

UNIVERSIDAD CATÓLICA DE SANTA MARÍA
FACULTAD DE CIENCIAS E INGENIERÍAS FÍSICAS Y FORMALES
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**Reconocimiento de Patrones de Movimiento Basado en
Raspberry Pi y Cámaras Megapíxel Para Mejorar la
Atención de Pacientes Hospitalizados**

**Tesis presentada por el bachiller:
Jesús Antonio Alpaca Rendón**
Para optar por el Título profesional de Ingeniero de
Sistemas: Especialidad en Ingeniería de Software

Asesor(a): Ing. Karina Rosas Paredes

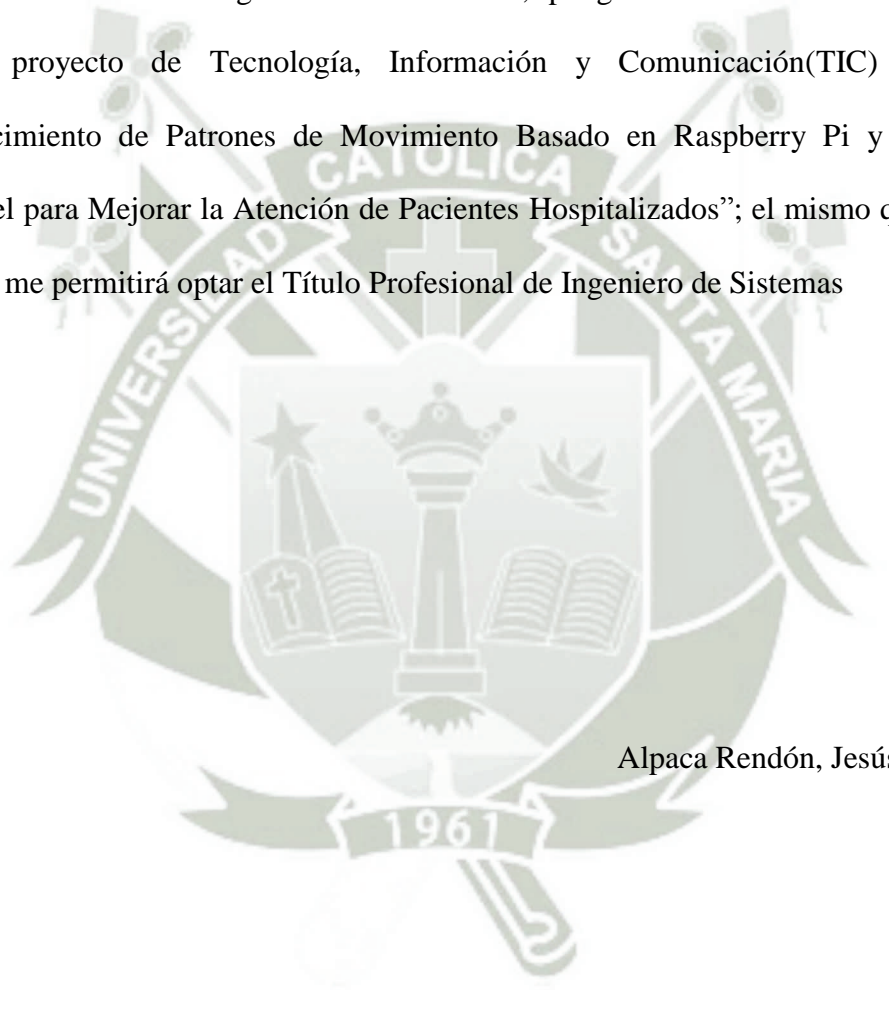
Arequipa, Perú 2017

PRESENTACIÓN

Sr. Director de la Escuela Profesional de Ingeniería de Sistemas

Sres. miembros del Jurado

De conformidad con las disposiciones del Reglamento de Grados y Títulos de la Escuela Profesional de Ingeniería de Sistemas, pongo a vuestra consideración el presente proyecto de Tecnología, Información y Comunicación(TIC) titulado: “Reconocimiento de Patrones de Movimiento Basado en Raspberry Pi y Cámaras Megapíxel para Mejorar la Atención de Pacientes Hospitalizados”; el mismo que de ser aprobado me permitirá optar el Título Profesional de Ingeniero de Sistemas



Alpaca Rendón, Jesús Antonio

AGRADECIMIENTOS

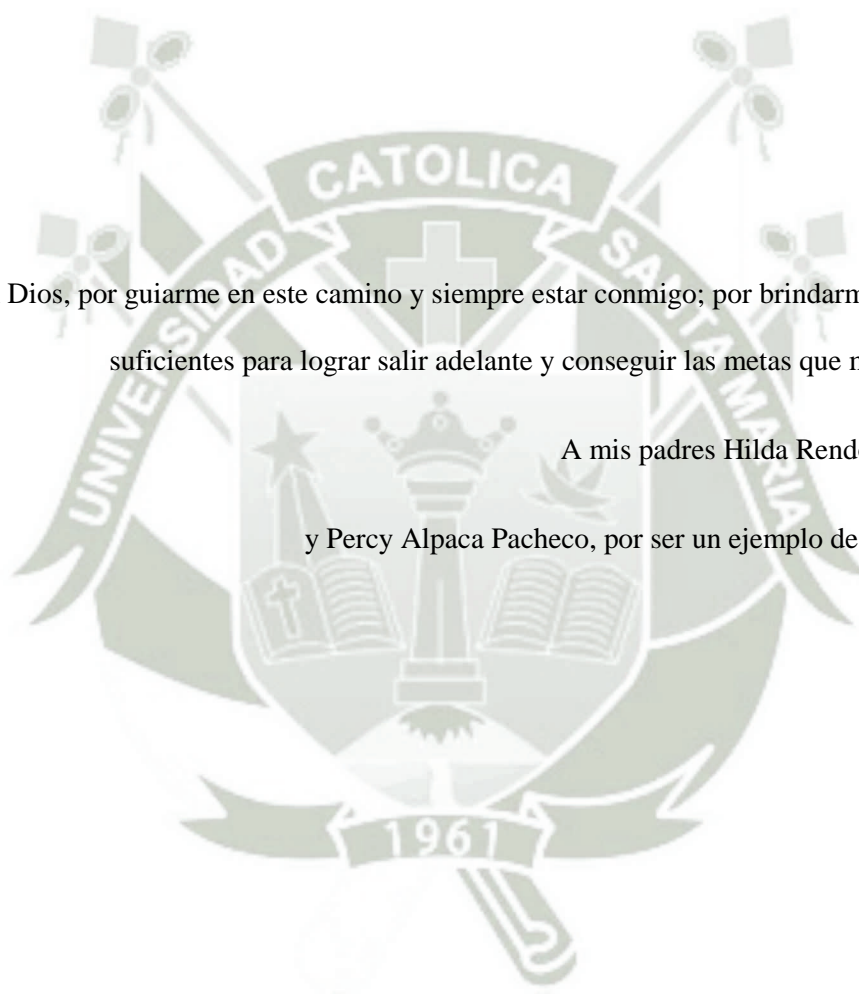
A mis padres, por apoyarme en todo momento, tenerme la paciencia suficiente y la enorme comprensión en mis malos momentos; siempre estaré agradecido con ellos por estar a mi lado, corregir mis errores y felicitarme por mis buenos momentos.

Al Ing. José Esquicha y la Ing. Karina Rosas, por ser de gran ayuda en el desarrollo de este proyecto y apoyo en mis años de estudiante en la Universidad

DEDICATORIA

A Dios, por guiarme en este camino y siempre estar conmigo; por brindarme las fuerzas suficientes para lograr salir adelante y conseguir las metas que me propongo.

A mis padres Hilda Rendón de Alpaca
y Percy Alpaca Pacheco, por ser un ejemplo de vida para mí



ÍNDICE

RESUMEN	1
ABSTRACT	2
INTRODUCCION	3
CAPITULO 1: PLANTEAMIENTO TEORICO	5
1.1. Descripción del Problema	5
1.2. Objetivos	5
1.3. Alcances y limitaciones.....	6
1.4. Justificación.....	7
1.5. Aspectos relevantes del Proyecto	7
CAPITULO 2: MARCO TEORICO	8
2.1. Estado del Arte	8
2.2. Bases Teóricas.....	12
2.2.1. Reconocimiento de patrones	12
2.2.2. Frameworks de reconocimiento de patrones	18
2.2.3. Hardware Libre	22
2.2.4. Ejemplos de Hardware Libre.....	25
2.2.5. Django	39
CAPITULO 3: DESCRIPCIÓN DEL SISTEMA Y DESARROLLO	41
3.1. Recolección y análisis de la información	41
3.1.1. Definición.....	41
3.1.2. Método de recolección de la información	41
3.2. Requerimientos del Sistema	58
3.2.1. Requerimientos del Software	58
3.2.2. Requerimientos del Hardware	77
3.2.3. Especificación de requisitos del software	81
3.3. Modelo Propuesto	98
3.3.1. Arquitectura del Software	98
3.3.2. Especificación del diseño	99
CAPITULO 4: VALIDACIÓN Y RESULTADOS	153
4.1.1. Descripción de pruebas	153
4.1.2. Pruebas	155
4.1.3. Resultados	157
Conclusiones	190
Recomendaciones.....	192

Referencias bibliográficas	194
Apéndices	197



ÍNDICE DE FIGURAS

<i>Figura 1. Proceso Básico de Inteligencia Artificial.....</i>	<i>13</i>
<i>Figura 2. Relación de la Visión Computacional con otros avances.</i>	<i>15</i>
<i>Figura 3. Ciclo de Vida de un sistema de Reconocimiento de Patrones</i>	<i>16</i>
<i>Figura 4. Ejemplo de Aforge.Net</i>	<i>21</i>
<i>Figura 5. Ciclo del diseño de hardware reconfigurable.....</i>	<i>24</i>
<i>Figura 6. Imagen de varias placas de Hardware Libre.....</i>	<i>25</i>
<i>Figura 7. Foto de un UzeBox con el respectivo mando de juegos conectado.....</i>	<i>27</i>
<i>Figura 8. CubieBoard 5 Octa-Core</i>	<i>28</i>
<i>Figura 9. Expander Board y sus partes</i>	<i>31</i>
<i>Figura 10. Arduino Uno.....</i>	<i>34</i>
<i>Figura 11. ESP8266 Camera para Arduino</i>	<i>35</i>
<i>Figura 12. Tarjeta Raspberry Pi Beta</i>	<i>36</i>
<i>Figura 13. Módulo de Cámara Megapíxel.....</i>	<i>37</i>
<i>Figura 14. Módulo de Cámara Megapíxel.....</i>	<i>38</i>
<i>Figura 15. Gráfico de Respuestas de Médicos, Pregunta 1.....</i>	<i>43</i>
<i>Figura 16. Gráfico de Respuestas de Enfermeras, Pregunta 1.....</i>	<i>44</i>
<i>Figura 17. Gráfico de Respuestas de Médicos, Pregunta 2.....</i>	<i>45</i>
<i>Figura 18. Gráfico de Respuestas de Enfermeras, Pregunta 2.....</i>	<i>45</i>
<i>Figura 19. Gráfico de Respuestas de Médicos, Pregunta 3.....</i>	<i>46</i>
<i>Figura 20. Gráfico de Respuestas de Enfermeras, Pregunta 3.....</i>	<i>47</i>
<i>Figura 21. Gráfico de Respuestas de Médicos, Pregunta 4.....</i>	<i>48</i>
<i>Figura 22. Gráfico de Respuestas de Enfermeras, Pregunta 4.....</i>	<i>48</i>
<i>Figura 23. Gráfico de Respuestas de Médicos, Pregunta 5.....</i>	<i>49</i>
<i>Figura 24. Gráfico de Respuestas de Enfermeras, Pregunta 5.....</i>	<i>50</i>
<i>Figura 25. Gráfico de Respuestas de Médicos, Pregunta 6.....</i>	<i>51</i>
<i>Figura 26. Gráfico de Respuestas de Enfermeras, Pregunta 6.....</i>	<i>51</i>
<i>Figura 27. Gráfico de Respuestas de Médicos, Pregunta 7.....</i>	<i>52</i>
<i>Figura 28. Gráfico de Respuestas de Enfermeras, Pregunta 7.....</i>	<i>52</i>
<i>Figura 29. Gráfico de Respuestas de Médicos, Pregunta 8.....</i>	<i>53</i>
<i>Figura 30. Gráfico de Respuestas de Enfermeras, Pregunta 8.....</i>	<i>54</i>
<i>Figura 31. Gráfico de Respuestas de Médicos, Pregunta 9.....</i>	<i>55</i>
<i>Figura 32. Gráfico de Respuestas de Enfermeras, Pregunta 9.....</i>	<i>55</i>
<i>Figura 33. Gráfico de Respuestas de Médicos, Pregunta 10.....</i>	<i>56</i>
<i>Figura 34. Gráfico de Respuestas de Enfermeras, Pregunta 10.....</i>	<i>57</i>
<i>Figura 35. Prueba de detección de colores</i>	<i>63</i>
<i>Figura 36. Estilos de desenfoque</i>	<i>64</i>
<i>Figura 37. Resultado de uno de los filtros de Nitidez.....</i>	<i>65</i>
<i>Figura 38. Imagen resultado del análisis de contornos y defectos.....</i>	<i>70</i>
<i>Figura 39. Jerarquía en Contornos de una Imagen.....</i>	<i>71</i>
<i>Figura 40. Gráfico de Distribución Normal</i>	<i>72</i>
<i>Figura 41. Plano Tridimensional de la Distribución Normal.....</i>	<i>72</i>
<i>Figura 42. Kernel de datos</i>	<i>73</i>
<i>Figura 43. Kernel de resultados</i>	<i>74</i>
<i>Figura 44. Selección de Contornos con el Algoritmo de Suzuki.....</i>	<i>75</i>
<i>Figura 45. Selección de Contornos con el Algoritmo de Suzuki.....</i>	<i>76</i>
<i>Figura 46. Raspberry Pi 3 B Model.....</i>	<i>79</i>
<i>Figura 47. Diagrama de Entidad-Relación</i>	<i>100</i>
<i>Figura 48. Diagrama de Entidad-Relación</i>	<i>101</i>

<i>Figura 49. Diagrama de Despliegue.....</i>	<i>103</i>
<i>Figura 50. Diagrama de Actividades del sistema en general</i>	<i>105</i>
<i>Figura 51. Diagrama de Actividades de manejo de alertas.....</i>	<i>106</i>
<i>Figura 52. Diagrama de Actividades generación de reportes</i>	<i>107</i>
<i>Figura 53. Diagrama de Actividades manejo de usuarios en el sistema</i>	<i>108</i>
<i>Figura 54. Ventana de Login</i>	<i>109</i>
<i>Figura 55. Ventana de Menú Principal</i>	<i>109</i>
<i>Figura 56. Ventana de Alertas</i>	<i>110</i>
<i>Figura 57. Ventana de Editar/Registrar Alertas.....</i>	<i>110</i>
<i>Figura 58. Ventana de Lista de Atenciones</i>	<i>111</i>
<i>Figura 59. Ventana de Edición de Alerta</i>	<i>111</i>
<i>Figura 60. Ventana de Menú Principal</i>	<i>112</i>
<i>Figura 61. Ventana de Alertas Administrador.....</i>	<i>112</i>
<i>Figura 62. Ventana de Lista de Atenciones (Administrador)</i>	<i>113</i>
<i>Figura 63. Ventana de Ver Alerta.....</i>	<i>113</i>
<i>Figura 64. Ventana de Administrador</i>	<i>114</i>
<i>Figura 65. Ventana de Generar Reportes.....</i>	<i>114</i>
<i>Figura 66. Ventana de Crear Usuario.....</i>	<i>115</i>
<i>Figura 67. Ventana de Búsqueda Avanzada</i>	<i>115</i>
<i>Figura 68. Ventana de Restablecer Contraseña</i>	<i>116</i>
<i>Figura 69. Ventana de Información de Usuario.....</i>	<i>116</i>
<i>Figura 70. Diagrama de Clases Del módulo del Raspberry Pi</i>	<i>118</i>
<i>Figura 71. Raspberry Pi 3 Modelo B para el proyecto.....</i>	<i>123</i>
<i>Figura 72. Componentes de Hardware para el Proyecto.....</i>	<i>123</i>
<i>Figura 73. Protoboard y componentes conectados.....</i>	<i>124</i>
<i>Figura 74. Placa de Raspberry Pi conectado</i>	<i>124</i>
<i>Figura 75. Cámara Megapíxel Prendida.....</i>	<i>125</i>
<i>Figura 76. Prueba de Hardware Prendido.....</i>	<i>125</i>
<i>Figura 77. Prueba de Conexión Wifi conectado a la red local.....</i>	<i>126</i>
<i>Figura 78. Pantalla de Alertas del Sistema Web</i>	<i>133</i>
<i>Figura 79. Pantalla de Edición de Alertas del Sistema Web</i>	<i>134</i>
<i>Figura 80. Pantalla del Menú de Manejo de Entrada y Salida de Pacientes</i>	<i>137</i>
<i>Figura 81. Pantalla de Ingreso de Paciente</i>	<i>137</i>
<i>Figura 82. Pantalla de Reasignación de Paciente.....</i>	<i>137</i>
<i>Figura 83. Pantalla de Selección de área.....</i>	<i>138</i>
<i>Figura 84. Pantalla de Asignación de Cama.....</i>	<i>138</i>
<i>Figura 85. Pantalla de Asignación de Cama (Paciente asignado a una cama).....</i>	<i>138</i>
<i>Figura 86. Pantalla de Salida de Pacientes.....</i>	<i>139</i>
<i>Figura 87. Pantalla de Lista de Alertas del Sistema Web.....</i>	<i>141</i>
<i>Figura 88. Pantalla de Administración del Sistema Web</i>	<i>142</i>
<i>Figura 89. Pantalla de Cambiar de contraseña.....</i>	<i>142</i>
<i>Figura 90. Pantalla de Creación de Usuarios.....</i>	<i>143</i>
<i>Figura 91. Pantalla de Búsqueda Avanzada de Registros.....</i>	<i>144</i>
<i>Figura 92. Pantalla de Información de Usuarios</i>	<i>144</i>
<i>Figura 93. Mockup de Reporte Diario del Sistema.....</i>	<i>146</i>
<i>Figura 94. Pantalla de Reporte diario del Sistema.....</i>	<i>147</i>
<i>Figura 95. Mockup de Reporte Mensual del Sistema</i>	<i>148</i>
<i>Figura 96. Pantalla principal de Reportes</i>	<i>150</i>
<i>Figura 97. Pantalla de Reporte Mensual del Sistema.....</i>	<i>151</i>
<i>Figura 98. Pantalla de Constancia de Atención.....</i>	<i>152</i>
<i>Figura 99. Interpretación de Defectos de un Contorno.....</i>	<i>159</i>

<i>Figura 100. Array de Defectos de la Imagen Capturada.....</i>	<i>159</i>
<i>Figura 101. Array y longitud de contornos de la Imagen.....</i>	<i>160</i>
<i>Figura 102. Interpretación del Teorema del Coseno.....</i>	<i>162</i>
<i>Figura 103. Valor de Conversión de Radíán a Grado Sexagesimal.....</i>	<i>162</i>
<i>Figura 104. Mensaje Enviado una vez calculado la cantidad de Defectos</i>	<i>164</i>
<i>Figura 105. Prueba de Falso Positivo.....</i>	<i>165</i>
<i>Figura 106. Prueba de Falso Positivo superada</i>	<i>165</i>
<i>Figura 107. Comparación de Áreas de Contornos para detectar el movimiento.....</i>	<i>166</i>
<i>Figura 108. Ingreso a la categoría 2(Detección de movimiento).....</i>	<i>167</i>
<i>Figura 109. Detección de movimiento funcional.....</i>	<i>168</i>
<i>Figura 110. Maqueta de prueba</i>	<i>169</i>
<i>Figura 111. Componentes conectados en el Raspberry Pi dentro de la maqueta.....</i>	<i>169</i>
<i>Figura 112. Maqueta con Led azul prendido.....</i>	<i>170</i>
<i>Figura 113. Maqueta con Led naranja prendido.....</i>	<i>170</i>
<i>Figura 114. Prueba de detección exitosa.....</i>	<i>171</i>
<i>Figura 115. Prueba de detección con funcionamiento de componentes.....</i>	<i>171</i>
<i>Figura 116. Prueba de detección desde otra perspectiva.....</i>	<i>172</i>
<i>Figura 117. Atributos de la clase Sql.....</i>	<i>172</i>
<i>Figura 118. Mensajes Enviados a la Base de Datos.....</i>	<i>173</i>
<i>Figura 119. Mensaje Registrado y Comprobado en la misma Base de Datos.....</i>	<i>173</i>
<i>Figura 120. Alertas mostradas en el Sistema Web</i>	<i>174</i>
<i>Figura 121. Verificación de la Existencia de la Cama 13 en la base de Datos.....</i>	<i>174</i>
<i>Figura 122. Comprobación del funcionamiento del Led 1</i>	<i>175</i>
<i>Figura 123. Comprobación del funcionamiento del Led 2</i>	<i>176</i>
<i>Figura 124. Comprobación del funcionamiento del Led 3</i>	<i>176</i>
<i>Figura 125. Prueba de avisos de cámara apagada</i>	<i>177</i>
<i>Figura 126. Test Ejecutados y Pasados.....</i>	<i>178</i>
<i>Figura 127. Test Ejecutados y Pasados en Consola.....</i>	<i>179</i>
<i>Figura 128. Grafica de resultados de la pregunta a los pacientes.....</i>	<i>180</i>
<i>Figura 129. Grafica de resultados de la pregunta a los pacientes.....</i>	<i>180</i>
<i>Figura 130. Grafica de resultados de la pregunta a los pacientes.....</i>	<i>181</i>
<i>Figura 131. Grafica de resultados de la pregunta a los pacientes.....</i>	<i>181</i>
<i>Figura 132. Grafica de resultados de la pregunta a los pacientes.....</i>	<i>182</i>
<i>Figura 133. Grafica de resultados de la pregunta a los pacientes.....</i>	<i>182</i>
<i>Figura 134. Grafica de resultados de la pregunta al personal.....</i>	<i>184</i>
<i>Figura 135. Grafica de resultados de la pregunta al personal.....</i>	<i>184</i>
<i>Figura 136. Grafica de resultados de la pregunta al personal.....</i>	<i>185</i>
<i>Figura 137. Grafica de resultados de la pregunta al personal.....</i>	<i>185</i>
<i>Figura 138. Grafica de resultados de la pregunta al personal.....</i>	<i>186</i>
<i>Figura 139. Grafica de resultados de la pregunta al personal.....</i>	<i>186</i>
<i>Figura 140. Grafica de resultados de la pregunta al personal.....</i>	<i>187</i>
<i>Figura 141. Grafica de resultados de la pregunta al personal.....</i>	<i>187</i>
<i>Figura 142. Grafica de resultados de la pregunta al personal.....</i>	<i>188</i>

ÍNDICE DE TABLAS

<i>Tabla 1. Librerías que ofrece Aforge</i>	20
<i>Tabla 2. Comparación entre CubieBoard y Raspberry Pi</i>	30
<i>Tabla 3. Tabla de comparación de Frameworks de Reconocimiento de Patrones</i>	60
<i>Tabla 4. Tabla de comparación de Hardware Libres</i>	78
<i>Tabla 5. Matriz de requerimientos del Sistema</i>	82
<i>Tabla 6. Matriz de requerimientos del sistema y del programa</i>	83
<i>Tabla 7. Tabla de Señales predeterminadas del sistema</i>	157
<i>Tabla 8. Tabla de Resultados de las coordenadas</i>	163



RESUMEN

En la actualidad, el reconocimiento de patrones es una ciencia bastante difundida; podemos observar reconocimiento facial, identificación de huellas dactilares, detectores de movimiento, etc. Sin embargo, a pesar de toda la tecnología a nuestro alrededor, algunos procesos se siguen realizando como se hacía hace años atrás, sin buscar una nueva solución, un ejemplo de ello es la actual manera de atender a un paciente hospitalizado.

El presente trabajo presenta un sistema web capaz de manejar información de atención a los pacientes hospitalizados y del personal, con la finalidad de registrarlos y evaluarlos. Para obtener esa información, se hace uso del Raspberry Pi y su módulo de Cámara Megapíxel el cual evalúa las señales que el paciente indique a la cámara; interpretándolas como una solicitud de atención con mensajes definidos por el sistema según el tipo de señal que se le haya enviado a la cámara.

Aprovechando que el Raspberry Pi es un pequeño procesador, este dispositivo es el que utiliza los algoritmos de reconocimiento de patrones para interpretar la señal; además almacena en la base de datos del servidor la información obtenida, la cual el sistema web procesará y brindará la opción de tener un registro completo de la cual un operador haga uso para tener un control de las solicitudes de atención que se generen, al igual que la búsqueda y actualización de estas. Identifica también quién atendió al paciente, cómo lo hizo, si hubo algún incidente, cuándo se solicitó, etc.

Las herramientas por utilizarse en este proyecto son librerías de Python, Frameworks que facilitan la creación de la plataforma web y el uso de reconocimiento de patrones.

PALABRAS CLAVE: RECONOCIMIENTO DE PATRONES, RASPBERRY PI, CÁMARA MEGAPÍXEL.

ABSTRACT

At present, pattern recognition is a science that is already more common than before, we can observe facial recognition, fingerprint identification, motion detectors, etc. However, some processes, despite all the technology around us, are still being carried out as was done years ago, without looking for a new solution, an example of which is the current way of treating a patient hospitalized.

The present work consists of presenting a web system capable of handling care information for hospitalized patients, in order to register and evaluate hospital staff. To obtain this information, a Raspberry Pi and its Megapixel Camera model are used which will allow us to evaluate the signals that the patient indicates to the camera interpreting them as a request of attention with messages defined by the system according to the type of signal sent to the camera.

Taking advantage of the fact that Raspberry Pi is a small processor, it will be the one that makes use of the algorithms of pattern recognition to interpret the signal, as well as to store in the database of the server, the information obtained, which the web system will process and give the option to have a complete record of which an operator uses to have a control of the requests for care that are generated, as well as the search and update of these, with this it will also be possible to identify who attended the patient, as he did, If there were any incidents, when requested, etc.

The tools to be used in this project are Python libraries and frameworks that facilitate the creation of the web platform and the use of pattern recognition.

KEYWORDS: PATTERNS RECOGNITION, RASPBERRY PI, CAMERA MEGAPÍXEL.

INTRODUCCIÓN

Actualmente, en los hospitales de nuestra ciudad, la calidad de atención a los pacientes ha disminuido debido al poco personal y los pacientes han aumentado. Según el Diario la Republica (01 de enero, 2016), El Jefe Nacional del SIS informó que los hospitales de Arequipa devolvieron 14 millones de soles que deberían haber sido utilizados para mejorar la atención de los pacientes, eso incluye invertir no solamente en medicinas para los pacientes, sino también en diferentes elementos para mejorar su atención, lo que también, incluye proyectos para el hospital. Sin embargo, los hospitales como el Honorio Delgado, solamente en el 2015, el Seguro Integral de Salud (SIS) les entregó S/. 31 millones 109 mil, pero solo gastó S/. 19 millones 840 mil. Por lo que el hospital devolvió S/. 11 millones 269 mil.

Los pacientes que están internados en los hospitales son quienes sufren las consecuencias. Personas que pasaron por esa situación conocen que en el hospital la única forma de comunicarse con una enfermera o auxiliar, es alterando el orden y calma del lugar con escandalosos llamados al personal; lo cual incomoda a varios de los pacientes que están en la habitación, o también de otras habitaciones. Existen los botones de atención, pero no funcionan.

Este proyecto plantea crear un prototipo utilizando la tecnología de Raspberry Pi y módulos de cámara del dispositivo en conjunto con inteligencia artificial para mejorar la atención a los pacientes internos del hospital. Principalmente, la cámara vigilará a los pacientes y hará uso de un código de programación para lograr detectar gestos e informar a los encargados de su cuidado, registrando también a las personas que los atendieron.

La tesis está constituida de cuatro capítulos en los cuales se tratarán los siguientes puntos:

- En el capítulo 1, Planeamiento del Proyecto; definimos los objetivos, los alcances y limitaciones, justificación, identificar las variables del proyecto y mencionar algunos aspectos relevantes del Proyecto
- En el capítulo 2, Marco Teórico, resaltamos proyectos que son similares a este, algunas ideas planteadas y presentadas anteriormente en la sección del Estado del Arte y conceptos importantes y necesarios para entender el proyecto
- En el capítulo 3, Descripción del Sistema y Desarrollo, se explica las actividades que se realizaron para obtener la información y los datos para el proyecto; resaltando la opinión del estado actual del escenario donde se plantea aplicar el proyecto. Los requerimientos del software y hardware lo definimos a partir de la información y datos adquiridos; además de identificar los requerimientos del hardware necesario para la creación del prototipo y generar una propuesta del modelo, en la cual se explica la arquitectura del sistema, el cual incluye la base de datos, Diagramas de casos de uso, Diagramas que expliquen cómo está compuesto el sistema y los usuarios que harán uso del sistema, detallando los subsistemas del proyecto
- En el capítulo 4, Pruebas y Resultados, se comprueba el funcionamiento del prototipo y el logro de los objetivos del proyecto mediante los datos de entrada que se indiquen.
- Finalmente presentamos las conclusiones y recomendaciones del proyecto

CAPITULO 1: PLANTEAMIENTO TEÓRICO

1.1. Descripción del Problema

En la ciudad de Arequipa, el servicio de salud que más impacto tiene es el de EsSalud al poseer más de 5 locales en la ciudad blanca los cuales brindan servicios a las personas aseguradas.

En el área de hospitalización, se ofrece un servicio de cuidado y atención al paciente en su periodo de recuperación. Actualmente, cuando un paciente hospitalizado requiere ayuda, debe usar un sistema de botoneras que permiten avisar al personal que necesita ayuda; sin embargo, este sistema es de baja calidad en cuanto a funcionamiento y eficacia, ya que la mayoría de pacientes al ver que el sistema no funciona de manera correcta para ellos, necesitan gritar o pedirle a otro paciente que use su botonera para que pueda recibir la atención necesaria. Es una realidad de la cual solo se piensa en arreglar un sistema que es ineficiente y vulnerable a fallos, en vez de cambiar a otro sistema que sea eficaz, complementado, económico, sobre todo, moderno.

1.2. Objetivos

Implementar un prototipo de reconocimiento de patrones de movimiento para mejorar la atención de los pacientes internados de un nosocomio haciendo uso de Raspberry Pi con su módulo de cámara Megapíxel

Objetivos Específicos

1. Explorar la tecnología, software y hardware que se utilizará para el desarrollo del proyecto, indicando los pasos realizados.
2. Identificar las herramientas para implementar un sistema de reconocimiento de señales en una cámara

3. Explorar el estado actual de atención a pacientes internos
 4. Diseñar una plataforma web para el procesamiento de datos, tanto como la creación, observación y actualización de datos, así como la elaboración de reportes y otras labores afines.
 5. Conectar el Raspberry Pi con la Plataforma Web para envío de datos capturados
 6. Desarrollar un prototipo funcional a pequeña escala haciendo uso de herramientas de visión computacional para el análisis de las imágenes y generar una interpretación clara de ellas.
 7. Verificar y validar el prototipo
- 1.3. Alcances y limitaciones
- Alcances
- El proyecto tiene como alcance implementar el prototipo que tomará como referencia los cuartos de hospitalización de EsSalud.
 - Crear un prototipo de sistema web capaz de trabajar mutuamente con los datos que el Raspberry Pi y su Cámara Megapíxel brinden al sistema
 - Reconocimiento de señales indicadas solo por las manos
- Limitaciones
- La limitación principal es la implementación en un ambiente real
 - Contar con un solo Raspberry Pi para su implementación.

1.4. Justificación

Principalmente, la atención a pacientes internados en EsSalud es un problema no tanto por la capacidad del personal; sino por la misma atención en sí. La vigilancia no puede ser constante al momento de tener un paciente internado, justamente, existen varios pacientes internados, y cada miembro del personal médico tiene labores y obligaciones que cumplir no solo con uno, sino con varios, es por ello, que ocurren problemas. Actualmente, en los centros de EsSalud de Paucarpata y Yanahuara, tienen el sistema de botones en bajo funcionamiento para atención, no obstante, los pacientes tienen que gritar o golpear las paredes o camas para lograrlo. El Hospital Carlos Alberto Segúin Escobedo, cuenta con el sistema de botones, pero el personal no colabora con el caso ni atiende como debería a los pacientes; entonces, requiere de un sistema de vigilancia de 24 horas del día y 7 días de la semana (24x7) para poder corregir todos esos problemas, tanto del paciente como del personal, para así mejorar el servicio de atención de EsSalud

1.5. Aspectos relevantes del Proyecto

Este proyecto a través de un sistema web desarrollado en Python y con Django, un framework muy conocido para el desarrollo web. Nos servirá para recibir alertas generadas por los mismos pacientes en la cámara megapíxel del Raspberry Pi, revisar detalles de atención, generar reportes y visualizar a los pacientes.

CAPITULO 2: MARCO TEÓRICO

2.1. Estado del Arte

La tecnología cada vez es más eficaz y el uso de Raspberry Pi no es la excepción, son muchos los proyectos e ideas materializadas gracias a esta pequeña computadora, y el reconocimiento de patrones es una ciencia que existe ya desde hace muchos años y con ayuda de nuevos Frameworks y softwares es más fácil implementarlos. Aunque, actualmente, no hay un proyecto desarrollado exclusivamente para mejorar la atención en áreas de hospitalización de los centros de salud que incluya el uso de Raspberry Pi y uso de Frameworks de visión computacional; este proyecto sería el primero de su tipo.

Abaya, Basa, Abad y Dadios (2014) desarrollaron un sistema de video vigilancia nocturna el cual era especial para la detección del movimiento en una habitación y el control de humos; ellos utilizaron Open CV, pero aplicado en C++ el cual incluso transmite una alerta vía Wi-Fi al correo electrónico con una imagen de lo registrado por la cámara en ese momento; se hicieron diferentes pruebas las cuales registran verdaderos y falsos positivos o negativos, dando a concluir resultados alentadores, con la mayoría de ellos positivos verdaderos y un par de negativos falsos. No solo se consideraron los algoritmos típicos para su desarrollo, sino también los factores lógicos y externos, como en el caso de la detección del humo, físicamente, factores externos como el viento y sonido no afectan el movimiento del humo, este siempre se mueve con mayor fuerza en dirección horizontal hacia arriba, lo cual también es parte considerable a tomar en cuenta, para la modificación del algoritmo de detección.

Otros de los proyectos realizados es el de Merin y Razmid (2014) el cual trata de la detección del flujo de movimiento en forma óptica con Raspberry Pi. Principalmente, este proyecto se realizó para que la cámara detecte el más mínimo movimiento en el registro; se trabajó con Open CV y Python; los algoritmos se trabajaron con límites matemáticos, estos algoritmos son los de Lucas y Kanade analizando el movimiento de la imagen, por la cantidad de píxeles desplazados, esto indica el movimiento; sin embargo, indicar la dirección del movimiento depende de un tiempo de diferencia entre ambos, si se analiza la imagen estativa, se tendrá los píxeles vecinos, al moverse los vecinos también se moverán en una dirección igual, haciendo un análisis de los vecinos con el centro de la primera imagen con la segunda, se obtiene la dirección a donde se produjo el movimiento.

Soetedjo, Mahmudi, Ibrahim y Nakhoda (2014) presentan un trabajo realizado con Raspberry Pi y Open CV para lo que es una puntería laser de mayor eficiencia. Este sistema está escrito en lenguaje C con Open CV, el algoritmo está diseñado a la variación de color, teniendo en cuenta los valores RGB en píxeles que toma cada color; es decir, de 0 a 255 es la escala de valor de los colores, dependiendo del color el algoritmo define el color del puntero, si está en una superficie no tan blanca, el puntero se tornará azul con borde rojo no tan claro, y si la imagen es más densa, más oscura, el borde rojo se expandirá y el azul resaltará de esa manera. La cámara que se utilizó fue una webcam Logitech C920 la cual tiene una resolución de 1920 x 1080px. con drivers fáciles de instalar en Debian, en este caso Raspbian. Se evaluó los resultados con la distancia desde 0.5m a 3m, las cuales en su mayoría fueron exitosas. Variando la luz o sin variar.

Tan, Hoong, Ken Hong y Zhi Wen (2014) presentaron una implementación para estacionamientos de carros, utilizando Raspberry Pi, Open CV y Twitter. Principalmente, el objetivo de su proyecto es brindar a los usuarios vía Twitter información acerca del parque de carros; indicando cuales están disponibles y cuáles no, esto durante el día únicamente, la solución se logra a bajo costo y con poco esfuerzo. El flujo de trabajo es primero la captura de imagen, pre procesamiento de esta, detección del parqueo disponible, por último twittear las opciones disponibles. Sin entrar a detalles sobre el algoritmo, este únicamente hace un procesamiento de la imagen y al tener los datos disponibles, los convierte a información que será twitteada para los interesados.

Sutoyo, Prayoga, Suryani y Shodiq (2015) presentaron una implementación realizada con una librería Open CV además de usar visión computacional.

El proyecto consta de lo siguiente:

Principalmente, plantearon que en una presentación o clase, los profesores usualmente utilizan el teclado del computador o incluso los punteros; pero como hemos visto, esto puede ser fastidioso o incluso molesto para los presentadores ya que deben darse un tiempo para que puedan llegar al computador y presionar siguiente para continuar exponiendo o en el caso de los punteros o herramientas, suelen fallar. Es una herramienta útil que ayuda al presentador permitiendo desplazarse a su voluntad, sin depender de la instalación del puntero o controlador al computador.

Esto se implementó un sistema de reconocimiento de las manos y dedos de los presentadores en pantalla del reflector para que básicamente se toque en cualquier parte de la pizarra y esta pueda reaccionar de acuerdo con el gesto implementado. Para analizar el planteamiento, se generaron algunas variables

como el número de data entrenada que necesita la aplicación, la intensidad de luminosidad para detectar, la distancia entre el objeto y la cámara, etc. Los objetivos primordiales que presenta la aplicación son el de construir una aplicación capaz de interpretar patrones de la mano para generar eventos como el click o doble click del mouse; implementar patrones para controlar la misma presentación que incluye el desplazarse a la primera diapositiva, a la última, la anterior, la siguiente, y finalizar la presentación.

Otro de los documentos al cual el anterior también hizo referencia es el descrito por Junhao, Junji y Yiye (2012). Crean una aplicación para el control de la música y videos vía webcam incorporando el reconocimiento de gestos, ayudando a las personas a utilizar y controlar los elementos multimedia como Spotify y iTunes sin necesidad del uso del teclado o mouse, solo el uso de una webcam que reconozca gestos los cuales puedan interpretar funciones como Play, Pause, Stop, Next/Preview, y Reload. Diseñado para computadoras Mac únicamente, este sistema se diseña en Matlab con algoritmos de movimiento y reconocimiento implementando incluso la distancia que tendría que tener la mano, gestos principales y color de piel.

El último paper que también es referenciado en el paper de Sutoyo, es el presentado por Nana Ramadijanti y Raga Mukti (2013). Presentan un sistema de reconocimiento de gestos mediante webcam para aplicar simples acciones como el movimiento del mouse, cursores, abrir aplicaciones, operar el sistema a media capacidad básicamente. Para esto, hacen uso de una data para entrenar el sistema de tamaño moderado, incluye 2500 negativos y 2400 positivos; también utilizan la conversión en cascada dentro de XML y detección de objetos, depende de la data base para que el sistema entrene sus resultados y brinde mejor información.

Otros puntos que relatan el paper que no se tocaron es la intensidad de la luz, distancia y movimiento de objetos continuos.

Los trabajos anteriores brindan bastantes mejoras a diversas áreas o procesos. Unos proponían trabajos desarrollados en Raspberry Pi para la detección del movimiento, otros con reconocimiento de gestos, etc. La implementación para el sector salud es nula ya que no se encontraron proyectos que brinden una mejora a centros de salud que requieren de tecnología para mejorar la calidad de atención al paciente. Esto es justamente lo que se propone con este trabajo, acomodar las ideas que se tuvieron en estos proyectos y plasmarlas en el área de salud, más concretamente, en la atención a pacientes hospitalizados, ya que la tecnología de estos Frameworks de visión computacional y hardware libre es una alternativa bastante accesible, económica y eficaz para implementar un sistema que mejore la calidad de atención de las personas mediante un sistema que interactúe con ellos en el mundo real; reconociendo gestos, movimientos que el computador pueda interpretar.

2.2. Bases Teóricas

2.2.1. Reconocimiento de Patrones

El reconocimiento de patrones se puede definir con la obtención de data del mundo real, dentro del área de aprendizaje de máquinas, el cual tiene como propósito lograr hacer que el sistema reconozca diversos patrones, darles un significado, un valor y que sean interpretados por el computador (Ver Figura 1).

Lograr que las computadoras puedan procesar, analizar e interpretar imágenes igual que nosotros es una nueva escala en el futuro de la computación; si bien ya existe, está por desarrollarse a mayor escala, todo esto gracias a la inteligencia

artificial. Primeramente, veamos qué es la inteligencia artificial. John McCarthy dijo en 1956 en el instituto de Tecnología de Massachusetts la siguiente definición: “La inteligencia artificial es el puente entre las ciencias de la computación con crear computadoras que actúen igual que un humano”. Esta definición abrió la puerta a nuevas formas de desarrollar tecnología capaz de lograr dicho objetivo. Alan Turing, conocido criptólogo de la segunda guerra mundial, dijo que si ponemos a conversar a una persona con una máquina y durante 10 min la persona no se da cuenta que es una máquina, se puede considerar inteligente, esta prueba se le denominó “El test de Turing”. Actualmente, el desarrollo de chat-bot es la prueba más cercana a que se está llegando a superar esa prueba.

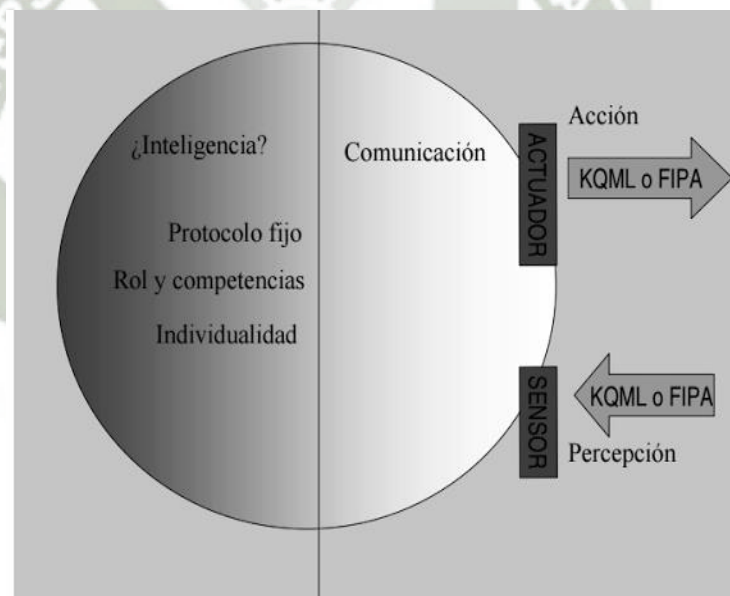


Figura 1. Proceso Básico de Inteligencia Artificial

Fuente: Recuperado de <http://www.giaa.inf.uc3m.es/>

La inteligencia artificial tiene varios campos de estudio, entre los cuales, los principales son (Ver Figura 2):

- Programación Lógica

- Lógica de Primer Orden
- Reconocimiento de Patrones
- Redes Neuronales
- Sistemas Expertos
- Criptografía
- Conciencia Humana
- Programación por Hechos
- Robótica
- Web Semántica
- Sistemas de apoyo a la decisión
- Sistemas multi-agente e IA distribuida
- Algoritmos Genéticos

En este proyecto, se hace uso de herramientas de reconocimiento de patrones, el cual es uno de los más pioneros y resalta lo que es la Visión Computacional. Como lo dijo Marr, Rosenfeld, Lawrence y Horn (1995); la visión computacional es aquello que incluye la adquisición, procesamiento, análisis y entendimiento de imágenes, es decir, entendimiento de data de alto nivel la cual es adquirida del mundo real en orden de procesos numéricos e información simbólica.

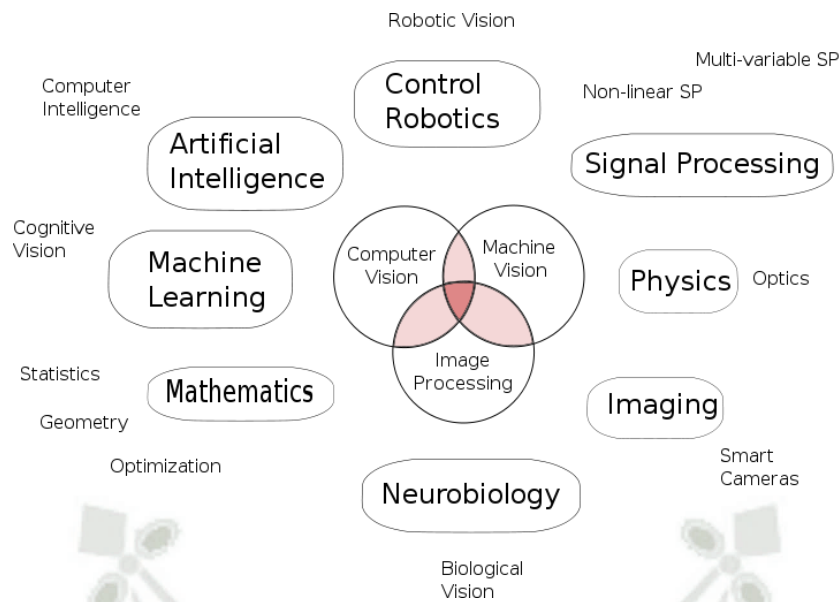


Figura 2. Relación de la Visión Computacional con otros avances.

Fuente: KDNuggets.com, 2012

Primeramente, la inteligencia artificial fue fruto de los estudios de psicología cognitiva y la lógica matemática, intentando encontrar una relación al comportamiento y reconocimiento. A mediados de los años 60, el MIT (Massachusetts Institute of Technology) se planteó el estudio de la visión computacional, lo cual implica no solo que se capte una imagen, sino que esta llegue a ser comprendida por el sistema. Entre los avances y prácticas de sus investigaciones estuvo la creación de un robot que podía reconocer y apilar objetos sobre una mesa; el proyecto estuvo a cargo de Larry Roberts, Gerald Sussman, Adolfo Guzmán, Patrick Winston, David Huffman, entre otros.

Los avances en los años 70 se dieron principalmente en la representación del conocimiento, empezando por mapas conceptuales hasta bases del conocimiento, las cuales eran sentencias lógicas que, aunque efectivas, se veían limitadas por el almacenamiento de la época. Un ejemplo de esas bases del conocimiento es la siguiente que se tomó como ejemplo de Facebook:

If facebook.contactos=contactos.amigos then contactos.amigos=fotos.amigos

If facebook.contactos=contactos.amigos then contactos.amigos=informacion.amigos

If facebook.contactos=contactos.amigos then contactos.amigos=comentarios.amigos

En una BC(Base de Conocimiento) se tienen las siguientes características:

- Almacenar Hechos
- Almacenar Inferencias
- Generar Nuevos Hechos
- Responder Preguntas

Actualmente, el reconocimiento de patrones es un hecho real; algo que ya el mundo puede ver que existe y usarlo. Básicamente, consta de 3 simples procesos (Ver Figura 3):

- Adquirir Datos
- Extraer Características
- Tomar Decisiones

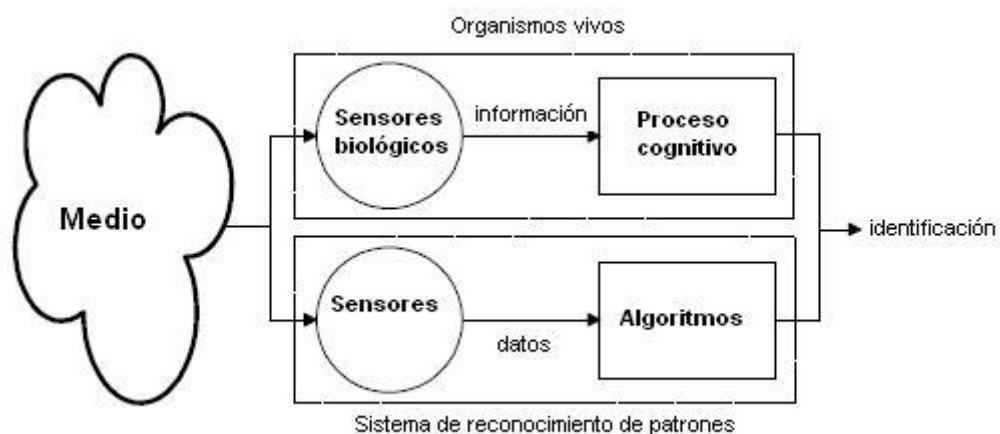


Figura 3. Ciclo de Vida de un sistema de Reconocimiento de Patrones

Fuente: Wikiwand

Sistema Básico de Reconocimiento

Como se puede ver en la anterior imagen, un sistema básico de reconocimiento tiene las siguientes partes:

- **Sensor**

Es el encargado de la adquisición de datos, puede tomar diferentes tipos de datos dependiendo de su función; temperatura, densidad, iluminación, aceleración, etc.

En nuestro caso es interpretar movimientos.

- **Extractor de características**

Es aquí donde se procesa la data obtenida por los sensores, principalmente, se debe hacer una limpieza de datos o preprocesamiento ya que siempre existen unas pequeñas derivaciones para poder pasar esa data a la respectiva extracción de características que van a influir a futuro en la toma de decisiones.

- **Clasificador**

Aquí se empieza a agrupar o clasificar los datos para que puedan ser interpretados de manera adecuada según el vector de características obtenidas.

Este método se denomina aprendizaje automático, el cual es el desarrollo de procesos que permiten a las computadoras aprender nuevos conocimientos.

Estos se dividen en diferentes tipos:

- **Clasificación Supervisada:** Se trata de áreas en las que se conoce a priori la clase a las que pertenecen los datos y sus características
- **Clasificación Parcialmente Supervisada:** Solo se conocen algunas clases mientras que otras no están definidas, pero tienen que ser tomadas en cuenta

- Clasificación No Supervisada: Es más complicada ya que se hace uso de algoritmos los cuales crean clasificación multivariante haciendo que los grupos se vayan creando dependiendo de los individuos (Duda , Hart y Stork, 2001)

2.2.2. Frameworks de Reconocimiento de Patrones

Algunos Frameworks de reconocimiento de patrones ayudan al desarrollo del software rápido de este tipo, ya que contienen varios de los algoritmos implementados; a continuación, algunos de estos Frameworks y el que se escogió para el desarrollo del proyecto:

1) IGesture

Es un framework basado en Java que se encarga del reconocimiento de gestos, enfocado en extensibilidad y reusabilidad, incluye herramientas para la creación de gestos que complementan el framework.

Este framework es libre, está basado en Apache 2.0; la página principal ofrece publicar trabajos para motivar a demás desarrolladores a la retroalimentación y crecimiento del framework.

Para su uso, primeramente, nos vamos a la página principal:
<http://www.igesture.org/download.html>

La cual nos brinda los archivos de descarga desde repositorio o también está viable en SourceForge.net. Ofrece un paquete de instalación en Eclipse, esta plataforma reconoce las librerías de IGesture instaladas

- **Ventajas**

- Es un framework para Java
- Se basa principalmente, en el reconocimiento de gestos

- **Desventajas**

- No tiene actualizaciones desde 2007

- Es anticuado
- No existen avances en el framework ni otras aplicaciones.

IGesture solo tiene soporte para el lenguaje de Java, además de ser anticuado y carecer de nuevas actualizaciones es un framework útil en cuanto a detección, pero solo en Java y con algoritmos limitados (IGesture,2006).

2) OpenCV

Es el framework más usado para visión computacional, además de ser de licencia libre para la comunidad, desarrollado primeramente para Python, pero extendido para los demás lenguajes como C#, C/C++, Android, etc.

Este framework actualmente está en la versión 3.0.0 está en constante crecimiento y mejora por parte de la comunidad, la cual contribuye con nuevas mejoras, al ser de código abierto, cada mejora es libre de utilizar por la comunidad.

La información y documentación brindada sobre este framework en la página oficial es completa en cuanto a su desarrollo de la versión 3.0.0, podemos trabajar con diversos elementos de multimedia para generar Visión Computacional (OpenCV.org, 2016).

- **Ventajas**

- Es multiplataforma
- Está en continua modificación y contribuciones por parte de la comunidad
- Es totalmente gratis
- Conjuntos de datos para hacer uso de algoritmos que requieren datos en conjunto para aprendizaje

- **Desventajas**

- La instalación de la librería requiere compilación, lo que complica un poco la instalación del framework
- Poca documentación sobre los algoritmos utilizados (LearnOpenCV,2015).

3) AForge.NET

Es un framework C# diseñado para desarrolladores e investigadores en los campos de Visión Computacional e Inteligencia Artificial.

Esta contiene diversas sub-librerías que tienen funciones específicas para el reconocimiento de patrones (Ver Tabla 1).

Tabla 1. Librerías que ofrece Aforge

AForge.Imaging	Librería con rutinas de procesamiento de imágenes
AForge.Vision	Librería de visión computacional
AForge.Video	Conjunto de librerías para el procesamiento de video
AForge.Neuro	Librería para el desarrollo de redes neuronales
AForge.Genetic	Librería para el desarrollo de evolución y algoritmos genéticos
AForge.Fuzzy	Librería de lógica difusa
AForge.Robotics	Librería que brinda herramientas para el desarrollo robótico
AForge.MachineLearning	Librería para el aprendizaje de la máquina.

Fuente: Aforge.Net

Actualmente, está en la versión 2.2.5, es parte de la licencia LGPL v3, esta licencia básicamente nos indica que está permitido copiar y distribuir, sin embargo, no se pueden realizar cambios a las librerías. Solo está desarrollada para plataformas .NET (Ver Figura 4).

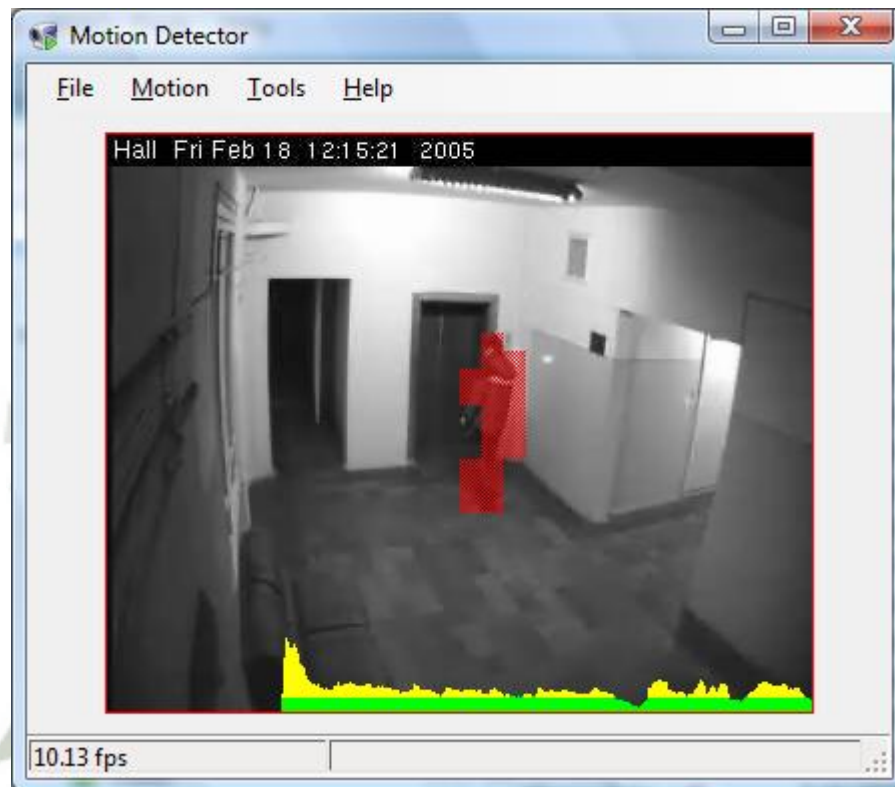


Figura 4. Ejemplo de Aforge.Net

Fuente: AForge.NET, 2013

- **Ventajas**
 - Este framework contiene una extensa cantidad de algoritmos para diversos temas de inteligencia artificial, no solo para reconocimiento de patrones
 - Aplicabilidad para plataformas .NET
 - Facilidad de combinar algoritmos de diferentes temas de visión computacional, es decir, combinar algoritmos genéticos con redes neuronales o reconocimiento de patrones.

- **Desventajas**

- No tiene varianza en otros lenguajes de programación como Java y Python.
- No se puede hacer cambios a la librería ni mejoras por parte de la comunidad.
- No tiene soporte para Hardware Libre.

Más adelante se hablará del hardware libre, pero lamentablemente los hardware libres no tienen un soporte otro hardware que no sea un CPU o notebook que soporte sistemas operativos de Windows, y los hardware libres usan software libre como derivados de Debian/Linux. (Aforge.NET, 2008-2012).

2.2.3. Hardware Libre

Principalmente, el hardware libre se divide en función a su naturaleza en 2 tipos: el estático y el reconfigurable.

Hardware Estático

El hardware libre estático se refiere a aquel hardware que puede ser construido y modificado por cualquier persona, es decir, se le brinda la suficiente información para que la persona pueda construir uno desde 0. Como se sabe, existen cuatro libertades en lo que es el software libre:

- La libertad de usar el software
- La libertad de estudiarlo cómo funciona y modificarlo
- La libertad de distribuir copias del programa
- La libertad de buscar mejorarlo y hacer públicas esas mejoras con el fin de que toda la comunidad informática se beneficie

Si se intenta aplicar estas cuatro libertades al hardware, se comprueba que es imposible, por el simple hecho de la natural diferencia de ambas; un código se puede modificar directamente, sin embargo, un hardware, debe tener las piezas

que lo constituyen agrupadas y unidas; caso contrario no sería el mismo hardware. Los problemas que aparecen son los siguientes:

- Un diseño físico es único: se refiere a que el hardware es un componente físico, difícil de cambiarlo tal como es; es decir, o bien se usa una placa armada o bien se construye una igual, pero en esencia no se puede alterar.
- La compartición tiene asociado un coste: quiere decir que la construcción del hardware requiere de un diseño y componentes, y estos tienen un coste
- Disponibilidad de los componentes: puede que los componentes para su construcción estén disponibles en el país del creador; sin embargo, si alguna otra persona en otro lugar desea construir debe percatarse que los componentes estén disponibles en el lugar donde vive para su construcción.

Una propuesta de definición que ofrece la fuente es: *“Un diseño se considera hardware abierto si ofrece las 4 libertades del software libre en el esquemático, PCB y fichero para la fabricación (este último puede no estar disponible)”*

Ventajas del Hardware Libre

- Cualquier persona la puede fabricar
- Cualquier persona la puede modificar
- Cualquier empresa la puede comercializar

Hardware Reconfigurable

Un hardware puede ser descrito mediante lenguajes; los lenguajes HDL son aquellos que describen los diseños, estructura y operaciones de

circuitos los cuales también pueden ser utilizados para el análisis automático y simulaciones.

Ahora, para probar físicamente estos diseños tenemos lo que son los FPGAs; estos son dispositivos que permiten implementar circuitos digitales, están compuestos por componentes configurables llamados CLBs, estos componentes se pueden unir de diferentes maneras, dependiendo del diseño para formar un dispositivo y otro. La característica fundamental que tienen los FPGAs es que son universales y se pueden convertir en cualquier diseño digital.

Entonces, viendo el lenguaje HDL y los FPGAs, juntos forman el hardware reconfigurable, el cual con el primero se genera las simulaciones digitales de nuestros circuitos y en los FPGAs podemos hacer que se materialice, esta parte se denomina síntesis (Ver Figura 5.).

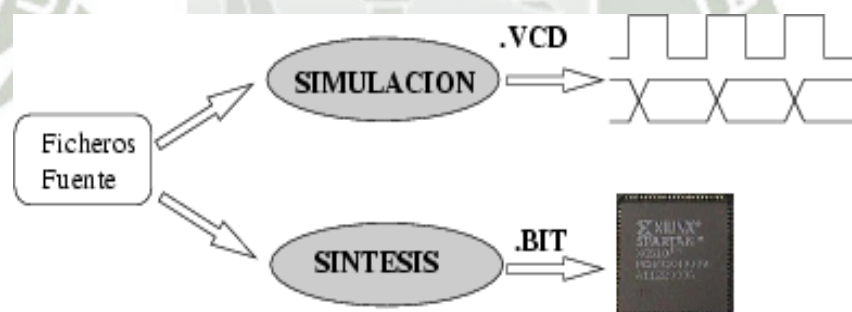


Figura 5. Ciclo del diseño de hardware reconfigurable

Fuente: Iearobotics, 2013

Principalmente, si se observa desde el punto de vista de los lenguajes que se utilizan para el diseño, se puede decir que, si se habla de hardware reconfigurable libre, tenemos las siguientes características:

- Se ofrece las 4 libertades del software libre en los ficheros HDL
- Surgen comunidades de hardware libre

- Se pueden crear repositorios donde se da acceso a otros interesados
- Aparecen distribuciones que recopilen todo el hardware libre existente
(González Iván, González Juan, Gómez Arribas Francisco, 2003).

2.2.4. Ejemplos de Hardware Libre

Existen actualmente varias placas que salen al Mercado y al público con diferentes usos y funcionalidades dependiendo lo que el usuario quiera (Ver Figura 6.), lo que más resalta en el hardware libre es el bajísimo costo a comparación de hardware garantizado por empresas grandes, ya que las funcionalidades muchas veces las mismas que conseguir hardware patentado, y lo mejor es que este tipo de hardware se puede mejorar a diferencia del patentado que solo se utilizará para una función específica sin opción a ser modificada por ningún motive (J.J. Velazco, 2013).

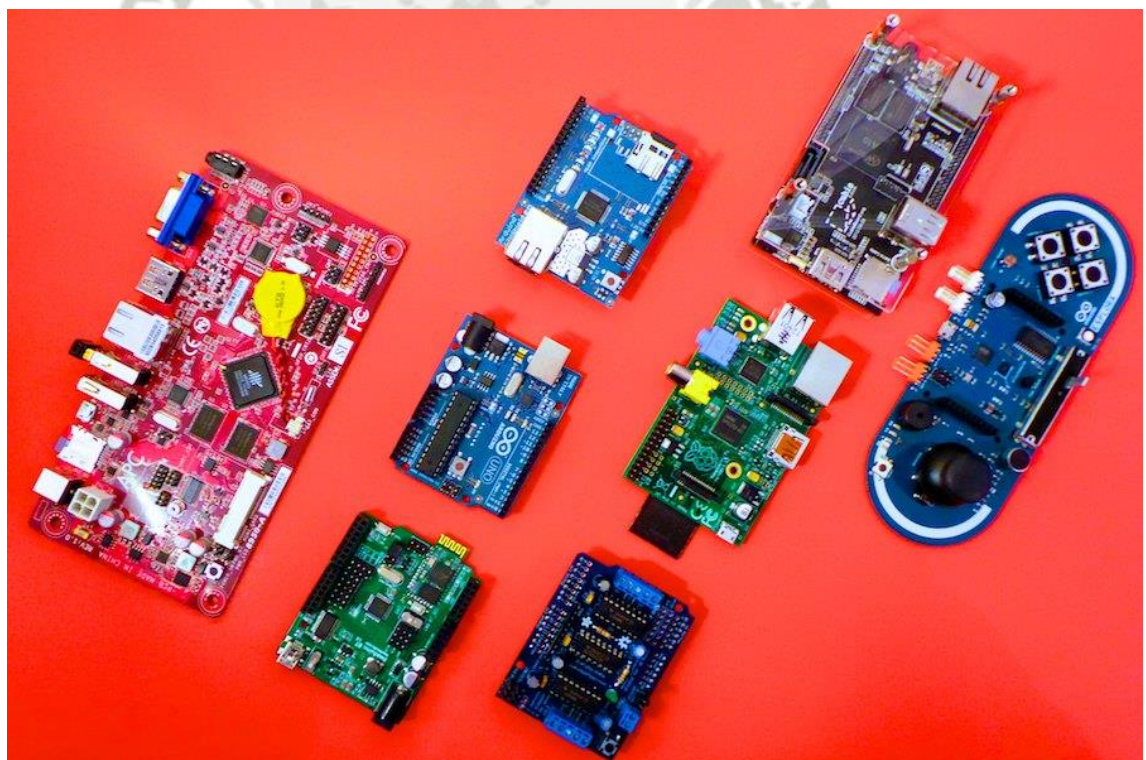


Figura 6. Imagen de varias placas de Hardware Libre

Fuente: techwatch.keeward, 2013

1) UzeBox

Este hardware, es definido como una consola la cual se puede desarrollar en casa sin necesidad de hacer mucho gasto en los componentes, Se basa en el microcontrolador AVR de 8 bits de Atmel. Se pueden desarrollar juegos programados en C, el objetivo es brindar con pocos componentes una consola con buen sonido y buenos gráficos (Ver Figura 7).

Entre sus características son:

- CPU: Microcontrolador ATmega644
- Total RAM: 4K
- Programa de la memoria: 64K
- Velocidad: 28.61818Mhz (overclocking)
- Colores: 256 colores simultáneos dispuestos en un espacio de color 03:03:02 (rojo: 3 bits, verde: 3 bits, azul: 2 bits)
- Salida de video: NTSC compuesto y S-Video
- Sonido: 8-bit mono, mezclado en ~15kHz y la salida vía PWM
- Entradas: Dos joypads NES/SNES compatibles
- Almacenamiento externo: SD/MicroSD
- Opciones: interfaz MIDI



Figura 7. Foto de un UzeBox con el respectivo mando de juegos conectado

Fuente: Belogic Uzebox , 2007

A partir del año 2010 se empezó a comercializar, ya como un componente de juegos y su sucesor, la Fuzebox, es con ese nombre con el que se distribuye, con entradas para mandos de SNES.

Lamentablemente, este hardware libre no funcionaría para el desarrollo del proyecto ya que es exclusivamente para juegos y no para procesar algoritmos o conectar cámaras web o de otro tipo.

Pero cabe mencionarlo ya que es parte del hardware libre, aunque no sea lo que se busca para el proyecto, es parte de otras funciones a las que si podría ayudar de mejor manera (Uzebox,2015-2016).

2) CubieBoard

Es la competencia directa del Raspberry Pi, ya que al igual que este, es un mini ordenador. El CubieBoard es una tarjeta en desarrollo, pero que esta por buen camino actualmente y sigue creciendo su producción y mejora del producto.

Actualmente, esta tecnología de hardware libre se encuentra en la versión 5 denominada CubieTruck Plus/Cubieboard5, el paquete incluye un cable usb/miniUsb, un cable SATA, un cable USB hembra, cable de poder, una base para el CubieBoard, enfriadores de aluminio y tornillos para fijar la base a la placa (Ver Figura 8).

Puede tener sistemas operativos derivados de Linux, y versiones especiales de Android para la placa; pero no tiene soporte o versión de SO de Windows o Apple para esta placa.

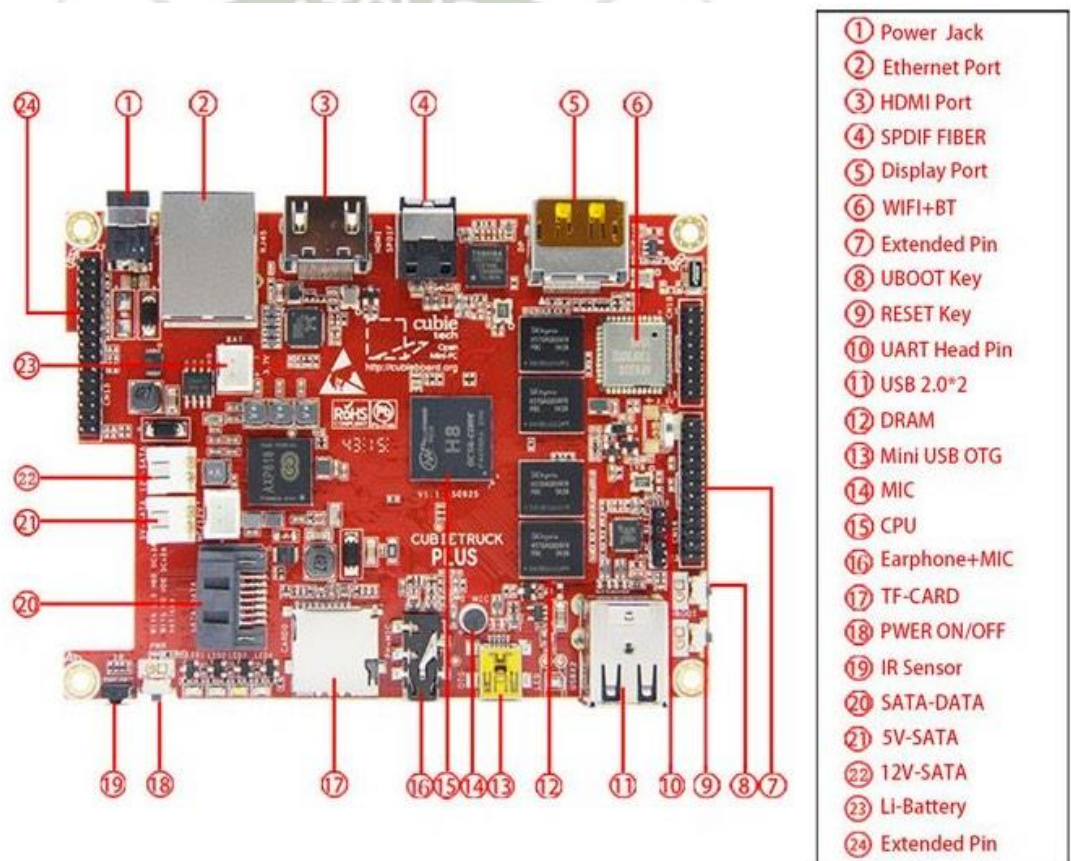


Figura 8. CubieBoard 5 Octa-Core

Fuente: Geeky Gadgets , 2016

Se desarrolló en el año 2012 en una empresa China, desde esa fecha se internacionalizó la CubieBoard en una licencia Open Source, de la cual se pueden descargar los planos y diseño de la página principal de la placa.

Las características que ofrece la CubieBoard 5 son las siguientes:

- SoC: Allwinner H8
- CPU: ARM Cortex-A7 @ 2 GHz octa-core
- GPU: PowerVR SGX544 @ 700 MHz
- display controller: Toshiba TC358777XBG, supports HDMI 1.4 1080p y DisplayPort, no soporta LVDS
- 2 GiB DDR3
- 8 GB EMMC flash built-in, 1x microSD slot, 1 puerto x SATA 2.0 (Hard Disk of 2,5") via USB bridge.
- 10/100/1000M RJ45 Gigabit Ethernet
- 2x USB Host, 1x USB OTG, 1x CIR.
- S/PDIF, auriculares, salida de audio HDMI, micrófono cableado de 3.5mm jack, y micrófono incorporado.
- Wi-Fi (dual-radio 2.4 and 5 GHz) y Bluetooth incorporado con PCB antenna
- 70 pines incluyendo I²C, SPI
- Dimensiones: 11 cm × 8 cm

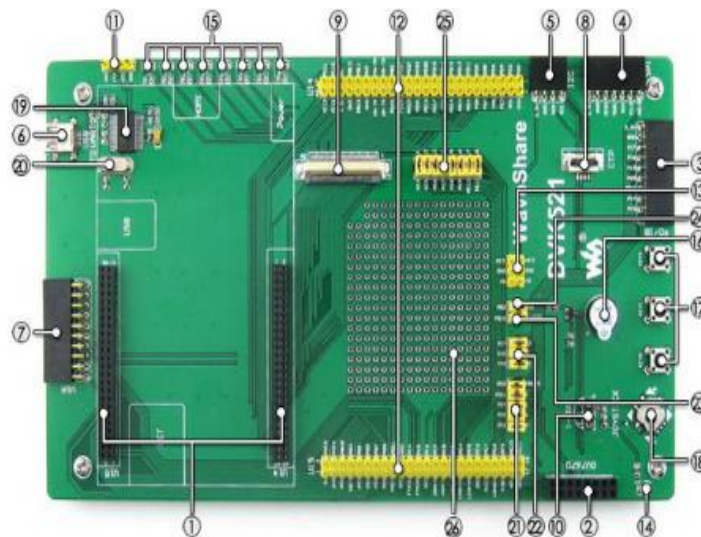
Actualmente, tiene como competidor directo a la Raspberry Pi, aquí se muestra una pequeña tabla de comparación entre ambos (Ver Tabla 2):

Tabla 2. Comparación entre CubieBoard y Raspberry Pi

CubieBoard	Raspberry Pi
Precio: 46 – 60\$	Precio: 35 – 45\$ /Raspberry Pi Zero: 5\$
Capacidad: Discos Duros, USB, SD, etc.	Capacidad: SD, USB, etc.
Placa de mayor tamaño	Placa Pequeña
Jala mayor electricidad	Es ahorrativa en cuestión de gasto de energía
Viene con botón de encendido y apagado	No viene con botones externos
Resiste sistemas operativos como Ubuntu	Resiste sistemas operativos que son derivaciones pequeñas de Debian/Ubuntu y Windows 10

Fuente: Elaboración propia

Como se puede apreciar, ambas placas son muy útiles para el desarrollo de proyectos; aunque la Raspberry primeramente se desarrolló para el aprendizaje de la tecnología y computación, la CubieBoard se desarrolló para ser utilizada como computador, por lo que su desarrollo compensa hasta ahora el objetivo. Está en constante desarrollo y a futuro se verán nuevos cambios y mejoras en la CubieBoard, aunque la venta del dispositivo original es algo costosa; estaría totalmente capacitada para este proyecto, con una excepción. La CubieBoard no viene con una entrada de módulo de cámara propio, para esto se desarrolló lo que se denomina Expander Boards, que se observa en la siguiente imagen (Ver Figura 9):



Specs:

1. Cubieboard socket
2. OV7670 interface: for connecting OV7670 Camera Module
3. 8I/Os interface: easily connects to modules controlled by I/Os, such as 8 Push Buttons
4. SPI interface: easily connects to SPI modules such as AT45DBXX Dataflash, etc.
5. I2C interface: easily connects to I2C modules such as PCF8574 Expansion Module, PCF8563 RTC Module, et
6. USB interface: USB TO UART, convenient for debugging
7. VGA interface: for connecting VGA display Module
8. Capacitive touch screen socket: for connecting capacitive touch screen using I2C interface
9. 7inch LCD interface: for connecting 7inch LCD
10. ONE-WIRE interface: easily connects to ONE-WIRE devices (TO-92 package), such as temperature sensor (D
11. UART interface (PL2303TA): connects to the UART interface of Cubieboard
12. Cubieboard Ext Ports
13. 5V/3.3 V power input/output: usually used as power output, also common-grounding with other user board
14. Power indicator
15. User LEDs
16. Buzzer
17. User Keys
18. Joystick: five positions
19. PL2303TA: onboard USB TO UART convertor
20. 12M crystal: for PL2303TA
21. Joystick jumper
22. User Keys jumper
23. ONE-WIRE jumper
24. Buzzer jumper
25. User LEDs jumper
26. Prototyping area: can be used to place user components for experiments

Figura 9. Expander Board y sus partes

Fuente: Cubieboard.org , 2016

La Expander Board como se vio, brinda a la CubieBoard mayores elementos de entrada y salida para este dispositivo con la finalidad de conectar varios periféricos necesarios para proyectos distintos; esto tiene un costo aparte de la placa, pero las posibilidades son variadas, ya que se puede conectar básicamente casi todo lo que se puede conectar a un computador normal: cámaras web, mouse, joysticks, pantallas VGA, touchs screens, etc.

Para el proyecto de esta tesis, se requería varios componentes para trabajar con esta placa; primero adquirir todo el kit de funcionamiento de la placa, la Expander Board y una cámara web para su funcionamiento, lo cual se tendría un gasto mucho mayor del esperado:

- Gasto 1: Comprar 1 CubieBoard y el kit de la placa
- Gasto 2: Comprar una Expander Board para que se le pueda añadir nuevos periféricos necesarios del proyecto
- Gasto 3: Comprar una Cámara Web, ya que se requiere de una cámara para que capture las imágenes

Estos gastos son elevados y lo que se desea en este proyecto es minimizar los gastos y ofrecer un proyecto útil y eficiente, sin la necesidad de hacer gastos grandes a la empresa, ya que, si se desea implementar a futuro, se tendría que repetir esos 3 gastos la cantidad de veces que la empresa considere necesario para mejorar la atención de pacientes hospitalizados (Doutel Fernando, 2013-2016).

3) Arduino

Arduino es un controlador en el cual se pueden conectar diversos sensores, luces, botones o incluso mensajes de Twitter; es una placa open source la cual se puede programar dándole indicaciones a los sensores y periféricos que se conecten que hagan lo que el usuario desee. Principalmente, está desarrollado con el fin de generar la creatividad de las personas para desarrollar proyectos que contribuyan a la sociedad sin tener problemas con las licencias, además de ser sencillo para el aprendizaje de las personas.

Actualmente, existen varios otros microcontroladores como Parallax Basic Stamp, Netmedía's BX-24, Phidgets, MIT's Handyboard, etc. Son más complejos a la hora de programar los sensores, Arduino simplifica esto reconociendo de manera automática el sensor que se conecta, y solo mediante la programación indica qué componente se desea utilizar y qué debe hacer sin tener que instalar nada más que la plataforma de programación de Arduino (Ver Figura 10).

Algunas de las características que ofrece Arduino son:

- **Económica:** Su costo no sobrepasa los \$50, que en comparación con las otras mencionadas previamente (Parallax Basic Stamp, Netmedía's BX-24, etc) tienen un costo más elevado.
- **Multiplataforma:** Su software corre en cualquier sistema operativo
- **Fácil comprensión IDE:** Su plataforma de programación Arduino(IDE) es sencilla, nada compleja, directo a la pantalla de programación, muy parecida a un bloc de notas.
- **Open Source y extensible:** Es libre, puede ser modificada y mejorada por los usuarios, no tienen ningún costo la descarga.

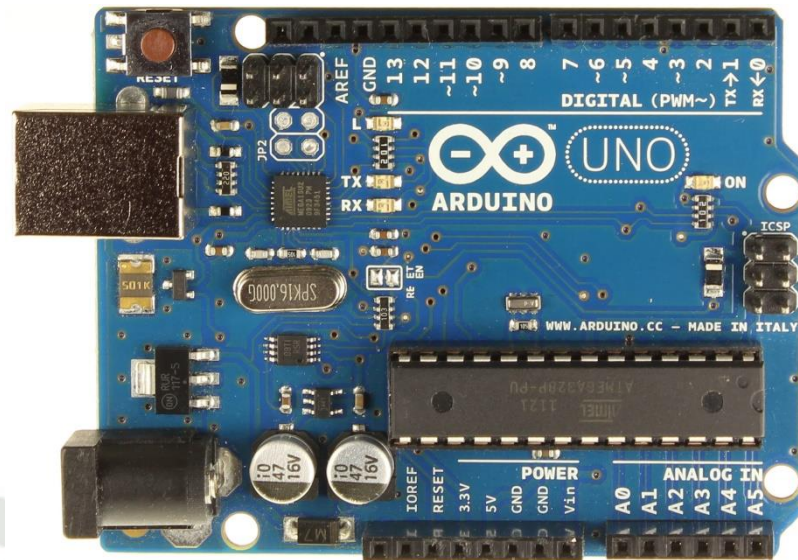


Figura 10. Arduino Uno

Fuente: Arduino.org , 2013

Algunas de las desventajas que ofrece el Arduino son:

- **Baja eficiencia:** Ya que algunas de las funciones implementadas en las librerías, no son las mejores alternativas para las tareas que deberían cumplir; esto en muchos casos, amerita que el usuario cree sus propios métodos para que funcione como se espera.
- **Estructura:** Algunas placas de Arduino tiene un tamaño más grande en comparación a otras, lo que dificulta un poco al crear proyectos de pequeña escala en cuanto a medida, ya que ocupa un espacio algo grande.

El Arduino puede incluir un módulo de cámara para registrar capturas y videos, ofrece hasta 5 MP de imagen (Ver Figura 11).



Figura 11. ESP8266 Camera para Arduino

Fuente: Arducam.com , 2016

Con estos elementos y el costo no tan elevado, es posible que este hardware sea útil para el proyecto, sin embargo, la programación que ofrece Arduino no es lo suficientemente completa para hacer uso de software de reconocimiento de patrones, ya que se buscaba hacer uso de Frameworks que agilicen el trabajo, y en Arduino, no se puede hacer uso de Frameworks de reconocimiento de patrones por el momento, se tendría que implementar numerosas líneas de código y es posible que no se llegue a implementar como se busca en este proyecto, sino que tales se reducirían algunos requerimientos e incluso facilitar el trabajo, lo cual no se busca en este proyecto, al contrario, tal vez añadir más requerimientos dependiendo el avance del proyecto (Arduino, 2016).

4) Raspberry Pi

Según la definición de su página oficial, es un dispositivo de bajo costo. Tarjeta de computadora pequeña que se puede conectar a un monitor de computadora o TV; además de utilizar un teclado, mouse e incluso conexión a un cable de red. Este dispositivo ofrece al usuario la capacidad de interactuar como si fuese una computadora real, pero en versión pequeña y portable.

Principalmente, se creó con la finalidad de brindar mayor educación a los niños acerca de informática y tecnología, ya que su diseño pequeño y fácil de usar, era interesante para los niños, lo que generaba curiosidad y muchas ganas de aprender de ello (Ver Figura 12).

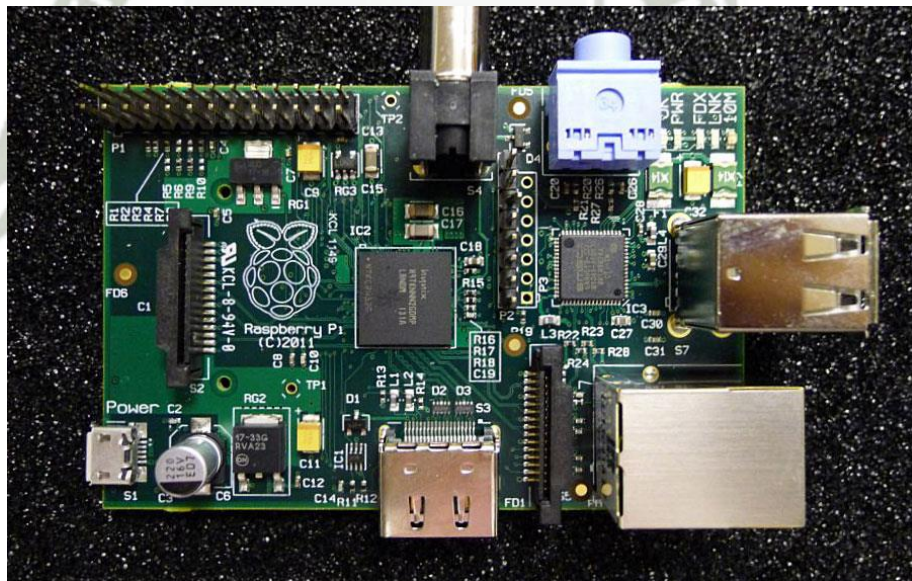


Figura 12. Tarjeta Raspberry Pi Beta

Fuente: RaspberryPi.org, 2012

La página oficial brinda un sistema operativo libre para el hardware el cual es denominado Raspbian, es una variación de Debian, Linux ARM; también se ofrece otros tipos de sistemas operativos como son Ubuntu Mate, Snappy

Ubuntu Core o incluso un paquete de Windows 10 IOT Core. Sin embargo, los más recomendables son NOOBS o Raspbian.

En esta ocasión para el proyecto, se hace uso de Raspbian.

Otro elemento del que se hace uso y vale la pena hablar de él, es el módulo de cámara megapíxel, en si el módulo de cámara megapíxel es una cámara que se conecta directamente con la tarjeta mediante un puerto CSI localizado al costado del puerto Ethernet. Este dispositivo nos permite integrar al Raspberry una cámara de 5 megapíxeles de foto y 1 Mp de video (Raspberry, 2016) (Ver Figura 13).

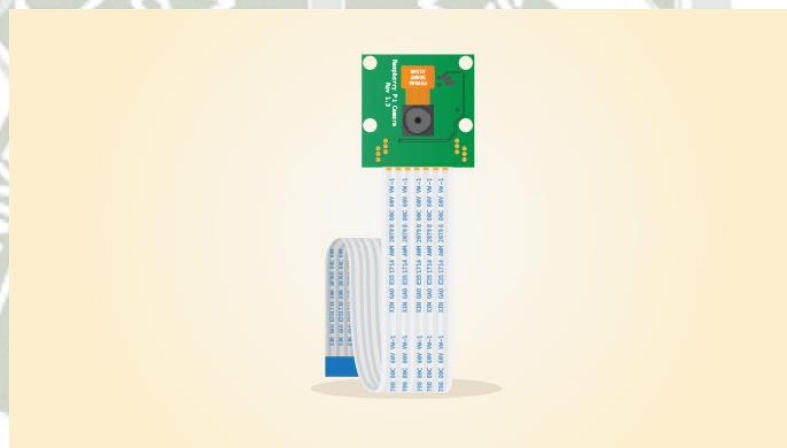


Figura 13. Módulo de Cámara Megapíxel

Fuente: RaspberryPi.org, 2012

Cada Raspberry Pi viene con un conector de módulo de cámara megapíxel el cual es parte del hardware directamente de la Raspberry Pi, evitando así hacer uso de otro dispositivo como una cámara web o celular para capturar imágenes; claro que también es posible, como se mencionó anteriormente, escatimar gastos utilizando el módulo de cámara megapíxel, ya que no es para nada una mala cámara, por el contrario su capacidad y captura de imagen es la adecuada.

Se puede añadir luminosidad en caso de que se quiera hacer uso de ella en la noche, se puede instalar un pequeño led o hasta 4 a sus costados en los 4 espacios alrededor para tener una visión iluminada durante la noche. También conectar un infrarrojo para que la cámara pueda obtener una imagen clara como si fuera de día (Raspberry Pi.org, 2014) (Ver Figura 14).

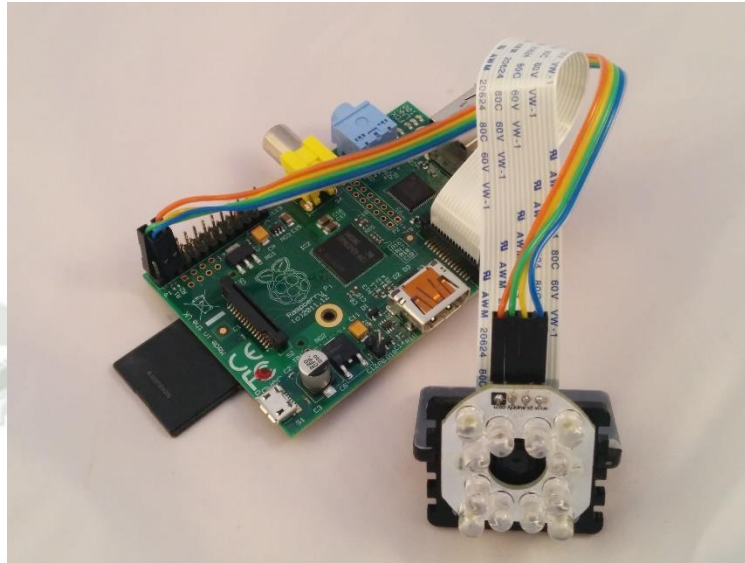


Figura 14. Módulo de Cámara Megapíxel

Fuente: Pi-Supply.org, 2012

En comparación con otras placas de funciones parecidas, puede que no sea la mejor en algunos factores; pero esto dependerá del uso que se le dé. Principalmente para el proyecto, se considera que esta es la mejor placa, ya que debido a su poco costo e inclusión de una cámara más económica y de buena calidad, un sistema operativo derivado de Debian/Linux y con soporte a programación en Python, Java, etc.

2.2.5. Django

Django es un framework de Python para desarrollo web el cual es gratis y la documentación está abierta al público.

Django tiene archivos .py por defecto los cuales ayudan al funcionamiento del sistema según el MVC(Modelo Vista Controlador). Tal como se menciona, este sistema en los archivos .py que se originan al momento de crear una app, se puede diferenciar las partes del programa; se tiene la parte de models, views y los templates.

1) Models

En los Models, se establece la conexión y representación de la base de datos. Django, primeramente, crea una base de datos estándar en Sqlite; pero para este proyecto se cambió la configuración para que trabaje con Postgresql. Django brinda la facilidad de trabajar directamente con la base de datos mediante los models, ejecutando comandos de Django se puede sincronizar cambios entre la base de datos y las sentencias de los models de Django:

- MakeMigrations <nombre de la app>: Reconoce los cambios hechos en el Model.py creando un archivo .py donde reconoce esos cambios, ejecutándolo para alterar la base de datos; este comando solo genera el archivo de migración
- Migrate: Ejecuta el último archivo de migración creado para completar la modificación

2) Views

Los views actúan como controladores, que son el intermedio entre el Template y los models, interactúa con los models para trabajar los datos, y mostrarlos en los templates

3) Templates

Los templates son los archivos html que son la interfaz con el usuario. Django ofrece una manera sencilla de relacionar un template con los controladores que es por medio de las URL, las URL del sistema se encuentran en un archivo denominado url.py, su estructura de cada uno es similar a la siguiente:

url.(<Direccion URL>,<View Relacionado>,<Nombre del Url>)

Este framework se seleccionó por ser la librería de la que más conocimiento se tenía para el desarrollo web, además de utilizar el lenguaje Python para su programación (Django, 2016).

CAPITULO 3: DESCRIPCIÓN DEL SISTEMA Y DESARROLLO

3.1. Recolección y análisis de la información

3.1.1. Definición

Es necesario indagar qué es lo que realmente pasa en cuanto a la atención de los pacientes hospitalizados. Primeramente, se recolectó la información para definir los requerimientos del sistema a desarrollar, esto con la finalidad de tener en claro las funciones que debe cumplir el sistema en conjunto con el programa de reconocimiento de señales para mejorar la atención a los pacientes, además del rendimiento y cuidado del personal.

3.1.2. Método de recolección de información

Para recolectar los datos, se aplicaron encuestas al personal de hospitalización para que nos brinden información sobre la atención actual a los pacientes hospitalizados. Con el debido permiso de las autoridades del hospital, se tuvo acceso al Hospital Edmundo Escomel de Paucarpata, cuya población aproximada es de 33 doctores y 50 enfermeras.

Planeamiento de las Encuestas

Para empezar, se elaboró una encuesta para saber cómo está actualmente la atención a los pacientes y extraer de él algunos requerimientos para el sistema.

La encuesta generada y aplicada implicó las siguientes preguntas:

1. ¿Cómo solicitan atención los pacientes?
2. ¿Qué tan efectivo considera este método?

3. ¿Existen quejas de los pacientes internados con respecto a la tardía atención del personal?
4. ¿Existen pacientes que no pueden solicitar atención de la misma manera?
5. En caso un paciente no pueda solicitar atención de la misma manera y no se encuentren familiares presentes, ¿cómo solicitan ayuda?
6. ¿Considera que se mantiene en constante vigilancia a los pacientes internos?
7. ¿Ha habido problemas por algún descuido a los pacientes internos?
8. ¿En qué momento del día considera que es más fácil atender a los pacientes internos?
9. Durante la noche, ¿los pacientes internos pueden solicitar ayuda al personal de la misma manera que durante el día?
10. Si la respuesta anterior fuese no, ¿cómo hacen los pacientes para solicitar ayuda en la noche?

Se mencionó que actualmente están haciendo uso de botoneras, las cuales funcionan de la siguiente manera:

Un paciente presiona el botón ubicado al costado de su cama y en el pasillo se emite una luz que está afuera de la habitación (dentro de una habitación hay hasta 4 camas dependiendo del espacio); entonces, la enfermera puede atender al paciente al ver la luz encenderse, se acerca a la habitación y pregunta qué paciente hizo el llamado para ver qué necesita. Sin embargo, aproximadamente el 50% de las botoneras no funciona adecuadamente, el paciente tiene que presionar más fuerte o simplemente no enciende la luz. También, se menciona que muchas veces, el personal no atiende adecuadamente a los pacientes y surge el malestar por esta razón u otras veces no es culpa del personal; sino de los

pacientes imprudentes o irrespetuosos. Las encuestas que se aplicaron al personal médico se verán en el siguiente punto.

Procesamiento de las encuestas

El tamaño de la muestra encuestada al final fue de 10 médicos y 10 enfermeras del área de hospitalización, la cual se divide a su vez en Pediatría, Cirugía, Hospitalización y Neonato.

Las respuestas se representan con sus conclusiones:

➤ **Pregunta Nro. 1:**

Respuesta de los médicos:



Figura 15. Gráfico de Respuestas de Médicos, Pregunta 1

Fuente: Elaboración propia

Respuesta de las enfermeras:



Figura 16. Gráfico de Respuestas de Enfermeras, Pregunta 1

Fuente: Elaboración propia

- Al indagar un poco en el tema de atención, según los médicos, algunas botoneras funcionan y otras no; estas botoneras cumplen la función de encender un foquito en el pasadizo con un timbre, así el personal puede estar atento a las necesidades del paciente. Las enfermeras son las que están más al cuidado de los pacientes por lo que su respuesta es considerada como la más acertada. En conclusión, la mayoría de las botoneras funcionan y es el método que utilizan los pacientes, en su mayoría.

➤ Pregunta Nro. 2:

Respuesta de los médicos



Figura 17. Gráfico de Respuestas de Médicos, Pregunta 2

Fuente: Elaboración propia

Respuesta de las enfermeras:



Figura 18. Gráfico de Respuestas de Enfermeras, Pregunta 2

Fuente: Elaboración propia

La escala de calificación de este método es de 1 a 6, donde 1 es “malo” y 6 es “efectivo”. El método de “llamado” es considerado por encima del promedio como efectivo. Como se vio en la anterior pregunta, los médicos en su mayoría contestaron que usaban “llamados” los pacientes para solicitar ayuda; por eso le dan una calificación que está casi justo al 50%. Por el contrario, las enfermeras dieron una calificación por encima del 50% ya que ellas mencionaron que en su mayoría el método que se utilizaba era de las botoneras, al parecer es de gran ayuda.

➤ Pregunta Nro. 3:

Respuesta de los médicos:



Figura 19. Gráfico de Respuestas de Médicos, Pregunta 3

Fuente: Elaboración propia

Respuesta de las enfermeras:



Figura 20. Gráfico de Respuestas de Enfermeras, Pregunta 3

Fuente: Elaboración propia

Al parecer, ambas partes están muy de acuerdo con las quejas de los pacientes por la tardía atención. Esto hace preguntar el porqué de una tardía atención, para lo cual se planteó otra pregunta: ¿Considera que el personal actual es suficiente para atender a los pacientes hospitalizados?, el 100% de ambas partes contestaron que no es suficiente. Actualmente, en el Hospital I, Edmundo Escomel Paucarpata, se cuenta con 3 ambientes de hospitalización ubicados en el tercer piso; estos son de hospitalización general, pediatría y neonato. En el área de pediatría, conversando con una de las técnicas en enfermería, mencionó que solo ella se encarga de cuidar aproximadamente 13 camas con pacientes, aunque está permitido actualmente el ingreso de los padres de familia para el cuidado de sus hijos y adultos mayores

➤ Pregunta Nro. 4:

Respuesta de los médicos:



Figura 21. Gráfico de Respuestas de Médicos, Pregunta 4

Fuente: Elaboración propia

Respuesta de las enfermeras:



Figura 22. Gráfico de Respuestas de Enfermeras, Pregunta 4

Fuente: Elaboración propia

Aquí solo se comprueba que en un 60-64% de los pacientes no pueden solicitar el llamado de la manera actual.

➤ Pregunta Nro. 5:

Respuesta de los médicos:

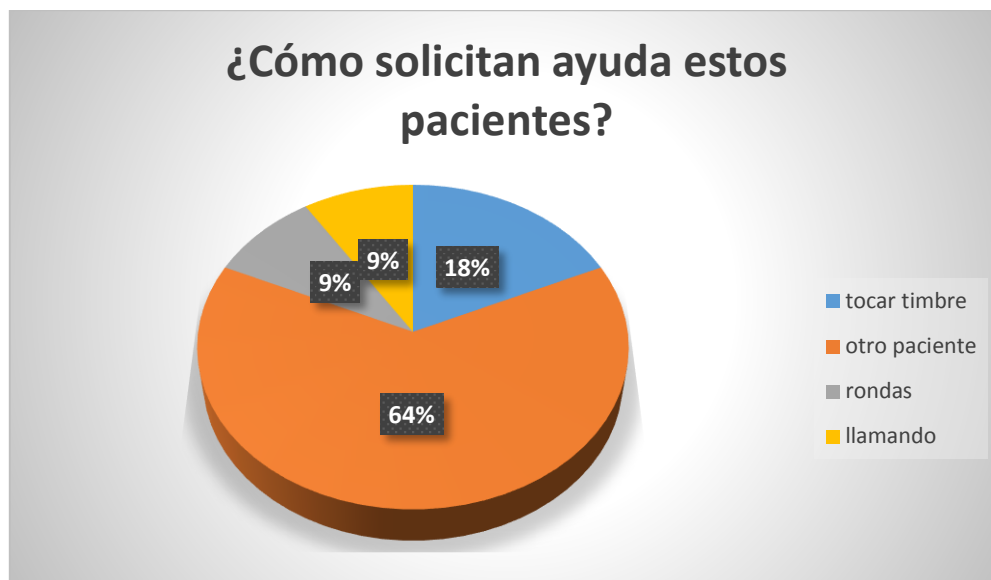


Figura 23. Gráfico de Respuestas de Médicos, Pregunta 5

Fuente: Elaboración propia

Respuesta de las enfermeras:



Figura 24. Gráfico de Respuestas de Enfermeras, Pregunta 5

Fuente: Elaboración propia

Comparando las respuestas, hay un 4% de variación. Se puede definir que sí existen pacientes que no pueden pedir atención con la botonera y necesitan decirle a su compañero(a) que lo haga por él y esto está comprobado con las respuestas de esta pregunta como se ven en los gráficos; indagando el porqué no lo hacen, es porque el paciente no tiene mucha movilidad para alcanzar el botón, sin embargo, este puede mover y levantar ligeramente los brazos, se puede aprovechar este dato para el proyecto

➤ Pregunta Nro. 6:

Respuesta de los médicos:

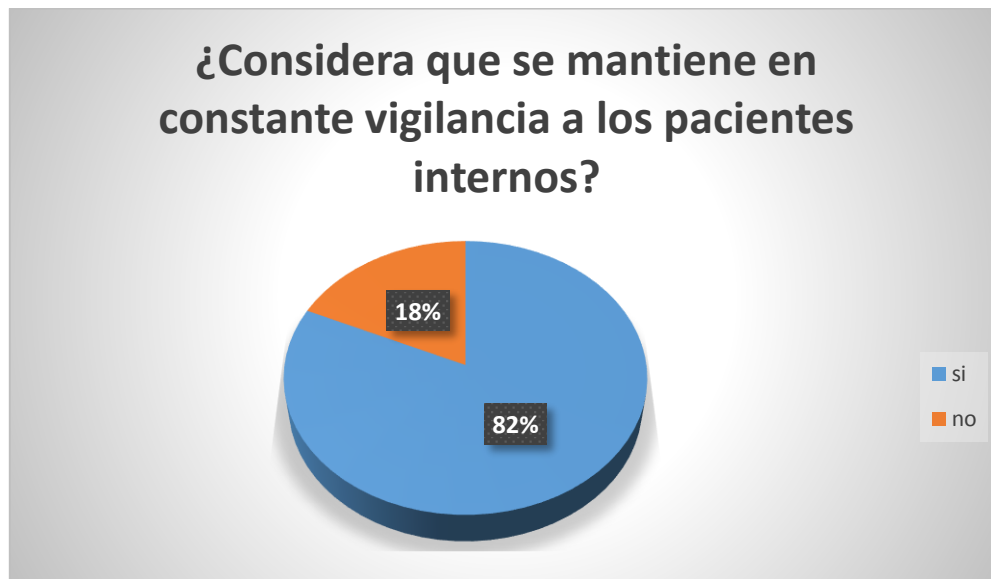


Figura 25. Gráfico de Respuestas de Médicos, Pregunta 6

Fuente: Elaboración propia

Respuesta de las enfermeras:

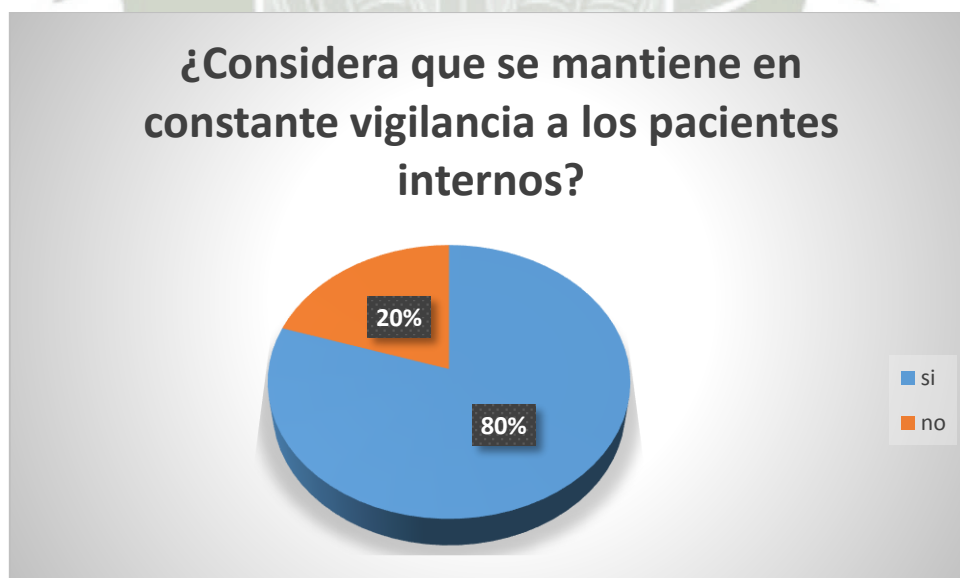


Figura 26. Gráfico de Respuestas de Enfermeras, Pregunta 6

Fuente: Elaboración propia

En un promedio del 81% manifiesta que es la vigilancia a los pacientes.

➤ Pregunta Nro. 7:

Respuesta de los médicos:

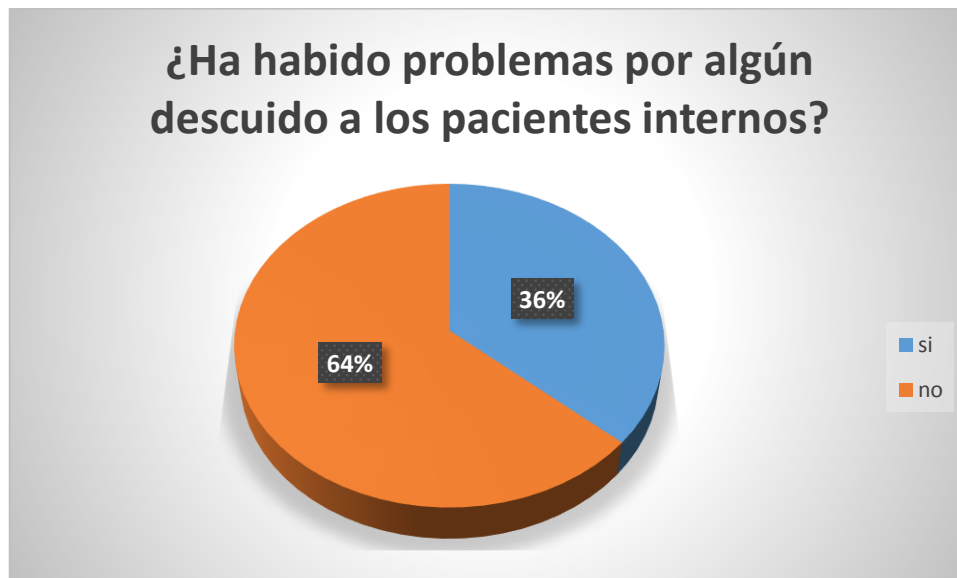


Figura 27. Gráfico de Respuestas de Médicos, Pregunta 7

Fuente: Elaboración propia

Respuesta de las enfermeras:

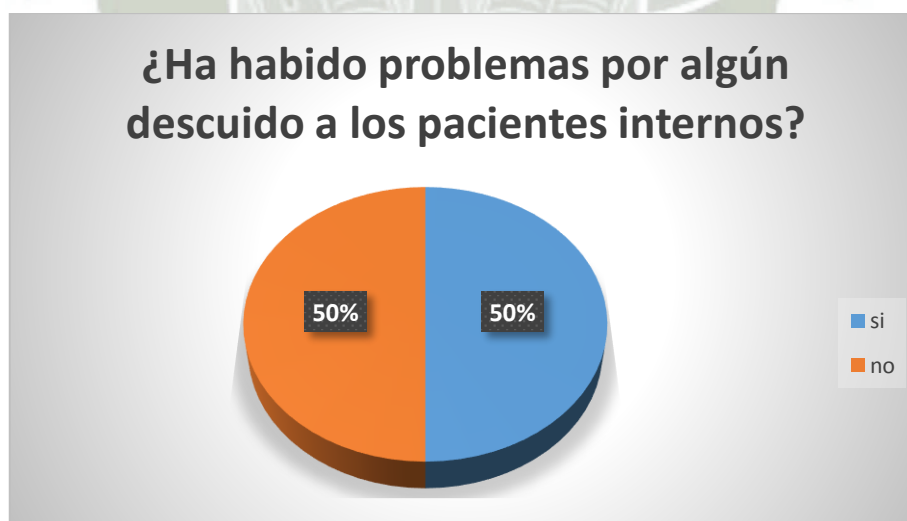


Figura 28. Gráfico de Respuestas de Enfermeras, Pregunta 7

Fuente: Elaboración propia

Un 43% contestó que, sí hubo problemas por descuido a los pacientes; indagando en esta pregunta, se tuvo algunos ejemplos de pacientes que no pueden pedir ayuda por sí solos. Aunque suena un poco contradictorio a la pregunta anterior, ya que se tiene en vigilancia constante a los pacientes según la mayoría del personal. No deberían ocurrir estos problemas.

➤ Pregunta Nro. 8:

Respuesta de los médicos:

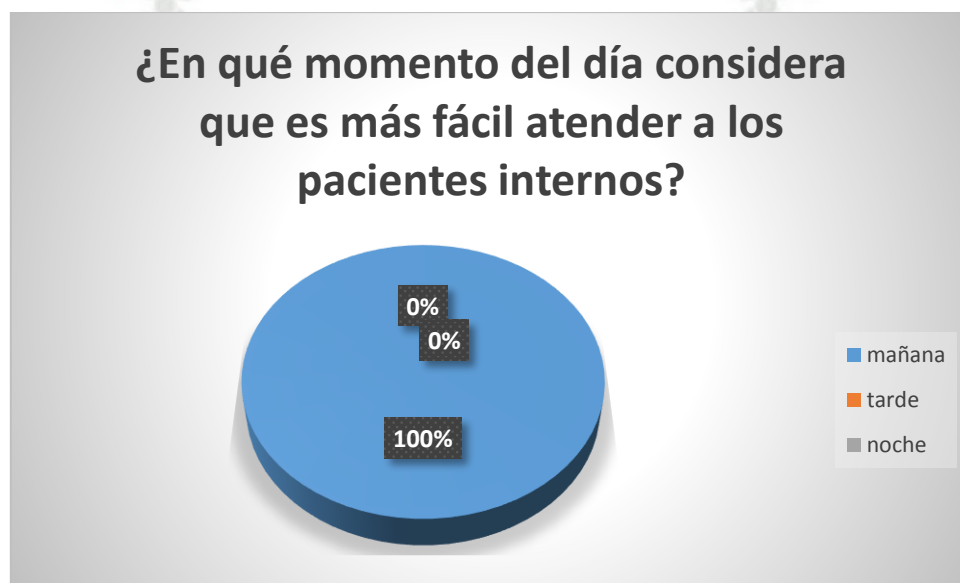


Figura 29. Gráfico de Respuestas de Médicos, Pregunta 8

Fuente: Elaboración propia

Respuesta de las enfermeras:

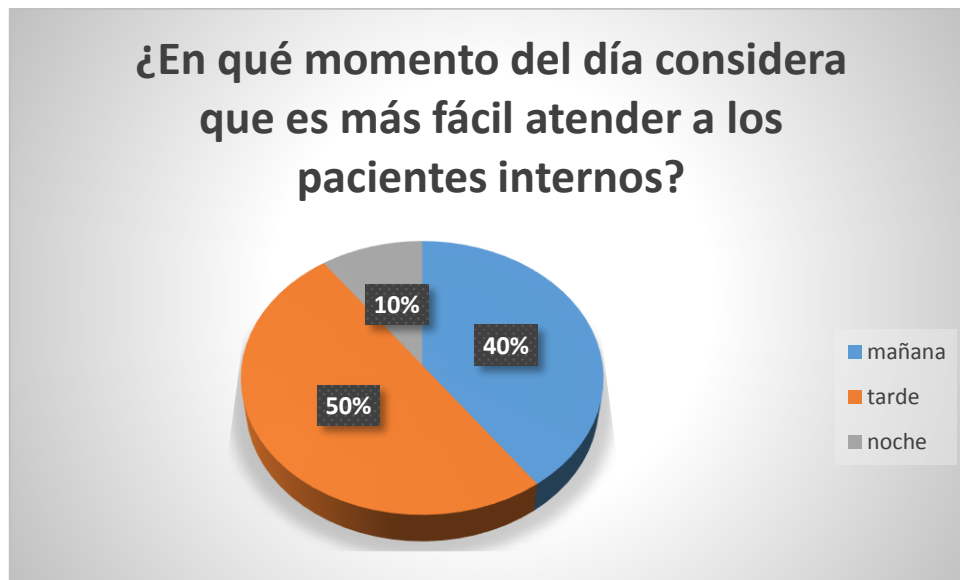


Figura 30. Gráfico de Respuestas de Enfermeras, Pregunta 8

Fuente: Elaboración propia

Los médicos no están al 100% atentos al paciente ya que esa es la labor principal de la enfermera; según la mayoría, es más fácil atenderlos por la tarde ya que tienen mayor cantidad de personal, esto facilita la atención.

➤ Pregunta Nro. 9:

Respuesta de los médicos:

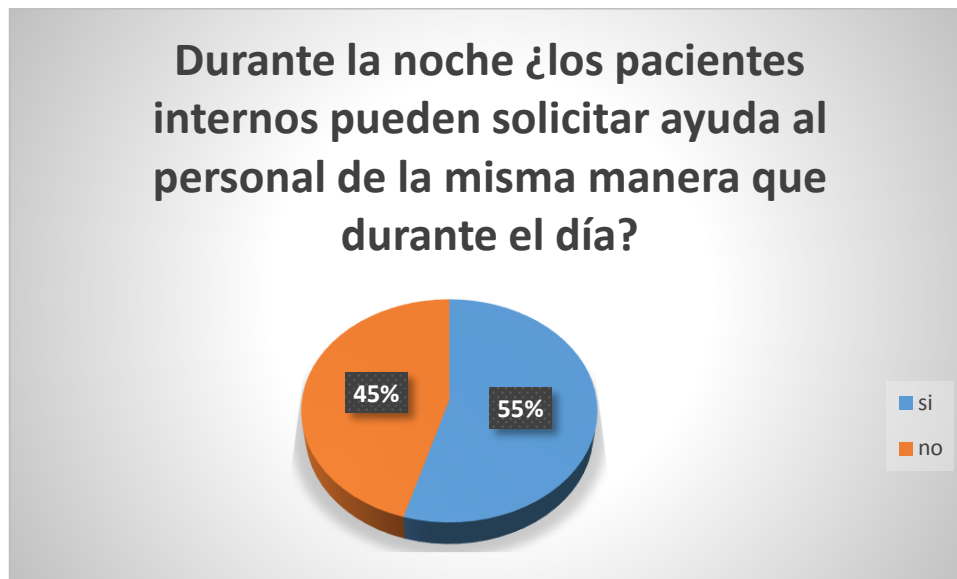


Figura 31. Gráfico de Respuestas de Médicos, Pregunta 9

Fuente: Elaboración propia

Respuesta de las enfermeras:

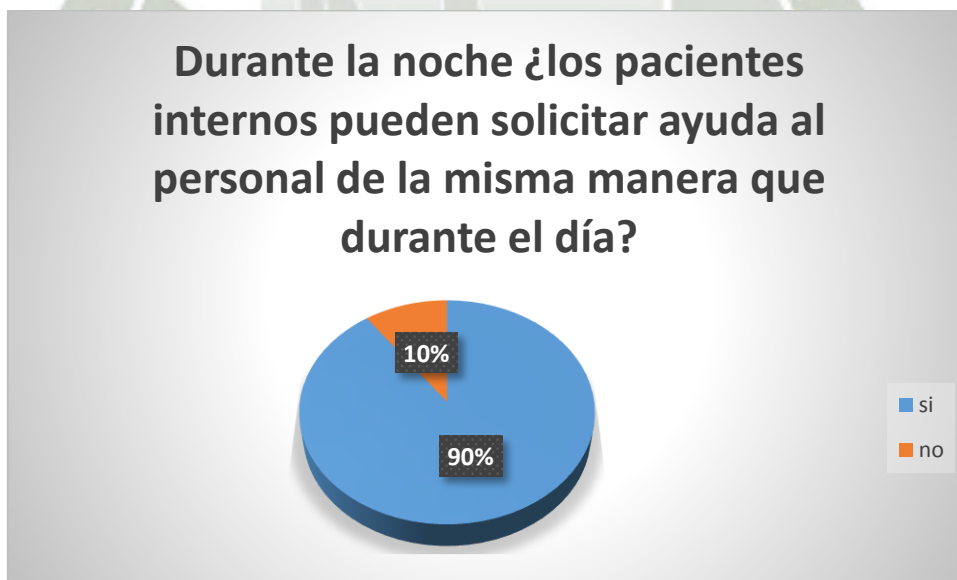


Figura 32. Gráfico de Respuestas de Enfermeras, Pregunta 9

Fuente: Elaboración propia

Se solicita de la misma manera atención de los pacientes durante la noche y el día.

Por lo que el proyecto estará adaptado también para operar de noche a futuro.

Pregunta Nro. 10:

Respuesta de los médicos:

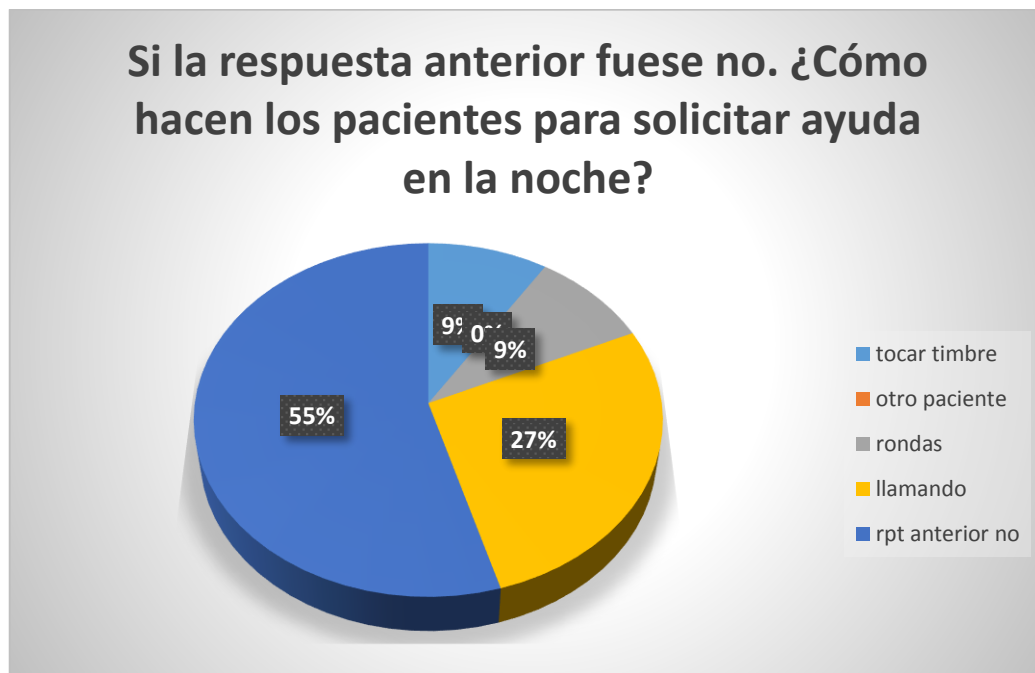


Figura 33. Gráfico de Respuestas de Médicos, Pregunta 10

Fuente: Elaboración propia

Respuesta de las enfermeras:

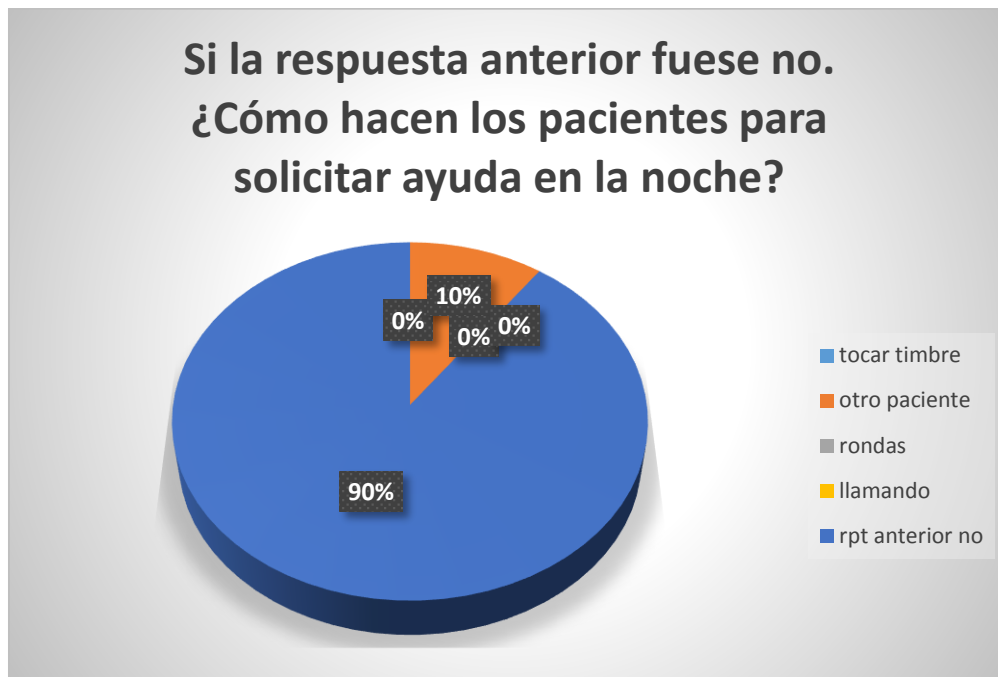


Figura 34. Gráfico de Respuestas de Enfermeras, Pregunta 10

Fuente: Elaboración propia

La mayoría contestó que no se puede llamar al personal con las botoneras (aunque en anteriores respuestas dijeron lo contrario); eso incluye el uso de las botoneras, indagando porqué no podían, la respuesta es porque algunas botoneras no funcionan o que se contradice con la primera pregunta, haciendo pensar que tal vez no sea muy efectivo este método.

Visto los resultados de la encuesta, se considera que el proyecto sería un beneficio para la atención, vigilancia y control del personal para con sus pacientes.

Por ahora, se puede planteando algunos requerimientos con esta información:

- Identificar una atención
- Registrar una atención
- Reconocer quién atendió al paciente
- Identificar qué paciente o cama fué la que solicitó la atención

- Alertar una atención
- Registrar una descripción de lo sucedido en la atención

3.2. Requerimientos del Sistema

3.2.1. Requerimientos de software

Sistema Operativo

Para el funcionamiento del Raspberry Pi, es necesario que tenga un sistema operativo funcional en la tarjeta SD. El sistema operativo seleccionado es Raspbian Jessie, software libre diseñado para el control del Raspberry Pi, es una derivación de Linux y viene con programas necesarios para programación, juegos, etc. Entre ellos, Python fue instalado en ambas versiones.

Para el desarrollo del sistema web se requiere Windows o Linux/Ubuntu, ya que la programación de este se puede hacer en ambas plataformas sin ningún problema de compatibilidad.

Python

El sistema completo se desarrolla en este lenguaje, se trabaja con Python 2.7 ya que es considerado el más estable hasta la fecha; además de ser más compatible con los Frameworks a utilizar tanto en el ordenador como en el Raspberry Pi

Django Web Framework

El framework de Django para Python que se instaló una de las últimas versiones; la 1.9.3, a diferencia de las demás, es la más completa, con diversas funciones, incluso más compleja para páginas web de mayor escala a la de este proyecto; al ser compatible con Python 2.7, se considera una versión eficiente para el desarrollo del prototipo

ReportLab

Este framework es utilizado para la generación de reportes a partir de la información capturada por el sistema de cámaras Raspberry Pi; está también asociado a Python y es de reportes para sistemas web; se instaló la última versión la cual ofrece trabajar con gráficos avanzados en pdf.

Postgresql

Es un gestor de Base de datos que fue hecho para la comunidad y estos pueden mejorarlo; es un gestor que actualmente presenta diversas herramientas para conectarse con la mayoría de lenguajes de programación. Contiene un lenguaje propio denominado PL/pgsql. Para el desarrollo de este proyecto, es bastante útil ya que el anterior framework denominado Django permite conectarse a este gestor de manera sencilla.

OpenCV

Como se vio anteriormente, existían más Frameworks capaces de utilizarse en el desarrollo del proyecto; sin embargo, se planteó desde el inicio que el análisis de imágenes, procesamiento y envío de datos lo haga el hardware que será utilizado como cámara que se seleccionó para el proyecto; para ello, estos hardware utilizan en su mayoría sistemas operativos derivados de Linux, por lo que el framework, lenguaje, base de datos y demás herramientas del software que se utilizarán deben ser compatibles con Linux. A continuación, se muestra una tabla de comparación entre los Frameworks vistos anteriormente para basarnos en la elección del más útil para el proyecto (Ver Tabla 3).

Tabla 3. Tabla de comparación de Frameworks de Reconocimiento de Patrones

Comparación Frameworks	IGesture	Aforge.NET	OpenCV
<u>Interfaces para diversos lenguajes</u>	Solo Java	Solo C#	C++, C#, Java, Python, Android
<u>Documentación</u>	Poca Documentación	Poca Documentación	Documentación Completa
<u>De Código Libre</u>	Libre	Libre	Libre
<u>Actualizaciones</u>	Ultima Actualización 2007	Ultima Actualización 2013	Ultima Actualización 2016
<u>Soporte para Sistemas Operativos</u>	Windows/Linux	Windows	Windows/Linux /Mac OS/Android /IOS

Fuente: Elaboración propia

Como se puede apreciar, el más completo y mejor desarrollado es OpenCV, ya que tiene compatibilidad con distintos sistemas operativos y distintos lenguajes; además de ser el más actualizado.

Se requerirá la instalación del OpenCV en el Raspberry Pi (se verá más adelante por qué fue seleccionado), ya que esta se encargará de analizar las imágenes y hacer uso de los algoritmos de reconocimiento de patrones. Por lo que la instalación se debe hacer dentro del entorno Raspbian, en este caso se hace uso de la versión 3.0.0 ya que es la última versión desarrollada

Como se mencionó anteriormente, existen varias herramientas que se pueden utilizar; pero Python es el lenguaje acorde a lo que vamos a plantear. Se decidió por OpenCV, el cual nos brinda varios algoritmos de reconocimiento de patrones implementados, para darnos la facilidad de implementar el proyecto en menos tiempo. Primeramente, se verán algunas de las funciones que se utilizaron para el proyecto y, posteriormente, cómo funcionan los algoritmos que nos brinda esta librería.

Para ello, se simplifica el desarrollo y explicación en las partes que son necesarias para el desarrollo del proyecto:

- Preprocesamiento de imágenes

OpenCV ofrece funciones que están desarrolladas de acuerdo con algoritmos diseñados para el procesamiento de imágenes, algunas de las funciones que ofrece son las siguientes:

Imagen y Video

OpenCV trabaja exclusivamente con archivos de imagen y video, procesando su información como si fuera un editor de imágenes; pero con código. Se puede realizar tareas básicas como leer un archivo, mostrarlo por pantalla, definir una tecla de pausa a la ejecución del script, cerrar las ventanas con una tecla, etc. Primeramente, con imágenes, el resultado es la lectura; guardado y mostrar no hay problemas; el problema surge en el video.

OpenCV ofrece la oportunidad de tanto leer videos ya hechos, como la posibilidad de interactuar con un video que está siendo transmitido a través de una cámara que tengamos en el computador como una webcam, por ejemplo.

Para esto, se crea un objeto VideoCapture; este objeto nos permite usar un hardware que es utilizado como cámara para el script, enviando simplemente un parámetro que nos dirá qué número de cámara se puede utilizar. En pocas palabras, si se tiene solo una cámara conectada, se le enviará como parámetro 0; caso contrario, este número variará dependiendo del orden.

Lo complicado es al momento de guardar un video, ya que se tiene que especificar el nombre de salida y el FourCC, el cual es la extensión que tendrá el video, puede ser .avi .mp4, etc. Otro parámetro que recibe es el IsColor, el cual especifica si es una calidad de color o escala de grises. FourCC es un código de

4 bits, que especifica el códec del video. Existen codecs para los diversos SO que son más viables y no ocasionan conflicto al momento de guardar. En el caso de Windows, el más factible es el DIVX, y en Linux Fedora, contiene la gran mayoría de codecs sin problemas.

Color

OpenCV trabaja con los tipos de colores BGR, HSV y GrayScale; estos tipos de colores son utilizados para mejorar la calidad de imagen de los archivos media con los que se interactuará y para hacer uso de otras herramientas que tiene OpenCV. Mayormente, es utilizado el de GrayScale ya que las futuras herramientas trabajan de mejor manera a esa escala de colores. La función que permite cambiar este modelo de colores es `cvtColor`, el cual recibe como parámetros el archivo en una variable y el tipo de escala de conversión; es decir, de BGR a HSV, de BGR a Grises, etc.

Gracias a esta herramienta se puede llegar a diferenciar los colores y hacerle entender al programa qué color es azul, qué color rojo; en pocas palabras, diferenciar cada color en una imagen o video (Ver Figura 35).



Figura 35. Prueba de detección de colores

Fuente: OpenCV-Python-Tutorials, 2011

Nitidez

La nitidez es usada mayormente para tener una mejor perspectiva en la imagen, que los pixeles se adapten a sus vecinos para aclarar la imagen y resaltar algunos colores. OpenCV también hace uso de esto, y es que con la función `cv2.Thresholding`, esta función recibe como parámetros la imagen, el valor de desenfoque que se desea usar, y el tercer argumento será el máximo valor que se desea obtener con la nitidez. OpenCV nos ofrece variables ya definidos para diferentes estilos de nitidez, incluso una nitidez adaptativa que utiliza algoritmos de gaussian, además de ofrecer el `thresholding` de Otsu's el cual calcula automáticamente un valor para la nitidez; esto nos ayuda a evitar estar probando valores para ver el mejor valor de nitidez en la imagen (Ver Figura 36).

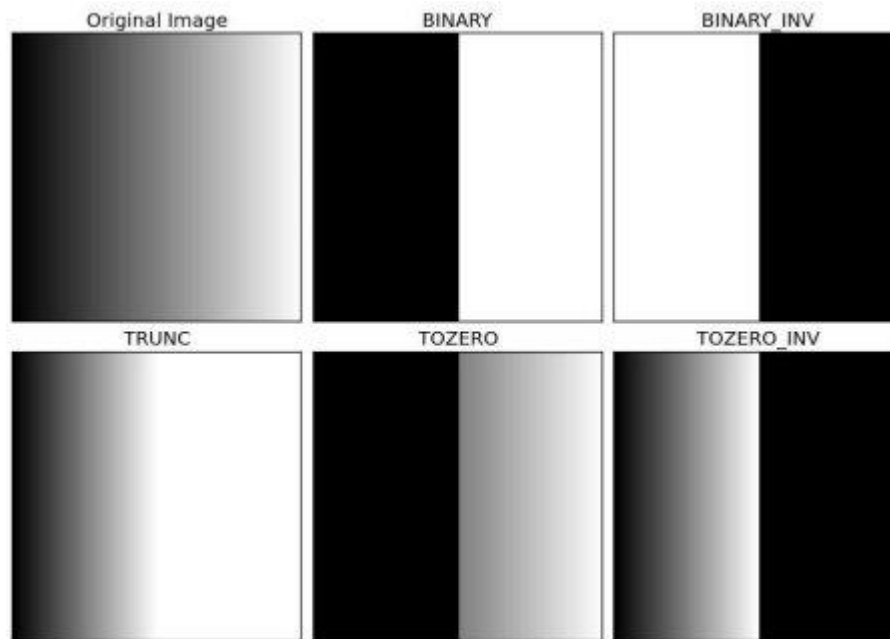


Figura 36. Estilos de desenfoco

Fuente: *OpenCV-Python-Tutorials*

Otsu's funciona de la siguiente manera:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

t = valor de nitidez

Donde:

$$q_1(t) = \sum_{i=1}^t P(i) \quad \& \quad q_2(t) = \sum_{i=t+1}^I P(i)$$

$$\mu_1(t) = \sum_{i=1}^t \frac{iP(i)}{q_1(t)} \quad \& \quad \mu_2(t) = \sum_{i=t+1}^I \frac{iP(i)}{q_2(t)}$$

$$\sigma_1^2(t) = \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \quad \& \quad \sigma_2^2(t) = \sum_{i=t+1}^I [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

P = Valores de los pixeles de la imagen

q1 y q2 son las probabilidades de 2 clases separadas (pixeles de primer plano y

pixeles de fondo) mientras que σ_1^2 y σ_2^2 son variaciones de estas 2 clases, de esta

manera se obtiene el umbral óptimo para aplicar una mejor nitidez. El método de

Otsu calcula el valor umbral ($\sigma_w^2(t)$) de manera que la dispersión dentro de cada

parte sea lo más pequeña posible, pero al mismo tiempo la dispersión sea lo más alta posible entre partes distintas, manteniendo así un equilibrio compartido y eficiente.

Desenfoque

El desenfoque o suavizado de una imagen se realiza para que algunos puntos o píxeles en la imagen sean redefinidos para adaptarse a los que están más a su alrededor. Para esto, OpenCV tiene diversos filtros entre los que se hace uso de LPF(Low Pass Filter) y HPF(High Pass Filter); pero la mayoría de estos necesitan de un kernel de por medio para funcionar. Para esto, se puede definir una matriz con la herramienta Numpy para que sirva de kernel y hacer funcionar el filtro (Ver Figura 37).

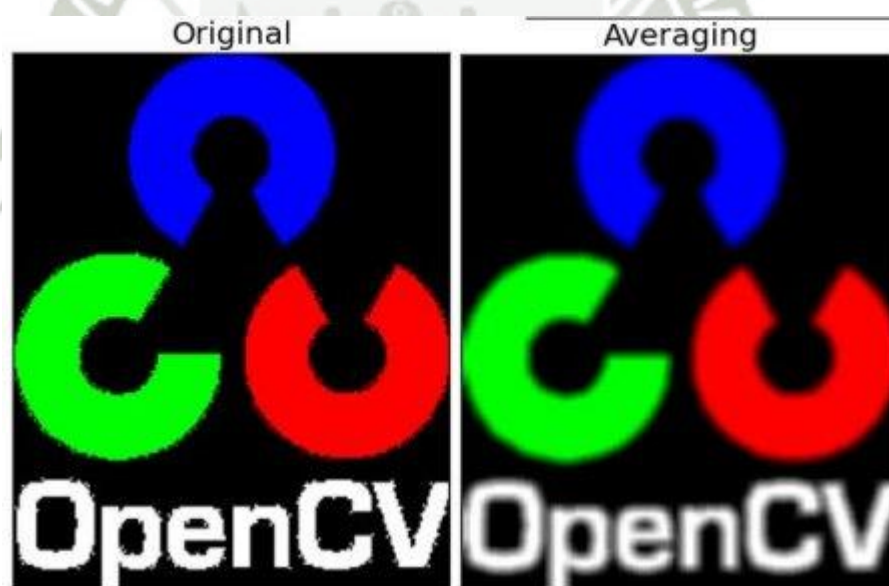


Figura 37. Resultado de uno de los filtros de Nitidez

Fuente: *OpenCV-Python-Tutorials*

Como se puede observar, los métodos que contiene OpenCV requieren una imagen limpia sin ruido y en su mayoría a escala de grises, cosa que ya se vio anteriormente. Existen varios métodos de filtrado como:

- Averaging
- Gaussian Filtered
- Media Filtering
- Bilateral Filtering

Cada uno de estos métodos de filtrado, utiliza kernels ya definidos en el mismo método programado; pero también se puede utilizar otros kernel con la finalidad de obtener el resultado esperado. Más adelante se mostrará cuál se utilizó en el proyecto para dar más suavidad a la imagen obtenida por video.

- Extracción de Contornos

Una vez se tenga una imagen procesada, lo que se debe hacer es indicarle que reconozca nuestra mano; para hacer esto, OpenCV nos brinda una herramienta la cual está basada en el Algoritmo de Suzuki Satoshi (1985).

Contornos

Contornos es un tema amplio en OpenCV; principalmente, nos permite poder diferenciar objetos dentro de una imagen, dibujando o reconociéndolos mediante limitantes o cambios de estructura en cierto punto de la imagen, generando así contornos alrededor de lo que se desea resaltar. A diferencia del Photoshop, este no reconoce la mínima curva; pero el algoritmo utilizado es efectivo para lo que se desarrolla en este proyecto. Primeramente, existen 2 métodos: findContours y drawContours.

La primera función trabaja buscando patrones de colores en conjunto que se diferencien del resto, no es muy complejo; pero para hacer un correcto uso de esta herramienta, es necesario procesar la imagen con thresholding y grises para un mejor análisis, además de que este método modifica la imagen por lo que es necesario almacenar la imagen original en otra variable. Esta función devuelve 3

valores: la imagen original, el modo de recuperación de contornos, y el método de aproximación; la segunda variable que nos retorna es básicamente una lista de los contornos de la imagen que están en un array tipo Numpy.

Para dibujarlos simplemente se usa la otra función `drawContours` la cual recibe como parámetros, la imagen, la lista de contornos, el contorno que se quiere dibujar (por defecto si se coloca -1, dibujara todos los contornos encontrados), los colores y entre otros parámetros que también se pueden enviar para estilo como grosor, tipo de línea, etc. Ya que se envía los contornos, se puede seleccionar alguno de ellos en el 3er parámetro; es decir, quiero dibujar el tercer contorno, pongo el nro. 2(ya que trabaja como cualquier lista Numpy), o también simplemente como segundo parámetro se pone un contorno, es decir, en una variable coloco el valor de un contorno en la lista de contornos y como tercer parámetro en la función coloco 0, ya que solo es un valor y no una lista

```
#Lista
img = cv2.drawContours(img, contours, -1, (0,255,0), 3)
#un contorno
img = cv2.drawContours(img, contours, 3, (0,255,0), 3)
#un valor de la lista de contornos
cnt = contours[4]
img = cv2.drawContours(img, [cnt], 0, (0,255,0), 3)
```

Como se mencionó anteriormente, el segundo y tercer argumento de `findContours`, ¿con qué valores específicamente tengo que llenarlos? La respuesta es que OpenCV ya tiene valores definidos y solo hay que probar entre ellos para escoger el más efectivo.

Ahora como se mencionó anteriormente, OpenCV no puede trabajar al igual que un Photoshop al seleccionar los contornos; sin embargo, mediante unas técnicas y métodos que brinda, esto se puede lograr. Primeramente, tener el concepto claro de lo que es un `image moment` o momento de imagen; el concepto es tener

un peso promedio de la imagen dependiendo la intensidad de pixeles de la imagen. La fórmula para tal proceso (ya definida en OpenCV) es:

$$m_{ji} = \sum_{x,y} (\text{array}(x, y) \cdot x^j \cdot y^i)$$

El array será el conjunto de contornos o unos contornos específicos, que contienen coordenadas X y Y, realizando una sumatoria para encontrar tanto el área dentro de los contornos como los límites.

Para extraer los momentos de un contorno y hallar un valor promedio de este se tiene que seleccionar un contorno del array, con esto se obtendrá data importante como área, centroide, etc. Un ejemplo para hallar el centroide en puntos X y Y es el siguiente en código:

```
cnt = contours[0]
M = cv2.moments(cnt)
cx = int(M['m10']/M['m00'])
cy = int(M['m01']/M['m00'])
```

Donde:

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

Siendo X y Y los centroides a hallar.

Se tiene otras funciones también como el área del contorno(contourArea), el perímetro (arcLength, tiene un segundo argumento, si el contorno es cerrado, poner True, de lo contrario False), Contour Approximation(utiliza el algoritmo de Dougle Peucker, utilizado para reducir el nro de puntos en una curva y dar forma), etc.

El que se usó Principalmente, en la aplicación es ConvexHull ya que, según la documentación, la cual incluso nos muestra una imagen de la mayor utilidad de este método, sirve en sí para analizar una curva de defectos convexos y corregirlos; básicamente, una curva convexa es aquella que se abulta en el

interior o es menos plana, este método corrige eso y lo vuelve más circular y agradable a la vista.

```
hull = cv2.convexHull(points[, hull[, clockwise[, returnPoints]]]
```

Los parámetros son los siguientes:

Points: los contornos

Hull: la salida de la función, normalmente el contorno mejorado

Clockwise: la orientación según las manillas del reloj, True si es normal, False anti horario

Return Points: True retorna las coordenadas de los puntos; False, los índices, simplemente para definir con qué tipos de dato trabajar.

La última variable se puede entender mejor si se dice que se tiene 5 defectos encontrados en un contorno, si le pongo True me retornará las coordenadas:

[125,112],[0,15]... y si se pone True retornará un índice, es decir: [10][15]

[105], y básicamente el valor será el mismo: [105] = [125,112]

Existen también propiedades de Contours las cuales son:

- Aspect Ratio
- Extent
- Solidity
- Equivalent Diameter
- Orientation
- Mask and Pixels Points
- Max Value, Min Value
- Mean Color, Mean Intensity
- Extreme Points

Algunas funciones que tiene también son:

- Point Polygon Test

- March Shapes
- Convexity Defects

Este último es importante para el proyecto por la siguiente razón:

Como se ve, en ConvexHull se consideraba cualquier desviación como defecto; pues con este método se puede extraer esos defectos y ubicarlos en los contornos, es decir, crear un punto de ubicación de estos defectos y usarlos como datos. Si se pone un `returnPoints = false` retornará el índice nada más, pues de esta manera trabaja `convexdefects` con los índices (Ver Figura 38):



Figura 38. Imagen resultado del análisis de contornos y defectos.

Fuente: OpenCV-Python-Tutorials

Explicándolo mejor, la función retorna un array que en cada fila contiene los siguientes datos: [start point, end point, farthest point, approximate distance to farthest point]

Ahora, también existe la jerarquía o herencia en los contornos. Como se ve al inicio, el método `findContours` recibía 3 parámetros: la imagen, el modo de recuperación y el método de aproximación. Estas 2 últimas son variables definidas por OpenCV que ayudan justamente a encontrar mejor los contornos; entre las variables de salida se tiene la imagen, los contornos y la tercera es la jerarquía. OpenCV tiene variables principales para jerarquía: `cv2.RETR_LIST`, `cv2.RETR_TREE`, `cv2.RETR_EXTERNAL`, `cv2.RETR_CCOMP`

¿Cómo se da esta jerarquía y a qué se refiere con la jerarquía de contornos? Esta imagen explica mejor ello (Ver Figura 39):

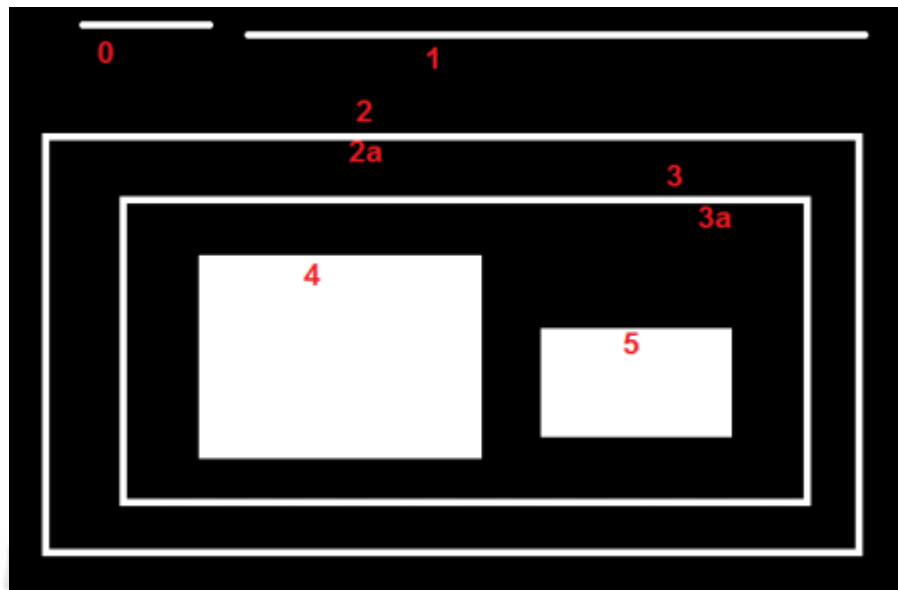


Figura 39. Jerarquía en Contornos de una Imagen

Fuente: OpenCV-Python-Tutorials

Esto nos ayuda a identificar, quién es el contorno padre, hijo, etc.

Como se sabe que nos retorna todos los contornos la función, se puede navegar a través de ellos mediante la jerarquía; poder identificar más rápidamente como si fuera una lista vertical los niveles de los contornos. Los diversos métodos mencionados anteriormente son maneras para crear correctamente esta jerarquía; todos funcionan con arrays Numpy, pero son organizados dentro de manera diferente, depende de cada uno el método de jerarquía que desee usar.

Analizar Contornos (Reconocimiento de la mano)

Ahora entre los algoritmos que brinda se encuentran los siguientes:

- Gaussian Blur (Desenfoque Gaussiano)

Se guía de la función Gaussian la cual genera un filtro de desenfoque en la imagen calculando la transformación que se aplica a cada uno de los pixeles en la imagen. La ecuación varía de acuerdo a dimensiones:

1 dimensión:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

2 dimensiones:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Como podemos interpretar esto, primero tenemos que enfocarnos en la distribución normal, la cual tiene la siguiente gráfica (Ver Figura 40):

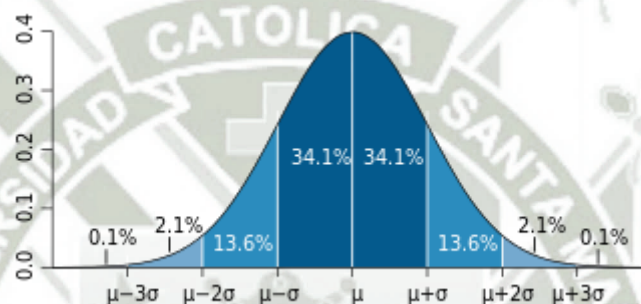


Figura 40. Gráfico de Distribución Normal

Fuente: Recuperado de <http://mathworld.wolfram.com/>

Como se puede observar, el mayor valor se encuentra al centro en un plano bidimensional, y a su alrededor son números menores que van en un descenso continuo y ordenado; si lo vemos en un plano tridimensional, estaríamos observando el resultado de la ecuación de 2 dimensiones (Ver Figura 41.)

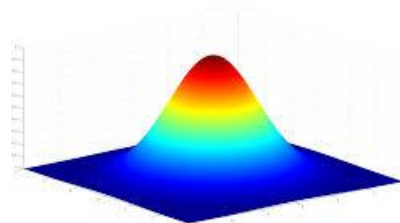


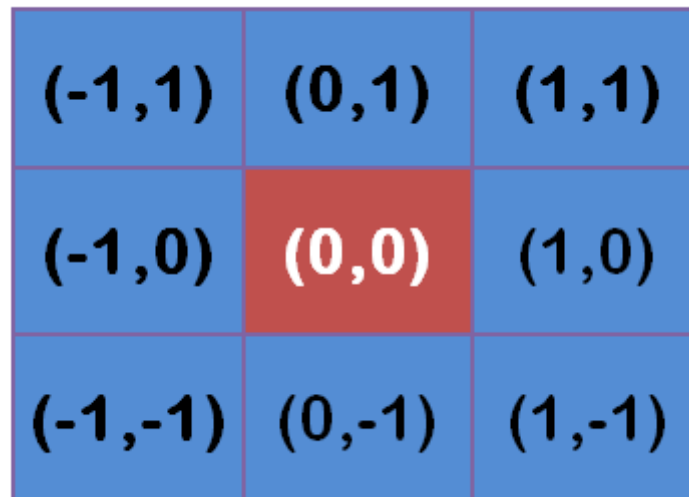
Figura 41. Plano Tridimensional de la Distribución Normal

Fuente: Recuperado de <http://mathworld.wolfram.com/>

En el caso de 2 dimensiones trabajamos con los puntos X Y en una imagen.

Estos puntos nos darán la coordenada del pixel dentro de la imagen.

La segunda parte para entender correctamente este algoritmo es el uso de un kernel, el cual es una matriz de números que nos servirán para calcular la media del valor de pixeles que están alrededor de uno seleccionado. Tomando un ejemplo, digamos que seleccionamos una imagen y aplicamos el algoritmo, creamos un kernel con todos los puntos alrededor del pixel y trabajamos con los valores del kernel (Ver Figura 42).



(-1,1)	(0,1)	(1,1)
(-1,0)	(0,0)	(1,0)
(-1,-1)	(0,-1)	(1,-1)

Figura 42. Kernel de datos

Fuente: Recuperado de <http://mathworld.wolfram.com/>

Teniendo la matriz de puntos queremos que el valor del desenfoque sea en este caso de 1.5, por lo que asignamos ese valor a la fórmula, $\sigma=1.5$ y aplicando la fórmula tenemos el siguiente resultado (Ver Figura 43.):

0.0453542	0.0566406	0.0453542
0.0566406	0.0707355	0.0566406
0.0453542	0.0566406	0.0453542

Figura 43. Kernel de resultados

Fuente: Recuperado de <http://mathworld.wolfram.com/>

Para lo cual calculamos el promedio que en este caso saldría de la suma de todos los puntos divididos entre 9: 0.4787147, así obtenemos la matriz de pesos.

Ahora para aplicar el filtrado trabajamos con la matriz de pesos obteniendo de la imagen la escala de gris que tenga cada pixel y multiplicamos por cada peso asignado en la matriz, obteniendo un nuevo valor para cada pixel y pixeles cercanos, formando así el desenfocado necesario.

- Algoritmo de Suzuki

Este algoritmo fue presentado en 1985 por Satoshi Suzuki que estudió en la universidad de Shizuoka en Japón. Este algoritmo es utilizado en la herramienta OpenCV para el reconocimiento de contornos en una imagen, para lo cual solicita primero que la imagen sea preprocesada convirtiendo sus pixeles en ceros y unos; es decir, en blanco y negro, puesto que si la imagen es a colores, el algoritmo no funciona, ahora una vez procesada la imagen, lo que hace el algoritmo es tener unas variables de almacenamiento que actuarán como buffer de pixeles recorridos, que partirá de una copia de la imagen, un contador de

contornos encontrados y almacenamiento de los pixeles pertenecientes al contorno. Para encontrar los contornos, el algoritmo recorre la imagen de arriba hacia abajo y de izquierda a derecha. Básicamente, va en búsqueda de todos los puntos 1 que estén en conjunto unidos alrededor del pixel analizado; en caso encuentre un pixel de 0 a 1, indica que se crea una transición de un contorno a otro por lo que se almacena hasta el punto 0 un contorno y se continúa con el otro. Una dificultad que presenta este algoritmo es cuando se llega a los bordes de la imagen, para ello el algoritmo crea una capa de pixeles con valor 0 alrededor para marcar un inicio y fin del barrido de la imagen, obteniendo así un conjunto de contornos de una imagen. En caso de OpenCV se hicieron modificaciones como se mencionó anteriormente, para que brinde coordenadas de una imagen y otros métodos de salida que se deseen utilizar dependiendo de las necesidades del proyecto; pero el uso del algoritmo en sí, es el mismo que presentó Suzuki en su paper (Ver Figura 44).

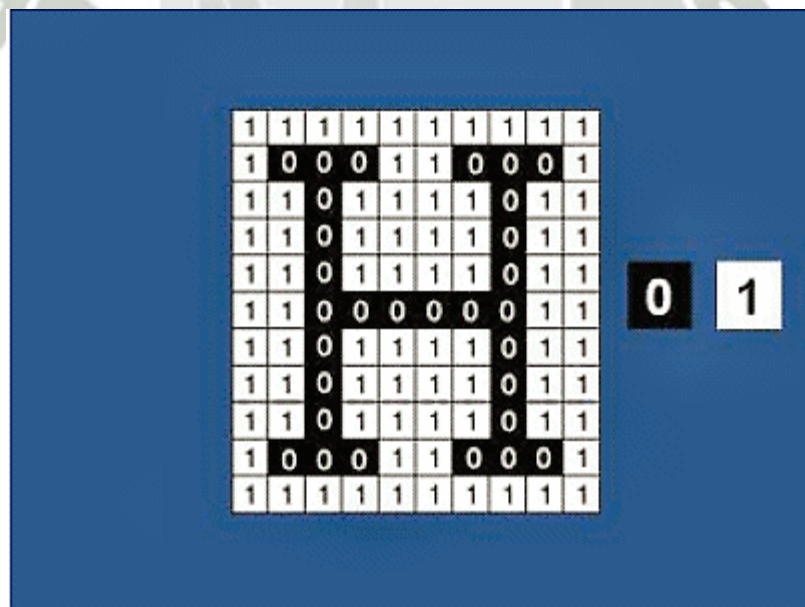


Figura 44. Selección de Contornos con el Algoritmo de Suzuki

Fuente: imageprocessingplace.com

- Algoritmo de Graham (Método de Graham)

Este algoritmo se hace uso en la función Convexhull de OpenCV, el cual nos permite detectar puntos convexos en un plano 2D; tiene una complejidad de $O(N \log N)$, pero su funcionalidad y acertamiento son bien validados por la comunidad. Principalmente, para esto el algoritmo trabaja con puntos o coordenadas de un plano X Y, ubicando diferentes puntos de este con el fin de crear un polígono que encierre a los puntos dentro, utilizando los puntos que estén más orientados al exterior (Ver Figura 45).

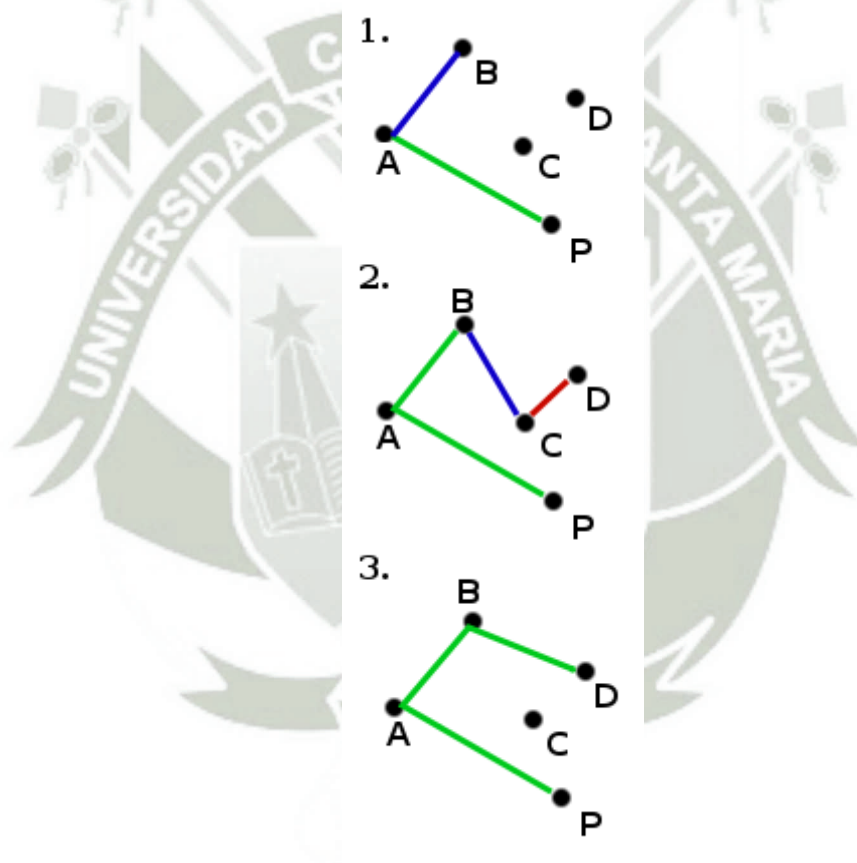


Figura 45. Selección de Contornos con el Algoritmo de Suzuki

Fuente: Marek Baczyński, Gdańsk, Poland

Para empezar con el algoritmo, primero se ubica el punto que esté en menor escala en el eje Y; en caso hallan 2 o más puntos en esa misma escala, se busca el que esté en la menor escala en X. Este punto es denominado como P, ahora a

partir de este punto, se busca ordenar los puntos según el ángulo del segmento que los une con el punto P y el eje ordenado; para ellos se utiliza un algoritmo de ordenamiento que los ponga en forma ordenada creciente. Ahora, para cada punto se considera también si el movimiento es un “giro a la derecha” o “giro a la izquierda”. Si el movimiento es a la derecha, entonces este no es considerado y se pasa al siguiente punto; cuando se halla un giro a la izquierda, el algoritmo analiza el siguiente punto. En caso de que existan puntos alineados pertenecientes a la envolvente, los centrales pueden ser descartados o considerados como parte de la misma. Ahora, para saber si se va a la derecha o a la izquierda, un cálculo de producto vectorial entre 2 vectores definidos por las coordenadas x_1, y_1 ; x_2, y_2 y $x_1, y_1; x_3, y_3$ quedando una ecuación como la siguiente $(x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1)$ en donde si el resultado es 0, los puntos son alineados; si es positivo, el giro es a la izquierda; de lo contrario, es a la derecha. Así es como funciona el algoritmo implementado por OpenCV en la función ConvexHull

3.2.2. Requerimientos de Hardware

Como se mencionó anteriormente, se tiene varios ejemplos de hardware que se puede utilizar para el proyecto. Sin embargo, se debe seleccionar el idóneo para este proyecto; a continuación, se muestra una tabla de comparación con los hardware libres más utilizados y reconocidos (Ver Tabla 4).

Tabla 4. Tabla de comparación de Hardware Libres

Requerimientos de Hardware								
Hardware	Última Actualización	Comunidad	Procesador	Memoria Ram	Proyectos	Puerto Cámara	Documentacion	Costo
Arduino Uno R3	Enero, 2013	Muchas personas	Microcontrolador ATmega328	32kb	Muchos	Si, Adaptador	Si	\$25
Raspberry Pi	Febrero, 2016	Muchas personas	Quad Core 1.2GHz Broadcom BCM2837 64bit CPU	1GB	Muchos	Si	Si	\$35
CubieBoard	Marzo, 2016	Pocas personas	Allwinner H8 ARM Cortex A7	2GB	Pocos	Si	Si	\$99
GooseBerry	Marzo, 2013	Pocas personas	ARM Cortex - A8	no tiene RAM	Pocos	No	Poca	\$63

Fuente: Elaboración propia

Como se ve en la tabla, la comunidad de usuarios utiliza más Arduino y Raspberry Pi; por lo que la documentación, la ayuda y los ejemplos que brinde la comunidad ayudan a crecer el desarrollo y mejorar del uso de estas 2 placas. Sin embargo, también se debe ver la parte del software en ambas y la que está más ubicada en cuanto al uso de lenguajes de programación es Raspberry Pi.

Raspberry Pi

Existen 3 versiones actualmente del Raspberry Pi, la cual para este proyecto se seleccionó el Raspberry Pi 3 B, ya que es mucho más rápido que su primera versión; además de que unas de las diferencias principales con la anterior versión, es que el Raspberry Pi 3 incluye Wifi y Bluetooth incorporado, además de tener ahora 4 núcleos de procesamiento. Tiene las siguientes características (Ver Figura 46):



Figura 46. Raspberry Pi 3 B Model

Fuente: Raspberrypi.org

- 1.2 GHz de 64 bits Quad Core ARMv8 CPU
- 1 GB RAM
- 4 puertos USB
- 40 GPIO pins
- Puerto Full HDMI
- Puerto Ethernet
- Conector de audio de 3.5 mm combinado y video compuesto
- Interfaz de Cámara (CSI)
- Interfaz Display (DSI)
- Puerto de tarjeta MicroSD
- VideoCore IV 3D Graphics Core
- 802.11n Wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)

Gracias a estas características, es fácil correr con el procesador distribuciones de Linux que se ofrecen para Raspberry Pi; incluyendo también la versión del Windows 10 para Raspberry Pi.

Módulo de Cámara Megapíxel

Existen 2 versiones de este módulo de cámara megapíxel; sin embargo, por ser más novedoso la versión 2, su costo aún es elevado, por lo que se trabajó para este proyecto con la versión 1 del módulo la cual es suficiente calidad de imagen la que ofrece para el proyecto.

Este hardware es una cámara de 5 MP, soporta hasta imagen HD de 1080 y 720 y para video es VGA; se conecta a través del puerto CSI del Raspberry Pi directamente, se configura como habilitado y está listo para ser usado.

3.2.3. Especificación de requisitos del software

Obtención de requerimientos

Para obtener los requerimientos necesarios para la realización del proyecto, fue necesario hacer algunas encuestas y conversaciones con personas que realmente están interesadas en mejorar la calidad de atención del paciente. Primeramente, las encuestas se realizaron al personal que es parte del área de hospitalización del Hospital I, Edmundo Escomel Paucarpata, estas encuestas se verán en el siguiente punto, por ahora se mencionará algunos requerimientos que se obtuvieron conversando con personas del Hospital interesadas en el proyecto.

Matriz de Trazabilidad de Requerimientos

Se mostrará la matriz de requerimientos necesaria para el proyecto, los cuales están estrechamente relacionadas a las entrevistas y encuestas aplicadas (Ver Tabla 5). Además, se muestra la tabla de trazabilidad de los requerimientos obtenidos del sistema con los del programa (Ver Tabla 6).

Tabla 5. Matriz de requerimientos del Sistema

MATRIZ DE TRAZABILIDAD DE REQUISITOS							
ID	REQUISITO	TIPO	PRIORIDAD	ESTADO	OBJETIVO	ESTADO	VALIDACION
1	Reconocer una solicitud del paciente	Raspberry Pi	A	Activo	Objetivo 6	Completado	VALIDADO
2	Registrar la solicitud de atención del paciente	Raspberry Pi	A	Activo	Objetivo 5	Completado	VALIDADO
3	Registrar personal que atendió al paciente	Plataforma Web	A	Activo	Objetivo 4	Completado	VALIDADO
4	Registrar breve descripción de la atención	Plataforma Web	A	Activo	Objetivo 4	Completado	VALIDADO
5	Identificar que paciente fue atendido	Plataforma Web	A	Activo	Objetivo 4	Completado	VALIDADO
6	Mantener en constante vigilancia al paciente	Raspberry Pi	M	Activo	Objetivo 6	Completado	VALIDADO
7	Ver lista de atenciones registradas	Plataforma Web	M	Activo	Objetivo 4	Completado	VALIDADO
8	Alertar sobre atención del paciente al personal	Plataforma Web	A	Activo	Objetivo 6	Completado	VALIDADO
9	Tener Constancia de la Atención	Plataforma Web	B	Activo	Objetivo 4	Completado	VALIDADO
10	Buscar atenciones en el registro	Plataforma Web	M	Activo	Objetivo 4	Completado	VALIDADO
11	Generar reportes de atenciones	Plataforma Web	B	Activo	Objetivo 4	Completado	VALIDADO
12	Editar atenciones ya registradas	Plataforma Web	B	Activo	Objetivo 4	Completado	VALIDADO
13	Identificar usuarios que hagan uso del sistema	Plataforma Web	B	Activo	Objetivo 4	Completado	VALIDADO
14	Alertar al paciente que su petición ha sido registrado	Raspberry Pi	B	Activo	Objetivo 6	Completado	VALIDADO

		Cantidad de Req.
A	Alta	6
M	Media	3
B	Baja	5

Fuente: Elaboración propia

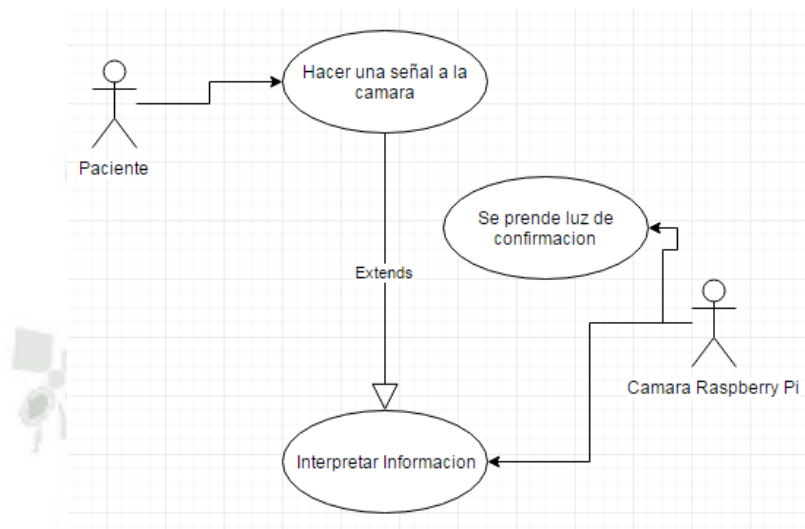
Tabla 6. Matriz de requerimientos del sistema y del programa

RP	RP1: La cámara sólo detecta la señal correcta del paciente	RP2: La cámara también puede detectar a pacientes que no pueden hacer señales	RP3: La cámara debe indicar al paciente que se reconoció su señal	RP4: La cámara debe enviar esos datos al sistema	RP5: La cámara registrará la información en la base de datos	RP6: El sistema, se manejará mediante usuarios en una ventana de login	RP7: Se tendrá una pantalla principal con las opciones del sistema	RP8: Se podrá visualizar las alertas generadas	RP9: Se podrá elegir un estado de la alerta	RP10: Se podrá editar el personal que atendió en dicha alerta	RP11: Se podrá editar la descripción de dicha alerta	RP12: Se podrá generar una constancia que valide la atención	RP13: Se podrá detectar quien registró la atención (usuario)	RP14: Se contabilizará la cantidad de atenciones de dicha alerta	RP15: No se podrá cambiar el estado de la atención	RP16: Se dará un aviso en el computador, cuando ocurra una nueva alerta	RP17: Se notificará cuantas alertas nuevas hay	RP18: Se podrá visualizar las alertas del sistema	RP19: Se podrá visualizar los pacientes y sus respectivas camas	RP20: Se tendrá un sistema de registro de pacientes	RP21: Se podrá indicar si se cambio de cama a un paciente	RP22: Se podrá indicar cuando un paciente ya no se dio de alta	RP23: Se tendrá un modulo de administrador	RP24: Se podrá realizar búsquedas por ID(administrador)	RP25: Se podrá crear nuevos usuarios(administrador)	RP26: Se podrá ver la información de un usuario	RP27: Se podrá generar un reporte según el día indicado	RP28: Se podrá generar un reporte mensual según la fecha indicada	RP29: Se podrá poner en suspensión la cámara	RP30: Se podrá ingresar a la IP de un Raspberry Pi y ver a los pacientes		
RS																																
RS1: Reconocer una solicitud del paciente	↑	↑																														↑
RS2: Registrar la solicitud de atención del paciente				↑	↑		↑	↑	↑																							
RS3: Registrar personal que atendió al paciente										↑																						
RS4: Registrar breve descripción de la atención											↑																					
RS5: Identificar que paciente fue atendido																		↑	↑	↑	↑											
RS6: Mantener en constante vigilancia al paciente																																↑
RS7: Ver lista de atenciones registradas																		↑														
RS8: Alertar sobre atención del paciente al personal																↑	↑															
RS9: Tener Constancia de la Atención											↑																					
RS10: Buscar atenciones en el registro																									↑							
RS11: Generar reportes de atenciones																						↑						↑	↑			
RS12: Editar atenciones ya registradas														↑	↑																	
RS13: Identificar usuarios que hagan uso del sistema						↑							↑	↑	↑										↑	↑						
RS14: Alertar al paciente que su petición ha sido registrado			↑																													

Fuente: Elaboración propia

Detalles de Requerimientos

Requerimientos 1(A): Tiene como propósito centrarse en la tarea de que cuando un paciente haga una señal, el sistema pueda reconocer esa señal; identificar que el paciente necesita atención, siendo interpretado como una especie de solicitud por parte del paciente para que se le pueda atender



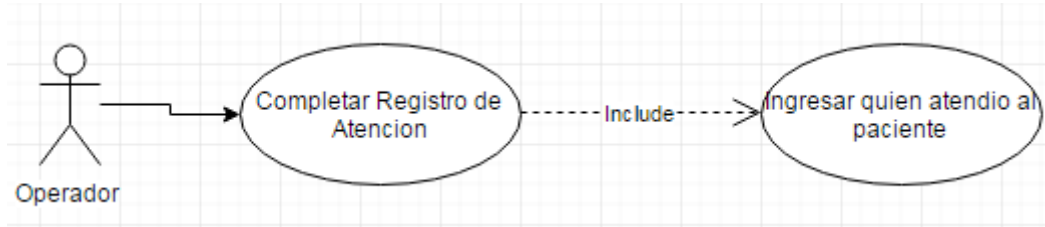
Documentación del Caso de Uso		
Nro.	Sección	Contenido
1	Nro. de Caso de uso	Caso 1
2	Nombre	Reconocer una solicitud del paciente
3	Autor	Jesús Alpaca Rendón
4	Prioridad	Alta
5	Estado crítico	Puede anular el uso del sistema
6	Personal responsable	Doctores, enfermeras
7	Descripción	Se realiza la detección de señales por parte del paciente
8	Actores	Paciente, Cámara Raspberry
9	Precondición	Encender la cámara y ejecutar el programa
10	Poscondición	Esperar un tiempo para permitir reconocer otra señal por parte del paciente
11	Escenario principal	1.- El paciente realiza una señal válida a la cámara 2.- La cámara después de 5 segundos, la reconoce 3.- La cámara envía datos adquiridos después de interpretar la señal a la base de datos 4.- Se notifica al sistema web
12	Escenarios alternativos	1.- El paciente realiza una señal inválida a la cámara 2.- La cámara después de 5 segundos, la reconoce; pero no de la manera correcta 3.- La cámara envía los datos adquiridos a la base de datos 4.- Se notifica al sistema web, con un mensaje para revisar
13	Resultado	Una solicitud generada por el paciente

Requerimientos 2(A): Una vez que el sistema reciba una solicitud de atención, esta será registrada en la base de datos; creando un nuevo elemento en la tabla de Atenciones, completando hasta cierto punto el registro



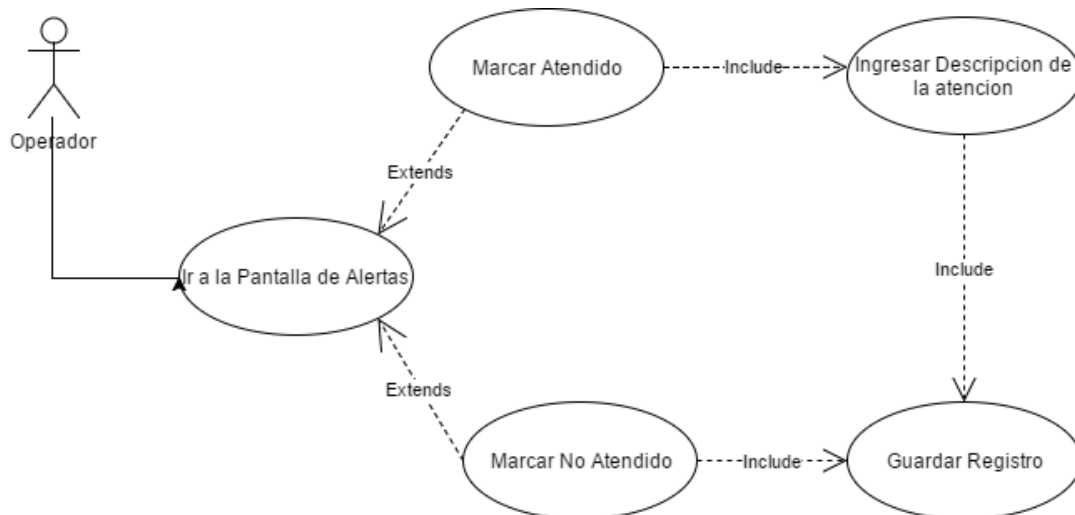
Documentación del Caso de Uso		
Nro.	Sección	Contenido
1	Nro. de Caso de uso	Caso 2
2	Nombre	Registrar solicitud de atención del paciente
3	Autor	Jesús Alpaca Rendón
4	Prioridad	Alta
5	Estado crítico	Puede anular el uso del sistema
6	Personal responsable	Operador
7	Descripción	Se registrará cada solicitud de atención por parte del paciente
8	Actores	Cámara Raspberry Pi
9	Precondición	Envío de datos del Raspberry a la base de datos
10	Poscondición	Aparece en la página web la solicitud
11	Escenario principal	1.- La cámara Raspberry Pi envía los datos a la página web 2.- La página web recibe esos datos directamente a la base de datos 3.- La página web almacena el registro y lo muestra como nuevo registro 4.- Se solicita actualizar la página web
12	Escenarios alternativos	No hay escenario alternativo
13	Resultado	Se registra satisfactoriamente la solicitud

Requerimientos 3(A): Se debe registrar y tener conocimiento de quién fue el que atendió al paciente; tener en el registro de atención, qué personal se encargó de brindar atención al paciente



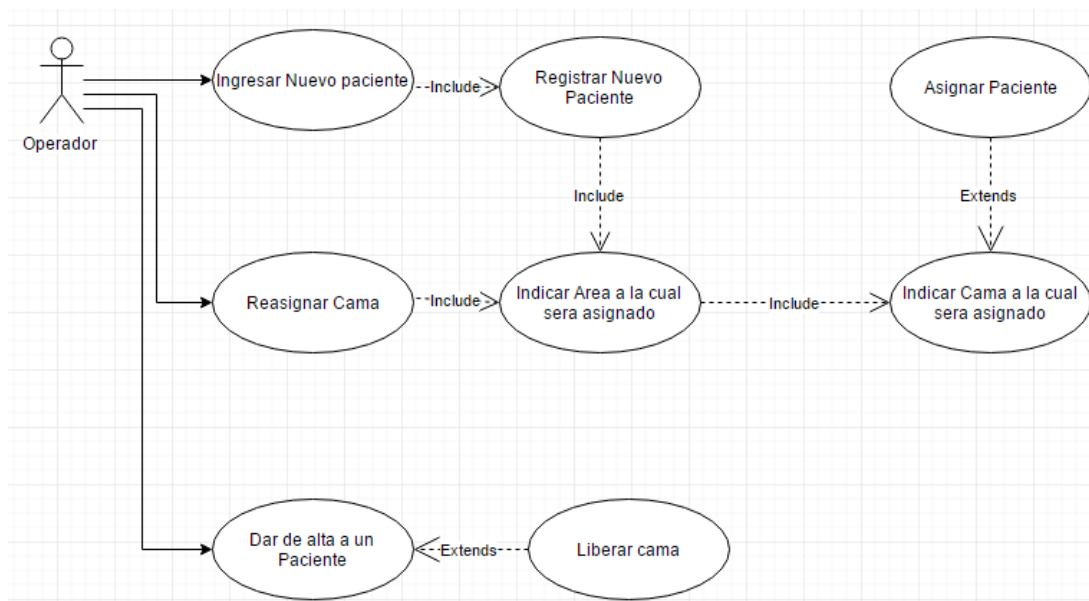
Documentación del Caso de Uso		
Nro.	Sección	Contenido
1	Nro. de Caso de uso	Caso 3
2	Nombre	Registrar el personal que atendió al paciente
3	Autor	Jesús Alpaca Rendón
4	Prioridad	Alta
5	Estado crítico	No genera errores en el sistema
6	Personal responsable	Operador, personal medico
7	Descripción	Se registra el personal que atendió al paciente
8	Actores	Operador
9	Precondición	Tener una solicitud del paciente
10	Poscondición	Tener el registro completo de la atención
11	Escenario principal	1.- En el menú principal, ir a la opción Alertas 2.- Registrar si la atención generada fue atendida o no 3.1.- En caso si fuese atendida, marcar en Atendida 3.2.- en la ventana de edición que se abrirá, seleccionar qué personal atendió al paciente 3.3.- Hacer clic en guardar registro 4.1.- En caso no fue atendida. marcar en No Atendida 4.2.- Se guardará el registro sin tener un miembro del personal asignado a la atención ni descripción de la atención
12	Escenarios alternativos	1.- En caso no se marque nada en una atención, esta seguirá apareciendo en la lista de atenciones urgentes 2.- Por defecto, estará como No Atendida hasta que se registre lo contrario 3.- No se tendrá registrado a un personal que atendió o no atendió
13	Resultado	Se tendrá el personal registrado

Requerimientos 4(A): Al momento de registrar se debe colocar una breve descripción de la atención; si hubo algún inconveniente o algún hecho que se considere importante, colocarlo en el registro y guardarlo



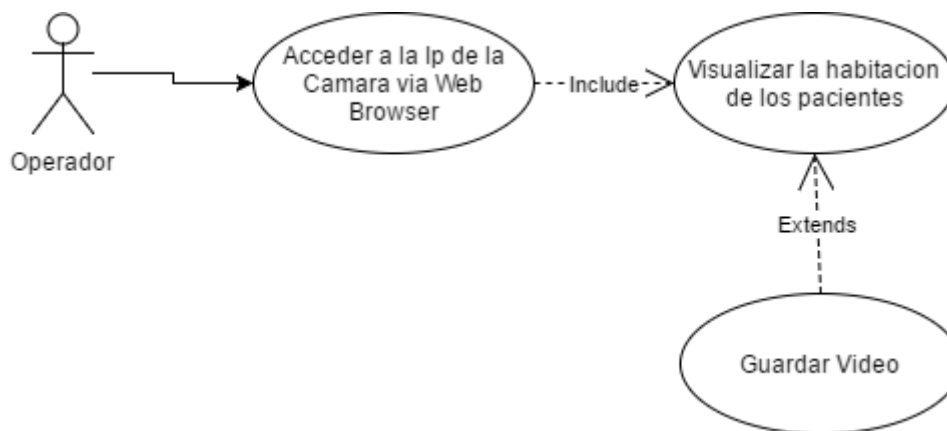
Documentación del Caso de Uso		
Nro.	Sección	Contenido
1	Nro. de Caso de uso	Caso 4
2	Nombre	Registrar una descripción de la atención
3	Autor	Jesús Alpaca Rendón
4	Prioridad	Alta
5	Estado crítico	No genera errores en el sistema
6	Personal responsable	Operador, Personal Médico, Pacientes
7	Descripción	Se registra una pequeña descripción de lo sucedido en la atención
8	Actores	Operador
9	Precondición	Tener una solicitud del paciente
10	Poscondición	Tener el registro completo de la atención
11	Escenario principal	1.- En el menú principal, ir a la opción Alertas 2.- Registrar si la atención generada fue atendida o no 3.1.- En caso si fuese atendida, marcar en atendida 3.2.- En la ventana de edición que se abrirá, redactar una breve descripción de la atención 3.3.- Hacer clic en guardar registro 4.1.- En caso no fue atendida. marcar en No Atendida 4.2.- Se guardará el estado como No Atendida automáticamente, dejando la posibilidad de editarlo posteriormente
12	Escenarios alternativos	1.- En caso no se marque nada en una atención, esta seguirá apareciendo en la lista de atenciones urgentes 2.- Por defecto, estará como No Atendida hasta que se registre lo contrario 3.- No se tendrá registrado una descripción de la atención
13	Resultado	Se tendrá el personal registrado

Requerimientos 5(A): Cuando se registra también se desea tener información de a quién se le brindó esa atención, quién solicitó.



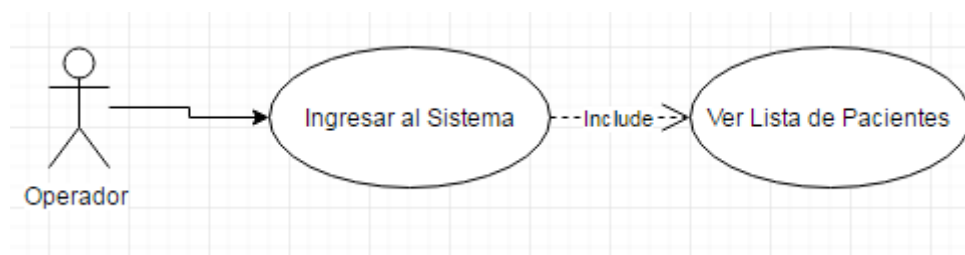
Documentación del Caso de Uso		
Nro.	Sección	Contenido
1	Nro. de Caso de uso	Caso 5
2	Nombre	Identificar qué paciente está solicitando atención
3	Autor	Jesús Alpaca Rendón
4	Prioridad	Alta
5	Estado crítico	Puede generar error en el sistema
6	Personal responsable	Operador
7	Descripción	Se identifica al usuario que generó la alerta mediante un pequeño módulo de asignación de camas, el cual también es parte del sistema
8	Actores	Operador
9	Precondición	Tener al paciente y cama creados en la base de datos
10	Poscondición	Se podrá verificar quién solicitó la atención
11	Escenario principal	1.- En el menú principal, ir a la opción Pacientes 2.- Registrar al paciente en Ingreso de Paciente 3.- Ingresar los datos del paciente 4.- Seleccionar el área al que será hospitalizado 5.- Asignar al paciente una cama libre 6.- Esperar que el paciente solicite atención, el sistema ya lo identificará
12	Escenarios alternativos	1.- En el menú principal, ir a la opción Pacientes 2.- Cambiar de cama al paciente en Reasignación de Cama 3.- Seleccionar el área al que será hospitalizado 4.- Asignar al paciente una cama libre 5.- Esperar que el paciente solicite atención, el sistema ya lo identificará
13	Resultado	Se tendrá registrado al paciente y en qué cama

Requerimientos 6(M): Se desea mantener en constante vigilancia a los pacientes, por lo que el sistema debe tener una manera de estar vigilando al paciente



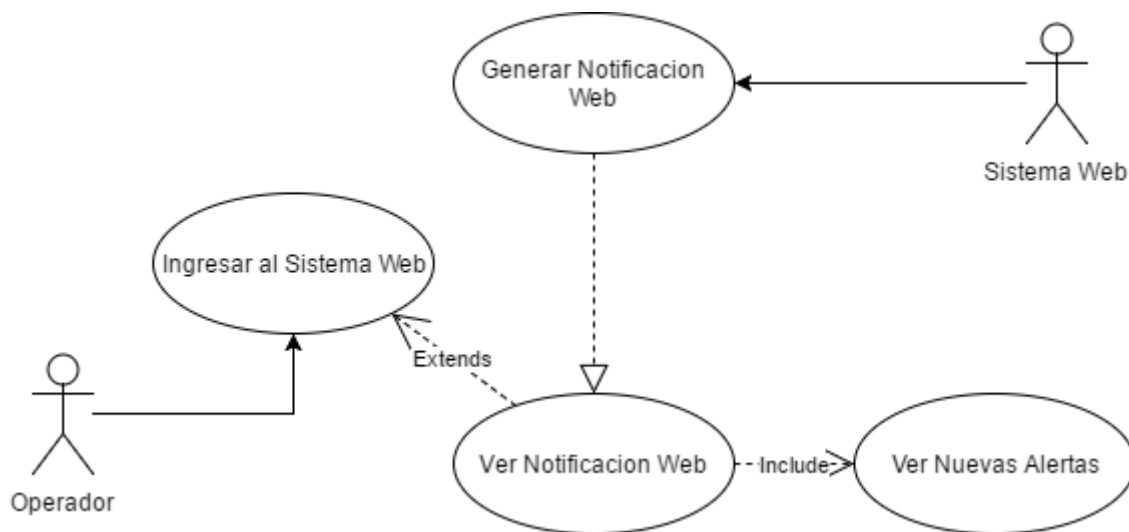
Documentación del Caso de Uso		
Nro.	Sección	Contenido
1	Nro. de Caso de uso	Caso 6
2	Nombre	Mantener en constante vigilancia al paciente
3	Autor	Jesús Alpaca Rendón
4	Prioridad	Media
5	Estado crítico	No genera problemas al sistema
6	Personal responsable	Operador
7	Descripción	Mantener en constante vigilancia al paciente mediante la cámara de Raspberry Pi
8	Actores	Operador
9	Precondición	Encender la Cámara Raspberry
10	Poscondición	Visualizar desde el computador el envío de imágenes
11	Escenario principal	1.- Ingresar la IP correcta del Raspberry Pi de vigilancia 2.- Visualizar las imágenes obtenidas por la cámara 3.- Opcionalmente, guardar el video de la vigilancia.
12	Escenarios alternativos	No hay escenarios alternativos
13	Resultado	Se tiene bajo vigilancia al paciente

Requerimientos 7(M): el propósito de este requerimiento es brindar al operador una manera de poder ver una lista de todas las atenciones que se realizaron



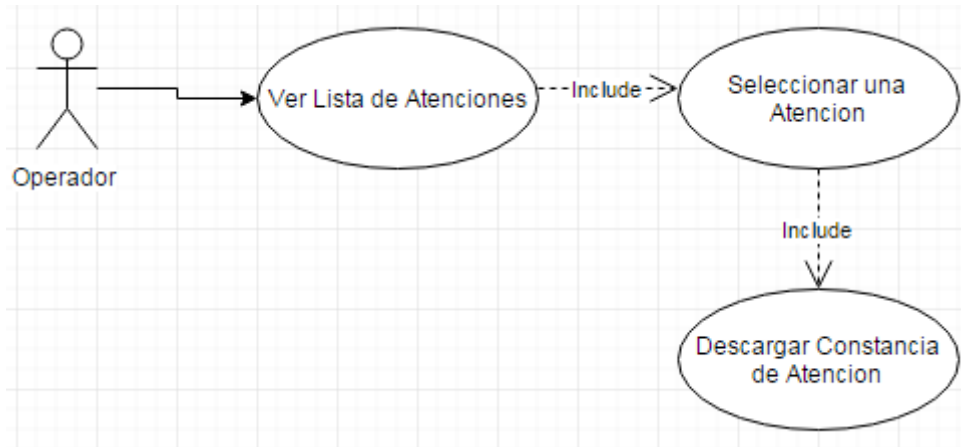
Documentación del Caso de Uso		
Nro.	Sección	Contenido
1	Nro. de Caso de uso	Caso 7
2	Nombre	Ver lista de atenciones registradas
3	Autor	Jesús Alpaca Rendón
4	Prioridad	Media
5	Estado crítico	No genera problemas en el sistema
6	Personal responsable	Operador
7	Descripción	Se puede visualizar una lista de las atenciones registradas
8	Actores	Operador
9	Precondición	Ingresar al sistema web
10	Poscondición	Se podrá visualizar las atenciones registradas
11	Escenario principal	1.- Ingresar a la opción de Lista de Atenciones 2.- Visualizar y seleccionar atenciones 3.1.- Filtrar las atenciones de acuerdo con las columnas seleccionadas 3.2.- Realizar búsqueda de atenciones con palabras claves
12	Escenarios alternativos	No hay escenarios alternativos
13	Resultado	Se puede buscar seleccionar y visualizar atenciones registradas

Requerimientos 8(A): Se desea que los registros generados, muestren una alerta para que el personal esté atento a las atenciones correspondientes



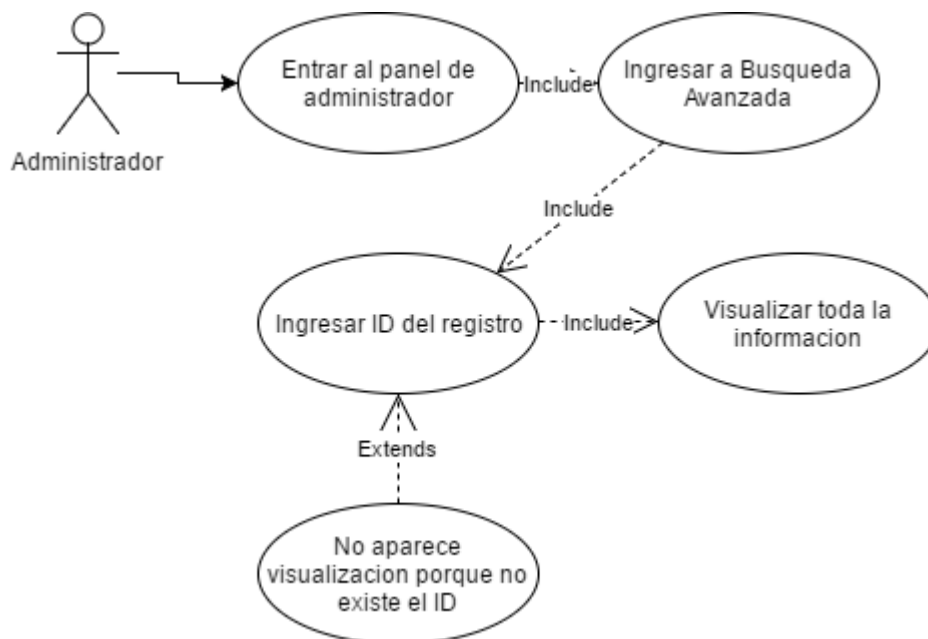
Documentación del Caso de Uso		
Nro.	Sección	Contenido
1	Nro. de Caso de uso	Caso 8
2	Nombre	Alertar sobre atención del paciente a personal
3	Autor	Jesús Alpaca Rendón
4	Prioridad	Alta
5	Estado crítico	No genera problemas al sistema
6	Personal responsable	Operador/personal medico
7	Descripción	Se indicará al personal de turno que atienda al paciente apenas se genere una alerta
8	Actores	Operador/Raspberry
9	Precondición	Actualizar la página
10	Poscondición	Marcar la atención como “Atendida”
11	Escenario principal	1.- El Raspberry emitirá un sonido mediante un buzzer el cual indicará que se ha generado una nueva solicitud 2.- Actualizar la página web 3.- Verificar nuevos registros en la tabla de alertas de la pantalla principal 4.- Dar aviso al personal médico en caso no haya escuchado el sonido emitido por la Cámara Raspberry Pi 4.- Completar registro de las nuevas atenciones
12	Escenarios alternativos	No hay escenarios alternativos
13	Resultado	Atender al paciente correctamente

Requerimientos 9(B): Como medida de comprobación, una vez se complete una atención al paciente, el personal puede solicitar que se genere una constancia de haber atendido al paciente



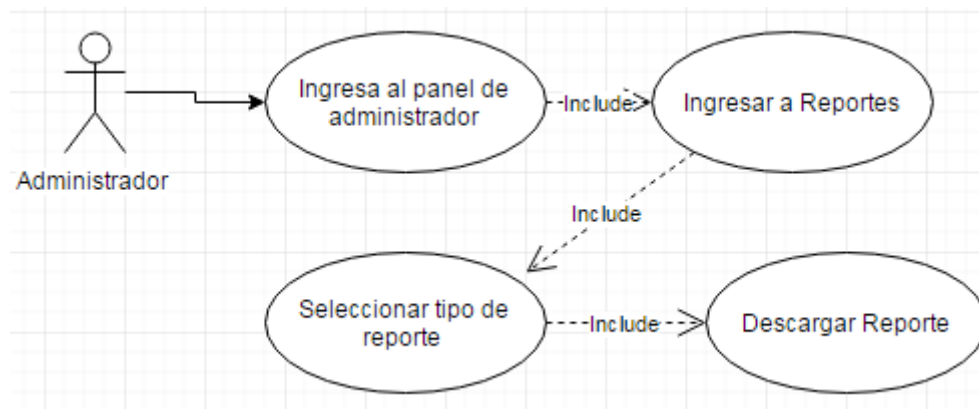
Documentación del Caso de Uso		
Nro.	Sección	Contenido
1	Nro. de Caso de uso	Caso 9
2	Nombre	Tener constancia de Atención
3	Autor	Jesús Alpaca Rendón
4	Prioridad	Baja
5	Estado crítico	No genera errores en el sistema
6	Personal responsable	Operador/personal médico/pacientes
7	Descripción	Tener un documento que respalde la atención o no atención por parte del personal
8	Actores	Operador
9	Precondición	Tener una alerta registrada
10	Poscondición	Tener el documento en físico
11	Escenario principal	1.- Seleccionar la atención de la lista 2.- Ingresar a Ver o Editar 3.- Al final de la información, dar clic en descargar constancia de atención 4.- Abrir el PDF descargado 5.- Imprimirlo si se desea
12	Escenarios alternativos	No hay escenarios alternativos
13	Resultado	Se obtiene una constancia de atención que puede ser solicitada opcionalmente

Requerimientos 10(M): Se desea tener un método de búsqueda de una atención en especial



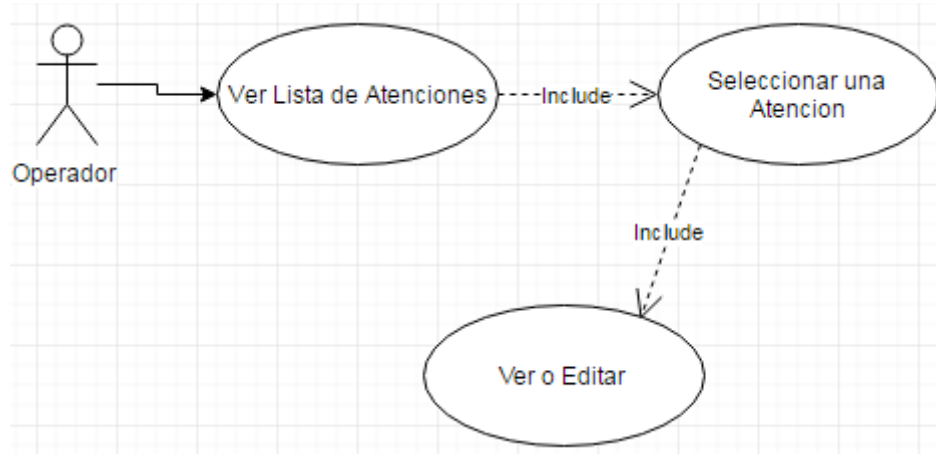
Documentación del Caso de Uso		
Nro.	Sección	Contenido
1	Nro. de Caso de uso	Caso 10
2	Nombre	Buscar atenciones en el registro
3	Autor	Jesús Alpaca Rendón
4	Prioridad	Media
5	Estado crítico	No genera problemas en el sistema
6	Personal responsable	Operador
7	Descripción	Se realiza la búsqueda de una atención o varias
8	Actores	Operador/Administrador
9	Precondición	Tener varias atenciones registradas
10	Poscondición	Dar la opción de editar la atención encontrada
11	Escenario principal	1.- Ingresar a Lista de Atenciones 2.- Ingresar una palabra clave de búsqueda 3.- Encontrar la(s) solicitud(es) que se indicaron
12	Escenarios alternativos	1.- En caso de Administrador. seleccionar Búsqueda Avanzada 2.- Ingresar ID de la atención 3.- Verificar toda la información de la atención seleccionada
13	Resultado	Encontrar la atención por medio de un parámetro

Requerimientos 11(B): Se desea que el sistema genere reportes necesarios para que los jefes del cuerpo médico tomen decisiones



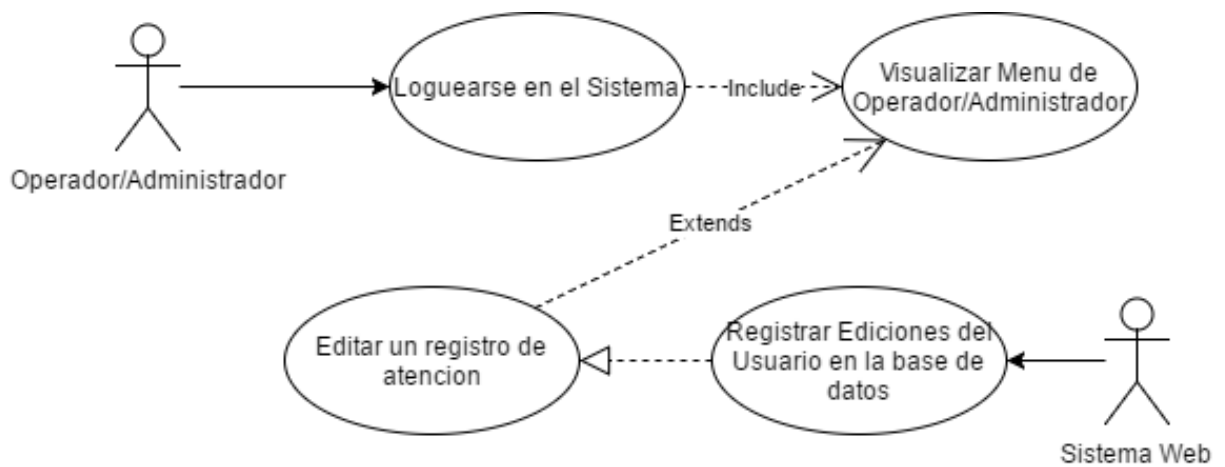
Documentación del Caso de Uso		
Nro.	Sección	Contenido
1	Nro. de Caso de uso	Caso 11
2	Nombre	Generar reportes de atención
3	Autor	Jesús Alpaca Rendón
4	Prioridad	Baja
5	Estado crítico	No genera errores en el sistema
6	Personal responsable	Administrador/ jefes del cuerpo médico
7	Descripción	Genera reportes diarios y mensuales sobre las atenciones realizadas
8	Actores	Administrador
9	Precondición	Tener registros completos de atenciones
10	Poscondición	Tener reportes generados automáticamente por el sistema
11	Escenario principal	1.- Seleccionar Reportes 2.- Seleccionar el tipo de reporte 3.1.- En caso selecciono Reporte Diario, haga clic en descargar 3.2.- Abrir el PDF e imprimirlo si desea 4.1.- En caso selecciono Reporte Mensual, haga clic en descargar 4.2.- Abrir el PDF e imprimirlo si desea
12	Escenarios alternativos	No hay escenarios alternativos
13	Resultado	Tener el reporte para la toma de decisiones y hacer seguimiento al estado actual de las áreas de hospitalización

Requerimientos 12(B): Tiene como propósito corregir atenciones que se registraron mal o con datos erróneos



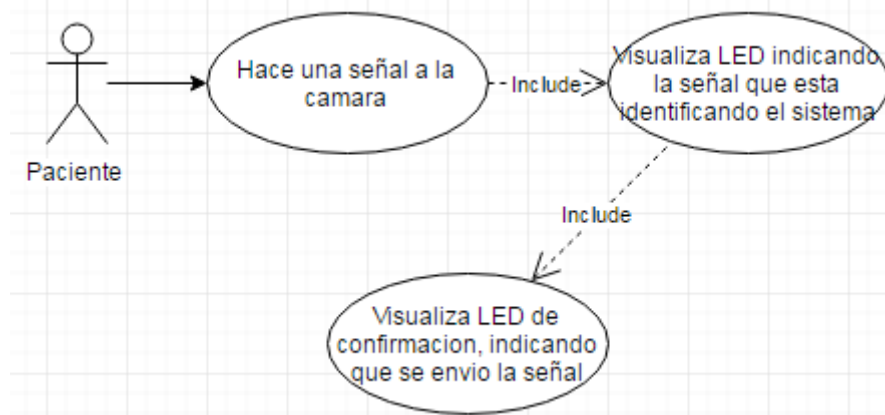
Documentación del Caso de Uso		
Nro.	Sección	Contenido
1	Nro. de Caso de uso	Caso 12
2	Nombre	Corregir atenciones mal registradas
3	Autor	Jesús Alpaca Rendón
4	Prioridad	Baja
5	Estado crítico	No genera errores en el sistema
6	Personal responsable	Operador/Administrador
7	Descripción	Permitir modificar registros de atención y saber quién realizó dicha modificación
8	Actores	Operador/Administrador
9	Precondición	Tener un registro con datos erróneos
10	Poscondición	Tener un registro actualizado
11	Escenario principal	1.- Seleccionar el registro a corregir 2.- Hacer clic en Ver o Editar 3.- Corregir los Datos del registro 4.- Guardar Datos 5.- Visualizar cambios 6.- El sistema web registrará la edición de esa alerta.
12	Escenarios alternativos	1.- En caso del administrador, ingresar al menú Administrador 2.- Hacer clic en Búsqueda Avanzada 3.- Ingresar el ID y presionar Enter 4.- Mostrará entre la información, quién y cuántas veces editó el usuario
13	Resultado	Se tiene un registro de quién editó y se puede actualizar un registro, pero no se puede eliminar

Requerimientos 13(B): Tiene como propósito identificar al operador que hace uso del sistema



Documentación del Caso de Uso		
Nro.	Sección	Contenido
1	Nro. de Caso de uso	Caso 13
2	Nombre	Identificar usuarios que hagan uso del sistema
3	Autor	Jesús Alpaca Rendón
4	Prioridad	Baja
5	Estado crítico	Puede generar errores al momento
6	Personal responsable	Operador/Administración
7	Descripción	Se identifica a la persona que está de operador
8	Actores	Operador/Administrador
9	Precondición	Tener una cuenta en el sistema
10	Poscondición	Realizar acciones en el sistema
11	Escenario principal	1.- Visualizar la pantalla de Login 2.- Ingresar Usuario 3.- Ingresar Contraseña 4.- Darle click en Ingresar 5.1.- En caso estén mal los datos, volverá a pedir 5.2.- En caso estén bien los datos, ingresará mostrando su nombre de usuario en la esquina superior derecha de la pantalla 6.- Dependiendo de los permisos del usuario, podrá visualizar en el menú la opción de Administrador
12	Escenarios alternativos	No hay escenarios alternativos
13	Resultado	Se identifica al personal que está de operador en ese turno

Requerimientos 14(B): Informar de alguna manera al paciente que la cámara detectó la solicitud de atención que él generó al hacer la señal correspondiente



Documentación del Caso de Uso		
Nro.	Sección	Contenido
1	Nro. de Caso de uso	Caso 14
2	Nombre	Alertar al paciente que su solicitud ha sido enviada
3	Autor	Jesús Alpaca Rendón
4	Prioridad	Baja
5	Estado crítico	No genera errores en el sistema
6	Personal responsable	Paciente/Raspberry Pi
7	Descripción	Se indica al usuario la señal que está realizando y se avisa del envío
8	Actores	Paciente
9	Precondición	Hacer una señal válida a la cámara Raspberry
10	Poscondición	Encender Leds indicando al paciente el envío
11	Escenario principal	1.- Se prenderá un LED indicando que la señal está reconociendo el sistema 2.- Después de 5 segundos, se enviará la señal al sistema 3.- Se encenderá el LED azul indicando que se envió la señal exitosamente 4.- Después de 10 segundos, se podrá volver a enviar otra señal.
12	Escenarios alternativos	No hay otro escenario alternativo
13	Resultado	Avisar al usuario la señal que envió y si se logró enviar

3.3. Modelo propuesto

3.3.1. Arquitectura del Software

En esta parte, se detallará el planeamiento que se tuvo previsto para la realización de la tesis con el fin de dar a conocer a detalle lo que se tiene pensado lograr y obtener con el presente proyecto, qué hacer en cada paso y cómo estará estructurado el prototipo.

La arquitectura utilizada para este proyecto es la de MVC, las cuales son:

- Capa Modelo
- Capa Vistas
- Capa Controlador

Se decidió por esta arquitectura debido a que se piensa en una aplicación escalable; independiente de los demás módulos, capaz de mejorarse sin afectar a lo ya realizado, nos permite diferenciar las vistas al usuario (HTML, Javascript), los controladores (Python), y los modelos que son mapeados de la base de datos. El acceso a datos se realiza mediante Querys Django, es un sistema de acceso a datos el cual permite hacer uso de consultas a la base de datos sin necesidad de usar SQL; es decir, Django interpreta mediante Python las consultas que se está realizando, un sistema muy parecido a Linq en C#. Esto facilita la extracción de datos, inserción, actualización y eliminación en la base de datos directamente.

Además, para los reportes es muy útil este proceso ya que las consultas se realizan en menos líneas de código.

Se utilizó la metodología SCRUM con el fin de tener entregables listos y funcionales del proyecto según los requerimientos, además de pensar a futuro que el proyecto podría crecer y requerir más gente en el proceso, teniendo así la información de los entregables ya lista para su definición

3.3.2. Especificación de Diseño

Arquitectura del Sistema

La arquitectura del sistema se refiere a cómo estará compuesto el sistema, qué partes lo constituirán, cómo será el envío de datos, el modelado, etc. Esta sección se dividió en 3 secciones; el diseño físico, explicará cómo será el sistema en términos de hardware; el diseño lógico, en términos de manejo de datos y flujo de la información y los mockups que serán diseños rápidos de las ventanas de la información.



El diagrama de la arquitectura del sistema completo es el siguiente (Ver Figura 47):

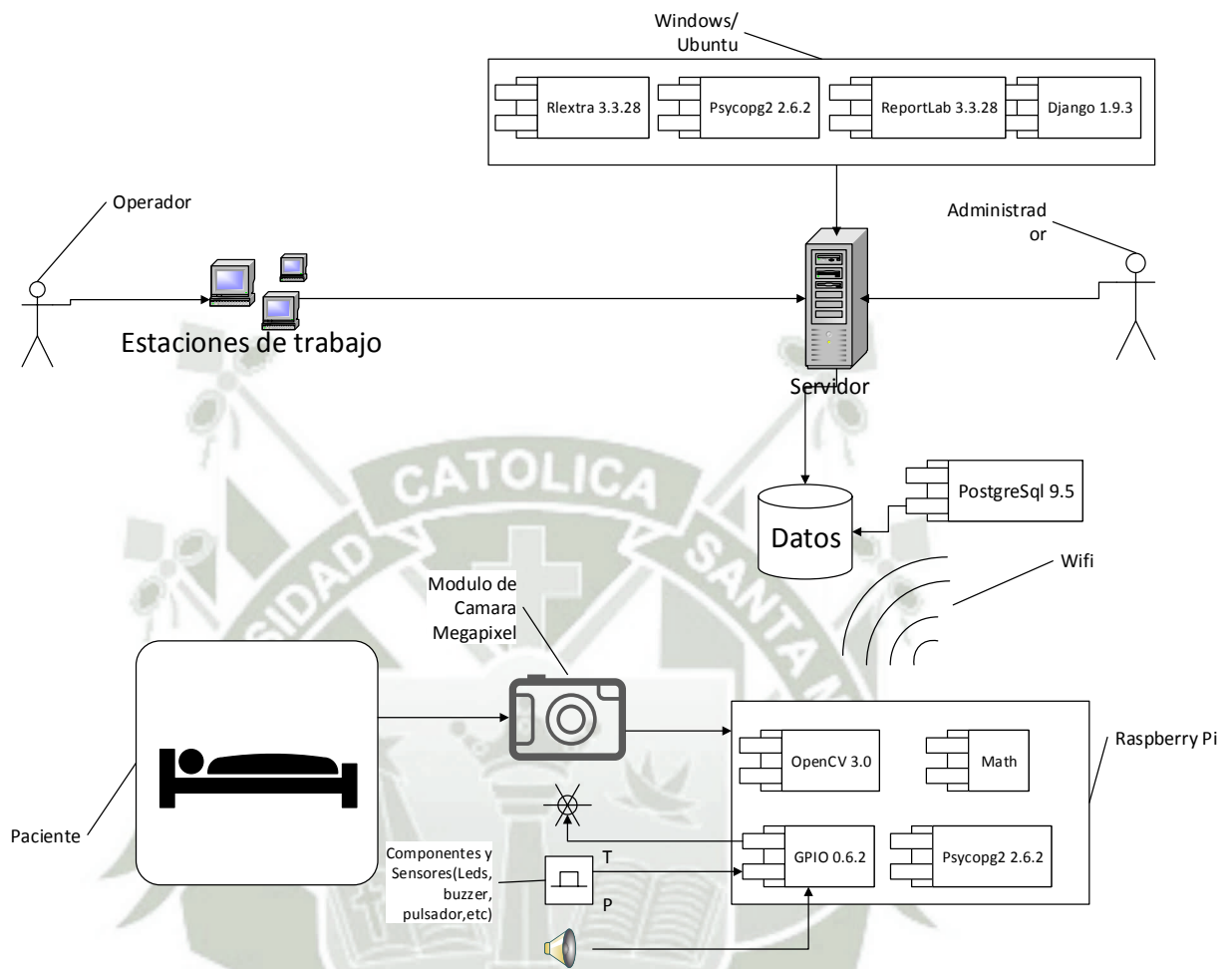


Figura 47. Diagrama de Entidad-Relación

Fuente: Elaboración propia

- **Base de Datos**

La base de datos del proyecto está hecha en PostgreSQL ya que al ser una base de datos libre y tener una conexión sencilla con Python y Django, es factible trabajar con ella; Principalmente, cabe recalcar que la mayoría de tablas de la base de datos que se creó ya existen en EsSalud, por lo que se infiere que cualquier otro centro de salud tenga tablas parecidas, además de considerar en

este proyecto solo las columnas necesarias para que el proyecto pueda funcionar de manera correcta.

El modelo Entidad-Relación es el siguiente (Ver Figura 48):

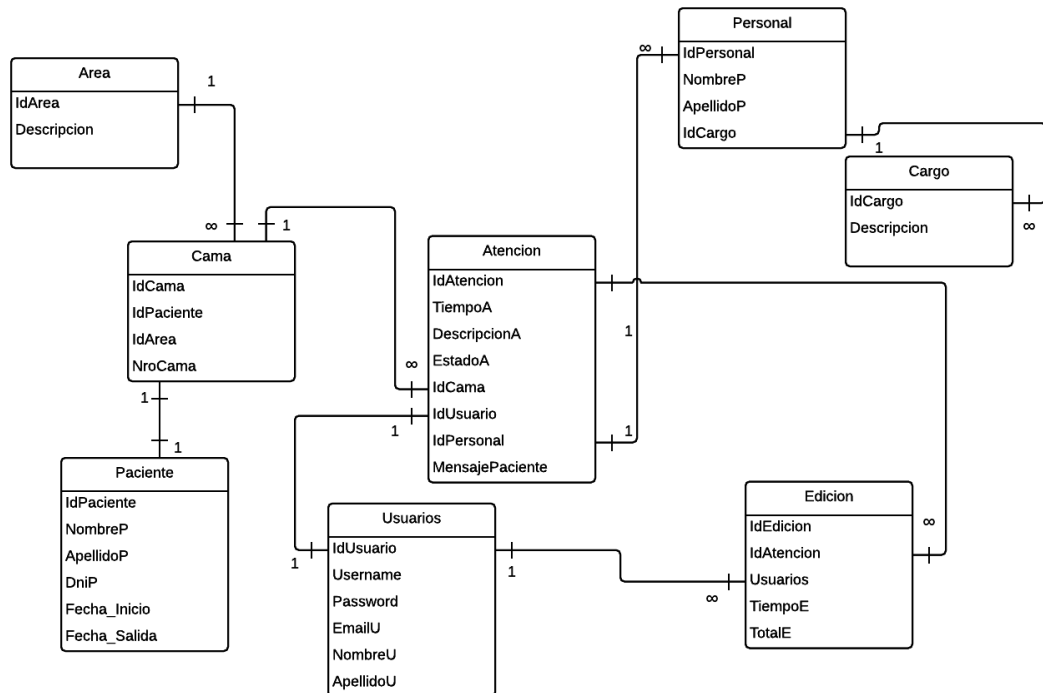


Figura 48. Diagrama de Entidad-Relación

Fuente: Elaboración propia

Tabla Atención:

Column	Type	Size	Nulls	Auto	Default	Children	Parents
id	serial	10		√	nextval('principal_atencion_id_seq'::regclass)	principal_edicion.idAtencion_id	
NroCamaA_id	int4	10					principal_cama.id
TiempoA	timestamptz	35,6					
DescripcionA	text	2147483647	√			null	
Estado	bool	1					
ID_P_id	int4	10	√			null	principal_personal.id
Nombre_U_id	int4	10	√			null	auth_user.id
AlertaMensajeA	varchar	50					
PacienteA_id	int4	10	√			null	principal_paciente.id

Tabla Cama:

Column	Type	Size	Nulls	Auto	Default	Children	Parents
id	serial	10		√	nextval('principal_cama_id_seq'::regclass)	principal_atencion.NroCamaA_id	
Nro_Cama	varchar	3					
Area_id	int4	10	√			null	principal_area.id
Id_paciente_id	int4	10	√			null	principal_paciente.id
ActivaC	bool	1					
EstadoC	bool	1					

Tabla Área:

Column	Type	Size	Nulls	Auto	Default	Children	Parents
id	serial	10		√	nextval("principal_area_id_seq"::regclass)	principal_cama.Area_id	
Descripcion_Area	varchar	60					

Tabla Paciente:

Column	Type	Size	Nulls	Auto	Default	Children	Parents
id	serial	10		√	nextval("principal_paciente_id_seq"::regclass)	principal_atencion principal_cama	
Nombre_P	varchar	50					
Apellido_P	varchar	50					
Fecha_Ingreso	date	13					
Fecha_Salida	date	13	√			null	
Dni_P	varchar	8	√			null	
Activo_P	bool	1					

Tabla Personal:

Column	Type	Size	Nulls	Auto	Default	Children	Parents
id	serial	10		√	nextval("principal_personal_id_seq"::regclass)	principal_atencion.ID_P_id	
ID_S	varchar	10					
NombreS	varchar	30					
ApellidoS	varchar	30					
Cargo_id	int4	10					principal_puesto.id
ActivoS	bool	1					

Tabla Cargo:

Column	Type	Size	Nulls	Auto	Default	Children	Parents
id	serial	10		√	nextval("principal_puesto_id_seq"::regclass)	principal_personal.Cargo_id	
DescripcionP	varchar	50					

Tabla Edición:

Column	Type	Size	Nulls	Auto	Default	Children	Parents
id	serial	10		√	nextval("principal_edicion_id_seq"::regclass)		
TiempoE	timestampz	35,6					
TotalE	int4	10					
IdUsuarios_id	int4	10					auth_user.id
idAtencion_id	int4	10					principal_atencion.id

- **Diseño Físico**

El diseño físico indica de qué elementos se constituye el proyecto físicamente, es decir, qué componentes o tecnología física se utilizará para el proyecto y cómo se usarán.

Para representar el diseño lógico, se empleó el diagrama de distribución. Estos diagramas se encargan de mostrar la arquitectura física de un sistema informático, al igual que sus interconexiones y software que usará cada equipo (Ver Figura 49).

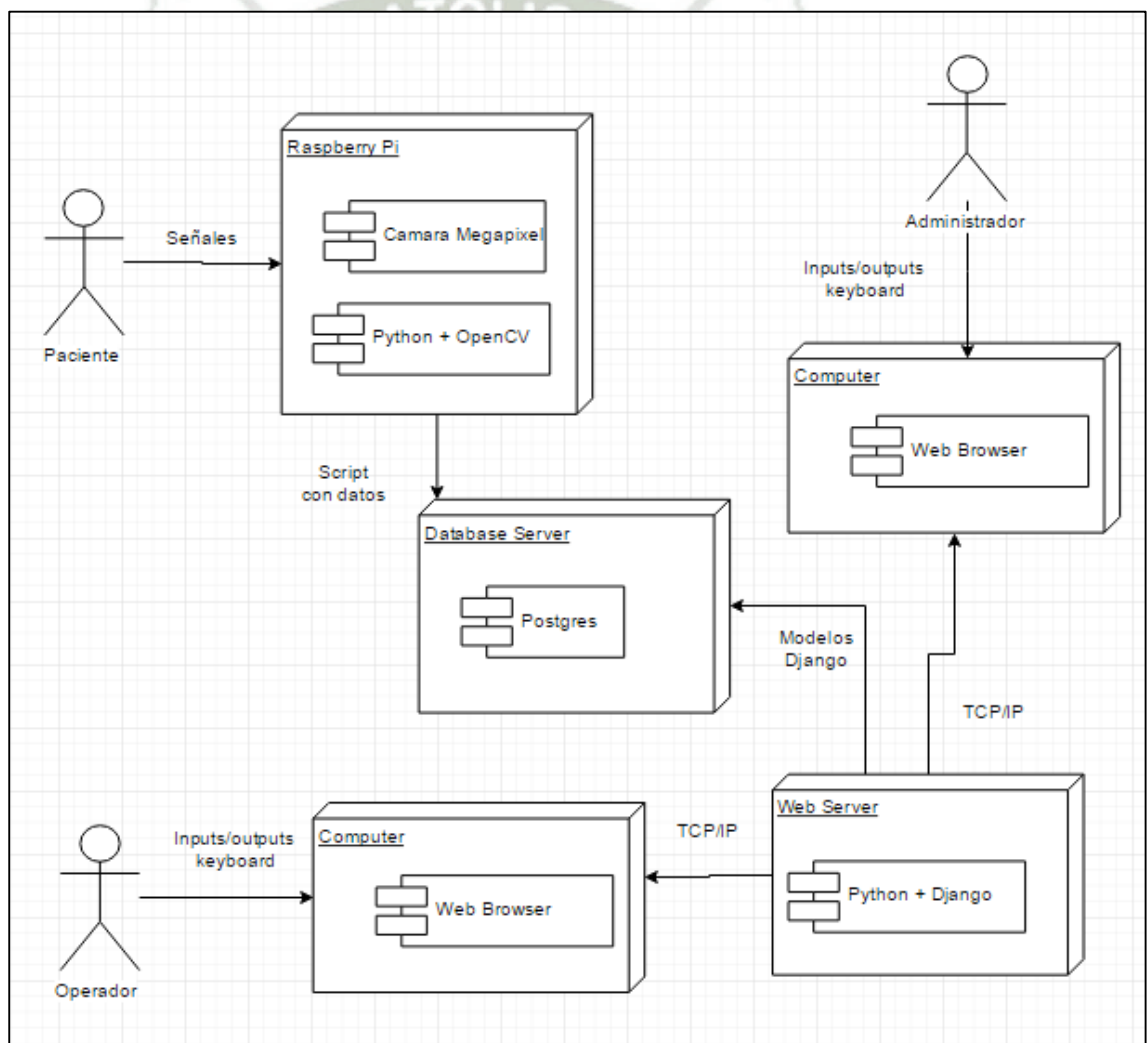


Figura 49. Diagrama de Despliegue

Fuente: Elaboración propia

- **Raspberry Pi:** Este bloque es la representación del Sistema y tecnologías que estarán dentro del Raspberry Pi; actuará con inteligencia artificial haciendo uso de su cámara Megapíxel y Python + OpenCV, para lo cual interpretará las señales correspondientes que le haga el paciente y al recibirlas y procesarlas enviará mediante un script la información a la base de datos.
- **Database Server:** El servidor de base de datos, se encargará de almacenar la información además de recibir la información que brinden los Raspberry para almacenarla y ser extraída por el servidor Web mediante modelos generados por Django
- **Web Server:** Es el servidor donde se almacenará el sistema web, con el cual todos los demás computadores tendrán acceso a él para poder hacer uso del sistema. El servidor tiene instalado Python y Django para el funcionamiento de la página web y la respectiva conexión con PostgreSQL mediante modelos que son generados por Django ya que es un framework del tipo MVC.
- **Computer:** Es desde aquí de donde el operador y el administrador accederán al sistema del servidor, más adelante se mencionará a detalle las funciones de los usuarios.

- **Diseño Lógico**

El diseño lógico mostrará cómo funciona por dentro el sistema, para representar esto se hizo uso de un diagrama de actividades el cual indica cuáles son las partes internas del software y cómo funcionan, qué opciones tienen, qué nos permite hacer, etc. (Ver Figura 50).

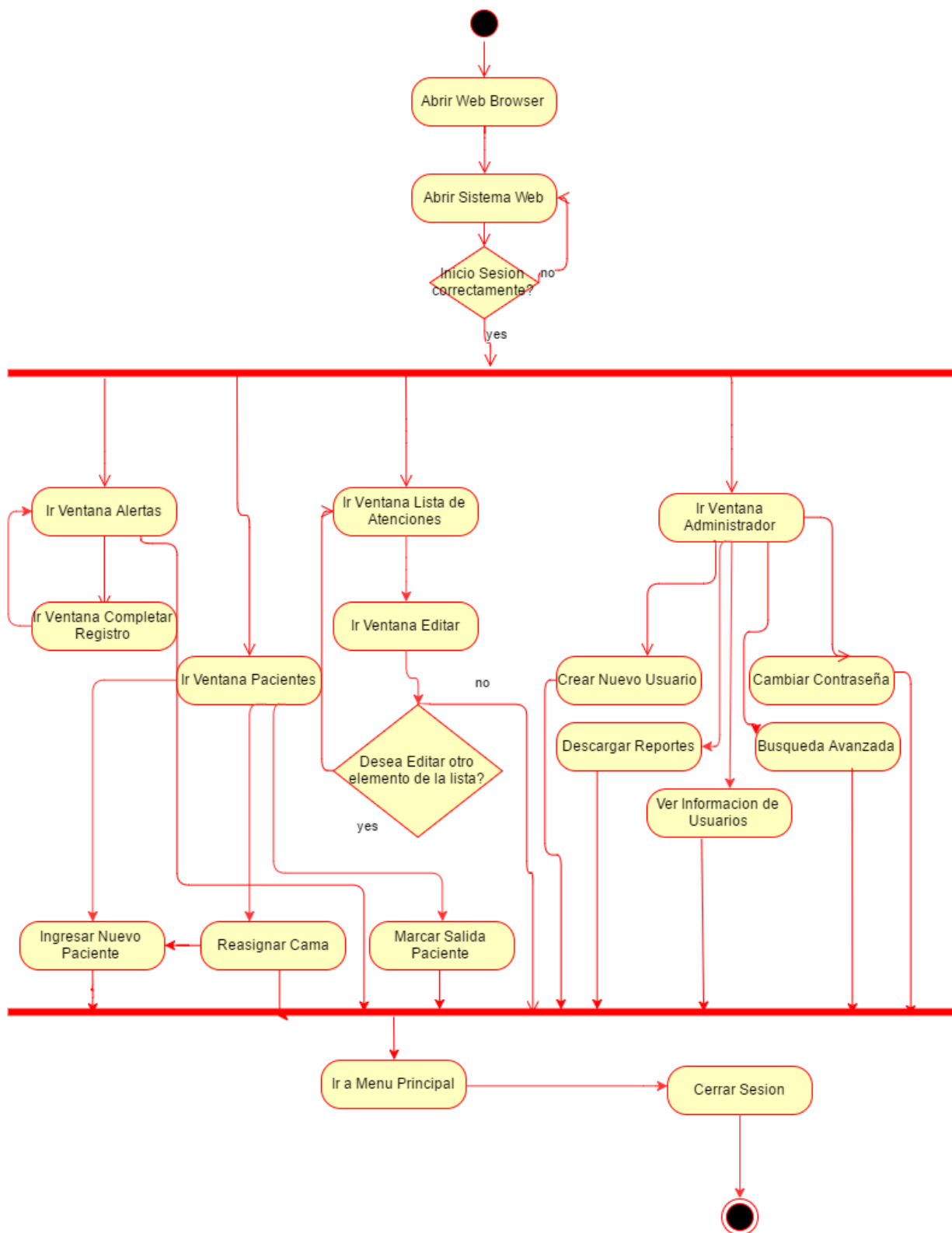


Figura 50. Diagrama de Actividades del sistema en general

Fuente: Elaboración propia

• **Flujo de Trabajo**

El flujo de trabajo es la manera en que se trabajará con el Sistema web en el ambiente real, para lo cual se desarrolló 3 diagramas de actividades para visualizar el flujo de trabajo de los usuarios del sistema (Ver Figuras 51, 52, 53)

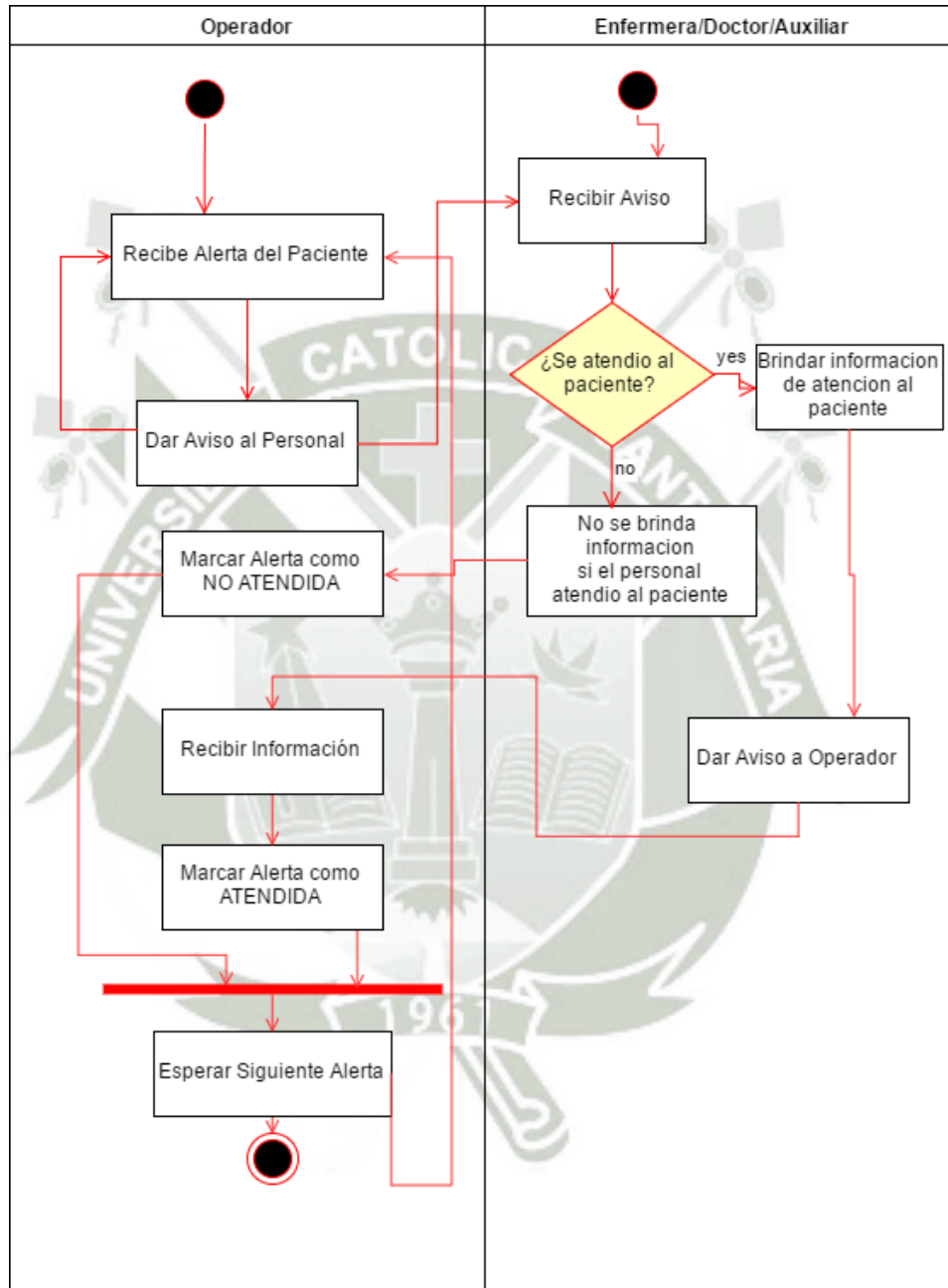


Figura 51. Diagrama de Actividades de manejo de alertas

Fuente: Elaboración propia

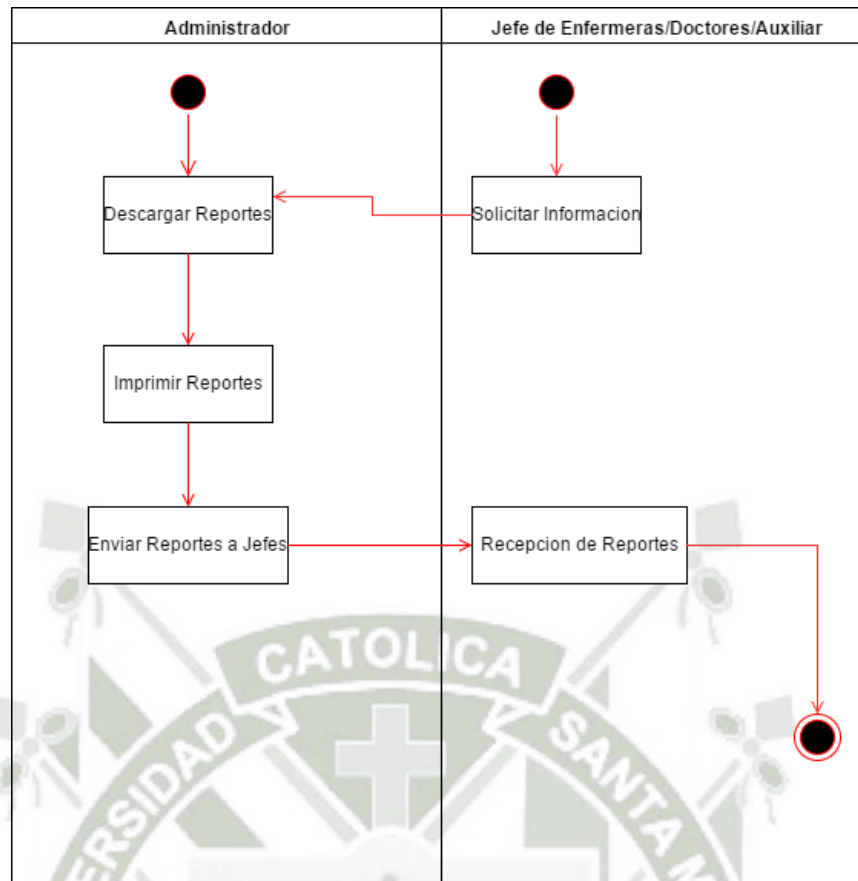


Figura 52. Diagrama de Actividades generación de reportes

Fuente: Elaboración propia

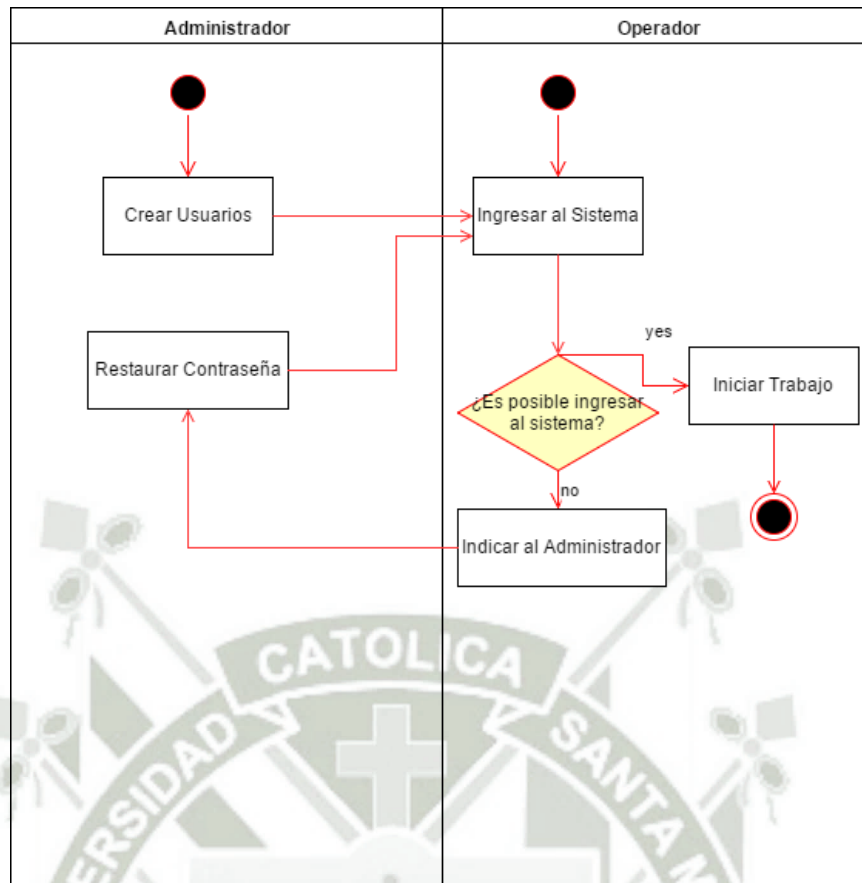


Figura 53. Diagrama de Actividades manejo de usuarios en el sistema

Fuente: Elaboración propia

Mockups del Sistema


	
	<p>Registro de Atención al Paciente Hospitalizado</p> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <p style="text-align: center;">LOGIN</p> <p style="text-align: center;">Usuario</p> <input style="width: 80%; margin-bottom: 5px;" type="text"/> <p style="text-align: center;">Contraseña</p> <input style="width: 80%; margin-bottom: 5px;" type="password"/> <p style="text-align: center; margin-top: 10px;">Iniciar Sesión</p> </div>

Figura 54. Ventana de Login

Fuente: Elaboración propia

VENTANAS DEL OPERADOR



<p>RAPH Registro de Atención al Paciente Hospitalizado</p>		<p>Hola, operador, ahora tiene acceso al sistema</p>
<p>Menú Principal</p>	<div style="border: 1px solid black; padding: 20px; width: 80%; margin: 0 auto;">  </div> <p>Video de Introducción y Tutorial</p>	<p>El Sistema de Registro de Atención al Paciente Hospitalizado es el encargado de brindar reportes y resultados en mutua comunicación con las cámaras Raspberry Pi instaladas en las camas de los pacientes hospitalizados con el fin de mejorar la atención, facilitar la vigilancia al personal médico y el pedido de atención del paciente</p>
<p>Alertas</p>		
<p>Lista de Atención</p>		
<p>Cerrar Sesión</p>		

Figura 55. Ventana de Menú Principal

Fuente: Elaboración propia


RAPH Registro de Atención al Paciente Hospitalizado				Hola, operador, ahora tiene acceso al sistema	
Menú Principal Alertas Lista de Atención Cerrar Sesión	ALERTAS DE PACIENTES				
	Nro.	Información	Estado		
	1	Cama 15// 30/05/16 7:35 pm	Atendido	No Atendido	
	2	Cama 14// 30/05/16 7:01 pm	Atendido	No Atendido	
	3	Cama 25// 30/05/16 6:25 pm	Atendido	No Atendido	
4	Cama 06// 30/05/16 6:15 pm	Atendido	No Atendido		

Figura 56. Ventana de Alertas

Fuente: Elaboración propia




RAPH Registro de Atención al Paciente Hospitalizado				Hola, operador, ahora tiene acceso al sistema	
Menú Principal Alertas Lista de Atención Cerrar Sesión	Registrar Alerta				
	Seleccione el Medico o Enfermera que atendió al paciente:				
	Dr. Miguel Adolfo Jiménez Barriga 				
	Agregue una descripción o acontecimientos importantes de la atención				
	<input type="button" value="Registrar Atención"/>				
Si desea descargar una constancia de atención haga click aquí					

Figura 57. Ventana de Editar/Registrar Alertas

Fuente: Elaboración propia

RAPH
Registro de
Atención al
Paciente
Hospitalizado



EsSalud
MAS SALUD PARA MAS PERUANOS

Hola, operador, ahora tiene acceso al sistema

Menú Principal

Lista de Registros de Atención


Buscar:

Id	Fecha-Tiempo	Personal Medico	Nro Cama	Estado	Opciones
1	14/05/16 07:00 pm	Dr Alfonso Cornejo Fuentes	15	Atendido	Editar
2	14/05/16 06:50 pm	Dr Alfonso Cornejo Fuentes	13	No Atendido	Editar
3	14/05/16 06:40 pm	Dr Alfonso Cornejo Fuentes	05	Atendido	Editar
4	14/05/16 06:30 pm	Dr Alfonso Cornejo Fuentes	09	Atendido	Editar

Figura 58. Ventana de Lista de Atenciones

Fuente: Elaboración propia

RAPH
Registro de
Atención al
Paciente
Hospitalizado



EsSalud
MAS SALUD PARA MAS PERUANOS

Hola, operador, ahora tiene acceso al sistema

Menú Principal

Editar Alerta

Seleccione el Medico o Enfermera que atendió al paciente:

Dr. Miguel Adolfo Jiménez Barriga
▾

Agregue una descripción o acontecimientos importantes de la atención

Editar Atención

Si desea descargar una constancia de atención haga [click aquí](#)

Figura 59. Ventana de Edición de Alerta

Fuente: Elaboración propia

VENTANAS DE ADMINISTRADOR

<p>RAPH Registro de Atención al Paciente Hospitalizado</p>			<p>Hola, <u>administrador</u>, ahora tiene acceso al sistema</p>
<p>Menú Principal</p>		<p>Video de Introducción y Tutorial</p>	<p>El Sistema de Registro de Atención al Paciente Hospitalizado es el encargado de brindar reportes y resultados en mutua comunicación con las cámaras Raspberry Pi instaladas en las camas de los pacientes hospitalizados con el fin de mejorar la atención, facilitar la vigilancia al personal médico y el pedido de atención del paciente</p>
Alertas			
Lista de Atención			
Administrador			
Cerrar Sesión			

Figura 60. Ventana de Menú Principal


Fuente: Elaboración propia

<p>RAPH Registro de Atención al Paciente Hospitalizado</p>			<p>Hola, <u>administrador</u>, ahora tiene acceso al sistema</p>										
<p>Menú Principal</p>		<p align="center">ALERTAS DE PACIENTES</p> <table border="1"> <thead> <tr> <th align="center">Nro.</th> <th align="center">Información</th> </tr> </thead> <tbody> <tr> <td align="center">1</td> <td align="center">Cama 15// 30/05/16 7:35 pm</td> </tr> <tr> <td align="center">2</td> <td align="center">Cama 14// 30/05/16 7:01 pm</td> </tr> <tr> <td align="center">3</td> <td align="center">Cama 25// 30/05/16 6:25 pm</td> </tr> <tr> <td align="center">4</td> <td align="center">Cama 06// 30/05/16 6:15 pm</td> </tr> </tbody> </table>		Nro.	Información	1	Cama 15// 30/05/16 7:35 pm	2	Cama 14// 30/05/16 7:01 pm	3	Cama 25// 30/05/16 6:25 pm	4	Cama 06// 30/05/16 6:15 pm
Nro.	Información												
1	Cama 15// 30/05/16 7:35 pm												
2	Cama 14// 30/05/16 7:01 pm												
3	Cama 25// 30/05/16 6:25 pm												
4	Cama 06// 30/05/16 6:15 pm												
Alertas													
Lista de Atención													
Administrador													
Cerrar Sesión													

Figura 61. Ventana de Alertas Administrador

Fuente: Elaboración propia

RAPH
Registro de
Atención al
Paciente
Hospitalizado



EsSalud
MAS SALUD PARA MAS PERUANOS

Hola, administrador, ahora tiene
acceso al sistema

**Menú
Principal**

Alertas

Lista de Atención

Administrador

Cerrar Sesión

Lista de Registros de Atención


Buscar:

Id	Fecha-Tiempo	Personal Medico	Nro Cama	Estado	Opciones
1	14/05/16 07:00 pm	Dr Alfonso Cornejo Fuentes	15	Atendido	Ver
2	14/05/16 06:50 pm	Dr Alfonso Cornejo Fuentes	13	No Atendido	Ver
3	14/05/16 06:40 pm	Dr Alfonso Cornejo Fuentes	05	Atendido	Ver
4	14/05/16 06:30 pm	Dr Alfonso Cornejo Fuentes	09	Atendido	Ver

Figura 62. Ventana de Lista de Atenciones (Administrador)

Fuente: Elaboración propia

RAPH
Registro de
Atención al
Paciente
Hospitalizado



EsSalud
MAS SALUD PARA MAS PERUANOS

Hola, administrador, ahora tiene
acceso al sistema

**Menú
Principal**

Alertas

Lista de Atención

Administrador

Cerrar Sesión

Editar Alerta

Médico o Enfermera seleccionada que atendió al paciente:

Dr. Miguel Adolfo Jiménez Barriga

Descripción de acontecimientos importantes de la atención

Se editó esta alerta 2 veces: Usuario1=1 Usuario2=1

Si desea descargar una constancia de atención haga [click aquí](#)

Figura 63. Ventana de Ver Alerta

Fuente: Elaboración propia

RAPH Registro de Atención al Paciente Hospitalizado		Hola, administrador, ahora tiene acceso al sistema
Menú Principal	Administración del Sistema	
Alertas	<input type="button" value="Generar Reportes"/>	<input type="button" value="Crear Usuario"/>
Lista de Atención	<input type="button" value="Búsqueda Avanzada"/>	<input type="button" value="Restablecer una Contraseña"/>
Administrador	<input type="button" value="Información de Usuario"/>	
Cerrar Sesión	<p>La ventana de administración es para poder controlar de mejor manera el sistema, adquiriendo permisos mayores a los operarios, verificando sus acciones, creando reportes, creando nuevos usuarios, etc.</p>	

Figura 64. Ventana de Administrador

Fuente: Elaboración propia

RAPH Registro de Atención al Paciente Hospitalizado		Hola, administrador, ahora tiene acceso al sistema
Menú Principal	Generar Reportes del Sistema	
Alertas	Ingrese el tipo de reporte: <input type="text" value="Reporte Diario//Reporte Mensual"/>	<input type="button" value="▼"/>
Lista de Atención	Ingrese Día <input type="text"/>	Ingrese Mes <input type="text"/>
Administrador	<div style="border: 1px solid gray; padding: 5px;"> <p style="text-align: center;">Atenciones</p> <p style="text-align: center;">■ Atendidas ■ No Atendidas</p> </div>	<p>Hubo un total de # atenciones</p> <p>Miembro del Personal que atendió más pacientes: <u>Enf. Beatriz Cornejo Zeballos</u></p>
Cerrar Sesión	<p>El reporte se <u>descargara</u> en formato PDF</p> <input type="button" value="Descargar Reporte"/>	

Figura 65. Ventana de Generar Reportes

Fuente: Elaboración propia


<p>RAPH Registro de Atención al Paciente Hospitalizado</p>		 <p>Hola, administrador, ahora tiene acceso al sistema</p>	
<p>Menú Principal</p>		<p>Crear Nuevo Usuario</p>	
Alertas		Ingrese un <u>username</u> :	
Lista de Atención		Ingrese un <u>Password</u> :	
Administrador		Ingrese nombres:	
Cerrar Sesión		Ingrese apellidos:	
		Ingrese un email:	
		¿El usuario es administrador? <input type="checkbox"/>	
		<p>Crear Nuevo Usuario</p>	

Figura 66. Ventana de Crear Usuario

Fuente: Elaboración propia


<p>RAPH Registro de Atención al Paciente Hospitalizado</p>		 <p>Hola, administrador, ahora tiene acceso al sistema</p>	
<p>Menú Principal</p>		<p>Búsqueda Avanzada del Sistema</p>	
Alertas		Ingrese ID del Registro: <input type="text" value="205"/>	
Lista de Atención		<p>Tiempo de la Atención: 31/05/16 11:50 am Estado de la Atención: ATENDIDO Cama de la Atención: Cama #15 Paciente de la Atención: Ignacio Cuevas Rondón Área de la Atención: Cirugía Personal de la Atención: Dr. Miguel Moscoso Villalba Usuario de la Atención: Rodrigo Castro Hinojosa Nro de Ediciones: 3</p>	
Administrador		<p>Descripción:</p> <p>Hubo una pequeña complicación con el paciente debido a que no estuvo en reposo como se le indico, sin embargo, quedo fuera de peligro y se encuentra estable</p>	
Cerrar Sesión		<p>Usuarios que Editaron el Registro: User1 = 2 - User2 =1 Tiempo de la última Edición: 05/06/16 06:15 am</p> <p>Imprimir Información:</p>	

Figura 67. Ventana de Búsqueda Avanzada

Fuente: Elaboración propia


RAPH Registro de Atención al Paciente Hospitalizado				Hola, administrador, ahora tiene acceso al sistema	
Menú Principal		Restablecer Contraseña			
Alertas		Seleccione un Usuario:		<input type="text" value="Usuario1//José Gómez Mamani"/>	
Lista de Atención		Ingrese la nueva Contraseña:		<input type="password" value="*****"/>	
Administrador		Ingrese Nuevamente la Contraseña:		<input type="password" value="*****"/>	
Cerrar Sesión		<input type="button" value="Restablecer Contraseña"/>			

Figura 68. Ventana de Restablecer Contraseña

Fuente: Elaboración propia


RAPH Registro de Atención al Paciente Hospitalizado				Hola, administrador, ahora tiene acceso al sistema	
Menú Principal		Información de Usuario			
Alertas		Seleccione un Usuario:		<input type="text" value="Usuario1//José Gómez Mamani"/>	
Lista de Atención		Nombre de Usuario: José Apellidos de Usuario: Gómez Mamani Email: Josegomezmamani@gmail.com Activo: Si			
Administrador		Administrador: No Total de Registros Manejados: 1025 Total de Registros Editados: 21			
Cerrar Sesión					

Figura 69. Ventana de Información de Usuario

Fuente: Elaboración propia

Usuarios

Django ofrece una clase ya creada e incorporada en su instalación llamada User, esta clase viene con los datos de usuario para Login/Logout necesarios para cualquier aplicación, encriptación de password y permisos para usuarios; esto nos permite generar el usuario de operador y administrador que necesita la aplicación. El inicio de sesión será el mismo para ambos; sin embargo, al momento de entrar al sistema, habrá diversas opciones que estarán habilitadas y deshabilitadas.

- **Administrador**

El administrador se encargará de crear reportes, principalmente. Tendrá acceso a ventanas de la web desde las cuales puede realizar ediciones más avanzadas, borrar registros y controlar permisos a los operadores.

- **Operador**

El operador será el encargado de estar pendiente de las alertas generadas, ya que tendrá a su disposición el módulo de alertas, lista de atenciones, constancias y reportes; tendrá la posibilidad de editar también, pero solo algunos campos en los que puede haberse equivocado en el momento del registro.

- **Paciente**

El paciente hará uso de la cámara megapíxel del Raspberry Pi la cual será un usuario que envíe la data necesaria para que actúe el sistema; las señales o el exceso de movimiento serán los que activen el sistema en sí, así que el paciente es directamente un usuario, ya que interactúa con el sistema sin necesidad de estar operando un teclado y mouse para hacer uso de él, solo usando señales

- Jefe de Enfermeras/Jefe de Personal Médico

Los jefes de cada área serán los encargados de solicitar los reportes necesarios para tomar decisiones, ver el progreso, las fallas, y más información de importancia para los jefes de personal

Arquitectura de Raspberry Pi

Principalmente, de lo que se encargara el Raspberry es de funcionar como cámara que registra movimientos, obteniendo datos e información de las imágenes que capture, ¿cómo se hará eso posible?, por medio de la librería OpenCV la cual preprocesará y procesará las imágenes obtenidas por el módulo de cámara megapíxel para luego enviar la información al sistema. Todo realizado mediante scripts y configuración previa del dispositivo Raspberry Pi para que pueda realizar dichas tareas.

A continuación, un Diagrama de clases que representa el funcionamiento del script programado en el Raspberry Pi (Ver Figura 70).

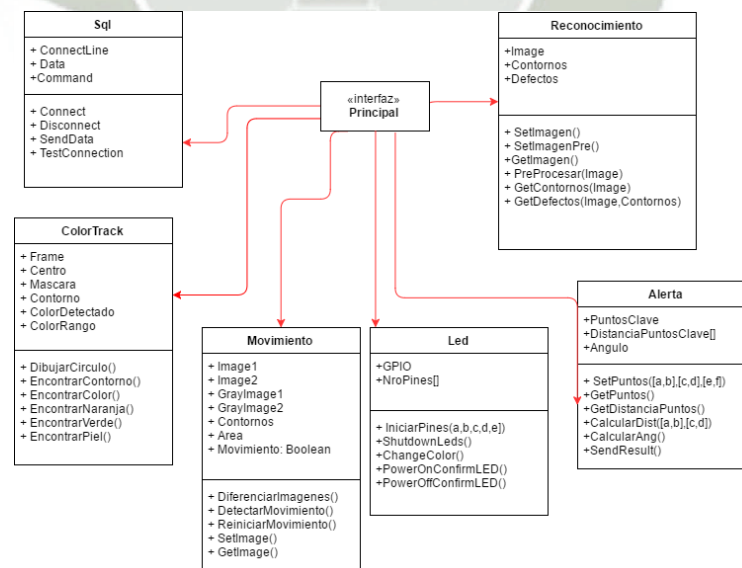


Figura 70. Diagrama de Clases Del módulo del Raspberry Pi

Fuente: Elaboración propia

- Clase Alerta: Esta clase será la mediadora para generar una alerta, para lo cual Principalmente, se constituye en 3 partes sus funciones:

- Solicitar puntos clave de la imagen
- Hacer uso de los cálculos necesarios para generar data
- Enviar los datos a la clase encargada de enviar la alerta a la base de datos

Más adelante se explicará cómo es que hará los cálculos correspondientes, por ahora tendrá como atributos lo que son:

- Puntos clave de la imagen obtenidos mediante el reconocedor
 - Un array de la distancia de todos esos puntos
 - Ángulo el cual se utilizará para confirmar la alerta
 - Un reconocedor que será de la clase Recognition
 - Un conector con la base de datos que será de la clase Sql
- Clase Recognition: Esta clase se encargará de reconocer los datos necesarios para que se puedan realizar cálculos matemáticos necesarios para extraer la información necesaria. Principalmente, aquí se hace uso del OpenCV ya que se encargará de extraer la imagen, preprocesarla, procesarla y generar puntos clave en las imágenes que se enviarán a la clase Alerta.
 - Clase Sql: Esta clase se encargará principalmente, de conectar con la base de datos en PostgreSQL desde el Raspberry Pi, registra la alerta para que posteriormente la página web pueda mostrarla en su pantalla. La clase Alerta tiene una variable SQL la cual recibirá la información ya lista para registrar en la base de datos.

Definición de Prototipo

Primeramente, para el desarrollo del prototipo, se debe realizar la parte más importante para cumplir con el objetivo del proyecto, que es el reconocimiento de señales de la mano; signos que el Raspberry Pi pueda identificar y usarlos en código,

Para esto se tiene que diseñar un conjunto de signos el cual nuestro sistema pueda reconocer y enviar alertas al sistema. Como se ve en el anterior punto, hay un script de OpenCV que reconoce algunos básicos gestos, este script puede ser utilizado como base para el módulo principal del proyecto. Además, se debe asociar con la página web los datos que recopile el Raspberry Pi con el módulo de Cámara Megapíxel.

Módulo Raspberry Pi

- **Objetivo**

El objetivo principal que tiene el Raspberry Pi es el siguiente:

- Identificar señales del paciente

Para lograr esto básicamente se debe lograr algunos objetivos específicos:

- Reconocer el movimiento del paciente
- Reconocer la señal que está realizando
- Convertir esa señal en datos
- Enviar los datos recopilados a la página web

- **Planteamiento**

Se plantea adaptar el algoritmo de reconocimiento de señales al proyecto con la finalidad de cumplir con los objetivos; una vez adaptado, debe incluir un

algoritmo de reconocimiento de movimiento el cual indique que el paciente quiere hacer un pedido de atención.

Una vez reconocida la señal se debe convertir en datos que se puedan enviar a la página web; para esto, en la página web se muestran las peticiones que se acaban de crear en la base de datos con columnas estándar, y se completan los datos del registro. Principalmente, se busca que el módulo de Raspberry envíe la información a la página web.

El Raspberry Pi tiene un adaptador Wifi por el cual es capaz de enviar la información sin necesidad que esté conectada más que al cable de poder para su funcionamiento.

Primeramente, se debe analizar las piezas del hardware que se necesita; posteriormente, configurar como es debido el Raspberry Pi y, al final, empezar a programar con los algoritmos y herramientas que ya se mencionaron previamente.

- Prueba de Hardware

El hardware Raspberry Pi es de bajo costo, pero bastante eficaz. Lo que es más importante del hardware que se utiliza es el módulo de cámara Megapíxel del Raspberry Pi. Para que en programación sea utilizado, es necesario instalar el módulo Pi Camera; para esto, en la consola se ejecuta el siguiente comando:

```
sudo apt-get install python-picamera
```

Para probar que importó la librería instalada y funciona se ejecuta el siguiente código, que es como un “Hola Mundo” en el uso de la PiCamera

```
import time
import picamera
```

```
with picamera.PiCamera() as picam:
    picam.start_preview()
    time.sleep(10)
    picam.stop_preview()
    picam.close()
```

se guarda el archivo con extensión .py y se ejecuta con el comando: python <Nombre del archivo>.py. mostrará una ventana que abre el video que toma la Raspberry Pi durante 10 segundos y posteriormente cierra la ventana. Debido a que en este proyecto se trabajó con un código más avanzado, no se tomó fotos o capturas de la ejecución de este; más adelante se mostrará el funcionamiento de la cámara con fotos de la ejecución del algoritmo.

El hardware, probando esto, estaría completo para empezar a trabajar; pero se necesita de un elemento principal que es la pantalla, y para mayor comodidad un teclado y mouse.

La conexión de un teclado y mouse USB es posible, ya que como se vio anteriormente el Raspberry Pi contiene puertos USB; un televisor o monitor con entrada HDMI sería útil para poder visualizar el Raspberry Pi por pantalla, como se verá en la imagen, es un televisor AOC con entrada HDMI conectada al Raspberry Pi



Figura 71. Raspberry Pi 3 Modelo B para el proyecto

Fuente: Elaboración propia

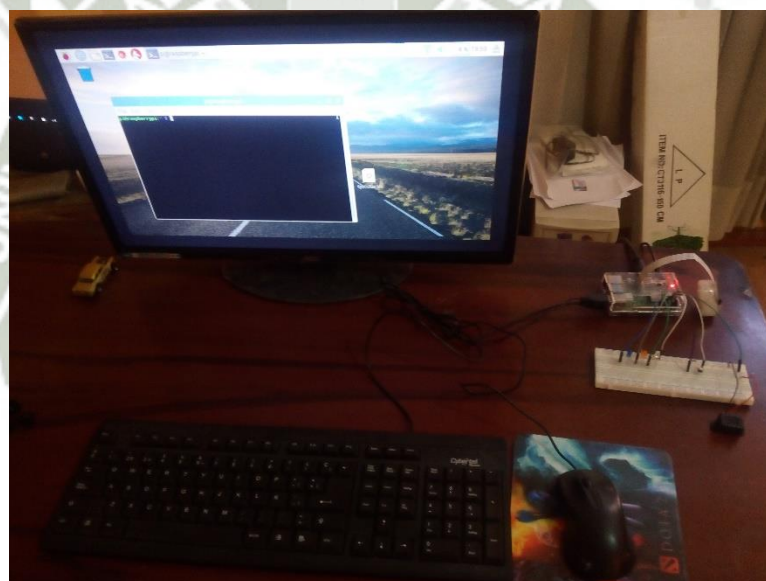


Figura 72. Componentes de Hardware para el Proyecto

Fuente: Elaboración propia

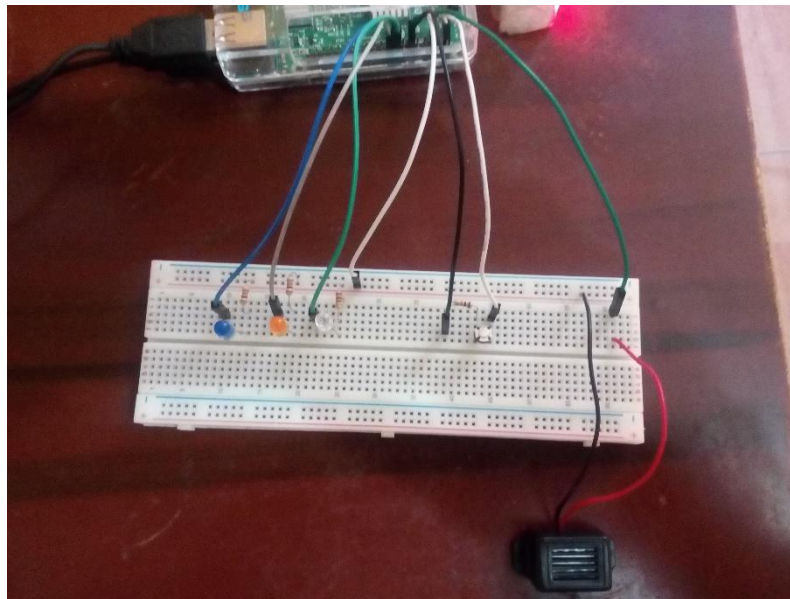


Figura 73. Protoboard y componentes conectados

Fuente: Elaboración propia

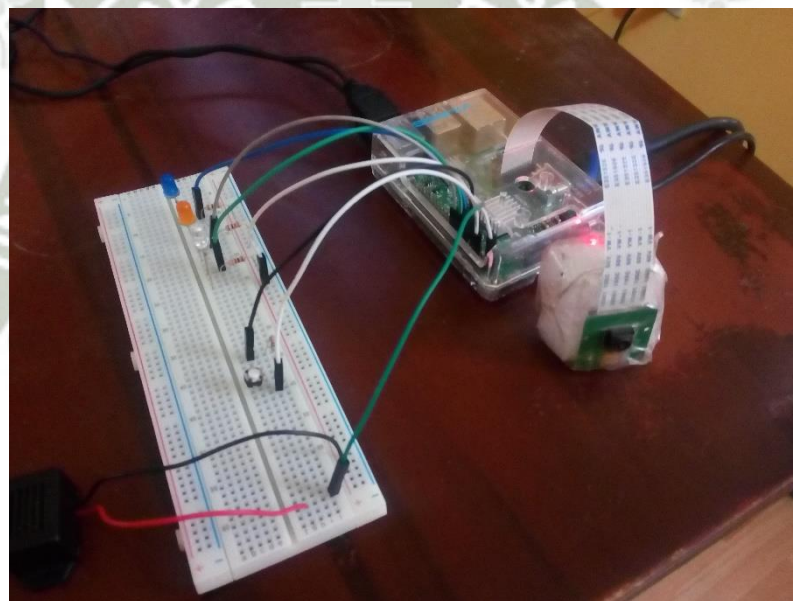


Figura 74. Placa de Raspberry Pi conectado

Fuente: Elaboración propia

Se conectó un protoboard con sus respectivos cables de conexión con el GPIO del Raspberry Pi, en los cuales también se conectó los leds, botón y buzzer para el proyecto, todo controlado desde Python.

Cuando la cámara del Raspberry Pi está en funcionamiento, emite una luz roja de un extremo indicando que está en uso por algún software.

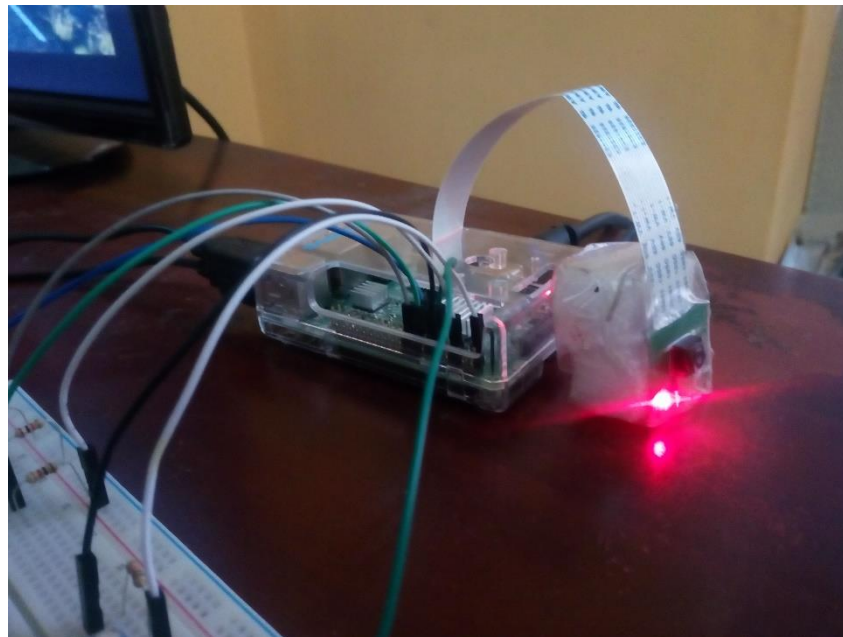


Figura 75. Cámara Megapíxel Prendida

Fuente: Elaboración propia

Para probar que los elementos del Hardware están en un 100% de capacidad, se procede a enchufar el Raspberry Pi con los dispositivos conectados y comprobar su funcionamiento.

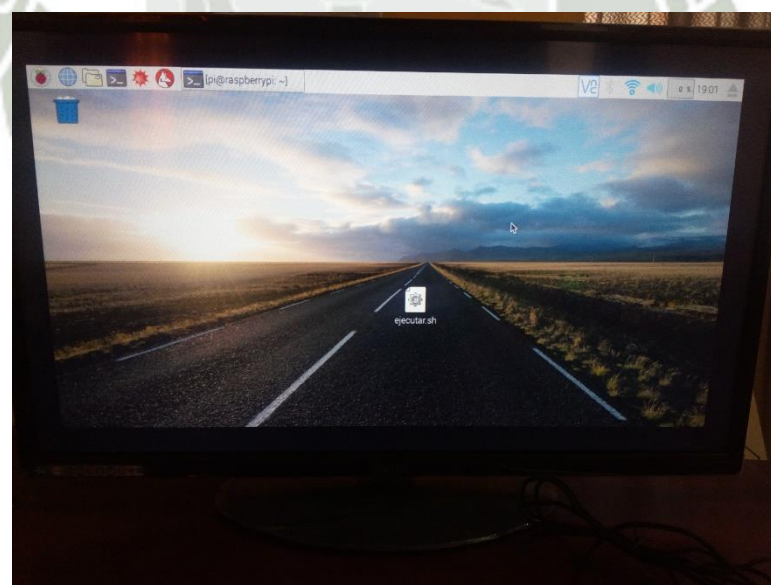


Figura 76. Prueba de Hardware Prendido

Fuente: Elaboración propia

En la siguiente imagen, se puede observar que la conexión Wifi funciona a la perfección ingresado a la red local.

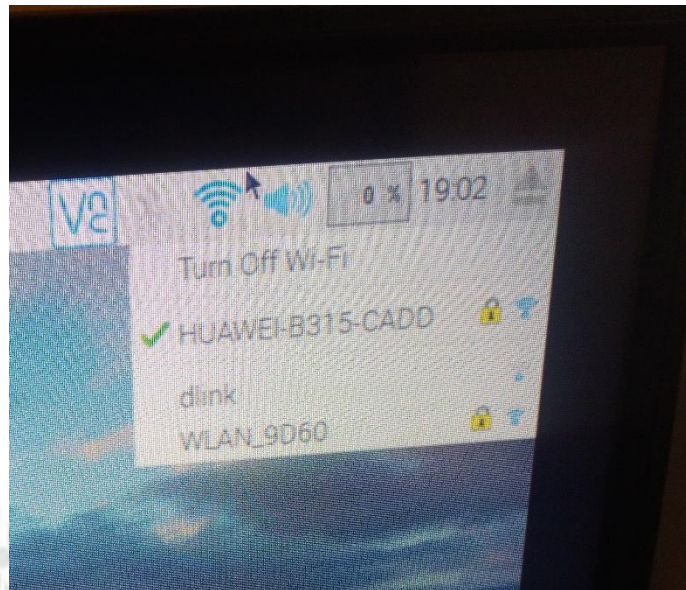


Figura 77. Prueba de Conexión Wifi conectado a la red local

Fuente: Elaboración propia

- Creación del Algoritmo

El algoritmo planteado para el Raspberry Pi se puede ver en el siguiente pseudocódigo:

```

Proceso Main
  PiCámara<-RaspberryPiCam
  F<-0
  Mensaje <-""
  Time <-Null
  TimeDelta <-Null
  Patron1<-Patron_Reconocimiento()
  Patron2<-Patron_Movimiento()
  Patron3<-Patron_Color()
  Repetir
    Si Patron3.EncontroColor==True Entonces
      Si Patron2.Movimiento==True Entonces
        Patron1.Preprocesar(PiCámara.Frame)
        Patron1.BuscarContornos
        Patron1.BuscarDefectos
        Nro_Defectos <- 0
        Para x<-0 Hasta Patron1.DEFECTOS Con Paso 1 Hacer
          Patron3<-CalculoMatematico()
          Patron3.RecibirDatos(x)
          Patron3.CalcularDistancia()
          Patron3.CalcularAngulo()
          Si Patron3.CalcularAngulo()<=90 Entonces
            Nro_Defectos<-Nro_Defectos+1
          Fin Si
        Fin Para
        Mensaje <- GenerarMensaje(Nro_Defectos)
        Tiempo <- Time.Now()
      F<-1
  
```

```
MostrarVideo()
Sino
Si F==1 Entonces
    TiempoDelta <- Tiempo + 3
    F=0
Sino
    Si TiempoDelta==Tiempo Entonces
        Patron4 <- SqlClass()
        Patron4.EnviaDatos(Mensaje)
    Fin Si
Fin Si
Fin Si
Hasta Que Key=="Esc"
FinProceso
```

Se planteó que el Raspberry Pi, a manera de explicar el pseudocódigo de arriba, se divida en 6 clases Principalmente, :

- SqlClass
- AlertaClass
- ReconocimientoClass
- MovimientoClass
- LedClass
- ColorClass

Estas clases tienen las siguientes funciones,

1. SqlClass: Se encarga Principalmente, del envío de los datos recopilados hacia la base de datos del servidor, las funciones que tiene la clase sql son:
 - 1.1. EstablecerConexion(): Para establecer conexión con el servidor, retornará un True si estableció la conexión correctamente.
 - 1.2. EnviarDatos(): Envía los datos recopilados por el análisis de las otras clases creadas; retorna un True si se hizo correctamente el Insert en la base de datos, y False si hubo un error.

2. AlertaClass: Se encarga de hacer los cálculos matemáticos para definir si la señal registrada es válida, entre las funciones o métodos que contiene están:
 - 2.1. EnviarPuntos: Recibe los puntos clave de la imagen para tenerlos entre sus datos
 - 2.2. CalcularDistancia: Calcula la distancia entre 2 puntos
 - 2.3. CalcularAngulo: Calcula el ángulo con el teorema del coseno y convertirlo a grados
3. MovimientoClass: Principalmente, se encarga de analizar si hay o no movimiento en la Cámara; esto sirve para indicar si el paciente quiere enviar una alerta o no, entre sus funciones están:
 - 3.1. CompararÁrea: Compara el área de los contornos encontrados en la imagen, si el área es similar o igual retornará un False, indicando que no hubo movimiento, y True si es distinto, indicando lo contrario
 - 3.2. RecibirFrame: Recibe un frame del video para analizarlo
 - 3.3. PreProcesar: Preprocesa el frame adquirido
4. ReconocimientoClass: Se encarga del reconocimiento de señales en sí, indicando si la señal es válida, reconociendo diversos datos para procesarlos, los métodos principales que contiene son:
 - 4.1. Encontrar Contornos: Analiza los contornos que tiene la imagen, retorna True si encontró algunos, y False si no encontró o hubo algún error
 - 4.2. Encontrar Defectos: Analiza los contornos y busca defectos, retornando True si encontró defectos y un error si nos los encuentra, para el cual, retornará un False
 - 4.3. Pre procesar Imagen: Convierte la imagen a Gris, extrae saturación, brillo y limpia la imagen para el siguiente análisis, retorna True si se hizo

correctamente el pre procesamiento y False si hubo algún error (Vipul Sharma, 2014).

5. LedClass: Se encarga de indicar al paciente la señal que está siendo reconocida por el sistema
 - 5.1. Shutdown Leds: se encarga de apagar todos los leds
 - 5.2. Indicar Envío de Datos: Encenderá el Led que significará que se envió correctamente el Led
 - 5.3. ActivarLed: Activa el Led de la señal que está dando el paciente a la cámara
6. ColorClass: Se encarga de reconocer un color clave en la imagen la cual activará la detección
 - 6.1. FindColor: Se encarga de llamar a la función para detectar un color específico
 - 6.2. Find Green/Red/Blue/Yellow/Orange/Skin: Se encarga de indicar si se encontró o no un color indicado en la imagen
 - 6.3. Find Contourn: Encuentra los contornos en la imagen, resaltando así la figura que contiene el color especificado
 - 6.4. DrawCircle: Dibuja un círculo en el centro de la figura resaltada por los contornos.

Módulo Web de Registro y Alertas

- **Objetivo**

El objetivo principal que tiene el Sistema Web es:

- Manejar y controlar la información obtenida con el fin de mejorar la calidad de atención del centro de salud

Los objetivos específicos son los siguientes:

- Registrar correctamente las alertas que brinde el Raspberry Pi
- Facilitar el control de vigilancia hacia los pacientes
- Visualizar las alertas generadas por los pacientes
- Especificar permisos para los usuarios que hagan uso del sistema
- Analizar y evaluar el desempeño del personal para la atención a los pacientes
- Generar reportes sobre la información que se maneja

- **Planeamiento**

El objetivo de este módulo es generar un manejo más tradicional y eficaz de la información que se generará a través de los datos que nos brinde el Raspberry Pi.

Especialmente, se desarrolló para gestionar la información que se tenga, con datos de los pacientes, personal, y las camas ubicadas en las diferentes áreas de hospitalización del hospital; se puede generar incluso reportes, los cuales se verá más adelante.

El sistema web se dividió en diferentes partes para entender bien las funciones de cada módulo que se creó para la página web, los módulos son los siguientes:

- Alertas
- Manejo de Entrada y Salida de Pacientes

- Lista
- Administración
- Reportes

1. Alertas

El módulo de alertas se encarga principalmente, de mostrar las últimas alertas enviadas por el Raspberry Pi.

El Raspberry Pi como se mencionó anteriormente, crea en la tabla de Atenciones un nuevo registro llenando, únicamente, los campos de Estado, Tiempo, Nro de Cama y Mensaje de Atención.

Estos datos serán visualizados desde el módulo de alertas en el orden en que se generan, para así el usuario saber qué atención fue la primera que se generó y desde ahí empezar para abajo a registrar las alertas.

Primeramente, la tabla de Atención tiene los siguientes datos:

- Id
- Tiempo
- Estado
- Descripción
- IdUsuario
- IdPersonal
- IdCama
- MensajeAlerta

Este módulo tendrá 2 partes:

- Lista de alertas
- Edición y Registro de Alertas

- Lista de Alertas

Esta parte principalmente, será para mostrar la lista de las alertas que se generen en el sistema mediante el Raspberry Pi, además de brindar la opción de completar el registro de las alertas generadas; esto se hace posible mediante 2 botones que estarán junto a cada alerta del sistema, los cuales serán para modificar el estado de la alerta.

Cuando se crea una alerta, está definida con un estado False que significa que está sin atender. Los botones que se coloquen al costado de cada alerta ayudarán a indicar el estado al cual pasa la alerta; es decir, una vez que está la alerta creada y se muestra en esta pantalla de Lista de Alertas, estará por default con un estado de False indicando que no se atendió, pero el operador tendrá la opción de hacer clic en 2 botones de acuerdo con la atención brindada; estas opciones son de Atendido o No Atendido.

El botón de Atendido puede ser presionado cuando alguien del personal fue a atender al paciente de manera casi inmediata o a tiempo

El botón de No Atendido puede ser presionado cuando el personal llegó muy tarde al auxilio del paciente o simplemente no fue.

Esto dependerá del usuario que opere el sistema. Sin embargo, en caso el usuario busque ayudar al personal marcando siempre Atendido, se podrá comprobar mediante la tabla de Edición que se verá más adelante.

Cuando se presiona alguno de esos 2 botones para cada alerta, el sistema abre la siguiente ventana de Edición y Registro de Alertas de la cual se hablará en el siguiente punto. La pantalla de Lista de Atenciones se vería como la siguiente imagen (Ver Figura 78)

Registro de Atención de Pacientes Hospitalizados
R-A-P-H

Hola, alpha, ahora tiene acceso al sistema

Recuerde verificar siempre las alertas

Menu Principal

Alertas

Ver Lista de Atenciones

Administrador

Cerrar Sesión

ALERTAS EN TIEMPO REAL

Descripcion de Alerta	Opciones de Alerta
Cama Nro: 34 // July 3, 2016, 7:45 p.m.	<input checked="" type="checkbox"/> Atendido <input type="checkbox"/> No Atendido
Cama Nro: 2 // July 3, 2016, 7:45 p.m.	<input checked="" type="checkbox"/> Atendido <input type="checkbox"/> No Atendido

Figura 78. Pantalla de Alertas del Sistema Web

Fuente: Elaboración propia

- Edición y Registro de Alertas

En esta pantalla, se editará el registro seleccionado en la pantalla anterior para completar los siguientes campos:

- IdPersonal
- IdUsuario
- Descripción

Basicamente, se mostrará un Select el cual tendrá todos los miembros del personal del área de Hospitalización los cuales están a total disposición de los pacientes internos; entonces, el usuario podrá seleccionar ahí el personal que atendió al paciente, más abajo en la misma pantalla, tendrá un espacio donde podrá llenar una breve descripción de la atención de acuerdo con lo que el usuario vea por conveniente o considere cómo fue la atención. Además de esos datos, tendrá la posibilidad de generar una constancia de atención de acuerdo a la información de dicha alerta; es decir, si en el caso de que un paciente niegue que un miembro del personal haya ido atenderlo y si lo haya hecho, o un miembro del personal no atienda al paciente y este le diga que cambie el estado o información como si lo hubiera hecho, el usuario por medida de seguridad

opcional, puede imprimir una constancia generada por el sistema con los datos que este haya ingresado. Entonces en caso un paciente niegue atención o un miembro del personal pida una ayuda al usuario para evitarle problemas, puede generar una constancia y después cambiar los datos de la alerta actual. Sin embargo el usuario también será evaluado por los cambios que haga, ya que para demostrar que hubo un cambio en los datos de la alerta, la tabla Edición será la encargada de registrar los cambios que haga dicho usuario a la alerta actual; es decir, que un cambio hecho por el usuario es registrado, posteriormente se verá cómo es que el usuario puede editar, pero la pantalla para editar, será la misma, esto quiere decir, que el estado actual de una atención no se puede cambiar, si se marcó como atendida, no se podrá cambiar como no atendida o viceversa (Ver Figura 79).

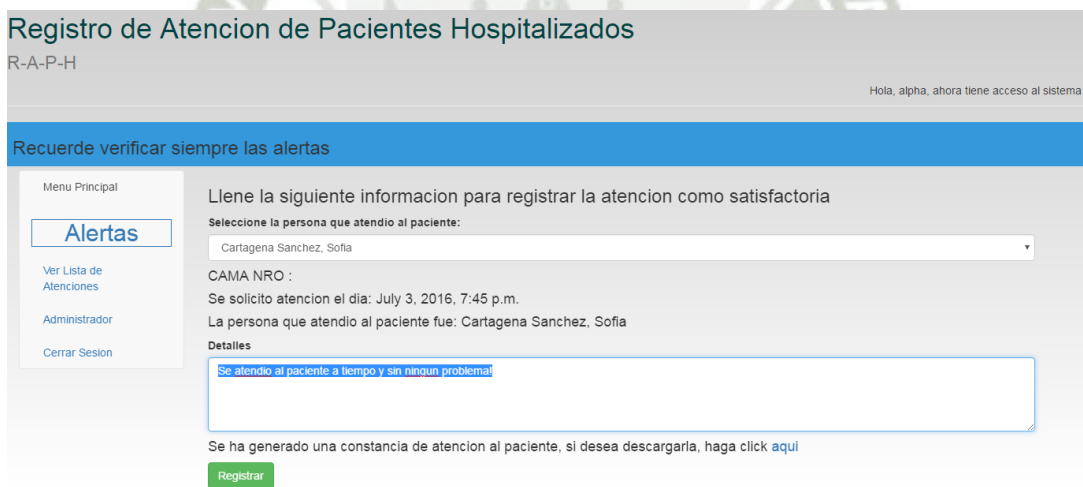


Figura 79. Pantalla de Edición de Alertas del Sistema Web

Fuente: Elaboración propia

2. Manejo de Entrada y Salida de Pacientes

Este sistema además de trabajar con las alertas y ayuda a mejorar la calidad del paciente, debe trabajar con otros datos que son afines a las alertas, los pacientes, es por eso que en el Diagrama de entidad-relación se colocó la tabla de

pacientes; los cuales tendrán registrados a los pacientes que están en ese momento hospitalizados, por lo que esa tabla tendrá datos de entrada y salida de los pacientes, para ser más exactos, las fechas de entrada y salida del paciente; pero esto lamentablemente no se puede controlar automáticamente con las cámaras de Raspberry Pi, esto tiene que ser por ingreso manual, por lo que este módulo deberá ser trabajado por el operador del sistema para controlar las fechas de entrada y salida del paciente hospitalizado. Primeramente, este módulo se dividirá en 5 partes:

- Ingreso de pacientes
- Modificación de Ingreso
- Asignación de Cama
- Visualización de los pacientes
- Salida del paciente.

El primer módulo será para registrar un nuevo paciente que será hospitalizado, un registro simple para el ingreso rápido del paciente.

El segundo módulo será de modificación, aquí se podrá reasignar a un paciente, para lo cual nos saldrá un módulo de selección del paciente; posteriormente nos mostrará el módulo de asignación de cama, que se explicará en el siguiente punto.

El módulo de ingreso principalmente, tendrá esas 2 partes, una en la que se ingresa el nombre del paciente completo y posteriormente la asignación de cama, ya que un paciente que es hospitalizado pasará a ocupar una de las camas del área de internado, por lo que se debe tener en el sistema un registro de que paciente está asignado a esa cama. Esto se desarrollará mediante un módulo que represente las camas disponibles en el sistema las cuales se dividirán por área. El

operador, deberá seleccionar un área, y actualizará automáticamente la vista de las camas disponibles en dicha área, solo bastará con arrastrar la imagen del paciente que se identificará con su DNI hacia una de las camas que aparecen disponibles (color verde) y nos aparecerá disponible el botón de Asignar Cama, se registrará en el sistema que dicho paciente está en esa cama.

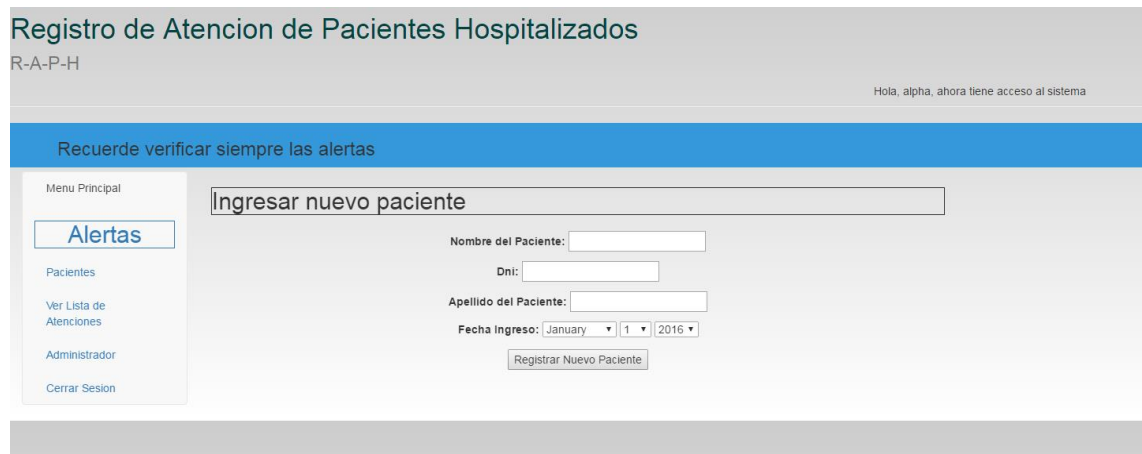
El módulo de Salida del Paciente simplemente será para registrar la salida del paciente que estaba hospitalizado, llenando el campo de TiempoSalida de la tabla Pacientes, el cual registrará la hora de salida del paciente desde el momento en que el usuario haga clic en el botón correspondiente. Este módulo, mostrará una lista de pacientes actuales en las camas, y el usuario podrá marcar uno de los pacientes mostrando un mensaje de confirmación, indicando así que ya abandonó el área de hospitalización, tomando el tiempo desde que el usuario marcó e hizo clic en el botón de Registrar Salidas, aparecerá una pantalla indicando que se registraron satisfactoriamente. En caso que se cometa un error, deberá contactar al administrador indicando que registró mal una salida, para que este pueda volver a asignarle la cama en la que estaba, ya que una vez que se registra la salida, la cama pasa a estar libre.

Las pantallas para el Manejo de Entrada y Salida de Pacientes serían las siguientes:



Figura 80. Pantalla del Menú de Manejo de Entrada y Salida de Pacientes

Fuente: Elaboración propia



Registro de Atención de Pacientes Hospitalizados
R-A-P-H

Hola, alpha, ahora tiene acceso al sistema

Recuerde verificar siempre las alertas

Menu Principal

- Alertas
- Pacientes
- Ver Lista de Atenciones
- Administrador
- Cerrar Sesión

Ingresar nuevo paciente

Nombre del Paciente:

Dni:


Apellido del Paciente:

Fecha Ingreso: January 1 2016

Registrar Nuevo Paciente

Figura 81. Pantalla de Ingreso de Paciente

Fuente: Elaboración propia



Registro de Atención de Pacientes Hospitalizados
R-A-P-H

Hola, alpha, ahora tiene acceso al sistema

Recuerde verificar siempre las alertas

Menu Principal

- Alertas
- Pacientes
- Ver Lista de Atenciones
- Administrador
- Cerrar Sesión

Reasignación del Paciente

Seleccione un paciente: Corrales Santana, Joaquin Antonio

Paciente seleccionado: Corrales Santana, Joaquin

Ver Camas Disponibles

Figura 82. Pantalla de Reasignación de Paciente

Fuente: Elaboración propia

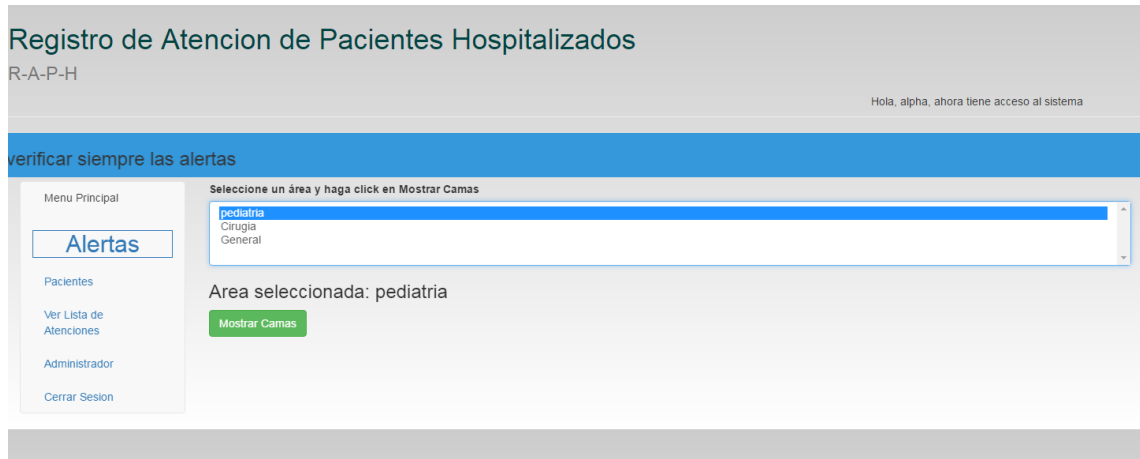


Figura 83. Pantalla de Selección de área

Fuente: Elaboración propia

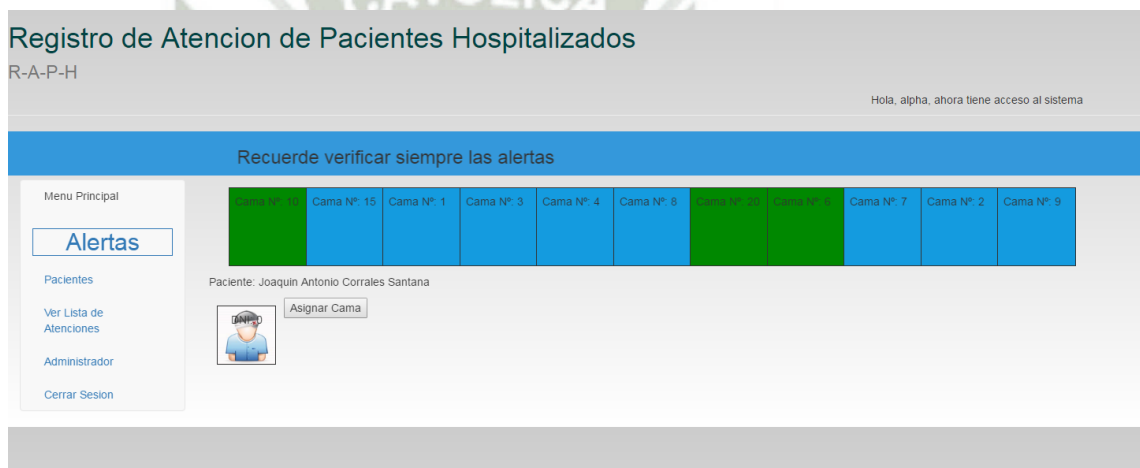


Figura 84. Pantalla de Asignación de Cama

Fuente: Elaboración propia



Figura 85. Pantalla de Asignación de Cama (Paciente asignado a una cama)

Fuente: Elaboración propia

Recuerde verificar siempre las alertas

Cama Asignada	Nombre del Paciente	Dni del Paciente	Indicar Salida
Cama Nro: 15	Pacori Santana, Joaquin Miguel	0	MARCAR SALIDA
Cama Nro: 1	Portugal Santana, Jose Carlos	0	MARCAR SALIDA
Cama Nro: 3	Sanchez Ramirez, Julian Jose	0	MARCAR SALIDA
Cama Nro: 4	Dominguez Santana, Sofia Alejandra	0	MARCAR SALIDA
Cama Nro: 8	Mamani Quispe, Pedro Miguel	0	MARCAR SALIDA
Cama Nro: 13	Linares Niebles, Santiago	55635186	MARCAR SALIDA
Cama Nro: 7	Sanchez Santana, Joaquin Pedro	0	MARCAR SALIDA
Cama Nro: 2	Espenser Brooke, Ally Stefania	0	MARCAR SALIDA
Cama Nro: 9	Corrales Santana, Joaquin Antonio	0	MARCAR SALIDA

Menu Principal

Alertas

Pacientes

Ver Lista de Atenciones

Administrador

Cerrar Sesión

Figura 86. Pantalla de Salida de Pacientes

Fuente: Elaboración propia



3. Lista

Esta parte del sistema es más simple, ya que solo es para visualizar las alertas que han llegado al sistema desde que inició; principalmente, muestran los siguientes datos en una lista:

- Id
- Nro. de Cama
- Descripción
- Tiempo de Creación
- Estado
- Link de “Editar”

Como se mencionó anteriormente, los usuarios podían tener la opción de editar las alertas que ya se registraron; todo esto mediante la pantalla de Edición y Registro de Alertas. Se puede recurrir a diversas formas para personalizar la Lista de Atenciones, filtrar por las columnas, ordenar ascendente o descendentemente buscar por palabras clave... La pantalla sería muy parecida a la siguiente (Ver Figura 87).

Recuerde verificar siempre las alertas

Menu Principal

Alertas

[Ver Lista de Atenciones](#)

[Administrador](#)

[Cerrar Sesión](#)

Show 10 entries

Search:

ID ▲	Nro Cama	Fecha	Detalles	Estado	Opciones
29	15	June 2, 2016, 4:04 p.m.	No hubo problemas y el paciente fue obediente a las indicaciones	Atendido	Ver o Editar
30	20	June 2, 2016, 4:06 p.m.	No hubo problemas y el paciente fue obediente a las indicaciones	Atendido	Ver o Editar
31	5	June 2, 2016, 4:06 p.m.	No hubo problemas y el paciente fue obediente a las indicaciones	No Atendido	Ver o Editar
32	34	June 2, 2016, 4:06 p.m.	No hubo problemas y el paciente fue obediente a las indicaciones	Atendido	Ver o Editar
34	34	June 2, 2016, 4:09 p.m.	SE ATENDIO AL PACIENTE DE MANERA CORRECTA	No Atendido	Ver o Editar
35	5	June 2, 2016, 4:10 p.m.	None	No Atendido	Ver o Editar
36	5	June 2, 2016, 4:10 p.m.	None	No Atendido	Ver o Editar
37	15	June 2, 2016, 4:10 p.m.	None	No Atendido	Ver o Editar
38	15	June 22, 2016, 12:34 p.m.	None	No Atendido	Ver o Editar
39	20	June 22, 2016, 12:34 p.m.	None	No Atendido	Ver o Editar

Showing 1 to 10 of 39 entries

Previous 1 2 3 4 Next

Figura 87. Pantalla de Lista de Alertas del Sistema Web

Fuente: Elaboración propia

4. Administración

Esta parte es una de las más importantes del sistema ya que se requiere de un administrador el cual puede ser el encargado del área de sistemas o alguien a fin a la carrera de Ing. de Sistemas. Es posible que algunos malos ingresos de datos tengan que ser modificados en la misma base de datos o se requiera de ciertas funciones que posee el sistema; ser ejecutadas por alguien que no esté involucrado directamente en el área de hospitalización. Algunas de estas funciones que se mencionan son las siguientes:

- Cambiar de contraseña
- Crear Nuevo usuario del Sistema
- Generar Reportes
- Realizar Búsquedas Avanzadas
- Ver información y Rendimiento de los Usuarios

Antes que nada, cuando un usuario es administrador del sistema, tiene mayores privilegios que otros usuarios, Principalmente, porque puede ver información más avanzada. Sin embargo, a pesar de ser administrador, no puede ejecutar las tareas destinadas a un operador como son el Registro y Edición de Alertas y el Manejo de Entrada y Salida de Pacientes. Puede visualizar estas ventanas, mas no operarlas o modificar (Ver Figura 88).



Figura 88. Pantalla de Administración del Sistema Web

Fuente: Elaboración propia

El módulo de cambio de contraseña es en el caso que un usuario no recuerde su contraseña, para esto podrá pedir ayuda al administrador del sistema para que pueda cambiar su contraseña (Ver Figura 89).

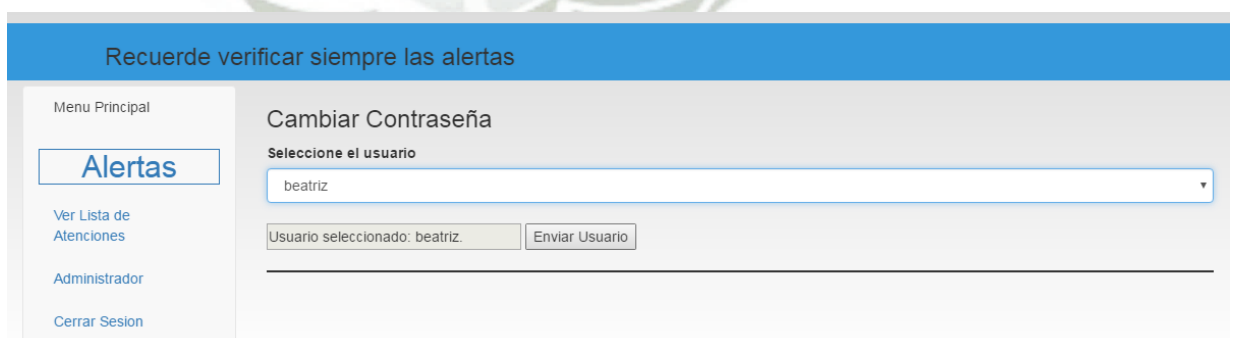


Figura 89. Pantalla de Cambiar de contraseña

Fuente: Elaboración propia

El siguiente módulo de creación de usuario es simple; solo tendrá que ingresar el nombre, apellidos, username, contraseña, indicar si será administrador o no y listo (Ver Figura 90).



Figura 90. Pantalla de Creación de Usuarios

Fuente: Elaboración propia

El módulo de reportes es más detallado por lo que será explicado en el siguiente punto:

El módulo de Realizar búsquedas avanzadas es más sencillo y a la vez eficaz, aunque puedes buscar mediante la Lista de Atenciones: este módulo permite buscar al administrador por medio del Id del Registro y mostrar información mucho más detallada de ese registro: proyectará datos asociados al personal que atendió, qué paciente fue, el total de ediciones que tuvo ese registro, quién las realizó, y los demás datos de la tabla de Atenciones (Ver Figura 91).

Recuerde verificar s

Menu Principal

Alertas

[Ver Lista de Atenciones](#)

[Administrador](#)

[Cerrar Sesión](#)

Tiempo de la Atencion:
June 30, 2016, 3:16 p.m.

Estado de la Atencion:
ATENDIDO

Nro de Cama:
7

Paciente Atendido:
Luis Ediberto, Gonzales Linares

Area de Atencion:
pediatria

Personal que atendio:
Sofia, Cartagena Sanchez

Usuario que registro atencion:
Jesus, Alpaca Rendon

Total de veces editado:
Total de veces editado:
1

Usuarios que Editaron:
, : 1 Ediciones al registro
Fue editado por ultima vez:
July 1, 2016, 3:33 p.m.

Descripcion:

Se atendio correctamente al paciente

Figura 91. Pantalla de Búsqueda Avanzada de Registros

Fuente: Elaboración propia

El último módulo de Información de Usuario, mostrará los datos del usuario seleccionado en una etiqueta Select de la pantalla; además de un botón llamado Estadísticas de Usuario, el cual nos brindará un reporte autogenerado sobre el desempeño del usuario hasta la fecha, pero se verá más a detalle esto en el punto de Reportes (Ver Figura 92).

Recuerde verificar siempre las alertas

Ver Información de Usuario

Seleccione el usuario y haga click en Ver Información

Nombre de Usuario:

Apellido de Usuario:

Email de Usuario:

¿Usuario Activo?:

¿Administrador?:

Si desea ver como se desempeño el usuario haga click aqui

Figura 92. Pantalla de Información de Usuarios

Fuente: Elaboración propia

5. Reportes

En la parte de reportes, se diseñó para que se maneje información más compleja que ayudará a altos mandos a tomar buenas decisiones y ver el rendimiento del personal que trabaja en el área de hospitalización. Para esto se estandarizó que los reportes fueran en pdf, ya que son documentos especiales de lectura y la programación en Python para pdf es sencilla, aunque con bastantes líneas de código, pero es efectiva.

Principalmente, en el sistema se considera que hay unos 3 reportes los cuales son:

- Reporte Diario
- Reporte Mensual
- Constancia de Atención

El primer reporte diario es el encargado de mostrar un resumen de las atenciones que se dieron durante el día; una lista de las atenciones para ser más específico y mostrar ahí los datos resaltantes que se quieren ver como cuántas atenciones hubo en total, quienes las ejecutaron, etc.

Principalmente, para este reporte solo se trabaja con la tabla Atenciones, ya que es un reporte simple y estructurado. A continuación, se muestra un prototipo del informe (Ver Figura 93).

RAPH

Registro de Atención al Paciente Hospitalizado



REPORTE DEL DIA

Nombre de Usuario: Jesus Alpaca Rendón
 Fecha de Emisión: 09/05/2016
 Cantidad de Atenciones: 14

Tabla de Atenciones:

A = Atendido
 N = No Atendido


Nro	Cama	Hora	Personal	Descripción	Ediciones	Estado
5	12	15:25	Dr. Juan Magallanes Deposito	Atencion realizada Atencion sin problemas	1	A
6	16	05:25	Enf. Santa Milagros	El paciente indica que no se le atendio correctamente	1	N
7	20	21:15	Dr. Juan Magallanes Deposito	Atencion realizada Atencion sin problemas	2	A
8	05	18:40	Dr. Juan Magallanes Deposito	Atencion realizada Atencion sin problemas	2	A
9	08	12:15	Enf. Santa Milagros	Atencion realizada Atencion sin problemas	1	A
10	12	12:15	Enf. Santa Milagros	La enfermera indica que el paciente no se dejo atender correctamente	1	N
11	36	12:20	Enf. Santa Milagros	Atencion realizada Atencion sin problemas	1	A

Figura 93. Mockup de Reporte Diario del Sistema

Fuente: Elaboración propia

El reporte diario no tiene muchos datos de interés ya que solo se analiza información diaria, no se puede extraer mucha información que pueda derivarse a conocimiento aquí. Sin embargo, un control del trabajo del personal en cuanto al día estaría bien visto por los jefes de cada cuerpo de personal (Enfermeras, Médicos). El reporte Diario se vería de esta manera en la web, más adelante se explica en qué pantalla se genera el reporte; por el momento, este es el pdf descargable que genera la página web (Ver Figura 94).

RAPH
Registro de Atención de Pacientes Hospitalizados



REPORTE DEL DIA

Nombre de Usuario: alpha

Fecha de Emision: 2016-06-30

Cantidad de Atenciones: 2

Tabla de Atenciones

Atendidos = A
No Atendidos = N

Nro	Cama	Hora	Personal	Nro Ediciones	Estado
51	7	15:16:49	No se registro nadie	0	N
52	8	15:53:39	No se registro nadie	0	N

Reporte del Dia: 30-06-16 - Registro de Atención de Pacientes Hospitalizados

Figura 94. Pantalla de Reporte diario del Sistema

Fuente: Elaboración propia

El reporte Mensual es mucho más complejo, puesto que extrae data de todas las tablas del sistema. A continuación, se presenta un prototipo del informe (Ver Figura 95).

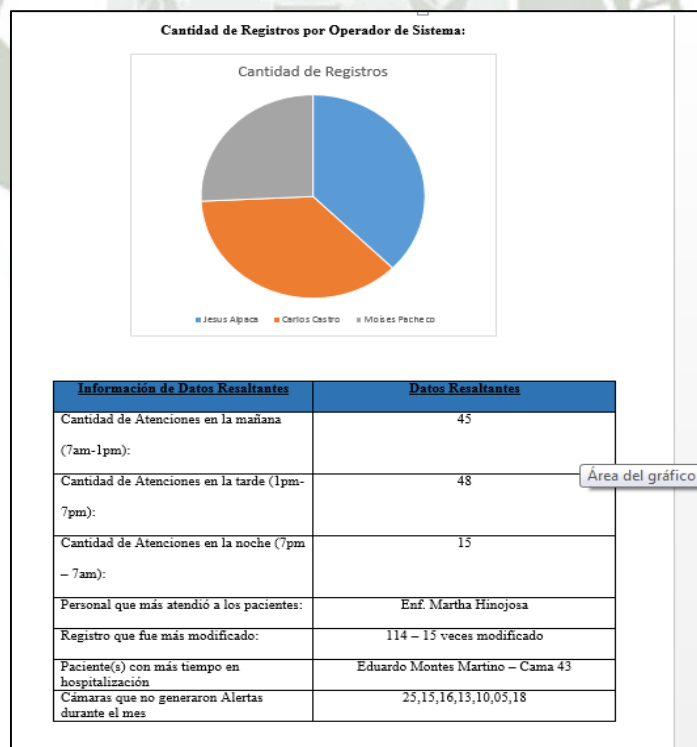
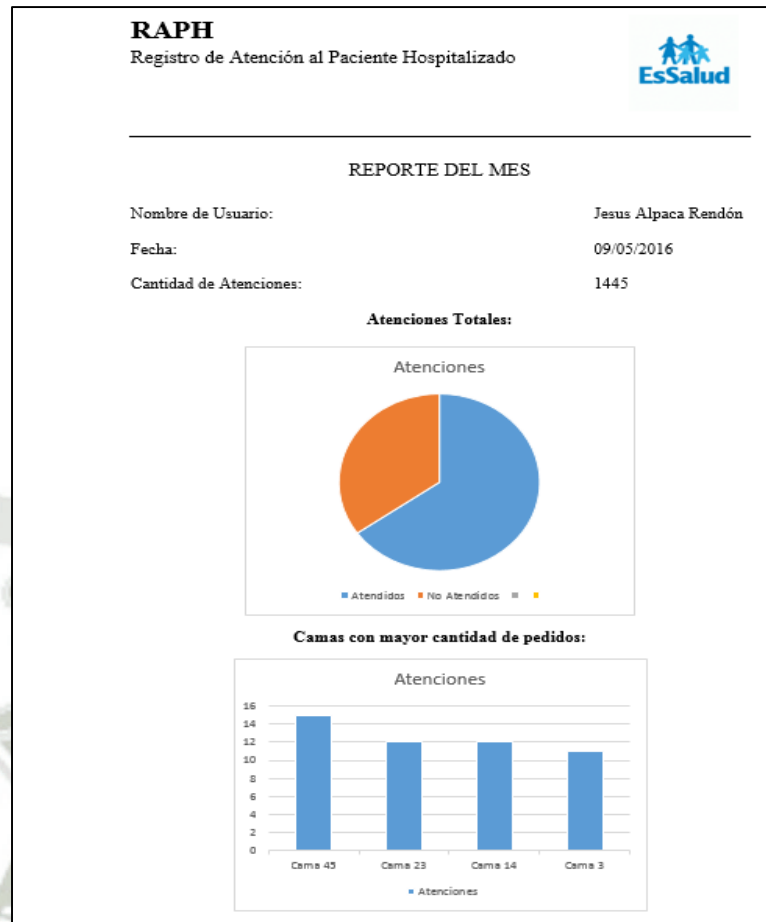


Figura 95. Mockup de Reporte Mensual del Sistema

Fuente: Elaboración propia

Como se vio, no solo se está evaluando datos de una tabla; se está sacando información conjunta de las tablas para generar conocimiento, siendo más exactos, se trabaja con las siguientes tablas:

- Tabla Atención
- Tabla Personal
- Tabla Edición
- Tabla Cama
- Tabla Usuarios

Las tablas de Paciente y Área aún no son consideradas ya que la tabla de pacientes es muy variada; los pacientes en promedio no pasan más de 5 días en hospitalización, excepto casos extremos, pero la mayoría, sí tiene un tiempo de estadía menor a esa cantidad o muy parecida.

La tabla Área solo es para identificar a qué área de hospitalización pertenece una cama, como un indicador simplemente, la cual es más utilizada en el módulo de entrada y salida de pacientes; para los reportes no ayudaría mucho, cada cama tiene un número único que la idéntica, es mejor trabajar con eso que con el área, posiblemente a futuro se puede incluir en el informe dependiendo de la escala que tenga el software.

En la parte baja, se mostrará una tabla con las atenciones brindadas por parte del personal. La pantalla de Reportes solo está disponible para el administrador y se puede apreciar de la siguiente manera (Ver Figura 96).

Registro de Atención de Pacientes Hospitalizados

R-A-P-H

Hola, alpha, ahora tiene acceso al sistema[Cerrar Sesión]

Recuerde verificar siempre



Figura 96. Pantalla principal de Reportes

Fuente: Elaboración propia

Para generar un reporte, vamos a la sección de Reportes de Administración y, como se ve en la imagen, se selecciona qué tipo de reporte se requiere, Reporte Diario o Reporte Mensual. Una vez que se selecciona uno de ellos, en la pantalla nos muestra un pequeño gráfico indicando el total de atenciones que se realizó, dividiéndolas en la cantidad de atendidas y no atendidas; es decir, por sus estados guardados.

Los reportes en la web generan un pdf que se descarga al hacer clic en el botón de descargar reportes; el Reporte Mensual se ve de esta manera por el momento (Ver Figura 97).

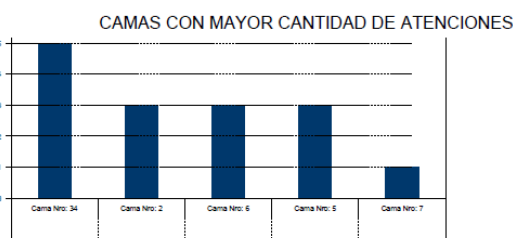
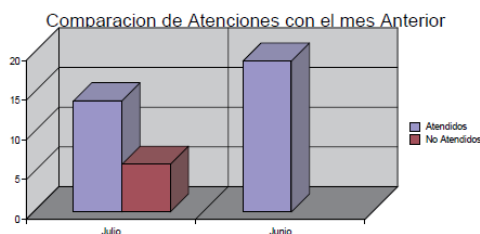
RAPH
Registro de Atención de Pacientes Hospitalizados



REPORTE DEL MES

Nombre de Usuario: alpha
 Fecha de Emisión: 2016-07-03
 Cantidad de Atenciones: 20

Estadísticas



RAPH
Registro de Atención de Pacientes Hospitalizados



TOTAL DE REGISTROS POR USUARIO

Juan	9
beatriz	7
alpha	2

INFORMACION	DATOS RESALTANTES
Total de Atenciones durante el turno 7:00 am - 1:00 pm	0
Total de Atenciones durante el turno 1:00 pm - 7:00 pm	14
Total de Atenciones durante el turno 7:00 pm - 7:00 am	0
Personal que mas atendio a los pacientes	Ana.Coppa Baluarte: 6
Registro de Atencion mas Editado o Modificado	Registro Nro: 71 Cama: 7 Registrada el: 2016-07-03 17:13:28.165015 Total de Ediciones: 3

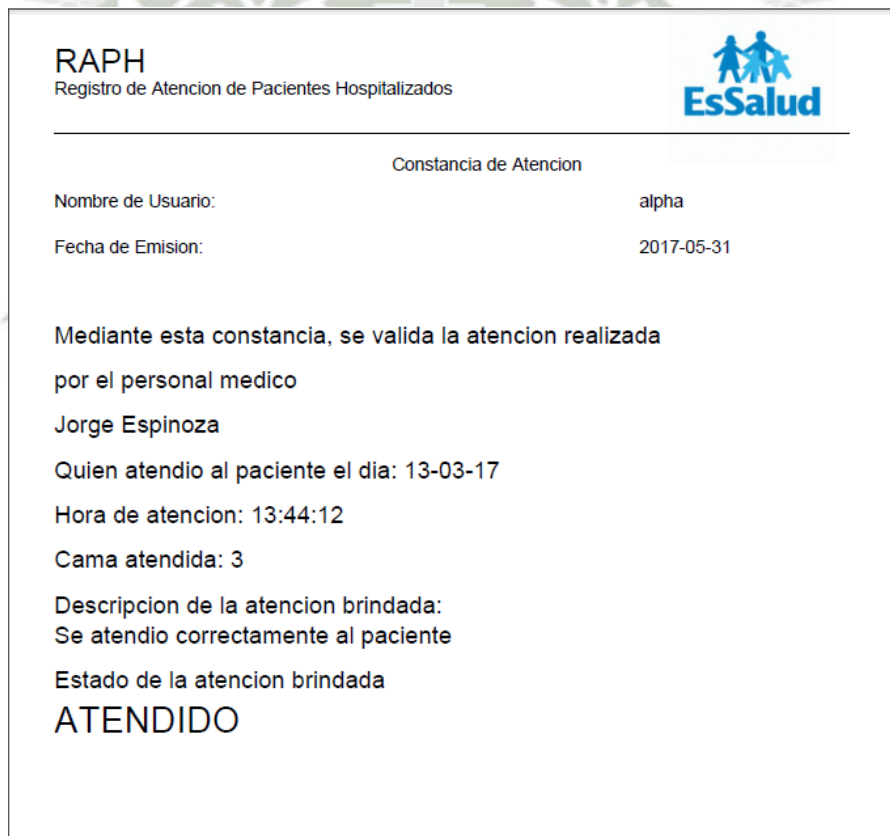
Figura 97. Pantalla de Reporte Mensual del Sistema

Fuente: Elaboración propia


En el siguiente reporte tenemos la Constancia de Atención

Es un simple informe que en un momento pueden ser necesarios, mas no es algo que sea obligatorio que el sistema tenga. Son requerimientos aparte que se desea implementar con la finalidad de mejorar el producto y brindar no solo interés en el operador, personal médico y pacientes que interactúan directamente con el sistema; sino también con autoridades de mayor rango como jefes o directores.

La constancia de Atención es un pdf que se imprime en forma de declaración jurada (no es una declaración jurada en sí) indicando que un miembro del personal fue atender al paciente, mostrando la descripción ingresada por el usuario (Ver Figura 98).



RAPH
Registro de Atención de Pacientes Hospitalizados



Constancia de Atención

Nombre de Usuario: alpha

Fecha de Emision: 2017-05-31

Mediante esta constancia, se valida la atencion realizada por el personal medico
Jorge Espinoza
Quien atendio al paciente el dia: 13-03-17
Hora de atencion: 13:44:12
Cama atendida: 3
Descripcion de la atencion brindada:
Se atendio correctamente al paciente
Estado de la atencion brindada
ATENDIDO

Figura 98. Pantalla de Constancia de Atención

Fuente: Elaboración propia

CAPITULO 4: VALIDACIÓN Y RESULTADOS

4.1.1. Descripción de Pruebas

El sistema se dividió en 2 categorías:

- Reconocimiento para pacientes en estado normal
- Reconocimiento para pacientes en estado crítico

La primera categoría funciona de la siguiente manera:

1. La Cámara Raspberry Pi analiza un área de la imagen capturada para verificar si hay o no movimiento
2. En caso se reconozca movimiento, se empieza a preprocesar esa pequeña área de la imagen para verificar qué tipo de señal se hizo.
3. La imagen extraerá algunos puntos clave de la imagen que se enviarán a sacar resultados con operaciones matemáticas para interpretar la señal
4. Los leds avisarán al paciente que su señal está siendo reconocida
5. Una vez interpretada la señal, se registra en la base de datos una nueva alerta y se visualiza en la página web
6. Después de 10 segundos, se activa la búsqueda de movimiento nuevamente en la cámara.

La segunda categoría funciona como se indica a continuación:

1. La Cámara Raspberry Pi analiza un área de la imagen capturada para verificar si hay o no movimiento
2. El paciente, al estar en estado crítico, se considera que no se puede mover mucho, por lo que se generarán alertas periódicamente cada 15 minutos, siempre y cuando no haya movimiento durante ese tiempo, además de una alerta obligatoria cada 30 min.

3. En caso el paciente quiera enviar una señal, tendrá que moverse durante 10 segundos seguidos.
4. Un led confirmará si está detectando el movimiento.
5. Otro led indicará cuando se haya enviado la señal
6. Después de 10 segundos, se activa la búsqueda de movimiento nuevamente en la cámara.
7. En caso el paciente solicite atención, unos 10 minutos antes de la revisión obligatoria, se correrá desde ese instante una atención obligatoria en los siguientes 30 minutos y una alerta periódica en los siguientes 15 minutos

Además, por evitar posibles falsos positivos durante una atención, se incluyó el uso de un botón que, al presionarlo, apagará temporalmente la cámara para que la atención se realice de manera normal, sin tener la preocupación de generar falsos positivos a la cámara. El funcionamiento del botón es el siguiente:

1. Al presionar el botón, la cámara se desactivará, pero el programa seguirá ejecutándose, esperando se vuelva a activar
2. Al presionarlo, sonará el buzzer durante 2 segundos y se prenderá el led que indica que la cámara está apagada
3. Cada 10 minutos, sonará el buzzer indicando que se active nuevamente la cámara
4. En caso no se haya prendido en los siguientes 30 min, sonará durante 5 segundos, encendiendo todos los leds parpadeando durante el mismo tiempo
5. Si no se encendió, se repetirá los puntos 3 y 4 hasta que se encienda, momento en el cual se apagará el led que indica que está apagado

Lo que se analizará en este capítulo, primeramente, será realizar las pruebas correspondientes para comprobar que el sistema funciona en todos sus módulos mencionados.

En el caso de la web, se presentará pantallas donde se verifique su funcionamiento.

Y, posteriormente, mostrar los resultados de las pruebas que se definieron en tablas e imágenes.

4.1.2. Pruebas

Algoritmo de Reconocimiento

El algoritmo de reconocimiento lo que hace es capturar, preprocesar y extraer puntos clave de la imagen para enviar a los demás algoritmos esos datos, para verificar las funciones del algoritmo, Se modifica directamente la imagen para verificar que el algoritmo está funcionando; además, se puede verificar en consola, los puntos clave que devuelve la imagen después de todo el reconocimiento

Algoritmo de Color

El algoritmo ya viene desarrollado en OpenCV, por lo que solo hay que brindar los valores en HSV de los colores que se quiere que se reconozca en un rango de mínimo a máximo; en el caso de las pruebas, se probó con el color Naranja, para ver si la detección del color es la correcta y no ocurra ninguna confusión

Algoritmo de Movimiento

Lo que se puede comprobar aquí es la varianza del área de la imagen de acuerdo con los contornos que reconozca, ya que básicamente se basa en ello el

algoritmo, para comprobar que funciona, se puede verificar que realmente reconoce una varianza en el área de la imagen capturada con la anterior inmediata.

Alerta

Aquí es más sencillo comprobar el funcionamiento, ya que se tiene que verificar que el cálculo matemático es el correcto; se verifica los datos, los resultados y se muestran por consola.

Conexión PostgreSQL

Simplemente se verifica que la conexión se ha realizado y puede registrar las alertas; para esto también se puede comprobar, yendo al sistema web a verificar que aparece una nueva alerta cuando se detecte una señal del paciente a la Cámara

Leds

Aquí se comprueba que los leds prendan de acuerdo con la alerta que se está generando, se tienen actualmente 3 Leds que son (Ver Tabla 7):

Tabla 7. Tabla de Señales predeterminadas del sistema

INTERPRETACIONES	MENSAJE	COLOR DE LED
Reconociendo una Señal	¡Necesito Atención urgente por favor! // Necesito un Doctor por favor! // Necesito una Enfermera por favor!	AZUL
Cámara Desactivada	Active la cámara por favor	ROJO
Alerta enviada al Sistema	¡Mensaje Enviado!	NARANJA

Fuente: Elaboración propia

El gesto puede ser cualquiera, se verifica más que toda la cantidad de dedos al hacer la señal y el color que se busque reconocer en la imagen para no confundir con otras señales.

También se tiene el botón y el buzzer como parte de los componentes electrónicos del proyecto, los cuales son utilizados para brindar alertas cuando la cámara está desactivada o activada.

4.1.3. Resultados

Como se pudo ver, existen muchos proyectos en cuanto a desarrollo de Reconocimiento de Patrones y sobre todo muchas herramientas las cuales podrían utilizarse para crear nuevas maneras de resolver procesos que hace años no se podía imaginar cómo solucionarlos mediante computadores; como se mencionó en los anteriores capítulos, se exploró las tecnologías actuales tanto en software como en hardware.

Se identificó qué herramientas se debían utilizar y aprender para el desarrollo de algoritmos facilitados por esta herramienta.

Se indagó sobre el estado actual de atención de pacientes hospitalizados en uno de los mayores centros de salud del Perú como es EsSalud.

Se diseñó un sistema web la cual será utilizada como sistema de procesamiento y almacenamiento de la información obtenida por los Raspberry. Ahora, lo que se mostrará es el funcionamiento del prototipo implementado en conjunto, además de comprobar la conectividad del Raspberry Pi con el sistema, verificando así que se cumplieron con los requerimientos del sistema, y listo para una próxima validación.

PROTOTIPO FUNCIONAL

Algoritmo de Reconocimiento

Se vio que principalmente, se reconozcan los puntos claves.

Para esto se mostró por consola los valores que retorna el algoritmo en la siguiente tabla, en cuanto a los defectos que reconoce en la imagen:

Lo que nos tiene que retornar son valores parecidos a la siguiente imagen (Ver Figura 99).

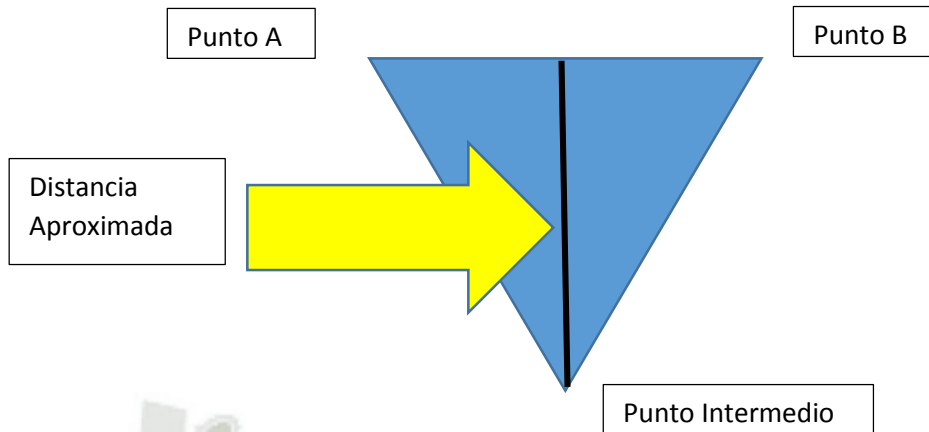


Figura 99. Interpretación de Defectos de un Contorno

Fuente: Elaboración propia

ARRAY DE DEFECTOS EN LA IMAGEN:

```

pi@raspberrypi: ~/RAPH
Archivo  Editar  Pestañas  Ayuda
defectos
[[ 525  608  139 60672]]
[[ 608  920  715 22860]]
[[ 920  922  921  114]]
[[ 922 1133 1044 22633]]
[[  0   39   17  324]]
[[  39  55   42  210]]
[[  55 122   86 1430]]
[[ 122 126  123  114]]
[[ 126 130  129  154]]
[[ 130 139  135  181]]
[[ 227 241  229  229]]
[[ 241 248  243  202]]
[[ 248 256  253  210]]
[[ 256 271  258  162]]
[[ 271 491  446 11483]]
    
```

Figura 100. Array de Defectos de la Imagen Capturada

Fuente: Elaboración propia

Los 3 primeros puntos son los índices de las coordenadas de contornos donde se encuentran, lo que se procede hacer es extraer todos estos puntos en el array de

contornos, el cual es bastante extenso como se ve a continuación (Ver Figura 101).

```
contornos
[[[139 23]]
 [ [138 24]]
 [ [137 24]]
 ...
 [[142 23]]
 [[141 23]]
 [[140 23]]]
longitud del contorno:
1145
End Motion Tracking
```

Figura 101. Array y longitud de contornos de la Imagen

Fuente: Elaboración propia

La longitud de contornos es de 1145, y la de defectos encontrados son 15 nada más; para comprobarlos se especifica en el siguiente código:

```
for i in range(DEFECTOS.shape[0]):
    PATRON_ALERTA = AlertClass.AlertClass()
    s, e, f, d = DEFECTOS[i, 0]
    PUNTO_1_INICIO = tuple(CONTORNO_SELECCIONADO[s][0])
    PUNTO_2_FINAL = tuple(CONTORNO_SELECCIONADO[e][0])
    PUNTO_3_MEDIO = tuple(CONTORNO_SELECCIONADO[f][0])
```

En el siguiente código, se recorre los defectos encontrados, se extrae los 4 datos que nos brinda cada fila, y al tener los índices de los puntos, se ingresa en la lista de contornos como se muestra en el código, para los 3 puntos clave de los defectos, y extraemos la tupla de coordenadas de esos 3 puntos.

Como vemos la extracción es correcta, esto se realiza por cada frame en el video que está capturando el Raspberry Pi; posteriormente, las otras clases harán el trabajo faltante al proyecto para obtener el resultado.

Algoritmo de Alertas

Como se ve, ya tenemos los puntos donde se encuentran todos los defectos en la imagen, ahora lo que se debe hacer es limitar todos esos defectos al clásico defecto que producen 2 dedos, la cual es una curvatura en forma de triángulo entre ambos; esto se considera como un defecto en la imagen y para encontrar ese defecto se deben hacer unos procedimientos matemáticos.

Primeramente, se tiene 3 puntos que forman básicamente un triángulo.

Primero calcular la distancia entre ambos puntos, para lo cual la fórmula es la siguiente:

```
def CalcDistancia(self,X,Y):  
    try:  
        Distancia = math.sqrt((X[0] - Y[0])**2 + (X[1] - Y[1])**2)  
    except Exception, e:  
        print "No se pasaron los puntos correctos"  
        return False  
    return Distancia
```

Se pasa a esta función dentro de la clase AlertClass los puntos para calcular la distancia entre todos, esto con la finalidad de luego usar lo que se conoce como el teorema del Coseno. Este teorema es una variación del teorema de Pitágoras para hallar la distancia entre 2 puntos conociendo el valor de los otros 2 lados (Ver Figura 102).

$$c^2 = a^2 + b^2 - 2ab \cos \gamma$$

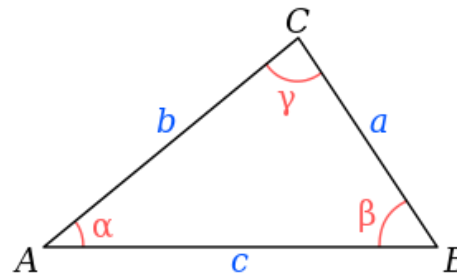


Figura 102. Interpretación del Teorema del Coseno

Fuente: Wikipedia.com

Gracias a este teorema también se puede hallar el ángulo faltante en un triángulo:

$$\gamma = \arccos \frac{a^2 + b^2 - c^2}{2ab}$$

Y esta es la fórmula que se utilizará para hallar el ángulo entre ambos puntos. Sin embargo, el resultado que brinda esta fórmula con los datos que tenemos está en radianes, por lo que es necesario pasarlo a grados para controlar la condición de que el ángulo sea menor a 90° para ser considerado. (Granville-Smith-Mikesh, 1982)

El valor necesario por el que hay que multiplicar el resultado, según Google es (Ver Figura 103).

Ángulo plano		▼
1	=	57.2958
Radián		Grado sexagesimal

Figura 103. Valor de Conversión de Radián a Grado Sexagesimal

Fuente: Google.com

A continuación, se verá los valores resultantes al hacer los cálculos correspondientes

Primero se evaluó con 12 conjuntos de coordenadas para evaluar el resultado

(Ver Tabla 8)

Tabla 8. Tabla de Resultados de las coordenadas

PUNTO A	PUNTO B	PUNTO C	DIST A-B	DIST A-C	DIST B-C	ÁNGULO
(172, 141)	(156, 93),	(114, 127)	50.5964425627	59.6657355607	54.0370243444	52.5614284157
(148, 93)	(1, 100),	(36, 126)	147.166572291	116.760438506	43.6004587132	126.975692328
(1, 177)	(92, 183),	(22, 159)	91.1975876874	27.6586333719	74.0	120.474060911
(96, 183),	(99, 182),	(97, 182)	3.16227766017	1.41421356237	2.0	134.999999969
(99, 182),	(101, 181),	(100, 181)	2.2360679775	1.41421356237	1.0	134.999999969
(101, 181),	(171, 144),	(117, 149)	79.1770168673	35.77708764	54.2309874518	121.855132355
(171, 144),	(172, 142),	(171, 143)	2.2360679775	1.0	1.41421356237	134.999999969
(173, 141),	(158, 93),	(115, 127)	50.2891638427	59.6657355607	54.8178802947	51.9037744824
(152, 92),	(1, 100),	(35, 126)	151.211772029	121.840059094	42.8018691181	126.390869125
(1, 179),	(94, 183),	(23, 159)	93.0859817588	29.7321374946	74.9466476902	119.049609235
(95, 183),	(100, 182),	(96, 182)	5.09901951359	1.41421356237	4.0	134.999999969
(100, 182),	(170, 145),	(119, 148)	79.1770168673	38.9486841883	51.0881590978	122.563946682

Fuente: Elaboración propia

De los resultados observados, se aprecia que solo 2 de los 12 conjuntos de coordenadas pasan la condición de ser menores o igual a 90 grado; Así, estos se van contabilizando y generando el mensaje. El resultado del mensaje sería el siguiente (Ver Figura 104).

```

coordenadas recibidas
[(1, 179), (94, 183), (23, 159)]
Distancias calculadas
93.0859817588
29.7321374946
74.9466476902
Angulo calculado
119.049609235
coordenadas recibidas
[(95, 183), (100, 182), (96, 182)]
Distancias calculadas
5.09901951359
1.41421356237
4.0
Angulo calculado
134.999999969
coordenadas recibidas
[(100, 182), (170, 145), (119, 148)]
Distancias calculadas
79.1770168673
38.9486841883
51.0881590978
Angulo calculado
122.563946682
Necesito Atencion urgente por favor!
Mensaje enviado!
coordenadas recibidas
[(173, 142), (173, 140), (1, 99)]
Distancias calculadas

```

Figura 104. Mensaje Enviado una vez calculado la cantidad de Defectos

Fuente: Elaboración propia

Además, para verificar que funciona y no reconoce falsos positivos, se puso imágenes cuyo valor podía engañar al algoritmo de color; sin embargo, por el algoritmo de reconocimiento, este no podía generar un falso positivo, ya que ambos filtros deben ser pasados para poder funcionar correctamente (Ver Figuras 105 y 106).

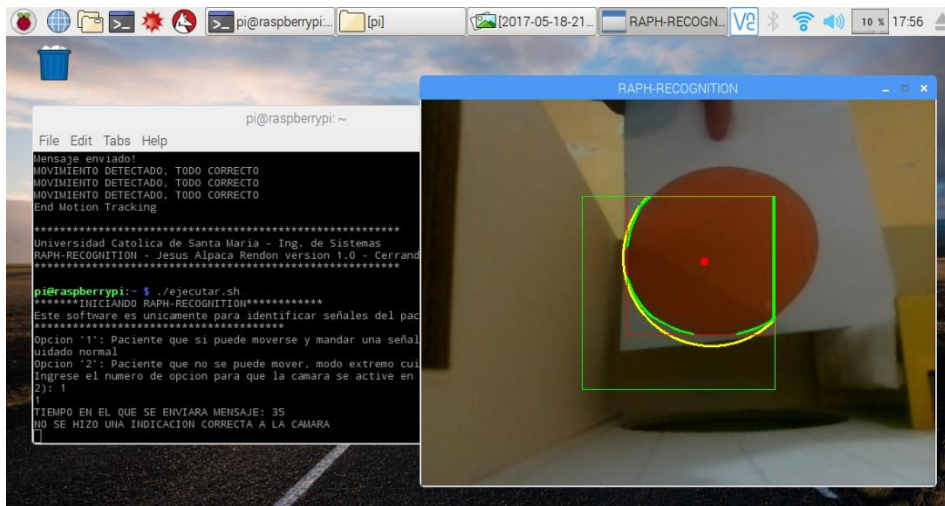


Figura 105. Prueba de Falso Positivo

Fuente: Elaboración propia

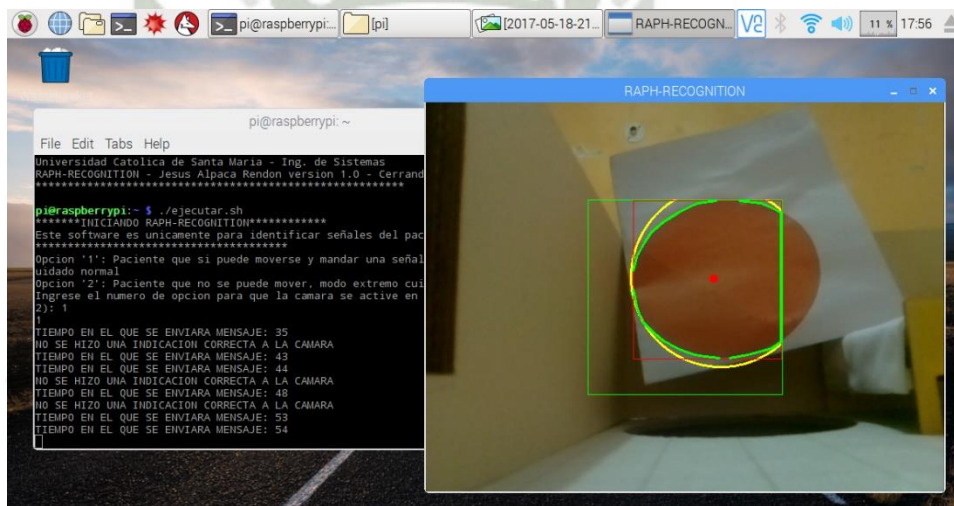


Figura 106. Prueba de Falso Positivo superada

Fuente: Elaboración propia

Algoritmo de Movimiento

El algoritmo de movimiento es más sencillo que los demás, ya que se consideró como prioridad del movimiento, considerar datos básicos que no se repetían y que la librería podría darnos como la cantidad de contornos, defectos, etc. Sin embargo, al parecer sí había variaciones en esos datos por lo que se decidió considerar datos un poco más profundos: distancia, área, altura, etc.

Al extraer contornos necesarios, se puede conocer el área que existe dentro; nos brinda un valor float, en el que si la imagen se encuentra estática, el área no cambia y los contornos se respetan. Así que se planteó usar este dato para verificar si existe o no movimiento.

Cuando inicia el programa, se asigna un valor pequeño al área inicial; en caso ocurra un movimiento, el área cambiará y se comparará con la anterior, si existe una variación, hay movimiento. El código que hace toda la comparación es el siguiente:

```
for C in self.Contours:  
    # get área of next contour  
    ÁREA_FOUND = cv2.contourArea(C)  
    # find the middle of largest bounding rectangle  
    if ÁREA_FOUND > self.BigÁrea:  
        self.Movement = True  
        self.BigÁrea = ÁREA_FOUND
```

La siguiente imagen muestra los resultados obtenidos (Ver Figura 107).

```
358.0  
29.5  
1.0  
1177.0  
Area inicial  
25  
Areas de cada arreglo de contornos  
28.0  
1580.0  
2511.0  
280.5  
Area inicial  
25  
Areas de cada arreglo de contornos  
20.0  
4654.5  
4820.0  
Area inicial  
25  
Areas de cada arreglo de contornos  
225.5  
4265.5  
234.5  
4683.0  
Area inicial  
25  
Areas de cada arreglo de contornos  
65.0  
30.5  
0.0  
29.0  
2396.0  
3743.5  
End Motion Tracking
```

Figura 107. Comparación de Áreas de Contornos para detectar el movimiento

Fuente: Elaboración propia

Como se puede apreciar, se pone un área inicial cada vez que la cámara captura un frame; si los contornos varían en el siguiente frame, se pasa el valor del área obtenida al atributo de la clase para almacenarlo, y así ir comparando las áreas para ver si hay o no movimiento. En este caso se puso como área inicial el valor de 25, pero se ve que puede variar extremadamente el valor de acuerdo a la variación detectada por la cámara.

Para verificar esto en un ambiente real, se puso a prueba el algoritmo con la segunda categoría que es la que más emplea el algoritmo de movimiento. Para esto como prueba, se muestran las siguientes imágenes:

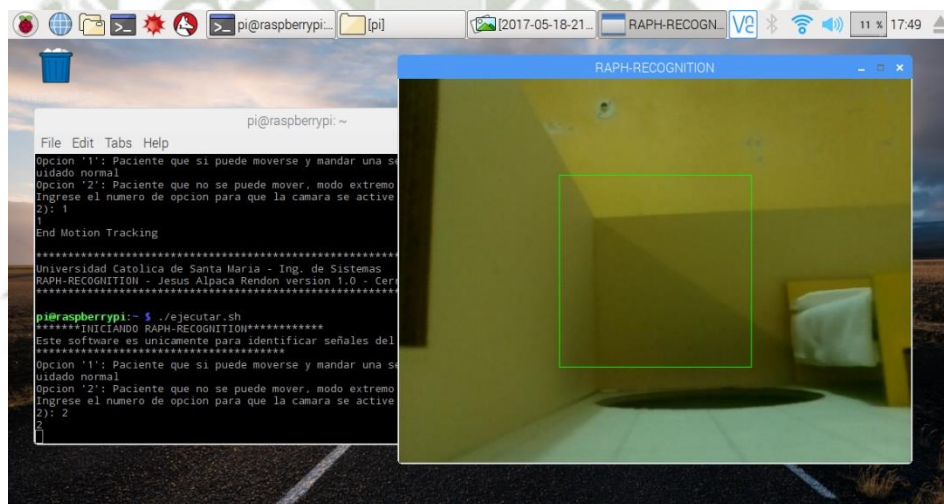


Figura 108. Ingreso a la categoría 2 (Detección de movimiento)

Fuente: Elaboración propia

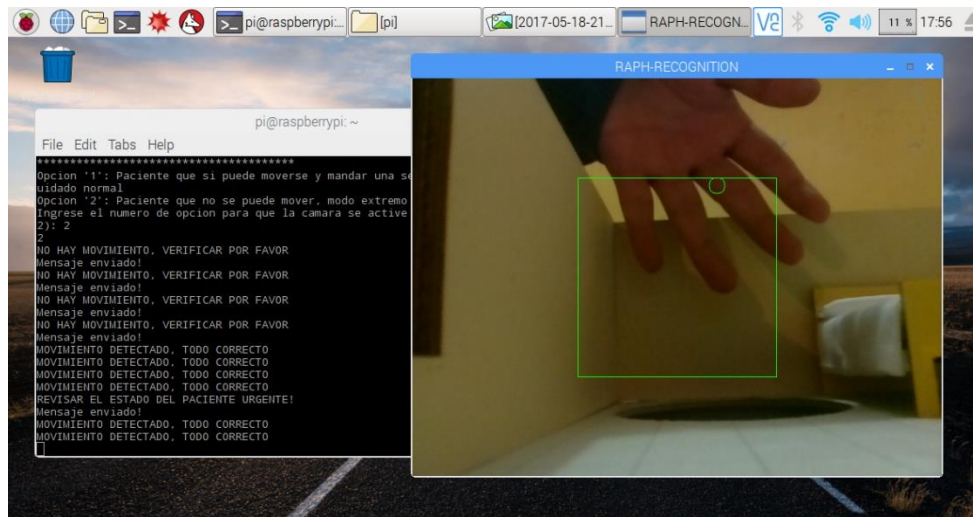


Figura 109. Detección de movimiento funcional

Fuente: Elaboración propia

Algoritmo de Color

Para comprobar el funcionamiento del color, ya que como se indicó previamente el algoritmo está ya definido en Open CV, simplemente se pondrá objetos imágenes que comprueban que no se reconoce otros colores que no sean color Naranja. A continuación, las imágenes:

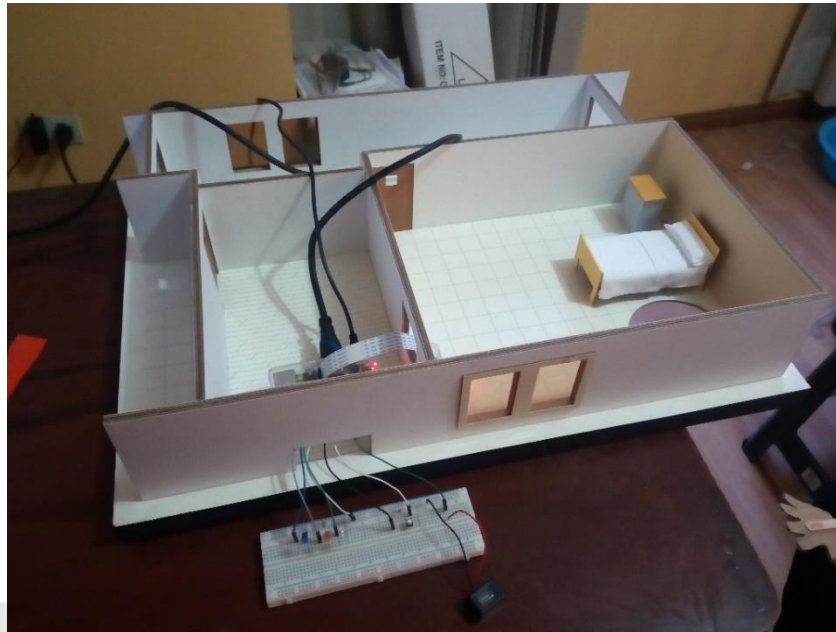


Figura 110. Maqueta de prueba

Fuente: Elaboración propia

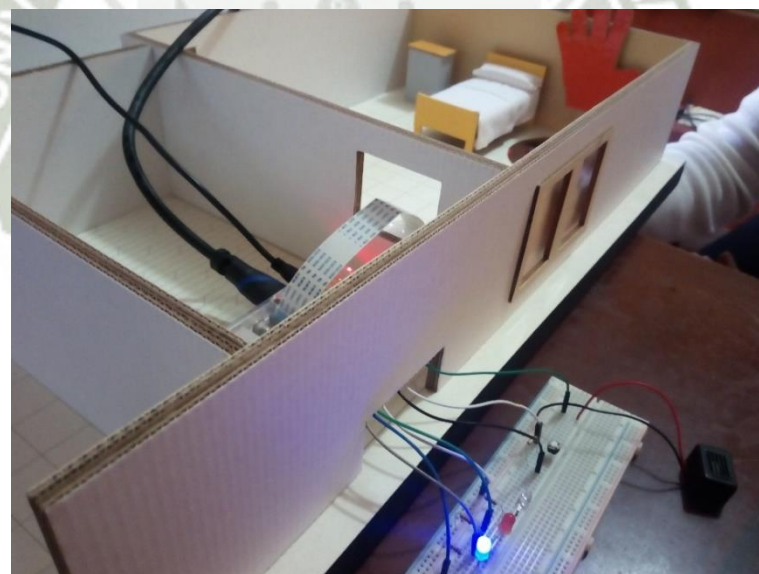


Figura 111. Componentes conectados en el Raspberry Pi dentro de la maqueta

Fuente: Elaboración propia

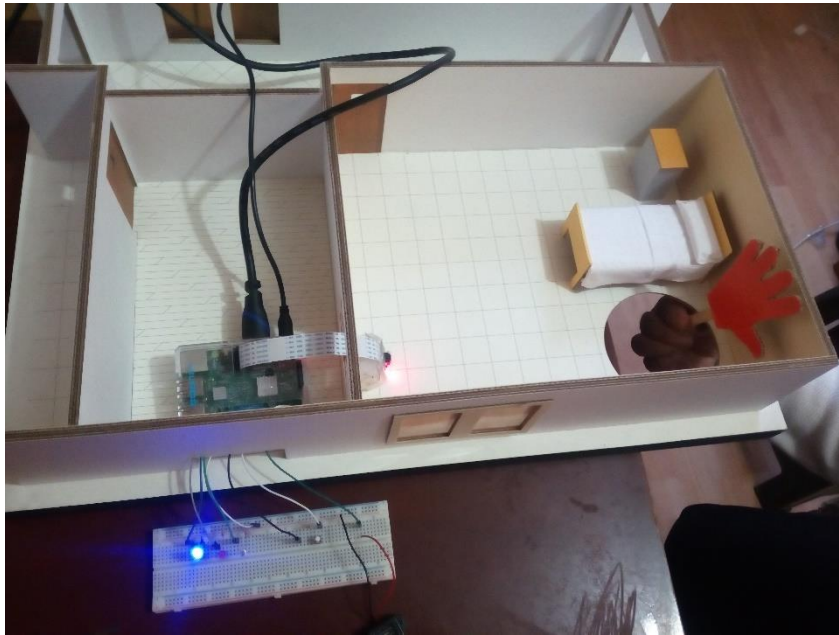


Figura 112. Maqueta con Led azul prendido

Fuente: Elaboración propia

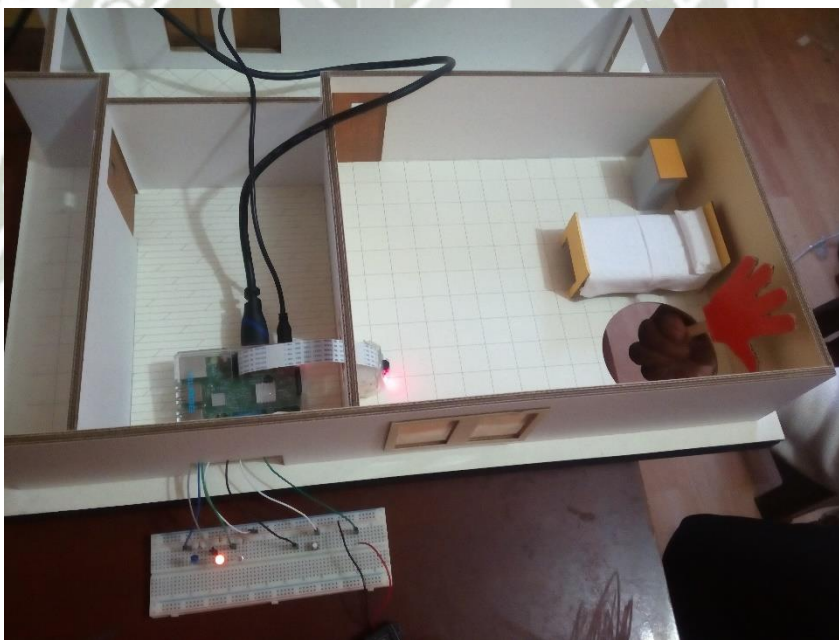


Figura 113. Maqueta con Led naranja prendido

Fuente: Elaboración propia

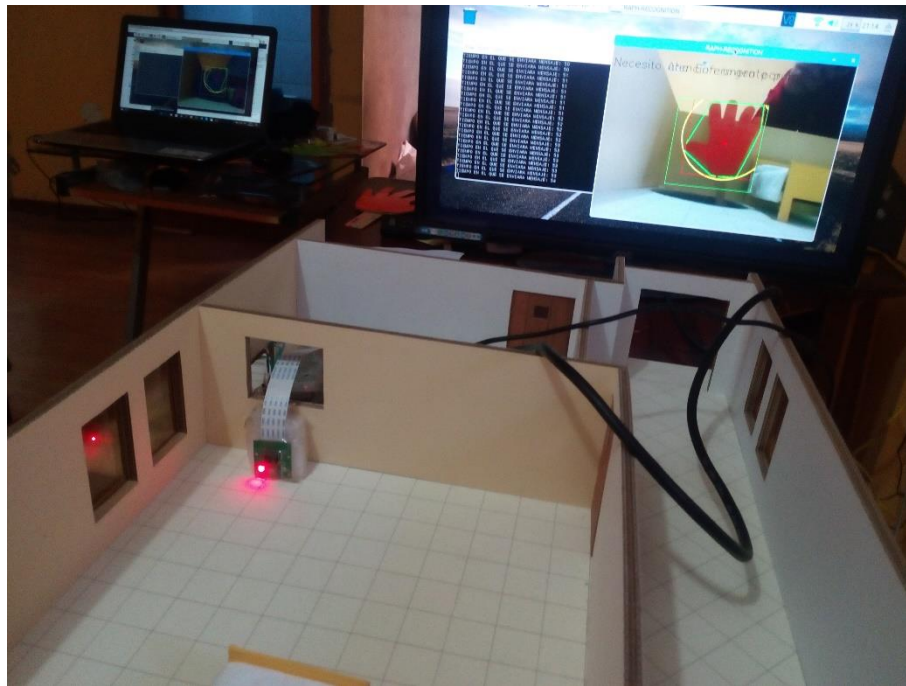


Figura 116. Prueba de detección desde otra perspectiva

Fuente: Elaboración propia

CONECTIVIDAD RASPBERRY PI – SISTEMA WEB

Conexión SQL

Es más sencillo comprobar este algoritmo ya que lo único que tenemos que comprobar es que se haya registrado una nueva alerta en la base de datos del sistema y que a la vez aparezca en el sistema web.

Primeramente, se asigna el ID de una cama que tendrá bajo vigilancia el Raspberry Pi, en este caso la cama 13 (Ver Figura 117).

```
class SqlClass(object):  
    """docstring for SqlClass"""  
    def __init__(self):  
        super(SqlClass, self).__init__()  
        self.stringconn = ""  
        self.ID = 13  
        self.Mensaje = ""
```

Figura 117. Atributos de la clase Sql

Fuente: Elaboración propia

Una vez ejecutado el programa y hecha la señal correcta, se pasa a comprobar que se haya enviado la alerta al sistema (Ver Figura 118).

```
(cv) pi@raspberrypi:~/RAPH $ python MainClass2.py
*****INICIANDO RAPH-RECOGNITION*****
Este software es unicamente para identificar señales del paciente
Necesito ir al Baño!
Mensaje enviado!
Necesito Atencion urgente por favor!
Mensaje enviado!
End Motion Tracking

*****
Universidad Catolica de Santa Maria - Ing. de Sistemas
RAPH-RECOGNITION - Jesus Alpaca Rendon version 1.0 - Cerrando programa
*****
(cv) pi@raspberrypi:~/RAPH $
```

Figura 118. Mensajes Enviados a la Base de Datos

Fuente: Elaboración propia

Se verifica si en la base de datos se registró (Ver Figura 119).

41	75	13	2016-07-06 17:53:46.865393-05		TRUE	2	Necesito una Enfermera por favor!		
42	76	9	2016-07-26 00:21:34.264013-05		TRUE	2	Necesito Atencion urgente!		
43	77	15	2016-07-26 00:21:43.702553-05	no hubo ningun problema	TRUE	7	2	Necesito Atencion urgente!	
44	78	15	2016-07-26 00:26:48.253955-05		TRUE	2	Necesito Atencion urgente!		
45	79	15	2016-07-26 00:29:58.638844-05		TRUE	2	Necesito Atencion urgente!		
46	80	15	2016-07-26 00:29:59.399888-05		FALSE	2	Necesito Atencion urgente!		
47	81	15	2016-07-26 00:33:20.172361-05	problema resuelto sin problemas	TRUE	7	2	Necesito Atencion urgente!	4
48	82	15	2016-07-26 00:34:35.570673-05	lalalalala	TRUE	7	2	Necesito Atencion urgente!	4
49	83	15	2016-07-26 00:42:45.8747-05	se atendio correctamente al paciente sin ningun problema	TRUE	7	2	Necesito Atencion urgente!	4
50	84	26	2016-07-26 00:46:27.428369-05	sdffsdffa	TRUE	7	2	Necesito Atencion urgente!	
51	85	13	2016-08-08 18:16:37.839087-05		FALSE			Necesito ir al Baño!	
52	86	13	2016-08-08 18:16:54.782056-05		FALSE			Necesito Atencion urgente por favor!	



Figura 119. Mensaje Registrado y Comprobado en la misma Base de Datos

Fuente: Elaboración propia

Ahora se verifica en el sistema si aparece

Registro de Atención de Pacientes Hospitalizados

R-A-P-H

Hola, beatriz, ahora tiene acceso al sistema

recuerde verificar siempre las alertas

Menu Principal

Alertas

Pacientes

Ver Lista de Atenciones

Cerrar Sesión

ALERTAS EN TIEMPO REAL

Descripcion de Alerta	Opciones de Alerta
Necesito ir al Baño! Cama Nro: 7 // Aug. 8, 2016, 6:16 p.m.	<input checked="" type="checkbox"/> Atendido <input type="checkbox"/> No Atendido
Necesito Atencion urgente por favor! Cama Nro: 7 // Aug. 8, 2016, 6:16 p.m.	<input checked="" type="checkbox"/> Atendido <input type="checkbox"/> No Atendido

Figura 120. Alertas mostradas en el Sistema Web

Fuente: Elaboración propia

	id [PK] serial	Nro_Cama character varying(3)	Area_id integer	Id_paciente_id integer	ActivaC boolean	EstadoC boolean
1	1	15	1	2	TRUE	TRUE
2	5	20	1		TRUE	FALSE
3	6	5	2		TRUE	FALSE
4	7	34	3		TRUE	FALSE
5	8	1	1	6	TRUE	TRUE
6	9	2	1	13	TRUE	TRUE
7	10	3	1	8	TRUE	TRUE
8	11	4	1	9	TRUE	TRUE
9	12	6	1		TRUE	FALSE
10	13	7	1	3	TRUE	TRUE
11	14	8	1	12	TRUE	TRUE
12	15	9	1	4	TRUE	TRUE
13	26	10	1		TRUE	FALSE
14	27	11	2		TRUE	FALSE
15	28	12	2		TRUE	FALSE
16	29	13	2	27	TRUE	TRUE
17	30	14	2		TRUE	FALSE
18	31	16	2		TRUE	FALSE
19	32	17	2		TRUE	FALSE
*						

Figura 121. Verificación de la Existencia de la Cama 13 en la base de Datos

Fuente: Elaboración propia

Se comprueba por medio del ID si el nro. de cama es el que corresponde, ya que el ID y el Nro. de Cama pueden ser distintos; en caso de un mal ingreso de datos al sistema, el ID asegura la identificación de la cama.

Como se puede ver, el Raspberry Pi, a través de su conexión por Wifi, puede conectarse a la base de datos y hacer el correcto registro de la alerta que el paciente está generando.

Funcionamiento de LEDS

Por último, comprobar que los LEDS indiquen al paciente qué señal está haciendo; como se mencionó al especificar las pruebas, cada color indica un mensaje del Raspberry Pi. Por lo que se comprobaba que se esté realizando el mensaje correcto para cada led.

Las siguientes imágenes son para confirmar la detección de la señal:

Señal Válida del Paciente a la Cámara:

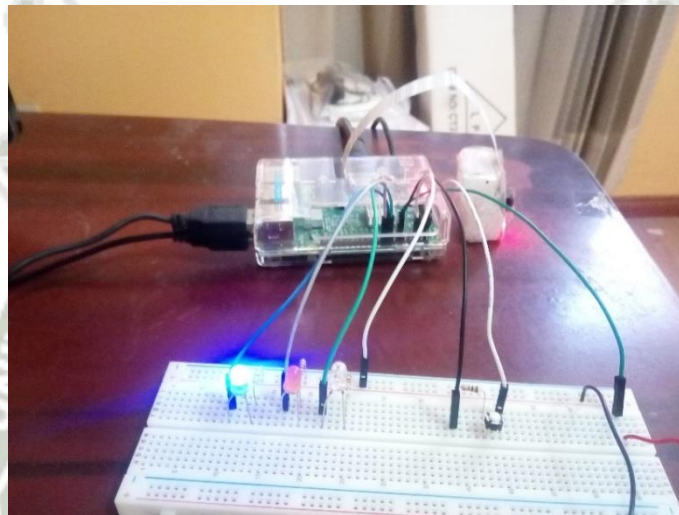


Figura 122. Comprobación del funcionamiento del Led 1

Fuente: Elaboración propia

Alerta enviada al sistema:

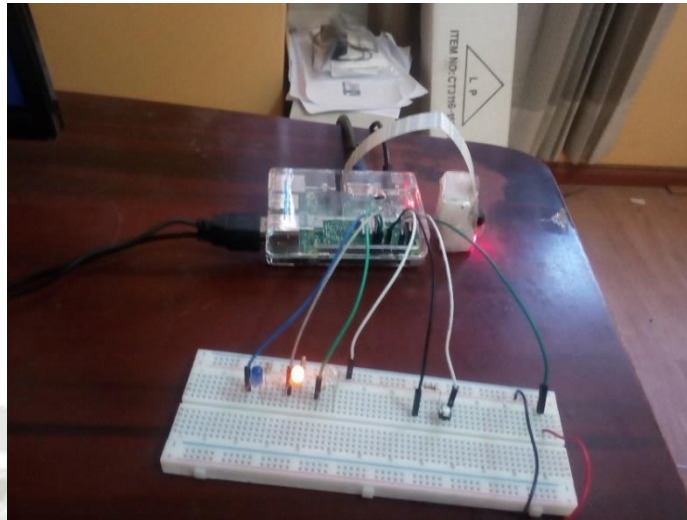


Figura 123. Comprobación del funcionamiento del Led 2

Fuente: Elaboración propia

Cámara Desactivada Temporalmente:

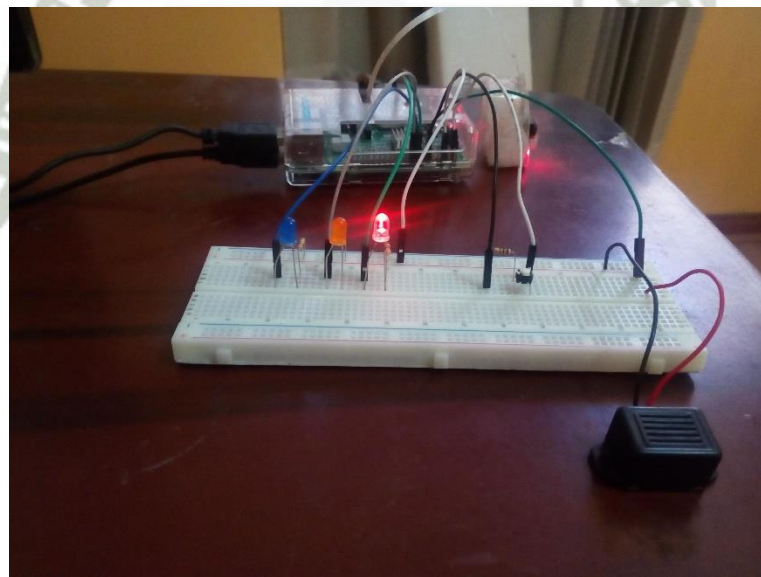
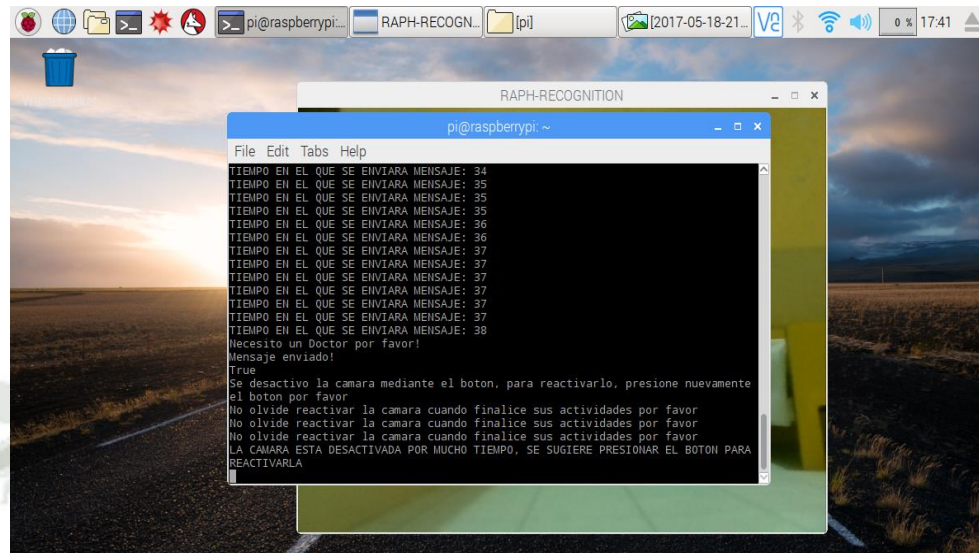


Figura 124. Comprobación del funcionamiento del Led 3

Fuente: Elaboración propia

Los leds prenden de manera correcta, por lo que se comprueba que el sistema en Raspberry Pi con su Cámara Megapíxel funciona de manera correcta.

Para comprobar el funcionamiento del botón y buzzer de apagado/encendido de la cámara, se tomó la siguiente pantalla en la cual se esperó el tiempo indicado al inicio de este punto, para comprobar que el buzzer y los mensajes que emitía eran los correctos (Ver Figura 125).



```

RAPH-RECOGNITION
pi@raspberrypi: ~
File Edit Tabs Help
TIEMPO EN EL QUE SE ENVIARA MENSAJE: 34
TIEMPO EN EL QUE SE ENVIARA MENSAJE: 35
TIEMPO EN EL QUE SE ENVIARA MENSAJE: 35
TIEMPO EN EL QUE SE ENVIARA MENSAJE: 35
TIEMPO EN EL QUE SE ENVIARA MENSAJE: 36
TIEMPO EN EL QUE SE ENVIARA MENSAJE: 36
TIEMPO EN EL QUE SE ENVIARA MENSAJE: 37
TIEMPO EN EL QUE SE ENVIARA MENSAJE: 37
TIEMPO EN EL QUE SE ENVIARA MENSAJE: 37
TIEMPO EN EL QUE SE ENVIARA MENSAJE: 37
TIEMPO EN EL QUE SE ENVIARA MENSAJE: 37
TIEMPO EN EL QUE SE ENVIARA MENSAJE: 37
TIEMPO EN EL QUE SE ENVIARA MENSAJE: 37
TIEMPO EN EL QUE SE ENVIARA MENSAJE: 38
Necesito un Doctor por favor!
Mensaje enviado!
True
Se desactivo la camara mediante el boton, para reactivarlo, presione nuevamente
el boton por favor
No olvide reactivar la camara cuando finalice sus actividades por favor
No olvide reactivar la camara cuando finalice sus actividades por favor
No olvide reactivar la camara cuando finalice sus actividades por favor
LA CAMARA ESTA DESACTIVADA POR MUCHO TIEMPO, SE SUGIERE PRESIONAR EL BOTON PARA
REACTIVARLA
    
```

Figura 125. Prueba de avisos de cámara apagada

Fuente: Elaboración propia

Verificación

El sistema desarrollado en Django Python se puede testear utilizando esta misma herramienta, Django brinda un módulo exclusivo de test, incluso al momento de crear la aplicación, nos crea archivos para realizar correctamente los test a nuestra aplicación, brindando varios componentes y sentencias que nos facilitan testear la aplicación correctamente.

Al momento de crear la aplicación, se crea un archivo llamado test.py, es aquí donde se podrá realizar los test correspondientes, para ello se importa la librería django.test, y de ella extraemos las clases Client y TestCase; esta última clase nos permitirá definir que lo que ejecutemos en este archivo es exclusivamente de test y no afectará la aplicación ni su funcionamiento. La clase Client nos permite

emular el navegador web, verificando si la página cargó correctamente y extrayendo información sobre el contenido y variables utilizadas dentro del código hecho en Python, comprobando incluso la extracción de la base de datos.

Un punto importante aquí es que, al momento de ejecutar los test, Django crea una base de datos nueva y vacía, es decir, que se tendrá que crear datos dentro de ella al momento de ejecutarla si se quiere realizar test con base de datos, permitiendo así la individualidad con la aplicación desarrollada. Para finalizar se debe crear una clase que derive de TestPage para poder crear ahí los métodos de test unitarios de la aplicación. (Django,2016)

Para la aplicación se crearon exactamente 41 test, los cuales evalúan cada una de las funciones que realiza la aplicación; además de evaluar las diferentes situaciones en las que podría fallar estos métodos. Los resultados de los test ejecutados de la aplicación, a continuación:

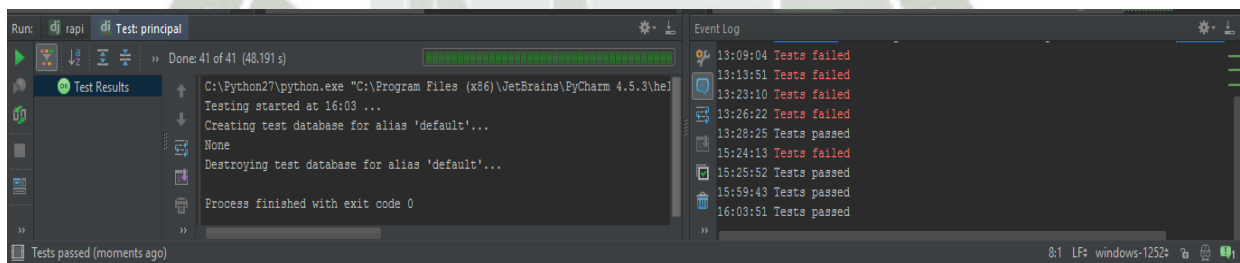
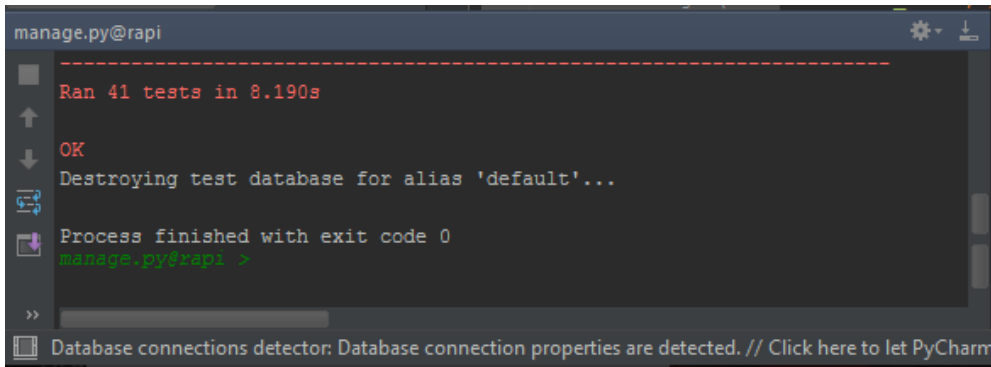


Figura 126. Test Ejecutados y Pasados

Fuente: Elaboración propia



```
manage.py@rapi
-----
Ran 41 tests in 8.190s
OK
Destroying test database for alias 'default'...
Process finished with exit code 0
manage.py@rapi >
Database connections detector: Database connection properties are detected. // Click here to let PyCharm
```

Figura 127. Test Ejecutados y Pasados en Consola

Fuente: Elaboración propia

Validación

Para validar el proyecto se realizaron encuestas de validación tanto a los pacientes como al personal, con el permiso que se consiguió al inicio del proyecto en el Hospital Edmundo Escobel I, fue factible realizar las encuestas de validación. Para este proceso se procedió a instalar el Raspberry Pi al costado de las camas de los pacientes, casi a la mitad, ya que a los pacientes les resultaba cómodo sin necesidad de realizar mucho movimiento; se colocó los leds y demás componentes cerca para que puedan visualizar todo, esto sobre una mesa pequeña que estaba colocada a un costado de la cama donde no pasan muchas personas para no incomodar a otras personas. Se llevó una laptop donde estaba el servidor del sistema web inicializado para que puedan visualizar su funcionamiento los miembros del personal médico.

A los pacientes se les preguntó específicamente sobre el Raspberry Pi y su cámara Megapíxel para ver si les parecía adecuado el método, sus respuestas a la encuesta fueron las siguientes:

RESULTADOS DE LAS PREGUNTAS A LOS PACIENTES

Pregunta 1:

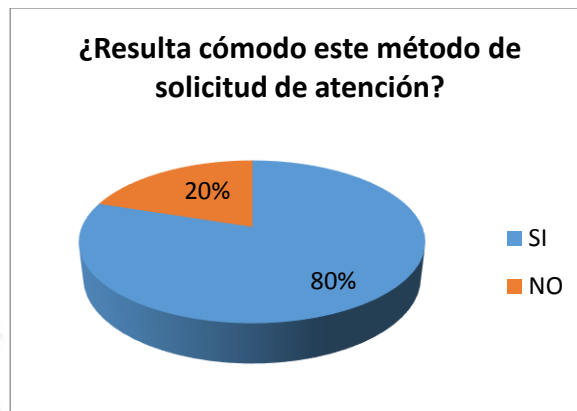


Figura 128. Grafica de resultados de la pregunta a los pacientes

Fuente: Elaboración propia

Pregunta 2:

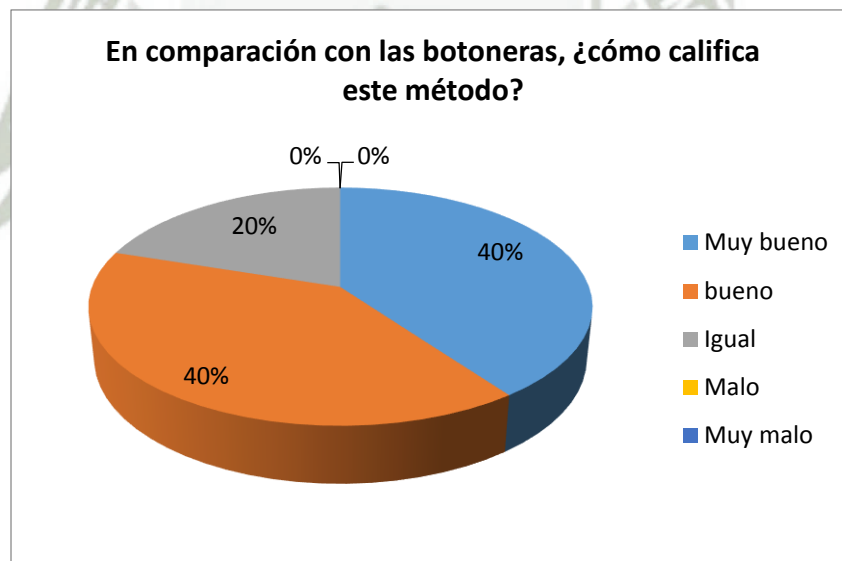


Figura 129. Grafica de resultados de la pregunta a los pacientes

Fuente: Elaboración propia

Pregunta 3:

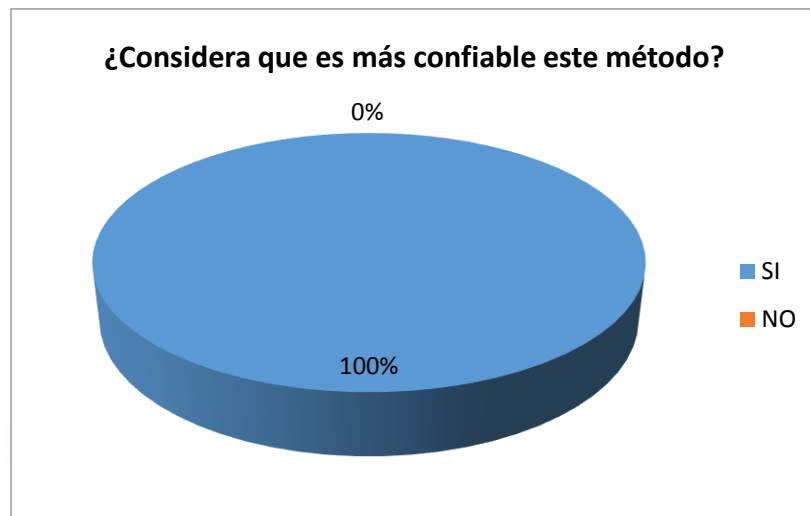


Figura 130. Grafica de resultados de la pregunta a los pacientes

Fuente: Elaboración propia

Pregunta 4:

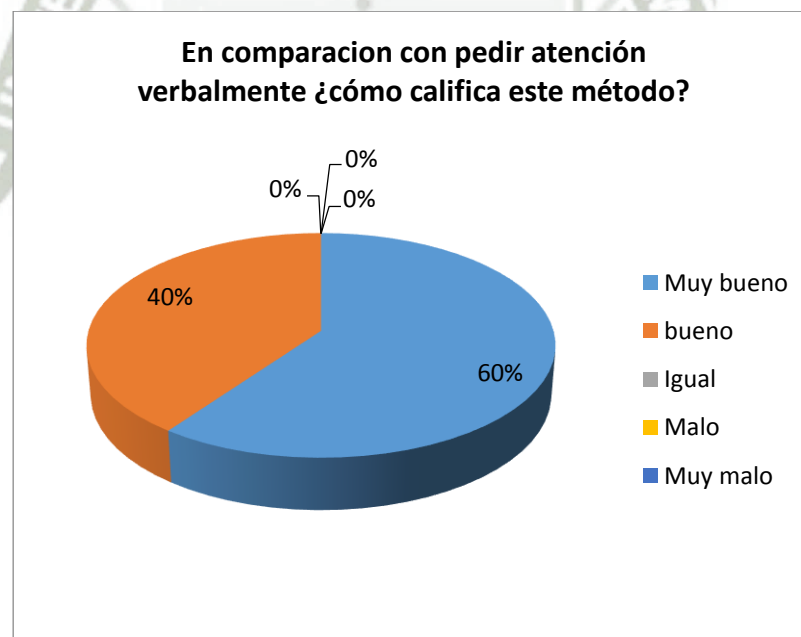


Figura 131. Grafica de resultados de la pregunta a los pacientes

Fuente: Elaboración propia

Pregunta 5:

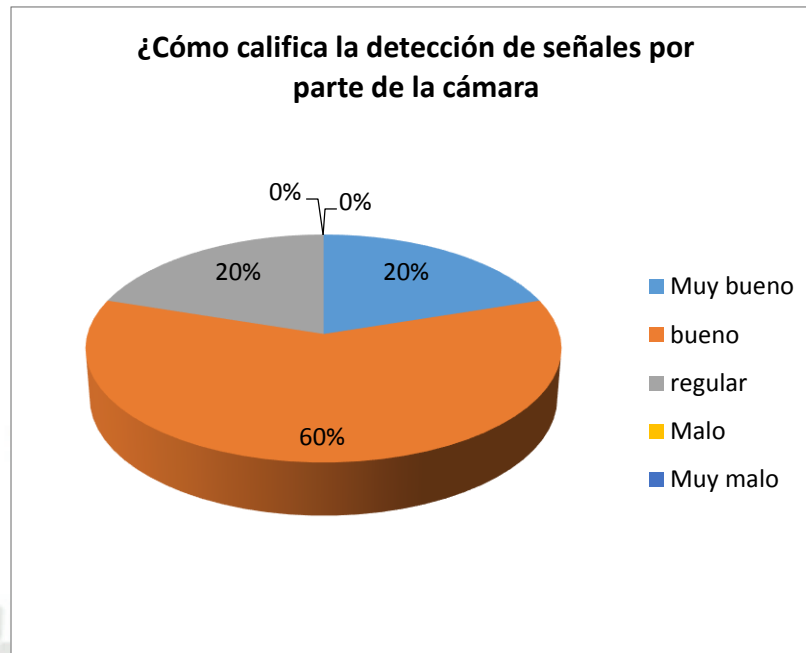


Figura 132. Grafica de resultados de la pregunta a los pacientes

Fuente: Elaboración propia

Pregunta 6:

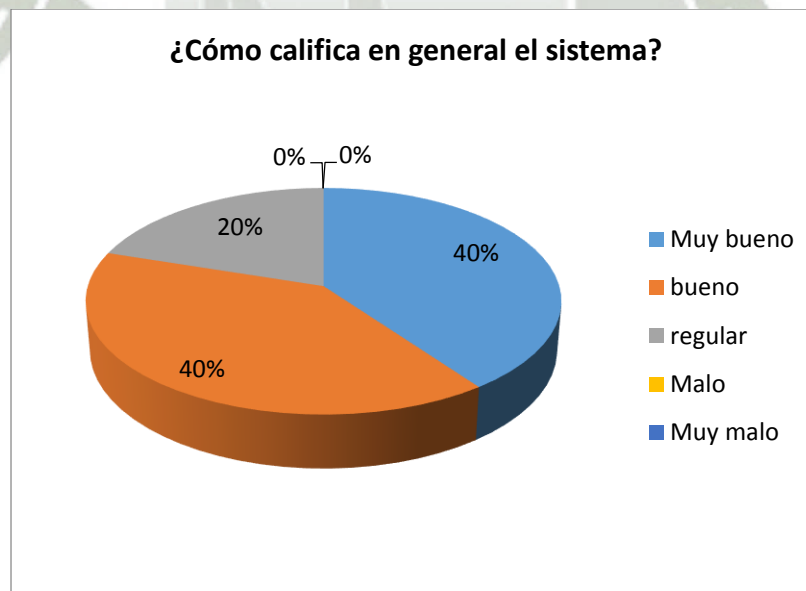


Figura 133. Grafica de resultados de la pregunta a los pacientes

Fuente: Elaboración propia

Como se puede observar en los resultados de las encuestas, se obtiene una conformidad estable por parte de los pacientes, ya que les agradó mucho el proyecto y la idea. Cuando se les explicó cómo funcionaría el sistema, indicaron que sería de gran utilidad ya que el sistema de botoneras no siempre ayuda, y usualmente los pacientes prefieren llamar al personal alzando la voz. Su incomodidad surgió al ver una mesa pequeña al costado de su cama; lo que los incomodaría un poco, pero con el personal presente en la habitación se vería a futuro una manera más factible de colocar la cámara de manera que no incomode al paciente. En general, les gustó el resultado mostrado, como retroalimentación adicional indicaron que en su caso se sentían cómodos con que la cámara reconociera su movimiento, los mensajes por defecto no necesariamente eran indispensables para ellos, pero sí les pareciero buena idea para una mejor comunicación con el personal.

En cuanto a las encuestas realizadas al personal entre los que se encuestó a 10 enfermeras y 5 doctores del área de internados, se les preguntó particularmente, sobre el sistema web ya que ellos serían los que en un momento harían uso o que tendrían que conseguir a un operador del sistema para poder apoyarlos.

Las preguntas que se les realizó fueron las siguientes:

RESULTADO DE LAS PREGUNTAS AL PERSONAL

Pregunta 1:

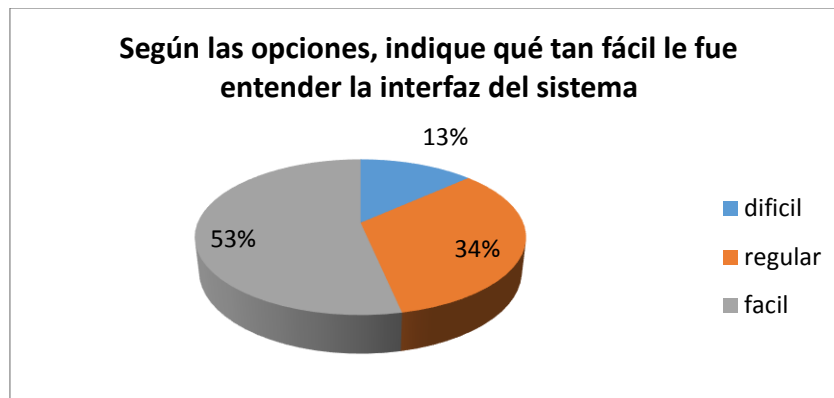


Figura 134. Grafica de resultados de la pregunta al personal

Fuente: Elaboración propia

Pregunta 2:

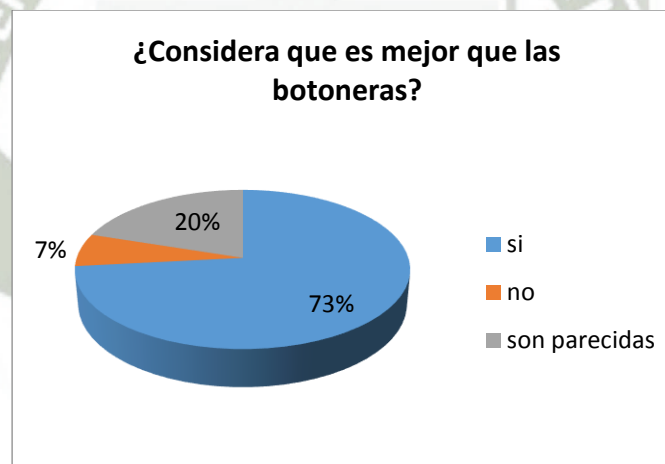


Figura 135. Grafica de resultados de la pregunta al personal

Fuente: Elaboración propia

Pregunta 3:

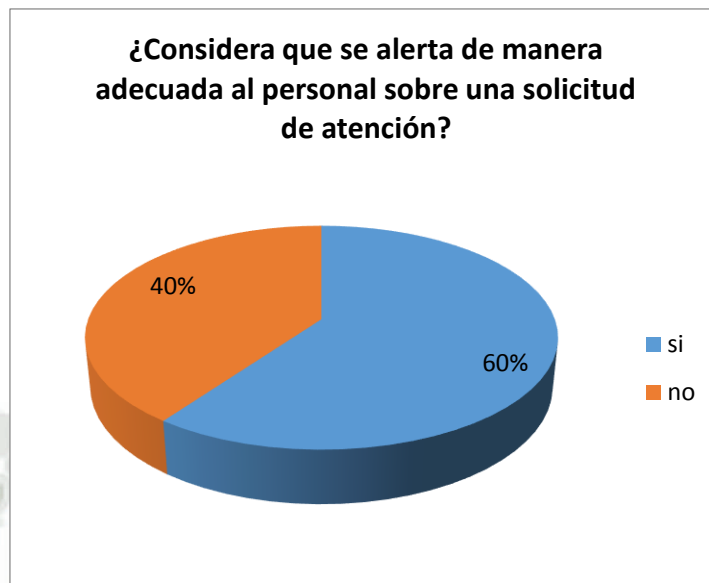


Figura 136. Grafica de resultados de la pregunta al personal

Fuente: Elaboración propia

Pregunta 4:



Figura 137. Grafica de resultados de la pregunta al personal

Fuente: Elaboración propia

Pregunta 5:

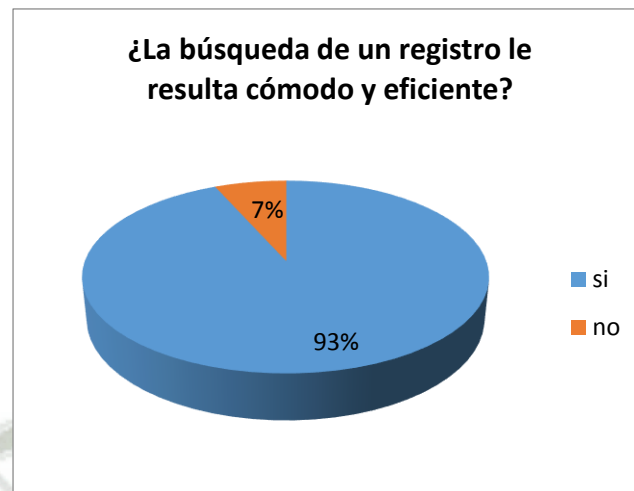


Figura 138. Grafica de resultados de la pregunta al personal

Fuente: Elaboración propia

Pregunta 6:

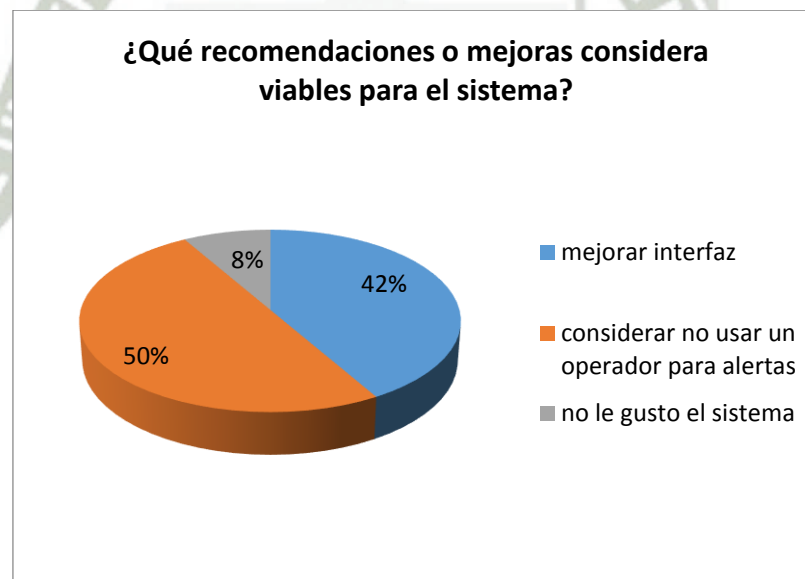


Figura 139. Grafica de resultados de la pregunta al personal

Fuente: Elaboración propia

Pregunta 7:

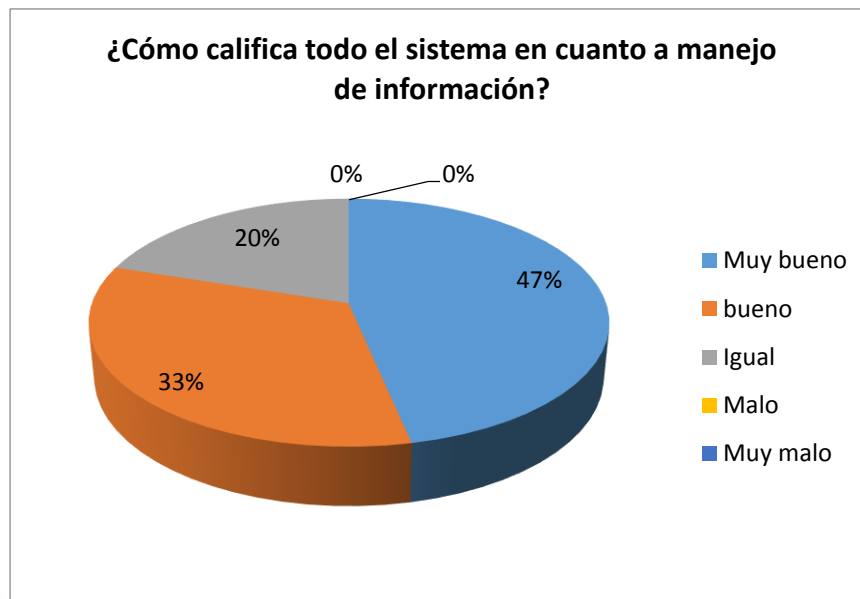


Figura 140. Grafica de resultados de la pregunta al personal

Fuente: Elaboración propia

Pregunta 8:

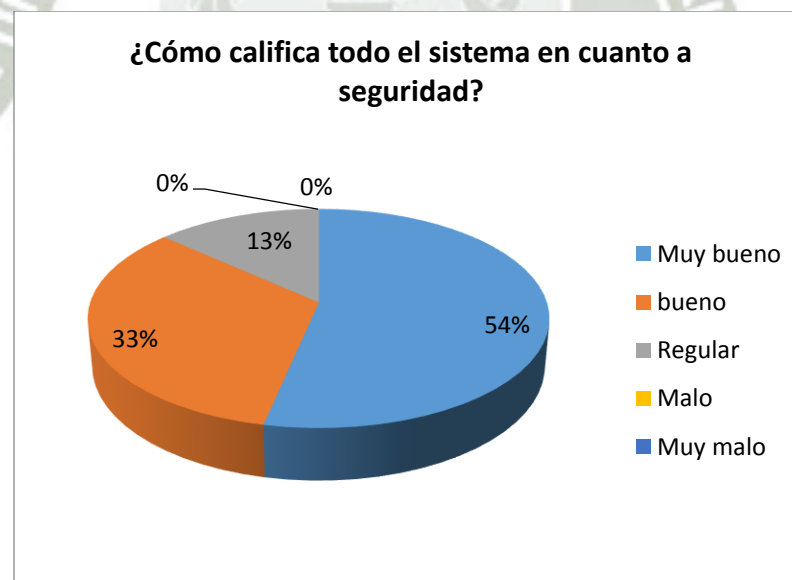


Figura 141. Grafica de resultados de la pregunta al personal

Fuente: Elaboración propia

Pregunta 9:

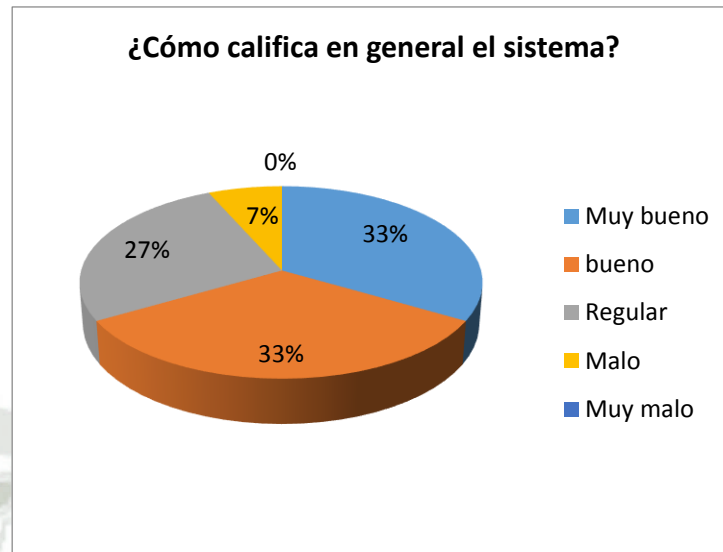


Figura 142. Gráfica de resultados de la pregunta al personal

Fuente: Elaboración propia

En conclusión, el sistema fue tanto del agrado como desagrado para el personal. En cuanto al uso de la aplicación, no les pareció difícil, por el contrario, no vieron muchas opciones, nada difícil de manejar, pocos clics, etc.

En cuanto a la interfaz, la mayoría indicó que estaba bien; algunos por el contrario indicaron que algunas letras estaban confusas, algunos botones requerían cambios, más que todo detalle estético; pero comparándolo con las botoneras, indicaron que el uso de cámaras era lo ideal y su conexión al sistema web les parecía interesante e innovador. En cuanto a su uso, se les indicó que para darles la alerta de una atención, sería bueno tener a un operador del sistema que se dedique a manejarlo, además de ver las alertas y dar aviso; esta fue justamente una de las recomendaciones para mejorar el sistema a futuro ya que eso implicaría contratar a una persona extra o capacitar a un miembro del personal para su manejo, y generaría costos adicionales al hospital y lo ven poco factible que se acceda a la contratación de personal extra, además que no ven el

registro de atenciones como algo necesario por ahora ya que se sentirían vigilados; sin embargo, los doctores y jefes del cuerpo médico que también fueron encuestados, se mostraron a favor de que sea así y consideraron que la contratación de un operador no sería complicado, además que los reportes les parecieron que se podrían mejorar pero a la vez muy útiles para saber cómo trabaja su personal médico. Se les indicó que también se vigilaría al operador ya que cada cambio realizado se registraría en la base de datos, a la cual solo el administrador o jefe del área de sistemas del hospital tendría acceso y generaría los reportes, esa medida les pareció lo ideal ya que la seguridad y veracidad de los registros es lo importante para los jefes del cuerpo médico. Para finalizar, brindaron una calificación general del sistema la cual fue calificada por la mayoría como Muy bueno y Bueno, un 27% indicó que se podría mejorar varias cosas para que se considerara como bueno, y un 7% indicó que no les gusta el sistema en sí ya que no les agrada el uso de un sistema así para las atenciones a pacientes, alegando que todos están cómodos con la forma actual de atender a los pacientes, a pesar de que según las encuestas realizadas al inicio del proyecto indican lo contrario.

Conclusiones

- PRIMERA:** Se desarrolló un prototipo del Sistema de reconocimiento de señales con la finalidad de mejorar la atención a los pacientes hospitalizados de un nosocomio
- SEGUNDA:** Se exploró las tecnologías y herramientas actuales que nos brinda la comunidad para lograr definir las indicadas para este proyecto
- TERCERA:** Se identificó las herramientas tanto de software y hardware para desarrollar el análisis de imagen y movimiento
- CUARTA:** Se definió los requerimientos del sistema mediante un trato directo con personal del área, aplicando encuestas y entrevistas con algunas autoridades del Hospital I, Edmundo Escomel Paucarpata, para tener los requerimientos necesarios y concretos
- QUINTA:** Se desarrolló una plataforma web el cual es capaz de procesar, administrar y almacenar la información obtenida por las cámaras, mejorando los módulos propuestos
- SEXTA:** Se cumplió con la arquitectura del sistema definido, garantizando así que el sistema esté bien documentado en cuanto a desarrollo, estructura y manejo de datos; la cual asegura la conectividad del Raspberry Pi con el sistema web
- SEPTIMA:** Se verificó mediante las pruebas y resultados, la funcionalidad del proyecto; comprobando así que el código desarrollado en el Raspberry Pi y el sistema web trabajan en conjunto los datos que manejan,

manteniendo la baja dependencia y escalabilidad aparte entre ambas partes del sistema.



Recomendaciones

PRIMERA: Aplicar la video vigilancia:

Se podría aplicar video vigilancia al sistema para mejorar el registro de datos y tener mayores pruebas sobre las atenciones; en este caso no se aplicó, debido a que se plantea implementar a futuro esta tesis, iniciando en la ciudad de Arequipa y extendiendo a futuro el software. Los hospitales no tienen el suficiente equipo para un flujo de datos tan pesado como el de video, y adquirir esos equipos es un costo grande para los centros de salud, se puede implementar unas capturas de imagen, pero garantizaría mayor confianza y seguridad la implementación en la video vigilancia y que muestre el video en vivo a través de la web

SEGUNDA: Mejorar la funcionalidad del sistema:

Se planea a futuro mejorar el sistema web y el programa de Raspberry Pi para que se tenga mayores funciones y no solo la de una cámara, sino que sea más intuitiva con el usuario, añadiendo más controladores y sensores para facilitar el control del paciente y a la vez la interacción del paciente con el Raspberry Pi; además, se puede añadir una cámara web para tener una mejor calidad de imagen.

TERCERA: Funcionamiento nocturno:

El software funcionaría de manera correcta si se consiguiera el último modelo de Cámara Megapíxel v2 de Raspberry Pi, ya que este cuenta con infrarrojo que mejoran la visión durante la noche. Actualmente, la cámara que se tiene funciona con un ambiente iluminado, si se apagan las luces del cuarto o es de noche, no reconocería ningún factor de movimiento, ni de color, solo estaría

esperando el sistema a que se aclare la imagen lo suficiente para que pueda trabajar.



Referencias Bibliográficas

- AFORGE.NET (2012), *Aforge.NET*, Cuba.plataform, Recuperado de: http://www.aforgenet.com/framework/samples/computer_vision.html
- Arduino (2016), *Getting Started with Arduino and Genuino products*, ARDUINO.cc, Recuperado de: <https://www.arduino.cc/en/Guide/HomePage>
- Bellamy, Benjamin (2013) *Arduino vs. Raspberry Pi vs. CubieBoard vs. Gooseberry vs. APC Rock vs. OLinUxino vs. Hackberry A10*, Techwatch, recuperado de: <http://techwatch.keeward.com/geeks-and-nerds/arduino-vs-raspberry-pi-vs-cubieboard-vs-gooseberry-vs-apc-rock-vs-olinuxino-vs-hackberry-a10/>
- Django (2016), *Meet Django*, DjangoProjects, Recuperado de: <https://www.djangoproject.com/>
- Django (2016), *Testing in Django*, Django, Recuperado de: <https://docs.djangoproject.com/en/1.11/topics/testing/>
- Doutel Fernando (2013), *Cubieboard: Primeras impresiones del rival de la Raspberry Pi*, Xatakahome, Recuperado de: <https://www.xatakahome.com/trucos-y-bricolaje-smart/cubieboard-primeras-impresiones-del-rival-de-la-raspberry-pi>
- Duda Richard O., Hart Peter E., Störk David G. (2001) *Pattern classification (2ª edición)*, Wiley, New York, ISBN 0-471-05669-3
- ETH Zurich (2007), *IGesture, Global Information Systems Group*, IGesture, Recuperado de: <http://www.igesture.org/apidocs/index.html>
- Feipeng Abaya Wilson, Basa Jimmy, Sy Michael, C. Abad Alexander, P Dadios Elmer (2014). Low cost smart security camera with night vision capability using Raspberry Pi and OpenCV, *IEEE/HNICEM/ISCII 2014*, At Hotel Centro, Puerto Prinsesa, Palawan, Philippines
- González Iván, González Juan, Gómez-Arribas Francisco (2003), Hardware libre: clasificación y desarrollo de hardware reconfigurable en entornos GNU/Linux, Escuela

Politécnica Superior, Universidad Autónoma de Madrid, España, Recuperado de:

<http://www.learobotics.com/personal/juan/publicaciones/art4/html/index.html>

Granville-Smith-Mikesh.(1982). Trigonometría plana y esférica. *UTEha, México D.F.* ISBN 968-438-774-1

Ian K. T. Tan , Poo Kuan Hoong, Chee Ken Hong, Low Zhi Wen (2014), Towards the Implementation of an Ubiquitous Car Park Availability Detection System, *Volume 301*, pages 875-884

Jiang Junhao, Ma Junji, Jin Yiye (2012), Computer Music Controller Based On Hand Gestures Recognition Through Web-cam, *EE 368 Final Project Proposal, Spring 2012*, Stanford

Leiva, Lucas. Acosta, Nelson. Vazquez Martin (2015) Herramienta para diseño automático de arquitecturas a medida basadas en redes neuronales para reconocimiento de patrones visuales, *Facultad de Ciencias Exactas, INCA/INTIA, UNCPBA*, Argentina

Merin Baby Rinu, Razmid Ahamed Rooha (2014), Optical Flow Motion Detection on Raspberry Pi, *Fourth International Conference on Advances in Computing and Communications*, Dwarahat

Nana Ramadijanti, Setiawardhana, Raga Mukti Alhaqqi (2013), TRACKING JARI DENGAN HAAR CASCADE DAN FILTER KALMAN PADA VIRTUAL KEYBOARD, *Jurusan Teknik Informatika Politeknik Elektronika Negeri Surabaya Kampus PENS Keputih Sukolilo Surabaya 60111*

OpenCV.org (2016), *OpenCV*, Itseez, Recuperado de: <http://opencv.org/>

Raspberry Pi.org (2016), *About Us*, Raspberry Pi, Recuperado de: <https://www.raspberrypi.org/>

Raspberry Pi.org(2014), *Camera Module*, Raspberry Pi. Recuperado de: <https://www.raspberrypi.org/products/camera-module/>

SATYA MALLICK (2015), *OpenCV (C++ vs Python) vs MATLAB for Computer Vision*, *Learn OpenCV*, Recuperado de: <https://www.learnopencv.com/opencv-c-vs-python-vs-matlab-for-computer-vision/>

Shapiro, Linda G., Stockman George C. (2001). *Computer Vision*. Prentice Hall. ISBN 0-13-030796-3.

Soetedjo Aryuanto, Mahmudi Ali, Ashari M. Ibrahim, Ismail Nakhoda Yusuf (2014), *Raspberry Pi Based Laser Spot Detection*, 2014 IEEE International Conference on Electrical Engineering and Computer Science, Bali, Indonesia

Sutoyo Rhio, Prayoga Bellinda, Fifilia, Suryani Dewi, Shodiq Muhsin (2015) , *The Implementation of Hand Detection and Recognition to Help Presentation Processes*, International Conference on Computer Science and Computational Intelligence (ICCSICI 2015), Procedia Computer Science, pages 550 – 558

Suzuki, S. and Abe, K. (1985), *Topological Structural Analysis of Digitized Binary Images by Border Following*. Vol. 30 Issue 1, pag. 32-46

UzeBox (2016), *UzeBoxProject*, Belogic, Recuperado de: <http://belogic.com/uzebox/index.asp>

Velazco J.J. (2013), *5 proyectos de Hardware Libre*, HiperTextual.com, Recuperado de: <https://hipertextual.com/2013/05/5-proyectos-de-hardware-libre-para-conocer>

Vipul Sharma (2014), *Gesture-OpenCV*, Github, Recuperado de: <https://github.com/vipul-sharma20/gesture-opencv>

Zenaida Condori (2016). *“Hospitales de Arequipa devuelven 14 millones que debían gastar en medicinas”*, La Republica, Arequipa, Perú, Recuperado de (25-01-2016): <http://larepublica.pe/impres/sociedad/734693-hospitales-de-arequipa-devuelven-14-millones-que-debian-gastar-en-medicinas>

Apéndices

Anexo A

Glosario de Términos

Raspberry Pi: Hardware desarrollado para entender el funcionamiento de un computador en la Fundación Raspberry Pi, la cual es un computador en un tablero de tamaño minimizado

CV: Son las siglas de Computer Vision, es una ciencia que incluye métodos para poder adquirir, interpretar, analizar y dar resultados de tipo numérico o simbólico que pueden ser tratados por un computador

BGR: Son las siglas de Blue-Green-Red, representa una estructura de números enteros de 0-255 en cada color, es decir (0-255) B, (0-255) G, (0-255) R. Existe otras siglas RGB, esto solo altera el orden de enteros, indicando que el Rojo pasa primero y al final el Azul

HSV: Es un modelo de colores, su significado es Hue, Saturation, Value. Se representa en un cono donde el borde de la base interpreta la matiz, la saturación el borde del cuerpo y el valor es definido, dependiendo el nivel de altura del cono

LPF y HPF: Son las siglas de Low Pass Filter y High Pass Filter, que se usa para interpretar el desenfoque que debe tener una imagen, se utiliza en el Open CV

Numpy: Es una librería de Python para manejar Arrays de manera más sencilla y fácil, es utilizada para el manejo de arrays en Open CV, siendo un requisito para su instalación y funcionamiento de la librería de manejo de imágenes

LGPL: Licencia Publica General en español. Es una licencia generada por la Free Software Foundation a mediados de los 80. Está orientada, Principalmente, a proteger la libre distribución, modificación y uso de software.

LED: Light-Emitting Diode, es un diodo que emite luz en el color según sus materiales de fabricación. Es activado a través de la electroluminiscencia de los semiconductores que lo conforman.

KQML: Es un lenguaje y protocolo de comunicación usado por agentes de software en sistemas basados en el conocimiento

FIPA: Fundación de Agentes Físicos Inteligentes, Es una propuesta de lenguaje estándar para comunicación de agentes de comunicaciones

BC: Una base de conocimiento es un tipo especial de base de datos para la gestión del conocimiento. Provee los medios para la recolección, organización y recuperación computarizada de conocimiento.

Puerto CSI: Camera Serial Interface, es lo que define la interface entre la cámara y el procesador de un hardware