


| | | |
|--|-------------------|--------------------------------------|
|  | Computación | Docente: Vladimir Robles Bykbaev |
| | VISIÓN ARTIFICIAL | Período Lectivo: Marzo – Agosto 2024 |

| | | | |
|---|---|--|---|
|  | | FORMATO DE GUÍA DE PROYECTO INTEGRADOR | |
| CARRERA: COMPUTACIÓN | | ASIGNATURA: VISIÓN ARTIFICIAL | |
| NRO. PRÁCTICA: | 2-2 | TÍTULO DEL PROYECTO: Proyecto Integrador Final – Desarrollo de un sistema de visión artificial para detectar/clasificar objetos empleando deep learning y extracción de características | |
| OBJETIVO: Reforzar los conocimientos adquiridos en clase sobre el trabajo con las etapas de preprocesamiento de imágenes y extracción de características locales y globales y el empleo de técnicas del dominio de la frecuencia y de deep learning a fin de realizar tareas de reconocimiento de patrones. | | | |
| INSTRUCCIONES: | | 1. Revisar el contenido teórico del tema, las presentaciones, tutoriales y video-lecturas disponibles en el ambiente virtual de aprendizaje. | |
| | | 2. Profundizar los conocimientos revisando los libros guías, los enlaces contenidos en los objetos de aprendizaje y la documentación disponible en fuentes académicas en línea | |
| | | 3. | |
| | | 4. Deberá generar un informe empleando una herramienta Web 2.0 y un video-blog en inglés explicando los principales aspectos de la propuesta planteada. | |
| ASIGNATURAS Y CONOCIMIENTOS RELACIONADOS AL PROYECTO INTEGRADOR | | | |
| Asignatura | Contenido de la materia | Contenido al que aporta de la asignatura de Visión por Computador o requerimientos del Proyecto | Porcentaje de aporte al proyecto |
| Álgebra Lineal | Matrices (operaciones con matrices) | Operaciones sobre grupos de puntos de la imagen (convolución, filtro de la media, filtro de la mediana, filtro Gaussiano). Reducción de ruido. Operaciones morfológicas. HOG. | 25% |
| Cálculo Diferencial | La derivada | Detección de Bordes (Operadores Básicos, Prewitt, Sobel, Laplaciano, Canny) | 20% |
| | | | |
| Estadística | Introducción a la estadística y conjuntos Probabilidad Variables aleatorias y distribuciones de probabilidad | Ecuación de Histograma | 25% |
| Programación y plataformas Web | Desarrollo de aplicaciones Web | | 15% |
| Inteligencia Artificial | Redes Neuronales | Aplicación de visión por computador con aprendizaje profundo. | 15% |
| Visión por Computador | Espacios de Color Operaciones sobre los puntos de la imagen Operaciones Morfológicas Detección de Bordes Histogram Oriented Gradients (HOG) Cascadas de Haar | N/A | N/A |
| ACTIVIDADES POR DESARROLLAR | | | |
| <ul style="list-style-type: none"> Desarrollar una aplicación basada en visión artificial y organizada en dos etapas. La primera etapa se debe presentar en el interciclo, y la segunda etapa previo a finalizar el ciclo lectivo. La aplicación deberá cumplir con las siguientes especificaciones: | | | |

a) Parte I (Interciclo):

- Debe desarrollar una aplicación móvil y una aplicación básica web que permitan realizar el proceso de fusión de dos fuentes de imágenes/vídeos, como se puede observar en la Ilustración 1.

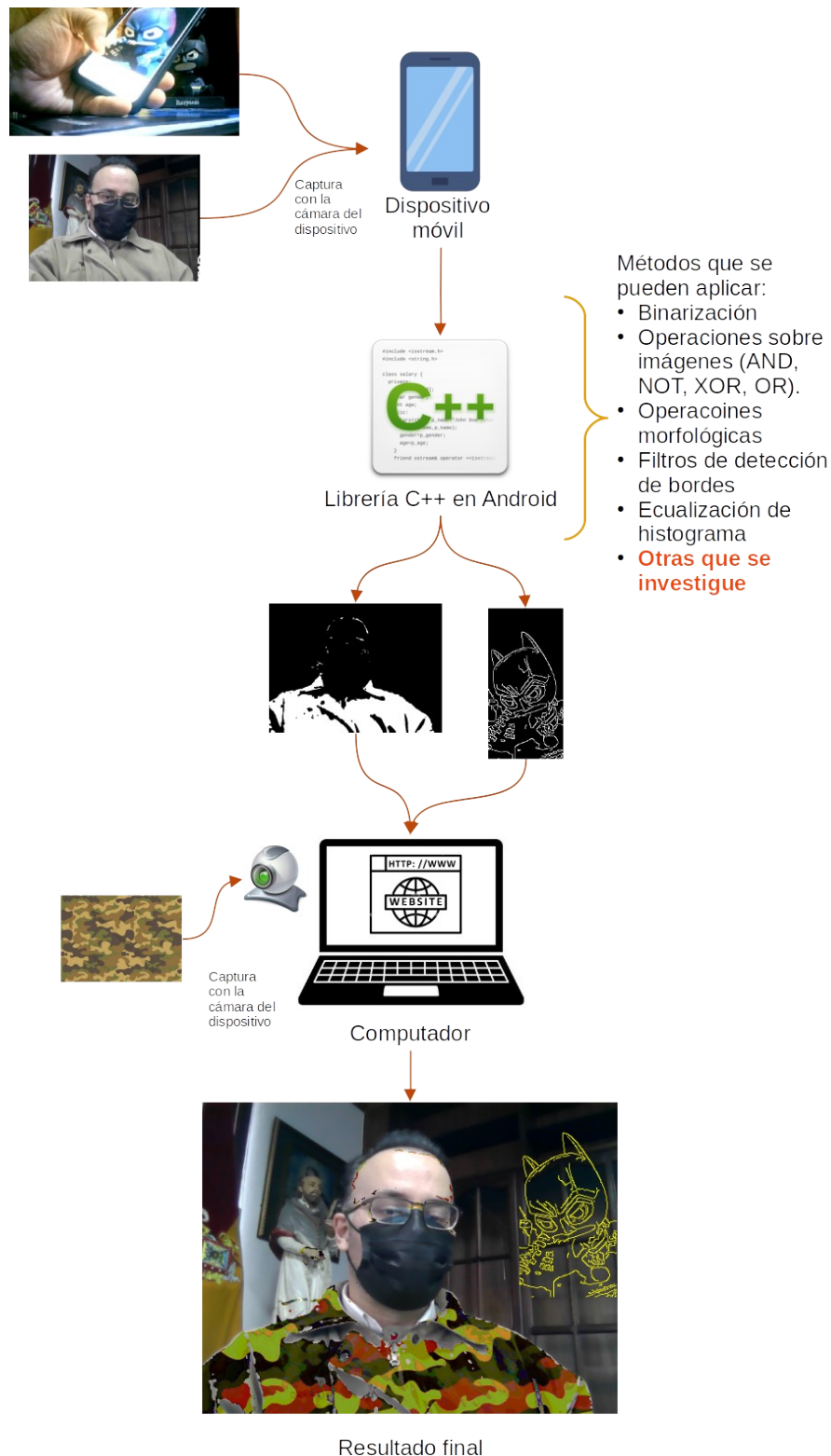


Ilustración 1. Diagrama que ilustra las etapas y procesos que se sugiere desarrollar en el proyecto. Como se observa, el dispositivo móvil captura dos imágenes, las procesa en una librería C++ dentro de Android y luego se envía al Servidor Web. El servidor Web aplica la textura a la ropa de la persona y fusiona con los bordes de la imagen de Batman.

- Los aspectos y funcionalidades con los que debe contar el proyecto desarrollado son los que se indican seguidamente.
 - La aplicación móvil permitirá capturar una secuencia de imágenes o vídeo y posterior a la captura, dentro de la aplicación móvil en la librería de C++ se realizará un pre-procesamiento de la imagen para luego generar un efecto.
 - Las imágenes pre-procesadas se envían a través de la red (HTTP o algún protocolo similar) a un servidor web.
 - El servidor web realiza una captura de imágenes o vídeo de un escenario o una fuente de vídeo que se fusionarán con las imágenes pre-procesadas y capturadas por el dispositivo móvil.
 - Como resultado final el sistema generará un vídeo o una secuencia de imágenes donde se han fusionado las imágenes origen tanto del dispositivo móvil como del servidor web.
- Las técnicas de visión artificial que deben emplearse en el pre-procesamiento tanto en el dispositivo móvil como en el servidor web:
 - Thersholding
 - Contrast Stretching
 - Binarización por umbral de color
 - Operaciones sobre los puntos de la imagen (NOT, AND, OR, XOR)
 - Detección de Bordes
 - Operaciones morfológicas
 - **Otras técnicas que se investigue**
- Aspectos generales a tomar en consideración:
 - Se debe diseñar la propuesta del efecto (de forma similar a cómo se presenta en la Ilustración 1).
 - Si usa códigos de terceros, debe indicar qué parte se ha tomado y de dónde (colocar citas), a fin de valorar correctamente el trabajo.
 - El proyecto debe presentarse funcionando en vivo durante la semana de revisión de exámenes. Considere que para ello puede ser que necesite traer su *router* o realizar configuración y pruebas previas en la red de la UPS.
 - **Fecha de entrega: Semana de revisión de exámenes interciclo.**

Documentación de Soporte:

- El enlace donde se encuentra el código de la aplicación móvil desarrollado y la explicación paso a paso es el siguiente:
 - <https://github.com/vlarobbyk/opencv4.2-android-studio-simple-app>
- Por ejemplo, como se aprecia en la Ilustración 2, en el enlace de github se muestra cómo desarrollar una aplicación móvil que permita trabajar con Visión Artificial usando librerías Nativas de C++:

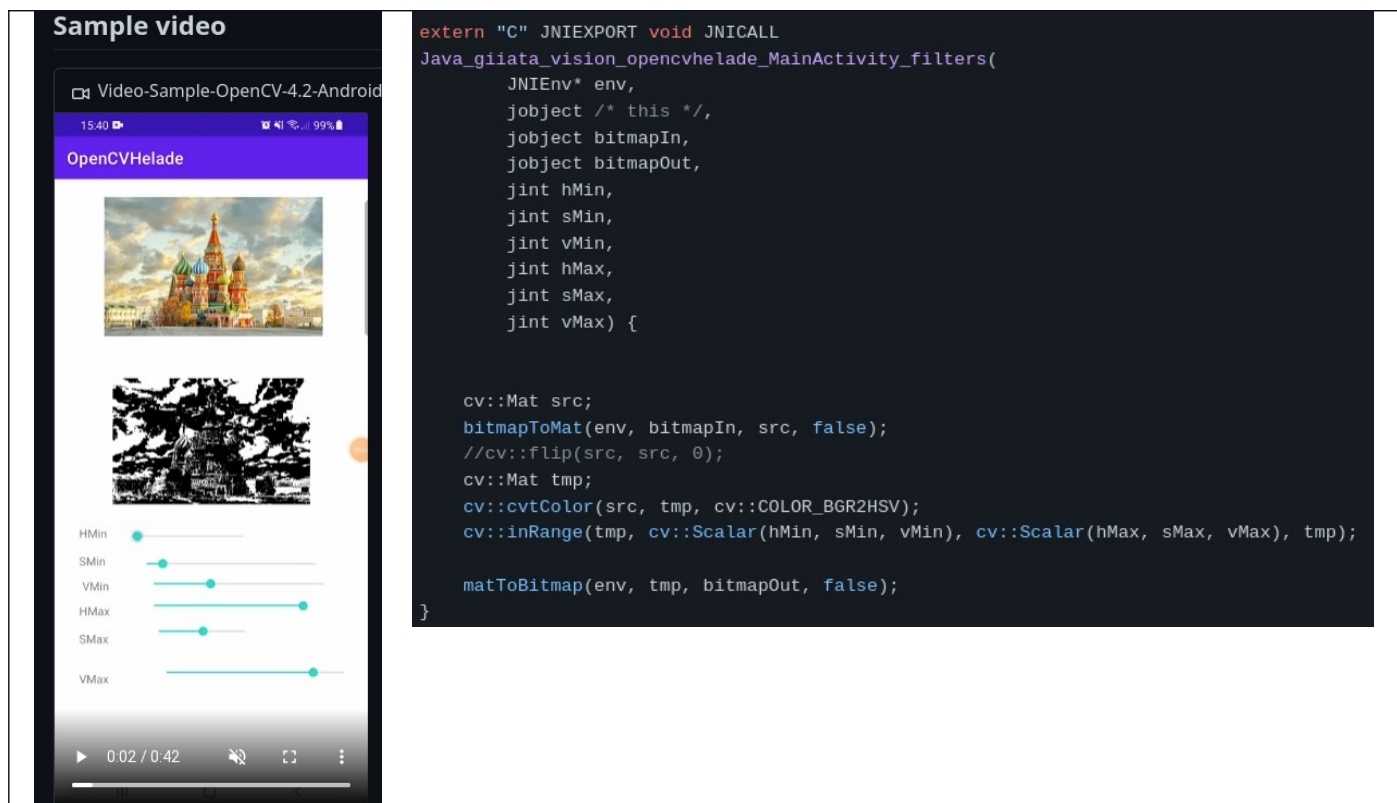


Ilustración 2. Aplicación móvil que emplea visión artificial y librerías nativas en Android. A la izquierda se aprecia la aplicación móvil desarrollada y la rapidez de respuesta que tiene cuando se usan librerías nativas, mientras que a la derecha se puede observar un fragmento de código para pasar datos de Java a C++.

RÚBRICA DE EVALUACIÓN PARA LA PARTE I PROYECTO (15 puntos):

| Proyecto Integrador | | | |
|---|-------------|-----|------|
| Criterio | Ponderación | | |
| El efecto que se emplea realiza la fusión de al menos 2 fuentes de imágenes o vídeos, y aplica al menos 3 de las operaciones vistas en clase | 20% | | |
| La imagen o vídeo resultante de la generación del efecto es nítida, no tiene ruido y se observa un resultado correctamente aplicado | 20% | | |
| El código está desarrollado usando una librería embebida de C++ donde se procesan las imágenes para posteriormente ser enviados a la aplicación web | 20% | | |
| Dentro del proceso se incluye una técnica nueva que se ha investigado por parte de los estudiantes | 15% | | |
| La aplicación móvil captura una imagen (10%) o realiza la captura de una secuencia de imágenes o videos (25%)* | 10% | 25% | |
| Total (sobre 100%) | | 85% | 100% |

*Este criterio tiene una ponderación distinta para el tipo de implementación que se realice. Si desarrolla la opción para capturar secuencias de imágenes (GIF o similares) o vídeo, el puntaje es de 25%. Si por otra parte, en lugar de ello, se captura únicamente una imagen, la ponderación es del 10%. Esto se da con el objetivo de que los estudiantes puedan optar por aplicar simplemente lo visto en clase si así lo requieren.

RÚBRICA DE EVALUACIÓN PARA EL INFORME DE LA PARTE I (5 puntos):

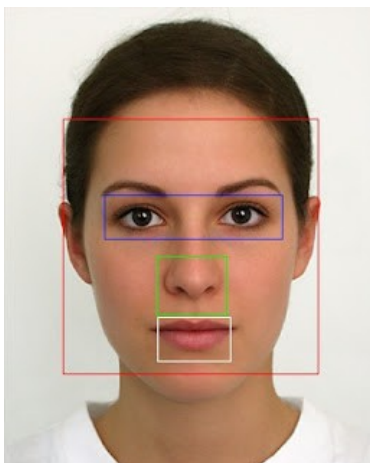
A continuación se especifican los criterios de evaluación:

Video – Blog en inglés (Sobre 4 Puntos)

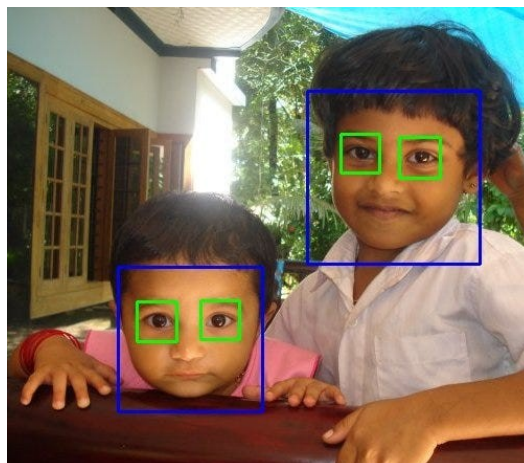
| Criterio | Ponderación |
|---|--------------|
| El informe incluye una descripción detallada del trabajo realizado (introducción, descripción del problema, propuesta de solución, conclusiones, bibliografía) | 35% |
| El informe incorpora resultados de pruebas realizadas con el sistema (gráficas, reportes, etc.) de los siguientes aspectos: precisión media, FPS (Frames Per Second), matriz de confusión, uso de memoria RAM, etc. | 25% |
| El informe está correctamente redactado, contiene citas a papers y un esquema explicativo de la solución planteada | 20% |
| El vídeo en inglés sigue un guion donde se explican los principales aspectos del proyecto de una manera clara y concisa | 20% |
| Total | 100% |
| Total | Total |

b) Parte II (Final):

- En base a la aplicación desarrollada en la Parte I, debe incorporar dos funcionalidades nuevas que se indican a continuación:
- Detección de rostros:
 - Debe programar un código que permita realizar la detección de rostros en el dispositivo móvil (librería en C++) usando Cascadas de Haar o un enfoque similar, de forma que se pueda detectar no solo el área donde está el rostro, sino también ojos, nariz y boca, como se puede apreciar en la Ilustración 3.



(a)



(b)

Ilustración 3. Ejemplos de la aplicación del algoritmo de Cascadas de Haar para detección de rostros. Fuente: (a) <http://opencvfacedetect.blogspot.com/2010/10/face-detectionfollowed-by-eyesnose.html>, (b) <https://www.analyticsvidhya.com/blog/2022/10/face-detection-using-haar-cascade-using-python/>

- Una vez detectado el rostro, deberá enviarse las imágenes o el vídeo capturado con las posiciones del rostro, ojos, nariz y boca al servidor web.

- El servidor web deberá aplicar un efecto sobre las partes del rostro, por ejemplo: colocar lentes o gafas, cambiar el color de los ojos, etc. Para ello, debe definir qué efecto desea aplicar y diseñar cómo se implementará el mismo.
- Clasificación de objetos usando Histogramas de Gradientes Orientados (HOG):
 - Deberá entrenar un clasificador de objetos empleando para ello el descriptor HOG y una Red Neuronal Perceptron Multicapa o similar.
 - Para ello, deberá seleccionar un corpus de imágenes sobre las cuales realizará la clasificación, se extraerá el descriptor HOG y posteriormente entrenar la red neuronal, como se puede observar en la Ilustración 4, donde se emplea el HOG para clasificar dígitos escritos a mano (corpus MNIST).

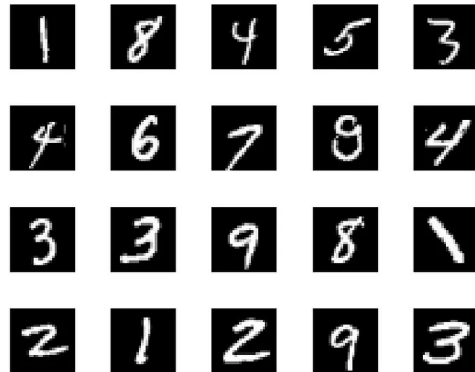


Ilustración 3. Ejemplo de imágenes del corpus MNIST con el que se realiza clasificación de dígitos empleando HOG. Fuente: <https://github.com/pransan/ComputerVisionAlgorithms/blob/master/Histogram%20of%20Oriented%20Gradients%20--%20MNIST/MNIST%20with%20HOG%20Features.ipynb>

- Una vez entrenada la red neuronal debe implementar el clasificador en el servidor web, de modo que cuando se cargue una imagen nueva se realice la predicción.
- Aspectos generales a tomar en consideración:
 - Se debe diseñar la propuesta del efecto (de forma similar a cómo se presenta en la Ilustración 1).
 - Si usa códigos de terceros, debe indicar qué parte se ha tomado y de dónde (colocar citas), a fin de valorar correctamente el trabajo.
 - El proyecto debe presentarse funcionando en vivo durante la semana de revisión de exámenes. Considere que para ello puede ser que necesite traer su *router* o realizar configuración y pruebas previas en la red de la UPS.
 - **Fecha de entrega: Semana de revisión de exámenes interciclo.**

Documentación de Soporte:

- Tutorial I para detección de rostros usando OpenCV:
 - <http://opencvfacedetect.blogspot.com/2010/10/face-detectionfollowed-by-eyesnose.html>
- Tutorial II para detección de rostros usando OpenCV:
 - <https://www.analyticsvidhya.com/blog/2022/10/face-detection-using-haar-cascade-using-python/>
- Tutorial HOG:
 - <https://waiyankyawmc.medium.com/histogram-of-oriented-gradients-90567ea6490a>
- Presentación de HOG por el Prof. Navneet Dalal:
 - http://vision.stanford.edu/teaching/cs231b_spring1213/slides/HOG_2011_Stanford.pdf
-

RÚBRICA DE EVALUACIÓN PARA EL PROYECTO PARTE II FINAL (20 puntos):

Proyecto Integrador

Criterio

Ponderación

| | | |
|--|-------------------|--------------------------------------|
|  | Computación | Docente: Vladimir Robles Bykbaev |
| | VISIÓN ARTIFICIAL | Período Lectivo: Marzo – Agosto 2024 |

| | | |
|---|-----|----------|
| El efecto que se emplea realiza la fusión de al menos 2 fuentes de imágenes o vídeos, y aplica al menos 3 de las operaciones vistas en clase | 20% | |
| Se detecta rostros (ojos, nariz y boca) con un nivel de precisión superior al 85% | 20% | |
| El código está desarrollado usando una librería embebida de C++ donde se procesan las imágenes para posteriormente ser enviados a la aplicación web | 20% | |
| El Detector HOG está implementado en una librería embebida en C++ y tiene un nivel de precisión superior al 70% | 15% | |
| La aplicación móvil captura una imagen (10%) o realiza la captura de una secuencia de imágenes o videos (25%)* | 10% | 25% |
| Total (sobre 100%) | | 85% 100% |

*Este criterio tiene una ponderación distinta para el tipo de implementación que se realice. Si desarrolla la opción para capturar secuencias de imágenes (GIF o similares) o vídeo, el puntaje es de 25%. Si por otra parte, en lugar de ello, se captura únicamente una imagen, la ponderación es del 10%. Esto se da con el objetivo de que los estudiantes puedan optar por aplicar simplemente lo visto en clase si así lo requieren.

RÚBRICA DE EVALUACIÓN PARA EL INFORME PARTE II FINAL (5 puntos):

A continuación se especifican los criterios de evaluación:

Video – Blog en inglés (Sobre 4 Puntos)

| Criterio | Ponderación |
|---|-------------|
| El informe incluye una descripción detallada del trabajo realizado (introducción, descripción del problema, propuesta de solución, conclusiones, bibliografía) | 35% |
| El informe incorpora resultados de pruebas realizadas con el sistema (gráficas, reportes, etc.) de los siguientes aspectos: precisión media, FPS (Frames Per Second), matriz de confusión, uso de memoria RAM, uso de procesar y tiempo de inferencia tanto con la red neuronal como con HOG. | 25% |
| El informe está correctamente redactado, contiene citas a papers y un esquema explicativo de la solución planteada | 20% |
| El vídeo en inglés sigue un guion donde se explican los principales aspectos del proyecto de una manera clara y concisa | 20% |
| Total | 100% |

RESULTADO(S) OBTENIDO(S):

Entender cómo se realiza el proceso para extraer descriptores globales y emplear deep learning para resolver problemas de clasificación o segmentación de imágenes.

CONCLUSIONES:

- Los estudiantes identifican qué técnicas de extracción de descriptores locales y globales permiten realizar seguimiento de objetos en vídeo.
- Los estudiantes identifican en que circunstancias se deben aplicar operaciones de preprocesamiento de la imagen que permiten realizar la extracción de descriptores con mayor precisión.
- Los estudiantes implementan soluciones que realizar identificación o clasificación de objetos usando técnicas como el Histogram of Oriented Gradients y la extracción de características.

RECOMENDACIONES:

- Revisar la información proporcionada por el docente previo a la práctica.
- Haber asistido a las sesiones de clase.

- Consultar con el docente las dudas que puedan surgir al momento de realizar la práctica.

Docente / Técnico Docente: Ing. Vladimir Robles Bykbaev

Firma: _____