

CUANTIFICACIÓN DEL NIVEL DE ESTRÉS METABOLICO A PARTIR DEL USO DE TERMOGRAFÍA INFRARROJA Y PROCESAMIENTO DE IMÁGENES.

MARIAJOSE DE JESUS VERA GALLARDO

Trabajo para optar al Título de
Ingeniero Civil Biomédico

Profesor Guía:

JULIO SOTELO PARRAGUEZ

Julio - 2022

Valparaíso - Chile

Resumen

Resumen:

Objetivo: Detectar variaciones de temperaturas asociadas al estrés metabólico. **Metodología:** Se sometieron a 19 voluntarios a pruebas estresantes mientras se realizaban registros termográficos y fisiológicos. Las imágenes obtenidas fueron analizadas utilizando un código Python con la finalidad de detectar cambios de temperatura durante las pruebas estresantes y evaluar que zona del rostro registró mayor variación de temperatura asociada al estrés. **Resultados:** Se registraron cambios significativos de temperatura en las pruebas que requerían mayor concentración y se detectaron como áreas de interés la nariz, ojos y boca. **Conclusión:** Es posible considerar la termografía infrarroja como una herramienta para la detección del estrés metabólico, sin embargo, es relevante evaluar una forma para disminuir los factores que alteran los resultados.

Palabras Clave (máximo 5): Termografía, detección del estrés, estrés metabólico.

TABLA DE CONTENIDO

1.	INTRODUCCION.....	1
1.1	Objetivo general.....	2
1.2	Objetivos específicos.....	2
2.	MARCO TEORICO.....	2
2.1	Termografía infrarroja.....	2
2.1.1	Aplicaciones médicas.....	3
3.	ESTADO DEL ARTE.....	3
4.	METODOLOGIA.....	4
4.1	Participantes.....	4
4.2	Materiales.....	5
4.3	Pruebas mentales.....	6
4.4	Procesamiento de imágenes.....	7
4.4.1	Extracción de frames.....	8
4.4.2	Alinear imágenes.....	8
4.4.3	ROI.....	8
4.4.4	Limpieza de datos.....	8
4.4.5	Análisis estadístico.....	9
4.4.6	Umbralización:.....	10
4.5	Parámetros fisiológicos.....	11
5.	RESULTADOS.....	11
6.	DISCUSION.....	13
7.	CONCLUSION.....	14
8.	REFERENCIAS.....	14
9.	ANEXOS.....	15
9.1	Resolución.....	15
9.2	Pruebas Mentales.....	15
9.2.1	Prueba mental 1 - Encuesta:.....	16
9.2.2	Prueba mental 2 – Juego de memoria:.....	17
9.2.3	Prueba mental 3 – Juego de palabras:.....	17
9.2.4	Prueba mental 4 - PASAT :.....	18
9.2.5	Encuesta de percepción:.....	18
9.3	Códigos Python.....	19
9.3.1	Código Prueba PASAT.....	19
9.3.2	Código para extraer frames:.....	19
9.3.3	Código para alinear imágenes:.....	21

9.3.4	Código para extraer ROI:	22
9.3.5	Código para análisis estadístico:	22
9.3.6	Gráfico de cajas:	24
9.3.7	Código para Mapeo:	43
9.3.8	Código de Umbralización:.....	43

CUANTIFICACION DEL NIVEL DEL ESTRÉS METABOICO A PARTIR DEL USO DE TERMOGRAFIA Y PROCESAMIENTO DE IMÁGENES.

Mariajosé de Jesús Vera Gallardo.

Escuela de Ingeniería Civil Biomédica

Facultad de Ingeniería, Universidad de Valparaíso, Chile

Palabras clave: Termografía, detección del estrés, estrés metabólico.

1. INTRODUCCION.

El estrés es un proceso subjetivo que comienza cuando un sujeto debe adaptarse a una situación que considera amenazante o problemática y necesita dar una respuesta de forma rápida, generando que el sujeto realice un esfuerzo físico y mental agotador [1]. Este proceso está conformado principalmente por tres etapas: Fase de alarma, fase de resistencia y fase de agotamiento [2]. La fase de alarma corresponde al momento en el cual el sujeto percibe una situación estresante y el organismo se prepara para enfrentarla [2]. La fase de resistencia comienza cuando el sujeto intenta resolver la situación con la finalidad de alejarse de la situación estresante, si el sujeto logra esto, el estrés no se concreta, sin embargo, de igual manera se produce una fatiga mental y física [2]. Finalmente, si el sujeto no logra resolver el problema de forma adecuada y la situación se mantiene por un tiempo prolongado, se produce la fase de agotamiento, lo cual puede causar graves daños en la salud de la persona, por ejemplo: trastornos físicos, psicológicos y conductuales [2]. Dado que el estrés es subjetivo, no es posible determinar una única causa para su desarrollo, no obstante, se ha logrado identificar factores que pueden generar las diversas situaciones de estrés, algunas pueden ser causadas por factores externos al sujeto, ya sean ambientales o químicos, otras se producen debido a factores sociales en los cuales puede estar involucrado personas del entorno del sujeto o incluso pueden estar relacionadas a como el sujeto piensa, siente y afronta la vida [3].

Actualmente, el estrés es uno de los problemas de salud mental más importantes del siglo XXI, afectando a un 54% de los estudiantes de nuestro país, según la primera encuesta nacional de salud mental universitaria, realizada el 2019, además, se estima un aumento de esta cifra debido al aislamiento y temor al contagio generado durante la pandemia [4]. Esto ha generado una gran preocupación con respecto a la salud mental tanto a nivel nacional, como a nivel mundial, lo que ha potenciado el interés en realizar estudios con el fin de detectar de forma temprana el estrés y así reducir los efectos adversos causados como el sobrepeso, aumento en consumo de alcohol y drogas, problemas cardiovasculares y problemas de ira [5]. Los diversos estudios han logrado demostrar que es posible analizar el estrés por medio de 3 factores: la epidemiología, la psicología y la biología, permitiendo aumentar las posibilidades de cuantificar el nivel de estrés en las personas, tanto para evaluaciones clínicas como para el área de investigación [1]. A nivel epidemiológico y psicológico se utilizan pruebas de estrés percibido por los sujetos en situaciones cotidianas de su rutina; a nivel fisiológico se han evaluado los efectos que tiene el estrés en el cuerpo humano, logrando detectar cambios en ritmo cardiaco, frecuencia respiratoria, presión arterial, insulina y tensión muscular, asociados a un proceso metabólico generado por el cuerpo humano para otorgarle al sujeto, la energía necesaria para resolver la situación [1].

Dado lo anterior, el siguiente trabajo tiene como objetivo detectar cambios de temperatura asociado al proceso metabólico generado por el cuerpo durante situaciones de estrés, en voluntarios pertenecientes a la escuela de Ingeniería Civil Biomédica de la Universidad de Valparaíso durante el primer semestre del año 2022, evaluando su respuesta metabólica mientras son sometidos a 4 pruebas estresante y se relacionan los resultados obtenidos con su carga académica. Se utilizó una cámara térmica Seek-Thermal-Shotpro de 640x480 pixeles. Como parámetros complementarios se realizaron mediciones de presión arterial, frecuencia cardíaca y oximetría en sangre.

1.1 OBJETIVO GENERAL.

Medir los niveles de estrés en voluntarios a través de termografía infrarroja, mientras los sujetos realizan pruebas cognitivas.

1.2 OBJETIVOS ESPECIFICOS.

- Reclutar voluntarios según criterios de inclusión y exclusión.
- Someter a los voluntarios a pruebas cognitivas mientras se registran los parámetros fisiológicos definidos.
- Procesar las imágenes térmicas adquiridas y detectar patrones térmicos relacionados con el estrés.
- Correlacionar resultados obtenidos con carga académica de los voluntarios.

2. MARCO TEORICO.

Para llevar a cabo este estudio, es importante entender que es la termografía infrarroja y que consideraciones se deben tener al momento de utilizarla en aplicaciones médicas.

2.1 TERMOGRAFIA INFRARROJA.

Las cámaras infrarrojas fueron desarrolladas en la década de 1950 con fines militares, que permite visualizar y registrar la temperatura de un objeto, sin necesidad de contacto y luz visible. En los últimos 50 años, su uso se expandió a áreas de la ingeniería como herramienta de apoyo en 4 principales áreas: mantenimiento predictivo, monitoreo de condiciones, diagnóstico de problemas y el área de investigación y desarrollo. Las cámaras infrarrojas son muy similares a las cámaras de video digital, teniendo como principal diferencia, que los dispositivos térmicos poseen un lente que enfoca la radiación infrarroja hacia un detector compuesto por materiales sensibles a las longitudes de ondas IR (aproximadamente entre $0,9 \mu m$ a $14 \mu m$). La termografía infrarroja, tiene como objetivo principal, convertir la radiación infrarroja que no es visible para el ojo humano, en una representación visual de las variaciones térmicas de un objeto u escena, siendo la radiación emitida directamente proporcional con la temperatura. Cabe destacar, que todo objeto con temperatura mayor a $0^{\circ}K$ emite radiación infrarroja [6].

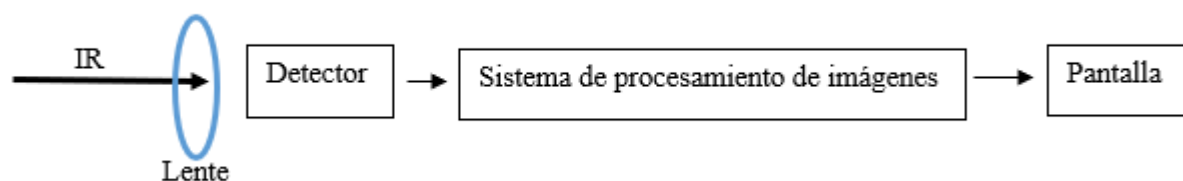


Figura 1: Diagrama en bloque de componentes básicos de sistema de termografía infrarroja.

2.1.1 Aplicaciones médicas.

Se ha estudiado el uso de la termografía para el área de la medicina desde principio de la década de 1960, sin embargo, recién en 1978 se publicó un artículo en el cual tenía como enfoque la estandarización de la termografía para aplicaciones patológicas y ensayos clínicos de fármacos, estos artículos fueron creados por la Asociación Europea de la Termografía [7].

La aplicación de la termografía en el estudio del cuerpo humano es posible debido a que las funciones mecánicas, físicas y bioquímicas de nuestro cuerpo involucran calor, el cual está regulado por nuestra piel mediante el uso de tres mecanismos de transferencia: conducción, convección y radiación. Estos mecanismos se ven evidenciados en nuestros procesos fisiológicos como, por ejemplo: circulación sanguínea, sudoración y generación de calor metabólico [6].

Según diversos estudios, las alteraciones en el sistema termorregulador del cuerpo humano pueden manifestarse principalmente en dos modalidades: cambios en la distribución espacial de la temperatura y cambios durante el calentamiento o enfriamientos de un área de la piel [6]. Es importante comprender que se denomina alteración a una diferencia de temperatura entre los 0,7° a 1°C [6]. A continuación, se describirán factores que se deben considerar al momento de adquirir imágenes con valor diagnóstico clínico utilizando termografía infrarroja:

- Entorno:
La habitación que se utilice para termografía debe tener un tamaño adecuado que permita una distancia prudente entre el sujeto de prueba, el examinador y el sistema de adquisición de imagen, idealmente se sugiere utilizar un espacio con tamaño mínimo de 2 x 3 metros [7]. El lugar debe tener buena ventilación y mantener una temperatura constante que sea cómoda para el sujeto [7].
- Paciente:
Es relevante considerar factores que puedan afectar la temperatura corporal percibida por el dispositivo de detección térmica, como el uso de cosméticos, ingesta de alcohol o tabaquismo, además, es relevante considerar el historial médico del paciente con el fin de asegurar la calidad de los resultados [7].
- Sistema de imagen:
Se recomienda el uso de trípodes para mantener un ajuste de altura y ángulo con respecto al paciente constante y reproducible [7]. Se recomienda el uso de software como sistema de apoyo para la detección de patologías [7].
- Reporte:
Para evitar errores de interpretación de los resultados y posibles comparaciones con imágenes anteriores, se recomienda estandarizar la escala de colores a utilizar [7].

3. ESTADO DEL ARTE.

Debido a la importancia que se tiene para este trabajo asegurar la calidad de los resultados obtenidos, se realizó una búsqueda sistemática con el fin de investigar la metodología y procesos utilizados para la obtención de resultados en otros estudios de aplicación médica de termografía para la detección de estrés en humanos. En la tabla 1 se puede apreciar el resumen de la información obtenida de los artículos seleccionados desde la base de datos PubMed utilizando las palabras clave: *Thermography, Stress detection*.

Autor	Año	Título	Metodología	Procesamiento de imagen
Robert Koprowski [8]	2017	Dedicated tool to assess the impact of a rhetorical task on human bodytemperature	Cuestionario para recopilar información básica (sexo, edad). Tiempo de aclimatación 20 minutos. Se definen criterios de exclusión.	Se utilizan herramientas de Matlab para procesamiento de imágenes. Se aplica un filtro mediano. Se ajusta la posición del sujeto en la imagen. Se comparan cambios de temperatura.
Katarzyna Baran [9]	2021	Stress detection and monitoring base don low-cost mobile thermography	Se recopilaron datos de los participantes previo a las mediciones. Se considera un tiempo de aclimatación. Temperatura y humedad ambiental constante. Distancia de cámara a 50 cm de la cara.	Se realizó procesamiento de imágenes utilizando la librería Open CV: Conversión de BGR a HSV Enmascaramiento Transformaciones morfológicas
Youngjun Cho [10]	2019	Instant Stress: Detection of Perceived Mental Stress Through Smartphone Photoplethysmography and Thermal Imaging	Se definieron criterios de exclusión. Se recopilaron datos de los participantes. Se realizaron mediciones en la punta de la nariz de los participantes.	Se utilizaron tres procesos para la obtención de resultados: Seguimiento de región de interés. Aplicación de filtro pasa bajo. Calcular diferencias de temperaturas.

Tabla 1: Resumen búsqueda sistemática.

4. METODOLOGIA.

El estudio fue realizado en la Facultad de Ingeniería de la Universidad de Valparaíso, durante el primer semestre del año 2022. Esta investigación fue evaluada y aprobada por el comité de ética oficial de la Universidad, resolución adjunta en anexos. Se establecieron criterios de inclusión y exclusión para reclutar voluntarios. Previo a iniciar el estudio se solicitó a los voluntarios leer y firmar el consentimiento informado. La investigación consistió en realizar dos mediciones, la primera con la finalidad de obtener parámetros basales y la segunda medición se realizó al transcurrir un mes para obtener resultados comparativos. Durante las mediciones se realizaron registros de presión arterial, frecuencia cardiaca, oxigenación en sangre y adquisición de imágenes termográficas. Las imágenes obtenidas fueron analizadas utilizando procesamiento de imágenes por medio de código Python.

4.1 PARTICIPANTES.

Durante este estudio, inicialmente participaron 20 estudiantes de la escuela de Ingeniería Civil Biomédica de la Universidad de Valparaíso, sin embargo, previo a las segundas mediciones, uno de los participantes anunció que no podría seguir en el estudio. Finalmente participaron 19 voluntarios de los cuales 13 fueron de sexo femenino y 6 de sexo masculino, con edad promedio de 24 ± 3 años.

Los voluntarios fueron seleccionados según el siguiente criterio de inclusión: Estudiantes en calidad de alumno regular de Ingeniería Civil Biomédica durante el primer semestre del año 2022. Fueron excluido aquellos voluntarios con: lesión o infección en el área del rostro, voluntarios con problemas de visión que no sean capaces de detectar objetos a una distancia de 1 metro sin lentes, voluntarios que consuman medicamentos de tipo ansiolíticos, voluntarios en estado de ebriedad o bajo el efecto de drogas, voluntarios con enfermedades que afecten presión arterial, frecuencia cardiaca y/o oxigenación en sangre.

Antes de realizar las mediciones se consideró un tiempo de aclimatación de 20 minutos, durante ese tiempo se le solicitó a los participantes quitarse la mascarilla, se posiciono la cámara a 1 metro de distancia del sujeto, se ajustó a la altura de este para enfocar el área del rostro y permaneció encendida con la finalidad que el detector se ajustara a la temperatura ambiente y del sujeto de prueba. Las pruebas fueron realizadas en una sala ubicada en la facultad de ingeniería de la Universidad de Valparaíso, donde se colocó al participante fuera del alcance de corrientes de aire y exposición a rayos UV. La sesión completa de medición duró aproximadamente 1 hora, durante ese tiempo se les solicitó a los participantes evitar movimientos innecesarios y responder mirando siempre a la cámara.

En la Tabla 2 se encuentra el registro de los voluntarios que participaron en el estudio, en el cual se encuentra el ID correspondiente a cada participante, la edad, genero, la generación de ingreso a la cual pertenece y carga académica.

ID	Edad	Genero	Generación	Carga académica
P_01	24	Femenino	2017	22 créditos
P_02	26	Femenino	2015	22 créditos
P_03	25	Femenino	2015	31 créditos
P_04	22	Femenino	2018	35 créditos
P_05	26	Femenino	2015	22 créditos
P_06	18	Masculino	2022	31 créditos
P_07	18	Masculino	2022	31 créditos
P_08	22	Femenino	2019	32 créditos
P_09	25	Femenino	2015	22 créditos
P_10	26	Femenino	2017	30 créditos
P_11	23	Femenino	2017	32 créditos
P_12	25	Femenino	2015	26 créditos
P_13	24	Femenino	2017	22 créditos
P_14	25	Masculino	2015	26 créditos
P_15	24	Femenino	2017	25 créditos
P_16	30	Masculino	2019	31 créditos
P_17	26	Masculino	2017	26 créditos
P_18	24	Masculino	2015	21 créditos
P_19	21	Femenino	2019	28 créditos

Tabla 2: Datos de voluntarios.

4.2 MATERIALES.

Para la obtención de imágenes de termografía se utilizó una cámara térmica, Seek-Thermal-Shotpro, este sistema de detección infrarrojo tiene un sensor térmico de 320x240 pixeles, resolución de pantalla de 640x480 pixeles, sensibilidad térmica: >70 mK y frame rate de 9 Hz. La cámara se ubicó a 1 m del voluntario enfocada en el área del rostro con un ángulo de 90°.

Para registrar la frecuencia cardiaca y oxigenación en sangre, se utilizó un pulsioxímetro Burer P30, el cual posee un rango de medición de 0-100 % \pm 2 % y 30-250 lpm \pm 2lpm para saturación de oxígeno y frecuencia cardiaca respectivamente. Con sensores de medición de SpO2 de luz roja (660 nm); infrarrojos (880 nm) y un diodo receptor de silicio.

Para registrar la presión arterial se utilizó un tensiómetro Burer BM35, de tipo oscilométrico, con una precisión de \pm 3 mmHg y rango de medición de 60-280 mmHg y 30-200 mmHg para sistólica y diastólica respectivamente.

Para el procesamiento de imágenes se utilizó el programa visual Studio para desarrollar un código Python que permitiera extraer frames de los videos y analizar las imágenes adquiridas.

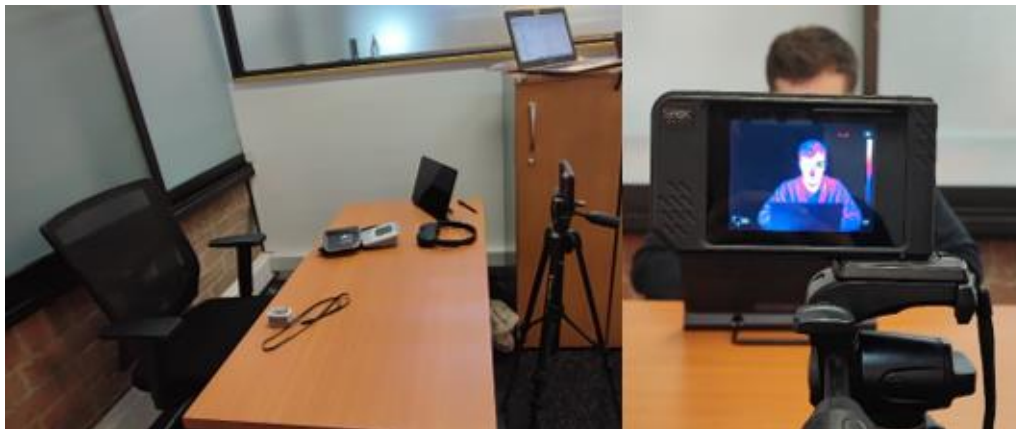


Figura 2: Implementación de materiales en la sala de medición y demostración de imagen captada por cámara térmica.

4.3 PRUEBAS MENTALES.

Para someter a los participantes a una situación de estrés, se realizaron 4 pruebas mentales de dificultad ascendente, además se realizó una encuesta de percepción del estrés al finalizar las mediciones. En los Anexos podrá visualizar las planillas de cada una de las pruebas y de la encuesta de percepción.

- Prueba mental 1 – Encuesta:
Esta prueba tiene como finalidad activar la primera fase del estrés, realizando preguntas fáciles de responder, por ejemplo: estación del año, lugar donde se encuentra la persona, etc. Finalizando con una pregunta de memoria, donde el sujeto deberá recordar tres palabras, las cuales deberá repetir luego de responder algunas preguntas más.
- Prueba mental 2 – Juego de memoria:
Esta prueba se realizó mediante un juego en un dispositivo móvil llamado “Test de Memoria Visuoespacial” descargado en Play Store. Este juego, tiene como finalidad que los participantes recuerden y repitan una secuencia de movimientos siguiendo las indicaciones del juego. Posteriormente se registró la cantidad de aciertos y la cantidad de intentos fallidos.
- Prueba mental 3 – Juego de palabras:
Durante esta prueba el participante deberá crear una secuencia de palabras, inicialmente la secuencia constara de una palabra y de irán añadiendo de a una hasta llegar a 30. En la figura 4 se puede apreciar las palabras que serán dictadas, con las respuestas esperadas y el registro de las secuencias que fueron completadas de forma exitosa.
- Prueba mental 4 – PASAT:

Para esta prueba se desarrolló un código Python para dictar de forma aleatoria 100 números entre el 1 y el 9, además crea un archivo CSV con los números aleatorios dictados y las respuestas correctas. El test PASAT es una prueba cognitiva aritmética, donde el participante debe memorizar el número mencionado y sumarlo con el número que se le dijo anteriormente y decir el valor en voz alta, es decir si el número mencionado fue n y el número anterior corresponde a $(n - 1)$ debe sumar $n + (n - 1)$. Para registrar los resultados del participante se extrajeron los datos del archivo CSV creado en un Excel y se fueron colocando '1' en las filas donde el participante resolvió la suma correctamente.

- Encuesta de percepción:
Esta prueba tiene como objetivo tener respuestas subjetivas sobre como el participante se siente en una escala del 1 al 7 con respecto a la carga académica que tiene en ese momento y sobre las pruebas mentales que acaba de realizar.

4.4 PROCESAMIENTO DE IMÁGENES.

Con la finalidad de obtener los resultados esperados, se realizaron diferentes procesos para realizar un análisis de las imágenes obtenidas, utilizando diversas técnicas de procesamiento de imágenes. Para esta investigación se utilizaron dos metodologías diferentes para la obtención de resultados, en la figura 3 se puede apreciar el flujo realizado para analizar las imágenes térmicas, a continuación, se describirán cada uno de ellos. Cada uno de los procesos fue realizado mediante un código Python, los cuales se encuentran en la sección de Anexos:

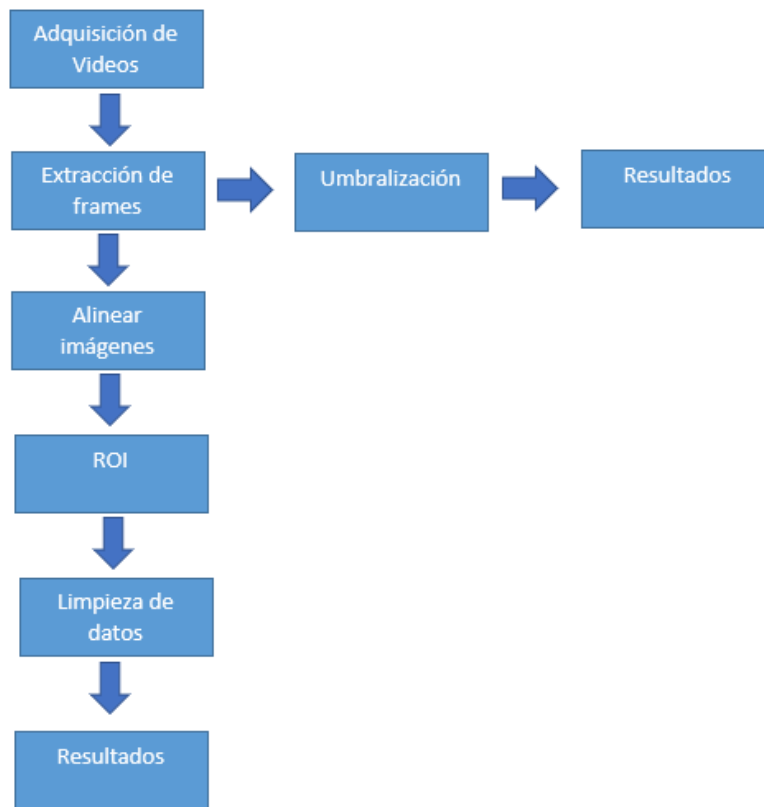


Figura 3: Flujo de procesamiento de imágenes.

4.4.1 Extracción de frames.

Debido que durante las pruebas se realizaron registros en formato de video, fue necesario extraer los frames que componen dichos registros, dado que el dispositivo utilizado registraba 9 imágenes por segundos, se optó por realizar un muestreo cada 2 segundos, es decir, cada 18 imágenes, con la finalidad de optimizar los procesos siguientes debido a que la temperatura humana no tiene variaciones significativas en un tiempo tan reducido en un entorno con temperatura controlada. Las imágenes extraídas fueron almacenadas en una carpeta para su posterior análisis.

4.4.2 Alinear imágenes.

Si bien, se les solicitó a los participantes realizar movimientos bruscos durante las pruebas realizadas, para el ser humano es muy complicado mantenerse quieto durante largos periodos de tiempo, por ende, al analizar los frames se detectaron movimientos que podrían afectar el análisis de las imágenes. Dado lo anterior, se aplicó una técnica conocida como registro de imágenes basado en optical flow, utilizando la librería skimage, esto permitió alinear las imágenes usando una de ellas como referencia, en la siguiente figura se aprecia un ejemplo de registro de imágenes aplicado en las imágenes obtenidas:



Figura 4: La siguiente secuencia de imagen muestra el resultado de registro de imagen, al alinear una imagen movida con respecto a una imagen de referencia que se encuentra centrada.

4.4.3 ROI.

Dado que las imágenes adquiridas poseen mas datos de los que deseamos analizar, se realizó un recorte de nuestra área de interés, es decir, del rostro de los participantes. El proceso de alineamiento realizado anteriormente permitió extraer la misma sección para todas las imágenes.



Figura 5: área de interés para analizar.

4.4.4 Limpieza de datos.

Previo a realizar el análisis estadístico, fue necesario revisar la calidad de las imágenes que se utilizaron para el análisis, con el fin de asegurar la calidad de los resultados. Al realizar esta revisión se detectaron

algunas imágenes que no pudieron ser reconstruidas de forma correcta al momento de realizar la alineación por ende fue necesario descartarlas.



Figura 6: Ejemplo de imagen alterada.

4.4.5 Análisis estadístico.

Finalmente, tras almacenar las áreas de interés de cada prueba realizada por los participantes, se realizó una conversión de intensidad de pixel a temperatura en grados Celsius para calcular el promedio de temperatura de cada ROI en función del tiempo y así determinar si durante las pruebas fue posible detectar variaciones térmicas que se puedan asociar al estrés. Además, utilizando los datos obtenidos de la varianza, se buscó obtener el patrón térmico, identificando las zonas del rostro que registraron mayor variación de temperatura. En la Figura 7 se aprecian los gráficos obtenidos de promedio y varianza de uno de los participantes. En la Figura 8, se puede ver las zonas con mayor variación de intensidad de pixel.

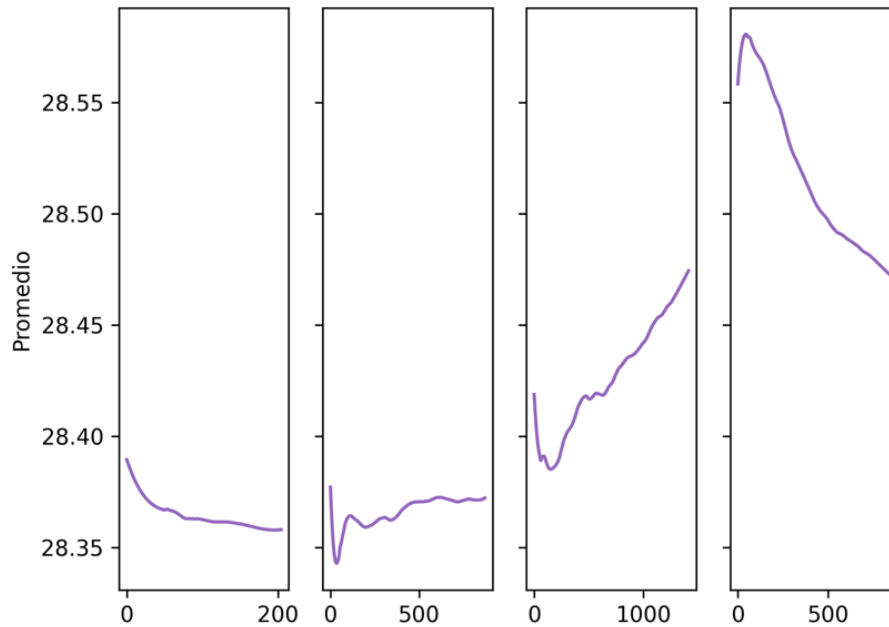


Figura 7: Ejemplo de datos obtenidos durante las cuatro pruebas en función del tiempo.



Figura 8: Zonas con mayor variación de temperatura durante las pruebas estresantes.

4.4.6 Umbralización:

Debido a que las zonas que presentan mayor variación de temperatura son muy pequeñas, no es posible, determinar cuantos grados de temperatura varía esta zona utilizando el promedio y varianza del rostro, es por esto, que se utilizó esta técnica para extraer los diversos colores que componen la escala de temperatura y evaluar los cambios producidos. Cada rango de color extraído representa una variación de 2 °C.



Figura 9: Umbralización de imágenes en tres tiempos distintos.

4.5 PARAMETRÓS FISIOLÓGICOS.

Como se mencionó anteriormente además de los registros térmicos se obtuvieron datos fisiológicos en 3 instancias diferentes durante la sesión de medición, el primer registro se realizó al finalizar el tiempo de aclimatación, el segundo registro se realizó al finalizar la segunda prueba estresante y el tercero se realizó al finalizar la cuarta prueba. La finalidad de estos datos fue realizar una correlación entre los datos térmicos obtenidos con las variaciones en los parámetros fisiológicos de presión arterial, frecuencia cardiaca y oximetría.

5. RESULTADOS.

Los resultados obtenidos usando estadística descriptiva de las ROI, muestran cambios de temperatura en la zona del rostro asociado al estrés producido por las diversas pruebas, sin embargo, dado que el estrés es subjetivo, las pruebas causaron diferentes efecto en los participantes, debido a esto, se detectó una alteración en la distribución de los datos, por este motivo, se tomó la decisión de agrupar los datos en un grafico de caja para analizar los promedios de temperatura en conjunto y así evaluar la tendencia de cambio de temperatura. Por otro lado, la pendiente en la gráfica de los promedios en función del tiempo nos permitió evaluar que pruebas causaron mayor variación en los participantes, ya que, a mayor pendiente, sea esta positiva o negativa significa un mayor cambio de temperatura, estos datos se recopilaron en una tabla y se calculó el porcentaje de participantes que presentaron pendientes elevadas en cada prueba.

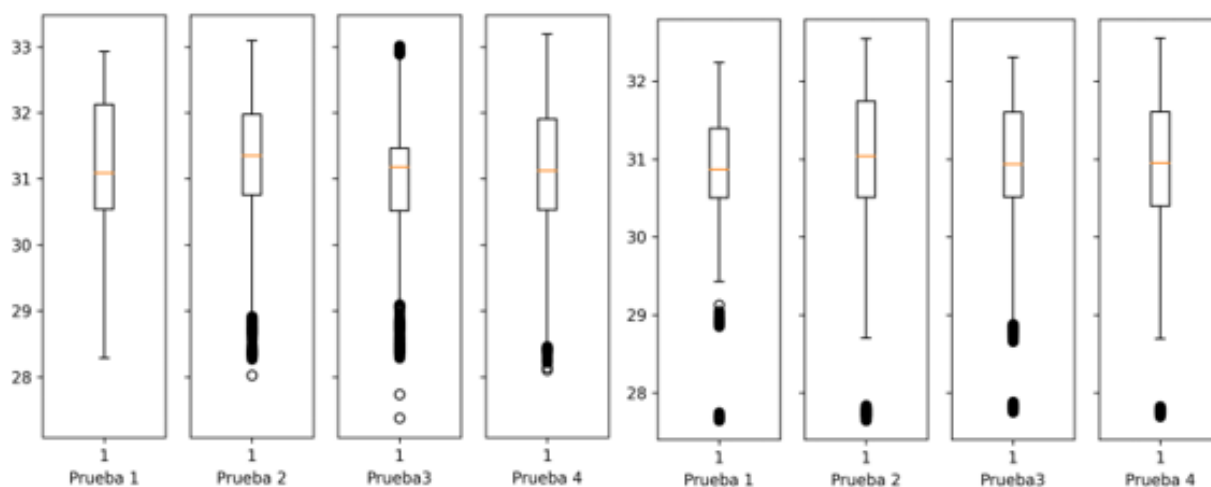


Figura 10: grafico de cajas durante medición 1 y 2.

Complementando los resultados anteriores, tras utilizar los datos de varianza para reconstruir una imagen y así evidenciar los puntos con mayor variación de temperatura, se logró identificar que el área con cambios significativos fue la nariz, también se detectaron cambios de menor intensidad en zona de ojos y boca. Debido a que las zonas que presentan cambios asociado con el estrés son muy pequeñas, no es posible visualizar cambios en los gráficos mostrados anteriormente. Dado lo anterior se decidió aplicar la técnica de umbralización en las imágenes extraídas de los videos, con la finalidad de evaluar como cambian las distribuciones de colores, es decir, de temperatura en el rostro. Utilizando esta técnica se pudo estimar que el promedio de variación en la zona de la nariz fue de 3°C y en ojos y boca el promedio fue de

1°C, en todos los casos los cambios fueron producidos por una disminución de temperatura. En la figura 11 se muestra las imágenes originales junto con el resultado obtenido de las umbralización

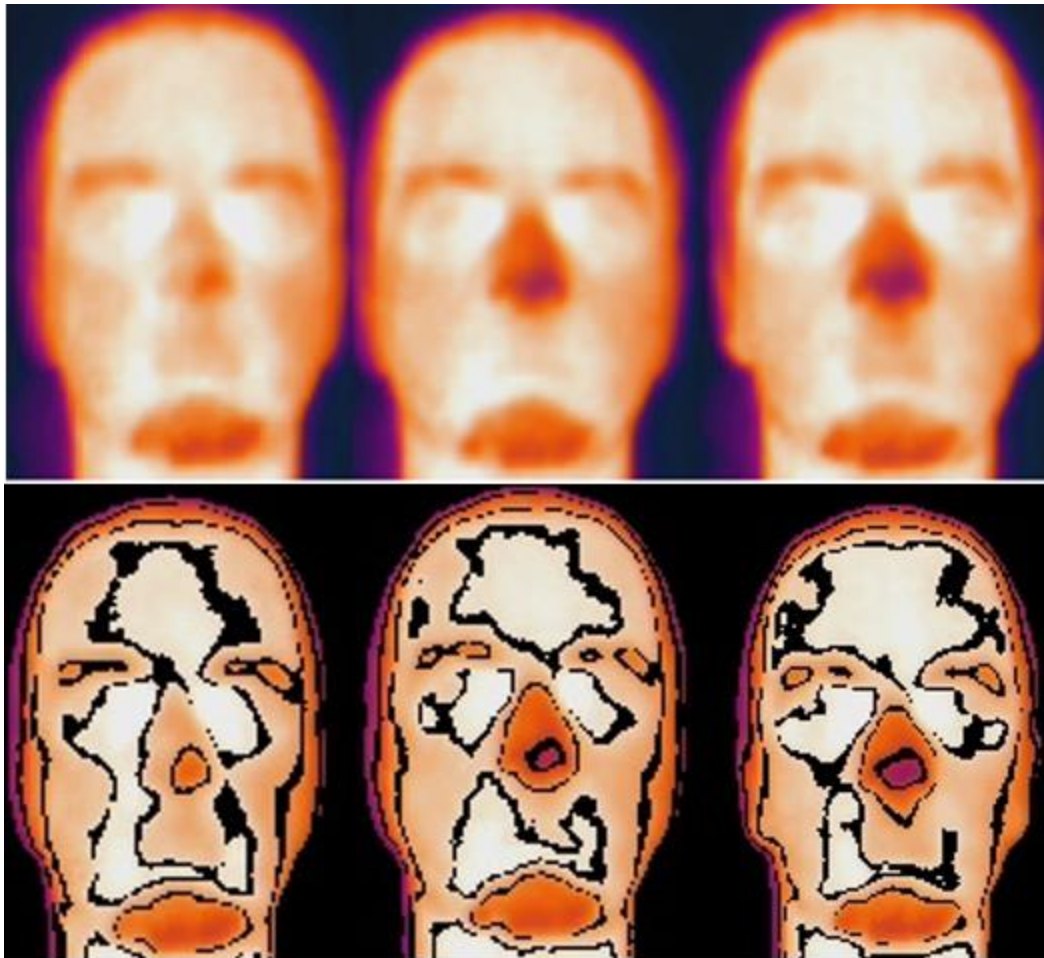


Figura 11: Imágenes originales e imágenes umbralizada.

Después de analizar los cambios de presión arterial y frecuencia cardíaca durante las pruebas estresantes se pudo determinar que los participantes que tuvieron cambios significativos de temperatura presentaron cambios presión arterial y frecuencia cardíaca, dado que el estrés genera alteraciones a corto plazo de estos parámetros, ninguno de los participantes presentó cambios de oxigenación en sangre.

Participante	Medición basal			Postprueba 2			Postprueba 4		
	Presion arterial	Frecuencia cardiaca	Oximetria	Presion arterial	Frecuencia cardiaca	Oximetria	Presion arterial	Frecuencia cardiaca	Oximetria
P_01	100/64	89	98	107/67	85	98	108/65	84	98
P_02	129/98	81	98	123/76	80	98	125/82	84	98
P_03	114/87	70	98	117/88	78	98	119/91	73	98
P_04	113/74	71	98	108/65	69	98	105/48	68	98
P_05	115/76	78	99	107/76	74	99	120/77	78	99
P_06	105/67	86	98	106/53	77	98	108/63	63	98
P_07	109/57	69	97	98/71	80	97	118/74	73	97
P_08	109/63	77	99	105/69	78	99	122/83	71	99
P_09	113/75	71	99	113/70	77	99	103/75	68	99
P_10	114/67	66	99	98/56	68	99	102/63	69	99
P_11	137/89	75	99	146/86	77	99	149/89	74	99
P_12	107/60	67	99	102/63	64	99	107/63	63	99
P_13	114/80	115	99	115/67	93	99	117/73	83	99
P_14	129/71	69	98	109/80	64	98	109/77	65	98
P_15	113/62	70	99	105/66	71	99	125/67	71	99
P_16	118/67	63	97	120/69	62	97	113/66	61	97
P_17	118/77	74	98	118/82	65	98	114/75	66	98
P_18	113/59	50	99	122/62	49	99	104/59	52	99
P_19	106/68	80	99	98/65	72	99	107/64	79	99

Ilustración 12: Variación de presión arterial, frecuencia cardiaca y oxigenación en sangre durante primera medición.

Dada la suspensión de evaluaciones a causa de la paralización estudiantil, los voluntarios no tuvieron cambios de percepción de su estrés académico durante las segundas mediciones. Por otro lado, la encuesta de percepción del estrés percibido durante las pruebas mentales no se pudo utilizar como parámetro comparativo con respecto a los resultados obtenidos.

6. DISCUSION.

Usualmente cuando nos referimos al estrés, solemos asociarlo a los síntomas causados por un estrés crónico, el cual es dañino para nuestra salud, sin embargo, como se ha mencionado anteriormente, el proceso del estrés comienza mucho antes, muchas veces sin que nos demos cuenta. Al analizar los datos obtenidos, fue posible determinar que es posible detectar cambios de temperatura, asociados al proceso metabólico causado por el estrés, generado mientras los participantes gastan energía en mantenerse concentrados para desarrollar las pruebas mentales, detectando cambios significativos durante las 3 ultimas pruebas, las cuales requerían mayor concentración para su desarrollo. Esto permitirá a futuro analizar desde otra perspectiva los cambios producidos en el cuerpo humano a causa del estrés, e incluso incorporar la temperatura corporal como parámetro para la detección temprana del estrés y seguimiento de este. Sin embargo, dado que la tasa de muestreo en términos de población en este estudio es muy baja, se recomienda replicar el estudio en un grupo mayor de personas para validar los resultados obtenidos, sin embargo, para aplicar este estudio en un grupo de muestreo mayor, se recomienda incorporar técnicas de inteligencia artificial para automatizar las técnicas de procesamiento de imágenes utilizadas.

Es importante considerar que, para el desarrollo de este estudio, no se consideraron factores hormonales ni de masa corporal, por ende, se propone evaluar cambios de la distribución de temperatura en situaciones estresantes comparando resultados obtenidos entre un grupo de control y un grupo con mayor masa corporal o mujeres durante la ovulación. Debido a que la participante P_11 presentó una alteración en el

parámetro de presión arterial durante las pruebas, tras investigar las posibles causas se señaló que podría ser un factor hormonal.

Una de las principales dificultades que se presentaron en este estudio es la distorsión de algunas imágenes causado por movimientos bruscos de los participantes, lo cual se vio reflejado en la gran cantidad de outliers en el gráfico de cajas. Dado el cese de actividades a causa de la paralización estudiantil, no fue posible determinar una relación entre los niveles de estrés detectados con la carga académica de los participantes. Se recomienda disminuir factores que puedan potenciar el movimiento de los voluntarios y explorar otras técnicas para alinear o estabilizar imágenes.

7. CONCLUSION.

Este estudio, ha permitido evaluar y demostrar cambios de temperatura en el rostro de los participantes, principalmente en nariz asociados al estrés, mediante el uso de termografía infrarroja, debido al proceso metabólico producido con la finalidad de generar energía suficiente para que los participantes pudieran mantenerse concentrados, y así, responder las pruebas a las cuales fueron sometidos y salir lo más pronto de esa situación estresante. Sin embargo, es importante disminuir los factores de movimiento, que causan alteraciones en los datos recopilados, con la finalidad de mejorar la calidad de los resultados obtenidos. Cabe mencionar que proceso de estrés generado en este estudio no fue dañino para los participantes.

8. REFERENCIAS.

- [1] S. Cohen, «A Stage Model of Stress and Disease,» *Perspect Psychol Sci.*, vol. 11, n° 4, p. 456–463, 2016.
- [2] C. B. G. Paltán., «DIAGNÓSTICO DEL NIVEL DE ESTRÉS Y SU INCIDENCIA EN EL DESEMPEÑO LABORAL, EN EL PERSONAL ADMINISTRATIVO DEL HOSPITAL REGIONAL “VICENTE CORRAL MOSCOSO “,» Cuenca, 2010.
- [3] A. O. Stepanian, «Síntomas, niveles de estrés y estrategias de afrontamiento en una muestra de estudiantes masculinos y femeninos de una institución de educación superior militar: Análisis comparativo.,» Bogotá, 2016.
- [4] INJUV, «Habemos de todo,» 2021. [En línea]. Available: <https://hablemosdetodo.injuv.gob.cl/wp-content/uploads/2021/02/Bullying-y-Cyberbullying.pdf>. [Último acceso: 2022].
- [5] A. J. N. CERMENATI, «DESARROLLO DE UN SISTEMA WEARABLE PARA LA DETECCIÓN DE ESTRÉS MEDIANTE APRENDIZAJE AUTOMÁTICO BASADO EN RITMO CARDIACO,» Concepción, 2021.
- [6] L. Izhar, «Chapter 2 - Thermal Imaging in Medicine,» de *Advances in Imaging and Electron Physics*, 2012, pp. 41-114.
- [7] E. Ring, «The Technique of Infra red Imaging in Medicine,» *Thermology international* , vol. 10, n° 1, pp. 7-14, 2000.
- [8] R. Koprowski, «Dedicated tool to assess the impact of a rhetorical task on human bodytemperature,» *International Journal of Psychophysiology*, n° 120, pp. 69-77, 2017.
- [9] K. Baran, «Stress detection and monitoring base don low-cost mobile thermography,» *Procedia Computer Science*, n° 192, p. 1102–1110, 2021.
- [10] Y. Cho, «Instant Stress: Detection of Perceived Mental Stress Through,» *JMIR Ment Health*, vol. 6, 2019.

9. ANEXOS.

9.1 RESOLUCIÓN.



ANT.: Revisión del proyecto "Cuantificación del nivel de estrés metabólico a partir del uso de termografía infrarroja y procesamiento de imágenes".

MAT.: Resultado de la revisión de la reformulación.

OFICIO N°002/CECFACING/2022

VALPARAISO, 8 de abril de 2022

A : JULIO SOTELO PARRAGUEZ
ACADÉMICO
ESCUELA DE INGENIERÍA CIVIL BIOMÉDICA,
VALPARAÍSO

DE : ANA AGUILERA FARACO
PRESIDENTA
COMITÉ DE ÉTICA CIENTÍFICA UNIVERSIDAD DE VALPARAÍSO

Por medio del presente, y de conformidad a lo indicado en el antecedente, sobre la presentación del trabajo de investigación de la estudiante María José Vera Gallardo para optar al título de Ingeniero Civil Biomédico, me permito informar lo siguiente.

Que, reunión ordinaria del Comité de Ética Científica de la Facultad de Ingeniería (cec-facing) de hoy viernes 8 de abril, los relatores designados presentaron los comentarios sobre la revisión de la reformulación del proyecto atendiendo a las recomendaciones realizadas por el cec-facing, que fueron enviadas el lunes 28 de marzo.

A partir de las aclaraciones y correcciones presentadas relacionadas con el resguardo de los individuos de estudio, el Comité ha considerado que las modificaciones responden a los requerimientos mencionados en la carta de recomendaciones, aprobándose de este modo el protocolo de investigación.

Sin otro en particular, saluda

ANA AGUILERA FARACO
PRESIDENTA
COMITÉ DE ÉTICA CIENTÍFICA CEC-FACING
UNIVERSIDAD DE VALPARAÍSO

Figura 13: Imagen de documento de probación del comité de ética

9.2 PRUEBAS MENTALES.

9.2.1 Prueba mental 1 - Encuesta:

A continuación, se realizarán preguntas enfocadas a la memoria. Por favor, intente responder lo mejor que pueda.		
¿Qué años es?		
¿Qué edad tienes?		
¿Con qué mano escribe normalmente?		
¿Qué estación del año es?		
¿Qué día de la semana es hoy?		
¿En que región del país estamos?		
¿En que piso del edificio estamos?		
A continuación, nombraré 3 objetos que deberá repetir y memorizar:	MESA	
	NARANJA	
	GLOBO	
Deberá recordarlos, ya que le pediré que los vuelva a repetir más adelante.		
Ahora le dire una palabra y cuando le indique le pediré que la deletree. LA palabra es HUEVO.	Hacia delante	Marcar con un 0 o un 1
	Hacia atrás	Marcar con un 0 o un 1
Repita la siguiente frase: "LA MANZANA DEL ARBOL"		
¿Podría repetir los tres objetos que le pedí que recordara anteriormente?		
LOGRADO	NO LOGRADO	

Figura 14: Planilla utilizada para realizar y registrar las respuestas correspondientes a la prueba 1

9.2.2 Prueba mental 2 – Juego de memoria:

Prueba 2 - Juego de memoria				
ID participante	Modo Directo		Modo inverso	
	Aciertos	Errores	Aciertos	Errores
P_01				
P_02				
P_03				
P_04				
P_05				
P_06				
P_07				
P_08				
P_09				
P_10				
P_11				
P_12				
P_13				
P_14				
P_15				
P_16				
P_17				
P_18				
P_19				
P_20				

Figura 15: Planilla utilizada para registrar resultado de prueba 2

9.2.3 Prueba mental 3 – Juego de palabras:

Secuencia Botella	Total=
Aciertos	Secuencia Botella Lapiz
Secuencia Botella Lapiz	Total=
Aciertos	Secuencia Botella Lapiz Silla
Secuencia Botella Lapiz Silla	Total=
Aciertos	Secuencia Botella Lapiz Silla Celular
Secuencia Botella Lapiz Silla Celular	Total=
Aciertos	Secuencia Botella Lapiz Silla Celular Reloj
Secuencia Botella Lapiz Silla Celular Reloj	Total=
Aciertos	Secuencia Botella Lapiz Silla Celular Reloj Cama
Secuencia Botella Lapiz Silla Celular Reloj Cama	Total=
Aciertos	Secuencia Botella Lapiz Silla Celular Reloj Cama Horno
Secuencia Botella Lapiz Silla Celular Reloj Cama Horno	Total=
Aciertos	Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto
Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto	Total=
Aciertos	Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared
Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared	Total=
Aciertos	Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma
Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma	Total=
Aciertos	Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma Crema
Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma Crema	Total=
Aciertos	Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma Crema Libro
Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma Crema Libro	Total=
Aciertos	Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma Crema Libro Perro
Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma Crema Libro Perro	Total=
Aciertos	Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma Crema Libro Perro Hilo
Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma Crema Libro Perro Hilo	Total=
Aciertos	Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma Crema Libro Perro Hilo Tierra
Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma Crema Libro Perro Hilo Tierra	Total=
Aciertos	Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma Crema Libro Perro Hilo Tierra Metal
Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma Crema Libro Perro Hilo Tierra Metal	Total=
Aciertos	Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma Crema Libro Perro Hilo Tierra Metal Lana
Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma Crema Libro Perro Hilo Tierra Metal Lana	Total=
Aciertos	Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma Crema Libro Perro Hilo Tierra Metal Lana Foto
Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma Crema Libro Perro Hilo Tierra Metal Lana Foto	Total=
Aciertos	Secuencia Botella Lapiz Silla Celular Reloj Cama Horno Auto Pared Pluma Crema Libro Perro Hilo Tierra Metal Lana Foto Palta

Figura 16: Fragmento de Planilla utilizada para registrar la respuesta de los participantes durante prueba 3

9.2.4 Prueba mental 4 - PASAT :

Column1	ID	num xi	num xi+1	suma	Aciertos
0	P_03	3	6	9	1
1	P_03	6	9	15	1
2	P_03	9	2	11	1
3	P_03	2	4	6	1
4	P_03	4	8	12	1
5	P_03	8	5	13	1
6	P_03	5	9	14	1
7	P_03	9	3	12	1
8	P_03	3	6	9	1
9	P_03	6	9	15	1
10	P_03	9	2	11	1
11	P_03	2	5	7	1
12	P_03	5	9	14	1
13	P_03	9	5	14	1
14	P_03	5	8	13	1
15	P_03	8	9	17	1
16	P_03	9	1	10	1
17	P_03	1	5	6	1
18	P_03	5	2	7	1
19	P_03	2	5	7	1
20	P_03	5	1	6	1

Figura 17: Fragmento del documento generado por el código Python para verificar las respuestas de los participantes durante la prueba 4.

9.2.5 Encuesta de percepción:

Encuesta Percepción								
ID	Pregunta	Respuesta						
P_01	¿Qué tan estresado te sientes con la carga académica?	1	2	3	4	5	6	7
	¿Qué tan estresante fueron las pruebas mentales?	1	2	3	4	5	6	7
P_02	¿Qué tan estresado te sientes con la carga académica?	1	2	3	4	5	6	7
	¿Qué tan estresante fueron las pruebas mentales?	1	2	3	4	5	6	7
P_03	¿Qué tan estresado te sientes con la carga académica?	1	2	3	4	5	6	7
	¿Qué tan estresante fueron las pruebas mentales?	1	2	3	4	5	6	7
P_04	¿Qué tan estresado te sientes con la carga académica?	1	2	3	4	5	6	7
	¿Qué tan estresante fueron las pruebas mentales?	1	2	3	4	5	6	7

Figura 18: Fragmento de planilla utilizada en encuesta de percepción

9.3 CÓDIGOS PYTHON.

9.3.1 Código Prueba PASAT

Prueba PASAT

```

from gtts import gTTS
import numpy as np
import os
from time import sleep
import pandas as pd

# guarda el audio de los numeros entre 1 y 9
mypath = 'NUMEROS/'
if not os.path.isdir(mypath):
    os.makedirs(mypath)

for n in range(9):
    myText = str(n+1)
    output = gTTS(text = myText, lang = "es", tld='com.mx', slow = False)
    output.save('NUMEROS/'+ myText + '.mp3')

ID= input("Ingresar ID participante:")
random_numbers = np.random.randint(1,10,100) # aqui se indica que se dictaran
100 numeros aleatorios entre 1 y 9
next_number_random = np.zeros(random_numbers.shape)
next_number_random[:-1] = random_numbers[1:]
sum_random = np.zeros(random_numbers.shape)
sum_random[:-1] = random_numbers[1:]+random_numbers[:-1]
print(random_numbers)
print(sum_random)
dict = {'ID': ID,
        'num xi': random_numbers,
        'num xi+1': next_number_random.astype(int),
        'suma': sum_random.astype(int)}
out = pd.DataFrame(dict)
out.to_csv("numeros.csv") #Se guardan los numeros y la suma en el archivo CSV

start= input('presione enter para continuar:')
# aca envio el audio por el audifono
cwd = os.getcwd()
for n in range(random_numbers.shape[0]):
    myText = str(random_numbers[n])
    os.system(cwd + '/NUMEROS/' + myText + ".mp3")
    sleep(4) # esta linea de comando nos da el tiempo de respuesta para que
mencione el numero siguiente

```

9.3.2 Código para extraer frames:

Extraer frames

```

#
import cv2
import os

```

```

Video_in='ingresar ruta de video' # ingresar nombre de video
video_out='ingresar ruta donde se guardará el nuevo video' # nombre de video
muestreado
carpeta_frames= 'ingresar ruta donde se guardarán los frames' #nombre de
carpeta con frames

#####
##### muestreo de video cada dos segundos #####
#####

captura= cv2.VideoCapture(Video_in) # cambiar nombre de video para cada
sujeto
fourcc = cv2.VideoWriter_fourcc(*'XVID')
ret, frame = captura.read() #primera captura del cuadro en la variable frame
fps_video=captura.get(cv2.CAP_PROP_FPS) #frames por segundo original del video
video_name
height,width = frame.shape[:2]
out = cv2.VideoWriter (video_out,fourcc, int(fps_video/9), (width,height))
count = 1

while ret:

    if count==1 or count%18==0: #Si el cuadro corresponde al primero o cada 5
cuadros, entonces guarda el cuadro/frame en el objeto 'out'

        frame      =      cv2.resize(frame,      None,      fx=1.0,      fy=1.0,
interpolation=cv2.INTER_AREA)
        print("saving frame: {}".format(count))
        out.write(frame)
        ret, frame = captura.read()
    else: #caso contrario: pasar al siguiente cuadro
        ret, frame = captura.read()
        count = count+1

captura.release()
out.release()

#
#####Extraer frames #
#####
captura= cv2.VideoCapture(video_out) # cambiar nombre de video para cada
sujeto
carpeta= os.mkdir(carpeta_frames) #crea una nueva carpeta
img_index= 0

while (captura.isOpened()):
    ret,frame= captura.read()
    if ret== True:
        path= carpeta_frames # nombre de carpeta que se creo anteriormente donde
se van a guardar los frames del video
        cv2.imwrite(os.path.join(path,str(img_index)+'.jpg'),frame) # se guarada
cada frame en la carpeta en formato jpg

        img_index +=2
        cv2.imshow('P_01', frame) #se visualiza el video umbralizado

```

```

    if ret== False:
        break

    if cv2.waitKey(1) & 0xFF == ord('s'): # si se oprime la letra S el ciclo se
para
        break

captura.release()
cv2.destroyAllWindows()

```

9.3.3 Código para alinear imagenes:

Registro de imágenes

```

#
import numpy as np
import cv2
import os
from matplotlib import pyplot as plt
from skimage.io import imread
from skimage.io import imsave
from skimage.color import rgb2gray
from skimage.data import stereo_motorcycle, vortex
from skimage.transform import warp
from skimage.registration import optical_flow_tvll1, optical_flow_ilk

carpeta_registro= 'ingresar ruta donde se guardarán la imagenes alineadas'
dirCarpeta = 'Ingresar ruta de carpeta de frames'
im_ref = imread('ingresar ruta de imagen de referencia')

tiempo=[]

im_ref = rgb2gray(im_ref)

archivos = os.listdir(dirCarpeta)
archivos.sort (key = lambda x: int (x.replace ("", "").split ('.') [0]))

carpeta= os.mkdir(carpeta_registro)

for name in archivos:
    # tiempo.append(int(name[:-4]))
    im_roi=rgb2gray(imread(dirCarpeta +'/' +name))
    v, u = optical_flow_tvll1(im_ref,im_roi)
    nr, nc = im_ref.shape
    row_coords, col_coords = np.meshgrid(np.arange(nr), np.arange(nc),
indexing='ij')
    imagen1_warp=warp(im_roi, np.array([row_coords + v, col_coords + u]),
mode='edge')

    path='reg_18_P2'
    imsave(os.path.join(path,str(name[:-4])+'.jpg'),imagen1_warp)
}

```

9.3.4 Código para extraer ROI:

Extracción ROI

```

#
import cv2
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.pyplot as img
import os

carpetal= os.mkdir('ingresar nombre de carpeta donde se guardarán los ROI')
dirCarpeta1 = 'Nombre de carpeta de imágenes donde se extraerán los ROI'
archivos1 = os.listdir(dirCarpeta1)
archivos1.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
img_ref=cv2.imread(dirCarpeta1+'/' + archivos1[0])
roi= cv2.selectROI(windowName="roi", img=img_ref, showCrosshair=True,
fromCenter=False)
x,y,w,h= roi
cv2.rectangle (img=img_ref, pt1=(x,y), pt2=(x+w, y+h), color= (0,0,255),
thickness=2)
cv2.imshow("roi", img_ref)
for name in archivos1 :
    im_roi=cv2.imread(dirCarpeta1+'/' + name)
    recorte= im_roi[y:(y+h), x:(x+w)]
    path= 'ingresar nombre de carpeta donde se guardarán los ROI')# nombre de
carpeta que se creo anteriormente donde se van a guardar los frames del video
    cv2.imwrite(os.path.join(path,str(name[:-4])+'.jpg'),recorte)
}

```

9.3.5 Código para análisis estadístico:

Análisis estadístico

```

#
from pathlib import Path
from signal import CTRL_C_EVENT
from turtle import color
import cv2
import os
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt
from PIL import Image
from skimage.io import imsave
from skimage.io import imread

carpetal= os.mkdir('nombre de carpeta donde se guardarán los gráficos') # se
crea carpeta para guardar graficos

# Ruta de carpetas con ROI alienados de cada prueba
dirCarpeta1 = 'ingresar nombre se carpeta con imágenes de Prueba 1'
dirCarpeta2 = 'ingresar nombre se carpeta con imágenes de Prueba 2'
dirCarpeta3 = 'ingresar nombre se carpeta con imágenes de Prueba 3'
dirCarpeta4 = 'ingresar nombre se carpeta con imágenes de Prueba 4'

```

```

archivos1 = os.listdir(dirCarpeta1)
archivos2 = os.listdir(dirCarpeta2)
archivos3 = os.listdir(dirCarpeta3)
archivos4 = os.listdir(dirCarpeta4)

#####
##### se calcula las estadisticas de la carpeta#####
#####

tiempo1 =[]
tiempo2 =[]
tiempo3 =[]
tiempo4 =[]
vector_mean1=[]
vector_mean2=[]
vector_mean3=[]
vector_mean4=[]
imag1=[]
imag2=[]
imag3=[]
imag4=[]

archivos1.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos2.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos3.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos4.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
## se crean ciclo for para calcular las variables deseadas en las imagenes que
están contenidas en las diferentes carpetas

for name in archivos1:

    tiempo1.append(int(name[:-4]))
    im_gris = np.array(Image.open(dirCarpeta1+'/' + name))
    imagen_new = (im_gris-np.amin(im_gris))
    imagen_new = imagen_new/np.amax(imagen_new)
    rango = 36-19
    imagen_new = imagen_new*rango + 19
    imag1.append(imagen_new)
    vector_mean1.append((np.mean(imag1)))

for name in archivos2:

    tiempo2.append(int(name[:-4]))
    im_gris = np.array(Image.open(dirCarpeta2+'/' + name))
    imagen_new = (im_gris-np.amin(im_gris))
    imagen_new = imagen_new/np.amax(imagen_new)
    rango = 36-19
    imagen_new = imagen_new*rango + 19
    imag1.append(imagen_new)
    vector_mean2.append((np.mean(imag1)))

for name in archivos3:

    tiempo3.append(int(name[:-4]))
    im_gris = np.array(Image.open(dirCarpeta3+'/' + name))
    imagen_new = (im_gris-np.amin(im_gris))

```

```

imagen_new = imagen_new/np.amax(imagen_new)
rango = 36-19
imagen_new = imagen_new*rango + 19
imag1.append(imagen_new)
vector_mean3.append((np.mean(imag1)))

for name in archivos4:

    tiempo4.append(int(name[:-4]))
    im_gris = np.array(Image.open(dirCarpeta4+'/' + name))
    imagen_new = (im_gris-np.amin(im_gris))
    imagen_new = imagen_new/np.amax(imagen_new)
    rango = 36-19
    imagen_new = imagen_new*rango + 19
    imag1.append(imagen_new)
    vector_mean4.append((np.mean(imag1)))

path= 'nombre de carpeta donde se guardarán los graficos'

## Se crea y guarda grafico de Promedio
fig, (ax1,ax2,ax3,ax4) = plt.subplots(1, 4, sharey = True)
ax1.plot(tiempo1, vector_mean1, color = 'tab:purple')
ax2.plot(tiempo2, vector_mean2, color = 'tab:purple')
ax4.plot(tiempo4, vector_mean4, color = 'tab:purple')
ax1.set_ylabel('Promedio')
plt.savefig(os.path.join(path,"Promedio.png"), dpi=300)

}

```

9.3.6 Gráfico de cajas:

Mapeo de imagen

```

#
import cv2
import numpy as np
import os
from skimage.color import rgb2gray
import matplotlib.pyplot as plt

#Se analizan imágenes de todos los participantes durante prueba 1
dirCarpeta1='ruta de carpeta'
dirCarpeta2='ruta de carpeta'
dirCarpeta3='ruta de carpeta'
dirCarpeta4='ruta de carpeta'
dirCarpeta5='ruta de carpeta'
dirCarpeta7='ruta de carpeta'
dirCarpeta8='ruta de carpeta'
dirCarpeta9='ruta de carpeta'
dirCarpeta10='ruta de carpeta'
dirCarpeta11='ruta de carpeta'
dirCarpeta12='ruta de carpeta'
dirCarpeta13='ruta de carpeta'
dirCarpeta14='ruta de carpeta'

```

```
dirCarpeta15='ruta de carpeta'  
dirCarpeta16='ruta de carpeta'  
dirCarpeta17='ruta de carpeta'  
dirCarpeta18='ruta de carpeta'  
dirCarpeta19='ruta de carpeta'  
dirCarpeta20='ruta de carpeta'
```

```
archivos1 = os.listdir('ruta archivos')  
archivos2 = os.listdir('ruta archivos')  
archivos3 = os.listdir('ruta archivos')  
archivos4 = os.listdir('ruta archivos')  
archivos5 = os.listdir('ruta archivos')  
archivos7 = os.listdir('ruta archivos')  
archivos8 = os.listdir('ruta archivos')  
archivos9 = os.listdir('ruta archivos')  
archivos10 = os.listdir('ruta archivos')  
archivos11 = os.listdir('ruta archivos')  
archivos12 = os.listdir('ruta archivos')  
archivos13 = os.listdir('ruta archivos')  
archivos14 = os.listdir('ruta archivos')  
archivos15 = os.listdir('ruta archivos')  
archivos16 = os.listdir('ruta archivos')  
archivos17 = os.listdir('ruta archivos')  
archivos18 = os.listdir('ruta archivos')  
archivos19 = os.listdir('ruta archivos')  
archivos20 = os.listdir('ruta archivos')
```

```
archivos1.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos2.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos3.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos4.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos5.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos7.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos8.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos9.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos10.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos11.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos12.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos13.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos14.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos15.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos16.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos17.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos18.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos19.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos20.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
vector_mean1=[]  
vector_mean2=[]  
vector_mean3=[]  
vector_mean4=[]
```

```
for name in archivos1:
```

```
    imagen=rgb2gray (cv2.imread (dirCarpeta1 + '/' + name))  
    imagen_new = (imagen-np.amin (imagen))
```

```
imagen_new = imagen_new/np.amax(imagen_new)

rango = 36-19
imagen_new = imagen_new*rango + 19
vector_mean1.append((np.mean(imagen_new)))

for name in archivos2:

    imagen=rgb2gray(cv2.imread(dirCarpeta2 +'/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean1.append((np.mean(imagen_new)))

for name in archivos3:

    imagen=rgb2gray(cv2.imread(dirCarpeta3 +'/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean1.append((np.mean(imagen_new)))

for name in archivos4:

    imagen=rgb2gray(cv2.imread(dirCarpeta4 +'/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean1.append((np.mean(imagen_new)))

for name in archivos5:

    imagen=rgb2gray(cv2.imread(dirCarpeta5 +'/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean1.append((np.mean(imagen_new)))

for name in archivos7:

    imagen=rgb2gray(cv2.imread(dirCarpeta7 +'/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean1.append((np.mean(imagen_new)))
```

```
for name in archivos8:

    imagen=rgb2gray(cv2.imread(dirCarpeta8 + '/' + name))
    imagen_new = (imagen - np.amin(imagen))
    imagen_new = imagen_new / np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new * rango + 19
    vector_mean1.append((np.mean(imagen_new)))

for name in archivos9:

    imagen=rgb2gray(cv2.imread(dirCarpeta9 + '/' + name))
    imagen_new = (imagen - np.amin(imagen))
    imagen_new = imagen_new / np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new * rango + 19
    vector_mean1.append((np.mean(imagen_new)))

for name in archivos10:

    imagen=rgb2gray(cv2.imread(dirCarpeta10 + '/' + name))
    imagen_new = (imagen - np.amin(imagen))
    imagen_new = imagen_new / np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new * rango + 19
    vector_mean1.append((np.mean(imagen_new)))

for name in archivos11:

    imagen=rgb2gray(cv2.imread(dirCarpeta11 + '/' + name))
    imagen_new = (imagen - np.amin(imagen))
    imagen_new = imagen_new / np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new * rango + 19
    vector_mean1.append((np.mean(imagen_new)))

for name in archivos12:

    imagen=rgb2gray(cv2.imread(dirCarpeta12 + '/' + name))
    imagen_new = (imagen - np.amin(imagen))
    imagen_new = imagen_new / np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new * rango + 19
    vector_mean1.append((np.mean(imagen_new)))

for name in archivos13:

    imagen=rgb2gray(cv2.imread(dirCarpeta13 + '/' + name))
    imagen_new = (imagen - np.amin(imagen))
    imagen_new = imagen_new / np.amax(imagen_new)

    rango = 36-19
```

```
imagen_new = imagen_new*rango + 19
vector_mean1.append((np.mean(imagen_new)))

for name in archivos14:

    imagen=rgb2gray(cv2.imread(dirCarpeta14 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean1.append((np.mean(imagen_new)))

for name in archivos15:

    imagen=rgb2gray(cv2.imread(dirCarpeta15 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean1.append((np.mean(imagen_new)))

for name in archivos16:

    imagen=rgb2gray(cv2.imread(dirCarpeta16 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean1.append((np.mean(imagen_new)))

for name in archivos17:

    imagen=rgb2gray(cv2.imread(dirCarpeta17 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean1.append((np.mean(imagen_new)))

for name in archivos18:

    imagen=rgb2gray(cv2.imread(dirCarpeta18 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean1.append((np.mean(imagen_new)))

for name in archivos19:

    imagen=rgb2gray(cv2.imread(dirCarpeta19 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
```

```
imagen_new = imagen_new/np.amax(imagen_new)

rango = 36-19
imagen_new = imagen_new*rango + 19
vector_mean1.append((np.mean(imagen_new)))

for name in archivos20:

    imagen=rgb2gray(cv2.imread(dirCarpeta20 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean1.append((np.mean(imagen_new)))

#Se analizan imágenes de participantes durante prueba 2
dirCarpeta1='ruta de carpeta'
dirCarpeta2='ruta de carpeta'
dirCarpeta3='ruta de carpeta'
dirCarpeta4='ruta de carpeta'
dirCarpeta5='ruta de carpeta'
dirCarpeta7='ruta de carpeta'
dirCarpeta8='ruta de carpeta'
dirCarpeta9='ruta de carpeta'
dirCarpeta10='ruta de carpeta'
dirCarpeta11='ruta de carpeta'
dirCarpeta12='ruta de carpeta'
dirCarpeta13='ruta de carpeta'
dirCarpeta14='ruta de carpeta'
dirCarpeta15='ruta de carpeta'
dirCarpeta16='ruta de carpeta'
dirCarpeta17='ruta de carpeta'
dirCarpeta18='ruta de carpeta'
dirCarpeta19='ruta de carpeta'
dirCarpeta20='ruta de carpeta'

archivos1 = os.listdir('ruta archivos')
archivos2 = os.listdir('ruta archivos')
archivos3 = os.listdir('ruta archivos')
archivos4 = os.listdir('ruta archivos')
archivos5 = os.listdir('ruta archivos')
archivos7 = os.listdir('ruta archivos')
archivos8 = os.listdir('ruta archivos')
archivos9 = os.listdir('ruta archivos')
archivos10 = os.listdir('ruta archivos')
archivos11 = os.listdir('ruta archivos')
archivos12 = os.listdir('ruta archivos')
archivos13 = os.listdir('ruta archivos')
archivos14 = os.listdir('ruta archivos')
archivos15 = os.listdir('ruta archivos')
archivos16 = os.listdir('ruta archivos')
archivos17 = os.listdir('ruta archivos')
archivos18 = os.listdir('ruta archivos')
archivos19 = os.listdir('ruta archivos')
```

```
archivos20 = os.listdir('ruta archivos')

archivos1.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos2.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos3.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos4.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos5.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos7.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos8.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos9.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos10.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos11.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos12.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos13.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos14.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos15.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos16.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos17.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos18.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos19.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos20.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))

for name in archivos1:

    imagen=rgb2gray(cv2.imread(dirCarpeta1 +'/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean2.append((np.mean(imagen_new)))

for name in archivos2:

    imagen=rgb2gray(cv2.imread(dirCarpeta2 +'/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean2.append((np.mean(imagen_new)))

for name in archivos3:

    imagen=rgb2gray(cv2.imread(dirCarpeta3 +'/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean2.append((np.mean(imagen_new)))

for name in archivos4:
```

```
imagen=rgb2gray(cv2.imread(dirCarpeta4 + '/' + name))
imagen_new = (imagen - np.amin(imagen))
imagen_new = imagen_new / np.amax(imagen_new)

rango = 36-19
imagen_new = imagen_new * rango + 19
vector_mean2.append((np.mean(imagen_new)))

for name in archivos5:

    imagen=rgb2gray(cv2.imread(dirCarpeta5 + '/' + name))
    imagen_new = (imagen - np.amin(imagen))
    imagen_new = imagen_new / np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new * rango + 19
    vector_mean2.append((np.mean(imagen_new)))

for name in archivos7:

    imagen=rgb2gray(cv2.imread(dirCarpeta7 + '/' + name))
    imagen_new = (imagen - np.amin(imagen))
    imagen_new = imagen_new / np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new * rango + 19
    vector_mean2.append((np.mean(imagen_new)))

for name in archivos8:

    imagen=rgb2gray(cv2.imread(dirCarpeta8 + '/' + name))
    imagen_new = (imagen - np.amin(imagen))
    imagen_new = imagen_new / np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new * rango + 19
    vector_mean2.append((np.mean(imagen_new)))

for name in archivos9:

    imagen=rgb2gray(cv2.imread(dirCarpeta9 + '/' + name))
    imagen_new = (imagen - np.amin(imagen))
    imagen_new = imagen_new / np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new * rango + 19
    vector_mean2.append((np.mean(imagen_new)))

for name in archivos10:

    imagen=rgb2gray(cv2.imread(dirCarpeta10 + '/' + name))
    imagen_new = (imagen - np.amin(imagen))
    imagen_new = imagen_new / np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new * rango + 19
    vector_mean2.append((np.mean(imagen_new)))
```

```
for name in archivos11:

    imagen=rgb2gray(cv2.imread(dirCarpeta11 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean2.append((np.mean(imagen_new)))

for name in archivos12:

    imagen=rgb2gray(cv2.imread(dirCarpeta12 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean2.append((np.mean(imagen_new)))

for name in archivos13:

    imagen=rgb2gray(cv2.imread(dirCarpeta13 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean2.append((np.mean(imagen_new)))

for name in archivos14:

    imagen=rgb2gray(cv2.imread(dirCarpeta14 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean2.append((np.mean(imagen_new)))

for name in archivos15:

    imagen=rgb2gray(cv2.imread(dirCarpeta15 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean2.append((np.mean(imagen_new)))

for name in archivos16:

    imagen=rgb2gray(cv2.imread(dirCarpeta16 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)
```

```
rango = 36-19
imagen_new = imagen_new*rango + 19
vector_mean2.append((np.mean(imagen_new)))

for name in archivos17:

    imagen=rgb2gray(cv2.imread(dirCarpeta17 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean2.append((np.mean(imagen_new)))

for name in archivos18:

    imagen=rgb2gray(cv2.imread(dirCarpeta18 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean2.append((np.mean(imagen_new)))

for name in archivos19:

    imagen=rgb2gray(cv2.imread(dirCarpeta19 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean2.append((np.mean(imagen_new)))

for name in archivos20:

    imagen=rgb2gray(cv2.imread(dirCarpeta20 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean2.append((np.mean(imagen_new)))

#Se analizan imágenes de participantes durante prueba 3
dirCarpeta1='ruta de carpeta'
dirCarpeta2='ruta de carpeta'
dirCarpeta3='ruta de carpeta'
dirCarpeta4='ruta de carpeta'
dirCarpeta5='ruta de carpeta'
dirCarpeta7='ruta de carpeta'
dirCarpeta8='ruta de carpeta'
dirCarpeta9='ruta de carpeta'
dirCarpeta10='ruta de carpeta'
dirCarpeta11='ruta de carpeta'
dirCarpeta12='ruta de carpeta'
dirCarpeta13='ruta de carpeta'
```

```
dirCarpeta14='ruta de carpeta'  
dirCarpeta15='ruta de carpeta'  
dirCarpeta16='ruta de carpeta'  
dirCarpeta17='ruta de carpeta'  
dirCarpeta18='ruta de carpeta'  
dirCarpeta19='ruta de carpeta'  
dirCarpeta20='ruta de carpeta'
```

```
archivos1 = os.listdir('ruta archivos')  
archivos2 = os.listdir('ruta archivos')  
archivos3 = os.listdir('ruta archivos')  
archivos4 = os.listdir('ruta archivos')  
archivos5 = os.listdir('ruta archivos')  
archivos7 = os.listdir('ruta archivos')  
archivos8 = os.listdir('ruta archivos')  
archivos9 = os.listdir('ruta archivos')  
archivos10 = os.listdir('ruta archivos')  
archivos11 = os.listdir('ruta archivos')  
archivos12 = os.listdir('ruta archivos')  
archivos13 = os.listdir('ruta archivos')  
archivos14 = os.listdir('ruta archivos')  
archivos15 = os.listdir('ruta archivos')  
archivos16 = os.listdir('ruta archivos')  
archivos17 = os.listdir('ruta archivos')  
archivos18 = os.listdir('ruta archivos')  
archivos19 = os.listdir('ruta archivos')  
archivos20 = os.listdir('ruta archivos')
```

```
archivos1.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos2.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos3.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos4.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos5.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos7.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos8.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos9.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos10.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos11.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos12.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos13.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos14.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos15.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos16.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos17.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos18.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos19.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))  
archivos20.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
```

```
for name in archivos1:
```

```
    imagen=rgb2gray(cv2.imread(dirCarpeta1 + '/' +name))  
    imagen_new = (imagen-np.amin(imagen))  
    imagen_new = imagen_new/np.amax(imagen_new)
```

```
    rango = 36-19
```

```
imagen_new = imagen_new*rango + 19
vector_mean3.append((np.mean(imagen_new)))

for name in archivos2:

    imagen=rgb2gray(cv2.imread(dirCarpeta2 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean3.append((np.mean(imagen_new)))

for name in archivos3:

    imagen=rgb2gray(cv2.imread(dirCarpeta3 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean3.append((np.mean(imagen_new)))

for name in archivos4:

    imagen=rgb2gray(cv2.imread(dirCarpeta4 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean3.append((np.mean(imagen_new)))

for name in archivos5:

    imagen=rgb2gray(cv2.imread(dirCarpeta5 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean3.append((np.mean(imagen_new)))

for name in archivos7:

    imagen=rgb2gray(cv2.imread(dirCarpeta7 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean3.append((np.mean(imagen_new)))

for name in archivos8:

    imagen=rgb2gray(cv2.imread(dirCarpeta8 + '/' +name))
```

```
imagen_new = (imagen-np.amin(imagen))
imagen_new = imagen_new/np.amax(imagen_new)

rango = 36-19
imagen_new = imagen_new*rango + 19
vector_mean3.append((np.mean(imagen_new)))

for name in archivos9:

    imagen=rgb2gray(cv2.imread(dirCarpeta9 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean3.append((np.mean(imagen_new)))

for name in archivos10:

    imagen=rgb2gray(cv2.imread(dirCarpeta10 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean3.append((np.mean(imagen_new)))

for name in archivos11:

    imagen=rgb2gray(cv2.imread(dirCarpeta11 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean3.append((np.mean(imagen_new)))

for name in archivos12:

    imagen=rgb2gray(cv2.imread(dirCarpeta12 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean3.append((np.mean(imagen_new)))

for name in archivos13:

    imagen=rgb2gray(cv2.imread(dirCarpeta13 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean3.append((np.mean(imagen_new)))
```

```
for name in archivos14:

    imagen=rgb2gray(cv2.imread(dirCarpeta14 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean3.append((np.mean(imagen_new)))

for name in archivos15:

    imagen=rgb2gray(cv2.imread(dirCarpeta15 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean3.append((np.mean(imagen_new)))

for name in archivos16:

    imagen=rgb2gray(cv2.imread(dirCarpeta16 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean3.append((np.mean(imagen_new)))

for name in archivos17:

    imagen=rgb2gray(cv2.imread(dirCarpeta17 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean3.append((np.mean(imagen_new)))

for name in archivos18:

    imagen=rgb2gray(cv2.imread(dirCarpeta18 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean3.append((np.mean(imagen_new)))

for name in archivos19:

    imagen=rgb2gray(cv2.imread(dirCarpeta19 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
```

```
imagen_new = imagen_new*rango + 19
vector_mean3.append((np.mean(imagen_new)))

for name in archivos20:

    imagen=rgb2gray(cv2.imread(dirCarpeta20 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean3.append((np.mean(imagen_new)))

#Se analizan imágenes de participantes durante prueba 4
dirCarpeta1='ruta de carpeta'
dirCarpeta2='ruta de carpeta'
dirCarpeta3='ruta de carpeta'
dirCarpeta4='ruta de carpeta'
dirCarpeta5='ruta de carpeta'
dirCarpeta7='ruta de carpeta'
dirCarpeta8='ruta de carpeta'
dirCarpeta9='ruta de carpeta'
dirCarpeta10='ruta de carpeta'
dirCarpeta11='ruta de carpeta'
dirCarpeta12='ruta de carpeta'
dirCarpeta13='ruta de carpeta'
dirCarpeta14='ruta de carpeta'
dirCarpeta15='ruta de carpeta'
dirCarpeta16='ruta de carpeta'
dirCarpeta17='ruta de carpeta'
dirCarpeta18='ruta de carpeta'
dirCarpeta19='ruta de carpeta'
dirCarpeta20='ruta de carpeta'

archivos1 = os.listdir('ruta archivos')
archivos2 = os.listdir('ruta archivos')
archivos3 = os.listdir('ruta archivos')
archivos4 = os.listdir('ruta archivos')
archivos5 = os.listdir('ruta archivos')
archivos7 = os.listdir('ruta archivos')
archivos8 = os.listdir('ruta archivos')
archivos9 = os.listdir('ruta archivos')
archivos10 = os.listdir('ruta archivos')
archivos11 = os.listdir('ruta archivos')
archivos12 = os.listdir('ruta archivos')
archivos13 = os.listdir('ruta archivos')
archivos14 = os.listdir('ruta archivos')
archivos15 = os.listdir('ruta archivos')
archivos16 = os.listdir('ruta archivos')
archivos17 = os.listdir('ruta archivos')
archivos18 = os.listdir('ruta archivos')
archivos19 = os.listdir('ruta archivos')
archivos20 = os.listdir('ruta archivos')
```

```
archivos1.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos2.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos3.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos4.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos5.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos7.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos8.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos9.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos10.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos11.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos12.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos13.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos14.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos15.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos16.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos17.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos18.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos19.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
archivos20.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))
```

```
for name in archivos1:
```

```
    imagen=rgb2gray(cv2.imread(dirCarpeta1 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean4.append((np.mean(imagen_new)))
```

```
for name in archivos2:
```

```
    imagen=rgb2gray(cv2.imread(dirCarpeta2 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean4.append((np.mean(imagen_new)))
```

```
for name in archivos3:
```

```
    imagen=rgb2gray(cv2.imread(dirCarpeta3 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean4.append((np.mean(imagen_new)))
```

```
for name in archivos4:
```

```
    imagen=rgb2gray(cv2.imread(dirCarpeta4 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)
```

```
rango = 36-19
imagen_new = imagen_new*rango + 19
vector_mean4.append((np.mean(imagen_new)))

for name in archivos5:

    imagen=rgb2gray(cv2.imread(dirCarpeta5 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean4.append((np.mean(imagen_new)))

for name in archivos7:

    imagen=rgb2gray(cv2.imread(dirCarpeta7 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean4.append((np.mean(imagen_new)))

for name in archivos8:

    imagen=rgb2gray(cv2.imread(dirCarpeta8 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean4.append((np.mean(imagen_new)))

for name in archivos9:

    imagen=rgb2gray(cv2.imread(dirCarpeta9 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean4.append((np.mean(imagen_new)))

for name in archivos10:

    imagen=rgb2gray(cv2.imread(dirCarpeta10 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean4.append((np.mean(imagen_new)))

for name in archivos11:

    imagen=rgb2gray(cv2.imread(dirCarpeta11 + '/' + name))
```

```
imagen_new = (imagen-np.amin(imagen))
imagen_new = imagen_new/np.amax(imagen_new)

rango = 36-19
imagen_new = imagen_new*rango + 19
vector_mean4.append((np.mean(imagen_new)))

for name in archivos12:

    imagen=rgb2gray(cv2.imread(dirCarpeta12 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean4.append((np.mean(imagen_new)))

for name in archivos13:

    imagen=rgb2gray(cv2.imread(dirCarpeta13 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean4.append((np.mean(imagen_new)))

for name in archivos14:

    imagen=rgb2gray(cv2.imread(dirCarpeta14 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean4.append((np.mean(imagen_new)))

for name in archivos15:

    imagen=rgb2gray(cv2.imread(dirCarpeta15 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean4.append((np.mean(imagen_new)))

for name in archivos16:

    imagen=rgb2gray(cv2.imread(dirCarpeta16 + '/' +name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean4.append((np.mean(imagen_new)))
```

```
for name in archivos17:

    imagen=rgb2gray(cv2.imread(dirCarpeta17 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean4.append((np.mean(imagen_new)))

for name in archivos18:

    imagen=rgb2gray(cv2.imread(dirCarpeta18 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean4.append((np.mean(imagen_new)))

for name in archivos19:

    imagen=rgb2gray(cv2.imread(dirCarpeta19 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean4.append((np.mean(imagen_new)))

for name in archivos20:

    imagen=rgb2gray(cv2.imread(dirCarpeta20 + '/' + name))
    imagen_new = (imagen-np.amin(imagen))
    imagen_new = imagen_new/np.amax(imagen_new)

    rango = 36-19
    imagen_new = imagen_new*rango + 19
    vector_mean4.append((np.mean(imagen_new)))

path= 'C:/Users/maria/Documents/tt2/analisis_3'

fig, (ax1,ax2,ax3,ax4) = plt.subplots(1,4, sharey = True)
ax1.boxplot(vector_mean1)
ax1.set_xlabel('Prueba 1')
ax2.boxplot(vector_mean2)
ax2.set_xlabel('Prueba 2')
ax3.boxplot(vector_mean3)
ax3.set_xlabel('Prueba3')
ax4.boxplot(vector_mean4)
ax4.set_xlabel('Prueba 4')
plt.savefig(os.path.join(path,"box_med1.png"), dpi=300)

plt.show()
}
```

9.3.7 Código para Mapeo:

Mapeo de imagen

```
#
import os
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
from matplotlib.pyplot import imshow
from matplotlib.pyplot import cm

dirCarpeta = 'C:/Users/maria/Documents/tt2/python/reg_06_P4'

im_save='Var_P06_P4.jpg'

archivos = os.listdir(dirCarpeta)
vector_var=[]
imag=[]

archivos.sort (key = lambda x: int (x.replace ("", ""). split ('.') [0]))

for name in archivos:
    im_gris = np.array(Image.open(dirCarpeta+'/' + name))
    imag.append(im_gris)
    vector_var.append((np.var(im_gris.flatten())))

imagenes = np.array(imag)
var_imag = np.var(imagenes, axis = 0)

fig, ax = plt.subplots()
ax.imshow(var_imag, cmap = 'hot')
imshow(im_save, var_imag, cmap=cm.hot, dpi=300)
}
```

9.3.8 Código de Umbralización:

Umbralizar imagen

```
#
import cv2
import numpy as np
import os

imagen= cv2.imread('nombre de imagen')

umbral1= cv2.inRange(imagen, (67, 6, 122), (127, 66, 182))
extraido1= cv2.bitwise_and(imagen, imagen, mask=umbral1)
umbral2= cv2.inRange(imagen, (2, 62, 193), (62, 142, 255))
extraido2= cv2.bitwise_and(imagen, imagen, mask= umbral2)
```

```
umbral3= cv2.inRange(imagen, (85,127,195), (175,227,255))
extraido3= cv2.bitwise_and(imagen, imagen, mask= umbral3)
umbral4= cv2.inRange(imagen, (175,217,175), (255,255,255))
extraido4= cv2.bitwise_and(imagen, imagen, mask= umbral4)
juntos= extraido1+extraido2+extraido3+extraido4
path= 'Ruta para guardar imagen'
cv2.imwrite(os.path.join(path, 'img_1_color_1.jpg'), extraido1)
cv2.imwrite(os.path.join(path, 'img_1_color_2.jpg'), extraido2)
cv2.imwrite(os.path.join(path, 'img_1_color_3.jpg'), extraido3)
cv2.imwrite(os.path.join(path, 'img_1_color_4.jpg'), extraido4)
cv2.imwrite(os.path.join(path, 'img_1_color_5.jpg'), juntos)
}
```
