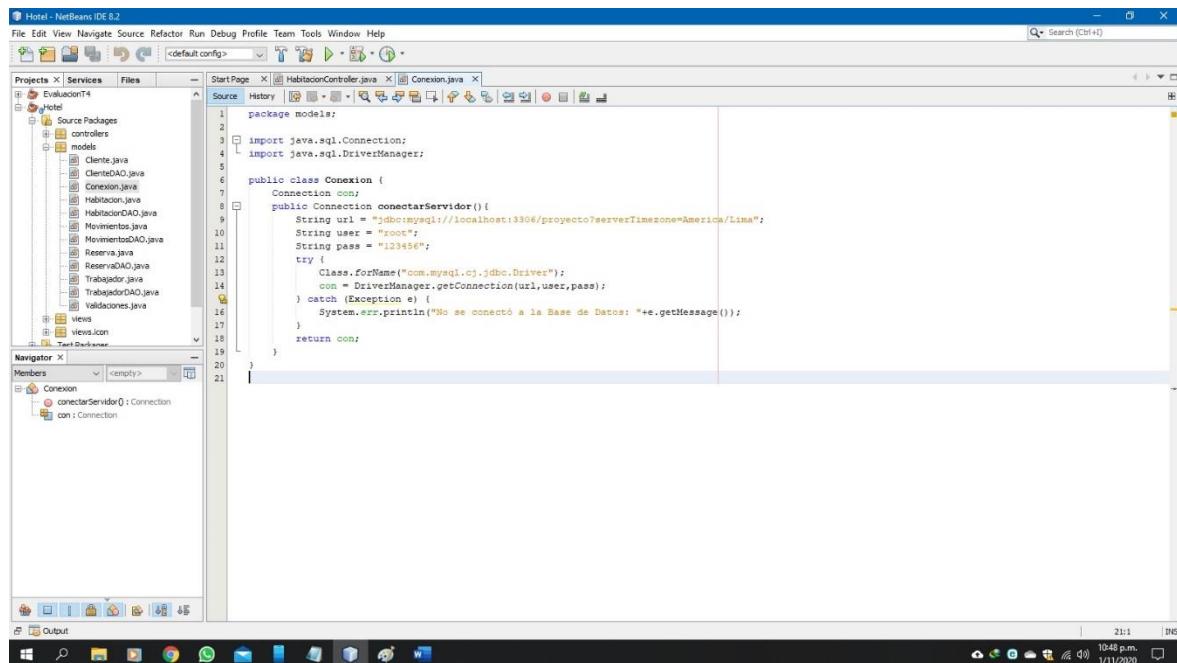


Ilustración 174: Funcionamiento del Software - Conexión a Base de Datos

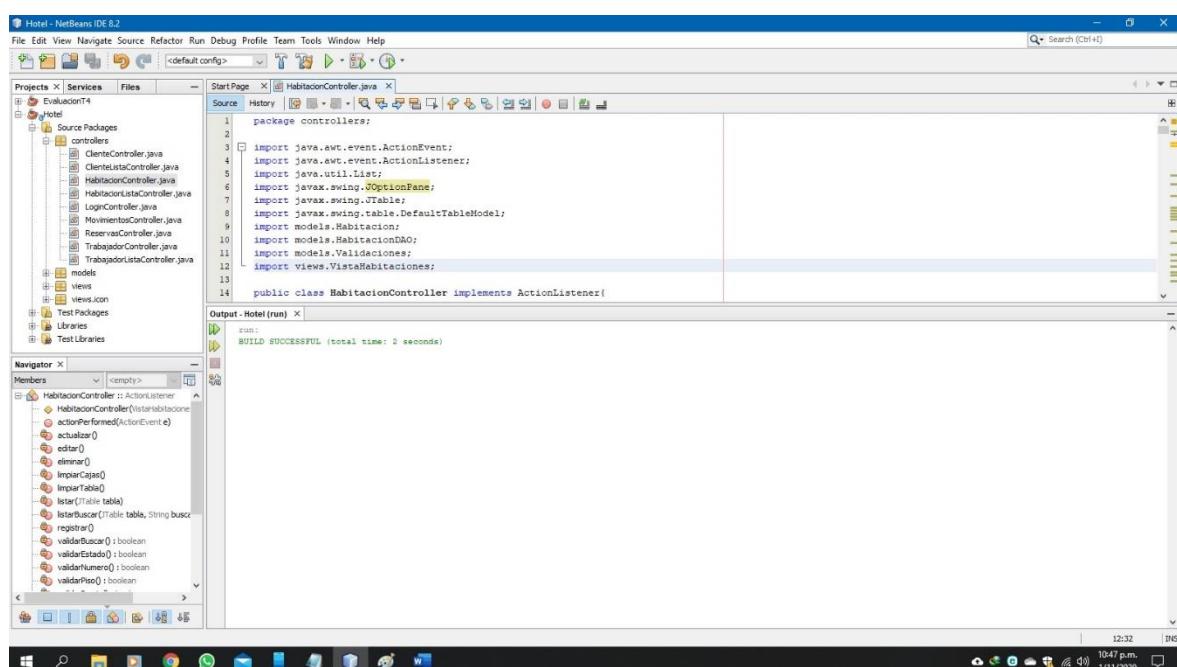


Ilustración 175: Funcionamiento del Software - Tiempo de respuesta

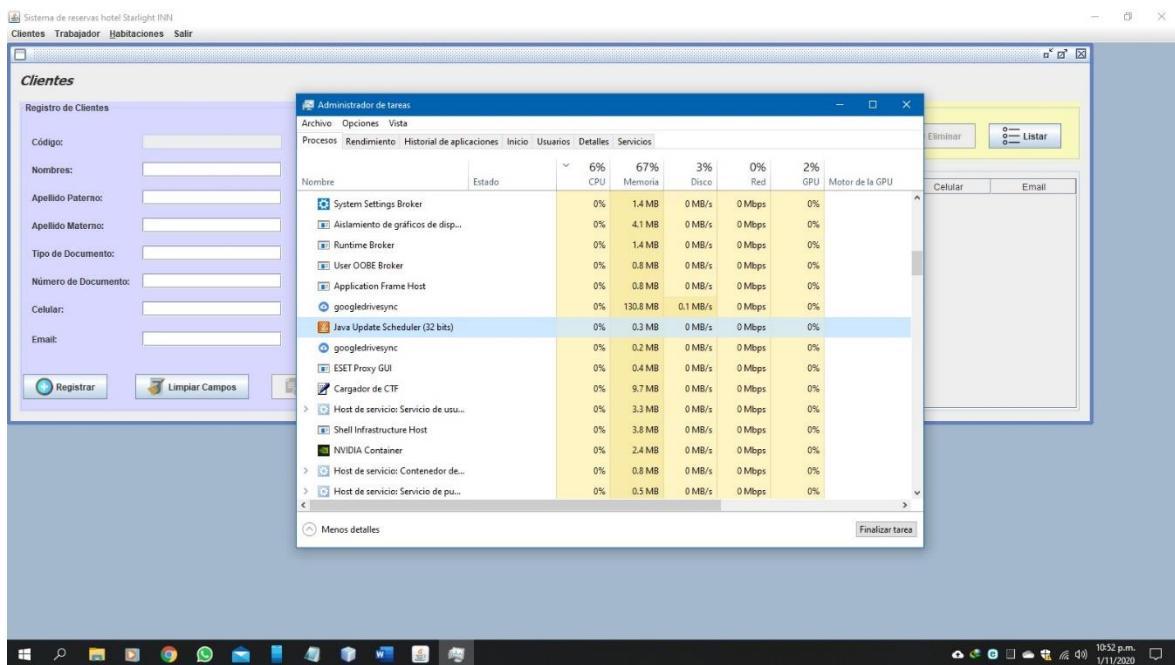


Ilustración 176: Funcionamiento del Software - Espacio mínimo de memoria