



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS
ELECTRÓNICA E INDUSTRIAL**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
COMUNICACIONES**

TEMA:

**“SISTEMA ELECTRÓNICO PARA ASISTIR A PACIENTES EN PROCESO
DE REHABILITACIÓN DE ÚLCERAS POR PRESIÓN”**

Trabajo de Graduación. Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero en Electrónica y Comunicaciones.

SUBLÍNEA DE INVESTIGACIÓN: Procesamiento digital de señales e
imágenes

AUTOR: Hugo Daniel Aguagüiña Pilataxi

TUTOR: Ing. Cesar Granizo, Mg.

**AMBATO – ECUADOR
OCTUBRE 2018**

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de investigación sobre el tema: “SISTEMA ELECTRÓNICO PARA ASISTIR A PACIENTES EN PROCESO DE REHABILITACIÓN DE ÚLCERAS POR PRESIÓN”, del señor Hugo Daniel Aguagüiña Pilataxi estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato, Octubre 2018.

TUTOR



Ing. Cesar Granizo, Mg.

AUTORÍA DEL TRABAJO

El presente Proyecto de Investigación titulado: “SISTEMA ELECTRÓNICO PARA ASISTIR A PACIENTES EN PROCESO DE REHABILITACIÓN DE ÚLCERAS POR PRESIÓN”, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, Octubre 2018.

AUTOR



Hugo Daniel Aguagüiña Pilataxi

CC: 1804785531

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato, Octubre 2018.

AUTOR



Hugo Daniel Aguagüiña Pilataxi

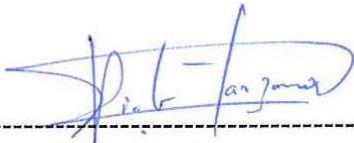
CC: 1804785531

APROBACIÓN DE LA COMISIÓN CALIFICADORA

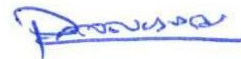
La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Santiago Manzano e Ing. Patricio Encalada, revisó y aprobó el Informe Final del Proyecto de Investigación titulado “SISTEMA ELECTRÓNICO PARA ASISTIR A PACIENTES EN PROCESO DE REHABILITACIÓN DE ÚLCERAS POR PRESIÓN”, presentado por el señor Hugo Daniel Aguagüiña Pilataxi de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.



Ing. Pilar Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL



Ing. Santiago Manzano, Mg.
DOCENTE CALIFICADOR



Ing. Patricio Encalada, Mg.
DOCENTE CALIFICADOR

DEDICATORIA

Principalmente a Dios, por ser el inspirador y darme fuerzas en este proceso de obtener uno de mis anhelos más deseados.

A mis padres Víctor y Patricia, por darme el apoyo necesario en cada etapa de mi vida, por su amor, trabajo y sacrificio en todos estos años, por educarme con valores y principios para ser una persona de bien.

Daniel Aguagüña

AGRADECIMIENTO

A Dios por ser mi fortaleza, por protegerme y darme fuerzas para superar los obstáculos a lo largo de mi vida.

A mi familia por sus palabras de aliento que han sido un soporte en los buenos y malos momentos.

Al Ing. Cesar Granizo por su tiempo, apoyo y paciencia brindada a lo largo del presente proyecto.

Al Hogar de Ancianos Sagrado Corazón de Jesús de la Ciudad de Ambato por darme la oportunidad de realizar mi proyecto de tesis.

Daniel Aguagüña

ÍNDICE GENERAL

CONTENIDOS	PÁGINAS
TEMA:	i
APROBACIÓN DEL TUTOR.....	ii
AUTORÍA DEL TRABAJO	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DE LA COMISIÓN CALIFICADORA	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL.....	viii
ÍNDICE DE TABLAS	x
ÍNDICE DE FIGURAS.....	xi
RESUMEN.....	xiii
SUMMARY	xiv
GLOSARIO DE TERMINOS Y ACRONIMOS.....	xv
INTRODUCCIÓN	xvi
CAPÍTULO I.....	1
EL PROBLEMA	1
1.1. Tema.....	1
1.2. Planteamiento del Problema.....	1
1.3. Delimitación.....	2
1.3.1. Delimitación de Contenido	3
1.3.2. Delimitación Espacial	3
1.3.3. Delimitación Temporal	3
1.4. Justificación.....	3
1.5. Objetivos	4
1.5.1. Objetivo General	4
1.5.2. Objetivos Específicos.....	4
CAPÍTULO II	5
MARCO TEÓRICO.....	5
2.1. Antecedentes Investigativos.....	5
2.2. Fundamentación Teórica.....	7
2.2.1. Fisioterapia.....	7
2.2.2. Úlceras por Presión (UPP)	8
2.2.3. Etiopatogenia de las Úlceras por Presión.....	8
2.2.4. Localizaciones más frecuentes de las Úlceras por Presión	12
2.2.5. Medidas Preventivas	13
2.2.6. Metodologías fisioterapéuticas en la rehabilitación de UPP.....	14
2.2.7. Procesamiento Digital de Imágenes	20
2.2.8. Visión Artificial	20
2.2.9. Hardware para desarrollo de Visión Artificial	23
2.2.10. Lenguajes de Programación para Visión Artificial.....	25
2.2.11. Algoritmos y librerías para Visión Artificial	29

2.2.12. Telegram	34
2.3 Propuesta de Solución	34
CAPÍTULO III	35
METODOLOGÍA	35
3.1. Modalidad de la Investigación	35
3.2. Población y Muestra.....	36
3.3. Recolección de Información	36
3.4. Procesamiento y Análisis de Datos	36
3.5. Desarrollo del Proyecto.....	36
CAPÍTULO IV	38
DESARROLLO DE LA PROPUESTA.....	38
4.1. Introducción	38
4.2. Análisis de Factibilidad.....	39
4.2.1. Factibilidad Técnica	39
4.2.2. Factibilidad Económica.....	39
4.2.3. Factibilidad Bibliográfica	39
4.3. Contexto de Nomenclatura.....	39
4.4. Requerimientos de SEAPULP	40
4.4.1. Captura	40
4.4.2. Procesamiento	42
4.4.3. Salida.....	45
4.5. Componentes de SEAPULP.....	45
4.6. Diseño de SEAPULP	46
4.6.1. Detección de la Posición del Paciente.....	46
4.6.2. Diagrama de Flujo SEAPULP para detección de posición.....	49
4.6.3. Desarrollo del código en Python para detectar la posición	50
4.6.4. Detección de Movimientos del Paciente	51
4.6.5. Diagrama de Flujo SEAPULP para detección de movimiento	54
4.6.6. Desarrollo del código en Python para detectar movimientos.....	54
4.6.7. Registro y Almacenamiento de Datos de los Pacientes	55
4.6.8. Salida del Sistema SEAPULP.....	56
4.6.9. Diseño de la Interfaz	60
4.7. Diagrama de Conexión de SEAPULP	62
4.8. Confiabilidad de SEAPULP.....	63
4.9. Pruebas y Resultados	65
4.10. Presupuesto del Proyecto	68
CAPÍTULO V	76
CONCLUSIONES Y RECOMENDACIONES.....	76
5.1. Conclusiones	76
5.2. Recomendaciones.....	77
BIBLIOGRAFÍA	78
ANEXOS	84
ANEXO A: INSTALACION DE SOFTWARE.....	85
ANEXO B: MANUAL DE USUARIO	93
ANEXO C: FOTOGRAFIAS	103
ANEXO D: CODIGO DE PROGRAMACION EN PYTHON.....	108

ÍNDICE DE TABLAS

Tabla 1. Metodologías Fisioterapéuticas en la Rehabilitación de UPP	18
Tabla 2. Comparación entre Cámaras Digitales.....	41
Tabla 3. Comparación entre micro ordenadores	42
Tabla 4. Comparación entre lenguajes de programación	43
Tabla 5. Tiempo promedio de reparación TR	64
Tabla 6. Detección de paciente con iluminación de 2500 lm.....	65
Tabla 7. Detección de paciente con menor iluminación	66
Tabla 8. Frame capturado en función de la distancia.....	67
Tabla 9. Presupuesto del Proyecto	69

ÍNDICE DE FIGURAS

Figura 1. Zonas de presión [15]	10
Figura 2. Zonas de Fricción [15].....	11
Figura 3. Zonas de Cizallamiento [15]	11
Figura 4. Zonas de presión en Decúbito Supino [16]	12
Figura 5. Zonas de presión en Decúbito Lateral [16]	12
Figura 6. Zonas de presión en Decúbito Prono [16]	13
Figura 7. Detección de Rostro	20
Figura 8. Etapas en un sistema de VA [39]	21
Figura 9. Diagrama de bloques de un Sistema SVA [41]	23
Figura 10. Placa Raspberry Pi 3 [42]	23
Figura 11. Placa Arduino Uno [43].....	24
Figura 12. Placa Orange Pi [44].....	24
Figura 13. Placa BeagleBone Black [45]	25
Figura 14. Interfaz de programación Visual Studio	26
Figura 15. Interfaz de programación Python	27
Figura 16. Interfaz de programación LabVIEW	28
Figura 17. Interfaz de programación Matlab	29
Figura 18. Estructura de ejecución del algoritmo Viola-Jones [52]	30
Figura 19. Clasificación usando el método AdaBoost. [53]	30
Figura 20. Imagen en escala de Gris	31
Figura 21. Características de dos, tres y cuatro rectángulos y su posición relativa a la ventana de búsqueda [54].....	32
Figura 22. Ejemplo de Algoritmo Haar [54].....	33
Figura 23. Diagrama de bloques de SEAPULP	40
Figura 24. Sistema SEAPULP	46
Figura 25. Captura de imágenes a través de una cámara web.....	47
Figura 26. Imagen aplicado filtros de escala Gris.....	48
Figura 27. Imagen con detección de color	48
Figura 28. Imagen con aplicación de método canny.....	48
Figura 29. Imagen de reconocimiento de rostro	49
Figura 30. Diagrama de flujo para detectar al paciente	49

Figura 31. Método supresión de fondo para detectar movimiento	52
Figura 32. Método operaciones morfológicas para detectar movimiento.....	52
Figura 33. Método Extracción de blobs para detectar movimiento	53
Figura 34. Método Extracción entre el segundo y primer plano para detectar movimiento	53
Figura 35. Diagrama de Flujo para detectar movimiento	54
Figura 36. Imagen cuando detecta movimiento	55
Figura 37. BotFather para iniciar la configuración	57
Figura 38. Inicia la configuración de usuario Bot.....	57
Figura 39. Asignación de usuario bot y nombre de usuario.....	58
Figura 40. Get id para iniciar la configuración	58
Figura 41. Código Chat_ID.....	59
Figura 42. Ventana básica en Python y Tkinter	60
Figura 43. Ventana básica y etiqueta de texto	61
Figura 44. Ventana básica y botón.....	61
Figura 45. Interfaz del sistema SEAPULP	62
Figura 46. Diagrama de conexión de SEAPULP	63

RESUMEN

En el presente proyecto de investigación se implementó un prototipo electrónico para asistir a pacientes en proceso de rehabilitación de Ulceras por Presión, denominado SEAPULP; cuyo desarrollo cuenta con librerías de visión artificial, basadas en la técnica de Viola-Jones, las cuales identifican, interpretan y detectan objetos en tiempo real. El objetivo de SEAPULP, es aportar al cuidado del paciente mientras se encuentra en proceso de rehabilitación, el prototipo detecta movimientos y posiciones en decúbito supino, lateral y sedestación de los pacientes con Ulceras por Presión, además proporciona al personal de turno la ventaja de recibir una alerta mediante mensaje de texto enviado a un dispositivo móvil que cuente con la aplicación de Telegram, estas notificaciones se envían cuando se detecta movimientos del cuerpo, o a su vez detecta que el paciente no está en posición.

El sistema SEAPULP utiliza un micro ordenador Raspberry PI 3 como tarjeta principal de procesamiento, una cámara web, una LCD de 7" para indicar la salida, los cuales son los elementos necesarios para el sistema funcione de forma ya preestablecida. La tarjeta Raspberry PI 3 recibe el video del entorno a través de la cámara web que se encuentra ubicada en la parte frontal del paciente en rehabilitación a una distancia predeterminada. Esta tarjeta electrónica procesa la información recibida de la cámara para ser mostrada en la pantalla LCD, la imagen capturada del paciente se enmarca en un recuadro de color visible de acuerdo a parámetros establecidos en la programación.

En la interfaz gráfica de usuario se observa el video en tiempo real cuando el sistema detecta algún evento, además cuenta con un menú para: registrar, ingresar, iniciar monitoreo, enviar reporte, etc. Los resultados obtenidos de las diferentes pruebas realizadas con el sistema concluyeron que, es un prototipo 95% confiable que se adapta en cualquier espacio que requiera de condiciones con iluminación en un rango de 1500-2500 lúmenes, de esta manera se pretende contribuir a la innovación tecnológica y a la calidad de vida de los pacientes.

Palabras Clave: Monitoreo, Raspberry Pi, Detección Facial, Detección de movimiento, Visión Artificial, UPP.

SUMMARY

In the present research project, an electronic prototype was implemented to assist patients in the process of rehabilitation of pressure ulcers, called SEAPULP; whose development has artificial vision libraries, based on the Viola-Jones technique, which identify, interpret and detect objects in real time. The goal of SEAPULP, is to provide care to the patient while it is in the process of rehabilitation, the prototype detects movements and positions in supine, lateral position and sitting of patients with pressure ulcers, also provides the shift staff with the benefit of receiving an alert by text message sent to a mobile device that has the Telegram application, these notifications are sent when body movements are detected, or in turn detects that the patient is not in position.

The SEAPULP system uses a Raspberry PI 3 micro computer as the main processing card, a webcam, a 7 "LCD to indicate the output, which are the elements necessary for the system to work in a pre-established way. The Raspberry PI 3 card receives the video of the environment through the webcam that is located on the front of the patient in rehabilitation at a predetermined distance. This electronic card processes the information received from the camera to be shown on the LCD screen, the captured image of the patient is framed in a visible color box according to parameters established in the programming.

In the graphical user interface, the video is observed in real time when the system detects an event. It also has a menu for: registering, entering, initiating monitoring, sending a report, etc. The results obtained from the different tests carried out with the system concluded that it is a 95% reliable prototype that adapts to any space that requires lighting conditions in a range of 1500-2500 lumens, in this way it is intended to contribute to innovation technology and the quality of life of patients.

Keywords: Monitoring, Raspberry Pi, Facial Detection, Motion Detection, Artificial Vision, UPP.

GLOSARIO DE TERMINOS Y ACRONIMOS

Frame: Se denomina a un fotograma o cuadro, una imagen particular dentro de una sucesión de imágenes que componen una animación. La continua sucesión de estos fotogramas producen a la vista la sensación de movimiento, fenómeno dado por las pequeñas diferencias que hay entre cada uno de ellos.

FAR: False Acceptance Rate – Tasa de Falsas Aceptaciones.

FRR: False Rejection Rate – Tasa de Falsos Rechazos.

LabView: Laboratory Virtual Instrumentation Engineering Workbench -Laboratorio de Instrumentación Virtual de Ingeniería Workbench

MatLab: MATrix LABoratory - Laboratorio de Matrices.

Umbral: Nivel mínimo necesario para la activación o toma de decisión de alguna función.

UPP: Ulcera por Presión

VA: Artificial Visión - Visión Artificial

OpenCV: Open Source Computer Vision Library.

LM: Lúmenes

IDLE: Integrated Development Environment – Entorno de Desarrollo Integrado

RGB: Red, Green, Blue - Rojo Verde Azul.

VGA: Video Graphics Array - Arreglo Gráfico de Video.

RAM: Random Access Memory - Memoria de Acceso Aleatorio

CPU: Unidad Central de Procesamiento.

LCD: Liquid Crystal Display – Pantalla de Cristal Líquido.

OCR: Optical Character Recognition – Reconocimiento óptico de Caracter.

OMS: Organización Mundial de la Salud.

USB: Universal Serial Bus - Conductor Universal en Serie.

INTRODUCCIÓN

En la actualidad, los sistemas de visión artificial o visión por computador están ganando cada vez más importancia en los últimos años. Esta disciplina científica que permite adquirir, procesar y analizar imágenes del mundo real, ha dejado de ser un paradigma, debido a que se cuenta con disposición de diferentes lenguajes de programación de código abierto y propietario. Uno de los que destaca es OpenCV, una herramienta informática que ha facilitado gratuitamente el acceso a todo el mundo al campo de la visión artificial.

La implementación de estos sistemas de ayuda, no son por un simple tema de comodidad, sino más bien por la alta capacidad de respuesta frente algún evento. Estos sistemas inteligentes se están desarrollando con más auge en las diferentes industrias a nivel mundial. A su vez en el presente proyecto de investigación se ha decidido utilizar los sistemas de visión artificial para desarrollar un prototipo que sea capaz de detectar posiciones y movimientos de pacientes en rehabilitación con UPP, con la finalidad de alertar algún suceso al personal de servicio a través de una notificación vía mensaje de texto.

Los sistemas automáticos de asistencia a pacientes cumplen una función muy importante dentro del ámbito de seguridad, es por ello que el sistema SEAPULP pretende aportar a la mejora en la rehabilitación; de esta manera también contribuye a la disminución del número de riesgos provocados principalmente por falta de atención. A continuación se presenta los capítulos estructurados que componen esta investigación:

En el Capítulo I se describe el problema a nivel global, donde se analiza la situación de pacientes con úlceras por presión (UPP); en todo el mundo hay un gran número de personas que necesitan tecnología de asistencia pero no todos tienen acceso a ella. En este apartado también se incluye la justificación del porque se busca realizar el prototipo, además de los objetivos con los cuales se cumple el propósito del proyecto.

El Capítulo II, abarca el desarrollo del Marco Teórico, donde se presenta antecedentes investigativos de sistemas de prevención y rehabilitación de Úlceras por Presión,

además se describen aspectos teóricos para comprender los sistemas de visión por computador así como las metodologías fisioterapéuticas en la rehabilitación de UPP. Por otra parte, se analiza los diferentes lenguajes de programación para posteriormente escoger el más idóneo para el sistema.

El Capítulo III presenta la Metodología utilizada donde se muestra las diferentes técnicas de investigación para la ejecución del proyecto, además de la recolección de información necesaria en conjunto con el procesamiento y análisis de datos para finalmente enumerar las actividades que se siguieron para el desarrollo del proyecto.

El Capítulo IV presenta el Desarrollo de la Propuesta, en este apartado se analiza los requerimientos de hardware y software necesarios para el desarrollo e implementación del sistema electrónico para asistir a pacientes en rehabilitación de UPP, por otra parte se describe los algoritmos empleados en la programación para la VA y la lógica del sistema en base a diagramas de flujo, finalmente se muestran las pruebas realizadas.

Por último, en el Capítulo V se redactan las Conclusiones y Recomendaciones que se muestran como resultado de la investigación realizada. Al término de la tesis se incorpora la Bibliografía y los respectivos Anexos.

CAPÍTULO I

EL PROBLEMA

1.1.Tema

“SISTEMA ELECTRÓNICO PARA ASISTIR A PACIENTES EN PROCESO DE REHABILITACIÓN DE ÚLCERAS POR PRESIÓN”.

1.2.Planteamiento del Problema

En todo el mundo hay un gran número de personas que necesitan tecnología de asistencia pero no todos tienen acceso a ella. Según la Organización Mundial de la Salud (OMS), calcula que cada año se hospitaliza a 421 millones de personas y que, en su estancia en el hospital, estos pacientes sufren aproximadamente 42,7 millones de eventos adversos. De acuerdo con datos recientes, el 15% de la actividad y del gasto total en los hospitales de los países de la Organización para la Cooperación y el Desarrollo Económicos (OCDE) son consecuencia directa de eventos adversos; los más numerosos son la tromboembolia venosa, la úlcera y las infecciones. Se calcula que el costo global de los daños causados a pacientes asciende a miles de billones de dólares cada año [1].

Las úlceras por presión (UPP) son una patología frecuente, se manifiestan en pacientes encamados y de movilidad reducida, constituyen un importante problema por la repercusión que alcanza en el nivel de salud y la calidad de vida de los pacientes [2]. En los últimos años, por sus múltiples implicaciones las UPP se han convertido en un problema de salud pública. Distintos estudios muestran un predominio de UPP en pacientes hospitalizados entre un 8 y 28% [3].

El manejo de las UPP implica desgastes para los profesionales de enfermería, pacientes y sus familias, lo que hace necesario que se cuantifique y analice el impacto de las intervenciones que se lleven a cabo, tanto en el ámbito de la prevención como de la rehabilitación [4]. Este problema de salud no es solo individual sino también colectivo como consecuencia de las distintas implicaciones que trae para el individuo, la familia, y los cuidadores. Por consiguiente se produce gastos de elementos de curación, aumento de la estancia hospitalaria, tiempo de atención de profesionales de enfermería.

Para rehabilitar a pacientes que padecen UPP se requiere realizar cambios posturales periódicamente a efecto de prevenir la formación de dichas lesiones isquémicas. La presencia de UPP hace que las personas afectadas limiten su desempeño motor, por lo que frecuentemente requiere de la ayuda del personal profesional de enfermería para su rehabilitación. En el sector público, la mayor parte de países brinda un escaso acceso a este tipo de tecnología. Muy pocos países disponen de un programa nacional de tecnología de asistencia. Incluso en países de ingresos altos, los productos de asistencia están a menudo racionados o excluidos de los sistemas de salud, lo que obliga a usuarios y familiares a desembolsar directamente fuertes sumas [5].

Los sistemas basados en visión artificial enfocados para el monitoreo de pacientes con UPP, actualmente no se han desarrollado en su totalidad, por consiguiente en el Ecuador, directamente en la ciudad de Ambato, en el Hogar de Ancianos Sagrado Corazón de Jesús de la Ciudad de Ambato no dispone de esta tecnología. Por lo que se convierte en una desventaja en cuestión de tecnología de monitoreo de pacientes con UPP; y esto a su vez conlleva a generar gastos económicos en los pacientes hospitalizados y por consiguiente prolongue la recuperación de los pacientes afectados.

1.3. Delimitación

Este apartado del proyecto de investigación describe la delimitación de contenido, espacial y temporal.

1.3.1. Delimitación de Contenido

Área académica:	Comunicaciones.
Línea de investigación:	Tecnologías de la Comunicación.
Sublínea de investigación:	Procesamiento digital de señales e imágenes.

1.3.2. Delimitación Espacial

La presente investigación se desarrolló en el Hogar de Ancianos Sagrado Corazón de Jesús de la Ciudad de Ambato.

1.3.3. Delimitación Temporal

La presente investigación se desarrolló en el periodo académico Septiembre 2017 – Julio 2018 de acuerdo en lo establecido en el Reglamento de Graduación para obtener el Título Terminal de Tercer Nivel de la Universidad Técnica de Ambato.

1.4. Justificación

Actualmente el mundo está viviendo una gran revolución de la tecnología de la información digital, sin conocer totalmente la tecnología que nos rodea, en muchas ocasiones se hace uso de ello. Las nuevas tecnologías relacionadas con el entorno, están agilizando, optimizando y perfeccionando algunas actividades que se realiza día a día. La electrónica y las comunicaciones permiten desarrollar circuitos avanzados de transmisión y recepción de la información, que en conjunto con la visión artificial pueden mejorar los sistemas actuales de monitoreo con elementos que interactúen entre sí de forma automática, con el fin de obtener una respuesta frente a un evento con clara intención de prevenir algún suceso. Las lesiones isquémicas normalmente requieren cuidados específicos y un manejo adecuado por el profesional de enfermería.

Los cuidados de enfermería que se ofrecen a los pacientes deben estar dirigidos hacia la integración de la curación, prevención y rehabilitación, por lo tanto el profesional de enfermería debe brindar una atención eficaz, con calidad y libre de riesgos con el fin de evitar el deterioro de la integridad de la piel de los pacientes. Al considerarse tan importante la valoración integral en los pacientes, nace la necesidad de desarrollar

el Sistema Electrónico con capacidad de monitorear pacientes en proceso de rehabilitación de Úlceras por Presión, esta tecnología es totalmente adaptable en cualquier espacio que requiera de condiciones con iluminación en un rango de 1500-2500 lm, de tal manera que los beneficiarios directos del presente proyecto, son principalmente los pacientes del Hogar de Ancianos Sagrado Corazón de Jesús de la Ciudad de Ambato que padecen de UPP.

El proyecto de investigación fue factible debido a que se cuenta con el conocimiento necesario por parte del investigador, además del respaldado de diversas fuentes de información y lo fundamental, el aval de la administradora del Hogar de Ancianos Sagrado Corazón de Jesús.

1.5. Objetivos

1.5.1. Objetivo General

- Implementar un prototipo de un Sistema Electrónico para asistir a pacientes en proceso de rehabilitación de Úlceras por Presión

1.5.2. Objetivos Específicos

- Analizar las metodologías de fisioterapia consideradas al momento de ejecutar una rutina de rehabilitación a pacientes con Ulceras por Presión.
- Diseñar la interfaz de monitoreo de asistencia para la rehabilitación de Úlceras por Presión.
- Diseñar el prototipo de un Sistema Electrónico de asistencia en pacientes con Úlceras por Presión.

CAPÍTULO II

MARCO TEÓRICO

2.1. Antecedentes Investigativos

Al momento de realizar la investigación sobre Sistemas Electrónicos asociados a pacientes que padecen UPP se encontró proyectos y artículos científicos que cuentan con un cierto grado de similitud.

D. Hayn, M. Falgenhauer, J. Morak, K. Wipfler, V. Willner, W. Liebhart y G. Schreier en su artículo científico titulado “An eHealth System for Pressure Ulcer Risk Assessment Based on Accelerometer and Pressure Data” desarrollan un sistema basado en la evaluación y prevención del riesgo de úlceras por presión, donde combinan las ventajas de los sensores de presión y el acelerómetro para monitorear los factores de riesgo de UPP. Los sensores son utilizados para detectar las reposiciones del paciente que se encuentra acostado sobre el colchón. Los datos del sensor son enviados a una Tablet donde se analizan y presentan gráficamente, tales ilustraciones gráficas pueden ser útiles para los cuidadores con el fin de optimizar el cuidado de las personas con riesgo de úlcera de presión media a alta. Los resultados indicaron que el sistema es capaz de detectar movimientos de personas mientras se encuentran acostados en la cama [6].

E. Zimlichman, Z. Shinar, R. Rozenblum, S. Levkovich, S. Skiano, M. Szyper-Kravitz, A. Altman, Y. Shoenfeld y H. Amital en su artículo científico titulado “Using continuous motion monitoring technology to determine patient's risk for development of pressure ulcers” realizaron una validación inicial mediante una tecnología de monitoreo de movimiento continuo con la utilización de un sistema denominado EverOn. EverOn es un sistema de medición continua sin ningún contacto, está basado

en un sensor piezoeléctrico que se coloca debajo del colchón del paciente. Por otra parte, desarrollaron un estudio no intervencionista en dos departamentos médicos donde obtuvieron datos de movimiento registrados de los pacientes inscritos y asignaron a los pacientes una puntuación de nivel de movimiento. Los resultados indicaron puntuaciones de movimiento para la primera noche de hospitalización los cuales se correlacionaron con la escala de Norton [7].

O. Chenu, N. Vuillerme, M. Bucki, B. Diot, Y. Payan y F. Cannard en su artículo científico titulado “TexiCare: An innovative embedded device for pressure ulcer prevention. Preliminary results with a paraplegic volunteer” desarrollan un dispositivo denominado TexiCare, el cual tiene como finalidad prevenir las UPP en personas con lesión de la medula espinal. Este dispositivo se encuentra destinado a ser montado en la silla de ruedas del paciente además de que cuenta con un sensor que permite la medición de presiones en la interface entre el cojín y las nalgas. Está conectado a una unidad de tamaño de caja de cigarrillos que mide las presiones en tiempo real, estima el riesgo de sobretensiones internas y alerta al usuario de la silla de ruedas cuando sea necesario. Se evaluó el dispositivo en un voluntario parapléjico donde los resultados arrojaron malos hábitos como una postura inadecuada al mirar televisión, raras maniobras de alivio y la aparición de altas presiones anormales [8].

M. Verbunt y C. Bartnec en su artículo científico titulado “Sensing senses: Tactile feedback for the prevention of Decubitus ulcers” desarrollan un sistema el cual utiliza una estera de sensor para detectar posturas problemáticas y proporciona información táctil al usuario. Los resultados del estudio preliminar con sujetos sanos mostraron que la retroalimentación táctil es una opción viable para la retroalimentación hablada. Además el sistema puede ser utilizado para juegos de rehabilitación, pero también para la prevención de las úlceras por presión [9].

M. Falgenhauer, S. Zöschner, J. Morak, C. Schneider, M. Gugerell, W. Liebhart y D. Hayne en su artículo científico titulado “A patient centered sensor system for decubitus risk prediction and prevention” desarrollan un sistema de monitoreo dirigido a la prevención del decúbito (UPP), que consta de cinco acelerómetros 3D y cuatro

sensores de presión en donde mediante una tableta visualizaron la cantidad de movimientos de la persona y estimaron el riesgo de decúbito individual. Los resultados arrojaron que el sistema es utilizable y puede proporcionar información útil para la prevención de decúbito [10].

2.2. Fundamentación Teórica

Los siguientes numerales describen la fundamentación teórica inmersa al tratamiento de pacientes con Ulcera por Presión, además del hardware y software de visión computacional.

2.2.1. Fisioterapia

Se define a la fisioterapia como una alternativa terapéutica sin la utilización de fármacos cuyo objetivo es mejorar las condiciones de vida de personas que tienen problemas de movilidad. Por otra parte ayuda a recuperar las funciones motoras, prevenir lesiones y mitigar el dolor haciendo uso de métodos como estimulación eléctrica, ejercicios, masajes y estiramientos [11].

Técnicas de Fisioterapia

Dentro de la fisioterapia se incluye el tratamiento de las patologías del cuerpo humano con la ayuda de medios físicos como: el movimiento, la temperatura, el masaje, la electricidad, el agua, etc. Estas opciones hacen que existan posibilidades terapéuticas al alcance del fisioterapeuta, por lo que se han clasificado en categorías [11].

- Cinesiterapia: En esta categoría existen tres tipos de ejercicio, se aplica dependiendo la situación del paciente.
- Ejercicios de Buerger: Se basa en cambios de postura, es utilizada en pacientes en rehabilitación con patología circulatoria periférica.
- Ejercicios de Codman y Chandler: Adecuados para tratar limitaciones o problemas de la articulación del hombro con la única diferencia que el ejercicio Chandler aporta una mayor seguridad y comodidad al paciente.
- Electroterapia: Existen tipos de corrientes que benefician en los tratamientos.
- Corriente Galvánica: Corriente de bajo voltaje, se mantiene constante durante el periodo que permanece la sesión de fisioterapia.

- Ultrasonido: Ondas sonoras que provocan vibraciones mecánicas, generando dilataciones en la materia y compresiones que se propagan a través de ella.
- Mecanoterapia: Utilización terapéutica de aparatos mecánicos destinados a provocar y dirigir movimientos corporales.
- Masajes
- Vendajes, otros.

2.2.2. Úlceras por Presión (UPP)

Definición

Una UPP es una lesión que se encuentra en la piel, normalmente sobre una prominencia ósea, como resultado de una presión que se ejerce y se mantiene entre dos planos duros y la tolerancia de los tejidos a esta presión en combinación con cizalla [12]. Gallart define a la UPP como una “lesión de origen isquémico que se encuentra en la piel y tejidos subyacentes con pérdida de sustancia cutánea ocasionada por presión prolongada o fricción entre dos planos duros” [13].

La UPP también se puede definir como toda lesión de la piel producida cuando se ejerce una presión sobre una prominencia ósea o plano, donde provoca un bloqueo del riego sanguíneo ocasionando una degeneración rápida de los tejidos [14]. La aparición de las UPP es un proceso relacionado directamente con los cuidados proporcionados por los profesionales fisioterapeutas debido a que cuentan con experiencia en el área de cuidados en lo que se refiere a su prevención, rehabilitación, valoración o curación de este tipo de lesiones en la piel.

2.2.3. Etiopatogenia de las Úlceras por Presión

En su mayoría las úlceras se pueden formar por presión ejercida y mantenida entre dos planos duros y la tolerancia de los tejidos a ésta, cabe destacar que no es el único medio para que se produzcan ya que actúa en conjunto con otras fuerzas mecánicas externas, su entorno y factores relacionados con el paciente. Por este motivo se encuentra lo que se define como factores de riesgo [15].

Factores de Riesgo

Al igual que la mayoría de enfermedades las UPP se pueden manifestar en cierto grupo poblacional que tienen lo que se denominan factores de riesgo. Se clasifican en:

Factores de Riesgo Intrínsecos

Se deben a condiciones propias del paciente que hace que tenga mayor probabilidad de padecer UPP como: [15]

- **Edad:** se presenta en dos grupos de edad, niños lactantes y en adultos mayores. Estas lesiones isquémicas aparecen con mayor frecuencia en adultos mayores, ya que su piel es menos resistente y fina por su edad avanzada, además de que presentan una disminución del tejido subcutáneo y una pérdida de elasticidad.
- **Inmovilidad:** pacientes con enfermedades neurológicas que tengan inmovilidad o alteraciones en su aparato locomotor.
- **Incontinencia Urinaria:** la humedad antes se hablaba como un factor externo o extrínseco al paciente, en el sentido de que puede controlarse, ahora se hace referencia a este problema como propio del paciente.
- **Incontinencia Fecal:** los pacientes que padecen este problema están predispuestos al desarrollo de úlceras por presión, tanto por la humedad como por los tóxicos que contiene las heces y orina.
- **Deficiencias Nutricionales:** la mala nutrición en el paciente afecta a la cicatrización de las heridas, debido al déficit de vitamina C, esta vitamina produce alteración en la síntesis del colágeno. Para una correcta síntesis del colágeno son necesarios los oligoelementos como el cobre, hierro o zinc. Para que se produzca un proceso de cicatrización es necesario la energía y la proteína son precisas para el sistema inmunitario
- **Disminución del peso corporal:** existe una relación entre el peso disminuido y las UPP. Si un paciente tiene peso deficiente, produce una disminución en el tejido graso debido a que tiene una menor protección de la presión en las prominencias óseas.
- **Linfopenia:** los linfocitos forman parte del sistema inmunitario, ayudando a que el organismo este protegido ante agresiones. La pérdida o disminución de estos se asocia con la aparición de las UPP.

- Estado de la piel: Las pieles con afecciones como: irritaciones, sequedad, eccemas, sequedad y falta de elasticidad son las que presentan predisposición a padecer UPP.
- Nivel de conciencia: los estados de: coma, desorientación y confusión contribuyen de manera significativa produciendo un deterioro de la conciencia la cual está asociada a la pérdida de sensibilidad y la inmovilidad.
- Enfermedad de base: la aparición de determinadas enfermedades en los pacientes como la diabetes o problemas vasculares son más propensos a su desarrollo por las alteraciones de la microcirculación. Del mismo modo, en personas con trastorno neurológico, en este caso implica situaciones como la pérdida de sensibilidad y movilidad. Tumores en tratamiento quimioterápico.

Factores de Riesgo Extrínsecos

Se consideran aquellos factores externos al paciente que hace que tenga mayor probabilidad de presentar UPP, donde intervienen tres tipos de fuerzas en su formación como: [15]

- Presión: fuerza que actúa perpendicular a la piel como consecuencia de la gravedad, provocando un aplastamiento tisular entre dos planos: óseo (pertenece al paciente) y externo (cama, sillón, etc.).

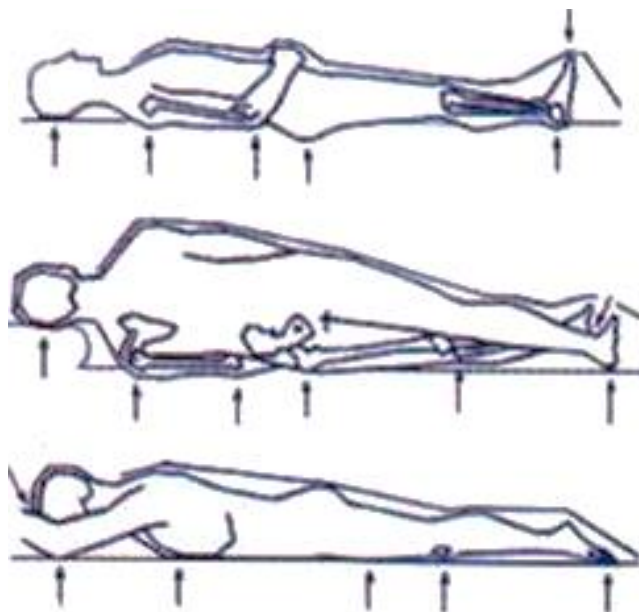


Figura 1. Zonas de presión [15]

- Fricción: fuerza tangencial, actúa paralelamente a la piel, provocando roces por arrastres o movimientos. Las fuerzas de fricción se producen en el paciente encamado o sentado en el roce con superficies rugosas o sábanas.

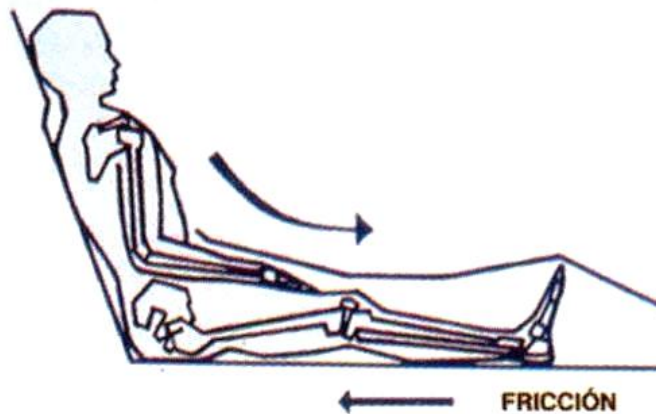


Figura 2. Zonas de Fricción [15]

- Cizallamiento: fuerzas tangenciales que combinan los efectos de presión y fricción, se ejercen en sentido contrario al desplazamiento del paciente sobre un plano duro.

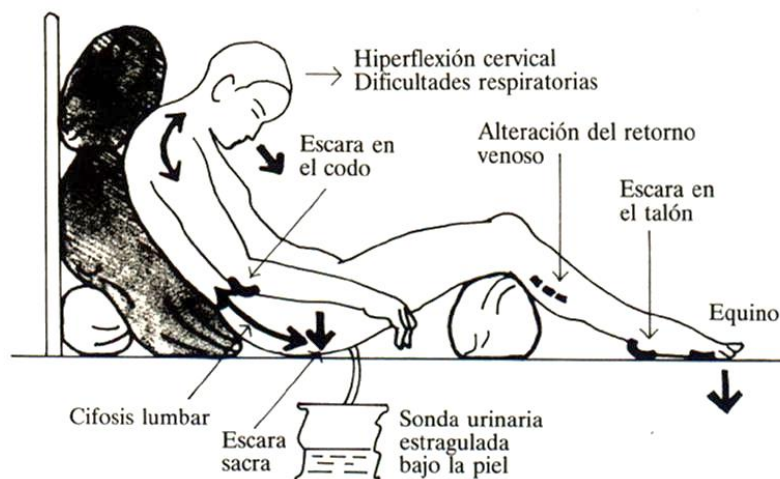


Figura 3. Zonas de Cizallamiento [15]

Por otro lado, existen factores externos al paciente que influyen en la formación de UPP como:

- Humedad: piel húmeda por orina, sudor o cualquier otro líquido aumenta el riesgo de desarrollar úlceras por presión. Otro factor de riesgo para que se desarrollen las UPP es la Incontinencia. El paciente que padece de incontinencia fecal tiene más

probabilidad de presentar una úlcera con respecto al paciente que no presenta este síntoma.

- Irritantes químicos: las heces y la orina, por sus componentes tóxicos e irritantes a la piel, contribuyen a la formación de la lesión. Otros irritantes pueden ser antisépticos, jabones inadecuados, alcohol, etc.

2.2.4. Localizaciones más frecuentes de las Úlceras por Presión

Se manifiestan con reiteración en sitios del organismo que tengan una prominencia ósea, varía de acuerdo a la posición que el paciente adopte [16].

- En decúbito supino (boca arriba): talón, codo, escapula (omoplato), occipital (parte posterior de la cabeza), sacro y pliegue interglúteo.
- En decúbito lateral: costillas, maléolos, orejas, trocánter, acromion y cóndilos.
- En decúbito prono: rodillas, acromion, dedos de los pies, mejilla, oreja. Hombres: órganos genitales. Mujeres: mamas.



Figura 4. Zonas de presión en Decúbito Supino [16]

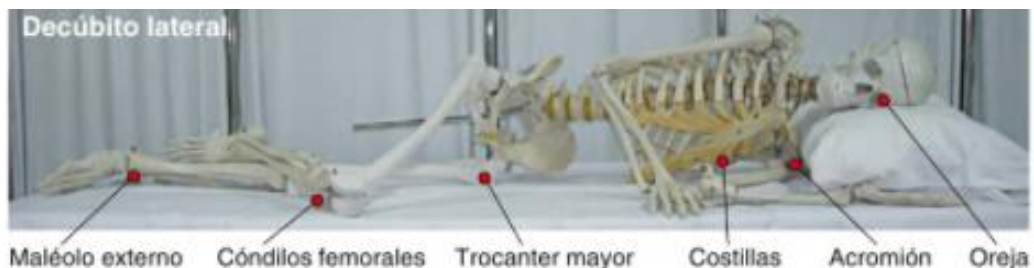


Figura 5. Zonas de presión en Decúbito Lateral [16]

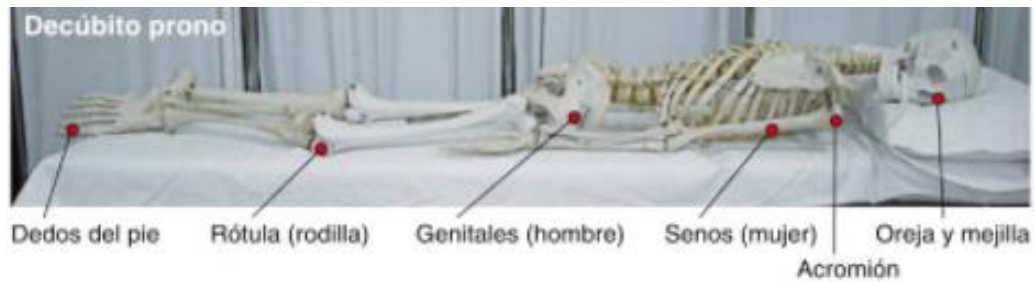


Figura 6. Zonas de presión en Decúbito Prono [16]

2.2.5. Medidas Preventivas

El modo más eficaz para tratar y cuidar a los pacientes con riesgo de desarrollar Ulceras por Presión es la prevención: [17]

Evitar y Aliviar la Presión

La disminución de la presión en las prominencias óseas se puede obtener mediante: cambios posturales (técnicas de posición), elección de una adecuada superficie de apoyo y otras específicas en función de la necesidad del paciente.

Cambios Posturales

La reposición disminuye la duración y la magnitud de la presión ejercida sobre zonas vulnerables.

- Como norma general es indispensable realizar cambios posturales cada una o dos horas de manera cíclica durante el día y como máximo, cada cuatro horas durante la noche para eliminar compresión en los puntos de apoyo [18,19].
- Reducen la magnitud y duración de la presión ejercida sobre las zonas vulnerables [18].
- La frecuencia de los cambios posturales vendrá establecido por las necesidades del paciente, el estado de la piel y la redistribución de las presiones de la superficie de apoyo [18,19].

Recomendación para cambios posturales

- No exponer la piel a la presión y cizalla, elevar y no arrastrar al paciente al momento que se realice un cambio postural [18].

- Seguir una rotación programada e individualizada de manera que alterne el decúbito supino, lateral derecho e izquierdo. Mantener: la distribución del peso, equilibrio del paciente, alineación fisiológica y corporal [18,19].
- Para evitar el contacto entre las prominencias óseas utilizar almohadas [20].
- Realizar pequeñas modificaciones dentro de la misma posición entre un cambio postural y otro [18].
- El paciente no debe ser colocado sobre prominencias óseas que presenten eritemas que no desaparecen al eliminar la presión.

Superficies especiales para el manejo de la presión

Para disminuir la presión, es fundamental reducir la duración, la magnitud, además de la fricción y el roce entre la superficie de apoyo y el paciente [21].

- Se debe utilizar colchones de espuma visco elástica (espuma de alta especificidad) en los pacientes que presenten riesgo de desarrollar úlceras por presión [22,23,24].
- El nivel de movilidad del paciente en la cama se debe tomar en cuenta al momento de seleccionar una superficie de apoyo apropiada.
- Mientras haya disponibilidad, se debe utilizar una superficie de apoyo activa.
- A pesar de que se utilice una superficie de apoyo para el manejo de la presión es necesario realizar cambios posturales siempre y cuando la situación del paciente lo permita.

2.2.6. Metodologías fisioterapéuticas en la rehabilitación de UPP

Existen diversas estrategias las cuales favorecen el proceso de rehabilitación para las UPP en donde se disminuye el tiempo de recuperación de las mismas. El tratamiento de este tipo de lesiones requiere de un manejo integral por parte del fisioterapeuta, ya que se emplean técnicas de enfermería. A continuación se presenta las metodologías aplicadas por el fisioterapeuta en la rehabilitación de UPP [25,26].

- Electroestimulación [25,27]

Esta metodología emplea propiedades de corriente eléctrica cuyo objetivo es la de favorecer los procesos de curación. Consiste en la aplicación de corriente eléctrica la cual se transmite mediante un electrodo con la finalidad de producir reacciones fisiológicas y biológicas. El electrodo se aplica directamente al lecho de la lesión

isquémica el mismo se conecta a un estimulador que se encuentra diseñado para producir una mínima carga eléctrica en los tejidos.

La ubicación del electrodo produce un flujo de corriente eléctrica que estimula a las células inactivas que se encuentran alrededor de la lesión isquémica, provocando el proceso denominado galvanotaxis con la finalidad de invadir el espacio vacío de la lesión iniciando así el proceso de cicatrización normal. El tiempo estipulado de aplicación por sesión es 20 minutos, 5 días por semana hasta completar 20 sesiones, el tratamiento puede ocasionar una ligera irritación de la piel.

- Ultrasonidos [25,28,29]

El ultrasonido se aplica alrededor de la lesión isquémica, lo cual acelera el proceso de cicatrización y reparación del tejido. El método ultrasónico transmite una vibración por medio de la piel reduciendo la probabilidad de infecciones, estimulando así el proceso de curación de las mismas. El profesional fisioterapeuta, antes de realizar el procedimiento deberá aplicar un gel conductor en el contorno de la herida del paciente de la misma manera procederá retirando el gel de manera cuidadosa con el fin de evitar que la UPP se contamine.

Esta metodología es una de las más utilizadas en el proceso de tratamiento de las lesiones isquémicas ya que sus efectos mecánicos provocan una eficiencia y una mejora del aspecto de la piel a partir de la tercera semana logrando así una muy buena regeneración de la piel en la cuarta semana, en este punto la piel muestra un alto grado de recuperación debido a que cambia el color y el tamaño del tejido de la parte afectada. Se lo debe realizar de manera pulsátil con un ciclo de trabajo del 20%, a una intensidad de 0,8-1 W/cm² y frecuencia de 3 MHz con un tiempo estipulado de 5 a 10 minutos, 3 veces por semana.

- Terapia electromagnética [25,30,31]

La Magnetoterapia se basa en dispositivos que inducen un campo eléctrico dentro de un tejido mediante la utilización de un electrodo enrollado el cual no es aplicado de manera directa al tejido sino que se lo coloca sobre el apósito. Para favorecer la regeneración tisular y la cicatrización es necesario la producción del colágeno, para

ello intervienen los campos magnéticos los mismos que producen efectos celulares, sistémicos, bioquímicos y tisulares.

La mayor parte de estos dispositivos emiten pulsos cortos no térmicos a los que se les denominan campos electromagnéticos pulsados. Con esta terapia se obtiene una acción favorable debido al aumento de la presión de oxígeno donde acelera los procesos de curación de las UPP. - La dosis para tratar una UPP son: intensidad de 25-50% y frecuencia de 50 Hz.

- Radiación Ultravioleta (UV) [25,32,33]

En lo referente a las UPP el tipo de radiación UV que se aplica a estas lesiones isquémicas son los rayos UVC, este tipo de rayos reduce el número de bacterias por su gran poder bactericida además de la cantidad de exudado purulento. La lámpara se coloca a una distancia de unos de 80 cm de manera que la radiación UVC incida perpendicularmente sobre el tejido. Lo recomendable es una aplicación diaria de esta manera mejora el aspecto de las lesiones por presión.

La Guía de Práctica Clínica (GPC) de las principales organizaciones internacionales National Pressure Ulcer Advisory Panel, European Pressure Ulcer Advisory Panel, & Pan Pacific Pressure Injury Alliance (EPUAP, NPUAP y PPIIA) recomiendan la aplicación a corto plazo de UVC en caso de que fallen la magnetoterapia, ultrasonidos y la electroestimulación debido a que este tipo de luz provoca cáncer de piel en exposiciones prolongadas.

- Laserterapia [25,32,34]

Esta metodología se viene utilizando hace más de 30 años, la aplicación de luz láser en fisioterapia ayuda en el proceso de cicatrización de las heridas. La función principal del láser es la de aumentar las tasas de regeneración epidérmica además de incrementar y estimular la proliferación de fibras para la producción del colágeno. J. Taradaj, T. Halski, M. Kucharzewski, T. Urbanek, U. Halska y C. Kucio realizaron un estudio en el cual evaluaron la eficacia de la terapia con láser en diferentes longitudes de onda para el tratamiento de las UPP, en donde concluyeron que la longitud de onda

empleada fue de 658 nm donde ofrece resultados significativamente mejores en comparación con longitudes de onda tratadas con láser a 808 nm y 940 nm.

Las UPP tratadas con longitud de onda de 658 nm consiguieron cerrarse a los tres meses de tratamiento. En la actualidad, la evidencia sobre la eficacia de la laserterapia empleada para tratar las UPP es limitada debido a que, la Guía de Práctica Clínica (GPC) de las principales organizaciones internacionales National Pressure Ulcer Advisory Panel, European Pressure Ulcer Advisory Panel, & Pan Pacific Pressure Injury Alliance (EPUAP, NPUAP y PPPIA) no recomiendan el empleo del láser, por lo que hay que hacer uso de esta técnica con cautela.

- Cambios Posturales [25,35,36]

Los profesionales fisioterapeutas cuentan con conocimientos en: movilización, cinesiterapia y biomecánica que pueden minimizar el riesgo de desarrollar UPP. La inmovilidad e inactividad se asocian a las lesiones en donde estos dos factores de riesgo son importantes para el desarrollo de UPP. Por consiguiente, es conveniente realizar con frecuencia cambios posturales con el fin de reducir la presión en las zonas vulnerables del cuerpo en donde aparece estas lesiones isquémicas mientras la persona este inmóvil.

Tanto para la prevención como la rehabilitación de las UPP se emplean cambios posturales. El plan de cuidados que realiza el fisioterapeuta para el reposicionamiento del paciente se debe realizar en referencia a las superficies en las que se sienta y acuesta incluyendo el apoyo de las extremidades como: pies, brazos y piernas. Si el paciente permanece en sedestación los cambios posturales se realizaran cada 15 minutos mientras que si se encuentra acostado se realizaran cada dos o tres horas. La frecuencia será mayor o menor dependiendo de la respuesta de cicatrización de la lesión isquémica.

Tabla 1. Metodologías Fisioterapéuticas en la Rehabilitación de UPP

Metodología	Características	Frecuencia de Aplicación
Electroestimulación	<ul style="list-style-type: none"> - Aplica corriente eléctrica la cual se transmite mediante un electrodo. - El electrodo genera un flujo de corriente eléctrica que estimula las células inactivas alrededor de la lesión isquémica. - Provoca el cierre total de la UPP. 	<ul style="list-style-type: none"> - Tiempo estipulado de aplicación por sesión 20 minutos. - 5 días por semana hasta completar 20 sesiones. - El tratamiento puede ocasionar una ligera irritación de la piel.
Ultrasonidos	<ul style="list-style-type: none"> - Transmite una vibración por medio de la piel. - Acelera el proceso de cicatrización y reparación del tejido. - Reduce la probabilidad de infecciones. - Provoca una eficiencia y una mejora del aspecto de la piel a partir de la tercera semana. 	<ul style="list-style-type: none"> - Realizar de manera pulsátil con un ciclo de trabajo del 20%, a una intensidad de 0,8-1 W/cm² y frecuencia de 3 MHz. - Tiempo estipulado de 5 a 10 minutos, 3 veces por semana.
Terapia electromagnética	<ul style="list-style-type: none"> - Inducen un campo eléctrico dentro de un tejido mediante la utilización de un electrodo enrollado. - El electrodo no se aplica de manera directa al tejido sino que se lo coloca sobre el apósito. - Se obtiene una acción favorable debido al aumento de la presión de oxígeno donde acelera los procesos de curación de las UPP. 	<ul style="list-style-type: none"> - La dosis para tratar una UPP son: intensidad de 25-50% y frecuencia de 50 Hz.

<p>Radiación Ultravioleta (UV)</p>	<ul style="list-style-type: none"> - El tipo de radiación UV que se aplica a estas lesiones isquémicas son los rayos UVC. - Reduce el número de bacterias por su gran poder bactericida además de la cantidad de exudado purulento. - Mejora el aspecto de las lesiones por presión. 	<ul style="list-style-type: none"> - La lámpara se coloca a una distancia de unos de 80 cm de manera que la radiación UVC incida perpendicularmente sobre el tejido. - Lo recomendable es una aplicación diaria. - Se recomienda la aplicación a corto plazo debido a que este tipo de luz provoca cáncer de piel en exposiciones prolongadas.
<p>Laserterapia</p>	<ul style="list-style-type: none"> - Aumenta las tasas de regeneración epidérmica. - Incrementa y estimula la proliferación de fibras para la producción del colágeno. - Ayuda en el proceso de cicatrización de las heridas. 	<ul style="list-style-type: none"> - Aplicar con una longitud de onda de 658 nm para mayor efectividad. - La Guía de Práctica Clínica (GPC) de las principales organizaciones internacionales: EPUAP, NPUAP y PPIIA) no recomiendan el empleo del láser, por lo que hay que hacer uso de esta técnica con cautela.
<p>Cambios Posturales</p>	<ul style="list-style-type: none"> - Se emplea para prevención y rehabilitación de las UPP. - Realizar con frecuencia cambios posturales con el fin de reducir la presión en las zonas vulnerables. - El reposicionamiento se debe realizar en referencia a las superficies en las que se sienta y acuesta incluyendo el apoyo de las extremidades como: pies, brazos y piernas. 	<ul style="list-style-type: none"> - En sedestación los cambios posturales se realizaran cada 15 minutos. - Acostado se realizaran cada dos o tres horas. - La frecuencia será mayor o menor dependiendo de la respuesta de cicatrización de la lesión isquémica

Elaborado por: Investigador basado en [25-36]

2.2.7. Procesamiento Digital de Imágenes

El procesamiento de imágenes se encuentra orientado a obtener imágenes mejoradas o modificadas con efectos especiales a través de herramientas de programación destinadas a un propósito específico. Asimismo, el procesamiento de imágenes tiene como finalidad mejorar y hacer más evidentes ciertos detalles que se desean hacer notar en las imágenes. Este tipo de procesamiento se lo puede hacer mediante métodos ópticos y a través de métodos digitales en una computadora [37].

Utilidad del Procesamiento de Imágenes

- Su uso es muy amplio y abarca demasiados campos, un ejemplo son aquellas imágenes obtenidas con fines de diagnóstico médico.
- Otro ejemplo son las imágenes obtenidas para realizar exámenes del terreno, a través de este método se analizan las fallas geológicas del terreno, los recursos naturales, etc [37].

2.2.8. Visión Artificial

La visión artificial, también denominada visión por computador es una técnica que se basa en la adquisición de imágenes que, mediante un procesamiento computacional permite el procesamiento, análisis y obtención de cualquier tipo de información obtenida a través de imágenes digitales. Se trata de una tecnología en donde combina cámaras de video con computadores con el fin de interpretar de forma a la inspección visual humana. Por tanto pretende capturar la información visual obtenida del entorno físico para después extraer características visuales relevantes [38].



Figura 7. Detección de Rostro
Elaborado por: Investigador

Etapas de un Sistema de Visión Artificial

En un sistema de visión artificial se presentan cuatro fases diferenciadas en el proceso de imágenes, como son: [39]

Primera fase: realiza la adquisición o captura de imágenes por medio de una cámara digital, estas imágenes son almacenadas en la memoria de un ordenador como un arreglo o matriz de puntos, llamados píxeles.

Segundo fase: tiene por objetivo el tratamiento de las imágenes. Se lo realiza mediante filtros digitales transformando a la imagen original a una imagen de escala gris para ayudar al posterior análisis.

Tercera fase: segmentación, consiste en particionar una imagen en regiones homogéneas con el fin de extraer características de dichas regiones (textura, color, forma, contorno, etc.).

Cuarta fase: lleva a cabo el reconocimiento o clasificación. Emplea el análisis de ciertas características que se establecen previamente en el entrenamiento de imágenes para detectar los objetos deseados.

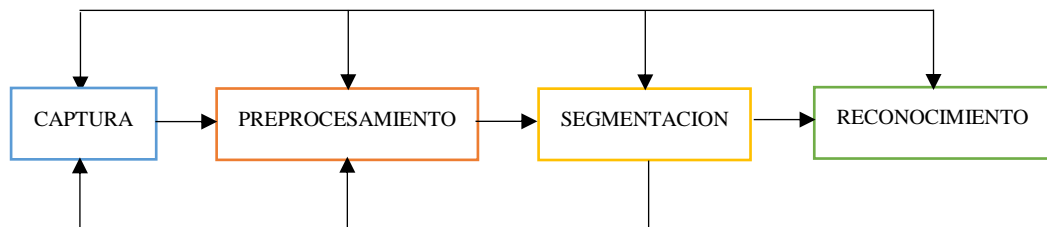


Figura 8. Etapas en un sistema de VA [39]

Ventajas de la Visión Artificial

La visión artificial trabaja fuera del espectro visible reconociendo detalles que no son visibles para el ojo humano en donde ejecutan tareas repetitivas con rapidez y precisión. Además cuenta con ventajas en tareas que necesitan un alto grado de repetición y grandes velocidades, un claro ejemplo son los procesos de inspección visual de una línea de ensamblaje. Las ventajas que nos ofrece la visión artificial son: [40]

Aumenta la producción

- Reducción de costos

- Eficiencia en tiempo de producción
- Bajos costos de implementación
- Medición sin contacto

Evita errores humanos

- Errores visuales
- Falta de atención
- Ausencia laboral
- Utilizado en lugares inaccesibles

Desventajas de la Visión Artificial

Los sistemas que utilizan visión artificial poseen desventajas considerando la capacidad visual de los ojos del ser humano, los cuales se evalúan de acuerdo al nivel de implementación de los diferentes sistemas, entre las cuales tenemos: [40]

- Menor velocidad de interpretación en comparación con el ser humano.
- Problemas con la iluminación.
- Sistemas en desarrollo e investigación.

Componentes de un Sistema de Visión

Los elementos de hardware necesarios para un sistema de visión artificial (SVA) son los siguientes: [41]

Sensor óptico: Una cámara puede ser un sensor óptico, su principal función es capturar la imagen proyectada en el sensor para después transferir información a un sistema electrónico. En visión artificial las cámaras requieren de características como la posición requerida en el disparo para capturar piezas que pasan por delante de ella.

Fuente de iluminación: La principal función de la iluminación es la de controlar la forma en que la cámara va a ver el objeto, esta es la parte más crítica dentro de un sistema de visión artificial debido a que la luz se refleja de diferentes formas dependiendo del objeto de interés. Por tanto, la iluminación se debe ajustar de acuerdo al objeto que se va a iluminar.

Computador: Compuesto por software y hardware, la tarjeta de adquisición de imagen se encarga de digitalizar la señal de video además que recibe, procesa y analiza los datos mediante algoritmos de análisis.

Interfaz Gráfica: Permite visualizar las escenas captadas como los resultados del procesamiento de las imágenes.

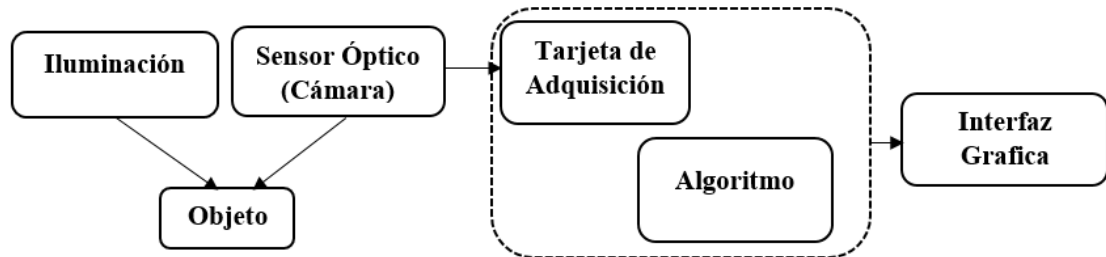


Figura 9. Diagrama de bloques de un Sistema SVA [41]

2.2.9. Hardware para desarrollo de Visión Artificial

Raspberry Pi 3

La tercera generación con un procesador ARMv8 de cuatro núcleos y 64bits, es más rápido y capaz que su predecesor debido a que trabaja a una frecuencia de 1.2 Ghz. Esta generación cuenta con mejores características que sus versiones anteriores como son: 1 GB de RAM, 4 puertos USB para conexión de diferentes periféricos, salida HDMI, interfaz GPIO de 40 pins, slot para micro SD, puerto ethernet, VideoCore IV 3D graphics, Jack de audio, interfaz para pantallas externas y cámara, además esta versión es compatible con internet inalámbrico ya que viene incorporado con Wi-Fi y Bluetooth [42].



Figura 10. Placa Raspberry Pi 3 [42]

Arduino

Arduino es una plataforma electrónica de hardware y software de código abierto, basada en una placa con un microcontrolador con entradas y salidas (E/S), analógicas y digitales en un entorno de desarrollo que implementa lenguaje de programación Processing. Se encuentra diseñado para utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin necesidad de licencia [43].



Figura 11. Placa Arduino Uno [43]

Orange Pi

Es una computadora de una sola placa de código abierto basada en Raspberry Pi desarrollada por Shenzhen Xunlong Software CO., Limited. Puede ejecutar Ubuntu, Android, Debian, Raspbian, Fedora, entre otros sistemas operativos. Cuenta con un procesador AllWinner H2, H3 y H5 SoC, A64 Quad-core Cortex-A53 64bit o ARM Cortex-A5 32bit, además en placas pequeñas posee una memoria de 256MB y en placas mayores de 2GB DDR3 SDRAM de RAM [44].



Figura 12. Placa Orange Pi [44]

BeagleBone Black

Es una tarjeta electrónica de bajo costo y de código abierto con un respaldo de una comunidad de rápido crecimiento destinada a desarrolladores y aficionados. Difiere ligeramente de la versión normal ya que cuenta con algunas características eliminadas y añadidas de las cuales se tiene: memoria DRAM DDRL3 DE 512 MB, puerto integrado micro HDMI, incorpora una memoria flash de 4GB además de un procesador AM3358 a 1 GHz [45].



Figura 13. Placa BeagleBone Black [45]

2.2.10. Lenguajes de Programación para Visión Artificial

Microsoft Visual Studio

Es un entorno de desarrollo integrado (IDE) que permite crear a los desarrolladores aplicaciones, sitios, servicios y aplicaciones web, en cualquier lenguaje de programación que soporte la plataforma tales como: Java, C++, F#, PHP, Visual Basic, .NET, C#, Python y Ruby. Se encuentra disponible para los sistemas operativos Windows y Mac, soporta extensiones esta ventaja permite mejorar el desempeño en el desarrollo de sistema de información también se puede realizar depuración del código, agregar o quitar proyectos además, trabaja al mismo tiempo con distintos lenguajes de programación [46].

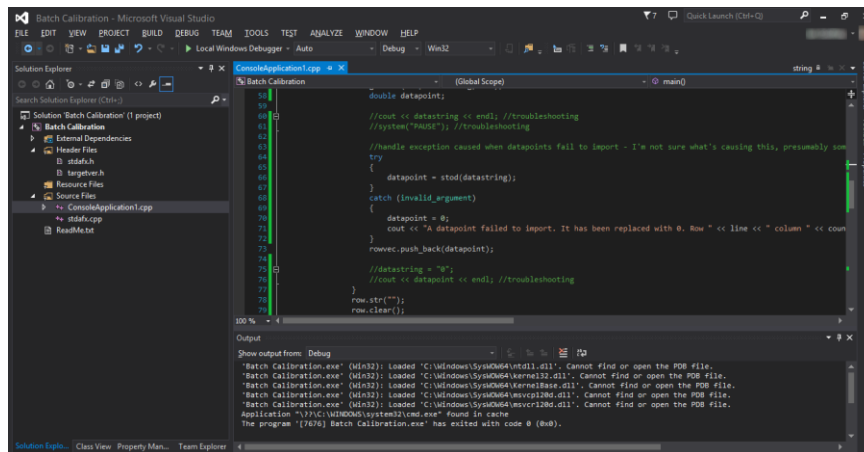


Figura 14. Interfaz de programación Visual Studio
Elaborado por: Investigador

Python

Es un lenguaje de programación de alto nivel y propósito general, destinado a objetos que puede ser utilizado para el desarrollo web. Python es un lenguaje interpretado debido a que no se necesita compilar el código fuente para poder ejecutarlo. Entre las características que se puede destacar de este lenguaje de programación son: [47]

- Propósito general: tiene la capacidad de crear todo tipo de programas, aunque no es un lenguaje destinado para la web entre sus posibilidades si se encuentra el desarrollo de páginas.
- Interpretado: quiere decir que no se compila a diferencia de otros lenguajes, sino que es interpretado en tiempo de ejecución.
- Multiplataforma: originalmente este lenguaje de programación se desarrolló para Unix, en la actualidad cualquier sistema operativo es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.
- Orientado a objetos: todo en Python es un objeto debido a que ofrece en muchos casos una manera fácil de realizar o crear programas con componentes reutilizables.
- Librerías y funciones: Existen librerías las cuales se pueden importar en los programas para temas específicos, por otra parte cuenta con funciones que se encuentran incorporadas en el lenguaje para el tratamiento de archivos, strings, números, etc.

- Sintaxis clara: el lenguaje cuenta con una sintaxis muy visual por su notación indentada, es decir que si en Python se desea separar código se realiza hacia adentro colocando un margen al código a diferencia de otros lenguajes que utilizan llaves.

```

SEAPULP.py - C:\Users\User\Desktop\SEAPULP\SEAPULP.py (2.7.9)
File Edit Format Run Options Windows Help
import numpy as np
import cv2
import pygame
import time
import sys
import os
import telepot
import datetime

s=0
msj=0
movi=0
size=5
pygame.mixer.init()
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_alt.xml')
#face_cascade = cv2.CascadeClassifier('haarcascade_profileface.xml')
cap = cv2.VideoCapture(0)
cap.set(3,400)
cap.set(4,600)
_,prev=cap.read()
prev=cv2.cvtColor(prev,cv2.COLOR_RGB2GRAY)
kernel=np.ones((5,5),np.uint8)
font=cv2.FONT_HERSHEY_SIMPLEX
PrevCen=np.array([0,0])
bot = telepot.Bot('481163786:AAHo20FR1yqjVf1dSj7zhQ1w3G1EiVFwPI')
ventana=" SEAPULP"

cv2.namedWindow(ventana, cv2.WINDOW_NORMAL)
cv2.moveWindow(ventana,0,0)
cv2.resizeWindow(ventana,800,400)
fondo=cv2.imread("imagen1.jpg",1)

#cv2.ocl.setUseOpenCL(False)
while 1:
    _,frame= cap.read()

    y=str(time.strftime("%A:%H:%M:%S"))
    formato = "%X"
    hora = time.strftime(formato) #variable tipo hora
  
```

Figura 15. Interfaz de programación Python
Elaborado por: Investigador

LabVIEW

Es un lenguaje de programación visual desarrollado por National Instruments, ofrece una programación grafica en donde se puede visualizar los aspectos de la aplicación a desarrollar, incluyendo datos de medidas, configuración del hardware y depuración. La visualización hace que sea más sencillo integrar el hardware, también representar una lógica compleja en el diagrama además de diseñar interfaces personalizadas y desarrollar algoritmos de análisis de datos [48].

Este lenguaje es similar a los sistemas de desarrollo comerciales que usan el lenguaje Basic o C y se encuentra dirigido al desarrollo de aplicaciones. Es comúnmente utilizado para control de instrumentos, adquisición de datos y automatización industrial, se encuentra disponible para los sistemas operativos Linux, Windows, versiones de Unix y macOS.

Las principales características que posee LabVIEW son: [48]

- Crea sus aplicaciones hace uso de iconos en lugar de líneas de texto.

- Utiliza programación por diagrama de bloques.
- Posee funciones básicas de todo lenguaje de programación además de extensas librerías de funciones y subrutinas.
- Posibilidad de incorporar aplicaciones escritas en otros lenguajes.
- Las librerías son específicas para diferentes ámbitos como el control de instrumentación VXI, adquisición de datos, análisis presentación y guardado de datos.
- Posee un compilador gráfico para logara la máxima velocidad de ejecución.
- Herramientas que hacen más fácil la depuración de los programas.

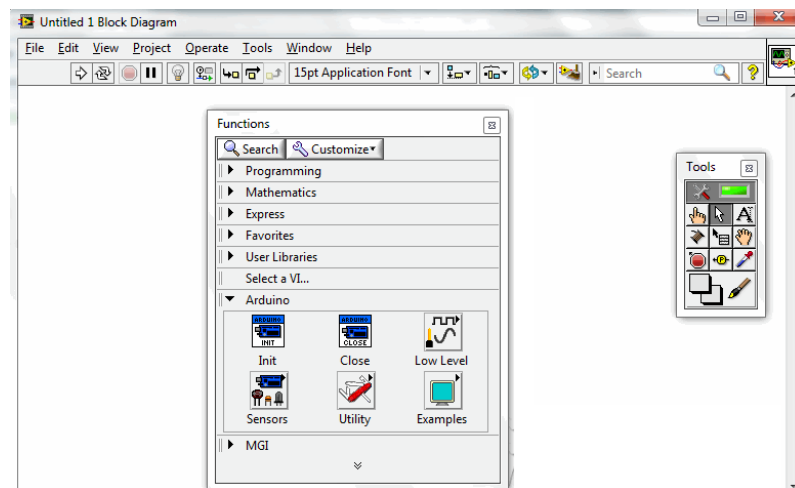


Figura 16. Interfaz de programación LabVIEW
Elaborado por: Investigador

Matlab

Es un lenguaje de programación de alto nivel y entorno interactivo que permite el trazado de funciones y datos, desarrollo de algoritmos, manipulaciones matriciales, creación de interfaces de usuario, computación numérica e interconexión con programas escritos en diferentes idiomas como Python, Fortran, C++, Java, C, C#. Es multiplataforma ya que se encuentra disponible para: Windows, Linux, Unix, Mac. Este lenguaje posee características como son: [49]

- Herramientas útiles para realizar aplicaciones con interfaces gráficas personalizadas.
- Interfaz gráfica que permite visualizar datos en dos y tres dimensiones.

- Herramientas para mejorar la calidad del código.
- Entorno interactivo para exploración iterativa, diseño y resolución de problemas.
- Integra algoritmos mediante funciones basadas en Matlab con aplicaciones externas.
- Funciones para tratar el análisis de Fourier, integración numérica, algebra lineal, resolución de ecuaciones diferenciales ordinarias, etc.
- Lenguaje de alto nivel para desarrollo y visualización de aplicaciones así como para computación numérica.

```

1 function [ x ] = Biseccion( a,b,eps,m)
2 %Este programa calcula las raices de la funcion 8*x*cos(x)-2*x^2 en un
3 %intervalo (a,b)
4 i=0;
5 while abs(b-a)>eps && (i<m)
6     i=i+1;
7     c=(b-a)/2;
8     fa=8*a*cos(a)-2*(a^2);
9     fc=8*c*cos(c)-2*(c^2);
10    if fa*fc<0;
11        b=c;
12    else
13        a=c;
14    end
15    x=(a+b)/2;
16 end
17 end

```

Figura 17. Interfaz de programación Matlab
Elaborado por: Investigador

2.2.11. Algoritmos y librerías para Visión Artificial

Algoritmo Viola – Jones

El algoritmo de Viola-Jones es una técnica ampliamente utilizada para la detección de objetos en tiempo real, su principal propiedad es que su entrenamiento es lento, pero la detección es rápida. Además este algoritmo realiza la clasificación mediante características en vez de pixel a pixel, lo que permite una cierta abstracción del algoritmo respecto al resultado. Su desarrollo fue destinado principalmente por el problema de detección de caras, pero puede ser aplicado en otra clase de objetos que, como las caras estén caracterizados por patrones típicos de iluminación [50] [51].

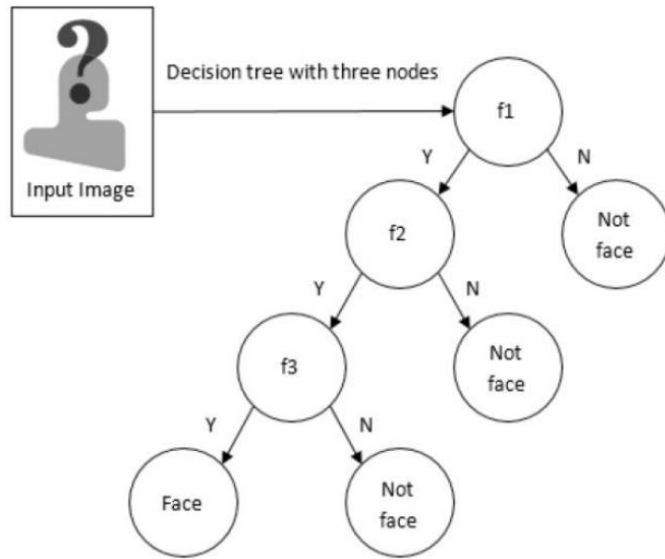


Figura 18. Estructura de ejecución del algoritmo Viola-Jones [52]

Viola-Jones utiliza una cascada de clasificadores que se ejecutan uno tras otro. Los primeros clasificadores son sencillos los cuales permiten aceptar un alto porcentaje de caras de la misma manera descartan una gran cantidad de no caras. Además, la cascada desecha gran parte de las regiones de la imagen y solamente permite concentrarse en las zonas en la que exista una cara, cada uno de estos clasificadores de cascada se entrenan con el algoritmo de boosting AdaBoost. En la figura 19 se observa la clasificación usando el método AdaBoost [53].

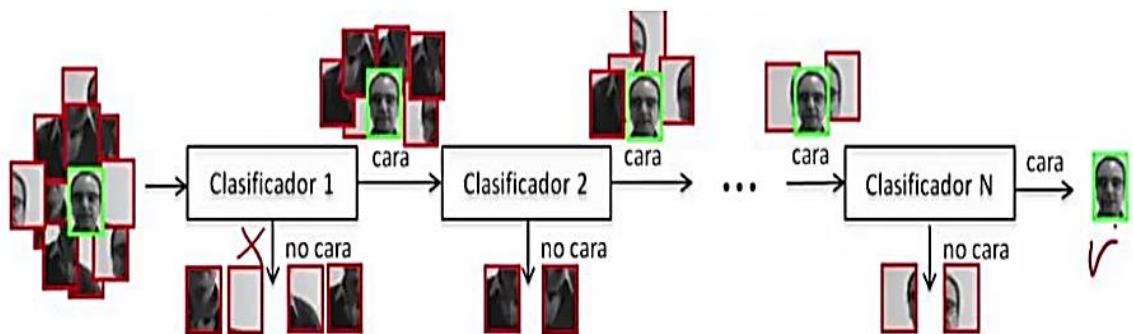


Figura 19. Clasificación usando el método AdaBoost. [53]

Clasificador Haar

Haar Classifier es un algoritmo clasificador en cascada que, reconoce la forma del objeto haciendo uso de cientos de muestras positivas y negativas, pasando por etapas diferentes que se aplican secuencialmente con un proceso de optimización. [54] Las características de este clasificador son elementos con los cuales se realiza la detección, de modo que se buscan en las imágenes y que consisten en la diferencia de intensidades luminosas entre regiones rectangulares adyacentes.

Este tipo de clasificadores definen regiones rectangulares sobre una imagen en escala de grises y al estar formada por un número finito de rectángulos, se puede obtener un valor escalar que consiste en sumar los píxeles de cada rectángulo, en base a una serie de clasificadores en cascada. Cada clasificador determina si la subregión se trata del objeto buscado o no. A diferencia de otros algoritmos, este solo invierte capacidad de procesamiento a las subregiones que posiblemente representen un rostro [54].

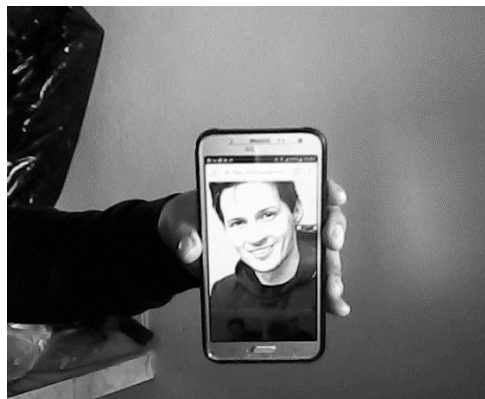


Figura 20. Imagen en escala de Gris
Elaborador por: Investigador

El algoritmo utiliza tres tipos de características representadas en la figura 21.

- Dos rectángulos: El resultado es la diferencia entre las sumas de los píxeles contenidos en ambos rectángulos.
- Tres rectángulos: Es la diferencia entre los rectángulos exteriores y el interior multiplicado por un peso para compensar la diferencia de áreas.
- Cuatro rectángulos: Es la diferencia de rectángulos que son par de su lado diagonal.

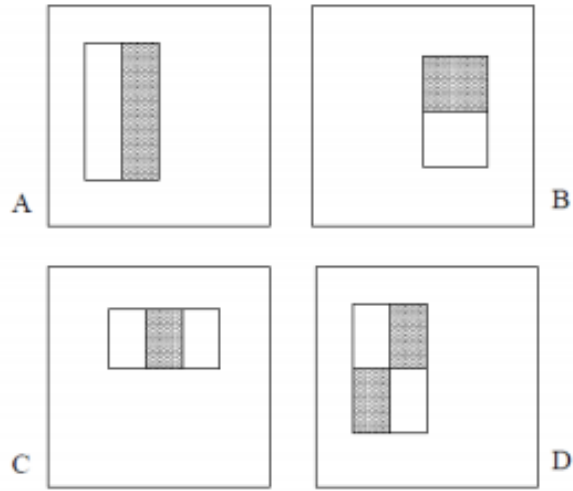


Figura 21. Características de dos, tres y cuatro rectángulos y su posición relativa a la ventana de búsqueda [54].

Para calcular de manera eficiente la suma de los píxeles de un rectángulo, se emplea la representación denominada imagen integral. Esta integral se calcula como la suma de los píxeles ubicados arriba y a la izquierda el punto (x, y) en la imagen original. La imagen integral se calcula con la siguiente ecuación:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (1)$$

donde $ii(x, y)$ es la imagen integral y $i(x', y')$ es la imagen original.

Empleando el par de ecuaciones (2) y (3). Se puede calcular en un solo barrido la imagen integral:

$$s(x, y) = s(x, y - 1) + i(x, y) \quad (2)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (3)$$

donde $s(x, y)$ es la suma acumulada de la fila x , con $s(x, -1) = 0$ y $ii(-1, y) = 0$

Con la imagen integral se puede calcular cualquier suma rectangular con cuatro referencias como se observa en la figura 24. Además, se muestra un ejemplo haciendo uso de los 4 puntos de la matriz, donde el valor de la suma de los píxeles en el

rectángulo D es igual a: el valor de la imagen integral en 4 (A+B+C+D) – valor de la imagen integral en 2(A+B) y 3(A+C) + valor de la imagen integral en 1(A).

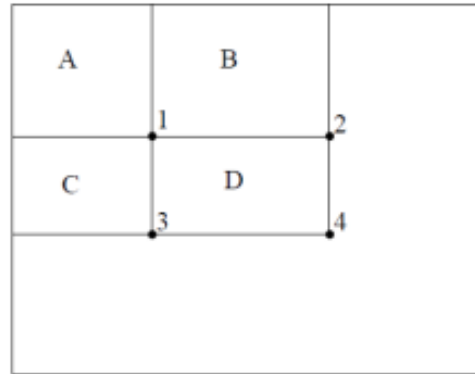


Figura 22. Ejemplo de Algoritmo Haar [54]

OpenCV

Es una librería o biblioteca de software de código abierto dedicada para visión artificial que puede ser utilizada en diferentes lenguajes de programación como C#, Objective C, Java y Python, además de que se puede ejecutar en diferentes sistemas operativos (Windows, Android, iOS, Linux y Mac OS X). Se encuentra diseñada especialmente para la captura, procesamiento y visualización de imágenes en diferentes áreas como: monitorización, reconocimiento de objetos, robótico, biométrica e interacción hombre-máquina. Al ser un producto de licencia BSD (Berkeley Software Distribution), OpenCv facilita a las empresas modificar y utilizar el código [55].

OpenCV implementa una gran variedad de herramientas para la interpretación de la imagen. La biblioteca cuenta con más de 2500 algoritmos, destinados para muchas aplicaciones como: [55]

- **Detectar y reconocer** rostros
- **Identificar** Objetos
- Rastrear el movimiento de objetos
- Eliminar los ojos rojos de las imágenes capturadas con flash.
- Clasificar las acciones humanas en videos
- Encontrar imágenes similares en una base de datos
- Seguimiento a los movimientos de la cámara
- Reconocer paisajes

- Seguir los movimientos del ojo humano
- Establecer marcas para recubrirlos con realidad aumentada
- Unir imágenes para producir en conjunto una imagen de alta resolución de una completa escena
- Inspeccionar etiquetas de los productos en las fabricas
- Producir nubes de puntos 3D de cámaras estéreo
- Extraer modelos 3D de objetos

2.2.12. Telegram

Es una aplicación de mensajería móvil dispuesta a competir con WhatsApp, desarrollada por los fundadores de VKontakte, la red social más popular en Rusia. Tras un primer vistazo, el parecido con WhatsApp es evidente, pero tiene peculiaridades importantes [56].

Plataforma abierta

Telegram dispone de una API que permite a aplicaciones de terceros conectarse a la red. Esto hace factible que haya clientes no oficiales como las versiones para Windows, Linux, Mac, web mientras que WhatsApp es una plataforma cerrada herméticamente. Además, Telegram brinda las funcionalidades para desarrollar proyectos, entre ellos se destaca la creación de un usuario `_bot`, el cual este nos proporciona un código API, y un código `chat_id`, para poder interactuar con dispositivos electrónicos [56].

2.3 Propuesta de Solución

La implementación del Sistema Electrónico para asistir a pacientes en proceso de rehabilitación de Úlceras por Presión permitió al personal de enfermería monitorear al paciente en tiempo real con la finalidad de evitar que estas lesiones isquémicas se agraven.

CAPÍTULO III

METODOLOGÍA

3.1. Modalidad de la Investigación

Investigación Aplicada

El presente proyecto tuvo una modalidad de investigación aplicada, debido a que interviene conocimientos adquiridos durante la carrera universitaria, para sustentar el desarrollo de una propuesta de ingeniería. Además, el proyecto está orientado a una solución específica de un problema que concierne a las necesidades del Hogar de Ancianos Sagrado Corazón de Jesús.

Investigación Bibliográfica-Documental

La investigación fue de modo bibliográfica-documental ya que se argumenta el tema conceptualmente a través de una estructura metodológica del proyecto, obteniendo un marco teórico enfocado al tema. Para este propósito se utilizó fuentes de información primaria y secundaria como libros, revistas, artículos científicos, tesis de grado y publicaciones electrónicas, lo cual permitió asociar criterios de diferentes autores con respecto al tema de investigación.

Investigación de Campo

También tuvo el modo investigación de campo, con la finalidad de obtener información de la situación en la que se encuentran los adultos mayores que padecen UPP. Se inició desde un problema, y llegó al desarrollo de procesos experimentales, recopilando información apegados a resultados técnicos y científicos, de tal forma que se sustentó de manera coherente el informe final del proyecto de investigación.

3.2. Población y Muestra

Por la naturaleza y características de la investigación no se requerirá población y muestra.

3.3. Recolección de Información

El presente proyecto de investigación se recopiló información de libros, revistas, artículos científicos, tesis de grado y publicaciones electrónicas; además de guías de observación como instrumentos metodológicos de investigación en el proceso de su desarrollo, conforme se identificó el problema, de los pacientes que padecen de UPP.

Se recopiló información relevante acerca de la tecnología de visión artificial aplicada a distintas áreas, como por ejemplo en seguridad vehicular, medicina, educación, etc. Se analizó los diferentes sistemas y se aplicó al sistemas de asistencia para pacientes con UPP, tomando en cuenta las técnicas y métodos más eficientes en dichos sistemas de visión computacional.

3.4. Procesamiento y Análisis de Datos

La información fue procesada y analizada esporádicamente, razón por la cual, se usó el siguiente procedimiento:

- Se organizó la información obtenida mediante la investigación bibliográfica-documental.
- Se analizó la información recolectada la misma que permitirá elaborar estrategias para el desarrollo de la solución de problemas.
- Se interpretó la información que permitirá una contribución para el desarrollo del problema planteado.

3.5. Desarrollo del Proyecto

Para el desarrollo del proyecto se tomarán en cuenta las siguientes actividades:

- Analizar la información sobre técnicas fisioterapia aplicadas en un proceso de rehabilitación a pacientes con Úlceras por Presión.
- Analizar la situación actual de los pacientes que padecen UPP en el Hogar de Ancianos Sagrado Corazón de Jesús.

- Determinar los requerimientos del sistema para establecer los parámetros necesarios que se considera al momento de ejecutar una rutina de rehabilitación para UPP.
- Diseñar una interfaz de monitoreo para asistir a los pacientes con Úlceras por Presión.
- Analizar los dispositivos, características técnicas para el desarrollo del sistema electrónico.
- Seleccionar los dispositivos adecuados para desarrollar sistema electrónico.
- Desarrollar el prototipo electrónico para el monitoreo de los pacientes que padecen Úlceras por Presión.
- Realizar pruebas de funcionamiento y corregir de errores del prototipo electrónico.
- Elaborar un Informe Final.

CAPÍTULO IV

DESARROLLO DE LA PROPUESTA

4.1. Introducción

En el mundo existe un gran número de personas que necesitan tecnología de asistencia pero no todos tienen acceso a ella. Según la Organización Mundial de la Salud (OMS), cada año se hospitaliza a 421 millones de personas y en su estancia en el hospital, estos pacientes sufren aproximadamente 42,7 millones de eventos adversos. En el sector público, la mayor parte de países brinda un escaso acceso a este tipo de tecnología. Muy pocos países disponen de un programa nacional de tecnología de asistencia. Incluso en países de ingresos altos, los productos de asistencia están a menudo racionados o excluidos de los sistemas de salud, lo que obliga a usuarios y familiares a desembolsar directamente fuertes sumas [5].

El manejo de las UPP implica desgastes para los profesionales de enfermería, pacientes y sus familias, lo que hace necesario que se cuantifique y analice el impacto de las intervenciones que se lleven a cabo, tanto en el ámbito de la prevención como de la rehabilitación [4]. Este problema de salud no es solo individual sino también colectivo como consecuencia de las distintas implicaciones que trae para el individuo, la familia, y los cuidadores.

Los sistemas basados en visión artificial enfocados para el monitoreo de pacientes con UPP, actualmente no se han desarrollado en su totalidad, por consiguiente en el Ecuador, directamente en la ciudad de Ambato, en el Hogar de Ancianos Sagrado Corazón de Jesús, no dispone de esta tecnología. Por lo que se convierte en una desventaja en cuestión de

tecnología de monitoreo de pacientes con UPP, a su vez esto conlleva a generar gastos económicos en los pacientes hospitalizados y por lo tanto se prolongue su recuperación.

4.2. Análisis de Factibilidad

Se enfoca en la disponibilidad de los recursos necesarios para llevar a cabo el desarrollo de la propuesta cumpliendo los objetivos establecidos. El presente proyecto de investigación es viable en varios escenarios entre los que destacan:

4.2.1. Factibilidad Técnica

El desarrollo del proyecto es técnicamente factible, debido a que los dispositivos electrónicos utilizados para el desarrollo del prototipo se encuentran a disposición en el mercado del país.

4.2.2. Factibilidad Económica

Económicamente el desarrollo del proyecto es factible, ya que no se requiere de gran cantidad de componentes electrónicos y por tanto no representa gastos elevados al investigador.

4.2.3. Factibilidad Bibliográfica

Bibliográficamente es factible ya que la información se encuentra en libros, revistas, artículos científicos, tesis de grado y publicaciones electrónicas con el fin de adquirir y utilizar la información dentro de la investigación.

4.3. Contexto de Nomenclatura

Para la implementación del sistema electrónico para asistir a pacientes en proceso de rehabilitación de úlceras por presión, se optó por comprimir el tema al acrónimo **S.E.A.P.UL.P.** (Sistema Electrónico para Asistir a Pacientes en proceso de rehabilitación de Úlceras por Presión).

4.4. Requerimientos de SEAPULP

SEAPULP tiene como finalidad detectar la posición en decúbito supino, lateral y sedestación del paciente además de los movimientos de las articulaciones del cuerpo. Se basa en la detección y notificación de algún suceso en el paciente, por ende cuenta con las siguientes características para su correcto funcionamiento:

- Adquisición del video en tiempo real mediante una cámara web.
- Procesamiento digital de las imágenes en cada fotograma.
- Detección de movimientos y cambios en la posición del paciente en rehabilitación.
- Notificación mediante mensaje de texto a la aplicación Telegram.
- Interfaz gráfica del prototipo.

En la Figura 23 muestra el diagrama de bloques con el cual trabaja el sistema.

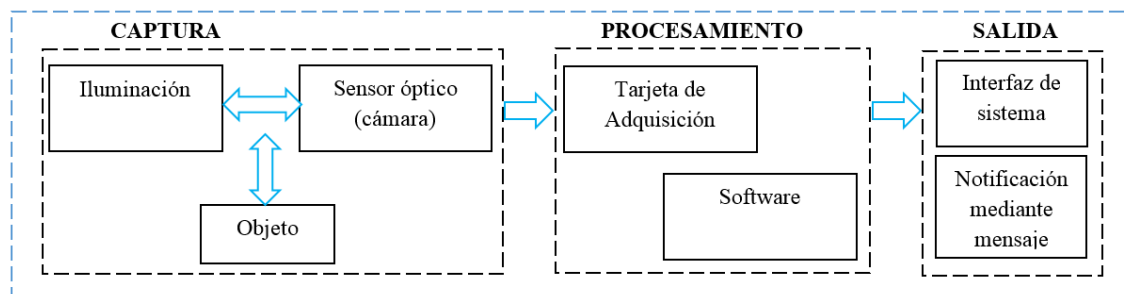


Figura 23. Diagrama de bloques de SEAPULP

Elaborado por: Investigador

Para el correcto funcionamiento del sistema electrónico es necesario detallar los requerimientos tanto en software como en hardware. El software mediante el uso de librerías como de un lenguaje de programación específico se encargan de realizar el procesamiento de la información, por otra parte el hardware a través de la selección de dispositivos adecuados permite la comunicación para el correcto funcionamiento del sistema.

4.4.1. Captura

La primera fase realiza la adquisición o captura de las imágenes en donde intervienen factores tales como:

Iluminación

La iluminación es el factor más importante para que el sistema funcione correctamente en ambientes no controlados, por tal razón es recomendable que el área donde se encuentre el sistema SEAPULP cuente con un sistema de iluminación en un rango de 1500-2500 lm, de esta manera se evita corregir el problema de iluminación mediante algoritmos de programación.

La luz de la lámpara debe distribuirse uniformemente por todo el cuerpo del paciente, esto facilita la detección dando como resultado una excelente calidad en la imagen. Para que no afecte la detección del rostro y de los movimientos de las articulaciones, es recomendable no dejar sombras en el rostro y en las zonas del cuerpo.

Sensor Óptico (Cámara)

La cámara es uno de los dispositivos importantes para la adquisición de imágenes en tiempo real, donde se debe analizar características como: formato, resolución, calidad de captura e iluminación.

Es necesario realizar un análisis comparativo entre las diferentes cámaras con el fin de que cumplan con los requerimientos de SEAPULP. A continuación en la tabla 2 se detalla cada una de ellas.

Tabla 2. Comparación entre Cámaras Digitales

Características	Genius Facecam	ALTEK	Microsoft Lifecam Hd3000	Web Omega
Interfaz	USB 2.0	USB 2.0	USB 2.0	USB 2.0
Zoom	Digital 3X	Lente Manual	8mpx	Exposición Automática
Sensor	VGA pixel CMOS	CMOS 640x480	CMOS	CMOS 300K
Formato de imagen y video	JPG/AVI	MJPEG/AVI	JPEG, BMP/AVI	JPGE/BMP
Resolución (píxeles)	640x 480	1280x960	720p HD	2560x1920

Driver	Plug and play	Plug and play	Plug and play	Requiere
Compatibilidad	Windows 2000/XP/Vista/7	Windows 7/10/vista/XP	Windows 7/10/vista/XP/MAC	Windows 7/8
Costo	\$17	\$20	\$57,40	\$16

Elaborador por: Investigador

Acorde al análisis previo realizado se optó por la adquisición de Genius Facecam, debido a que cumple con los requerimientos necesarios como mejor calidad del sensor de conversión de imagen A/D.

4.4.2. Procesamiento

La segunda fase tiene por objetivo el procesamiento de las imágenes donde intervienen factores como son: tarjeta de adquisición (hardware), software y algoritmo. A continuación se detalla cada uno de ellos.

Hardware

Comparación entre placas embebidas o micro ordenadores

Actualmente se puede encontrar tipos de ordenadores de placa reducida como se detallan en la tabla 3, donde es necesario diferenciar las características técnicas de cada uno de ellos.

Tabla 3. Comparación entre micro ordenadores

Características	Raspberry Pi 3	Arduino Uno	Orange Pi	BeagleBone Black
CPU	ARM11 4 núcleos 1,2 GHz a 64 bits	ATMega 328	Allwinner H3 4 núcleos 1GHZ	ARM CORTEX – A8
Tamaño	8.5 x 3.5 cm	6.8 x 5.3 cm	8.5x 5.5 cm	7.2 x 5.2 cm
RAM	1GB	2KB	1GB	512 MB
Salidas de video y resolución	HDMI 1920x1200	N/A	HDMI 1280x1024	HDMI 1280x1024
USB	2.0/4	2.0	2.0/3	2.0/1

Ethernet	Si 10/100Mbps	N/A	Si 10/100Mbps	Si 10/100Mbps
PUERTOS	4	N/A	1	2
Conexión Wireless 802.11n	SI	NO	NO	NO
Costo	\$65	\$29	\$83	\$92

Elaborador por: Investigador

A través del análisis de características técnicas entre las diferentes micro ordenadores se seleccionó la tarjeta Raspberry Pi 3 debido a características tales como:

- Cuenta con tarjeta integrada Wireless.
- Dispone de 4 puertos USB 2.0 para la conexión de diferentes periféricos.
- Mayor velocidad del procesador.
- Mayor RAM en comparación con las otras placas.
- Bajo costo.

Software

Existen diferentes tipos de software de código abierto y privado, los cuales manejan procesamiento de imágenes. En la siguiente tabla 4 se puede observar un análisis comparativo de los diferentes lenguajes de programación más destacables.

Tabla 4. Comparación entre lenguajes de programación

Características	Visual Studio	Python	Labview	Matlab
Sistema Operativo	Windows	Linux, Windows y Mac	Multiplataforma	Multiplataforma
Licencia	Propietaria	Libre	Propietaria	Propietaria
Idioma	Múltiple	Ingles	Ingles	Ingles
Última Versión	2017	3.6		R2018a
	-Visión Artificial -Diseño de plataformas web	-Software y juegos de desarrollo - Servicio Web	- Diseño de sistemas - Sistemas de	-Diseño de controladores - Visión Artificial

Aplicaciones	-Diseño de sistemas embebidos -Procesamiento de Imágenes	- Interfaz grafica - Procesamiento de imágenes -Programación red	control industrial - Procesamiento de imágenes	- Análisis estadístico - Software matemático - Procesamiento de imágenes
Ventajas	-Posee interfaz gráfica eficiente -Amplia documentación -Librerías de reconocimiento facial -Ejecución de código de forma remota, mediante conexión con Windows IoT	- Lenguaje de propósito general - Multiplataforma - Orientado a objetos - Portable	- Fácil de aprender y usar - Mejora en documentación y exportación de datos	- Lenguaje de bajo nivel - Facilidad de uso - Orientado a objetos - Soporta varias base de datos - Gráficos de muy buena calidad. - Robusto
Desventajas	- No es posible la exportación de código a otras plataformas	- Lentitud por ser un lenguaje interpretado	- Consume muchos recursos - Lento a la hora de compilar	- Mayor consumo de recursos de la Pc.

Elaborado por: Investigador

Una vez realizado el análisis comparativo de cada software de programación se seleccionó Python debido a características tales como:

- Es libre y ofrece código abierto, permitiendo crear propias aplicaciones en su desarrollo.
- Su sintaxis es de fácil entendimiento ya que es cercana al lenguaje natural, además de que los programas realizados en este lenguaje parecen pseudocódigos lo que brinda una gran ayuda en su mantenimiento.
- Es un lenguaje que se encuentra orientado a objetos.
- Dispone de librerías y funciones para el procesamiento de imágenes.
- No necesita una gran cantidad de memoria.

- Amplia información tanto para el aprendizaje como para su utilización.
- Se lleva bien con los datos debido a que los gestiona de manera eficiente.
- En conjunto con la placa electrónica Raspberry Pi 3 permite un monitoreo constante en forma remota.

4.4.3. Salida

En la fase final se encuentra la salida del sistema para ello se necesita una pantalla donde se visualice el monitoreo en tiempo real, en cuanto al envío de notificaciones utiliza un dispositivo móvil que aloje la aplicación Telegram.

Pantalla

Para indicar la salida del sistema se necesita una pantalla, la misma que ayuda a visualizar el proceso; de acuerdo a los requerimientos de SEAPULP se optó por la pantalla LCD de 7 pulgadas de la Raspberry Pi ya que, además de brindar una buena calidad de imagen, su pantalla es táctil lo que permite que el usuario interactúe con mayor facilidad al momento de hacer uso del sistema.

Notificaciones

Las notificaciones son enviadas a un dispositivo móvil que cuente con la aplicación Telegram. El personal de turno es notificado mediante un mensaje el cual indica si el paciente no está en posición o a su vez se detecta movimientos durante la rutina de rehabilitación. Para que estas notificaciones sean enviadas en tiempo real, el dispositivo debe mantenerse conectado a la red wi-fi o de datos.

4.5. Componentes de SEAPULP

Para el desarrollo del Sistema Electrónico para asistir a pacientes en proceso de rehabilitación de Ulceras por Presión, se requiere de los siguientes dispositivos para que puedan interactuar entre sí.

- Tarjeta electrónica con capacidades de un ordenador.
- Cámara digital.
- Pantalla LCD.
- Dispositivo inteligente para recibir notificaciones

- Fuente de alimentación

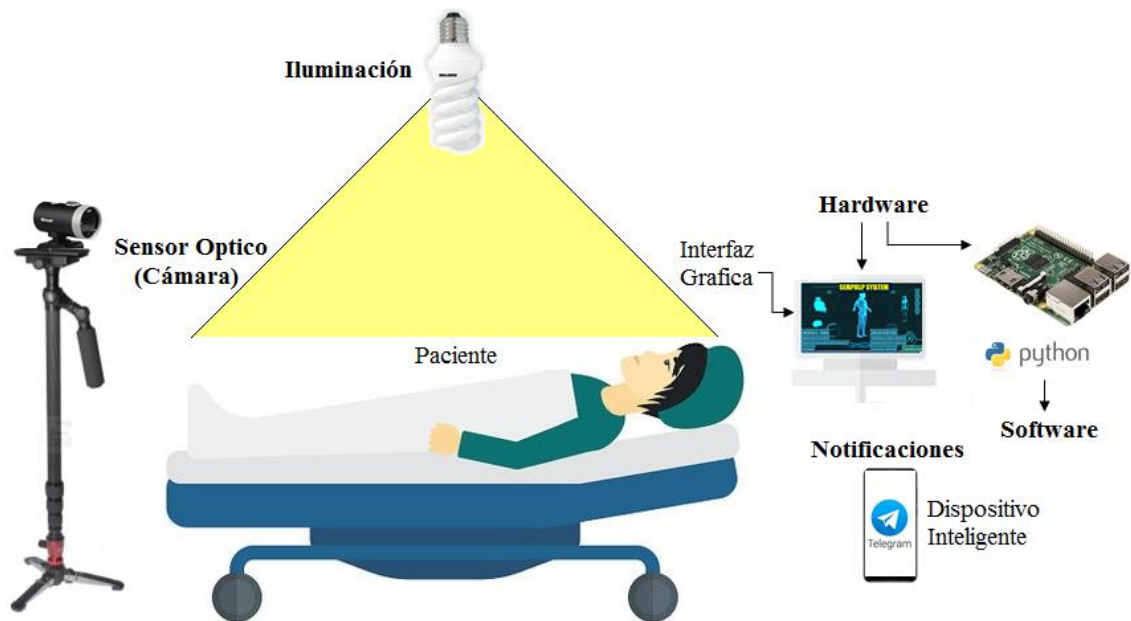


Figura 24. Sistema SEAPULP
Elaborado por: Investigador

4.6. Diseño de SEAPULP

4.6.1. Detección de la Posición del Paciente

En este bloque de información, es necesario tener en cuenta algunos conceptos básicos para entender con mayor claridad que es un "reconocimiento" y que es una "detección", ambos conceptos son muy populares hoy en día cuando se habla de análisis de imágenes, pero a veces dista de lo que en realidad significa. En el proyecto de desarrollo del sistema SEAPULP vamos a aplicar la "detección" facial, por lo tanto no debe confundirse con "reconocimiento". Si bien, es posible hacer un reconocimiento mediante OpenCV, el primer paso es crear un software o algoritmo que primero identifique lo se desea reconocer.

El reconocimiento de objetos, se diferencia porque además de haber pasado por una detección específica de la imagen, también es posible entregar información más precisa. En la actualidad existen distintos algoritmos que pueden ser implementados para una detección facial, muchos de ellos un tanto complejos o específicos según el área en donde se desea emplear, por ejemplo: redes neuronales, grafos, entre otros.

Para el sistema SEAPULP, se trabajó con un archivo .xml que viene adjunto en la instalación, este archivo tiene la características de identificar rostros frontales, el cual brinda una ventaja ya que el sistema necesita ser específico en la detección. A continuación se explica cuál es el proceso para la detección de objetos: Para la detección de objetos es necesario que se disponga de un archivo .xml, previamente entrenado y un software de programación que soporte visión artificial. Para el Sistema SEAPULP, se optó trabajar con Python 2.7.9 y OpenCv 2.4.12

Visión Artificial

La visión Artificial se representa en cuatro fases por ello hay que codificar siguiendo este proceso.

Primera fase: En la primera fase se realiza la captura o adquisición de las imágenes a través de una cámara digital.

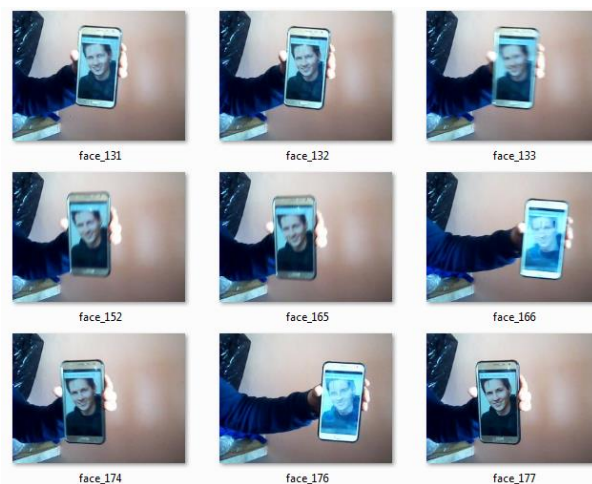


Figura 25. Captura de imágenes a través de una cámara web.

Elaborador por: Investigador

Segunda fase: Tiene por objetivo el tratamiento de las imágenes. Se lo realiza mediante filtros digitales transformando a la imagen original a una imagen de escala gris para ayudar al posterior análisis.



Figura 26. Imagen aplicado filtros de escala Gris
Elaborado por: Investigador

Tercera fase: Segmentación, consiste en particionar una imagen en regiones homogéneas con el fin de extraer características de dichas regiones (textura, color, forma, contorno, etc.).



Figura 27. Imagen con detección de color
Elaborado por: Investigador

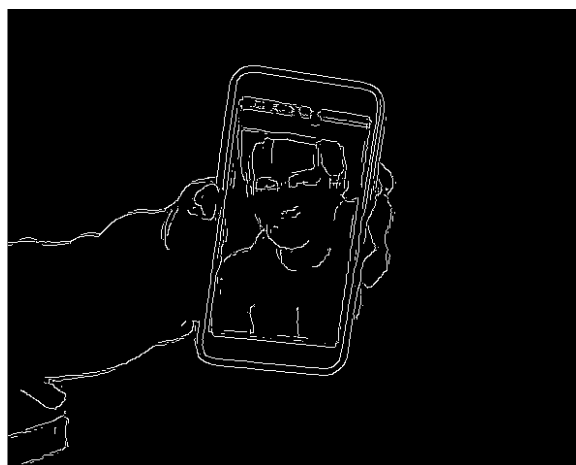


Figura 28. Imagen con aplicación de método canny
Elaborado por: Investigador

Cuarta fase: Lleva a cabo el reconocimiento o clasificación. Emplea el análisis de ciertas características que se establecen previamente en el entrenamiento de imágenes para detectar los objetos deseados.

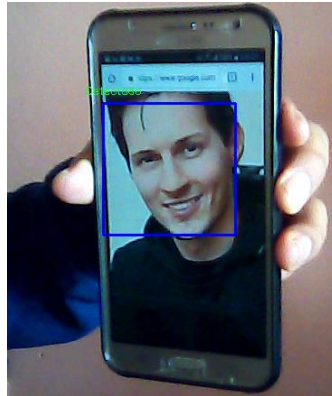


Figura 29. Imagen de reconocimiento de rostro
Elaborado por: Investigador

4.6.2. Diagrama de Flujo SEAPULP para detección de posición

Para la implementación del sistema SEAPULP, se ha establecido el siguiente diagrama de flujo para detectar la posición, como se muestra en la Figura 30.

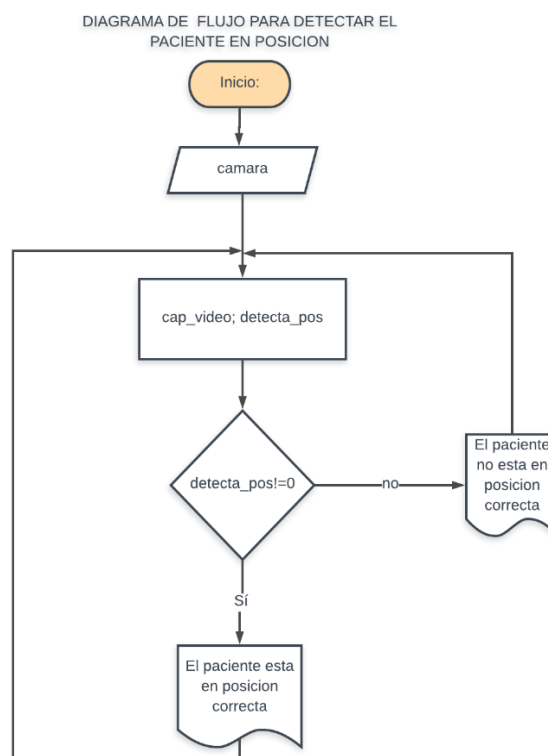


Figura 30. Diagrama de flujo para detectar al paciente
Elaborado por: Investigador

4.6.3. Desarrollo del código en Python para detectar la posición

Una vez instalado los paquetes necesarios, se procede con el desarrollo del código en Python. Para ello lo primero es fundamental indicar que el código se estructura en forma general en tres bloques más importantes, como son captura, procesamiento y salida del sistema, a continuación se detalla los pasos a seguir.

- **Importación de librerías y Captura de video**

Como básico es necesario importar las siguientes librerías para hacer uso de la visión artificial

```
import numpy as np
```

```
import cv2
```

Una vez importado las librerías, se procede a capturar el video a través de una cámara web, dando uso del método indicado a continuación.

```
cap = cv2.VideoCapture(0)
```

- **Procesamiento**

En este bloque el código se ejecuta en un bucle infinito, para que en cada ciclo haga la búsqueda de las características establecidas en el archivo .xml. Este archivo para detectar rostros viene adjunto en la carpeta de instalación de OpenCV. Normalmente se encuentra en la siguiente ruta C:\opencv\sources\data

En el bucle infinito, se procede a leer la variable *cap*. que se asignó anteriormente al método.

```
img = cap.read()
```

A continuación se convierte cada imagen capturada, a escala de Gris para ayudar a posteriores procesos.

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

El siguiente paso es hacer la búsqueda de características de imagen, en cada frame capturado, esta línea de código es muy importante; jugar con los parámetros del método para calibrar la sensibilidad de la detección, normalmente para detectar rostros es como se indica a continuación.

```
faces = face_cascade.detectMultiScale(gray, 1.3, 5)
```

- **Salida del Sistema**

Como salida del sistema, es la visualización del video indicando en un recuadro la característica de imagen que se ha detectado, con respuesta a la detección genera una notificación mediante un mensaje de texto enviado a la aplicación Telegram, como respuesta al sistema.

Con el siguiente código se construye un recuadro al rededor del objeto detectado.

```
cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
```

Y se visualiza la salida

```
cv2.imshow('imagen de salida',img)
```

4.6.4. Detección de Movimientos del Paciente

Una de las aplicaciones prácticas que podemos realizar con OpenCV es un sistema de detección de movimientos, por ejemplo usar una webcam para detectar movimientos y hacer que envíe una notificación a nuestro dispositivo móvil, para la detección de movimiento utilizaremos OpenCV, Python, y una cámara web.

Para esto se aplicó el método llamado “Differential Images” este consiste en comparar dos imágenes, la primera imagen actual capturada por la webcam y la segunda es la imagen anterior, si las dos imágenes son idénticas concluimos en que no hubo movimiento. Existen diferentes técnicas, métodos o algoritmos que posibilitan la detección de movimiento. Al igual que en otras materias, en la visión artificial no hay casos genéricos. Dependerá de cada situación usar uno u otro. En este desarrollo se indica 4 métodos para hacer la detección de movimiento con OpenCV y Python.

Supresión del fondo

Consiste en estimar un modelo del fondo y compararlo con el frame actual para detectar cambios. El resultado es una imagen binaria donde los píxeles se clasifican entre sí, forman parte del fondo o son del primer plano.



Figura 31. Método supresión de fondo para detectar movimiento
Elaborado por: Investigador

Operaciones morfológicas

En la imagen resultado de la operación anterior suelen aparecer regiones de pequeño tamaño marcadas como de primer plano debido al ruido en el frame original. Una solución muy común en estos casos es aplicar operaciones de dilatación y erosión con el objeto de suprimirlas.

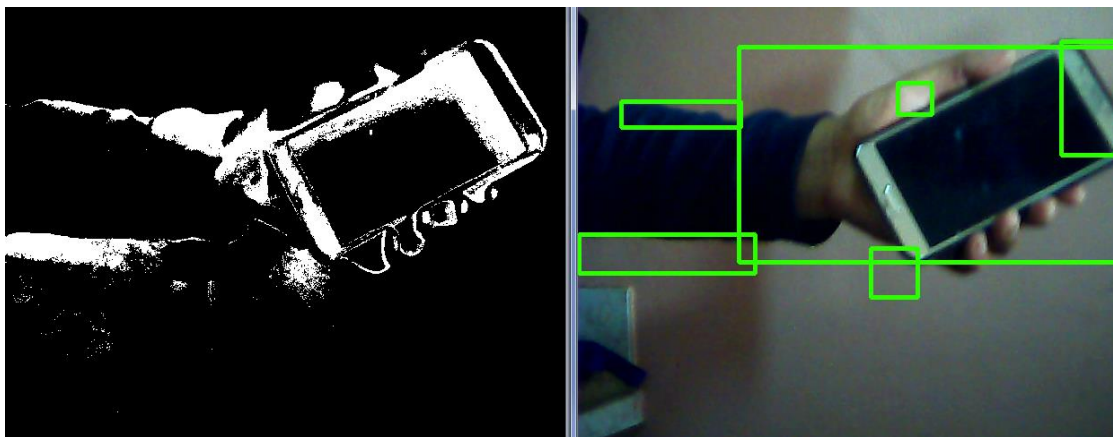


Figura 32. Método operaciones morfológicas para detectar movimiento
Elaborado por: Investigador

Extracción de blobs

Los píxeles clasificados como de primer plano suelen agruparse en regiones que corresponden a objetos en movimiento en el frame original. La extracción de blobs

permite identificar estas regiones, por ejemplo, marcarlas con un cuadro delimitador en la imagen original.

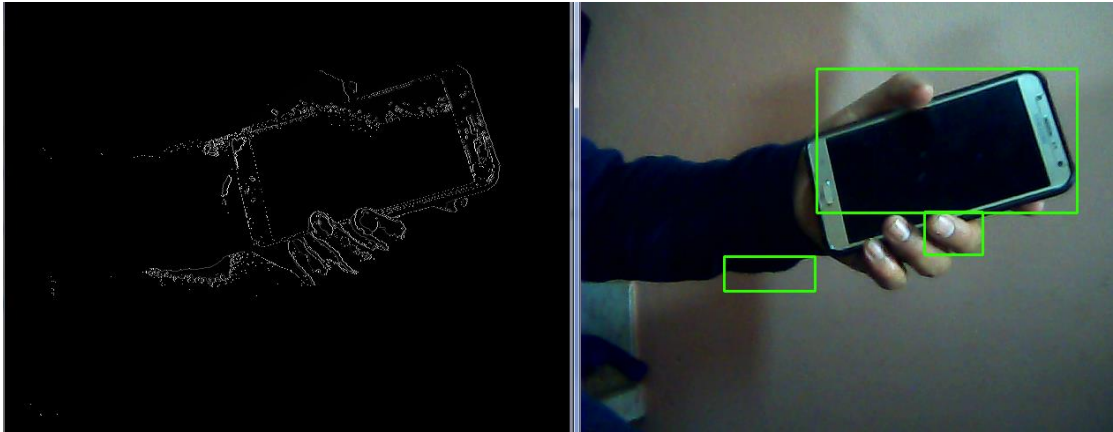


Figura 33. Método Extracción de blobs para detectar movimiento
Elaborado por: Investigador

Extracción entre el segundo plano y el primer plano

A la hora de restar una imagen con otra, se obtienen valores negativos, para evitar esto lo que se hace es restar y obtener los valores absolutos de cada píxel. OpenCV proporciona un método para hacer esto. En esta parte del proceso se debe quedar con aquellos píxeles que superen un umbral. El objetivo es binarizar la imagen es decir, tener dos posibles valores. Todos aquellos que superen el umbral serán píxeles blancos y los que no lo superen serán píxeles negros. Esto servirá para seleccionar el objeto en movimiento.

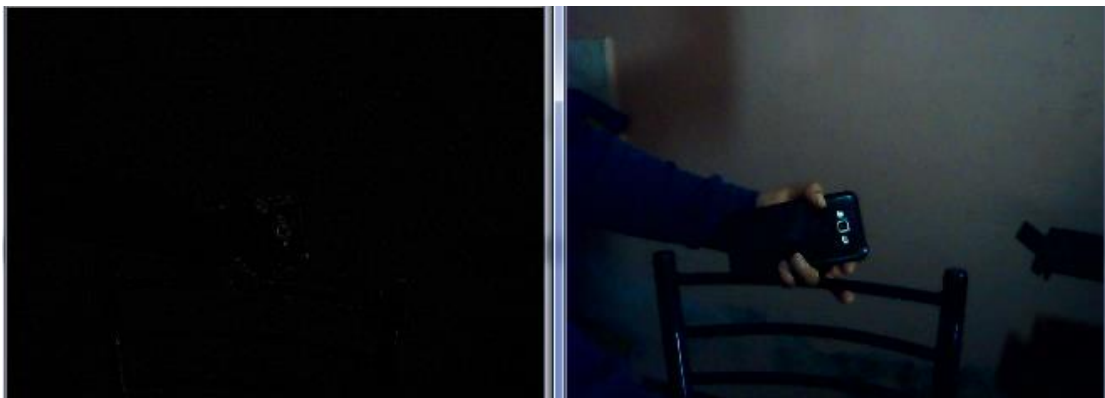


Figura 34. Método Extracción entre el segundo y primer plano para detectar movimiento
Elaborado por: Investigador

4.6.5. Diagrama de Flujo SEAPULP para detección de movimiento

Para la implementación del sistema SEAPULP, se ha establecido el siguiente diagrama de flujo para detectar la posición, como se muestra en la Figura 35.

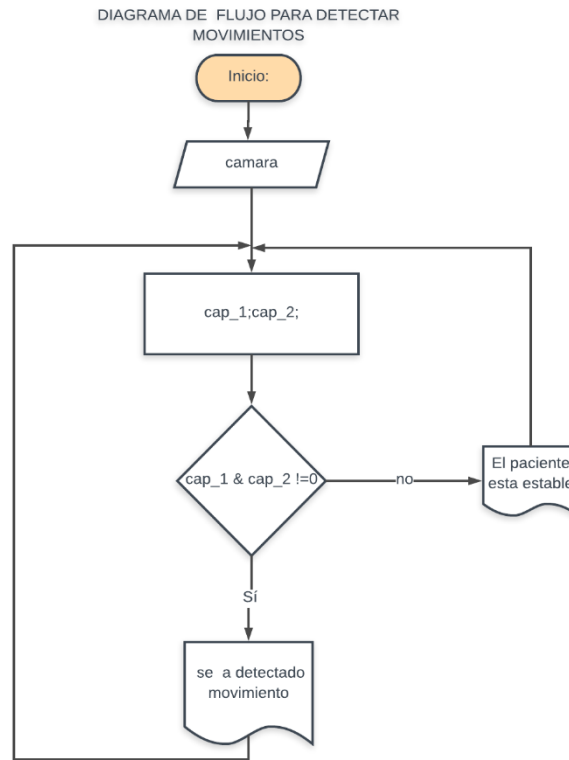


Figura 35. Diagrama de Flujo para detectar movimiento
Elaborado por: Investigador

4.6.6. Desarrollo del código en Python para detectar movimientos

El código para la detección de movimientos es muy similar al código de detección de posición del paciente, con clara diferencia que este código hace la comparación de los frames actual y el anterior, de esta manera puede identificar si hay cambios en los pixles en las imágenes para determinar si hay movimiento.

Tanto la primera parte de importar librería y como también la salida del sistema como respuesta, son iguales como se explicó en el apartado de detección de posición de los pacientes.

El bloque de código más importante para la detección de movimiento es la siguiente:

Lo primero que se debe tener son las dos capturas la actual y anterior.

```
_,anterior=cap.read()
```

```
_,actual=cap.read()
```

La variable ***anterior*** debe ser establecida antes del ciclo infinito. Una vez con las dos variables el siguiente código es quien hace la operación de diferencia para determinar si hay movimiento.

```
movim=np.array(abs(np.array(actual,np.float32)-np.array(anterior,np.float32)),np.uint8)
```

Esta línea de código entrega un resultado para realizar alguna operación, cuando el resultado es mayor que cero, nos indica que hay movimiento.



Figura 36. Imagen cuando detecta movimiento
Elaborado por: Investigador

4.6.7. Registro y Almacenamiento de Datos de los Pacientes

Para el registro de datos de monitoreo y pacientes en terapia, en este proyecto no se desarrolló en base a un servidor web, más bien se desarrolló de tal forma que el mismo ordenador almacene dichos datos en el Disco duro, estos datos se guardaron en un archivo .csv el cual le permite manipular los datos a través de un algoritmo desarrollado en Python. A continuación se explica en forma resumida:

El siguiente código realiza la función de abrir el archivo11.csv de forma que se puede sobre escribir los datos por el atributo “w”


```
archivo=open("archivo11.csv","w")
```

Estos atributos pueden ser diferentes tales como “r”, abre el archivo solo de lectura, y el parámetro “a” agrega nuevos datos a los datos ya existentes

El siguiente código escribe el dato en el documento, estos datos deben estar de forma de caracteres.

```
archivo.write(ent)
```

En donde ent = “este dato se guardara”, tiene el valor a guardar.

Y por último cierra el archivo para dar paso a un nuevo proceso

```
archivo.close()
```

De esta manera bajo una lógica de programación, conforme a los requerimientos del sistema se logró concretar los objetivos propuestos en el presente proyecto.

4.6.8. Salida del Sistema SEAPULP

Los dispositivos de salida son aquellos periféricos que se asocian a un ordenador y que tienen como finalidad comunicar información al usuario. Los dispositivos de salida muestran información que ya ha sido ingresada y procesada, con información que es devuelta al mundo real.

La salida del sistema SEAPULP, está conformado por:

- Pantalla
- Mensajes de texto a un celular inteligente

La pantalla está conectada directamente a la placa base, con la finalidad de indicar la salida del sistema frente algún evento. En cambio para la alerta a través de un mensaje de texto, se consideró como terminal a un teléfono móvil inteligente, este a su vez tenga la capacidad de alojar la aplicación Telegram.

Configuración de Telegram

Para dar uso de los servicios que ofrece, lo primero que se debe hacer es instalar la app de Telegram en un dispositivo móvil. Una vez ya instalada la app, se procede a buscar en el icono de Telegram: *BotFather*

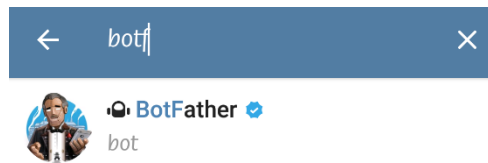


Figura 37. BotFather para iniciar la configuración
Elaborado por: Investigador

Ingresa en BotFather y escribir: */start*

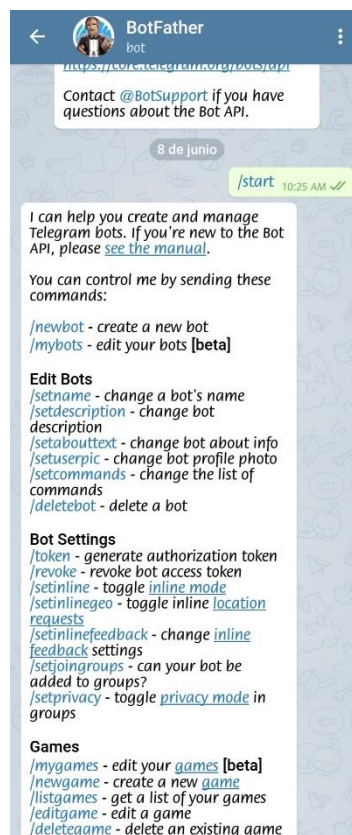


Figura 38. Inicia la configuración de usuario Bot
Elaborado por: Investigador

Para crear un usuario escribir */newbot* después un nombre en mi caso *daniel994_bot*, para modo de explicación.

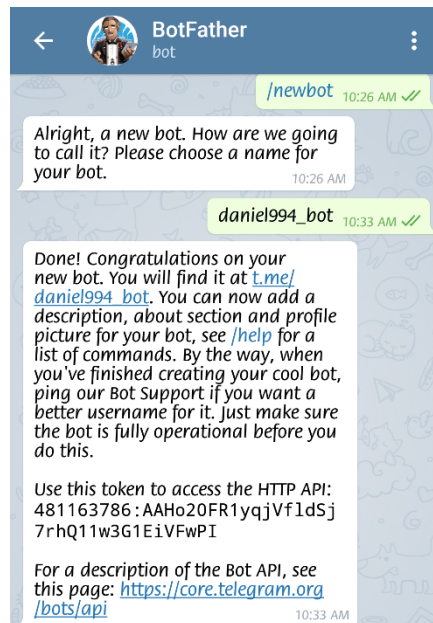


Figura 39. Asignación de usuario bot y nombre de usuario
Elaborado por: Investigador

Ahora devolverá un código HTTP API: copiar el código completo, este código es el que se usa más tarde.

Chat ID

Para obtener el código chat_Id, de modo que permita enviar y recibir mensajes se aplica el siguiente proceso.

Dentro de la aplicación Telegram buscar: *get id*

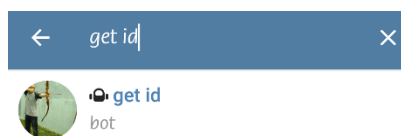


Figura 40. Get id para iniciar la configuración
Elaborado por: Investigador

Una vez adentro digitamos */start* donde muestra el usuario y el chat_Id, como se indica en la figura.

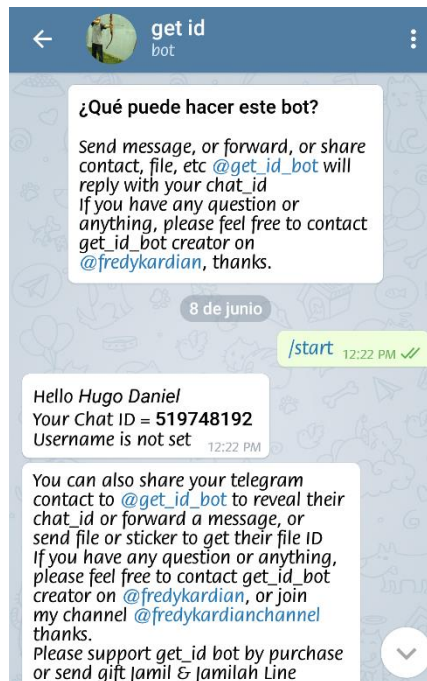


Figura 41. Código Chat_ID
Elaborado por: Investigador

Una vez finalizado este proceso, dar uso en la programación en Python a los códigos generados: API y Chat_id. Para dar uso de ello es necesario tener instalado la librería de Telegram sobre Python, la misma que intaló anteriormente.

Una vez instalado se procede a importar la librería desde Python

```
import telepot
```

Posterior a ello se procede a colocar la API, que anteriormente se había adquirido, con la siguiente línea de código:

```
bot = telepot.Bot('481163786:AAHo20FR1yqjVfldSj7rhQ11w3G1EiVFwPI')
```

Se puede dar uso de la API para enviar mensajes al dispositivo móvil, esto ya depende como se estructure su código, a continuación la línea de código empleado para enviar mensajes al celular.

```
bot.sendMessage(519748192, "Hola, aquí el mensaje para enviar")
```

4.6.9. Diseño de la Interfaz

Para que el Sistema SEAPULP pueda interactuar con el usuario, es necesario realizar una interfaz gráfica GUI. Para desarrollar una Interfaz se dispone actualmente diferentes plataformas que son compatibles con Python entre ellos tenemos: Qtcreator , Boa constructor, PyGTK, etc. Para este proyecto se utilizó TkInter una librería propia de Python, a continuación se detalla el desarrollo.

Lo primero que se debe hacer al igual que con otros módulos, es importarlo para comenzar a utilizarlo, y al igual que con otros módulos no hay una sola forma de hacerlo. La primer forma

```
from Tkinter import *
```

Y la segunda forma:

```
import Tkinter
```

De esta manera ya se tiene el modulo importado, ahora solo queda llamar a los métodos para crear una ventana en donde se alojaran los distintos botones y cuadros de textos, etiquetas, necesarios para la interfaz.

```
from Tkinter import *  
ventana=Tk()  
ventana.geometry ("250x170+0+0")  
ventana.mainloop()
```

En base a las líneas de código descritas anteriormente, se puede visualizar una ventana básica, del cual se empieza para hacer una variedad de aplicaciones. En la siguiente figura se puede ver el resultado tras compilar el código.

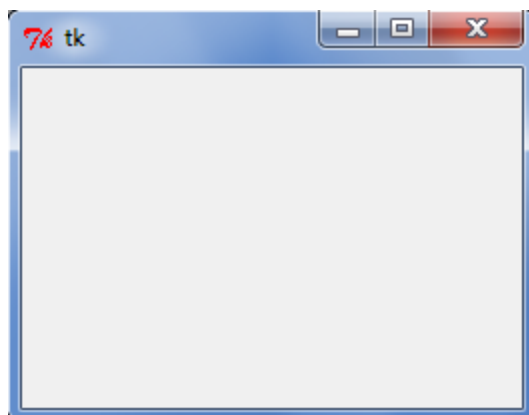


Figura 42. Ventana básica en Python y Tkinter
Elaborado por: Investigador

Con la siguiente línea de código se agrega una etiqueta de texto para que la interfaz sea mas entendible.

```
etiqueta=Label(text="SEAPULP SYSTEM ",font=("Impact", 12)).place(x=80,y=0)
```

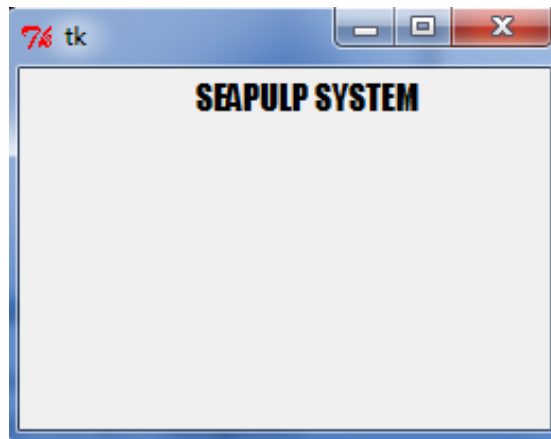


Figura 43. Ventana básica y etiqueta de texto
Elaborado por: Investigador

Además, permite crear botones de tal forma que podamos agregar algún evento a dicho botón este pueda realizar alguna operación.

```
boton1=Button(text="INICIAR SISTEMA ",font=("verdana", 10)).place(x=80,y=53)
```

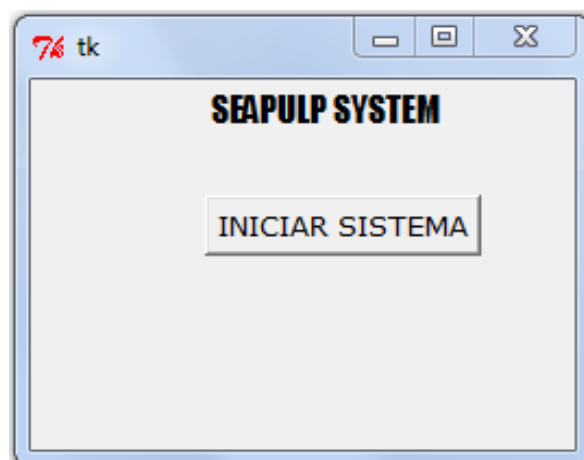


Figura 44. Ventana básica y botón
Elaborado por: Investigador

Una vez estructurado el código de tal forma que se pueda interactuar con los botones creados conforme a los requerimientos, la interfaz del sistema SEAPULP se puede apreciar de la siguiente manera.



Figura 45. Interfaz del sistema SEAPULP
Elaborado por: Investigador

4.7. Diagrama de Conexión de SEAPULP

El sistema SEAPULP utiliza un micro ordenador Raspberry PI 3 como tarjeta principal de procesamiento, una cámara web, una LCD de 7" para indicar la salida, los cuales son los elementos necesarios para el sistema funcione de forma ya preestablecida. La tarjeta Raspberry PI 3 recibe el video del entorno a través de la cámara web que se encuentra ubicada en la parte frontal del paciente en rehabilitación a una distancia predeterminada. Para que el sistema SEAPULP trabaje de manera eficiente hay que tener en cuenta parámetros como corriente y voltaje que soporta la tarjeta Raspberry Pi 3.

- Voltaje: 5V (DC)
- Corriente: 2.5(A)

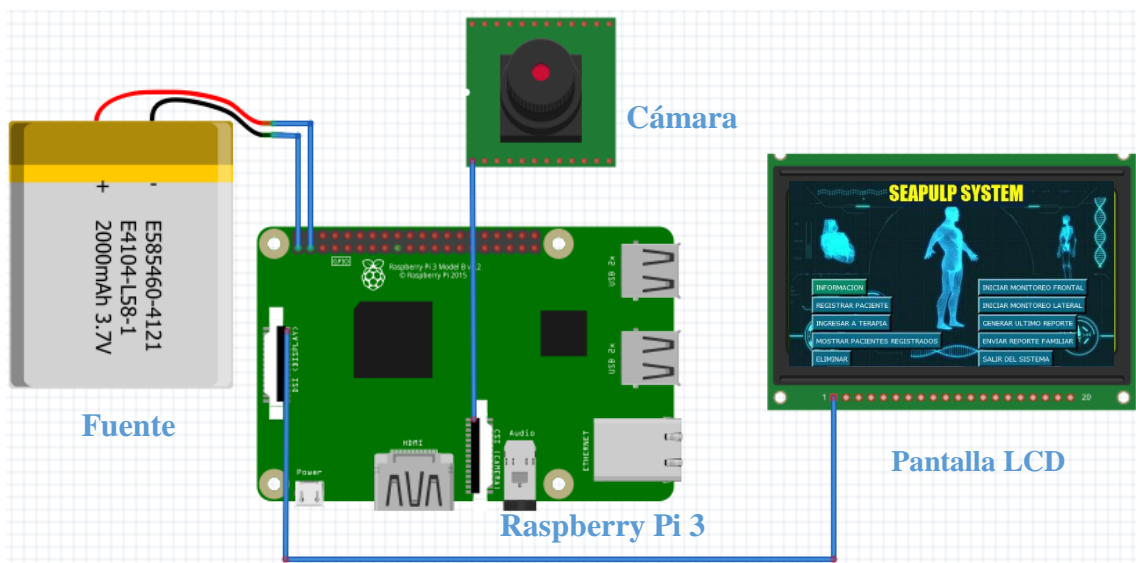


Figura 46. Diagrama de conexión de SEAPULP
Elaborado por: Investigador

4.8. Confiabilidad de SEAPULP

La confiabilidad del sistema es la probabilidad de que el sistema funcione para el cual fue diseñado bajo condiciones establecidas durante un periodo determinado bajo condiciones operativas específicas. Independientemente de la metodología que se utilice, la finalidad es evaluar e identificar los elementos críticos del sistema y determinar la sensibilidad del mismo.

SEAPULP está compuesto de diferentes componentes de acuerdo al diseño con el propósito de lograr el cumplimiento de determinadas funciones con una adecuación y fiabilidad aceptables. El tipo de componentes, la cantidad, calidad y el modo en que están dispuestas para el funcionamiento tiene un efecto directo a la confiabilidad de sistema. El sistema SEAPULP como componente principal cuenta con la Raspberry PI 3, es aquel que realiza la tarea principal y por consiguiente ocasiona que la tarjeta pueda alcanzar altas temperaturas, aunque los fabricantes anuncian que es normal que se caliente el procesador y que a la vez no es peligroso para la máquina ni para las personas, ya que está diseñado para Running Hot.

El componente como la cámara, es aquel que complementa el funcionamiento del sistema, por consiguiente se ha establecido componentes que proporciona un grado

de confiabilidad al sistema. La posible vibración principalmente en la cámara puede ocasionar que el sistema no detecte correctamente lo establecido y que este a su vez arroje falsas alarmas ocasionado alertas indebidas al personal de servicio, es por eso que hay que considerar mucho el aspecto de disminuir dicha anomalía, para que el sistema pueda brindar una confiabilidad frente a sus usuarios.

El sistema al estar desarrollado bajo una estructura de alto rendimiento proporciona gran ventaja al disponer del sistema SEAPULP, es decir factores externos pueden ocasionar que el sistema no funcione de tal manera ya establecida.

Para determinar la confiabilidad del sistema hay que considerar datos tales como el tiempo promedio de fallas y el tiempo promedio en reparación.

Evaluando el sistema por el tiempo de una hora, podemos sacar datos de la confiabilidad del sistema con la siguiente ecuación:

$$CF = \frac{TF}{TF + TR} \quad (4)$$

Donde:

TF = Tiempo en funcionamiento

TR = Tiempo de Reparación del sistema

CF = Confiabilidad del sistema

Para determinar el tiempo promedio de reparación TR , entre algunos de los posibles sucesos se consideró como indica en la siguiente tabla:

Tabla 5. Tiempo promedio de reparación TR

CORRECCION POSIBLES FALLOS	TIEMPO REPARACION (MIN)
Enfoque de la cámara a la posición adecuada	3
Cambio de cámara en caso de avería	5
Reinicio del sistema	1
Tiempo Promedio	3

Elaborador por: Investigador

Entonces:

Si, $TF = 60 \text{ min}$; $TR = 3 \text{ min}$

$$CF = \frac{TF}{TF + TR}$$






$$CF = \frac{60 \text{ min}}{(60 + 3) \text{ min}}$$


$$CF = 0.95 \times 100\% = 95\%$$

4.9. Pruebas y Resultados

Pruebas realizadas con iluminación de 2500 lm

Tabla 6. Detección de paciente con iluminación de 2500 lm






Paciente	Monitoreo Decúbito Supino	Monitoreo Decúbito Lateral
Encamado		
Encamado		
Sedestación	<p style="text-align: center;">Monitoreo Frontal</p> 	

Sedestación		
-------------	---	--

Elaborador por: Investigador

Pruebas realizadas con menor iluminación

Tabla 7. Detección de paciente con menor iluminación

Paciente	Monitoreo Decúbito Supino	Monitoreo Decúbito Lateral
Encamado		
Encamado		
Sedestación	Monitoreo Frontal	
		



Elaborador por: Investigador

Para que el sistema detecte la posición del paciente se necesita entre 1500 a 2500 lm en iluminación de esta manera el sistema no tiene problemas en la detección y trabaja eficientemente como se observa en la tabla 6 mientras que, si se trabaja con menor cantidad de lúmenes el sistema no detecta las posiciones como se observa en la tabla 7. Por lo tanto este es un factor muy importante que se debe tomar en cuenta, de esta manera se evita corregir mediante algoritmos de programación.

Resultados

Para la adquisición de los datos incluidos en la tabla 6 se realizó líneas de código extras de tal forma pueda realizar un conteo de las imágenes detectadas y no detectadas en cada Frame capturado durante un lapso de tiempo programado, de esta manera con los datos ya obtenidos en función de la distancia se procede hacer el cálculo respectivos, se detalla los resultados a continuación.

Tabla 8. Frame capturado en función de la distancia

PROCESO	DISTANCIA SEAPULP (m)	DETECCION CORRECTA (%)	VELOCIDAD DE PROCESAMIENTO (FPS)
MONITOREO DE POSICION	1,00	100,0 %	10
	1,25	100,0 %	10
	1,50	99,9 %	10
	1,80	99,9 %	10
	2,00	99,9 %	10
	2,25	99,92 %	10
	2,50	99,87 %	10
	3,00	99,87 %	10

		DETECCION DE FALSOS MOVIMIENTOS	
Movimientos	1,00	0,0 %	10
	1,25	0,0 %	10
	1,50	0,0 %	10
	1,80	0,0 %	10
	2,00	0,0 %	10
	2,25	0,0 %	10
	2,50	0,0 %	10
	2,75	0,0 %	10

Elaborador por: Investigador

Como se observa en la Tabla 8, el porcentaje de falsas alertas indican como resultados muy bajos de manera que el sistema responde bien conforme se ha establecido su funcionamiento. A continuación se explica los ítems detallados en la tabla 8.

Distancia SEAPULP: Es la distancia a la cual el paciente está en relación a la cámara para ser monitoreado.

Detección de correcta: Se refiere a que el sistema detecta según lo establecido en el código.

Detección de falsos movimientos: Hace referencia a que el sistema indica que hay movimiento sin existir alguno.

Velocidad de Procesamiento (FPS): Es la cantidad de imágenes por segundo que trabaja con una cámara.

4.10. Presupuesto del Proyecto

Para el desarrollo de SEAPULP intervienen costos relacionados que van desde el diseño hasta la implementación del prototipo, además de las horas trabajadas durante el desarrollo del sistema. Para obtener el costo por hora de trabajo de un Ingeniero Electrónico especialista en mantenimiento, se tomó en cuenta el salario mínimo que es de \$ 415.53, obtenidos de la página del Ministerio de Relaciones laboral 2018 [57]. Tomando en cuenta las 8 horas laborables por día, donde:

Mes: \$ 415.53

Día: \$ 415.53/30= \$ 13.85

Hora: $\$13.85/8 = \$ 1.73$

Para el desarrollo de SEAPULP se toman en cuenta un total de 200 horas, entonces el costo de trabajo será:

$$\text{Costo de trabajo} = 200 \times \$1.73 = \$ 346$$

En la tabla se detalla el costo total del sistema SEAPULP, donde intervienen costos de trabajo y de los diferentes elementos electrónicos utilizados para el desarrollo del prototipo, los mismos que se encuentran a disposición en el mercado del país.

Tabla 9. Presupuesto del Proyecto

N°	DETALLE	UNIDAD	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
1	LCD Rasp 7"	c/u	1	110,00	110,00
2	Raspberry PI 3	c/u	1	65,00	65,00
3	Ventilador	c/u	1	7,70	7,70
4	Cámara	c/u	1	17,00	17,00
5	Ensamblaje de prototipo	c/u	1	50,00	50,00
6	Diseño del Sistema	hora	200	1,73	346,00
SUB-TOTAL					595,7
Imprevistos (5%)					29,78
TOTAL					625,48

Elaborado por: Investigador

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Se determinó la ubicación de la cámara a una distancia predeterminada dependiendo de la posición del paciente para evitar falsas alertas, en sedestación la base de la cámara debe medir 1.10 m desde el piso y se debe colocar a 1.50 m partiendo desde los pies del paciente; en postura encamado, la base de la cámara debe tener 1.85m desde el piso y se debe colocar 0.8 m partiendo desde el final de la cama.
- En cuanto al tiempo de respuesta del sistema para detectar la posición del paciente, se ve afectado por un factor externo que es la iluminación, la zona donde se ubique el sistema debe contar con iluminación en un rango de 1500 a 2500 lm el cual debe distribuirse uniformemente por todo el cuerpo, evitando dejar sombras en el rostro del paciente. De esta manera se evita corregir el problema de iluminación mediante algoritmos de programación.
- El componente como la cámara, es aquel que complementa el funcionamiento del sistema. La posible vibración principalmente en la cámara puede ocasionar que el sistema no detecte correctamente lo establecido y que este a su vez arroje falsas alarmas ocasionado alertas indebidas al personal de turno, es por eso que hay que considerar mucho el aspecto de disminuir dicha anomalía fijando la cámara a un soporte para que el sistema pueda brindar una confiabilidad frente a sus usuarios.
- Los resultados finales del prototipo SEAPULP indica un 95% de confiabilidad con bajas tasas de error lo que proporciona una velocidad de procesamiento de 10 fotogramas por segundo, por lo cual el sistema es confiable haciendo que el sistema ayude a disminuir el número de riesgos que puedan agravar a los pacientes durante la rehabilitación.

5.2. Recomendaciones

- Se recomienda que el personal de turno previamente instale la aplicación Telegram en sus dispositivos móviles para que reciban las notificaciones, además de estar conectados a una red wi-fi o de datos con la finalidad de evitar errores en la ejecución del sistema, de esta manera las notificaciones son enviadas en tiempo real como respuesta de algún suceso en la rutina de rehabilitación.
- Se recomienda antes de realizar la adquisición de los frames tener en cuenta la ubicación de la cámara web debido a que cumple un papel crucial que influye en el desempeño del algoritmo desarrollado ya que si no se encuentra en la distancia predeterminada ocasiona falsas detecciones en el paciente.
- Se recomienda realizar mantenimiento constante a los componentes del sistema utilizando una pulsera antiestática, esto ayuda a reducir riesgos mecánicos los cuales afectarían directamente a los resultados del proceso.
- Se recomienda para futuras investigaciones utilizar cámaras específicamente diseñadas para sistemas de visión artificial, este tipo de cámaras reducen el nivel de ruido además de que no dependen en gran medida del factor iluminación, permitiendo que la etapa de detección sea más sencilla generando así mejores resultados del sistema.

BIBLIOGRAFÍA

- [1] Organización Mundial de la Salud. “Seguridad del Paciente”. [En línea]. Disponible en: <http://www.who.int/es/news-room/facts-in-pictures/detail/patient-safety>
- [2] Caple C; Pravikoff D; Úlceras de presión: descripción general; California; Cinahl Information Systems, 2012.
- [3] Chacón JM, Blanes L, Hochman B, Ferreira LM. Prevalence of pressure ulcers among the elderly living in long-stay institutions in São Paulo. *Sao Paulo Med J.* 2009.
- [4] Soldevilla Agreda JJ, Torra i Bou JE, Posnett J, Verdú Soriano J, San Miguel L, Mayan Santos JM. Una aproximación al impacto del coste económico del tratamiento de las úlceras por presión en España. *Gerokomos*, 2007 [En línea]. Disponible en: http://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S1134-928X2007000400007&lng=es
- [5] Organización Mundial de la Salud. “Tecnología de Asistencia”. [En línea]. Disponible en: <http://www.who.int/es/news-room/fact-sheets/detail/assistive-technology>
- [6] D. Hayn, M. Falgenhauer, J. Morak, K. Wipfler, V. Willner, W. Liebhart y G. Schreier. “An eHealth System for Pressure Ulcer Risk Assessment Based on Accelerometer and Pressure Data”. Hindawi Publishing Corporation, 2015. [En línea]. Disponible en: <https://www.hindawi.com/journals/js/2015/106537/>.
- [7] E. Zimlichman, Z. Shinar, R. Rozenblum, S. Levkovich, S. Skiano, M. Szyper-Kravitz, A. Altman, Y. Shoenfeld y H. Amital. “Using continuous motion monitoring technology to determine patient's risk for development of pressure ulcers,” *Journal of Patient Safety*, 2011. [En línea]. Disponible en: <https://journals.lww.com/journalpatientsafety/Abstract/2011/12000/Using-Continuous-Motion-Monitoring-Technology-to.3.aspx>
- [8] O. Chenu, N. Vuillerme, M. Bucki, B. Diot, Y. Payan y F. Cannard. “TexiCare: an innovative embedded device for pressure ulcer prevention. Preliminary results with a

paraplegic volunteer,” *Journal of Tissue Viability*, 2013. [En línea]. Disponible en: <https://www.sciencedirect.com/science/article/pii/S0965206X13000478>

[9] M. Verbunt y C. Bartneck, “Sensing senses: tactile feedback for the prevention of decubitus ulcers,” *Applied Psychophysiology Biofeedback.*, 2010. [En línea]. Disponible en: <https://link.springer.com/article/10.1007/s10484-009-9124-z>

[10] M. Falgenhauer, S. Zöschner, J. Morak, C. Schneider, M. Gugerell, W. Liebhart y D. Hayne. “A patient centered sensor system for decubitus risk prediction and prevention,” *Biomedizinische Technik.*, 2013. [En línea]. Disponible en: <http://europepmc.org/abstract/med/23920807>

[11] Gallego, T. “Bases teóricas y fundamentos de la fisioterapia”. Ed Médica Panamericana, 2007.

[12] Defloor T, Schoonhoven L. Inter-rater reliability of the EPUAP pressure ulcer classification system using photographs. *J Clin Nurs*, 2004.

[13] E. Gallart y cols. Estudio experimental para comprobar efectividad de los ácidos grasos hiperoxigenados en la prevención de úlceras por presión en pacientes ingresados. *Enfermería*, 2001.

[14] V. Ayala. “Cuidados en Úlceras por Presión Parte I”, 2012

[15] G. Fernández, P. Francisco, otros. “Úlceras por presión en el paciente crítico”, 2009.

[16] R. Torres, S. Agreda. “Juicio clínico o escalas de valoración para identificar a los pacientes en riesgo de desarrollar úlceras por presión”. *Gerokomos*, 2007.

[17] J. Jiménez. “Protocolo para la prevención y manejo de úlceras por presión en pacientes hospitalizados”. Dirección de Secretaría de Salud México, 2012.

[18] European Pressure Ulcer Advisory Panel and National Pressure Ulcer Advisory Panel. *Treatment of pressure ulcers: Quick Reference Guide*. Washington. Washington: DC: National Pressure Ulcer Advisory Panel, 2009.

- [19] Defloor T, De Bacquer D, Grypdonck MH. The effect of various combinations of turning and pressure reducing devices on the incidence of pressure ulcers. *Int J Nurs Stud*, 2005.
- [20] Registered nurses Association of Ontario. *Nursing Best Practice Guiderlines Program. Risk Assessment & Prevention of Pressure Ulcer*. 3rd ed. Toronto: SCO Health Services; 2011.
- [21] E. McInnes, A. Jammali-Blasi, Bell-Syer SEM, JC. Dumville, N. Cullum. “Support surfaces for pressure ulcer prevention”. *Cochrane Database of Systematic Reviews*, 2011.
- [22] R. Chou, T. Dana, C. Bougatsos, I. Blazina, A. Starmer, K. Reitel, D. Buckley. *Pressure Ulcer Risk Assessment and Prevention: Comparative Effectiveness. Comparative Effectiveness Review No. 87*, 2013.
- [23] D. Colin, J. Rochet, P. Ribinik, B. Barrois, Y. Passadori, J. Michel. What is the best support surface in prevention and treatment, for a patient at risk and/or suffering from pressure ulcer sore?. *Ann Phys Rehabil Med*, 2012.
- [24] The Joanna Briggs Institute. *JBIC Connect evidence summary: Pressure Ulcers (Prevention): Support Surfaces*, 2012.
- [25] Asociación Profesional de Enfermeras de Ontario. “Valoración y manejo de las lesiones por presión para equipos interprofesionales”. Tercera edición. Toronto. *Guía de buenas prácticas*, 2016.
- [26] Osakidetza. “Guía de recomendaciones basadas en la evidencia en Prevención y Tratamiento de las úlceras por presión en adultos”, 2015.
- [27] G. Koel, P. Houghton. “Electrostimulation: Current Status, Strength of Evidence Guidelines, and Meta-Analysis”. *Adv Wound Care (New Rochelle)*, 2014.
- [28] N. Maeshige, H. Fujiwara, H. Honda. “Evaluation of the combined use of ultrasound irradiation and wound dressing on pressure ulcers”. *J Wound Care*, 2010.

- [29] A. Polak, A. Franek, E. Blaszczyk. "A prospective, randomized, controlled, clinical Study to evaluate the efficacy of high-frequency ultrasound in the treatment of stage 2 and stage 3 pressure ulcers in geriatric patients". *Ostomy Wound Manage*, 2014.
- [30] M. Regan, R. Teasell, D. Wolfe. "A Systematic Review of Therapeutic Interventions for Pressure Ulcers Following Spinal Cord Injury", 2009.
- [31] H. McGaughey, S. Dhamija, L. Oliver. "Pulsed electromagnetic energy in management of chronic wounds: A systematic review", 2009.
- [32] National Pressure Ulcer Advisory Panel, European Pressure Ulcer Advisory Panel, & Pan Pacific Pressure Injury Alliance (NPUAP, EPUAP & PPPIA 2014). "Prevención y tratamiento de las úlceras por presión: Guía de consulta rápida". Segunda edición.
- [33] A. Onigbinde, R. Adedoyin, O. Ojoawo. "Effects of ultraviolet radiation (type B) on wound exudates, appearance and depth description". *Technology and Health Care*, 2010.
- [34] F. Andrade, R. Clark, M. Ferreira. "Effects of low-level laser therapy on wound healing". *Rev Col Bras Cir*, 2014.
- [35] Asociación Profesional de Enfermeras de Ontario. "Valoración del riesgo y prevención de las úlceras por presión. Guía de buenas prácticas en enfermería. Cómo enfocar el futuro de la enfermería". Hendrichova, 2011.
- [36] Osakidetza. "Guía de recomendaciones basadas en la evidencia en Prevención y Tratamiento de las úlceras por presión en adultos", 2015.
- [37] C. Gonzales, R. Woods. "Procesamiento de Imágenes Digitales". Segunda Edición.
- [38] R. Gonzalez, R. Woods. "Digital Image Processing". Second Edition. Editorial Prentice-Hall, United States of America, 2002.
- [39] J. Mantilla. "Visión Artificial". Universidad Pontificia Comillas, Escuela Técnica Superior de Ingeniería (ICAI), España, Madrid, 2015. [En línea]. Disponible en: <http://www.etitudela.com/ce/downloads/visionartificial.pdf>.

- [40] J. Cancelas. “Introducción a la Visión por Computador”. España. 2010. [En línea]. Disponible en: http://isa.uniovi.es/~cancelas/doctorado/sis_percep0.pdf.
- [41] A. González, F. Martínez, V. Pernía, M. Castejón, J. Ordieres. “Técnicas y Algoritmos Básicos de Visión Artificial”. Universidad de la Rioja. 2006. [En línea]. Disponible en: <https://publicaciones.unirioja.es/catalogo/online/VisionArtificial.pdf>
- [42] Raspberry Pi. “Características técnicas de Raspberry Pi 3”. [En línea]. Disponible en: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [43] TDRobotica. “Guía básica de Arduino: fundamentos y ejercicios prácticos”. [En línea]. Disponible en: http://tienda.tdrobotica.co/download/Libro_kit_Basico.pdf
- [44] Orange Pi. “La nueva generación de mini ordenadores”. [En línea]. Disponible en: <https://orangepiweb.es/>
- [45] BeagleBoard. “Trabajando con tarjetas electrónicas BeagleBoard” [En línea]. Disponible en: <https://beagleboard.org/black>
- [46] J. Paredes. “Características de Visual Studio”. [En línea]. Disponible en: <http://naimcruzado.blogspot.com/2010/06/caracteristicasdevisualstudio2010.html>.
- [47] M. Álvarez. “Lenguaje de programación Python: desarrollo práctico”, 2013. [En línea]. Disponible en: <http://www.desarrolloweb.com/articulos/1325.php>.
- [48] National Instruments. “Principales características de LabVIEW”. [En línea]. Disponible en: <http://www.ni.com/es-cr/shop/labview.html>
- [49] Control Tutorials for Matlab - Simulink. [En línea]. Disponible en: http://ctms.engin.umich.edu/CTMS/index.php?aux=Basics_Matlab.
- [50] R. Klette, 2014. Concise computer vision: An introduction into theory and algorithms. Springer, 375- 413.
- [51] K. Grauman, & B. Leibe, 2011. Visual object recognition. San Rafael, Calif.: Morgan & Claypool

[52] G. Ottado, «www.eva.fing.edu.uy,» 2010. [En línea]. Disponible en: https://eva.fing.edu.uy/file.php/514/ARCHIVO/2010/TrabajosFinales2010/informe_final_ottado.pdf

[53] P. Viola & M. Jones, “Rapid object detection and recognized using a boosted cascade of simple feactures”, in proceeding of the 2013, IEEE Computer Society Conference on computer vision and Patter Recognition.

[54] M. Guevara, D. Echeverry, W. Ureña. “Detección de Rostros en imágenes Digitales usando Clasificadores en Cascada”, Numero de publicación ISSN 0122-1701.

[55] OSL.ULL. “Programas para el estudio de visión artificial: guías prácticas”, Universidad de las Lagunas, 2016. [En línea]. Disponible en: <https://osl.ull.es/software-libre/opencv-libreria-vision-computador/>.

[56] Telegram Bot API. . [En línea]. Disponible en: <https://core.telegram.org/bots/api>

[57] Ministerio del Trabajo, “Sueldos mínimos sectoriales 2018”. [En línea]. Disponible en: <http://www.ecuadorlegalonline.com/laboral/tabla-salarios-minimos-sectoriales-2018/>

ANEXOS

ANEXO A: INSTALACION DE SOFTWARE

INSTALACION DE SOFTWARE

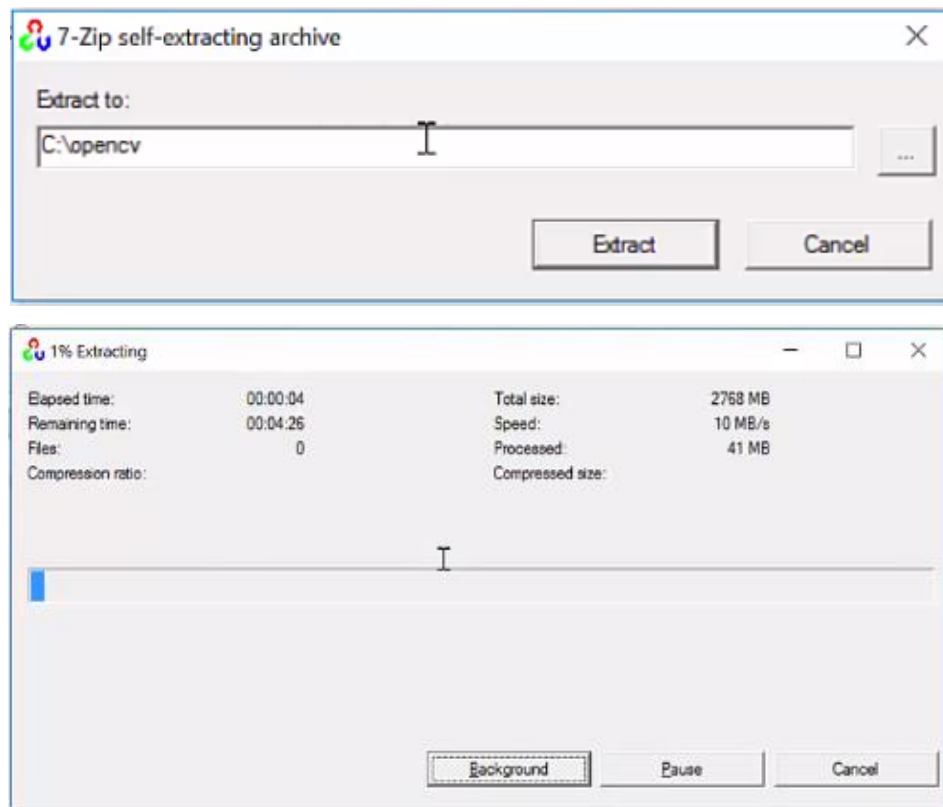
Para un correcto funcionamiento de SEAPULP es necesario instalar el software específico que trabaje con visión artificial. Para el sistema se necesita:

- OpenCV
- Python

INSTRUCCIONES PARA INSTALAR OPENCV Y PYTHON

OPENCV

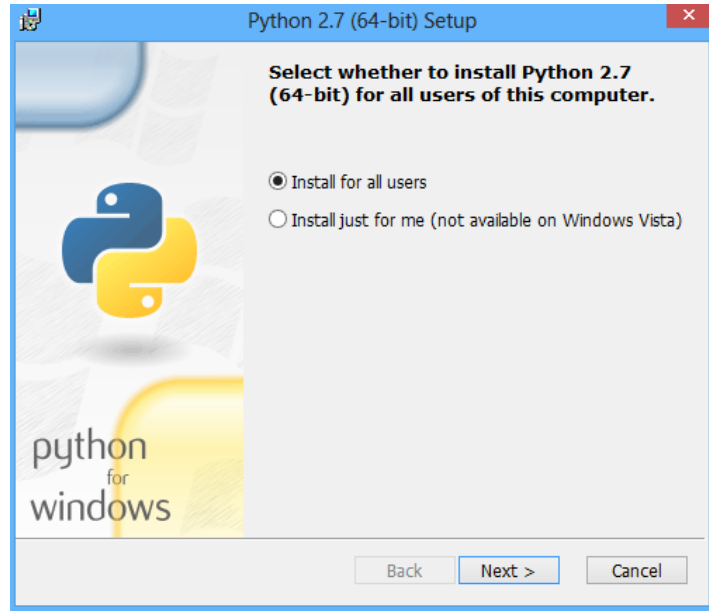
1. Descargar la versión 2.4.12 OpenCV.
2. Ejecutar opencv2.4.12 y direccionamos al disco C como se ve en la imagen. Ahora la librería OpenCV se encuentra descomprimida en la dirección C.



Elaborado por: Investigador

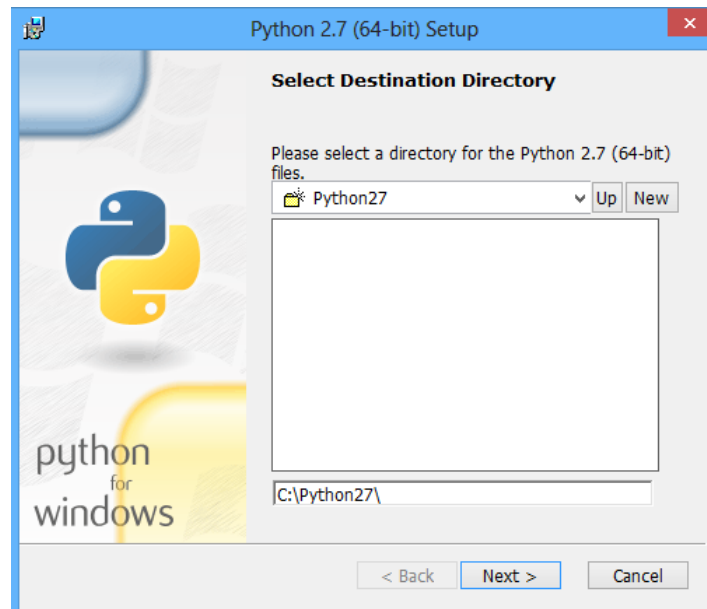
PYTHON

1. Descargar Python 2.7.9 y ejecutarlo como administrador.



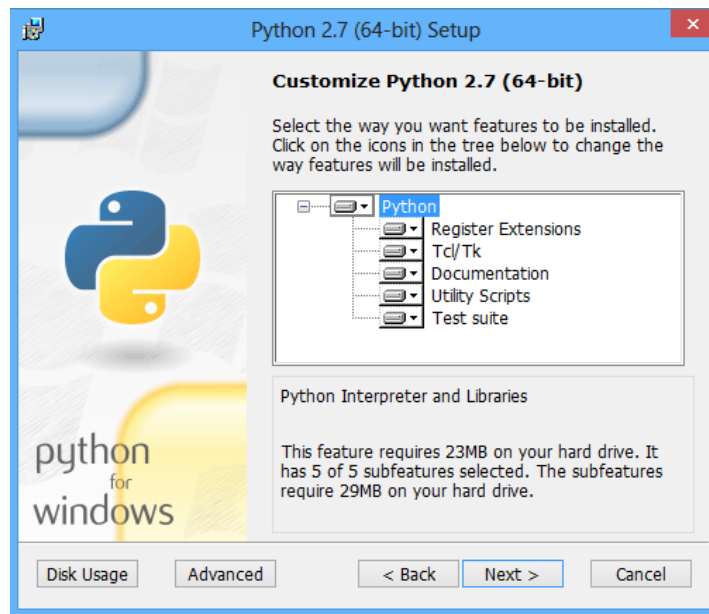
Elaborado por: Investigador

2. Direccionar al disco C.

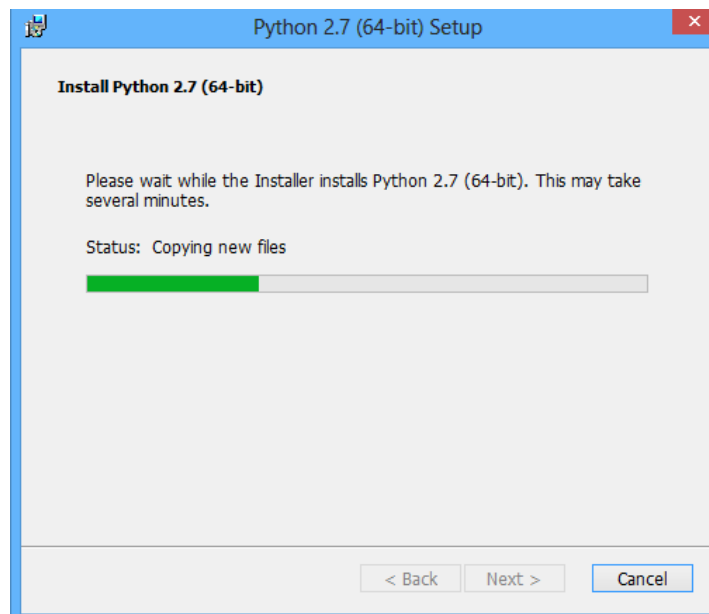


Elaborado por: Investigador

3. Clic en el botón Next.



Elaborado por: Investigador



Elaborado por: Investigador

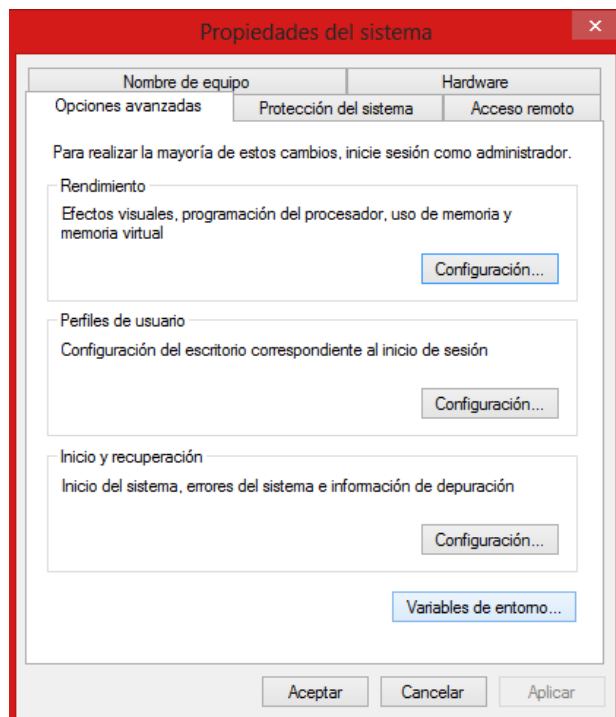
4. Clic en el botón Finish para terminar con la instalación.



Elaborado por: Investigador

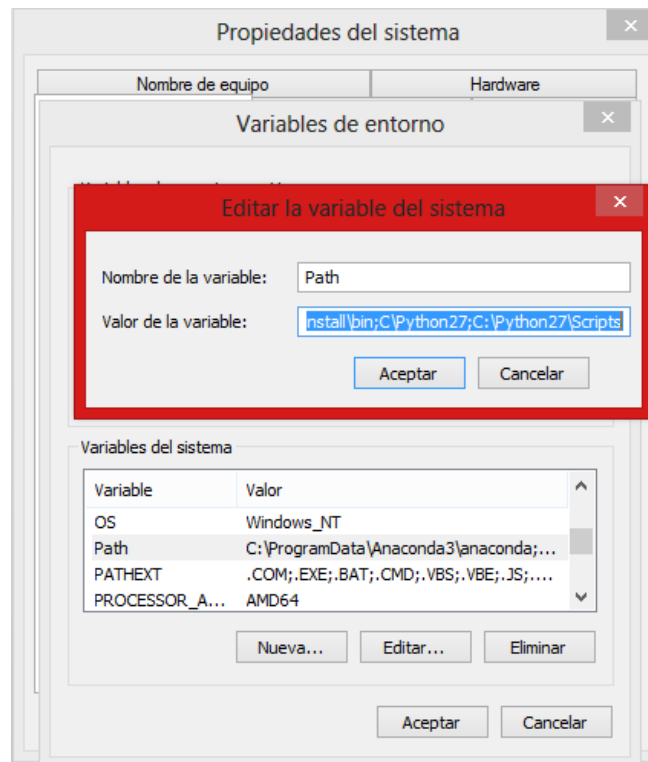
Después de su instalación es necesario configurar la “Variable de Entorno”

5. Clic derecho sobre el icono de Equipo->Propiedades y se abrirá una ventana Sistema, buscamos la opción Configuración Avanzada del sistema. Una vez situado ahí dar clic en el botón “Variables de Entorno”.



Elaborado por: Investigador

6. Dirigirse a “Variables del sistema” y damos clic en Path y en el campo Valor de la variable agregar un ; la ruta donde se instaló python. C:\Python27



Elaborado por: Investigador

INSTALACION DE LIBRERIAS Y PAQUETES

Para desarrollar el código es necesario instalar los siguientes paquetes para evitar conflictos en la compilación del mismo, para ello necesitamos:

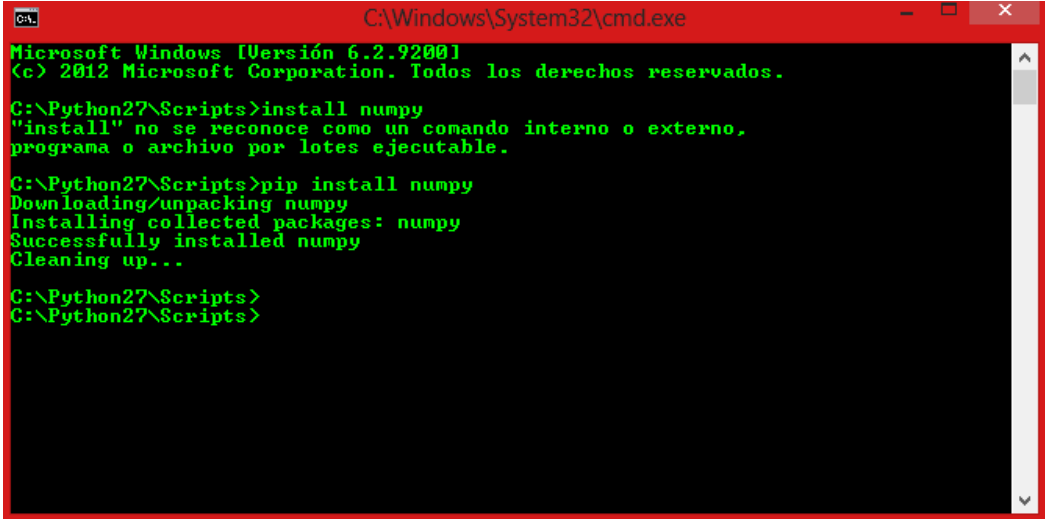
7. Ahora para instalar la librería de Opencv en Python, dirigirse al directorio de la carpeta C:\opencv\build\python\2.7\x64 (x64 o x86 según tú interprete) dentro vamos a encontrar un archivo llamado cv2.pyd, lo vamos a copiar dentro de C:\Python27\Lib\site-packages.
8. Si todo está correctamente instalado y seguiste estos pasos, abre tu IDLE de Python y ejecuta lo siguiente:

```
>>> import cv2
```

```
>>> cv2.__version__
```

```
'2.4.12'
```

9. Si muestra un resultado parecido o igual, quiere decir que ya estás listo para continuar.
10. Numpy, paquete fundamental para la informática científica con Python. Dirigirse al símbolo del sistema y ejecutar la siguiente línea: `install numpy`.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 6.2.9200]
(c) 2012 Microsoft Corporation. Todos los derechos reservados.

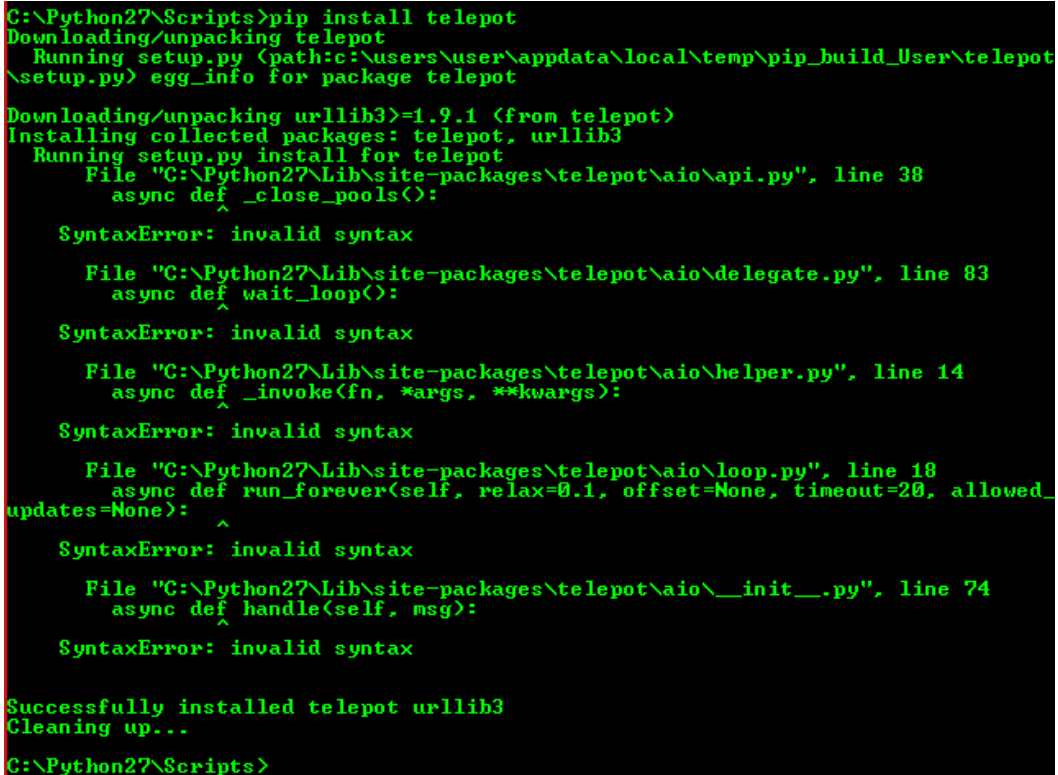
C:\Python27\Scripts>install numpy
"install" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Python27\Scripts>pip install numpy
Downloading/unpacking numpy
Installing collected packages: numpy
Successfully installed numpy
Cleaning up...

C:\Python27\Scripts>
C:\Python27\Scripts>
```

Elaborado por: Investigador

11. Telepot, este paquete ayuda a crear aplicaciones para la API de Telegram Bot, ejecutamos: `install telepot`.



```
C:\Python27\Scripts>pip install telepot
Downloading/unpacking telepot
  Running setup.py (path:c:\users\user\appdata\local\temp\pip_build_User\telepot\setup.py) egg_info for package telepot

Downloading/unpacking urllib3>=1.9.1 (from telepot)
Installing collected packages: telepot, urllib3
  Running setup.py install for telepot
    File "C:\Python27\Lib\site-packages\telepot\aiolib\api.py", line 38
      async def _close_pools():
        ^
    SyntaxError: invalid syntax

    File "C:\Python27\Lib\site-packages\telepot\aiolib\delegate.py", line 83
      async def wait_loop():
        ^
    SyntaxError: invalid syntax

    File "C:\Python27\Lib\site-packages\telepot\aiolib\helper.py", line 14
      async def _invoke(fn, *args, **kwargs):
        ^
    SyntaxError: invalid syntax

    File "C:\Python27\Lib\site-packages\telepot\aiolib\loop.py", line 18
      async def run_forever(self, relax=0.1, offset=None, timeout=20, allowed_updates=None):
        ^
    SyntaxError: invalid syntax

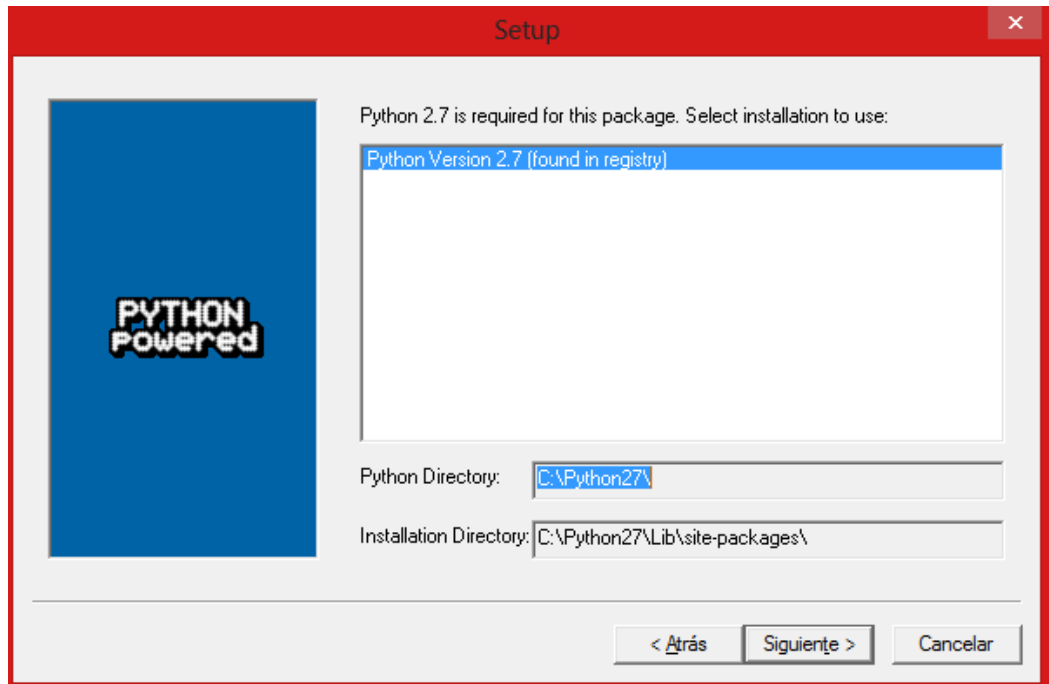
    File "C:\Python27\Lib\site-packages\telepot\aiolib\__init__.py", line 74
      async def handle(self, msg):
        ^
    SyntaxError: invalid syntax

Successfully installed telepot urllib3
Cleaning up...

C:\Python27\Scripts>
```

Elaborado por: Investigador

12. Pillow (PIL), es la biblioteca de imágenes de Python, es útil para crear la interfaz del sistema. Una vez descargado la biblioteca procedemos con la instalación para ello: Ejecutamos el archivo: Pillow-2.5.3.win-amd64-py2.7.exe como administrador -> Siguiente. La ruta de instalación debe ser como la figura C:\Python27\ ->Siguiente y Finalizar.



Elaborador por: Investigador

ANEXO B: MANUAL DE USUARIO

MANUAL DE USUARIO



ÍNDICE MANUAL DE USUARIO

1.INGRESO A SEAPULP	97
2. MENU DE SEAPULP	97
2.1. INFORMACION	97
2.2. DATOS DEL PACIENTE	98
2.3. MONITOREO.....	99
2.4. RESPALDO DE PACIENTES REGISTRADOS.....	100
2.5.REPORTES.....	101
2.6. SALIR	102

INTRODUCCIÓN

El presente manual tiene como finalidad proporcionar al usuario las indicaciones generales para la utilización del Sistema Electrónico para asistir a pacientes en proceso de rehabilitación de Ulceras por Presión denominado SEAPULP.

SEAPULP es un sistema electrónico destinado a la detección de la posición y movimientos de los pacientes con UPP en tiempo real. El prototipo cuenta con una estructura de hardware que puede ser adaptado en cualquier habitación o espacio con condiciones de iluminación normales.

Entre las características que ofrece el sistema se encuentran:

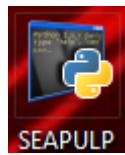
- Interfaz gráfica de fácil manejo para el usuario, donde puede monitorear al paciente en tiempo real.
- Acceso constante a la base de datos de los pacientes registrados, aquí se encuentran los datos de las personas que ingresan a terapia como son: nombre, cedula, dirección, teléfono, día y hora de ingreso además del nombre del fisioterapeuta en turno, por otra parte se puede eliminar la base de datos en caso de requerirlo.
- Envía un reporte al familiar una vez terminada la rutina de rehabilitación, en el reporte se incluyen datos como: día, hora de ingreso y salida, nombre del paciente, cedula, dirección, tiempo total del monitoreo y porcentaje de movimientos detectados.

MANUAL DE SEAPULP

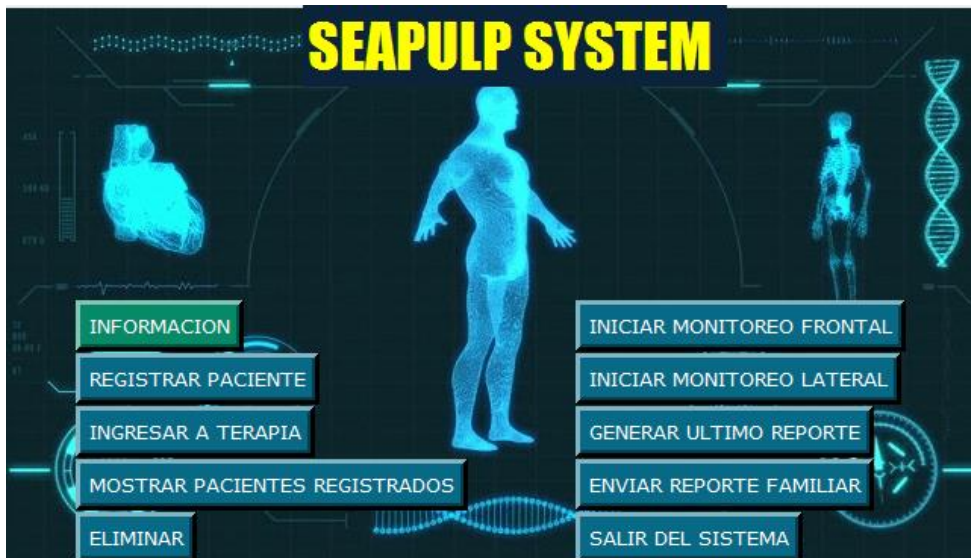
Antes de iniciar el sistema, conecte su dispositivo móvil y el prototipo a internet mediante wi-fi o datos móviles.

1. Ingreso a SEAPULP

Doble clic sobre el Icono SEAPULP para ingresar al sistema.



Debe visualizar la ventana principal donde se observa botones que cumplen diferentes funciones dentro del sistema.

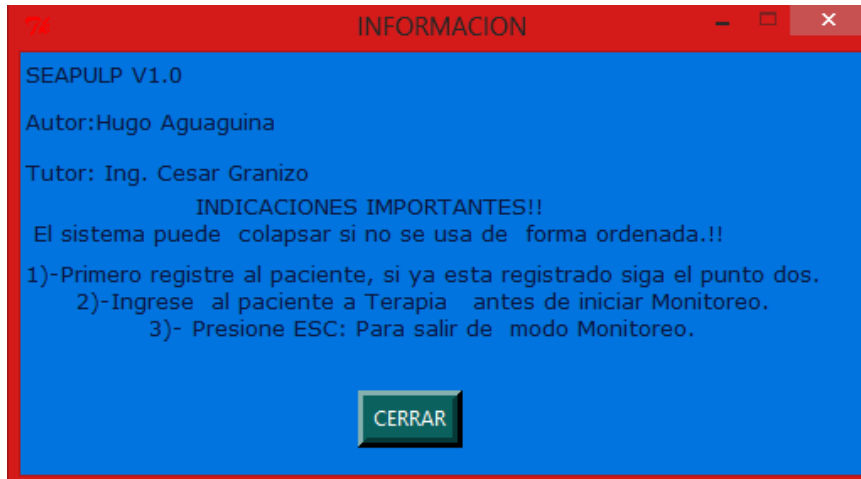


2. Menú de SEAPULP

El sistema dispone con las siguientes opciones:

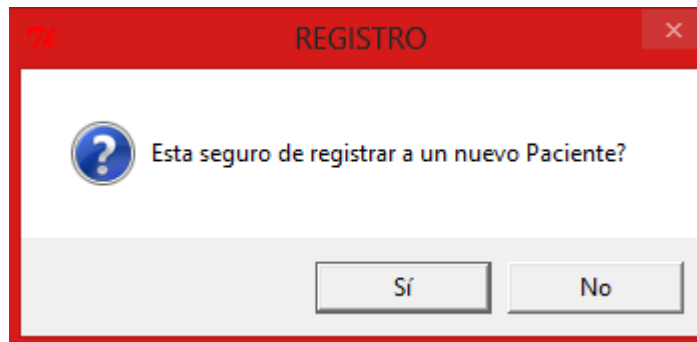
2.1 Información

Cuenta con indicaciones generales para darle un uso ordenado del sistema y evitar que colapse el mismo, además de datos del autor del sistema.



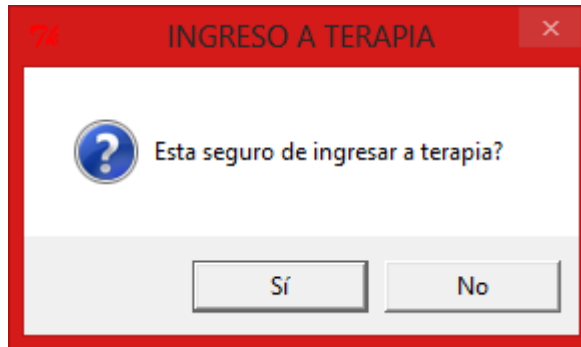
2.2 Datos del Paciente

Registrar Paciente: Ingresar datos personales del paciente.



```
BIENVENIDOS AL SISTEMA SEAPULP...  
Nuevo registro:  
IngreseCodigo ID:  
Ingrese Nombre y Apellido:  
Ingrese Cedula:  
Ingrese la direccion:  
Ingrese telefono:
```

Ingresar a Terapia: Datos del paciente que va a ingresar a terapia y nombre del fisioterapeuta que se encuentre en turno.



```
Nombre del paciente:  
Ingrese Cedula:  
Nombre del fisioterapeuta encargado:  
Ingrese la Direccion:
```

2.3. Monitoreo

Iniciar monitoreo Frontal: Detecta el rostro frontal del paciente así como el movimiento de sus articulaciones. Para salir de modo monitoreo presionar la tecla ESC.



Iniciar monitoreo Lateral: Detecta el rostro de perfil izquierdo del paciente así como el movimiento de sus articulaciones. Para salir de modo monitoreo presionar la tecla ESC.



2.4. Respaldo de pacientes registrados

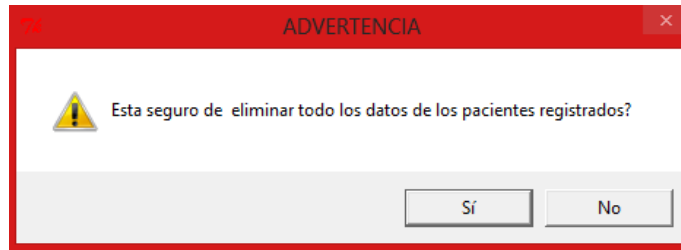
Mostrar pacientes Registrados: muestra todos los pacientes registrados durante la jornada de trabajo del fisioterapeuta.



Mostrar Registros

```
_Hora de registro:_Monday:17:46:34;_Codigo:_1;_Nombre:_Karla Sanchez;_Cedula:_0123456789;_Direccion:_Pelileo;_Celular:_0998416939;
_Hora de registro:_Monday:17:49:47;_Codigo:_2;_Nombre:_Daniel Aguaguina;_Cedula:_0123456789;_Direccion:_Pelilao;_Celular:_0369852147;
```

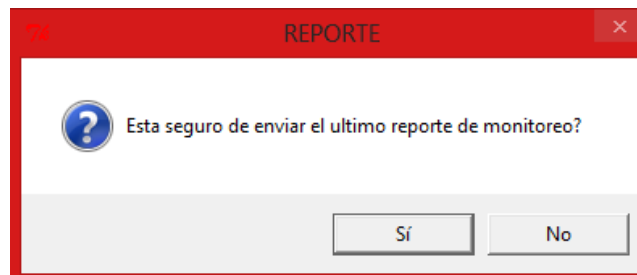
Eliminar: Borra todos los registros de los pacientes.



```
Eliminando la base de datos...
Registros eliminados
```

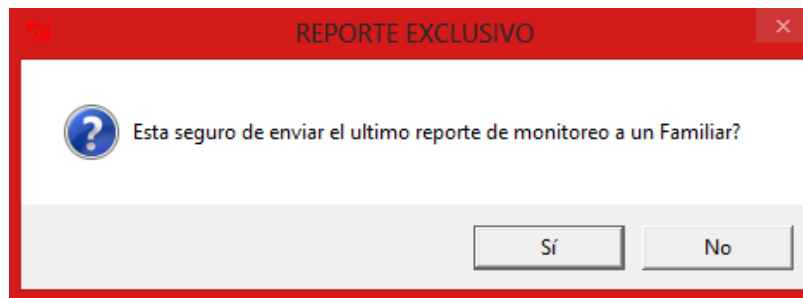
2.5. Reportes

Generar último Reporte: Envía el reporte al dispositivo móvil del fisioterapeuta encargado.



```
Detalle de ultimo monitoreo
del paciente: __Hora de
entrada: _Wednesday:
17:51:03; _Nombre y apellidos: _Daniel
Aguaguina; _Cedula: _1804785531; _Dir
eccion: Ambato;
_Monitoreo Finalizado a: _Wednesday:
18:21:09 _Tiempo promedio de
paciente No detectado: seg: _ 0
_Tiempo total de monitoreo: seg: _ 4.0
_Porcentaje Movimiento Detectados: _
%_87.5
6:21 PM
```


Enviar reporte familiar: Envía el reporte al dispositivo móvil del familiar del paciente.

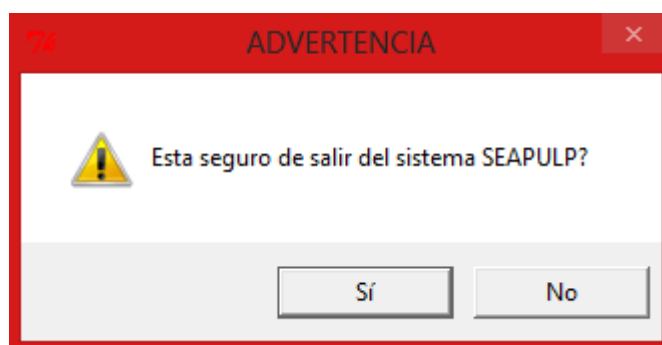


Ultimo reporte de paciente:
__Hora de entrada:_Monday:
17:49:52;_Nombre y
apellidos:_Daniel Aguaguina;_Cedula
:_0123456789;_Direccion:Ambato; 17:50

Datos Registrados: _Monitoreo
Finalizado:_Monday:
17:50:31;Tiempo no detectado
seg:_4;Tiempo total de monitoreo
seg:_15.0;Porcentaje de
movimientos %:_29; 17:50

2.6. Salir del Sistema

Salir: Si desea cerrar el sistema haga clic en el botón “SALIR”. Se muestra un cuadro de dialogo para confirmar.



ANEXO C: FOTOGRAFIAS

SISTEMA SEAPULP









**ANEXO D: CODIGO DE PROGRAMACION
EN PYTHON**

CODIGO DE PROGRAMACION EN PYTHON

```
import numpy as np
import cv2
import pygame
import time
import sys
import os
import telepot
import datetime
from Tkinter import*
import Tkinter as tk
from PIL import Image, ImageTk
import tkMessageBox
bot = telepot.Bot('481163786:AAHo20FR1yqjVfldSj7rhQ11w3G1EiVFwPI')

pygame.mixer.init()
def informacion():
    win=tk.Tk()
    win.geometry("500x250+100+115")
    win.title("INFORMACION")
    win.configure(background='#0174DF')

    etiquetaI=Label(win,text="SEAPULP
V1.0",fg="#0B173B",bg="#0174DF",font=("verdana",10)).place(x=0,y=2)
    etiquetaI2=Label(win,text="Autor:Hugo                               Aguaguña",
fg="#0B173B",bg="#0174DF",font=("verdana",10)).place(x=0,y=30)
    etiquetaI3=Label(win,text="Tutor:          Ing.          Cesar          Granizo",
fg="#0B173B",bg="#0174DF",font=("verdana",10)).place(x=0,y=60)
    etiquetaI4=Label(win,text="INDICACIONES IMPORTANTES!! \n El sistema puede
colapsar          si          no          se          usa          de          forma
ordenada.!!",fg="#0B173B",bg="#0174DF",font=("verdana",10)).place(x=0,y=80)
    etiquetaI5=Label(win,text="1)-Primero registre al paciente, si ya esta registrado siga
el punto dos.\n2)-Ingrese al paciente a Terapia antes de iniciar Monitoreo.\n3)- Presione
ESC:          Para          salir          de          modo
Monitoreo.",fg="#0B173B",bg="#0174DF",font=("verdana",10)).place(x=0,y=120)
    win.resizable(width=False, height=False)

    boton=tk.Button(win,text='CERRAR',fg="white",activebackground="green",command
=win.destroy,bg="#0B615E",anchor="center",bd=6,relief="raised").place(x=200,y=200
)

    win.mainloop()
def handle(msg):

    command = msg['text']
```



```

datos=[0]
if command == 'reporte':

    archivo=open("archivo11.csv","r")
    for x in archivo:
        datos[0]=x

        print x
        print "\n"

        bot.sendMessage(519748192," Ultimo reporte online
paciente:"+datos[0])

        print "enviado del Reporte"
    archivo.close()

    archivo1=open("registro.csv","r")
    for t in archivo1:

        print "\n"
        print t
        print "\n"
        bot.sendMessage(519748192,"Datos Registrados: "+t)

    archivo1.close()
    print "REPORTE ENVIADO"

elif command =='reporte t':

    archivo=open("archivo.csv","r")
    for x in archivo:
        datos[0]=x

        print x
        print "\n"

        bot.sendMessage(519748192," REPORTE DE REGISTROS DE
PACIENTES "+datos[0])

    archivo.close()
    print "REPORTE ENVIADO"

bot = telepot.Bot('481163786:AAHo20FR1yqjVfldSj7rhQ11w3G1EiVFwPI')
bot.message_loop(handle)
print 'Estoy escuchando...'
menu():

```

```

    print "BIENVENIDOS AL SISTEMA SEAPULP..."
menu()

def ingreso():
    result=tkMessageBox.askquestion("INGRESO A TERAPIA", "Esta seguro de
ingresar a terapia?")
    if result=='yes':
        print "\n\n"
        archivo=open("archivo11.csv","w")
        y=str(time.strftime("%A:%H:%M:%S"))
        paciente=raw_input("Nombre del paciente: ")
        #hora=raw_input("Hora de Entrada: ")
        #salida=raw_input("Hora de Salida: ")
        cedula=raw_input("Ingrese Cedula: ")
        doctor=raw_input("Nombre del fisioterapeuta encargado: ")
        direccion=raw_input("Ingrese la Direccion: ")
        bot.sendMessage(519748192," INGRESO A TERAPIA: Paciente:_ " +paciente+ "
_Hora de entrada:_ "+y+ "_Fisioterapeuta encargado:_ " +doctor+ " _Direccion:_ "
+direccion)

        ent="_Hora de entrada:_ " +y
        nomb="_Nombre y apellidos:_ " +paciente
        ced="_Cedula:_ " +cedula
        dirc="_Direccion:" +direccion
        doct="_Fisioterapeuta encargado:_ " +doctor
        archivo.write(ent)
        archivo.write(";")
        archivo.write(nomb)
        archivo.write(";")
        # archivo.write(salida)
        # archivo.write(";")
        archivo.write(ced)
        archivo.write(";")
        archivo.write(dirc)
        archivo.write(";")
        archivo.write("\n")

        archivo.close()
        print "\n"
        print "Ingreso terminado... vaya al Panel principal, e inicie Monitoreo.."
        print "....."
    else:
        print"\n"
        print" TENGA UN EXCELENTE DIA!!"

return

```

```

def Registro ():
    result=tkMessageBox.askquestion("REGISTRO", "Esta seguro de registrar a un
nuevo Paciente?")
    if result=='yes':
        print "Nuevo registro:\n"
        archivo=open("archivo.csv","a")
        codigo=raw_input("IngreseCodigo ID: ")
        nombre=raw_input("Ingrese Nombre y Apellido: ")
        cedula=raw_input("Ingrese Cedula: ")
        direccion=raw_input("Ingrese la direccion: ")
        telefono=raw_input("Ingrese telefono: ")
        y=str(time.strftime("%A:%H:%M:%S"))
        bot.sendMessage(519748192," SE REGITRO A:" + nombre + "_Con codigo:_ "
+ codigo + "_Hora de registro:_ " + y + "_Con Cedula:_ " + cedula + "_Direccion:_" +
direccion + "_Celular:_" +telefono)
        print "Se ha registradoo "+nombre+ " C.I.:"+cedula+ " DIR.:" +direccion+ " TEL.:"
+telefono
        HR="_Hora de registro:_" +y
        codg="_Codigo:_" +codigo
        NomR="_Nombre:_" + nombre
        cedR="_Cedula:_" +cedula
        dirR="_Direccion:_" +direccion
        tel="_Celular:_" +telefono
        archivo.write(HR)
        archivo.write(";")
        archivo.write(codg)
        archivo.write(";")
        archivo.write(NomR)
        archivo.write(";")
        archivo.write(cedR)
        archivo.write(";")
        archivo.write(dirR)
        archivo.write(";")
        archivo.write(tel)
        archivo.write(";")
        archivo.write("\n")

        archivo.close()
        print "\n"
        print "Nuevo Registro Terminado... vaya al Panel principal.."
        print "....."
    else:
        print"\n"
        print" TENGA UN EXCELENTE DIA!!"
        print"\n"

```

```

    return

def Mostrar():
    result=tkMessageBox.showinfo("REGISTROS", "SE MOSTRARA TODO LOS
PACIENTES REGISTRADOS!!")
    print "Mostrar Registros"
    archivo=open("archivo.csv","r")
    print "\n"
    datos=[0]
    for x in archivo:
        print x
    archivo.close()
    print "\n"
    print "Proceso Terminado... vaya al Panel principal....."
    return
def eliminar():
    result=tkMessageBox.askquestion("ADVERTENCIA", "Esta seguro de eliminar todo
los datos de los pacientes registrados?", icon='warning')
    if result=='yes':

        archivo=open("archivo.csv","a")
        archivo.truncate()
        print "Eliminando la base de datos..."
        archivo.close()
        print "Registros eliminados"
    else:
        print"\n"
        print" TENGA UN EXCELENTE DIA, LOS REGISTROS NO SE HAN
ELIMINADO!!"
        print"\n"
    return

#DETECCION DECUBITO LATERAL
def capturaP():
    inicio=time.time()
    s=0
    msj=0
    movi=0
    movi2=0
    size=2
    t2=0
    pygame.mixer.init()

    perfil_cascade = cv2.CascadeClassifier('haarcascade_profileface.xml')

    cap = cv2.VideoCapture(0)
    cap.set(3,250)

```

```

cap.set(4,400)
_,prev=cap.read()
prevG=cv2.cvtColor(prev,cv2.COLOR_RGB2GRAY)
kernel=np.ones((5,5),np.uint8)
font=cv2.FONT_HERSHEY_SIMPLEX
PrevCen=np.array( [0,0])
bot = telepot.Bot('481163786:AAHo20FR1yqjVfldSj7rhQ11w3G1EiVFwPT')
ventana=" DETECTOR PERFIL"

cv2.namedWindow(ventana, cv2.WINDOW_NORMAL)

cv2.resizeWindow(ventana,550,300)

while 1:
    _,frame= cap.read()

    y=str(time.strftime("%A:%H:%M:%S"))
    formato = "%X"
    hora = time.strftime(formato) #variable tipo hora
    cv2.putText(frame, y,(5,270),font,0.5,(25,255,20))
    cv2.putText(frame, "PRESIONE ESC: PARA
SALIR",(5,20),font,0.4,(100,235,135),1)
    fGray=cv2.cvtColor(frame,cv2.COLOR_RGB2GRAY)
    #Gray=cv2.cvtColor(frame1,cv2.COLOR_RGB2GRAY)
    mini = cv2.resize(fGray, (frame.shape[1] / size, frame.shape[0] / size))

    perfil=perfil_cascade.detectMultiScale(mini)
    # faces = face_cascade.detectMultiScale(fGray, 1.3, 5)
    flow=np.array(abs(np.array(fGray,np.float32)-
np.array(prevG,np.float32)),np.uint8)
    #cv2.imshow(ventana,frame)
    rang=cv2.inRange(flow,20,255)
    #cv2.imshow('rang',flow)
    opening=cv2.morphologyEx(rang, cv2.MORPH_OPEN,kernel)
    closing=cv2.morphologyEx(opening, cv2.MORPH_CLOSE,kernel)
    # cv2.imshow('CLOSING',closing)

    M=[0,0]
    nCaras=[0]
    nPerfil=[0]
    nMovi=[0]
    n=0

    nPerfil[0]=len(perfil)
    def click_cerrar(event,x,y,flags,param):
        if event == cv2.EVENT_FLAG_LBUTTON:
            if x<50 and x<50 :

```

```

        cap.release()
        cv2.destroyAllWindows()
        return

cv2.setMouseCallback(ventana,click_cerrar)
if nPerfil[0]!=0:
    contours,hierarchy=cv2.findContours(closing,1,2)
    nMovi[0]=len(contours)

    if nMovi[0]>=1:

        for cnt in contours:
            inicio2=time.time()

            #print (" movimientos",movi )

            x,y,w,h =cv2.boundingRect(cnt)
            if w>20 and h>20 and w<500 and h<500:
                movi+=1
                movi2+=1
                cv2.putText(frame,                                "MOVIMIENTO
DETECTADO!",(135,230),font,0.5,(0,0,255),1)
                cv2.rectangle(frame,(x,y),(x+w,y+h),(255,255,0),1)

                #print (" El paciente se movio",movi )
                #if movi==4:
                    #bot.sendMessage(518009346,"Hola, hay movimiento!!, Por
favor verifique")
                if movi==5:
                    pygame.mixer.music.load("movi1.wav")
                    pygame.mixer.music.play(1)

                    if movi==30:
                        movi=0
                if nMovi[0]==0:
                    cv2.putText(frame,                                "PACIENTE
ESTABLE!",(150,270),font,0.5,(200,255,20))
                    movi=0

            if nPerfil[0]==0:

                msj+=1

                cv2.putText(frame, "ALERTA 2!!",(100,100),font,1.3,(0,0,255),2)

```

```

        if msj==4:
            bot.sendMessage(519748192,"Hola, el paciente No esta en Posicion
Perfil!!, Por favor verifique")
            n=1
        if msj==6:
            pygame.mixer.music.load("paciente2.wav")
            pygame.mixer.music.play(1)
            n=n+3

            t=n

            t1=t
            t2=(t1+t)/2

        if msj==40:
            msj=5

    else:
        msj=0

        for f in perfil:
            (x, y, w, h) = [v * size for v in f]
            cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),thickness=4)
            cv2.putText(frame, 'POSICION DE PERFIL DETECTADO',(x-20,y-
10),font,0.4,(25,255,20))
            s=s+1

        if(s>400):
            s=0
            bot.sendMessage(519748192,"ES NECESARIO CAMBIAR DE POSICION
AL PACIENTE, EL TIEMPO DE RUTINA A TERMINADO ")

    prevG=fGray

    cv2.imshow(ventana,frame)
    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break
    final=time.time()
    tiempo=round( final-inicio,0)
    cap.release()
    cv2.destroyAllWindows()
    return reporte1( t2,tiempo,movi2)

```

```
#DETECCION DECUBITO SUPINO
```

```
def captura ():
```

```
    inicio=time.time()
    s=0
    msj=0
    movi=0
    movi2=0
    size=2
    ti=1
    t2=0
    t4=0
    #def draw_circle():
    face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_alt.xml')
    cap = cv2.VideoCapture(0)
    cap.set(3,400)
    cap.set(4,600)
    _,prev=cap.read ()
    prevG=cv2.cvtColor(prev,cv2.COLOR_RGB2GRAY)
    kernel=np.ones((5,5),np.uint8)
    font=cv2.FONT_HERSHEY_SIMPLEX
    PrevCen=np.array( [0,0])
    bot = telepot.Bot('481163786:AAHo20FR1yqjVfldSj7rhQ11w3G1EiVFwPT')
    ventana=" SEAPULP"

    ##cv2.ocl.setUseOpenCL(False)
    while 1:

        _,frame= cap.read()
        y=str(time.strftime("% A:%H:%M:%S"))
        tiempos=str(time.strftime("%M:%S"))
        formato = "%X"

        hora = time.strftime(formato) #variable tipo hora
        cv2.putText(frame, y,(5,270),font,0.5,(25,255,20))
        cv2.putText(frame, "PRESIONE ESC: PARA
SALIR",(5,20),font,0.4,(100,235,135),1)
        fGray=cv2.cvtColor(frame,cv2.COLOR_RGB2GRAY)
        mini = cv2.resize(frame, (frame.shape[1] / size, frame.shape[0] / size))
        faces = face_cascade.detectMultiScale(mini)
        # faces = face_cascade.detectMultiScale(fGray, 1.3, 5)
        flow=np.array(abs(np.array(fGray,np.float32)-
np.array(prevG,np.float32)),np.uint8)
        #cv2.imshow(ventana,frame)
        rang=cv2.inRange(flow,20,255)
```



```

#cv2.imshow('rang',flow)
opening=cv2.morphologyEx(rang, cv2.MORPH_OPEN,kernel)
closing=cv2.morphologyEx(opening, cv2.MORPH_CLOSE,kernel)
# cv2.imshow('CLOSING',closing)

M=[0,0]
nCaras=[0]
nMovi=[0]

nCaras[0]=len(faces)
def click_cerrar(event,x,y,flags,param):
    if event == cv2.EVENT_FLAG_LBUTTON:
        if x<50 and x<50 :

            cap.release()
            cv2.destroyAllWindows()
            return

cv2.setMouseCallback(ventana,click_cerrar)

if nCaras[0]!=0:

    contours,hierarchy=cv2.findContours(closing,1,2)
    nMovi[0]=len(contours)
    #print (" movimientoss",nMovi[0] )
    if nMovi[0]>=1:

        for cnt in contours:
            inicio2=time.time()

            #print (" movimientos",movi )

            x,y,w,h =cv2.boundingRect(cnt)
            if w>15 and h>15 and w<500 and h<500:
                movi2+=1
                movi+=1
                cv2.putText(frame, "MOVIMIENTO
DETECTADO!",(135,270),font,0.5,(0,0,255),1)
                cv2.rectangle(frame,(x,y),(x+w,y+h),(255,255,0),1)

                #print (" El paciente se movio",movi2 )
                if movi==6:

                    bot.sendMessage(519748192,"Hola, hay movimiento!!, Por
favor verifique")
                    #if movi==5:

```

```

        #pygame.mixer.music.load("movi1.wav")
        #pygame.mixer.music.play(1)

        if movi==30:

            movi=0

            if nMovi[0]==0:
                cv2.putText(frame, "PACIENTE
ESTABLE!",(150,270),font,0.5,(200,255,20))
                movi=0

            if nCaras[0]==0:

                msj+=1

                # print (" El paciente no esta en posicion Correcta",msj ) 518009346

                cv2.putText(frame, "ALERTA!!",(100,100),font,1.3,(0,0,255),2)

                if msj==4:
                    bot.sendMessage(519748192,"Hola, el paciente No esta en Posicion!!,
Por favor verifique")

                    n=1

                if msj==6:
                    n=n+3

                    t=n

                    t1=t
                    t2=(t1+t)/2

                    pygame.mixer.music.load("paciente2.wav")
                    pygame.mixer.music.play(1)

                    if msj==40:
                        msj=5

            else:
                msj=0

        for f in faces:

```

```

(x, y, w, h) = [v * size for v in f]
cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),thickness=4)
cv2.putText(frame, 'POSICION DE FRENTE DETECTADO',(x-20,y-
10),font,0.4,(25,255,20))
s=s+1

if(s>400):
    s=0
    bot.sendMessage(519748192,"ES NECESARIO CAMBIAR DE
POSICION AL PACIENTE, EL TIEMPO DE RUTINA A TERMINADO ")

prevG=fGray

cv2.imshow(ventana, frame)

key = cv2.waitKey(20) & 0xff
if key==27 :
    break
final=time.time()
tiempo=round( final-inicio,0)

cv2.destroyAllWindows()
cv2.VideoCapture(0).release()
return reporte1( t2,tiempo,movi2)

```

```
def reporte1(D,t,tm):
```

```

nD=str(D)
tT=str(t)
num=(tm/t)
porc=str((num*100)/t)

Nodete="Tiempo no detectado seg:_" +nD
TiempoT="Tiempo total de monitoreo seg:_" +tT
Movim="Porcentaje de movimientos %:_" +porc[:porc.find(".")]
y=str(time.strftime("%A:%H:%M:%S"))
Tf="Monitoreo Finalizado:_" +y
archivo1=open("registro.csv","w")
archivo1.write(Tf)
archivo1.write(";")
archivo1.write(Nodete)
archivo1.write(";")
archivo1.write(TiempoT)
archivo1.write(";")
archivo1.write(Movim)
archivo1.write(";")
archivo1.write("\n")
archivo1.close()

```

```

archivo=open("archivo11.csv","r")
for x in archivo:

    print "\n"
    print x
    print "\n"
    bot.sendMessage(519748192," Detalle de ultimo monitoreo del paciente:_ "+x+
"_Monitoreo Finalizado a:_ "+y+" _Tiempo promedio de paciente No detectado: seg:_ "
+nD+" _Tiempo total de monitoreo: seg:_ "+tT+ " _Porcentaje Movimiento Detectados:_
%_ "+porc[:porc.find(".")+2])
    print "REPORTE ENVIADO....."
    print "\n"
archivo.close()
return
def salir():
    result=tkMessageBox.askquestion("ADVERTENCIA", "Esta seguro de salir del
sistema SEAPULP?", icon='warning')
    if result=='yes':

        print "\n"
        print"HASTA LUEGO!!"
        print "\n"
        ventana.destroy()
        sys.exit(1)
    else:
        print "\n"
        print "TENGA UN EXCELENTE DIA "

def reporteonline():
    result=tkMessageBox.askquestion("REPORTE", "Esta seguro de enviar el ultimo
reporte de monitoreo?")
    if result=='yes':

        archivo=open("archivo11.csv","r")
        for x in archivo:
            print "\n"
            print x
            print "\n"
            bot.sendMessage(519748192,"Ultimo reporte de paciente: _"+x)

        archivo.close()
        archivo1=open("registro.csv","r")
        for t in archivo1:
            print "\n"
            print t
            print "\n"

```

```

        bot.sendMessage(519748192,"Datos Registrados: _"+t)
        print "REPORTE ENVIADO"
    archivo1.close()
else:
    print "\n"
    print "TENGA UN EXCELENTE DIA "

return
def reporteF():
    result=tkMessageBox.askquestion("REPORTE EXCLUSIVO", "Esta seguro de enviar
el ultimo reporte de monitoreo a un Familiar?")
    if result=='yes':
        botF = telepot.Bot('639265785:AAHsM461XxmIZK6rOB9G8Usq8hdEP19qCx0')
        archivo=open("archivo11.csv","r")
        for x in archivo:
            print "\n"
            print x
            print "\n"
            botF.sendMessage(595688583,"Ultimo reporte de paciente: _"+x)

        archivo.close()
        archivo1=open("registro.csv","r")
        for t in archivo1:
            print "\n"
            print t
            print "\n"
            botF.sendMessage(595688583,"Datos Registrados: _"+t)

        archivo1.close()
        print "REPORTE ENVIADO"
    else:
        print "\n"
        print "TENGA UN EXCELENTE DIA "

return

ventana=Tk()

ventana.geometry("650x370+0+0")
ventana.resizable(False, False)
ventana.config(bg="#0040FF")

fisei=PhotoImage(file="imagen3.gif")

lbl_fisei=Label(ventana,image=fisei).place(x=0,y=0)
#lbl_uta=Label(ventana,image=uta).place(x=500,y=0)
etiqueta=Label(text="SEAPULP
",fg="#FFFF00",bg="#0B243B",font=("Impact",30)).place(x=200,y=0)

```

SYSTEM

```

boton1=Button(text="INGRESAR A
TERAPIA",activebackground="green",fg="white",bg="#086A87",command=ingreso,an
chor="center",bd=5,relief="raised",font=("verdana",10)).place(x=50,y=265)
boton2=Button(text="REGISTRAR
PACIENTE",activebackground="green",fg="white",bg="#086A87",command=Registro
,anchor="center",bd=5,font=("verdana",10)).place(x=50,y=230)
boton3=Button(text="MOSTRAR PACIENTES
REGISTRADOS",activebackground="green",fg="white",bg="#086A87",command=M
ostrar,anchor="center",bd=5,font=("verdana",10)).place(x=50,y=300)
boton4=Button(text="ELIMINAR",activebackground="green",fg="white",bg="#086A8
7",command=eliminar,anchor="center",bd=5,font=("verdana",10)).place(x=50,y=335)
boton5=Button(text="INICIAR MONITOREO
FRONTAL",activebackground="green",fg="white",bg="#086A87",command=captura,a
nchor="center",bd=5,font=("verdana",10)).place(x=380,y=195)
boton11=Button(text="INICIAR MONITOREO
LATERAL",activebackground="green",fg="white",bg="#086A87",command=capturaP
,anchor="center",bd=5,font=("verdana",10)).place(x=380,y=230)
boton6=Button(text="GENERAR ULTIMO
REPORTE",activebackground="green",fg="white",bg="#086A87",command=reporteo
nline,anchor="center",bd=5,font=("verdana",10)).place(x=380,y=265)
boton7=Button(text="ENVIAR REPORTE
FAMILIAR",activebackground="green",fg="white",bg="#086A87",command=reporteF
,anchor="center",bd=5,font=("verdana",10)).place(x=380,y=300)
boton8=Button(text="SALIR DEL
SISTEMA",activebackground="green",fg="white",bg="#086A87",command=salir,anch
or="center",bd=5,font=("verdana",10)).place(x=380,y=335)
boton9=Button(text="INFORMACION",activebackground="green",fg="white",bg="#0
88A68",command=informacion,anchor="center",bd=5,font=("verdana",10)).place(x=50
,y=195)

ventana.mainloop()

```