UNIVERSIDAD PRIVADA DEL NORTE

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

CAPÍTULO 1. INTRODUCCIÓN 1
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

CAPÍTULO 2. MARCO TEÓRICO 2
9. Marco teórico 2
10. Marco metodológico 4

CAPÍTULO 3. DESARROLLO DE LA INVESTIGACIÓN 5
11. Implementación de la solución planteada 5

CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES 208
12. Conclusiones 208
13. Recomendaciones 209
14. Referencias Bibliográficas 210
15. Anexos 210

ÍNDICE DE TABLAS

Tabla 1: Proceso de Negocio 5

Tabla 2: Trazabilidad de Requisitos 13

Tabla 3: Especificación de Casos de uso - Versión 1 - 1.....52

Tabla 4: Especificación de Casos de uso - Versión 1 - 2.....54

Tabla 5: Especificación de Casos de uso - Versión 1 - 3.....56

Tabla 6: Especificación de Casos de uso - Versión 1 - 4.....58

Tabla 7: Especificación de Casos de uso - Versión 2 - 1.....60

Tabla 8: Especificación de Casos de uso - Versión 2 - 2.....62

Tabla 9: Plan de pruebas Incremental - Versión 1.....65

Tabla 10: Plan de pruebas Incremental - Versión 2.....66

Tabla 11: Hallazgos e Incidencias - Versión 1 118

Tabla 12: Hallazgos e Incidencias - Versión 2 119

Tabla 13: Modelo FURPS - 1.....126

Tabla 14: Modelo FURPS - 2.....133

ÍNDICE DE FIGURAS

Ilustración 1: Proceso de Negocio.....5

Ilustración 2: Proceso de Software.....6

Ilustración 3: Diagrama de Casos de uso 7

Ilustración 4: Prototipo - Versión 1 - 1.....7

Ilustración 5: Prototipo - Versión 1 - 2.....8

Ilustración 6: Prototipo - Versión 1 - 3.....8

Ilustración 7: Prototipo - Versión 1 - 4.....9

Ilustración 8: Prototipo - Versión 2 - 1.....9

Ilustración 9: Prototipo - Versión 2 - 2.....10

Ilustración 10: Diagrama de actividades con particiones o calles.....11

Ilustración 11: Requisitos no funcionales - Diego Tasaico - 1.....14

Ilustración 12: Requisitos no funcionales - Diego Tasaico - 2.....14

Ilustración 13: Requisitos no funcionales - Diego Tasaico - 3.....15

Ilustración 14: Requisitos no funcionales - Diego Tasaico - 4.....15

Ilustración 15: Requisitos no funcionales - Mauricio Villanueva - 1.....16

Ilustración 16: Requisitos no funcionales - Mauricio Villanueva - 2.....16

Ilustración 17: Requisitos no funcionales - Mauricio Villanueva - 3.....17

Ilustración 18: Requisitos no funcionales - Abihail Zapata - 1.....17

Ilustración 19: Requisitos no funcionales - Abihail Zapata - 2.....18

Ilustración 20: Requisitos no funcionales - Abihail Zapata - 3.....18

Ilustración 21: Requisitos no funcionales - Abihail Zapata - 4.....19

Ilustración 22: Requisitos no funcionales - Olenka Lazo - 1.....19

Ilustración 23: Requisitos no funcionales - Olenka Lazo - 2.....20

Ilustración 24: Requisitos no funcionales - Olenka Lazo - 3.....20

Ilustración 25: Requisitos no funcionales - Olenka Lazo - 4.....21

Ilustración 26: Repositorio GitHub 21

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: Diagrama de secuencia – Registrar cliente 134

Ilustración 161: Vista Cliente - 1 135

Ilustración 162: Vista Cliente - 2 135

Ilustración 163: Vista Cliente - 3 136

Ilustración 164: Vista Cliente - 4 136

Ilustración 165: Vista Cliente - 5 137

Ilustración 166: Vista Cliente - 6 137

Ilustración 167: Vista Cliente - 7 ..... **138**

Ilustración 168: Vista Cliente - 8 ..... **138**

Ilustración 169: Vista Cliente - 9 ..... **139**

Ilustración 170: Vista Cliente - 10 ..... **139**

Ilustración 171: Vista Cliente - 11 ..... **140**

Ilustración 172: Control Cliente - 1 ..... **140**

Ilustración 173: Control Cliente - 2 ..... **141**

Ilustración 174: Control Cliente - 3 ..... **141**

Ilustración 175: Control Cliente - 4 ..... **142**

Ilustración 176: Control Cliente - 5 ..... **142**

Ilustración 177: Control Cliente - 6 ..... **143**

Ilustración 178: Control Cliente - 7 ..... **143**

Ilustración 179: Control Cliente - 8 ..... **144**

Ilustración 180: Modelo Cliente - 1 ..... **144**

Ilustración 181: Modelo Cliente - 2 ..... **145**

Ilustración 182: Modelo Cliente - 3 ..... **145**

Ilustración 183: Modelo Cliente - 4 ..... **146**

Ilustración 184: Diagrama de secuencia - Registrar trabajador **147**

Ilustración 185: Vista Trabajador - 1 ..... **148**

Ilustración 186: Vista Trabajador - 2 ..... **148**

Ilustración 187: Vista Trabajador - 3 ..... **149**

Ilustración 188: Vista Trabajador - 4 ..... **149**

Ilustración 189: Vista Trabajador - 5 ..... **150**

Ilustración 190: Vista Trabajador - 6 ..... **150**

Ilustración 191: Vista Trabajador - 7 ..... **151**

Ilustración 192: Vista Trabajador - 8 ..... **151**

Ilustración 193: Vista Trabajador - 9 ..... **152**

Ilustración 194: Vista Trabajador - 10 ..... **152**

Ilustración 195: Vista Trabajador - 11 ..... **153**

Ilustración 196: Vista Trabajador - 12 ..... **153**

Ilustración 197: Vista Trabajador - 13 ..... **154**

Ilustración 198: Vista Trabajador - 14 ..... **154**

Ilustración 199: Vista Trabajador - 15 ..... **155**

Ilustración 200: Control Trabajador - 1 ..... **155**

Ilustración 201: Control Trabajador - 2 ..... **156**

Ilustración 202: Control Trabajador - 3 ..... **156**

Ilustración 203: Control Trabajador - 4 ..... **157**

Ilustración 204: Control Trabajador - 5 ..... **157**

Ilustración 205: Control Trabajador - 6 ..... **158**

Ilustración 206: Control Trabajador - 7 ..... **158**

Ilustración 207: Control Trabajador - 8 ..... **159**

Ilustración 208: Control Trabajador - 9 ..... **159**

Ilustración 209: Control Trabajador - 10 ..... **160**

Ilustración 210: Control Trabajador - 11 ..... **160**

Ilustración 211: Control Trabajador - 12 ..... **161**

Ilustración 212: Control Trabajador - 13 ..... **161**

Ilustración 213: Modelo Trabajador - 1 ..... **162**

Ilustración 214: Modelo Trabajador - 2 ..... **162**

Ilustración 215: Modelo Trabajador - 3 ..... **163**

Ilustración 216: Modelo Trabajador - 4 ..... **163**

Ilustración 217: Modelo Trabajador - 5 ..... **164**

Ilustración 218: Modelo Trabajador - 6 ..... **164**

Ilustración 219: Modelo Trabajador - 7 ..... **165**

Ilustración 220: Modelo Trabajador - 8 ..... **165**

Ilustración 221: Modelo Trabajador - 9 ..... **166**

Ilustración 222: Diagrama de secuencia - Registrar habitaciones **167**

Ilustración 223: Vista Habitaciones - 1 ..... **168**

Ilustración 224: Vista Habitaciones - 2 ..... **169**

Ilustración 225: Vista Habitaciones - 3 ..... **170**

Ilustración 226: Vista Habitaciones - 4 ..... **171**

Ilustración 227: Vista Habitaciones - 5 ..... **172**

Ilustración 228: Vista Habitaciones - 6 ..... **173**

Ilustración 229: Vista Habitaciones - 7 ..... **174**

Ilustración 230: Vista Habitaciones - 8 ..... **175**

Ilustración 231: Vista Habitaciones - 9 ..... **176**

Ilustración 232: Vista Habitaciones - 10 ..... **177**

Ilustración 233: Vista Habitaciones - 11 ..... **178**

Ilustración 234: Control Habitaciones - 1 ..... **179**

Ilustración 235: Control Habitaciones - 2 ..... **180**

Ilustración 236: Control Habitaciones - 3 ..... **181**

Ilustración 237: Control Habitaciones - 4 ..... **182**

Ilustración 238: Control Habitaciones - 5 ..... **183**

Ilustración 239: Control Habitaciones - 6 ..... **184**

Ilustración 240: Modelo Habitaciones - 1 ..... **185**

Ilustración 241: Modelo Habitaciones - 2 ..... **186**

Ilustración 242: Modelo Habitaciones - 3 ..... **187**

Ilustración 243: Diagrama de secuencia - Registrar reserva **188**

Ilustración 244: Vista Reserva - 1 ..... **189**

Ilustración 245: Vista Reserva - 2 ..... **189**

Ilustración 246: Vista Reserva - 3 ..... **190**

Ilustración 247: Vista Reserva - 4 ..... **190**

Ilustración 248: Vista Reserva - 5 ..... **191**

Ilustración 249: Vista Reserva - 6 ..... **191**

Ilustración 250: Vista Reserva - 7 ..... **192**

Ilustración 251: Vista Reserva - 8 192

Ilustración 252: Vista Reserva - 9 193

Ilustración 253: Vista Reserva - 10 193

Ilustración 254: Vista Reserva - 11 194

Ilustración 255: Vista Reserva - 12 194

Ilustración 256: Vista Reserva - 13 195

Ilustración 257: Vista Reserva - 14 195

Ilustración 258: Vista Reserva - 15 196

Ilustración 259: Control Reserva - 1 197

Ilustración 260: Control Reserva - 2 197

Ilustración 261: Control Reserva - 3 198

Ilustración 262: Control Reserva - 4 198

Ilustración 263: Control Reserva - 5 199

Ilustración 264: Control Reserva - 6 199

Ilustración 265: Control Reserva - 7 200

Ilustración 266: Control Reserva - 8 200

Ilustración 267: Control Reserva - 9 201

Ilustración 268: Control Reserva - 10 201

Ilustración 269: Control Reserva - 11 202

Ilustración 270: Control Reserva - 12 202

Ilustración 271: Control Reserva - 13 203

Ilustración 272: Control Reserva - 14 203

Ilustración 273: Control Reserva - 15 204

Ilustración 274: Modelo Reserva - 1 205

Ilustración 275: Modelo Reserva - 2 205

Ilustración 276: Modelo Reserva - 3 206

Ilustración 277: Modelo Reserva - 4 206

Ilustración 278: Modelo Reserva - 5 207

Ilustración 279: Videoconferencia con representante de la empresa 210

Ilustración 280: Integrante - Diego Tasaico 211

Ilustración 281: Integrante - Mauricio Villanueva 211

Ilustración 282: Integrante - Olenka Lazo.....212

Ilustración 283: Integrante - Abihail Zapata 212

Ilustración 284: Acta de compromiso.....213

Ilustración 285: Funcionamiento del Software - Acceso al sistema 214

Ilustración 286: Funcionamiento del Software - Menú Principal.....214

Ilustración 287: Funcionamiento del Software - Vista de los clientes.....215

Ilustración 288: Funcionamiento del Software - Vista de los trabajadores.....215

Ilustración 289: Funcionamiento del Software - Vista de los movimientos 215

Ilustración 290: Funcionamiento del Software - Vista de las reservas.....215

Ilustración 291: Funcionamiento del Software - Vista de las habitaciones.....215

Ilustración 292: Funcionamiento del Software - Detección de errores 215

Ilustración 293: Funcionamiento del Software - Conexión a Base de Datos 215

Ilustración 294: Funcionamiento del Software - Tiempo de respuesta 215

Ilustración 295: Funcionamiento del Software - Espacio mínimo de memoria.....215

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	• Registro de cliente. • Consulta de precio de habitación. • Selección de habitación para el cliente. • Verificar que la habitación se encuentre vacía y limpia. • Realización del pago del cliente al hotel. • Registrar salida del cliente del hotel.	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

Ilustración 5: Prototipo - Versión 1 - 2

Ilustración 6: Prototipo - Versión 1 - 3

Ilustración 7: Prototipo - Versión 1 - 4

Ilustración 8: Prototipo - Versión 2 - 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

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_I
N_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_S
UBSTITUTION';


-- -----------------------------------------------------

-- 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

Ilustración 42: Código de programación del software - 1

Ilustración 43: Código de programación del software - 2

```
70          if(e.getSource() == vista.btn_nuevo_cliente){
71              limpiarCajas();;
72              vista.btn_nuevo_cliente.setEnabled(false);
73              vista.btn_registrar_cliente.setEnabled(true);
74              vista.btn_actualizar_cliente.setEnabled(false);
75          }
76          if(e.getSource() == vista.btn_salir_cliente){
77              salir();
78          }
79      }
80      private void registrar(){
81          if (vista.txt_nombre_cliente.getText().equals("") || vista.txt_apellido_p_cliente.getText().equals("") || vista.txt_apellido_m_cliente.getText().equa
82              || vista.txt_tipo_doc_cliente.getText().equals("") || vista.txt_num_doc_cliente.getText().equals("") || vista.txt_cel_cliente.getText().equal
83              JOptionPane.showMessageDialog(vista, "Debe completar los campos de nombres, apellidos, tipo y numero de documento y celular");
84              listar = false;
85          }
86          else{
87              listar = true;
88              String nombres = vista.txt_nombre_cliente.getText();
89              String apellido_paterno = vista.txt_apellido_p_cliente.getText();
90              String apellido_materno = vista.txt_apellido_m_cliente.getText();
91              String tipo_documento = vista.txt_tipo_doc_cliente.getText();
92              String num_documento = vista.txt_num_doc_cliente.getText();
93              String celular = vista.txt_cel_cliente.getText();
94              String email = vista.txt_email_cliente.getText();
95
96              Cliente cliente = new Cliente(nombres, apellido_paterno, apellido_materno, tipo_documento, num_documento, celular, email);
97              int listo = this.dao.agregar(cliente);
98              if(listo == 1){
99                  JOptionPane.showMessageDialog(vista, "Cliente registrado con éxito");
100             }else{
101                 JOptionPane.showMessageDialog(vista, "Error al registrar cliente");
102             }
103         }
104     }
```

Ilustración 44: Código de programación del software - 3

```
144     private void limpiarCajasNoID(){
145         vista.txt_nombre_cliente.setText("");
146         vista.txt_apellido_p_cliente.setText("");
147         vista.txt_apellido_m_cliente.setText("");
148         vista.txt_tipo_doc_cliente.setText("");
149         vista.txt_num_doc_cliente.setText("");
150         vista.txt_cel_cliente.setText("");
151         vista.txt_email_cliente.setText("");
152     }
153
154     private void editar(){
155         int fila = vista.tb_datos_cliente.getSelectedRow();
156             if(fila == -1){
157                 JOptionPane.showMessageDialog(vista, "Debe seleccionar una fila");
158             }else{
159                 this.vista.btn_registrar_cliente.setEnabled(false);
160                 this.vista.btn_actualizar_cliente.setEnabled(true);
161                 this.vista.btn_nuevo_cliente.setEnabled(true);
162                 int codigo = Integer.parseInt((String)vista.tb_datos_cliente.getValueAt(fila, 0).toString());
163                 String nombres = (String)vista.tb_datos_cliente.getValueAt(fila, 1);
164                 String apellido_paterno = (String)vista.tb_datos_cliente.getValueAt(fila, 2);
165                 String apellido_materno = (String)vista.tb_datos_cliente.getValueAt(fila, 3);
166                 String tipo_Documento = (String)vista.tb_datos_cliente.getValueAt(fila, 4);
167                 String num_documento = (String)vista.tb_datos_cliente.getValueAt(fila, 5);
168                 String celular = (String)vista.tb_datos_cliente.getValueAt(fila, 6);
169                 String email = (String)vista.tb_datos_cliente.getValueAt(fila, 7);
170
171                 vista.txt_idCliente.setText(codigo+"");
172                 vista.txt_nombre_cliente.setText(nombres);
173                 vista.txt_apellido_p_cliente.setText(apellido_paterno);
174                 vista.txt_apellido_m_cliente.setText(apellido_materno);
175                 vista.txt_tipo_doc_cliente.setText(tipo_Documento);
176                 vista.txt_num_doc_cliente.setText(num_documento);
177                 vista.txt_cel_cliente.setText(celular);
178                 vista.txt_email_cliente.setText(email);
179             }
180     }
```

Ilustración 45: Código de programación del software - 4

```
182     private void actualizar(){
183         if (vista.txt_nombre_cliente.getText().equals("") || vista.txt_apellido_p_cliente.getText().equals("") || vista.txt_apellido_m_cliente.getText().
184             || vista.txt_tipo_doc_cliente.getText().equals("") || vista.txt_num_doc_cliente.getText().equals("") || vista.txt_cel_cliente.getText().eq
185             JOptionPane.showMessageDialog(vista, "Debe completar los campos de nombres, apellidos, tipo y numero de documento y celular");
186             listar = false;
187         }else{
188             actualizar = true;
189             vista.btn_registrar_cliente.setEnabled(true);
190             vista.btn_actualizar_cliente.setEnabled(false);
191             vista.btn_nuevo_cliente.setEnabled(false);
192             int idCliente = Integer.parseInt(vista.txt_idCliente.getText());
193             String nombres = vista.txt_nombre_cliente.getText();
194             String apellido_paterno = vista.txt_apellido_p_cliente.getText();
195             String apellido_materno = vista.txt_apellido_m_cliente.getText();
196             String tipo_documento = vista.txt_tipo_doc_cliente.getText();
197             String num_documento = vista.txt_num_doc_cliente.getText();
198             String celular = vista.txt_cel_cliente.getText();
199             String email = vista.txt_email_cliente.getText();
200
201             Cliente cliente = new Cliente(idCliente, nombres, apellido_paterno, apellido_materno, tipo_documento, num_documento, celular, email);
202             int r = this.dao.actualizar(cliente);
203             if(r == 1){
204                 JOptionPane.showMessageDialog(vista, "Cliente actualizado con éxito");
205             }else{
206                 JOptionPane.showMessageDialog(vista, "Error al actualizar cliente");
207             }
208         }
209     }
```

Ilustración 46: Código de programación del software - 5

```
208         }
209     }
210
211     private void eliminar(){
212         int fila = vista.tb_datos_cliente.getSelectedRow();
213         if(fila == -1){
214             JOptionPane.showMessageDialog(vista, "Debe seleccionar una fila");
215         }else{
216             int id = Integer.parseInt((String)vista.tb_datos_cliente.getValueAt(fila, 0).toString());
217             int iden = this.dao.eliminar(id);
218             if (iden == 1){
219                 JOptionPane.showMessageDialog(vista, "Cliente eliminado con éxito");
220                 vista.btn_registrar_cliente.setEnabled(true);
221                 vista.btn_nuevo_cliente.setEnabled(false);
222                 vista.btn_actualizar_cliente.setEnabled(false);
223             }else{
224                 JOptionPane.showMessageDialog(vista, "Error al eliminar cliente");
225             }
226         }
227     }
228     private void salir(){
229         System.exit(0);
230     }
231 }
```

Ilustración 47: Código de programación del software - 6

```java
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.Habitacion;
import models.HabitacionDAO;
import views.VistaHabitaciones;

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

    DefaultTableModel modeloTabla = new DefaultTableModel();
    VistaHabitaciones vista = new VistaHabitaciones();
    HabitacionDAO dao = new HabitacionDAO();

    public HabitacionController(VistaHabitaciones vistaHabitaciones){
        this.vista=vistaHabitaciones;
        this.vista.btnListar_hab.addActionListener(this);
        this.vista.btnnuevo_hab.addActionListener(this);
        this.vista.btnguardar_hab.addActionListener(this);
        this.vista.btncancelar_hab.addActionListener(this);
        this.vista.btnbuscar_hab.addActionListener(this);
        this.vista.btneliminar_hab.addActionListener(this);
        this.vista.btnsalir_hab.addActionListener(this);
        this.vista.btneditar_hab.addActionListener(this);
    }

    @Override
    public void actionPerformed(ActionEvent e) {

    }
}
```

Ilustración 48: Código de programación del software - 7

```java
    private void listar(JTable tabla){
        this.vista.btn_editar_cliente.setEnabled(true);
        this.vista.btn_eliminar_cliente.setEnabled(true);
        this.modeloTabla = (DefaultTableModel)tabla.getModel();
        List<Cliente> listarClientes = this.dao.listar();
        Object[] objeto = new Object[9];
        for(int i=0;i<listarClientes.size();i++){
            objeto[0] = listarClientes.get(i).getIdCliente();
            objeto[1] = listarClientes.get(i).getNombres();
            objeto[2] = listarClientes.get(i).getApellido_paterno();
            objeto[3] = listarClientes.get(i).getApellido_materno();
            objeto[4] = listarClientes.get(i).getTipo_documento();
            objeto[5] = listarClientes.get(i).getNum_documento();
            objeto[6] = listarClientes.get(i).getCelular();
            objeto[7] = listarClientes.get(i).getEmail();
            modeloTabla.addRow(objeto);
        }
        vista.tb_datos_cliente.setModel(modeloTabla);
    }

    private void limpiarTabla (){
        for(int i=0;i<vista.tb_datos_cliente.getRowCount();i++){
            modeloTabla.removeRow(i);
            i = i-1;
        }
    }

    private void limpiarCajas(){
        vista.txt_idCliente.setText("");
        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("");
    }
```

Ilustración 49: Código de programación del software - 8

Ilustración 50: Código de programación del software - 9

Ilustración 51: Código de programación del software - 10

```java
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

```java
    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 = ?, emai
        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();
```

Ilustración 53: Código de programación del software - 12

Ilustración 54: Código de programación del software - 13

Ilustración 55: Código de programación del software - 14

```java
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

```java
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.err.println("Error al listar: "+e.getMessage());
        }
        return listarHabitacion;
    }
```

Ilustración 57: Código de programación del software - 16

Ilustración 58: Código de programación del software - 17

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 nuestro 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	Escenario 1: Validar trabajador. • Administrador ingresa su usuario y su contraseña • El sistema valida el usuario y contraseña Escenario 2: Consulta de trabajadores • Administrador solicita la consulta de todos los trabajadores. • El sistema le muestra los datos de todos los trabajadores. Escenario 3: Modificar al trabajador • Administrador solicita modificar los datos del trabajador • El sistema le permite modificar los datos del trabajador y se realiza los cambios respectivos Escenario 4: Eliminar al trabajador • Administrador solicita eliminar un trabajador • El sistema le permite eliminar al trabajador. ➢ El administrador registra todos los datos del trabajador ➢ El sistema valida todos los datos del trabajador ➢ El sistema registra el nombre del trabajador ➢ El sistema registra el apellido paterno del trabajador ➢ El sistema registra el apellido materno del trabajador ➢ El sistema registra el tipo de documento del trabajador ➢ El sistema registra el número de documento del trabajador ➢ El sistema registra el celular del trabajador ➢ El sistema registra el email del trabajador ➢ El sistema registra el sueldo del trabajador

		➢ El sistema registra el acceso del trabajador ➢ El sistema registra el login del trabajador ➢ El sistema registra el password del trabajador ➢ El sistema registra el estado del trabajador
8	Escenario Alternativo	➢ Si no se registra el trabajador, enviar mensaje de ayuda. ➢ Si faltan datos del trabajador, enviar mensaje de ayuda. ➢ Si el usuario no es válido, enviar mensaje de ayuda. ➢ Si la contraseña no es válida, enviar mensaje de ayuda. ➢ Si se consulta datos de un trabajador no seleccionado, enviar mensaje de ayuda. ➢ Si se modifica datos de un trabajador no seleccionado, enviar mensaje de ayuda. ➢ Si se elimina un trabajador no seleccionado, enviar mensaje de ayuda.
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 nuestro 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	Escenario 1: Validar trabajador. • Administrador o recepcionista ingresa su usuario y su contraseña • El sistema valida el usuario y contraseña Escenario 2: Consulta de clientes • Administrador o recepcionista solicita la consulta de todos los clientes. • El sistema le muestra los datos de todos los clientes. Escenario 3: Modificar al cliente • Administrador o recepcionista solicita modificar los datos del cliente • El sistema le permite modificar los datos del cliente y se realiza los cambios respectivos Escenario 4: Eliminar al cliente • Administrador o recepcionista solicita eliminar un cliente • El sistema le permite eliminar al cliente. ➢ El administrador o recepcionista registra todos los datos del cliente ➢ El sistema valida todos los datos del cliente ➢ El sistema registra el nombre del cliente ➢ El sistema registra el apellido paterno del cliente ➢ El sistema registra el apellido materno del cliente

		➢ El sistema registra el tipo de documento del cliente ➢ El sistema registra el número de documento del cliente ➢ El sistema registra el celular del cliente ➢ El sistema registra el email del cliente
8	Escenario Alternativo	➢ Si no se registra el cliente, enviar mensaje de ayuda. ➢ Si faltan datos del cliente, enviar mensaje de ayuda. ➢ Si el usuario no es válido, enviar mensaje de ayuda. ➢ Si la contraseña no es válida, enviar mensaje de ayuda. ➢ Si se consulta datos de un cliente no seleccionado, enviar mensaje de ayuda. ➢ Si se modifica datos de un cliente no seleccionado, enviar mensaje de ayuda. ➢ Si se elimina un cliente no seleccionado, enviar mensaje de ayuda.
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	Escenario 1: Validar trabajador. • Administrador ingresa su usuario y su contraseña • El sistema valida el usuario y contraseña Escenario 2: Consulta de ingreso de clientes • Administrador solicita la consulta de ingreso de todos los clientes. • El sistema le muestra el ingreso de todos los clientes. Escenario 3: Modificar el ingreso de cliente • Administrador solicita modificar el ingreso del cliente • El sistema le permite modificar el ingreso del cliente y se realiza los cambios respectivos Escenario 4: Eliminar el ingreso del cliente • Administrador solicita eliminar el ingreso de cliente • El sistema le permite eliminar el ingreso de cliente. ➢ El administrador registra el ingreso del cliente ➢ El sistema valida los datos de ingreso del cliente ➢ El sistema registra el ingreso del cliente
8	Escenario Alternativo	➢ Si no se registra el ingreso del cliente, enviar mensaje de ayuda. ➢ Si faltan datos en el ingreso de cliente, enviar mensaje de ayuda. ➢ Si el usuario no es válido, enviar mensaje de ayuda. ➢ Si la contraseña no es válida, enviar mensaje de ayuda.

		➢ Si se consulta el ingreso del cliente no seleccionado, enviar mensaje de ayuda. ➢ Si se modifica el ingreso del cliente no seleccionado, enviar mensaje de ayuda. ➢ Si se elimina el ingreso del cliente no seleccionado, enviar mensaje de ayuda.
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	Escenario 1: Validar trabajador. • Administrador ingresa su usuario y su contraseña • El sistema valida el usuario y contraseña Escenario 2: Consulta de salida de clientes • Administrador solicita la consulta de salida de todos los clientes. • El sistema le muestra la salida de todos los clientes. Escenario 3: Modificar la salida de cliente • Administrador solicita modificar la salida del cliente • El sistema le permite modificar la salida del cliente y se realiza los cambios respectivos Escenario 4: Eliminar la salida del cliente • Administrador solicita eliminar la salida de cliente • El sistema le permite eliminar la salida de cliente. ➢ El administrador registra la salida del cliente ➢ El sistema valida los datos de salida del cliente ➢ El sistema registra la salida del cliente
8	Escenario Alternativo	➢ Si no se registra la salida del cliente, enviar mensaje de ayuda. ➢ Si faltan datos en la salida de cliente, enviar mensaje de ayuda. ➢ Si el usuario no es válido, enviar mensaje de ayuda. ➢ Si la contraseña no es válida, enviar mensaje de ayuda.

		➢ Si se consulta la salida del cliente no seleccionado, enviar mensaje de ayuda. ➢ Si se modifica la salida del cliente no seleccionado, enviar mensaje de ayuda. ➢ Si se elimina la salida del cliente no seleccionado, enviar mensaje de ayuda.
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	Escenario 1: Validar trabajador. • Administrador ingresa su usuario y su contraseña • El sistema valida el usuario y contraseña Escenario 2: Consulta de habitaciones • Administrador solicita la consulta de todas las habitaciones. • El sistema le muestra los datos de todas las habitaciones. Escenario 3: Modificar habitación • Administrador solicita modificar los datos de la habitación • El sistema le permite modificar los datos de la habitación y se realiza los cambios respectivos Escenario 4: Eliminar habitación • Administrador solicita eliminar una habitación • El sistema le permite eliminar la habitación ➢ El administrador registra todos los datos de la habitación ➢ El sistema valida todos los datos de la habitación ➢ El sistema registra el número de la habitación ➢ El sistema registra el piso de la habitación ➢ El sistema registra el precio de la habitación ➢ El sistema registra el tipo de habitación ➢ El sistema registra el estado de la habitación
8	Escenario Alternativo	➢ Si no se registra la habitación, enviar mensaje de ayuda. ➢ Si faltan datos de la habitación, enviar mensaje de ayuda.

		➢ Si el usuario no es válido, enviar mensaje de ayuda. ➢ Si la contraseña no es válida, enviar mensaje de ayuda. ➢ Si se consulta datos de una habitación no seleccionada, enviar mensaje de ayuda. ➢ Si se modifica datos de una habitación no seleccionada, enviar mensaje de ayuda. ➢ Si se elimina una habitación no seleccionada, enviar mensaje de ayuda.
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	• Validar trabajador mediante el Login • Validar registro del cliente
6	Post condiciones	La reserva será registrada exitosamente.
7	Escenario básico	Escenario 1: Validar trabajador. • Administrador o recepcionista ingresa su usuario y su contraseña • El sistema valida el usuario y contraseña Escenario 2: Consulta de reservas • Administrador o recepcionista solicita la consulta de todas las reservas. • El sistema le muestra los datos de todas las reservas. Escenario 3: Modificar reserva • Administrador o recepcionista solicita modificar reserva del cliente. • El sistema le permite modificar los datos de la reserva y se realiza los cambios respectivos Escenario 4: Eliminar reserva • Administrador o recepcionista solicita eliminar una reserva. • El sistema le permite eliminar la reserva. ➢ El administrador o recepcionista registra todos los datos de la reserva ➢ El sistema valida todos los datos de la reserva ➢ El sistema registra el tipo de reserva ➢ El sistema registra la fecha de reserva

		➢ El sistema registra la fecha de ingreso de la reserva ➢ El sistema registra la fecha de salida de la reserva ➢ El sistema registra el costo de la reserva ➢ El sistema registra el estado de la reserva
8	Escenario Alternativo	➢ Si no se registra la reserva, enviar mensaje de ayuda. ➢ Si faltan datos de la reserva, enviar mensaje de ayuda. ➢ Si el usuario no es válido, enviar mensaje de ayuda. ➢ Si la contraseña no es válida, enviar mensaje de ayuda. ➢ Si se consulta datos de una reserva no seleccionada, enviar mensaje de ayuda. ➢ Si se modifica datos de una reserva no seleccionada, enviar mensaje de ayuda. ➢ Si se elimina una reserva no seleccionada, enviar mensaje de ayuda.
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 de 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 de Entrada de Cliente	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 la Entrada de Cliente	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 la Entrada de Cliente	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 la Entrada 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
Registro de Salida de Cliente	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 la Salida de Cliente	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 la 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 Registrar 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

Código	Nombres	Apellido Paterno	Apellido Materno	Tipo de documento	Nro de documento	Celular	Email
1	Veronica	Bonilla	Correa	DNI	72914278	991576132	veronica@gmail.c...
2	Ruben	Calle	Paniagua	DNI	45129453	990123942	ruben@gmail.com
3	Mauricio	Villanueva	Yaranga	DNI	74125201	931382482	mauricio@gmail.c...
4	Abihail	Zapata	Zamora	DNI	73924682	999145234	abihail@gmail.com
5	Olenka	Lazo	Pinchi	DNI	45212342	992133542	olenka@gmail.com
6	Diego	Tasaico	Chumpitaz	DNI	77124221	992132421	diego@gmail.com
7	Gianluca	Lapadula	Vargas	DNI	72120123	913124121	gianluca@gmail.c...
8	Fabrizio	Crisanto	Narvaez	DNI	43142321	912312353	fabrizio@gmail.com
9	Paolo	Guerrero	Gonzales	DNI	10392213	912124211	paolo@gmail.com
10	Adolfo	Rojas	Reyes	DNI	31242421	912123153	adolfo@gmail.com
11	Andres	Hurtado	Mamani	DNI	72312321	911864932	andres@gmail.com
12	Justin	Bieber	Gomez	DNI	42123201	991232124	justino@gmail.com
13	Aaron	Huamani	Narvaez	DNI	31221234	998812313	aaronhua@gmail....
14	Federico	Zevallo	Montenegro	DNI	43192421	900212455	federico@gmail.co...
15	Mariana	Gutierrez	Saenz	DNI	73219244	991321543	mariana@gmail.c...
16	Juliana	Arce	Soria	DNI	43122134	987355521	juliana@gmail.com
17	Jorge	Astengo	Perez	DNI	42391245	992157731	jorge@gmail.com
18	Patricio	Marquez	Castillo	DNI	73432355	911245211	patricio@gmail.com

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

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

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 Registrar 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 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 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 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 Registrar 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 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 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 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

Código	Número	Piso	Precio	Tipo de habitación	Estado
1	101	1	25.0	Individual	Limpio
2	102	1	25.0	Individual	Limpio
3	103	1	40.0	Doble	Mantenimiento
4	104	1	25.0	Individual	Limpio
5	105	1	40.0	Doble	Limpio
6	201	2	30.0	Individual	Limpio
7	202	2	30.0	Individual	Limpio
8	203	2	50.0	Doble	Mantenimiento
9	204	2	30.0	Individual	Mantenimiento
10	205	2	50.0	Individual	Limpio
11	301	3	50.0	Individual	Limpio
12	302	3	100.0	Individual	Mantenimiento
13	303	3	100.0	Doble	Mantenimiento
14	304	3	120.0	Doble	Limpio
15	305	3	100.0	Individual	Limpio
16	401	4	150.0	Individual	Limpio
17	402	4	150.0	Doble	Limpio

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	ACCCIÓN 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	0%
	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 muestra las imágenes de la carpeta de software y de informe en el GitHub?	GitHub	X					Sección repositorio GitHub	20	25	80%
	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	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			
Confiablidad	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	20%
	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			
	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	60%

	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	0%
	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	ACCCIÓN 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	100%
	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			
Confiablidad	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	100%
	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

Diagrama de secuencia del caso de uso Registrar cliente

Ilustración 160: Diagrama de secuencia – Registrar cliente

Vista del Cliente

Ilustración 161: Vista Cliente - 1

Ilustración 162: Vista Cliente - 2

```
58          btn_eliminar_cliente = new javax.swing.JButton();
59          btn_editar_cliente = new javax.swing.JButton();
60          jLabel17 = new javax.swing.JLabel();
61          txt_buscar_cliente = new javax.swing.JTextField();
62          btn_buscar_cliente = new javax.swing.JButton();
63          btn_actualizar_cliente = new javax.swing.JButton();
64
65          setClosable(true);
66          setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
67          setIconifiable(true);
68          setMaximizable(true);
69
70          jLabel1.setFont(new java.awt.Font("Tahoma", 3, 18)); // NOI18N
71          jLabel1.setText("Clientes");
72
73          jPanel1.setBackground(new java.awt.Color(216, 216, 255));
74          jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Registro de Clientes"));
75
76          jLabel2.setText("Código:");
77
78          jLabel3.setText("Nombres: ");
79
80          jLabel4.setText("Apellido Paterno: ");
81
82          jLabel5.setText("Apellido Materno: ");
83
84          jLabel6.setText("Tipo de Documento: ");
85
86          jLabel7.setText("Número de Documento:");
87
88          jLabel8.setText("Celular: ");
89
90          jLabel9.setText("Email: ");
91
92          txt_idCliente.setEditable(false);
93          txt_idCliente.addActionListener(new java.awt.event.ActionListener() {
94              public void actionPerformed(java.awt.event.ActionEvent evt) {
95                  txt_idClienteActionPerformed(evt);
96              }
```

Ilustración 163: Vista Cliente - 3

```
97          });
98
99          txt_apellido_p_cliente.addActionListener(new java.awt.event.ActionListener() {
100             public void actionPerformed(java.awt.event.ActionEvent evt) {
101                 txt_apellido_p_clienteActionPerformed(evt);
102             }
103         });
104
105         btn_registrar_cliente.setIcon(new javax.swing.ImageIcon(getClass().getResource("/views/icon/agregar.png"))); // NOI18N
106         btn_registrar_cliente.setText("Registrar");
107
108         btn_limpiarCampos_cliente.setIcon(new javax.swing.ImageIcon(getClass().getResource("/views/icon/limpiar.png"))); // NOI18N
109         btn_limpiarCampos_cliente.setText("Limpiar Campos");
110
111         btn_nuevo_cliente.setIcon(new javax.swing.ImageIcon(getClass().getResource("/views/icon/nuevo.png"))); // NOI18N
112         btn_nuevo_cliente.setText("Nuevo");
113         btn_nuevo_cliente.setEnabled(false);
114
115         javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
116         jPanel1.setLayout(jPanel1Layout);
117         jPanel1Layout.setHorizontalGroup(
118             jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
119             .addGroup(jPanel1Layout.createSequentialGroup()
120                 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
121                     .addGroup(jPanel1Layout.createSequentialGroup()
122                         .addContainerGap()
123                         .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
124                             .addComponent(jLabel2)
125                             .addComponent(jLabel3)
126                             .addComponent(jLabel4)
127                             .addComponent(jLabel5)
128                             .addComponent(jLabel6)
129                             .addComponent(jLabel7)
130                             .addComponent(jLabel8)
131                             .addComponent(jLabel9)))
132                     .addComponent(btn_registrar_cliente))
133                 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
134                     .addGroup(jPanel1Layout.createSequentialGroup()
135                         .addGap(18, 18, 18)
136                         .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

Ilustración 164: Vista Cliente - 4

```
137                     .addComponent(txt_num_doc_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, 190, javax.swing.GroupLayout.PREFERRED_SIZE)
138                     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
139                         .addComponent(txt_apellido_m_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, 190, javax.swing.GroupLayout.PREFERRED_SIZE)
140                         .addComponent(txt_tipo_doc_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, 190, javax.swing.GroupLayout.PREFERRED_SIZE)
141                         .addComponent(txt_apellido_p_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, 190, javax.swing.GroupLayout.PREFERRED_SIZE)
142                         .addComponent(txt_nombre_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, 190, javax.swing.GroupLayout.PREFERRED_SIZE)
143                         .addComponent(txt_idCliente, javax.swing.GroupLayout.PREFERRED_SIZE, 190, javax.swing.GroupLayout.PREFERRED_SIZE))
144                     .addComponent(txt_cel_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, 190, javax.swing.GroupLayout.PREFERRED_SIZE)
145                     .addComponent(txt_email_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, 190, javax.swing.GroupLayout.PREFERRED_SIZE)))
146                 .addGroup(jPanel1Layout.createSequentialGroup()
147                     .addGap(8, 8, 8)
148                     .addComponent(btn_limpiarCampos_cliente)
149                     .addGap(31, 31, 31)
150                     .addComponent(btn_nuevo_cliente)))
151             .addContainerGap(31, Short.MAX_VALUE))
152         );
153         jPanel1Layout.setVerticalGroup(
154             jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
155             .addGroup(jPanel1Layout.createSequentialGroup()
156                 .addGap(27, 27, 27)
157                 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
158                     .addComponent(jLabel2)
159                     .addComponent(txt_idCliente, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
160                 .addGap(18, 18, 18)
161                 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
162                     .addComponent(jLabel3)
163                     .addComponent(txt_nombre_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
164                 .addGap(18, 18, 18)
165                 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
166                     .addComponent(jLabel4)
167                     .addComponent(txt_apellido_p_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
168                 .addGap(18, 18, 18)
169                 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
170                     .addComponent(jLabel5)
171                     .addComponent(txt_apellido_m_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
172                 .addGap(18, 18, 18)
173                 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
174                     .addComponent(jLabel6)
175                     .addComponent(txt_tipo_doc_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
```

Ilustración 165: Vista Cliente - 5

```
176                 .addGap(18, 18, 18)
177                 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
178                     .addComponent(jLabel7)
179                     .addComponent(txt_num_doc_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
180                 .addGap(18, 18, 18)
181                 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
182                     .addComponent(jLabel8)
183                     .addComponent(txt_cel_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
184                 .addGap(21, 21, 21)
185                 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
186                     .addComponent(jLabel9)
187                     .addComponent(txt_email_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
188                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 38, Short.MAX_VALUE)
189                 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
190                     .addComponent(btn_registrar_cliente)
191                     .addComponent(btn_limpiarCampos_cliente)
192                     .addComponent(btn_nuevo_cliente))
193                 .addGap(12, 12, 12))
194         );
195 
196         jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder("Lista de Clientes"));
197 
198         tb_datos_cliente.setModel(new javax.swing.table.DefaultTableModel(
199             new Object [][] {
200 
201             },
202             new String [] {
203                 "Código", "Nombres", "Apellido Paterno", "Apellido Materno", "Tipo de documento", "Nro de documento", "Celular", "Email"
204             }
205         ) {
206             boolean[] canEdit = new boolean [] {
207                 false, false, false, true, false, false, true, true
208             };
209 
210             public boolean isCellEditable(int rowIndex, int columnIndex) {
211                 return canEdit [columnIndex];
212             }
213         });
214         jScrollPane1.setViewportView(tb_datos_cliente);
```

Ilustración 166: Vista Cliente - 6

```
        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jScrollPane1))
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 314, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 0, Short.MAX_VALUE))
        );

        jPanel3.setBackground(new java.awt.Color(249, 249, 188));
        jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder("Opciones"));

        btn_listar_cliente.setIcon(new javax.swing.ImageIcon(getClass().getResource("/views/icon/listar.png"))); // NOI18N
        btn_listar_cliente.setText("Listar");

        btn_eliminar_cliente.setIcon(new javax.swing.ImageIcon(getClass().getResource("/views/icon/eliminar.png"))); // NOI18N
        btn_eliminar_cliente.setText("Eliminar");
        btn_eliminar_cliente.setEnabled(false);

        btn_editar_cliente.setIcon(new javax.swing.ImageIcon(getClass().getResource("/views/icon/editar.png"))); // NOI18N
        btn_editar_cliente.setText("Editar");
        btn_editar_cliente.setEnabled(false);
        btn_editar_cliente.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_editar_clienteActionPerformed(evt);
            }
        });

        jLabel17.setText("Buscar por nombre: ");

        btn_buscar_cliente.setText("Buscar");
```

Ilustración 167: Vista Cliente - 7

```
        btn_actualizar_cliente.setIcon(new javax.swing.ImageIcon(getClass().getResource("/views/icon/actualizar.png"))); // NOI18N
        btn_actualizar_cliente.setText("Actualizar");
        btn_actualizar_cliente.setEnabled(false);
        btn_actualizar_cliente.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_actualizar_clienteActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(
            jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel17)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(txt_buscar_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, 188, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(btn_buscar_cliente)
                .addGap(27, 27, 27)
                .addComponent(btn_editar_cliente, javax.swing.GroupLayout.DEFAULT_SIZE, 107, Short.MAX_VALUE)
                .addGap(18, 18, 18)
                .addComponent(btn_actualizar_cliente)
                .addGap(18, 18, 18)
                .addComponent(btn_eliminar_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, 112, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(btn_listar_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, 99, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(22, 22, 22))
        );
        jPanel3Layout.setVerticalGroup(
            jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel17)
                    .addComponent(btn_editar_cliente)
                    .addComponent(btn_eliminar_cliente)
                    .addComponent(btn_listar_cliente)
```

Ilustración 168: Vista Cliente - 8

```
                    .addComponent(btn_buscar_cliente)
                    .addComponent(txt_buscar_cliente, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(btn_actualizar_cliente))
                .addContainerGap())
        );

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                            .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
                    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 82, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel1)
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents
```

Ilustración 169: Vista Cliente - 9

```
    private void txt_apellido_p_clienteActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_txt_apellido_p_clienteActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_txt_apellido_p_clienteActionPerformed

    private void txt_idClienteActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_txt_idClienteActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_txt_idClienteActionPerformed

    private void btn_actualizar_clienteActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_btn_actualizar_clienteActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_btn_actualizar_clienteActionPerformed

    private void btn_editar_clienteActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_btn_editar_clienteActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_btn_editar_clienteActionPerformed

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        VistaClientes vista = new VistaClientes();
        ClienteController controlador = new ClienteController(vista);
        vista.setVisible(true);
        //vista.setLocationRelativeTo(vista);

    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    public javax.swing.JButton btn_actualizar_cliente;
    public javax.swing.JButton btn_buscar_cliente;
    public javax.swing.JButton btn_editar_cliente;
    public javax.swing.JButton btn_eliminar_cliente;
    public javax.swing.JButton btn_limpiarCampos_cliente;
    public javax.swing.JButton btn_listar_cliente;
    public javax.swing.JButton btn_nuevo_cliente;
    public javax.swing.JButton btn_registrar_cliente;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel17;
    private javax.swing.JLabel jLabel2;
```

Ilustración 170: Vista Cliente - 10

Ilustración 171: Vista Cliente - 11

Control del Cliente

Ilustración 172: Control Cliente - 1

```
18      DefaultTableModel modeloTabla = new DefaultTableModel();
19      VistaClientes vista = new VistaClientes();
20      ClienteDAO dao = new ClienteDAO();
21      Validaciones val =new Validaciones();
22      boolean var;
23
24      public ClienteController(VistaClientes vistaClientes) {
25          this.vista = vistaClientes;
26          this.vista.btn_listar_cliente.addActionListener(this);
27          this.vista.btn_registrar_cliente.addActionListener(this);
28          this.vista.btn_limpiarCampos_cliente.addActionListener(this);
29          this.vista.btn_editar_cliente.addActionListener(this);
30          this.vista.btn_actualizar_cliente.addActionListener(this);
31          this.vista.btn_eliminar_cliente.addActionListener(this);
32          this.vista.btn_nuevo_cliente.addActionListener(this);
33          this.vista.btn_buscar_cliente.addActionListener(this);
34      }
35
36      @Override
37      public void actionPerformed(ActionEvent e) {
38          if(e.getSource() == vista.btn_listar_cliente){
39              limpiarTabla();
40              listar(vista.tb_datos_cliente);
41              vista.btn_actualizar_cliente.setEnabled(false);
42          }
43          if(e.getSource() == vista.btn_registrar_cliente){
44              registrar();
45              if(listar){
46                  vista.btn_editar_cliente.setEnabled(true);
47                  vista.btn_eliminar_cliente.setEnabled(true);
48                  limpiarCajas();
49                  limpiarTabla();
50                  listar(vista.tb_datos_cliente);
51              }
52          }
53          if(e.getSource() == vista.btn_limpiarCampos_cliente){
54              limpiarCajas();
55              vista.btn_actualizar_cliente.setEnabled(false);
56              vista.btn_editar_cliente.setEnabled(false);
```

Ilustración 173: Control Cliente - 2

```
45              if(listar){
46                  vista.btn_editar_cliente.setEnabled(true);
47                  vista.btn_eliminar_cliente.setEnabled(true);
48                  limpiarCajas();
49                  limpiarTabla();
50                  listar(vista.tb_datos_cliente);
51              }
52          }
53          if(e.getSource() == vista.btn_limpiarCampos_cliente){
54              limpiarCajas();
55              vista.btn_actualizar_cliente.setEnabled(false);
56              vista.btn_editar_cliente.setEnabled(false);
57              vista.btn_eliminar_cliente.setEnabled(false);
58              limpiarTabla();
59          }
60          if(e.getSource() == vista.btn_editar_cliente){
61              editar();
62          }
63          if(e.getSource() == vista.btn_actualizar_cliente){
64              actualizar();
65              if(actualizar){
66                  limpiarTabla();
67                  limpiarCajas();
68                  listar(vista.tb_datos_cliente);
69              }
70          }
71          if(e.getSource() == vista.btn_eliminar_cliente){
72              eliminar();
73              limpiarTabla();
74              limpiarCajas();
75              listar(vista.tb_datos_cliente);
76          }
77
78          if(e.getSource() == vista.btn_nuevo_cliente){
79              limpiarCajas();
80              vista.btn_nuevo_cliente.setEnabled(false);
81              vista.btn_registrar_cliente.setEnabled(true);
82              vista.btn_actualizar_cliente.setEnabled(false);
83          }
```

Ilustración 174: Control Cliente - 3

```
86              limpiarTabla();
87              if(validarBuscar()==true){
88                  listarBuscar(vista.tb_datos_cliente,vista.txt_buscar_cliente.getText());
89                  vista.btn_actualizar_cliente.setEnabled(false);
90              }
91          }
92      }
93      private void registrar(){
94          if (vista.txt_nombre_cliente.getText().equals("") || vista.txt_apellido_p_cliente.getText().equals("") || vista.txt_apellido_m_cliente.getText().equals("")
95                  || vista.txt_tipo_doc_cliente.getText().equals("") || vista.txt_num_doc_cliente.getText().equals("") || vista.txt_cel_cliente.getText().equals("")){
96              JOptionPane.showMessageDialog(vista, "Debe completar todos los campos");
97              listar = false;
98          }
99          else{
100             if(validarNombre()==true && validarApePaterno()==true && validarApeMaterno()==true && validarTipo_documento()==true
101                 && validarNumero_documento()==true && validarCelular()==true && validarEmail()==true){
102                 listar = true;
103                 String nombres = vista.txt_nombre_cliente.getText();
104                 String apellido_paterno = vista.txt_apellido_p_cliente.getText();
105                 String apellido_materno = vista.txt_apellido_m_cliente.getText();
106                 String tipo_documento = vista.txt_tipo_doc_cliente.getText();
107                 String num_documento = vista.txt_num_doc_cliente.getText();
108                 String celular = vista.txt_cel_cliente.getText();
109                 String email = vista.txt_email_cliente.getText();
110
111                 Cliente cliente = new Cliente(nombres, apellido_paterno, apellido_materno, tipo_documento, num_documento, celular, email);
112                 int listo = this.dao.agregar(cliente);
113                 if(listo == 1){
114                     JOptionPane.showMessageDialog(vista, "Cliente registrado con éxito");
115                 }else{
116                     JOptionPane.showMessageDialog(vista, "Error al registrar cliente");
117                 }
118             }
119         }
120     }
121
122     private void listarBuscar(JTable tabla,String buscar){
123         this.vista.btn_editar_cliente.setEnabled(true);
124         this.vista.btn_eliminar_cliente.setEnabled(true);
```

Ilustración 175: Control Cliente - 4

```
267     private boolean validarNombre(){
268         var = val.validacionNomApe(vista.txt_nombre_cliente.getText());
269         if(var == false){
270             JOptionPane.showMessageDialog(vista, "Ingrese un nombre valido");
271             vista.txt_nombre_cliente.setText("");
272             return false;
273         }else{
274             return true;
275         }
276     }
277
278     private boolean validarApePaterno(){
279         var = val.validacionNomApe(vista.txt_apellido_p_cliente.getText());
280         if(var == false){
281             JOptionPane.showMessageDialog(vista, "Ingrese un apellido valido");
282             vista.txt_apellido_p_cliente.setText("");
283             return false;
284         }else{
285             return true;
286         }
287     }
288
289     private boolean validarApeMaterno(){
290         var = val.validacionNomApe(vista.txt_apellido_m_cliente.getText());
291         if(var == false){
292             JOptionPane.showMessageDialog(vista, "Ingrese un apellido valido");
293             vista.txt_apellido_m_cliente.setText("");
294             return false;
295         }else{
296             return true;
297         }
298     }
299
300     private boolean validarTipo_documento(){
301         var = val.validacionTipoDocumento(vista.txt_tipo_doc_cliente.getText());
302         if(var == false){
303             JOptionPane.showMessageDialog(vista, "Solo puede ingresar DNI o Pasaporte");
304             vista.txt_tipo_doc_cliente.setText("");
305             return false;
```

Ilustración 176: Control Cliente - 5

Ilustración 177: Control Cliente - 6

Ilustración 178: Control Cliente - 7

Ilustración 179: Control Cliente - 8

Modelo del Cliente

Ilustración 180: Modelo Cliente - 1

Ilustración 181: Modelo Cliente - 2

Ilustración 182: Modelo Cliente - 3

Ilustración 183: Modelo Cliente - 4

Diagrama de secuencia del caso de uso Registrar trabajador

Ilustración 184: Diagrama de secuencia - Registrar trabajador

Vista del Trabajador

Ilustración 185: Vista Trabajador - 1

Ilustración 186: Vista Trabajador - 2

Ilustración 187: Vista Trabajador - 3

Ilustración 188: Vista Trabajador - 4

```
114         txt_codigo_trabajador.setEditable(false);
115         txt_codigo_trabajador.addActionListener(new java.awt.event.ActionListener() {
116             public void actionPerformed(java.awt.event.ActionEvent evt) {
117                 txt_codigo_trabajadorActionPerformed(evt);
118             }
119         });
120
121         txt_nombre_trabajador.addActionListener(new java.awt.event.ActionListener() {
122             public void actionPerformed(java.awt.event.ActionEvent evt) {
123                 txt_nombre_trabajadorActionPerformed(evt);
124             }
125         });
126
127         btn_registrar_trabajador.setIcon(new javax.swing.ImageIcon(getClass().getResource("/views/icon/agregar.png"))); // NOI18N
128         btn_registrar_trabajador.setText("Registrar");
129         btn_registrar_trabajador.addActionListener(new java.awt.event.ActionListener() {
130             public void actionPerformed(java.awt.event.ActionEvent evt) {
131                 btn_registrar_trabajadorActionPerformed(evt);
132             }
133         });
134
135         btn_limpiar_campos_trabajador.setIcon(new javax.swing.ImageIcon(getClass().getResource("/views/icon/limpiar.png"))); // NOI18N
136         btn_limpiar_campos_trabajador.setText("Limpiar Campos");
137
138         btn_nuevo_trabajador.setIcon(new javax.swing.ImageIcon(getClass().getResource("/views/icon/nuevo.png"))); // NOI18N
139         btn_nuevo_trabajador.setText("Nuevo");
140         btn_nuevo_trabajador.setEnabled(false);
141         btn_nuevo_trabajador.addActionListener(new java.awt.event.ActionListener() {
142             public void actionPerformed(java.awt.event.ActionEvent evt) {
143                 btn_nuevo_trabajadorActionPerformed(evt);
144             }
```

Ilustración 189: Vista Trabajador - 5

```
144             }
145         });
146
147         javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
148         jPanel1.setLayout(jPanel1Layout);
149         jPanel1Layout.setHorizontalGroup(
150             jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
151             .addGroup(jPanel1Layout.createSequentialGroup()
152                 .addContainerGap()
153                 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
154                     .addGroup(jPanel1Layout.createSequentialGroup()
155                         .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
156                             .addComponent(jLabel7)
157                             .addComponent(jLabel8)
158                             .addComponent(jLabel9)
159                             .addComponent(jLabel10)
160                             .addComponent(jLabel11)
161                             .addComponent(jLabel12)
162                             .addComponent(jLabel13)
163                             .addComponent(jLabel14)
164                             .addComponent(btn_registrar_trabajador))
165                         .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
166                             .addGroup(jPanel1Layout.createSequentialGroup()
167                                 .addGap(20, 20, 20)
168                                 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
169                                     .addComponent(txt_estado_trabajador, javax.swing.GroupLayout.PREFERRED_SIZE, 201, javax.swing.GroupLayout.PREFERRED_SIZE)
170                                     .addComponent(txt_password_trabajador, javax.swing.GroupLayout.PREFERRED_SIZE, 201, javax.swing.GroupLayout.PREFERRED_SIZE)
171                                     .addComponent(txt_login_trabajador, javax.swing.GroupLayout.PREFERRED_SIZE, 201, javax.swing.GroupLayout.PREFERRED_SIZE)
172                                     .addComponent(txt_acceso_trabajador, javax.swing.GroupLayout.PREFERRED_SIZE, 201, javax.swing.GroupLayout.PREFERRED_SIZE)
173                                     .addComponent(txt_sueldo_trabajador, javax.swing.GroupLayout.PREFERRED_SIZE, 201, javax.swing.GroupLayout.PREFERRED_SIZE)
174                                     .addComponent(txt_email_trabajador, javax.swing.GroupLayout.PREFERRED_SIZE, 201, javax.swing.GroupLayout.PREFERRED_SIZE)
175                                     .addComponent(txt_celular_trabajador, javax.swing.GroupLayout.PREFERRED_SIZE, 201, javax.swing.GroupLayout.PREFERRED_SIZE)
```

Ilustración 190: Vista Trabajador - 6

Ilustración 191: Vista Trabajador - 7

Ilustración 192: Vista Trabajador - 8

```
                    .addComponent(jLabel10)
                    .addComponent(txt_sueldo_trabajador, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel11)
                    .addComponent(txt_acceso_trabajador, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED_SIZE, 14, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(txt_login_trabajador, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel13)
                    .addComponent(txt_password_trabajador, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE
                .addGap(14, 14, 14)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(txt_estado_trabajador, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel14))
                .addGap(18, 18, 18)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(btn_nuevo_trabajador)
                    .addComponent(btn_limpiar_campos_trabajador)
                    .addComponent(btn_registrar_trabajador))
                .addContainerGap(15, Short.MAX_VALUE))
        );

        jLabel1.setFont(new java.awt.Font("Tahoma", 3, 18)); // NOI18N
        jLabel1.setText("Trabajadores y Accesos");

        jPanel2.setBackground(new java.awt.Color(255, 255, 144));
        jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder("Opciones"));

        btn_actualizar_trabajador.setIcon(new javax.swing.ImageIcon(getClass().getResource("/views/icon/actualizar.png"))); // NOI18N
```

Ilustración 193: Vista Trabajador - 9

```
        btn_actualizar_trabajador.setText("Actualizar");
        btn_actualizar_trabajador.setEnabled(false);

        btn_editar_trabajador.setIcon(new javax.swing.ImageIcon(getClass().getResource("/views/icon/editar.png"))); // NOI18N
        btn_editar_trabajador.setText("Editar");
        btn_editar_trabajador.setEnabled(false);

        btn_eliminar_trabajador.setIcon(new javax.swing.ImageIcon(getClass().getResource("/views/icon/eliminar.png"))); // NOI18N
        btn_eliminar_trabajador.setText("Eliminar");
        btn_eliminar_trabajador.setEnabled(false);

        btn_listar_trabajador.setIcon(new javax.swing.ImageIcon(getClass().getResource("/views/icon/listar.png"))); // NOI18N
        btn_listar_trabajador.setText("Listar");

        jLabel15.setText("Buscar por nombre: ");

        btn_buscar_trabajador.setText("Buscar");

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel15)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(txt_buscar_trabajador, javax.swing.GroupLayout.PREFERRED_SIZE, 176, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(btn_buscar_trabajador)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 25, Short.MAX_VALUE)
                .addComponent(btn_editar_trabajador, javax.swing.GroupLayout.PREFERRED_SIZE, 117, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(btn_actualizar_trabajador, javax.swing.GroupLayout.PREFERRED_SIZE, 125, javax.swing.GroupLayout.PREFERRED_SIZE)
```

Ilustración 194: Vista Trabajador - 10

Ilustración 195: Vista Trabajador - 11

Ilustración 196: Vista Trabajador - 12

Ilustración 197: Vista Trabajador - 13

Ilustración 198: Vista Trabajador - 14

Ilustración 199: Vista Trabajador - 15

Control del Trabajador

Ilustración 200: Control Trabajador - 1

```
/**
 *
 * @author Diego
 */
public class TrabajadorController implements ActionListener{
    boolean listar;
    boolean actualizar;

    DefaultTableModel modeloTabla = new DefaultTableModel();
    VistaTrabajadores vista = new VistaTrabajadores();
    TrabajadorDAO dao = new TrabajadorDAO();
    Validaciones val =new Validaciones();
    boolean var;

    public TrabajadorController(VistaTrabajadores vistaTrabajadores) {
        this.vista = vistaTrabajadores;
        this.vista.btn_listar_trabajador.addActionListener(this);
        this.vista.btn_registrar_trabajador.addActionListener(this);
        this.vista.btn_limpiar_campos_trabajador.addActionListener(this);
        this.vista.btn_editar_trabajador.addActionListener(this);
        this.vista.btn_actualizar_trabajador.addActionListener(this);
        this.vista.btn_eliminar_trabajador.addActionListener(this);
        this.vista.btn_nuevo_trabajador.addActionListener(this);
        this.vista.btn_buscar_trabajador.addActionListener(this);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == vista.btn_listar_trabajador){
            limpiarTabla();
            listar(vista.tb_datos_trabajador);
            vista.btn_actualizar_trabajador.setEnabled(false);
```

Ilustración 201: Control Trabajador - 2

```
        }
        if(e.getSource() == vista.btn_registrar_trabajador){
            registrar();
            if(listar){
                vista.btn_editar_trabajador.setEnabled(true);
                vista.btn_eliminar_trabajador.setEnabled(true);
                limpiarCajas();
                limpiarTabla();
                listar(vista.tb_datos_trabajador);
            }
        }
        if(e.getSource() == vista.btn_limpiar_campos_trabajador){
            vista.btn_actualizar_trabajador.setEnabled(false);
            vista.btn_editar_trabajador.setEnabled(false);
            vista.btn_eliminar_trabajador.setEnabled(false);
            limpiarCajas();
        }
        if(e.getSource() == vista.btn_editar_trabajador){
            editar();
        }
        if(e.getSource() == vista.btn_actualizar_trabajador){
            actualizar();
            if(actualizar){
                limpiarTabla();
                limpiarCajas();
                listar(vista.tb_datos_trabajador);
            }
        }
        if(e.getSource() == vista.btn_eliminar_trabajador){
            eliminar();
            limpiarTabla();
            limpiarCajas();
            listar(vista.tb_datos_trabajador);
```

Ilustración 202: Control Trabajador - 3

MOANSO-Proyecto/TrabajadorC
github.com/mau98git/MOANSO-Proyecto/blob/proyecto/src/controllers/TrabajadorController.java

```
80
81          if(e.getSource() == vista.btn_nuevo_trabajador){
82              limpiarCajas();
83              vista.btn_nuevo_trabajador.setEnabled(false);
84              vista.btn_registrar_trabajador.setEnabled(true);
85              vista.btn_actualizar_trabajador.setEnabled(false);
86          }
87
88          if(e.getSource() == vista.btn_buscar_trabajador){
89              limpiarTabla();
90              if(validarBuscar()==true){
91                  listarBuscar(vista.tb_datos_trabajador,vista.txt_buscar_trabajador.getText());
92                  vista.btn_actualizar_trabajador.setEnabled(false);
93              }
94          }
95
96      }
97      private void registrar(){
98          if (vista.txt_nombre_trabajador.getText().equals("") || vista.txt_ape_paterno_trabajador.getText().equals("") || vista.txt_ape_materno_trabajador.getText().equals("")
99                  || vista.txt_tipo_documento_trabajador.getText().equals("") || vista.txt_num_documento_trabajador.getText().equals("") || vista.txt_celular_trabajador.getText(
100                 vista.txt_sueldo_trabajador.getText().equals("") || vista.txt_acceso_trabajador.getText().equals("") || vista.txt_login_trabajador.getText().equals("") ||
101                 vista.txt_password_trabajador.getText().equals("") || vista.txt_estado_trabajador.getText().equals("")){
102             JOptionPane.showMessageDialog(vista, "Debe completar todos los campos");
103             listar = false;
104         }
105         else{
106             if(validarNombre()==true && validarApePaterno()==true && validarApeMaterno()==true && validarTipo_documento()==true
107             && validarNumero_documento()==true && validarCelular()==true && validarEmail()==true && validarSueldo()==true && validarAcceso()==true &&
108             validarEstado()==true){
109                 listar = true;
110                 String nombres = vista.txt_nombre_trabajador.getText();
111                 String apellido_paterno = vista.txt_ape_paterno_trabajador.getText();
112                 String apellido_materno = vista.txt_ape_materno_trabajador.getText();
```

Escribe aquí para buscar Escritorio ESP ES 21:00 10/11/2020

Ilustración 203: Control Trabajador - 4

MOANSO-Proyecto/TrabajadorC
github.com/mau98git/MOANSO-Proyecto/blob/proyecto/src/controllers/TrabajadorController.java

```
113                 String tipo_documento = vista.txt_tipo_documento_trabajador.getText();
114                 String num_documento = vista.txt_num_documento_trabajador.getText();
115                 String celular = vista.txt_celular_trabajador.getText();
116                 String email = vista.txt_email_trabajador.getText();
117                 double sueldo = Double.parseDouble(vista.txt_sueldo_trabajador.getText());
118                 String acceso = vista.txt_acceso_trabajador.getText();
119                 String login = vista.txt_login_trabajador.getText();
120                 String password = vista.txt_password_trabajador.getText();
121                 int estado = Integer.parseInt(vista.txt_estado_trabajador.getText());
122
123                 Trabajador trabajador = new Trabajador(nombres, apellido_paterno, apellido_materno, tipo_documento, num_documento, celular, email, sueldo, acceso, login, passw
124                 int listo = this.dao.agregar(trabajador);
125                 if(listo == 1){
126                     JOptionPane.showMessageDialog(vista, "Trabajador registrado con éxito");
127                 }else{
128                     JOptionPane.showMessageDialog(vista, "Debe completar todos los campos");
129                 }
130             }
131         }
132     }
133
134     private void listar(JTable tabla){
135         this.vista.btn_editar_trabajador.setEnabled(true);
136         this.vista.btn_eliminar_trabajador.setEnabled(true);
137         this.modeloTabla = (DefaultTableModel)tabla.getModel();
138         List<Trabajador> listarTrabajadores = this.dao.listar();
139         Object[] objeto = new Object[13];
140         if(listarTrabajadores.size()>0){
141             for(int i=0;i<listarTrabajadores.size();i++){
142                 objeto[0] = listarTrabajadores.get(i).getId_trabajador();
143                 objeto[1] = listarTrabajadores.get(i).getNombre();
144                 objeto[2] = listarTrabajadores.get(i).getApe_paterno();
```

Escribe aquí para buscar Escritorio ESP ES 21:01 10/11/2020

Ilustración 204: Control Trabajador - 5

```
        }

    }

    private boolean validarNombre(){
        boolean nombres = val.validacionNomApe(vista.txt_nombre_trabajador.getText());
        if(nombres == false){
            JOptionPane.showMessageDialog(vista, "Ingrese un nombre valido");
            vista.txt_nombre_trabajador.setText("");
            return false;
        }else{
            return true;
        }
    }

    private boolean validarApePaterno(){
        var = val.validacionNomApe(vista.txt_ape_paterno_trabajador.getText());
        if(var == false){
            JOptionPane.showMessageDialog(vista, "Ingrese un apellido valido");
            vista.txt_ape_paterno_trabajador.setText("");
            return false;
        }else{
            return true;
        }
    }

    private boolean validarApeMaterno(){
        var = val.validacionNomApe(vista.txt_ape_materno_trabajador.getText());
        if(var == false){
            JOptionPane.showMessageDialog(vista, "Ingrese un apellido valido");
            vista.txt_ape_materno_trabajador.setText("");
            return false;
        }else{
```

Ilustración 205: Control Trabajador - 6

```
            return true;
        }
    }

    private boolean validarTipo_documento(){
        var = val.validacionTipoDocumento(vista.txt_tipo_documento_trabajador.getText());
        if(var == false){
            JOptionPane.showMessageDialog(vista, "Solo puede ingresar DNI o Pasaporte");
            vista.txt_tipo_documento_trabajador.setText("");
            return false;
        }else{
            return true;
        }
    }

    private boolean validarNumero_documento(){
        var = val.validacionNumeros(vista.txt_num_documento_trabajador.getText());
        if(var == false){
            JOptionPane.showMessageDialog(vista, "Ingrese un numero de documento valido");
            vista.txt_num_documento_trabajador.setText("");
            return false;
        }else{
            return true;
        }
    }

    private boolean validarCelular(){
        var = val.validacionNumeros(vista.txt_celular_trabajador.getText());
        if(var == false){
            JOptionPane.showMessageDialog(vista, "Ingrese un numero de celular valido");
            vista.txt_celular_trabajador.setText("");
            return false;
        }else{
```

Ilustración 206: Control Trabajador - 7

```
            return true;
        }
    }

    private boolean validarEmail(){
        var = val.validacionEmail(vista.txt_email_trabajador.getText());
        if(var == false){
            JOptionPane.showMessageDialog(vista, "Ingrese un email valido");
            vista.txt_email_trabajador.setText("");
            return false;
        }else{
            return true;
        }
    }

    private boolean validarSueldo(){
        var = val.validacionNumeros(vista.txt_sueldo_trabajador.getText());
        if(var == false){
            JOptionPane.showMessageDialog(vista, "Ingrese un sueldo valido");
            vista.txt_sueldo_trabajador.setText("");
            return false;
        }else{
            return true;
        }
    }

    private boolean validarAcceso(){
        var = val.validacionAcceso(vista.txt_acceso_trabajador.getText());
        if(var == false){
            JOptionPane.showMessageDialog(vista, "Solo puede ingresar Administrador o Recepcion");
            vista.txt_acceso_trabajador.setText("");
            return false;
        }else{
```

Ilustración 207: Control Trabajador - 8

```
        }else{
            return true;
        }
    }

    private boolean validarEstado(){
        var = val.validacionEstado(vista.txt_estado_trabajador.getText());
        if(var == false){
            JOptionPane.showMessageDialog(vista, "Solo puede ingresar 0->(inactivo) o 1->(activo)");
            vista.txt_estado_trabajador.setText("");
            return false;
        }else{
            return true;
        }
    }

    private boolean validarBuscar(){
        var = val.validacionLetras(vista.txt_buscar_trabajador.getText());
        if(var == false){
            JOptionPane.showMessageDialog(vista, "Ingrese un nombre de trabajador valido");
            vista.txt_buscar_trabajador.setText("");
            return false;
        }else{
            return true;
        }
    }
}
```

Ilustración 208: Control Trabajador - 9

Ilustración 209: Control Trabajador - 10

Ilustración 210: Control Trabajador - 11

```
46          Object[] objeto = new Object[13];
47          if(listarTrabajadores.size()>0){
48              for(int i=0;i<listarTrabajadores.size();i++){
49                  objeto[0] = listarTrabajadores.get(i).getId_trabajador();
50                  objeto[1] = listarTrabajadores.get(i).getNombre();
51                  objeto[2] = listarTrabajadores.get(i).getApe_paterno();
52                  objeto[3] = listarTrabajadores.get(i).getApe_materno();
53                  objeto[4] = listarTrabajadores.get(i).getTipo_documento();
54                  objeto[5] = listarTrabajadores.get(i).getNum_documento();
55                  objeto[6] = listarTrabajadores.get(i).getCelular();
56                  objeto[7] = listarTrabajadores.get(i).getEmail();
57                  objeto[8] = listarTrabajadores.get(i).getSueldo();
58                  objeto[9] = listarTrabajadores.get(i).getAcceso();
59                  objeto[10] = listarTrabajadores.get(i).getLogin();
60                  objeto[11] = listarTrabajadores.get(i).getPassword();
61                  objeto[12] = listarTrabajadores.get(i).getEstado();
62                  modeloTabla.addRow(objeto);
63              }
64              vista.tb_vista_trabajador.setModel(modeloTabla);
65          }else{
66              JOptionPane.showMessageDialog(vista, "No hay trabajdores para listar");
67          }
68
69      }
70
71      private void limpiarTabla (){
72          for(int i=0;i<vista.tb_vista_trabajador.getRowCount();i++){
73              modeloTabla.removeRow(i);
74              i = i-1;
75          }
76      }
77
78      private boolean validarBuscar(){
```

Ilustración 211: Control Trabajador - 12

```
64              vista.tb_vista_trabajador.setModel(modeloTabla);
65          }else{
66              JOptionPane.showMessageDialog(vista, "No hay trabajdores para listar");
67          }
68
69      }
70
71      private void limpiarTabla (){
72          for(int i=0;i<vista.tb_vista_trabajador.getRowCount();i++){
73              modeloTabla.removeRow(i);
74              i = i-1;
75          }
76      }
77
78      private boolean validarBuscar(){
79          var = val.validacionLetras(vista.txt_buscar_trabajador.getText());
80          if(var == false){
81              JOptionPane.showMessageDialog(vista, "Ingrese un nombre de trabajador valido");
82              vista.txt_buscar_trabajador.setText("");
83              return false;
84          }else{
85              return true;
86          }
87      }
88  }
```

Ilustración 212: Control Trabajador - 13

Modelo del Trabajador

Ilustración 213: Modelo Trabajador - 1

Ilustración 214: Modelo Trabajador - 2

```
        this.sueldo = sueldo;
        this.acceso = acceso;
        this.login = login;
        this.password = password;
        this.estado = estado;
    }

    public int getId_trabajador() {
        return id_trabajador;
    }

    public void setId_trabajador(int id_trabajador) {
        this.id_trabajador = id_trabajador;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getApe_paterno() {
        return ape_paterno;
    }

    public void setApe_paterno(String ape_paterno) {
        this.ape_paterno = ape_paterno;
    }

    public String getApe_materno() {
        return ape_materno;
```

Ilustración 215: Modelo Trabajador - 3

```
    }

    public void setApe_materno(String ape_materno) {
        this.ape_materno = ape_materno;
    }

    public String getTipo_documento() {
        return tipo_documento;
    }

    public void setTipo_documento(String tipo_documento) {
        this.tipo_documento = tipo_documento;
    }

    public String getNum_documento() {
        return num_documento;
    }

    public void setNum_documento(String num_documento) {
        this.num_documento = num_documento;
    }

    public String getCelular() {
        return celular;
    }

    public void setCelular(String celular) {
        this.celular = celular;
    }

    public String getEmail() {
        return email;
```

Ilustración 216: Modelo Trabajador - 4

Ilustración 217: Modelo Trabajador - 5

Ilustración 218: Modelo Trabajador - 6

Ilustración 219: Modelo Trabajador - 7

Ilustración 220: Modelo Trabajador - 8

Ilustración 221: Modelo Trabajador - 9

Diagrama de secuencia del caso de uso Registrar habitaciones

Ilustración 222: Diagrama de secuencia - Registrar habitaciones

Vista de las Habitaciones


```
xml versión = " 1.0 " codificación = " UTF-8 " ?>

/ersión de formulario  = " 1.5 " maxVersion = " 1.9 " type = " org.netbeans.modules.form.forminfo.JInternalFrameFormInfo " >
: Propiedades >
  < Nombre de propiedad  = " closable " type = " boolean " value = " true " />
  < Nombre de propiedad  = " defaultCloseOperation " type = " int " value = " 3 " />
  < Nombre de propiedad  = " iconifiable " type = " boolean " value = " true " />
  < Nombre de propiedad  = " maximizable " type = " boolean " value = " true " />
:/ Propiedades >
: Propiedades sintéticas >
  < SyntheticProperty  name = " formSizePolicy "  type = " int "  value = " 1 " />
:/ Propiedades sintéticas >
: AuxValues >
  < AuxValue  name = " FormSettings_autoResourcing "  type = " java.lang.Integer "  value = " 0 " />
  < AuxValue  name = " FormSettings_autoSetComponentName "  type = " java.lang.Boolean "  value = " false " />
  < AuxValue  name = " FormSettings_generateFQN "  type = " java.lang.Boolean "  value = " true " />
  < AuxValue  name = " FormSettings_generateMnemonicsCode "  type = " java.lang.Boolean "  value = " false " />
  < AuxValue  name = " FormSettings_i18nAutoMode "  type = " java.lang.Boolean "  value = " false " />
  < AuxValue  name = " FormSettings_layoutCodeTarget "  type = " java.lang.Integer "  value = " 1 " />
  < AuxValue  name = " FormSettings_listenerGenerationStyle "  type = " java.lang.Integer "  value = " 0 " />
  < AuxValue  name = " FormSettings_variablesLocal "  type = " java.lang.Boolean "  value = " false " />
  < AuxValue  name = " FormSettings_variablesModifier "  type = " java.lang.Integer "  value = " 2 " />
:/ AuxValues >

: Diseño >
  < DimensionLayout  dim = " 0 " >
    < Tipo de grupo  = " 103 " groupAlignment = " 0 " atributos = " 0 " >
        < Tipo de grupo  = " 102 " alineación = " 0 " atributos = " 0 " >
            < EmptySpace  max = " -2 "  atributos = " 0 " />
            < Tipo de grupo  = " 103 " groupAlignment = " 0 " atributos = " 0 " >
                < Tipo de grupo  = " 102 " atributos = " 0 " >
                    < ID de componente  = " jLabel1 " min = " -2 " pref = " 127 " max = " -2 " atributos = " 0 " />
                    < EmptySpace  min = " 0 "  pref = " 0 "  max = " 32767 "  atributos = " 0 " />
                </ Grupo >
                < Tipo de grupo  = " 102 " atributos = " 0 " >
                    < ID de componente  = " jPanel1 " min = " -2 " max = " -2 " atributos = " 0 " />
                    < EmptySpace  max = " -2 "  atributos = " 0 " />
                    < Tipo de grupo  = " 103 " groupAlignment = " 0 " atributos = " 0 " >
                        < ID de componente  = " jPanel2 " max = " 32767 " atributos = " 0 " />
                        < ID de componente  = " jPanel3 " max = " 32767 " atributos = " 0 " />
                    </ Grupo >
                </ Grupo >
            </ Grupo >
            < EmptySpace  max = " -2 "  atributos = " 0 " />
        </ Grupo >
    </ Grupo >
  </ DimensionLayout >
  < DimensionLayout  dim = " 1 " >
    < Tipo de grupo  = " 103 " groupAlignment = " 0 " atributos = " 0 " >
        < Tipo de grupo  = " 102 " alineación = " 0 " atributos = " 0 " >
            < EmptySpace  pref = " 26 "  max = " 32767 "  atributos = " 0 " />
            < ID de componente  = " jLabel1 " min = " -2 " max = " -2 " atributos = " 0 " />
            < EmptySpace  type = " no relacionados "  max = " -2 "  atributos = " 0 " />
            < Tipo de grupo  = " 103 " groupAlignment = " 0 " atributos = " 0 " >
                < ID de componente  = " jPanel1 " min = " -2 " max = " -2 " atributos = " 0 " />
                < Tipo de grupo  = " 102 " alineación = " 0 " atributos = " 0 " >
                    < ID de componente  = " jPanel3 " min = " -2 " max = " -2 " atributos = " 0 " />
                    < EmptySpace  max = " -2 "  atributos = " 0 " />
                    < ID de componente  = " jPanel2 " min = " -2 " max = " -2 " atributos = " 0 " />
                </ Grupo >
            </ Grupo >
        </ Grupo >
    </ Grupo >
  </ DimensionLayout >
:/ Diseño >
: Subcomponentes >
  < Componente  class = " javax.swing.JLabel "  name = " jLabel1 " >
    < Propiedades >
      < Nombre de propiedad  = " font " type = " java.awt.Font " editor = " org.netbeans.beaninfo.editors.FontEditor " >
        < Font  name = " Tahoma "  size = " 18 "  style = " 3 " />
      </ Propiedad >
      < Nombre de propiedad  = " texto " tipo = " java.lang.String " value = " Habitaciones " />
    </ Propiedades >
  </ Componente >
  < Container  class = " javax.swing.JPanel "  name = " jPanel1 " >
    < Propiedades >
      < Nombre de propiedad  = " background " type = " java.awt.Color " editor = " org.netbeans.beaninfo.editors.ColorEditor " >
        < Color  blue = " e6 "  green = " ed "  red = " fc "  type = " rgb " />
      </ Propiedad >
      < Nombre de propiedad  = " border " type = " javax.swing.border.Border " editor = " org.netbeans.modules.form.editors2.Borde
        < Border  info = " org.netbeans.modules.form.compat2.border.TitledBorderInfo " >
          < TitledBorder  title = " Registro de Habitaciones " />
        </ Borde >
```


Ilustración 223: Vista Habitaciones - 1

```
4G+ 70% 15:25
</ Propiedades >

< Diseño >
  < DimensionLayout  dim = " 0 " >
    < Tipo de grupo  = " 103 " groupAlignment = " 0 " atributos = " 0 " >
        < Tipo de grupo  = " 102 " atributos = " 0 " >
            < EmptySpace  max = " -2 "  atributos = " 0 " />
            < Tipo de grupo  = " 103 " groupAlignment = " 0 " atributos = " 0 " >
                < ID de componente  = " jLabel2 " alineación = " 0 " min = " -2 " max = " -2 " atributos = " 0 " />
                < ID de componente  = " jLabel3 " alineación = " 0 " min = " -2 " max = " -2 " atributos = " 0 " />
                < ID de componente  = " jLabel4 " alineación = " 0 " min = " -2 " max = " -2 " atributos = " 0 " />
                < ID de componente  = " jLabel6 " alineación = " 0 " min = " -2 " max = " -2 " atributos = " 0 " />
                < ID de componente  = " jLabel5 " alineación = " 0 " min = " -2 " max = " -2 " atributos = " 0 " />
                < ID de componente  = " jLabel8 " alineación = " 0 " min = " -2 " max = " -2 " atributos = " 0 " />
            </ Grupo >
            < EmptySpace  min = " -2 "  pref = " 27 "  max = " -2 "  atributos = " 0 " />
            < Tipo de grupo  = " 103 " groupAlignment = " 0 " atributos = " 0 " >
                < ID de componente  = " txt_estado_hab " min = " -2 " pref = " 125 " max = " -2 " atributos = " 0 " />
                < Tipo de grupo  = " 103 " groupAlignment = " 0 " max = " -2 " atributos = " 0 " >
                    < ID de componente  = " txt_id_habitacion " max = " 32767 " atributos = " 0 " />
                    < ID de componente  = " txt_numero_hab " max = " 32767 " atributos = " 0 " />
                    < ID de componente  = " txt_piso_hab " alineación = " 0 " max = " 32767 " atributos = " 0 " />
                    < ID de componente  = " txt_precio_hab " alineación = " 0 " max = " 32767 " atributos = " 0 " />
                    < ID de componente  = " txt_tipo_hab " pref = " 125 " max = " 32767 " atributos = " 0 " />
                </ Grupo >
            </ Grupo >
            < EmptySpace  min = " 0 "  pref = " 0 "  max = " 32767 "  atributos = " 0 " />
        </ Grupo >
        < Tipo de grupo  = " 102 " atributos = " 0 " >
            < ID de componente  = " btn_registrar_hab " min = " -2 " max = " -2 " atributos = " 0 " />
            < EmptySpace  type = " separado "  max = " -2 "  atributos = " 0 " />
            < ID de componente  = " btn_limpiar_campos_hab " min = " -2 " max = " -2 " atributos = " 0 " />
            < EmptySpace  type = " separado "  max = " -2 "  atributos = " 0 " />
            < ID de componente  = " btn_nuevo_hab " min = " -2 " max = " -2 " atributos = " 0 " />
            < EmptySpace  max = " 32767 "  atributos = " 0 " />
        </ Grupo >
    </ Grupo >
  </ DimensionLayout >
  < DimensionLayout  dim = " 1 " >
    < Tipo de grupo  = " 103 " groupAlignment = " 0 " atributos = " 0 " >
        < Tipo de grupo  = " 102 " alineación = " 0 " atributos = " 0 " >
            < EmptySpace  min = " -2 "  pref = " 14 "  max = " -2 "  atributos = " 0 " />
            < Tipo de grupo  = " 103 " groupAlignment = " 3 " atributos = " 0 " >
                < ID de componente  = " jLabel8 " alineación = " 3 " min = " -2 " max = " -2 " atributos = " 0 " />
                < ID de componente  = " txt_id_habitacion " alineación = " 3 " min = " -2 " max = " -2 " atributos = " 0 " />
            </ Grupo >
            < EmptySpace  min = " -2 "  pref = " 30 "  max = " -2 "  atributos = " 0 " />
            < Tipo de grupo  = " 103 " groupAlignment = " 3 " atributos = " 0 " >
                < ID de componente  = " jLabel2 " alineación = " 3 " min = " -2 " max = " -2 " atributos = " 0 " />
                < ID de componente  = " txt_numero_hab " alineación = " 3 " min = " -2 " max = " -2 " atributos = " 0 " />
            </ Grupo >
            < EmptySpace  min = " -2 "  pref = " 29 "  max = " -2 "  atributos = " 0 " />
            < Tipo de grupo  = " 103 " groupAlignment = " 3 " atributos = " 0 " >
                < ID de componente  = " jLabel3 " alineación = " 3 " min = " -2 " max = " -2 " atributos = " 0 " />
                < ID de componente  = " txt_piso_hab " alineación = " 3 " min = " -2 " max = " -2 " atributos = " 0 " />
            </ Grupo >
            < EmptySpace  min = " -2 "  pref = " 30 "  max = " -2 "  atributos = " 0 " />
            < Tipo de grupo  = " 103 " groupAlignment = " 3 " atributos = " 0 " >
                < ID de componente  = " txt_precio_hab " alineamiento = " 3 " min = " -2 " max = " -2 " atributos = " 0 " />
                < ID de componente  = " jLabel4 " alineación = " 3 " min = " -2 " max = " -2 " atributos = " 0 " />
            </ Grupo >
            < EmptySpace  min = " -2 "  pref = " 34 "  max = " -2 "  atributos = " 0 " />
            < Tipo de grupo  = " 103 " groupAlignment = " 3 " atributos = " 0 " >
                < ID de componente  = " txt_tipo_hab " alineación = " 3 " min = " -2 " max = " -2 " atributos = " 0 " />
                < ID de componente  = " jLabel6 " alineación = " 3 " min = " -2 " max = " -2 " atributos = " 0 " />
            </ Grupo >
            < EmptySpace  min = " -2 "  pref = " 34 "  max = " -2 "  atributos = " 0 " />
            < Tipo de grupo  = " 103 " groupAlignment = " 3 " atributos = " 0 " >
                < ID de componente  = " txt_estado_hab " alineación = " 3 " min = " -2 " max = " -2 " atributos = " 0 " />
                < ID de componente  = " jLabel5 " alineación = " 3 " min = " -2 " max = " -2 " atributos = " 0 " />
            </ Grupo >
            < EmptySpace  pref = " 39 "  max = " 32767 "  atributos = " 0 " />
            < Tipo de grupo  = " 103 " groupAlignment = " 3 " atributos = " 0 " >
                < ID de componente  = " btn_registrar_hab " alineación = " 3 " min = " -2 " max = " -2 " atributos = " 0 " />
                < ID de componente  = " btn_limpiar_campos_hab " alineamiento = " 3 " min = " -2 " max = " -2 " atributos = "
                < ID de componente  = " btn_nuevo_hab " alineación = " 3 " min = " -2 " max = " -2 " atributos = " 0 " />
            </ Grupo >
            < EmptySpace  min = " -2 "  pref = " 35 "  max = " -2 "  atributos = " 0 " />
        </ Grupo >
    </ Grupo >
  </ DimensionLayout >
</ Diseño >
< Subcomponentes >
  < Component  class = " javax.swing.JTextField "  name = " txt_id_habitacion " >
```


Ilustración 224: Vista Habitaciones - 2

```
< Component  class = " javax.swing.JTextField "  name = " txt_id_habitacion " >
  < Propiedades >
    < Nombre de propiedad  = " editable " type = " boolean " value = " false " />
  </ Propiedades >
  < AuxValues >
    < AuxValue  name = " JavaCodeGenerator_VariableModifier "  type = " java.lang.Integer "  value = " 1 " />
  </ AuxValues >
</ Componente >
< Component  class = " javax.swing.JLabel "  name = " jLabel2 " >
  < Propiedades >
    < Nombre de propiedad  = " font " type = " java.awt.Font " editor = " org.netbeans.beaninfo.editors.FontEditor " >
      < Font  name = " Tahoma "  size = " 14 "  style = " 2 " />
    </ Propiedad >
    < Property  name = " text "  type = " java.lang.String "  value = " N & # xfa; mero de habitaci & # xf3; n: " />
  </ Propiedades >
</ Componente >
< Componente  class = " javax.swing.JLabel "  name = " jLabel3 " >
  < Propiedades >
    < Nombre de propiedad  = " font " type = " java.awt.Font " editor = " org.netbeans.beaninfo.editors.FontEditor " >
      < Font  name = " Tahoma "  size = " 14 "  style = " 2 " />
    </ Propiedad >
    < Nombre de propiedad  = " texto " tipo = " java.lang.String " value = " Piso: " />
  </ Propiedades >
</ Componente >
< Componente  class = " javax.swing.JLabel "  name = " jLabel4 " >
  < Propiedades >
    < Nombre de propiedad  = " font " type = " java.awt.Font " editor = " org.netbeans.beaninfo.editors.FontEditor " >
      < Font  name = " Tahoma "  size = " 14 "  style = " 2 " />
    </ Propiedad >
    < Nombre de propiedad  = " text " type = " java.lang.String " value = " Precio: " />
  </ Propiedades >
</ Componente >
< Componente  class = " javax.swing.JLabel "  name = " jLabel5 " >
  < Propiedades >
    < Nombre de propiedad  = " font " type = " java.awt.Font " editor = " org.netbeans.beaninfo.editors.FontEditor " >
      < Font  name = " Tahoma "  size = " 14 "  style = " 2 " />
    </ Propiedad >
    < Nombre de propiedad  = " texto " tipo = " java.lang.String " value = " Estado: " />
  </ Propiedades >
</ Componente >
< Componente  class = " javax.swing.JLabel "  name = " jLabel6 " >
  < Propiedades >
    < Nombre de propiedad  = " font " type = " java.awt.Font " editor = " org.netbeans.beaninfo.editors.FontEditor " >
      < Font  name = " Tahoma "  size = " 14 "  style = " 2 " />
    </ Propiedad >
    < Property  name = " text "  type = " java.lang.String "  value = " Tipo de habitaci & # xf3; n: " />
  </ Propiedades >
</ Componente >
< Component  class = " javax.swing.JTextField "  name = " txt_numero_hab " >
  < AuxValues >
    < AuxValue  name = " JavaCodeGenerator_VariableModifier "  type = " java.lang.Integer "  value = " 1 " />
  </ AuxValues >
</ Componente >
< Component  class = " javax.swing.JTextField "  name = " txt_precio_hab " >
  < AuxValues >
    < AuxValue  name = " JavaCodeGenerator_VariableModifier "  type = " java.lang.Integer "  value = " 1 " />
  </ AuxValues >
</ Componente >
< Component  class = " javax.swing.JButton "  name = " btn_registrar_hab " >
  < Propiedades >
    < Nombre de propiedad  = " icon " type = " javax.swing.Icon " editor = " org.netbeans.modules.form.editors2.IconEditor "
      < Image  iconType = " 3 "  name = " /views/icon/agregar.png " />
    </ Propiedad >
    < Nombre de propiedad  = " text " type = " java.lang.String " value = " Registrar " />
  </ Propiedades >
  < AuxValues >
    < AuxValue  name = " JavaCodeGenerator_VariableModifier "  type = " java.lang.Integer "  value = " 1 " />
  </ AuxValues >
</ Componente >
< Componente  class = " javax.swing.JButton "  name = " btn_limpiar_campos_hab " >
  < Propiedades >
    <Property name="icon" type="javax.swing.Icon" editor="org.netbeans.modules.form.editors2.IconEditor">
      <Image iconType="3" name="/views/icon/limpiar.png"/>
    </Property>
    <Property name="text" type="java.lang.String" value="Limpiar Campos"/>
  </Properties>
  <AuxValues>
    <AuxValue name="JavaCodeGenerator_VariableModifier" type="java.lang.Integer" value="1"/>
  </AuxValues>
</Component>
<Component class="javax.swing.JTextField" name="txt_piso_hab">
  <AuxValues>
    <AuxValue name="JavaCodeGenerator_VariableModifier" type="java.lang.Integer" value="1"/>
  </AuxValues>
```


Ilustración 225: Vista Habitaciones - 3

```
      < AuxValues >
        < AuxValue  name = " JavaCodeGenerator_VariableModifier "  type = " java.lang.Integer "  value = " 1 " />
      </ AuxValues >
    </ Componente >
    < Component  class = " javax.swing.JLabel "  name = " jLabel8 " >
      < Propiedades >
        < Nombre de propiedad  = " font " type = " java.awt.Font " editor = " org.netbeans.beaninfo.editors.FontEditor " >
          < Font  name = " Tahoma "  size = " 14 "  style = " 2 " />
        </ Propiedad >
        < Nombre de propiedad  = " text " type = " java.lang.String " value = " C & # xf3; digo: " />
      </ Propiedades >
    </ Componente >
  </ Subcomponentes >
</ Contenedor >
< Container  class = " javax.swing.JPanel "  name = " jPanel2 " >
  < Propiedades >
    < Nombre de propiedad  = " background " type = " java.awt.Color " editor = " org.netbeans.beaninfo.editors.ColorEditor " >
      < Color  azul = " ff "  verde = " ff "  rojo = " ff "  tipo = " rgb " />
    </ Propiedad >
    < Nombre de propiedad  = " border " type = " javax.swing.border.Border " editor = " org.netbeans.modules.form.editors2.Borde
      < Border  info = " org.netbeans.modules.form.compat2.border.TitledBorderInfo " >
        < TitledBorder  title = " Listado de Habitaciones " />
      </ Borde >
    </ Propiedad >
  </ Propiedades >

  < Diseño >
    < DimensionLayout  dim = " 0 " >
      < Tipo de grupo  = " 103 " groupAlignment = " 0 " atributos = " 0 " >
          < Tipo de grupo  = " 102 " alineación = " 0 " atributos = " 0 " >
              < EmptySpace  max = " -2 "  atributos = " 0 " />
              < ID de componente  = " jScrollPane1 " max = " 32767 " atributos = " 0 " />
              < EmptySpace  max = " -2 "  atributos = " 0 " />
          </ Grupo >
      </ Grupo >
    </ DimensionLayout >
    < DimensionLayout  dim = " 1 " >
      < Tipo de grupo  = " 103 " groupAlignment = " 0 " atributos = " 0 " >
          < Tipo de grupo  = " 102 " alineación = " 0 " atributos = " 0 " >
              < EmptySpace  max = " 32767 "  atributos = " 0 " />
              < ID de componente  = " jScrollPane1 " min = " -2 " pref = " 312 " max = " -2 " atributos = " 0 " />
          </ Grupo >
      </ Grupo >
    </ DimensionLayout >
  </ Diseño >
  < Subcomponentes >
    < Container  class = " javax.swing.JScrollPane "  name = " jScrollPane1 " >
      < AuxValues >
        < AuxValue  name = " autoScrollPane "  type = " java.lang.Boolean "  value = " true " />
      </ AuxValues >

      < Layout  class = " org.netbeans.modules.form.compat2.layouts.support.JScrollPaneSupportLayout " />
      < Subcomponentes >
        < Componente  class = " javax.swing.JTable "  name = " tb_datos_hab " >
          < Propiedades >
            < Nombre de propiedad  = " modelo " tipo = " javax.swing.table.TableModel " editor = " org.netbeans.modules.form.edi
              < Table  columnCount = " 6 "  rowCount = " 0 " >
                < Column  editable = " true "  title = " C & # xf3; digo "  type = " java.lang.Object " />
                < Column  editable = " true "  title = " N & # xfa; mero "  type = " java.lang.Object " />
                < Column  editable = " true "  title = " Piso "  type = " java.lang.Object " />
                < Column  editable = " true "  title = " Precio "  type = " java.lang.Object " />
                < Column  editable = " true "  title = " Tipo de habitaci & # xf3; n "  type = " java.lang.Object " />
                < Column  editable = " true "  title = " Estado "  type = " java.lang.Object " />
              </ Tabla >
            </ Propiedad >
            < Nombre de propiedad  = " columnModel " type = " javax.swing.table.TableColumnModel " editor = " org.netbeans.modul
              < TableColumnModel  selectionModel = " 0 " >
                < Columna  maxWidth = " -1 "  minWidth = " -1 "  prefWidth = " -1 "  resizable = " true " >
                  < Título />
                  < Editor />
                  < Procesador />
                </ Columna >
                < Columna  maxWidth = " -1 "  minWidth = " -1 "  prefWidth = " -1 "  resizable = " true " >
                  < Título />
                  < Editor />
                  < Procesador />
                </ Columna >
                < Columna  maxWidth = " -1 "  minWidth = " -1 "  prefWidth = " -1 "  resizable = " true " >
                  < Título />
                  < Editor />
                  < Procesador />
                </ Columna >
                < Columna  maxWidth = " -1 "  minWidth = " -1 "  prefWidth = " -1 "  resizable = " true " >
                  < Título />
```


Ilustración 226: Vista Habitaciones - 4

```
                < Editor />
                < Procesador />
              </ Columna >
              < Columna  maxWidth = " -1 "  minWidth = " -1 "  prefWidth = " -1 "  resizable = " true " >
                < Título />
                < Editor />
                < Procesador />
              </ Columna >
              < Columna  maxWidth = " -1 "  minWidth = " -1 "  prefWidth = " -1 "  resizable = " true " >
                < Título />
                < Editor />
                < Procesador />
              </ Columna >
              < Columna  maxWidth = " -1 "  minWidth = " -1 "  prefWidth = " -1 "  resizable = " true " >
                < Título />
                < Editor />
                < Procesador />
              </ Columna >
            </ TableColumnModel >
          </ Propiedad >
          <Property name="tableHeader" type="javax.swing.table.JTableHeader" editor="org.netbeans.modules.form.editors2.JTable
            <TableHeader reorderingAllowed="true" resizingAllowed="true"/>
          </Property>
        </Properties>
        <AuxValues>
          <AuxValue name="JavaCodeGenerator_VariableModifier" type="java.lang.Integer" value="1"/>
        </AuxValues>
      </Component>
    </SubComponents>
  </Container>
</SubComponents>
</Container>
<Container class="javax.swing.JPanel" name="jPanel3">
  <Properties>
    <Property name="background" type="java.awt.Color" editor="org.netbeans.beaninfo.editors.ColorEditor">
      <Color blue="c3" green="ff" red="ff" type="rgb"/>
    </Property>
    <Property name="border" type="javax.swing.border.Border" editor="org.netbeans.modules.form.editors2.BorderEditor">
      <Border info="org.netbeans.modules.form.compat2.border.TitledBorderInfo">
        <TitledBorder title="Opciones"/>
      </Border>
    </Property>
  </Properties>

  <Layout>
    <DimensionLayout dim="0">
      <Group type="103" groupAlignment="0" attributes="0">
          < Tipo de grupo  = " 102 " alineación = " 0 " atributos = " 0 " >
              < EmptySpace  max = " -2 "  atributos = " 0 " />
              < ID de componente  = " jLabel7 " min = " -2 " max = " -2 " atributos = " 0 " />
              < EmptySpace  max = " -2 "  atributos = " 0 " />
              <Component id="txt_buscar_hab" min="-2" pref="95" max="-2" attributes="0"/>
              <EmptySpace max="-2" attributes="0"/>
              <Component id="btn_buscar_hab" min="-2" max="-2" attributes="0"/>
              <EmptySpace min="-2" pref="60" max="-2" attributes="0"/>
              <Component id="btn_editar_hab" min="-2" pref="101" max="-2" attributes="0"/>
              <EmptySpace type="separate" max="-2" attributes="0"/>
              < ID de componente  = " btn_actualizar_hab " min = " -2 " max = " -2 " atributos = " 0 " />
              < EmptySpace  type = " separado "  max = " -2 "  atributos = " 0 " />
              <Component id="btn_eliminar_hab" pref="111" max="32767" attributes="0"/>
              <EmptySpace type="separate" max="-2" attributes="0"/>
              <Component id="btn_listar_hab" min="-2" pref="101" max="-2" attributes="0"/>
              <EmptySpace min="-2" pref="21" max="-2" attributes="0"/>
          </Group>
      </Group>
    </DimensionLayout>
    <DimensionLayout dim="1">
      <Group type="103" groupAlignment="0" attributes="0">
          <Group type="102" alignment="1" attributes="0">
              <EmptySpace max="32767" attributes="0"/>
              < Tipo de grupo  = " 103 " groupAlignment = " 3 " atributos = " 0 " >
                  < ID de componente  = " jLabel7 " alineación = " 3 " min = " -2 " max = " -2 " atributos = " 0 " />
                  <Component id="txt_buscar_hab" alignment="3" min="-2" max="-2" attributes="0"/>
                  < ID de componente  = " btn_buscar_hab " alineación = " 3 " min = " -2 " max = " -2 " atributos = " 0 " />
                  <Component id="btn_editar_hab" alignment="3" min="-2" max="-2" attributes="0"/>
                  <Component id="btn_actualizar_hab" alignment="3" min="-2" max="-2" attributes="0"/>
                  <Component id="btn_eliminar_hab" alignment="3" min="-2" max="-2" attributes="0"/>
                  <Component id="btn_listar_hab" alignment="3" min="-2" max="-2" attributes="0"/>
              </Group>
              <EmptySpace max="-2" attributes="0"/>
          </Group>
      </Group>
    </DimensionLayout>
  </Layout>
```

Ilustración 227: Vista Habitaciones - 5

```
                < ID de componente  = " btn_listar_hab " alineamiento = " 3 " min = " -2 " max = " -2 " atributos = " 0 " />
            </ Grupo >
            < EmptySpace  max = " -2 "  atributos = " 0 " />
        </ Grupo >
    </ Grupo >
  </ DimensionLayout >
</ Diseño >
< Subcomponentes >
  < Componente  class = " javax.swing.JLabel "  name = " jLabel7 " >
    < Propiedades >
      < Property  name = " text "  type = " java.lang.String "  value = " Buscar por n & # xfa; mero: " />
    </ Propiedades >
  </ Componente >
  < Component  class = " javax.swing.JTextField "  name = " txt_buscar_hab " >
    < Eventos >
      < EventHandler  event = " actionPerformed "  listener = " java.awt.event.ActionListener "  parameters = " java.awt.event
    </ Eventos >
    < AuxValues >
      < AuxValue  name = " JavaCodeGenerator_VariableModifier "  type = " java.lang.Integer "  value = " 1 " />
    </ AuxValues >
  </ Componente >
  < Componente  class = " javax.swing.JButton "  name = " btn_buscar_hab " >
    < Propiedades >
      < Nombre de propiedad  = " texto " tipo = " java.lang.String " value = " Buscar " />
    </ Propiedades >
    < AuxValues >
      < AuxValue  name = " JavaCodeGenerator_VariableModifier "  type = " java.lang.Integer "  value = " 1 " />
    </ AuxValues >
  </ Componente >
  < Componente  class = " javax.swing.JButton "  name = " btn_editar_hab " >
    < Propiedades >
      < Nombre de propiedad  = " icon " type = " javax.swing.Icon " editor = " org.netbeans.modules.form.editors2.IconEditor "
        < Image  iconType = " 3 "  name = " /views/icon/editar.png " />
      </ Propiedad >
      < Nombre de propiedad  = " texto " tipo = " java.lang.String " value = " Editar " />
      < Nombre de propiedad  = " habilitado " tipo = " booleano " valor = " falso " />
    </ Propiedades >
    < AuxValues >
      < AuxValue  name = " JavaCodeGenerator_VariableModifier "  type = " java.lang.Integer "  value = " 1 " />
    </ AuxValues >
  </ Componente >
  < Componente  class = " javax.swing.JButton "  name = " btn_actualizar_hab " >
    < Propiedades >
      < Nombre de propiedad  = " icon " type = " javax.swing.Icon " editor = " org.netbeans.modules.form.editors2.IconEditor "
        < Image  iconType = " 3 "  name = " /views/icon/actualizar.png " />
      </ Propiedad >
      < Nombre de propiedad  = " texto " tipo = " java.lang.String " value = " Actualizar " />
      < Nombre de propiedad  = " habilitado " tipo = " booleano " valor = " falso " />
    </ Propiedades >
    < AuxValues >
      < AuxValue  name = " JavaCodeGenerator_VariableModifier "  type = " java.lang.Integer "  value = " 1 " />
    </ AuxValues >
  </ Componente >
  < Component  class = " javax.swing.JButton "  name = " btn_eliminar_hab " >
    < Propiedades >
      < Nombre de propiedad  = " icon " type = " javax.swing.Icon " editor = " org.netbeans.modules.form.editors2.IconEditor "
        < Image  iconType = " 3 "  name = " /views/icon/eliminar.png " />
      </ Propiedad >
      < Property  name = " text "  type = " java.lang.String "  value = " Eliminar " />
      < Nombre de propiedad  = " habilitado " tipo = " booleano " valor = " falso " />
    </ Propiedades >
    < AuxValues >
      < AuxValue  name = " JavaCodeGenerator_VariableModifier "  type = " java.lang.Integer "  value = " 1 " />
    </ AuxValues >
  </ Componente >
  < Componente  class = " javax.swing.JButton "  name = " btn_listar_hab " >
    < Propiedades >
      < Nombre de propiedad  = " icon " type = " javax.swing.Icon " editor = " org.netbeans.modules.form.editors2.IconEditor "
        < Image  iconType = " 3 "  name = " /views/icon/listar.png " />
      </ Propiedad >
      < Nombre de propiedad  = " texto " tipo = " java.lang.String " value = " Listar " />
    </ Propiedades >
    < Eventos >
      < EventHandler  event = " actionPerformed "  listener = " java.awt.event.ActionListener "  parameters = " java.awt.event
    </ Eventos >
    < AuxValues >
      < AuxValue  name = " JavaCodeGenerator_VariableModifier "  type = " java.lang.Integer "  value = " 1 " />
    </ AuxValues >
  </ Componente >
</ Subcomponentes >
</ Contenedor >
/ Subcomponentes >
Formulario >
```


Ilustración 228: Vista Habitaciones - 6

```
/ *
 * Para cambiar este encabezado de licencia, elija Encabezados de licencia en Propiedades del proyecto.
* Para cambiar este archivo de plantilla, elija Herramientas | Plantillas
 * y abra la plantilla en el editor.
 * /
 vistas de paquetes ;

/ **
 *
* @autor Diego
 * /
La  clase  pública VistaHabitaciones  extiende  javax.swing . JInternalFrame {

    / **
     * Crea nueva forma VistaHabitaciones
     * /
     VistaHabitaciones públicas () {
        initComponents ();
    }

    / **
     * Este método se llama desde dentro del constructor para inicializar el formulario.
     * ADVERTENCIA: NO modifique este código. El contenido de este método es siempre
     * Regenerado por el Editor de formularios.
     * /
    @SuppressWarnings ( " sin marcar " )
    // <editor-fold defaultstate = "collapsed" desc = "Código generado"> // GEN-BEGIN: initComponents
    private  void  initComponents () {

        jLabel1 =  nuevo  javax.swing . JLabel ();
        jPanel1 =  nuevo  javax.swing . JPanel ();
        txt_id_habitacion =  nuevo  javax.swing . JTextField ();
        jLabel2 =  nuevo  javax.swing . JLabel ();
        jLabel3 =  nuevo  javax.swing . JLabel ();
        jLabel4 =  nuevo  javax.swing . JLabel ();
        jLabel5 =  nuevo  javax.swing . JLabel ();
        jLabel6 =  nuevo  javax.swing . JLabel ();
        txt_numero_hab =  nuevo  javax.swing . JTextField ();
        txt_precio_hab =  nuevo  javax.swing . JTextField ();
        btn_registrar_hab =  nuevo  javax.swing . JButton ();
        btn_limpiar_campos_hab =  nuevo  javax.swing . JButton ();
        txt_piso_hab =  nuevo  javax.swing . JTextField ();
        txt_estado_hab =  nuevo  javax.swing . JTextField ();
        txt_tipo_hab =  nuevo  javax.swing . JTextField ();
        btn_nuevo_hab =  nuevo  javax.swing . JButton ();
        jLabel8 =  nuevo  javax.swing . JLabel ();
        jPanel2 =  nuevo  javax.swing . JPanel ();
        jScrollPane1 =  nuevo  javax.swing . JScrollPane ();
        tb_datos_hab =  nuevo  javax.swing . JTable ();
        jPanel3 =  nuevo  javax.swing . JPanel ();
        jLabel7 =  nuevo  javax.swing . JLabel ();
        txt_buscar_hab =  nuevo  javax.swing . JTextField ();
        btn_buscar_hab =  nuevo  javax.swing . JButton ();
        btn_editar_hab =  nuevo  javax.swing . JButton ();
        btn_actualizar_hab =  nuevo  javax.swing . JButton ();
        btn_eliminar_hab =  nuevo  javax.swing . JButton ();
        btn_listar_hab =  nuevo  javax.swing . JButton ();

        setClosable ( verdadero );
        setDefaultCloseOperation ( javax.swing . WindowConstants . DISPOSE_ON_CLOSE );
        setIconifiable ( verdadero );
        setMaximizable ( verdadero );

        jLabel1 . setFont ( nueva  fuente java.awt . ( " Tahoma " , 3 , 18 )); // NOI18N
        jLabel1 . setText ( " Habitaciones " );

        jPanel1 . setBackground ( nuevo  java.awt . Color ( 252 , 237 , 230 ));
        jPanel1 . setBorder ( javax.swing . BorderFactory . createTitledBorder ( " Registro de Habitaciones " ));

        txt_id_habitacion . setEditable ( falso );

        jLabel2 . setFont ( nueva  fuente java.awt . ( " Tahoma " , 2 , 14 )); // NOI18N
        jLabel2 . setText ( " Número de habitación: " );

        jLabel3 . setFont ( nueva  fuente java.awt . ( " Tahoma " , 2 , 14 )); // NOI18N
        jLabel3 . setText ( " Piso: " );

        jLabel4 . setFont ( nueva  fuente java.awt . ( " Tahoma " , 2 , 14 )); // NOI18N
        jLabel4 . setText ( " Precio: " );

        jLabel5 . setFont ( nueva  fuente java.awt . ( " Tahoma " , 2 , 14 )); // NOI18N
        jLabel5 . setText ( " Estado: " );
```

Ilustración 229: Vista Habitaciones - 7

```
jLabel6 . setFont ( nueva  fuente java.awt . ( " Tahoma " , 2 , 14 )); // NOI18N
jLabel6 . setText ( " Tipo de habitación: " );

btn_registrar_hab . setIcon ( nuevo  javax.swing . ImageIcon (getClass () . getResource ( " /views/icon/agregar.png " ))); // NOI
btn_registrar_hab . setText ( " Registrador " );

btn_limpiar_campos_hab . setIcon ( nuevo  javax.swing . ImageIcon (getClass () . getResource ( " /views/icon/limpiar.png " ))); /
btn_limpiar_campos_hab . setText ( " Limpiar Campos " );

txt_tipo_hab . addActionListener ( nuevo  java.awt.event . ActionListener () {
    public  void  actionPerformed ( java.awt.event . ActionEvent  evt ) {
        txt_tipo_habActionPerformed (evt);
    }
});

btn_nuevo_hab . setIcon ( nuevo  javax.swing . ImageIcon (getClass () . getResource ( " /views/icon/nuevo.png " ))); // NOI18N
btn_nuevo_hab . setText ( " Nuevo " );
btn_nuevo_hab . setEnabled ( falso );

jLabel8 . setFont ( nueva  fuente java.awt . ( " Tahoma " , 2 , 14 )); // NOI18N
jLabel8 . setText ( " Código: " );

javax.swing . GroupLayout jPanel1Layout =  nuevo  javax.swing . GroupLayout (jPanel1);
jPanel1 . setLayout (jPanel1Layout);
jPanel1Layout . setHorizontalGroup (
    jPanel1Layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . LEADING )
    .addGroup (jPanel1Layout . createSequentialGroup ()
        .addContainerGap ()
        .addGroup (jPanel1Layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . LEADING )
            .addComponent (jLabel2)
            .addComponent (jLabel3)
            .addComponent (jLabel4)
            .addComponent (jLabel6)
            .addComponent (jLabel5)
            .addComponent (jLabel8))
        .addGap ( 27 , 27 , 27 )
        .addGroup (jPanel1Layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . LEADING )
            .addComponent (txt_estado_hab, javax.swing . GroupLayout . PREFERRED_SIZE , 125 , javax.swing . GroupLayout . PREFERF
            .addGroup (jPanel1Layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . LEADING , false )
                .addComponent (txt_id_habitacion)
                .addComponent (txt_numero_hab)
                .addComponent (txt_piso_hab)
                .addComponent (txt_precio_hab)
                .addComponent (txt_tipo_hab, javax.swing . GroupLayout . DEFAULT_SIZE , 125 , Short . MAX_VALUE )))
        .addGap ( 0 , 0 , corto . MAX_VALUE ))
    .addGroup (jPanel1Layout . createSequentialGroup ()
        .addComponent (btn_registrar_hab)
        .addGap ( 18 , 18 , 18 )
        .addComponent (btn_limpiar_campos_hab)
        .addGap ( 18 , 18 , 18 )
        .addComponent (btn_nuevo_hab)
        .addContainerGap ( javax.swing . GroupLayout . DEFAULT_SIZE , Short . MAX_VALUE ))
);
jPanel1Layout . setVerticalGroup (
    jPanel1Layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . LEADING )
    .addGroup (jPanel1Layout . createSequentialGroup ()
        .addGap ( 14 , 14 , 14 )
        .addGroup (jPanel1Layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . BASELINE )
            .addComponent (jLabel8)
            .addComponent (txt_id_habitacion, javax.swing . GroupLayout . PREFERRED_SIZE , javax.swing . GroupLayout . DEFAULT_SI
        .addGap ( 30 , 30 , 30 )
        .addGroup (jPanel1Layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . BASELINE )
            .addComponent (jLabel2)
            .addComponent (txt_numero_hab, javax.swing . GroupLayout . PREFERRED_SIZE , javax.swing . GroupLayout . DEFAULT_SIZE
        .addGap ( 29 , 29 , 29 )
        .addGroup (jPanel1Layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . BASELINE )
            .addComponent (jLabel3)
            .addComponent (txt_piso_hab, javax.swing . GroupLayout . PREFERRED_SIZE , javax.swing . GroupLayout . DEFAULT_SIZE ,
        .addGap ( 30 , 30 , 30 )
        .addGroup (jPanel1Layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . BASELINE )
            .addComponent (txt_precio_hab, javax.swing . GroupLayout . PREFERRED_SIZE , javax.swing . GroupLayout . DEFAULT_SIZE
            .addComponent (jLabel4))
        .addGap ( 34 , 34 , 34 )
        .addGroup (jPanel1Layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . BASELINE )
            .addComponent (txt_tipo_hab, javax.swing . GroupLayout . PREFERRED_SIZE , javax.swing . GroupLayout . DEFAULT_SIZE ,
            .addComponent (jLabel6))
        .addGap ( 34 , 34 , 34 )
        .addGroup (jPanel1Layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . BASELINE )
            .addComponent (txt_estado_hab, javax.swing . GroupLayout . PREFERRED_SIZE , javax.swing . GroupLayout . DEFAULT_SIZE
            .addComponent (jLabel5))
        .addPreferredGap ( javax.swing . LayoutStyle . ComponentPlacement . RELATED , 39 , Short . MAX_VALUE )
        .addGroup (jPanel1Layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . BASELINE )
            .addComponent (btn_registrar_hab)
```

Ilustración 230: Vista Habitaciones - 8

```
            .addComponent (btn_limpiar_campos_hab)
            .addComponent (btn_nuevo_hab))
        .addGap ( 35 , 35 , 35 ))
);

jPanel2 . setBackground ( nuevo  java.awt . Color ( 255 , 255 , 255 ));
jPanel2 . setBorder ( javax.swing . BorderFactory . createTitledBorder ( " Listado de Habitaciones " ));

tb_datos_hab . setModel ( nuevo  javax.swing.table . DefaultTableModel (
    nuevo  objeto [] [] {

    },
    nueva  cadena [] {
        " Código " , " Número " , " Piso " , " Precio " , " Tipo de habitación " , " Estado "
    }
));
jScrollPane1 . setViewportView (tb_datos_hab);

javax.swing . GroupLayout jPanel2Layout =  nuevo  javax.swing . GroupLayout (jPanel2);
jPanel2 . setLayout (jPanel2Layout);
jPanel2Layout . setHorizontalGroup (
    jPanel2Layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . LEADING )
    .addGroup (jPanel2Layout . createSequentialGroup ()
        .addContainerGap ()
        .addComponent (jScrollPane1)
        .addContainerGap ())
);
jPanel2Layout . setVerticalGroup (
    jPanel2Layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . LEADING )
    .addGroup (jPanel2Layout . createSequentialGroup ()
        .addContainerGap ( javax.swing . GroupLayout . DEFAULT_SIZE , Short . MAX_VALUE )
        .addComponent (jScrollPane1, javax.swing . GroupLayout . PREFERRED_SIZE , 312 , javax.swing . GroupLayout . PREFERRED_SIZ
);

jPanel3 . setBackground ( nuevo  java.awt . Color ( 255 , 255 , 195 ));
jPanel3 . setBorder ( javax.swing . BorderFactory . createTitledBorder ( " Opciones " ));

jLabel7 . setText ( " Buscar por número: " );

txt_buscar_hab . addActionListener ( nuevo  java.awt.event . ActionListener () {
    public  void  actionPerformed ( java.awt.event . ActionEvent  evt ) {
        txt_buscar_habActionPerformed (evt);
    }
});

btn_buscar_hab . setText ( " Buscar " );

btn_editar_hab . setIcon ( nuevo  javax.swing . ImageIcon (getClass () . getResource ( " /views/icon/editar.png " ))); // NOI18N
btn_editar_hab . setText ( " Editar " );
btn_editar_hab . setEnabled ( falso );

btn_actualizar_hab . setIcon ( nuevo  javax.swing . ImageIcon (getClass () . getResource ( " /views/icon/actualizar.png " ))); //
btn_actualizar_hab . setText ( " Actualizar " );
btn_actualizar_hab . setEnabled ( falso );

btn_eliminar_hab . setIcon ( nuevo  javax.swing . ImageIcon (getClass () . getResource ( " /views/icon/eliminar.png " ))); // NOI
btn_eliminar_hab . setText ( " Eliminar " );
btn_eliminar_hab . setEnabled ( falso );

btn_listar_hab . setIcon ( nuevo  javax.swing . ImageIcon (getClass () . getResource ( " /views/icon/listar.png " ))); // NOI18N
btn_listar_hab . setText ( " Listar " );
btn_listar_hab . addActionListener ( nuevo  java.awt.event . ActionListener () {
    public  void  actionPerformed ( java.awt.event . ActionEvent  evt ) {
        btn_listar_habActionPerformed (evt);
    }
});

javax.swing . GroupLayout jPanel3Layout =  nuevo  javax.swing . GroupLayout (jPanel3);
jPanel3 . setLayout (jPanel3Layout);
jPanel3Layout . setHorizontalGroup (
    jPanel3Layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . LEADING )
    .addGroup (jPanel3Layout . createSequentialGroup ()
        .addContainerGap ()
        .addComponent (jLabel7)
        .addPreferredGap ( javax.swing . LayoutStyle . ComponentPlacement . RELATED )
        .addComponent (txt_buscar_hab, javax.swing . GroupLayout . PREFERRED_SIZE , 95 , javax.swing . GroupLayout . PREFERRED_SI
        .addPreferredGap ( javax.swing . LayoutStyle . ComponentPlacement . RELATED )
        .addComponent (btn_buscar_hab)
        .addGap ( 60 , 60 , 60 )
        .addComponent (btn_editar_hab, javax.swing . GroupLayout . PREFERRED_SIZE , 101 , javax.swing . GroupLayout . PREFERRED_S
        .addGap ( 18 , 18 , 18 )
        .addComponent (btn_actualizar_hab)
        .addGap ( 18 , 18 , 18 )
```

Ilustración 231: Vista Habitaciones - 9

```
        .addGap ( 18 , 18 , 18 )
        .addComponent (btn_eliminar_hab, javax.swing . GroupLayout . DEFAULT_SIZE , 111 , Short . MAX_VALUE )
        .addGap ( 18 , 18 , 18 )
        .addComponent (btn_listar_hab, javax.swing . GroupLayout . PREFERRED_SIZE , 101 , javax.swing . GroupLayout . PREFERRED_S
        .addGap ( 21 , 21 , 21 ))
);
jPanel3Layout . setVerticalGroup (
    jPanel3Layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . LEADING )
    .addGroup ( javax.swing . GroupLayout . Alignment . TRAILING , jPanel3Layout . createSequentialGroup ()
        .addContainerGap ( javax.swing . GroupLayout . DEFAULT_SIZE , Short . MAX_VALUE )
        .addGroup (jPanel3Layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . BASELINE )
            .addComponent (jLabel7)
            .addComponent (txt_buscar_hab, javax.swing . GroupLayout . PREFERRED_SIZE , javax.swing . GroupLayout . DEFAULT_SIZE
            .addComponent (btn_buscar_hab)
            .addComponent (btn_editar_hab)
            .addComponent (btn_actualizar_hab)
            .addComponent (btn_eliminar_hab)
            .addComponent (btn_listar_hab))
        .addContainerGap ())
);

javax.swing . Diseño de GroupLayout =  nuevo  javax.swing . GroupLayout (getContentPane ());
getContentPane () . setLayout (diseño);
diseño . setHorizontalGroup (
    diseño . createParallelGroup ( javax.swing . GroupLayout . Alignment . LEADING )
    .addGroup (diseño . createSequentialGroup ()
        .addContainerGap ()
        .addGroup (layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . LEADING )
            .addGroup (diseño . createSequentialGroup ()
                .addComponent (jLabel1, javax.swing . GroupLayout . PREFERRED_SIZE , 127 , javax.swing . GroupLayout . PREFERRED_
                .addGap ( 0 , 0 , corto . MAX_VALUE ))
            .addGroup (diseño . createSequentialGroup ()
                .addComponent (jPanel1, javax.swing . GroupLayout . PREFERRED_SIZE , javax.swing . GroupLayout . DEFAULT_SIZE , j
                .addPreferredGap ( javax.swing . LayoutStyle . ComponentPlacement . RELATED )
                .addGroup (layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . LEADING )
                    .addComponent (jPanel2, javax.swing . GroupLayout . DEFAULT_SIZE , javax.swing . GroupLayout . DEFAULT_SIZE ,
                    .addComponent (jPanel3, javax.swing . GroupLayout . DEFAULT_SIZE , javax.swing . GroupLayout . DEFAULT_SIZE ,
        .addContainerGap ())
);
diseño . setVerticalGroup (
    diseño . createParallelGroup ( javax.swing . GroupLayout . Alignment . LEADING )
    .addGroup (diseño . createSequentialGroup ()
        .addContainerGap ( 26 , corto . MAX_VALUE )
        .addComponent (jLabel1)
        .addPreferredGap ( javax.swing . LayoutStyle . ComponentPlacement . UNRELATED )
        .addGroup (layout . createParallelGroup ( javax.swing . GroupLayout . Alignment . LEADING )
            .addComponent (jPanel1, javax.swing . GroupLayout . PREFERRED_SIZE , javax.swing . GroupLayout . DEFAULT_SIZE , javax
            .addGroup (diseño . createSequentialGroup ()
                .addComponent (jPanel3, javax.swing . GroupLayout . PREFERRED_SIZE , javax.swing . GroupLayout . DEFAULT_SIZE , j
                .addPreferredGap ( javax.swing . LayoutStyle . ComponentPlacement . RELATED )
                .addComponent (jPanel2, javax.swing . GroupLayout . PREFERRED_SIZE , javax.swing . GroupLayout . DEFAULT_SIZE , j
);

paquete();
* </editor-fold> // GEN-END: initComponents

vate  void  txt_tipo_habActionPerformed ( java.awt.event . ActionEvent  evt ) { // GEN-FIRST: event_txt_tipo_habActionPerformed
// TODO agregue su código de manejo aquí:
* ÚLTIMO GEN: event_txt_tipo_habActionPerformed

vate  void  txt_buscar_habActionPerformed ( java.awt.event . ActionEvent  evt ) { // GEN-FIRST: event_txt_buscar_habActionPerforme
// TODO agregue su código de manejo aquí:
* GEN-ÚLTIMO: event_txt_buscar_habActionPerformed

vate  void  btn_listar_habActionPerformed ( java.awt.event . ActionEvent  evt ) { // GEN-FIRST: event_btn_listar_habActionPerforme
// TODO agregue su código de manejo aquí:
* GEN-ULTIMO: event_btn_listar_habActionPerformed

*
param argumenta los argumentos de la linea de comando
*
ic  static  void  main ( String  args []) {
/ * Establece el aspecto y la sensación de Nimbus * /
// <editor-fold defaultstate = "collapsed" desc = "Código de configuración de apariencia (opcional)">
/ * Si Nimbus (introducido en Java SE 6) no está disponible, quédese con la apariencia predeterminada.
 * Para obtener más información, consulte http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
 * /
prueba {
    para ( javax.swing . UIManager . LookAndFeelInfo info :  javax.swing . UIManager . getInstalledLookAndFeels ()) {
        if ( " Nimbus " . es igual a (info . getName ())) {
            javax.swing . UIManager . setLookAndFeel (info . getClassName ());
            romper ;
        }
```

Ilustración 232: Vista Habitaciones - 10

```
vate  void  txt_buscar_habActionPerformed ( java.awt.event . ActionEvent  evt ) { // GEN-FIRST: event_txt_buscar_habActionPerforme
// TODO agregue su código de manejo aquí:
* GEN-ÚLTIMO: event_txt_buscar_habActionPerformed

vate  void  btn_listar_habActionPerformed ( java.awt.event . ActionEvent  evt ) { // GEN-FIRST: event_btn_listar_habActionPerforme
// TODO agregue su código de manejo aquí:
* GEN-ULTIMO: event_btn_listar_habActionPerformed

*
*param argumenta los argumentos de la línea de comando
*
lic  static  void  main ( String  args []) {
/ * Establece el aspecto y la sensación de Nimbus * /
// <editor-fold defaultstate = "collapsed" desc = "Código de configuración de apariencia (opcional)">
/ * Si Nimbus (introducido en Java SE 6) no está disponible, quédese con la apariencia predeterminada.
 * Para obtener más información, consulte http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
 * /
prueba {
    para ( javax.swing . UIManager . LookAndFeelInfo info :  javax.swing . UIManager . getInstalledLookAndFeels ()) {
        if ( " Nimbus " . es igual a (info . getName ())) {
            javax.swing . UIManager . setLookAndFeel (info . getClassName ());
            romper ;
        }
    }
} catch ( ClassNotFoundException ex) {
    java.util.logging . Logger . getLogger ( VistaHabitaciones . class . getName ()) . log ( java.util.logging . Level . SEVERE ,
} catch ( InstantiationException ex) {
    java.util.logging . Logger . getLogger ( VistaHabitaciones . class . getName ()) . log ( java.util.logging . Level . SEVERE ,
} catch ( IllegalAccessException ex) {
    java.util.logging . Logger . getLogger ( VistaHabitaciones . class . getName ()) . log ( java.util.logging . Level . SEVERE ,
} catch ( javax.swing . UnsupportedLookAndFeelException ex) {
    java.util.logging . Logger . getLogger ( VistaHabitaciones . class . getName ()) . log ( java.util.logging . Level . SEVERE ,
}
// </editor-fold>

/ * Crea y muestra el formulario * /
java.awt . EventQueue . invokeLater ( new  Runnable () {
    public  void  run () {
        nueva  VistaHabitaciones () . setVisible ( verdadero );
    }
});


eclaración de variables - no modificar // GEN-BEGIN: variables
lic  javax.swing . JButton btn_actualizar_hab;
lic  javax.swing . JButton btn_buscar_hab;
lic  javax.swing . JButton btn_editar_hab;
lic  javax.swing . JButton btn_eliminar_hab;
lic  javax.swing . JButton btn_limpiar_campos_hab;
lic  javax.swing . JButton btn_listar_hab;
lic  javax.swing . JButton btn_nuevo_hab;
lic  javax.swing . JButton btn_registrar_hab;
vax.swing privado . JLabel jLabel1;
vax.swing privado . JLabel jLabel2;
vax.swing privado . JLabel jLabel3;
vax.swing privado . JLabel jLabel4;
vax.swing privado . JLabel jLabel5;
vax.swing privado . JLabel jLabel6;
vax.swing privado . JLabel jLabel7;
vax.swing privado . JLabel jLabel8;
vax.swing privado . JPanel jPanel1;
vax.swing privado . JPanel jPanel2;
vax.swing privado . JPanel jPanel3;
vax.swing privado . JScrollPane jScrollPane1;
lic  javax.swing . JTable tb_datos_hab;
lic  javax.swing . JTextField txt_buscar_hab;
lic  javax.swing . JTextField txt_estado_hab;
lic  javax.swing . JTextField txt_id_habitacion;
lic  javax.swing . JTextField txt_numero_hab;
lic  javax.swing . JTextField txt_piso_hab;
lic  javax.swing . JTextField txt_precio_hab;
lic  javax.swing . JTextField txt_tipo_hab;
in de la declaración de variables // GEN-END: variables
```

© 2020 GitHub, Inc. Condiciones Intimidad Seguridad Estado Ayuda Póngase en contacto con GitHub Precios API Formación Blog Acerca de

Ilustración 233: Vista Habitaciones - 11

Control de las Habitaciones

319 líneas (294 sloc) | 12,7 KB

Crudo Culpa

```
 controladores de paquetes ;

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 ;
importar  modelos . Habitacion ;
importar  modelos.HabitacionDAO ;
importar  modelos.Validaciones ;
importar  vistas.VistaHabitaciones ;

La  clase  pública HabitacionController  implementa  ActionListener {
    listar booleano ;
    actualizar booleano ;

    DefaultTableModel modeloTabla =  nuevo  DefaultTableModel ();
    VistaHabitaciones vista =  new  VistaHabitaciones ();
    HabitacionDAO dao =  nueva  HabitacionDAO ();
    Validaciones val = new  Validaciones ();
     var booleana ;

    public  HabitacionController ( VistaHabitaciones  vistaHabitaciones ) {
        esto . vista = vistaHabitaciones;
        esto . vista . btn_listar_hab . addActionListener ( esto );
        esto . vista . btn_nuevo_hab . addActionListener ( esto );
        esto . vista . btn_registrar_hab . addActionListener ( esto );
        esto . vista . btn_actualizar_hab . addActionListener ( esto );
        esto . vista . btn_buscar_hab . addActionListener ( esto );
        esto . vista . btn_eliminar_hab . addActionListener ( esto );
        esto . vista . btn_limpiar_campos_hab . addActionListener ( esto );
        esto . vista . btn_editar_hab . addActionListener ( esto );
    }

    @Anular
    public  void  actionPerformed ( ActionEvent  e ) {
        if (e . getSource () == vista . btn_listar_hab) {
            limpiarTabla ();
            listar (vista . tb_datos_hab);
            vista . btn_actualizar_hab . setEnabled ( falso );
        }
        if (p . getSource () == vista . btn_registrar_hab) {
            registrador();
            if (listar) {
                vista . btn_editar_hab . setEnabled ( verdadero );
                vista . btn_eliminar_hab . setEnabled ( verdadero );
                limpiarCajas ();
                limpiarTabla ();
                listar (vista . tb_datos_hab);
            }
        }
        if (e . getSource () == vista . btn_limpiar_campos_hab) {
            limpiarCajas ();
            vista . btn_actualizar_hab . setEnabled ( falso );
            vista . btn_editar_hab . setEnabled ( falso );
            vista . btn_eliminar_hab . setEnabled ( falso );
            limpiarTabla ();
        }
        if (e . getSource () == vista . btn_editar_hab) {
            editar ();
        }
        if (e . getSource () == vista . btn_actualizar_hab) {
            actualizar ();
            if (actualizar) {
                limpiarTabla ();
                limpiarCajas ();
                listar (vista . tb_datos_hab);
            }
        }
        if (e . getSource () == vista . btn_eliminar_hab) {
            eliminar ();
            limpiarTabla ();
            limpiarCajas ();
            listar (vista . tb_datos_hab);
        }

        if (p . getSource () == vista . btn_nuevo_hab) {
            limpiarCajas ();
            vista . btn_nuevo_hab . setEnabled ( falso );
            vista . btn_registrar_hab . setEnabled ( verdadero );
```

Ilustración 234: Control Habitaciones - 1

```
78          if (p . getSource () == vista . btn_nuevo_hab) {
79              limpiarCajas ();
80              vista . btn_nuevo_hab . setEnabled ( falso );
81              vista . btn_registrar_hab . setEnabled ( verdadero );
82              vista . btn_actualizar_hab . setEnabled ( falso );
83          }
84
85          if (e . getSource () == vista . btn_buscar_hab) {
86              limpiarTabla ();
87              if (validarBuscar () == true ) {
88                  listarBuscar (vista . tb_datos_hab, vista . txt_buscar_hab . getText ());
89                  vista . btn_actualizar_hab . setEnabled ( falso );
90              }
91          }
92      }
93
94       registrador vacío  privado () {
95          if (vista . txt_numero_hab . getText () . equals ( " " ) || vista . txt_piso_hab . getText () . equals ( " " ) ||
96              || vista . txt_tipo_hab . getText () . es igual a ( " " ) || vista . txt_estado_hab . getText () . es igual a
97              JOptionPane . showMessageDialog (vista, " Debe completar todos los campos " );
98              listar =  falso ;
99          }
100         else {
101             if (validarNumero () == verdadero  && validarPiso () == verdadero  && validarPrecio () == verdadero  && valid
102                 && validarEstado () == verdadero ) {
103                 listar =  verdadero ;
104                 String numero = vista . txt_numero_hab . getText ();
105                 Cadena piso = vista . txt_piso_hab . getText ();
106                 precio doble =  Doble . parseDouble (vista . txt_precio_hab . getText ());
107                 String tipo_habitacion = vista . txt_tipo_hab . getText ();
108                 String estado = vista . txt_estado_hab . getText ();
109
110                 Habitacion hab = nueva  Habitacion (numero, piso, precio, tipo_habitacion, estado);
111                 int listo =  esto . dao . agregar (hab);
112                 si (listo ==  1 ) {
113                     JOptionPane . showMessageDialog (vista, " Habitacion registrada con éxito " );
114                 } más {
115                     JOptionPane . showMessageDialog (vista, " Error al registrar habitacion " );
116                 }
117             }
118         }
119     }
120
121     private  void  listarBuscar ( JTable  tabla , String  buscar ) {
122         esto . vista . btn_editar_hab . setEnabled ( verdadero );
123         esto . vista . btn_eliminar_hab . setEnabled ( verdadero );
124         esto . modeloTabla = ( DefaultTableModel ) tabla . getModel ();
125         Lista < Habitacion > listarHabitacion =  this . dao . listarBuscar (buscar);
126         Objeto [] objeto =  nuevo  Objeto [ 6 ];
127         if (listarHabitacion . size () > 0 ) {
128             for ( int i = 0 ; i < listarHabitacion . size (); i ++ ) {
129                 objeto [ 0 ] = listarHabitacion . obtener (i) . getIdHabitacion ();
130                 objeto [ 1 ] = listarHabitacion . obtener (i) . getNumero ();
131                 objeto [ 2 ] = listarHabitacion . obtener (i) . getPiso ();
132                 objeto [ 3 ] = listarHabitacion . obtener (i) . getPrecio ();
133                 objeto [ 4 ] = listarHabitacion . obtener (i) . getTipo_Habitacion ();
134                 objeto [ 5 ] = listarHabitacion . obtener (i) . getEstado ();
135                 modeloTabla . addRow (objeto);
136             }
137             vista . tb_datos_hab . setModel (modeloTabla);
138         } más {
139             JOptionPane . showMessageDialog (vista, " No hay habitaciones para listar " );
140         }
141     }
142
143     privada  vacío  listar ( JTable  tabla ) {
144         esto . vista . btn_editar_hab . setEnabled ( verdadero );
145         esto . vista . btn_eliminar_hab . setEnabled ( verdadero );
146         esto . modeloTabla = ( DefaultTableModel ) tabla . getModel ();
147         Lista < Habitacion > listarHabitacion =  this . dao . listar ();
148         Objeto [] objeto =  nuevo  Objeto [ 6 ];
149         if (listarHabitacion . size () > 0 ) {
150             for ( int i = 0 ; i < listarHabitacion . size (); i ++ ) {
151                 objeto [ 0 ] = listarHabitacion . obtener (i) . getIdHabitacion ();
152                 objeto [ 1 ] = listarHabitacion . obtener (i) . getNumero ();
153                 objeto [ 2 ] = listarHabitacion . obtener (i) . getPiso ();
154                 objeto [ 3 ] = listarHabitacion . obtener (i) . getPrecio ();
155                 objeto [ 4 ] = listarHabitacion . obtener (i) . getTipo_Habitacion ();
156                 objeto [ 5 ] = listarHabitacion . obtener (i) . getEstado ();
157                 modeloTabla . addRow (objeto);
158             }
159             vista . tb_datos_hab . setModel (modeloTabla);
160         } más {
161             JOptionPane . showMessageDialog (vista, " No hay habitaciones para listar " );
```

Ilustración 235: Control Habitaciones - 2

```
        } más {
            JOptionPane . showMessageDialog (vista, " No hay habitaciones para listar " );
        }
    }

    private  void  eliminar () {
        int fila = vista . tb_datos_hab . getSelectedRow ();
        si (fila ==  - 1 ) {
            JOptionPane . showMessageDialog (vista, " Debe seleccionar una fila " );
        } más {
            int id =  Entero . parseInt (( String ) vista . tb_datos_hab . getValueAt (fila, 0 ) . toString ());
            int iden =  esto . dao . eliminar (id);
            si (iden ==  1 ) {
                JOptionPane . showMessageDialog (vista, " Habitacion eliminada con éxito " );
                vista . btn_registrar_hab . setEnabled ( verdadero );
                vista . btn_nuevo_hab . setEnabled ( falso );
                vista . btn_actualizar_hab . setEnabled ( falso );
            } más {
                JOptionPane . showMessageDialog (vista, " Error al eliminar habitacion " );
            }
        }
    }

     vacío  privado editar () {
        int fila = vista . tb_datos_hab . getSelectedRow ();
            si (fila ==  - 1 ) {
                JOptionPane . showMessageDialog (vista, " Debe seleccionar una fila " );
            } más {
                esto . vista . btn_registrar_hab . setEnabled ( falso );
                esto . vista . btn_actualizar_hab . setEnabled ( verdadero );
                esto . vista . btn_nuevo_hab . setEnabled ( verdadero );
                int codigo =  Entero . parseInt (( String ) vista . tb_datos_hab . getValueAt (fila, 0 ) . toString ());
                String numero = ( String ) vista . tb_datos_hab . getValueAt (fila, 1 );
                String piso = ( String ) vista . tb_datos_hab . getValueAt (fila, 2 );
                Precio doble =  Doble . parseDouble (( String ) vista . tb_datos_hab . getValueAt (fila, 3 ) . toString (
                String tipo_habitacion = ( String ) vista . tb_datos_hab . getValueAt (fila, 4 );
                String estado = ( String ) vista . tb_datos_hab . getValueAt (fila, 5 );

                vista . txt_id_habitacion . setText (codigo + " " );
                vista . txt_numero_hab . setText (número);
                vista . txt_piso_hab . setText (piso);
                vista . txt_precio_hab . setText (precio + " " );
                vista . txt_tipo_hab . setText (tipo_habitacion);
                vista . txt_estado_hab . setText (estado);
            }
    }

     vacío  privado actualizar () {
         if (vista . txt_numero_hab . getText () . equals ( " " ) || vista . txt_piso_hab . getText () . equals ( " " ) |
            || vista . txt_tipo_hab . getText () . es igual a ( " " ) || vista . txt_estado_hab . getText () . es igual a
            JOptionPane . showMessageDialog (vista, " Debe completar todos los campos " );
            listar =  falso ;
        } más {
            if (validarNumero () == verdadero  && validarPiso () == verdadero  && validarPrecio () == verdadero  && valid
                && validarEstado () == verdadero ) {
                actualizar =  verdadero ;
                vista . btn_registrar_hab . setEnabled ( verdadero );
                vista . btn_actualizar_hab . setEnabled ( falso );
                vista . btn_nuevo_hab . setEnabled ( falso );

                int codigo =  Entero . parseInt (vista . txt_id_habitacion . getText ());
                String numero = vista . txt_numero_hab . getText ();
                Cadena piso = vista . txt_piso_hab . getText ();
                precio doble =  Doble . parseDouble (vista . txt_precio_hab . getText ());
                String tipo_habitacion = vista . txt_tipo_hab . getText ();
                String estado = vista . txt_estado_hab . getText ();

                Habitacion hab = nueva  Habitacion (codigo, numero, piso, precio, estado, tipo_habitacion);
                int r =  esto . dao . actualizar (hab);
                si (r ==  1 ) {
                    JOptionPane . showMessageDialog (vista, " Habitacion actualizada con éxito " );
                } más {
                    JOptionPane . showMessageDialog (vista, " Error al actualizar habitacion " );
                }
            }
        }
    }

     vacío  privado limpiarTabla () {
        para ( int i = 0 ; i < vista . tb_datos_hab . getRowCount (); i ++ ) {
            modeloTabla . removeRow (i);
            i = i - 1 ;
        }
    }
```

Ilustración 236: Control Habitaciones - 3

```
        }
    }

     vacio  privado limpiarCajas () {
        vista . txt_id_habitacion . setText ( " " );
        vista . txt_numero_hab . setText ( " " );
        vista . txt_piso_hab . setText ( " " );
        vista . txt_precio_hab . setText ( " " );
        vista . txt_tipo_hab . setText ( " " );
        vista . txt_estado_hab . setText ( " " );
    }

    privada  booleano  validarNumero () {
        var  = val . validacionHabitaciones (vista . txt_numero_hab . getText ());
        si ( var  ==  falso ) {
            JOptionPane . showMessageDialog (vista, " El numero de habitaciones es de 3 digitos " );
            vista . txt_numero_hab . setText ( " " );
            devolver  falso ;
        } más {
            devuelve  verdadero ;
        }
    }

    privada  booleano  validarPiso () {
       var  = val . validacionPiso (vista . txt_piso_hab . getText ());
        si ( var  ==  falso ) {
            JOptionPane . showMessageDialog (vista, " El número de piso es de un solo digito " );
            vista . txt_piso_hab . setText ( " " );
            devolver  falso ;
        } más {
            devuelve  verdadero ;
        }
    }

    privada  booleano  validarPrecio () {
        var  = val . validacionNumerosDecimales (vista . txt_precio_hab . getText ());
        si ( var  ==  falso ) {
            JOptionPane . showMessageDialog (vista, " Ingrese un precio valido " );
            vista . txt_precio_hab . setText ( " " );
            devolver  falso ;
        } más {
            devuelve  verdadero ;
        }
    }

    privada  booleano  validarTipo () {
       var  = val . validacionLetras (vista . txt_tipo_hab . getText ());
        si ( var  ==  falso ) {
            JOptionPane . showMessageDialog (vista, " Solo puede ingresar letras " );
            vista . txt_tipo_hab . setText ( " " );
            devolver  falso ;
        } más {
            devuelve  verdadero ;
        }
    }

    privada  booleano  validarEstado () {
       var  = val . validacionLetras (vista . txt_estado_hab . getText ());
        si ( var  ==  falso ) {
            JOptionPane . showMessageDialog (vista, " Solo puede ingresar letras " );
            vista . txt_estado_hab . setText ( " " );
            devolver  falso ;
        } más {
            devuelve  verdadero ;
        }
    }

    private  boolean  validarBuscar () {
        var  = val . validacionHabitaciones (vista . txt_buscar_hab . getText ());
        si ( var  ==  falso ) {
            JOptionPane . showMessageDialog (vista, " Ingrese un numero de habitacion valido " );
            vista . txt_buscar_hab . setText ( " " );
            devolver  falso ;
        } más {
            devuelve  verdadero ;
        }
```

Ilustración 237: Control Habitaciones - 4

```
1   controladores de paquetes ;
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   importar  modelos . Habitacion ;
10  importar  modelos.HabitacionDAO ;
11  importar  modelos.Validaciones ;
12  importar  vistas.VistaListaHabitacion ;
13  importar  vistas.VistaReserva ;
14
15  public  class  HabitacionListaController  implementa  ActionListener {
16
17      DefaultTableModel modeloTabla =  nuevo  DefaultTableModel ();
18      VistaListaHabitacion vista =  new  VistaListaHabitacion ();
19      HabitacionDAO dao =  nueva  HabitacionDAO ();
20      VistaReserva v =  nuevo  VistaReserva ();
21      Validaciones val =  new  Validaciones ();
22       var booleana ;
23
24      public  HabitacionListaController ( VistaListaHabitacion  vistalistahabitacion ) {
25          esto . vista = vistalistahabitacion;
26          esto . vista . btn_buscar_habitacion . addActionListener ( esto );
27          esto . vista . btn_salir_habitacion . addActionListener ( esto );
28      }
29
30      @Anular
31      public  void  actionPerformed ( ActionEvent  e ) {
32          if (e . getSource () == vista . btn_buscar_habitacion) {
33              limpiarTabla ();
34              if (validarBuscar () == true ) {
35                  listar (vista . tb_vista_habitacion, vista . txt_buscar_habitacion . getText ());
36              }
37          }
38          if (e . getSource () == vista . btn_salir_habitacion) {
39              esto . vista . disponer();
40          }
41      }
42
43      privada  vacío  listar ( JTable  tabla , Cadena  buscar ) {
44          esto . modeloTabla = ( DefaultTableModel ) tabla . getModel ();
45          Lista < Habitacion > listarHabitacion =  this . dao . listarBuscar (buscar);
46          Objeto [] objeto =  nuevo  Objeto [ 6 ];
47          if (listarHabitacion . size () > 0 ) {
48              for ( int i = 0 ; i < listarHabitacion . size (); i ++ ) {
49                  objeto [ 0 ] = listarHabitacion . obtener (i) . getIdHabitacion ();
50                  objeto [ 1 ] = listarHabitacion . obtener (i) . getNumero ();
51                  objeto [ 2 ] = listarHabitacion . obtener (i) . getPiso ();
52                  objeto [ 3 ] = listarHabitacion . obtener (i) . getPrecio ();
53                  objeto [ 4 ] = listarHabitacion . obtener (i) . getTipo_Habitacion ();
54                  objeto [ 5 ] = listarHabitacion . obtener (i) . getEstado ();
55                  modeloTabla . addRow (objeto);
56              }
57              vista . tb_vista_habitacion . setModel (modeloTabla);
58          } más {
59              JOptionPane . showMessageDialog (vista, " No hay habitaciones para listar " );
60          }
```

Ilustración 238: Control Habitaciones - 5

```
18      VistaListaHabitacion vista =  new  VistaListaHabitacion ();
19      HabitacionDAO dao =  nueva  HabitacionDAO ();
20      VistaReserva v =  nuevo  VistaReserva ();
21      Validaciones val =  new  Validaciones ();
22       var booleana ;
23
24      public  HabitacionListaController ( VistaListaHabitacion  vistalistahabitacion ) {
25          esto . vista = vistalistahabitacion;
26          esto . vista . btn_buscar_habitacion . addActionListener ( esto );
27          esto . vista . btn_salir_habitacion . addActionListener ( esto );
28      }
29
30      @Anular
31      public  void  actionPerformed ( ActionEvent  e ) {
32          if (e . getSource () == vista . btn_buscar_habitacion) {
33              limpiarTabla ();
34              if (validarBuscar () == true ) {
35                  listar (vista . tb_vista_habitacion, vista . txt_buscar_habitacion . getText ());
36              }
37          }
38          if (e . getSource () == vista . btn_salir_habitacion) {
39              esto . vista . disponer();
40          }
41      }
42
43      privada  vacío  listar ( JTable  tabla , Cadena  buscar ) {
44          esto . modeloTabla = ( DefaultTableModel ) tabla . getModel ();
45          Lista < Habitacion > listarHabitacion =  this . dao . listarBuscar (buscar);
46          Objeto [] objeto =  nuevo  Objeto [ 6 ];
47          if (listarHabitacion . size () > 0 ) {
48              for ( int i = 0 ; i < listarHabitacion . size (); i ++ ) {
49                  objeto [ 0 ] = listarHabitacion . obtener (i) . getIdHabitacion ();
50                  objeto [ 1 ] = listarHabitacion . obtener (i) . getNumero ();
51                  objeto [ 2 ] = listarHabitacion . obtener (i) . getPiso ();
52                  objeto [ 3 ] = listarHabitacion . obtener (i) . getPrecio ();
53                  objeto [ 4 ] = listarHabitacion . obtener (i) . getTipo_Habitacion ();
54                  objeto [ 5 ] = listarHabitacion . obtener (i) . getEstado ();
55                  modeloTabla . addRow (objeto);
56              }
57              vista . tb_vista_habitacion . setModel (modeloTabla);
58          } más {
59              JOptionPane . showMessageDialog (vista, " No hay habitaciones para listar " );
60          }
61      }
62
63       vacío  privado limpiarTabla () {
64          para ( int i = 0 ; i < vista . tb_vista_habitacion . getRowCount (); i ++ ) {
65              modeloTabla . removeRow (i);
66              i = i - 1 ;
67          }
68      }
69
70      private  boolean  validarBuscar () {
71          var  = val . validacionHabitaciones (vista . txt_buscar_habitacion . getText ());
72          si ( var  ==  falso ) {
73              JOptionPane . showMessageDialog (vista, " Ingrese un numero de habitacion valido " );
74              vista . txt_buscar_habitacion . setText ( " " );
75              devolver  falso ;
76          } más {
77              devuelve  verdadero ;
78          }
79      }
80  }
```

© 2020 GitHub, Inc. Condiciones Intimidad Seguridad Estado Ayuda Póngase en contacto con GitHub Precios API Formación Acerca de

Ilustración 239: Control Habitaciones - 6

Modelo de las Habitaciones

80 líneas (62 sloc) | 1,76 KB

Crudo Culpa

```
1   modelos de paquete ;
2
3  público  de clase  Habitacion {
4      private  int idHabitacion;
5      Private  String numero;
6       piso privado String ;
7       precio doble privado ;
8      private  String estado;
9      private  String tipo_Habitacion;
10
11     public  Habitacion ( int  idHabitacion , String  numero , String  piso , Double  precio , String  estado , String  ti
12         esto . idHabitacion = idHabitacion;
13         esto . numero = numero;
14         esto . piso = piso;
15         esto . precio = precio;
16         esto . estado = estado;
17         esto . tipo_Habitacion = tipo_Habitacion;
18     }
19
20     public  Habitacion ( String  numero , String  piso , Double  precio , String  estado , String  tipo_Habitacion ) {
21         esto . numero = numero;
22         esto . piso = piso;
23         esto . precio = precio;
24         esto . estado = estado;
25         esto . tipo_Habitacion = tipo_Habitacion;
26     }
27
28     pública  Habitacion () {
29     }
30
31     public  int  getIdHabitacion () {
32         return idHabitacion;
33     }
34
35     public  void  setIdHabitacion ( int  idHabitacion ) {
36         esto . idHabitacion = idHabitacion;
37     }
38
39     public  String  getNumero () {
40         return numero;
41     }
42
43     public  void  setNumero ( String  numero ) {
44         esto . numero = numero;
45     }
46
47     public  String  getPiso () {
48         volver piso;
49     }
50
51     public  void  setPiso ( String  piso ) {
52         esto . piso = piso;
53     }
54
55     public  Double  getPrecio () {
56         devolver precio;
57     }
58
59     pública  vacío  setPrecio ( doble  precio ) {
60         esto . precio = precio;
61     }
62
63     public  String  getEstado () {
64         return estado;
65     }
66
67     public  void  setEstado ( String  estado ) {
68         esto . estado = estado;
69     }
70
71     public  String  getTipo_Habitacion () {
72         return tipo_Habitacion;
73     }
74
75     public  void  setTipo_Habitacion ( String  tipo_Habitacion ) {
76         esto . tipo_Habitacion = tipo_Habitacion;
77     }
78
79
80 }
```

Ilustración 240: Modelo Habitaciones - 1

```
modelos de paquete ;

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

 clase  pública HabitacionDAO {
    Conexión conectar =  nueva  Conexión ();
    Conexión con;
    PreparedStatement ps;
    ResultSet rs;

    pública  Lista < Habitacion >  listar () {
        List < Habitacion > listarHabitacion =  new  ArrayList <> ();
        String sql =  " SELECT * FROM habitacion " ;
        prueba {
            con = conectar . conectarServidor ();
            ps = con . prepareStatement (sql);
            rs = ps . ejecutar la solicitud();
            while (rs . next ()) {
                Habitacion hab =  nueva  Habitacion ();
                hab . setIdHabitacion (rs . getInt ( 1 ));
                hab . setNumero (rs . getString ( 2 ));
                hab . setPiso (rs . getString ( 3 ));
                hab . setPrecio (rs . getDouble ( 4 ));
                hab . setTipo_Habitacion (rs . getString ( 5 ));
                hab . setEstado (rs . getString ( 6 ));
                listarHabitacion . añadir (hab);
            }
        } captura ( Excepción e) {
            Sistema . err . println ( " Error al listar: " + e . getMessage ());
        }
        return listarHabitacion;
    }

    public  List < Habitacion >  listarBuscar ( String  buscar ) {
        List < Habitacion > listarHabitacion =  new  ArrayList <> ();
        String sql =  " SELECT * FROM habitacion donde numero like '% " + buscar +  " %' " ;
        prueba {
            con = conectar . conectarServidor ();
            ps = con . prepareStatement (sql);
            rs = ps . ejecutar la solicitud();
            while (rs . next ()) {
                Habitacion hab =  nueva  Habitacion ();
                hab . setIdHabitacion (rs . getInt ( 1 ));
                hab . setNumero (rs . getString ( 2 ));
                hab . setPiso (rs . getString ( 3 ));
                hab . setPrecio (rs . getDouble ( 4 ));
                hab . setTipo_Habitacion (rs . getString ( 5 ));
                hab . setEstado (rs . getString ( 6 ));
                listarHabitacion . añadir (hab);
            }
        } captura ( Excepción e) {
            Sistema . err . println ( " Error al listar: " + e . getMessage ());
        }
        return listarHabitacion;
    }

    public  int  agregar ( Habitacion  hab ) {
        String sql =  " INSERT INTO habitacion VALUES (null,?,?,?,?,?) " ;
        prueba {
            con = conectar . conectarServidor ();
            ps = con . prepareStatement (sql);
            ps . setString ( 1 , hab . getNumero ());
            ps . setString ( 2 , hab . getPiso ());
            ps . setDouble ( 3 , hab . getPrecio ());
            ps . setString ( 4 , hab . getEstado ());
            ps . setString ( 5 , hab . getTipo_Habitacion ());
            ps . executeUpdate ();
```

Ilustración 241: Modelo Habitaciones - 2

```
    prueba {
        con = conectar . conectarServidor ();
        ps = con . prepareStatement (sql);
        rs = ps . ejecutar la solicitud();
        while (rs . next ()) {
            Habitacion hab =  nueva  Habitacion ();
            hab . setIdHabitacion (rs . getInt ( 1 ));
            hab . setNumero (rs . getString ( 2 ));
            hab . setPiso (rs . getString ( 3 ));
            hab . setPrecio (rs . getDouble ( 4 ));
            hab . setTipo_Habitacion (rs . getString ( 5 ));
            hab . setEstado (rs . getString ( 6 ));
            listarHabitacion . añadir (hab);
        }
    } captura ( Excepción e) {
        Sistema . err . println ( " Error al listar: " + e . getMessage ());
    }
    return listarHabitacion;
}

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

public  int  actualizar ( Habitacion  hab ) {
    int r =  0 ;
    String sql =  " ACTUALIZAR habitacion SET numero =?, Piso =?, Precio =?, Estado_habitacion =?, Tipo_habitacion =? WHERE idHa
    prueba {
        con = conectar . conectarServidor ();
        ps = con . prepareStatement (sql);
        ps . setString ( 1 , hab . getNumero ());
        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 ();
        si (r ==  1 ) {
            return  1 ;
        } más {
            return  0 ;
        }
    } captura ( Excepción e) {
        Sistema . err . println ( " Error al actualizar: " + e . getMessage ());
    }
    return r;
}

public  int  eliminar ( int  idHabitacion ) {
    String sql =  " BORRAR DE habitacion DONDE idHabitacion = " + idHabitacion;
    prueba {
        con = conectar . conectarServidor ();
        ps = con . prepareStatement (sql);
        ps . executeUpdate ();
        return  1 ;
    } captura ( Excepción e) {
        Sistema . err . println ( " Error al eliminar: " + e . getMessage ());
        return  0 ;
    }
}
```

© 2020 GitHub, Inc. Condiciones Intimidad Seguridad Estado Ayuda Póngase en contacto con GitHub Precios API Formación Blog
Acerca de

Ilustración 242: Modelo Habitaciones - 3

Diagrama de secuencia del caso de uso Registrar reserva

Ilustración 243: Diagrama de secuencia - Registrar reserva

Vista de la Reserva

Ilustración 244: Vista Reserva - 1

Ilustración 245: Vista Reserva - 2

Ilustración 246: Vista Reserva - 3

Ilustración 247: Vista Reserva - 4

Ilustración 248: Vista Reserva - 5

Ilustración 249: Vista Reserva - 6

Ilustración 250: Vista Reserva - 7

Ilustración 251: Vista Reserva - 8

Ilustración 252: Vista Reserva - 9

Ilustración 253: Vista Reserva - 10

Ilustración 254: Vista Reserva - 11

Ilustración 255: Vista Reserva - 12

Ilustración 256: Vista Reserva - 13

Ilustración 257: Vista Reserva - 14

Ilustración 258: Vista Reserva - 15

Control de la Reserva

Ilustración 259: Control Reserva - 1

Ilustración 260: Control Reserva - 2

Ilustración 261: Control Reserva - 3

Ilustración 262: Control Reserva - 4

Ilustración 263: Control Reserva - 5

Ilustración 264: Control Reserva - 6

Ilustración 265: Control Reserva - 7

Ilustración 266: Control Reserva - 8

Ilustración 267: Control Reserva - 9

Ilustración 268: Control Reserva - 10

Ilustración 269: Control Reserva - 11

Ilustración 270: Control Reserva - 12

Ilustración 271: Control Reserva - 13

Ilustración 272: Control Reserva - 14

Ilustración 273: Control Reserva - 15

Modelo de la Reserva

Ilustración 274: Modelo Reserva - 1

Ilustración 275: Modelo Reserva - 2

Ilustración 276: Modelo Reserva - 3

Ilustración 277: Modelo Reserva - 4

Ilustración 278: Modelo Reserva - 5

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 ingenieria 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 279: Videoconferencia con representante de la empresa

Ilustración 280: Integrante - Diego Tasaico

Ilustración 281: Integrante - Mauricio Villanueva

Ilustración 282: Integrante - Olenka Lazo

Ilustración 283: Integrante - Abihail Zapata

Starlight Inn

Pisco, 13 octubre del 2020

Srs. Representantes

Universidad Privada del Norte

Con la finalidad de completar y modernizar nuestro sistema administrativo, se autoriza para que los jóvenes estudiantes (Diego Tasaico, Mauricio Villanueva, Abihail Zapata y Olenka Lazo), realicen un programa virtual que nos facilite nuestro registro diario que actualmente usamos. De ser necesario, se aprueba la visita a las instalaciones del HOTEL, así como cualquier otra información que ayude en la elaboración de este importante proyecto.

Atte.

Eduardo Enrique Albarracin Trillo

Administrador

Ilustración 284: Acta de compromiso

Funcionamiento del Software:

Ilustración 285: Funcionamiento del Software - Acceso al sistema

Ilustración 286: Funcionamiento del Software - Menú Principal

Ilustración 287: Funcionamiento del Software - Vista de los clientes

Ilustración 288: Funcionamiento del Software - Vista de los trabajadores

Ilustración 289: Funcionamiento del Software - Vista de los movimientos

Ilustración 290: Funcionamiento del Software - Vista de las reservas

Ilustración 291: Funcionamiento del Software - Vista de las habitaciones

Ilustración 292: Funcionamiento del Software - Detección de errores

Ilustración 293: Funcionamiento del Software - Conexión a Base de Datos

Ilustración 294: Funcionamiento del Software - Tiempo de respuesta

Ilustración 295: Funcionamiento del Software - Espacio mínimo de memoria