

Programación Móvil

Dr. Marco Aurelio Nuño Maganda

Universidad Politecnica de Victoria
Ingeniería en Tecnologías de la Información
Cuatrimestre Septiembre - Diciembre 2024

mnunom@upv.edu.mx

August 29, 2024

Breve CV del Facilitador

- Doctor en Ciencias Computacionales por parte del INAOE (2009).
- Profesor de Tiempo Completo de la UPV desde 2009.
- Miembro del Sistema Nacional de Investigadores - Nivel Candidado (2014-2016), Nivel I (2020-2022), Nivel I (2023-2027)
- 17 tesis dirigidas a nivel maestría.
- Asignaturas impartidas en el pasado
 - Licenciatura: Cómputo en Dispositivos Mviles, Graficación por Computadora Avanzada, Lenguajes y Automátas, Programación Orientada a Objetos
 - Maestría: Visión por computadora, Tópicos Selectos de Imagenología, Fundamentos de Sistemas de Información
- Miembro del Núcleo Académico Básico (NAB) de la maestria en Ingeniería de la UPV.

Plataforma Virtual para el Curso

- Nombre de la clase: **Programación Móvil-Septiembre - Diciembre 2024**
- Código de clase en Classroom: **xv543pb**
- Enlace Meet para sesiones no presenciales:
<https://meet.google.com/jsb-nmxy-dht>

Horario de la Clase

■ Días y horas de clase

Clave de Grupo	Lunes	Martes	Miércoles	Jueves	Viernes
ITI-271311	14:00- 14:54	8:50- 10:40	14:00-14:54	9:45 - 10:40	12:05 - 13:00
ITI-271304		11:10 -13:00	9:45-12:05	11:10 -13:00	

■ Fechas Importantes:

- Inicio de Cursos: 2/Septiembre
- Fin de Cursos: 13/Diciembre
- Dias no hábiles oficiales: 16 de septiembre (lunes), 1 de octubre (martes) y 18 de noviembre (lunes).

Plataforma Virtual para el Curso

- Nombre de la clase: **Programación Móvil- Septiembre - Diciembre 2024**
- Código de clase en Classroom: **xv543pb**
- Enlace Meet para sesiones no presenciales:
<https://meet.google.com/jsb-nmxy-dht>

Reglas básicas

- Se recomienda puntualidad y asistencia a las sesiones.
- Respeto hacia el profesor y hacia sus compañeros y compañeras.
- No se permite el ingreso y/o ingestión de **Alimentos** ni **Bebidas** de ningún tipo a la clase.
- Solo se puede usar **AUDIFONOS O DISPOSITIVOS MANO-LIBRES EN CLASE** previa autorización por parte del profesor. Cualquier uso no AUTORIZADO es motivo de amonestación al estudiante, y expulsión en caso de reincidir sin derecho a réplica.

Uso del Teléfono Inteligente

- Se recomienda no utilizarlo durante el transcurso de la clase. Depende del comportamiento del grupo que esto no sea aplicado...

Resguardo del teléfono inteligente

De ser necesario, se solicitará al INICIO de la CLASE a todos los asistentes a la clase (incluyendo al profesor) guardar su telefono en una caja, la cual será cerrada, regresando su telefono al finalizar la SESION.



Pase de Lista

- Se pasa lista al inicio de la clase. En caso de reincorporación tardía, se pone un retardo.
- DOS RETARDOS equivalen a una INASISTENCIA, que no es JUSTIFICABLE.
- Para justificar una inasistencia, es necesario cumplir con los siguientes pasos:
 - Ingresar al apartado de Google classroom creado para dicho fin.
 - Agregar un PDF cuyo nombre debe
iti-clavegrupo-apat-amat-nombre-justifica-dd-mm-yy.pdf dd es el día de la inasistencia mm es el mes de su inasistencia yy es el año de su inasistencia
 - El interior de ese PDF debe contener documento que justifique su inasistencia. **Solo se reciben inasistencias por motivos medicos y de trabajo (SAT, Pasaporte, VISA, entrevista de trabajo)).** Los motivos meramente personales quedan cubiertos por el 20% de faltas que se conviene para el estudiante.

Alumnos con Empleo (1)

- Al NO alcanzar un 80% de asistencia, el estudiante pierde su derecho de ser EVALUADO

Alumnos VIPs

En caso de tener un empleo formal dentro o fuera de la ciudad, es necesario entregar una **constancia laboral** que acredite el horario que se esta cubriendo (en el caso de locales, este horario se debe empalmar con el de la materia). En esa constancia debe acreditar que se esta haciendo labores de manera presencial en tal ubicacion. Esto lo dispensa solo del requisito de las asistencias, mas no de los proyectos que deban entregarse. Incluso pudiera solicitarle presentar avance de manera “remota” durante alguna de las clases. Enviar esa constancia con copia para el director de carrera.

Alumnos con Empleo (2)

- La justificación de inasistencias por *actividad laboral* se considerará a partir del momento de la recepción de dicha constancia en el correo del instructor (y no a partir de la fecha indicada en la constancia), por lo que si se recibe de manera tardía (con mas de una semana de retardo), dichas inasistencias NO SERAN justificadas.
- La justificación será válida si el estudiante programa **POR LO MENOS** dos asesorías por semana. De no hacerlo, pierde el beneficio de la justificación y se aplican las reglas anteriormente establecidas.

- 1 Introducción a la Inteligencia Artificial
 - 1 Inteligencia Artificial y Sistemas Inteligentes
 - 2 Razonamiento y representación del conocimiento
 - 3 Estrategias de Búsqueda
- 2 Técnicas Básicas de Aprendizaje
 - 1 Aprendizaje Supervisado y No Supervisado
 - 2 Aprendizaje basado en Redes Neuronales
 - 3 Algoritmos Genéticos
- 3 Introducción a la percepción automática
 - 1 Percepción Visual
 - 2 Percepción Visual Estereoscópica
 - 3 Procesamiento de Lenguaje Natural

Evaluación (1)

- Para cada unidad del curso, se consideran 3 aspectos:
 - Ejercicios o investigaciones especiales (1)- 25%
 - Proyecto Individual - 35%
 - Proyecto en Equipo - 40%
- Para aprobar el curso, es obligatorio:
 - Tener calificación aprobatoria en todas las unidades (100-100-40 no da calificación aprobatoria).
 - Tener por lo menos dos asesorías por semana (Registrarlas por semana, no 30 asesorías al final del cuatrimestre)
 - Cumplir con el 80% de asistencia mínimo, incluyendo aquellas inasistencias justificadas debidamente mediante el SIITA

Evaluación (2)

Para cada unidad, habra sesiones de “teoria”, sesiones de seguimiento de proyectos y sesiones de esparcimiento

- En las sesiones de teoria, el profesor presentara uno o varios temas
- En las sesiones de seguimiento de proyectos, de manera aleatoria se nombrara al integrante de equipo individual o en equipo. En el caso de que un integrante individual no responda, se le bajarán 5 puntos a su calificación del proyecto
- En las sesiones de esparcimiento, se permitirá a los estudiantes trabajar en proyectos pendientes, pero se contabilizará la asistencia.

Evaluación (3)

Sesiones de Seguimiento de proyectos

- En el caso de que el integrante del equipo seleccionado aleatoriamente no responda satisfactoriamente lo cuestionado, se le bajaran 5 puntos a su calificación del proyecto a todos los integrantes del equipo
- En el caso de los proyectos en equipo, el integrante seleccionado es aleatorio. Si en una primera ronda le toco al integrante A, en una segunda ronda posiblemente le toque al integrante B

Evaluación (4)

Lo que se debe presentar en una sesion de seguimiento de proyectos

- En un trabajo individual
 - Compartir pantalla de la ejecucion del avance del proyecto
 - Explicar con recursos multimedia los pasos para la resolucion del proyecto
 - Establecer el avance desde la ultima entrega
- En un trabajo grupal
 - Compartir pantalla de la ejecucion del avance del proyecto
 - Explicar con recursos multimedia los pasos para la resolucion del proyecto
 - Desglosar como se repartio el trabajo entre los integrantes del equipo
 - Establecer el avance desde la ultima entrega

Evaluación (5)

Acerca de los proyectos

- Aleatorios y DIFERENTES para la mayoría (preferentemente para cada integrante)
- Equipos: Proyectos diferentes para cada equipo, e Integrantes de los mismos formados de manera ALEATORIA!!

Fragmentación de equipos

- Si llegar a ocurrir que en un proyecto en equipo no hay un acuerdo para trabajar en equipo (Hay dos o mas entregas del proyecto asignado por partes diferentes dentro del mismo equipo)

Penalización

Cada “fragmento” de equipo recibe una penalizacion de 25 puntos mas las penalizaciones acumuladas por otros rubros.

- esta regla **NO APLICA** cuando hay uno o varios “desertores” del equipo (y hay una sola entrega del proyectos en equipo)

Acerca de Exención

- Cuando el profesor realizar alguna mecánica para excentar un proyecto (Individual/Equipo/Asignación especial) y uno o varios estudiantes completan lo solicitado, existen dos posibilidades:
 - El estudiante acepta excentar la elaboración de dicho proyecto o actividad, pero al hacer esto asume que la calificación asignada es 70.
 - El estudiante decide hacer el proyecto a pesar de haber excentado. En este caso el estudiante se hace acreedor a 20 puntos que puede aplicar sobre la calificación de dicho proyecto.

Cartucho de Recuperación (REC)

- Estudiante tiene derecho a solicitar un ÚNICO proyecto de recuperación aplicable a un solo proyecto o actividad.
- Esta solicitud debe HACERLA es estudiante - El profesor NO ES RESPONSABLE de informar al estudiante cuando tiene un ADEUDO.
- Si el proyecto no entregado es individual, se asigna otro proyecto diferente.
- Si el proyecto es en equipo, de común acuerdo con los integrantes pueden trabajar en otro proyecto diferente en equipo, o recibir una asignación individual de un proyecto diferente.
- La calificación recuperada será asignada siempre y cuando cumpla con el porcentaje de falta mínimo necesario para aprobar. Además, debe haber agendado el % de asesorías proporcional al tiempo de cuatrimestre transcurrido.
- El nuevo proyecto asignado esta diseñado para que el estudiante invierta en él por lo menos 1 SEMANA. Si lo solicita un día antes de terminar el cuatrimestre, posiblemente no tendrá tiempo de llevarlo a cabo.

Reporte Técnico de Desarrollo de Práctica

- Para cada práctica realizada, entregar un documento (**únicamente en formato PDF***) con las siguientes secciones:
 - Introducción
 - Desarrollo Experimental
 - Resultados
 - Conclusiones
 - **Referencias**
- Para GENERAR este reporte es necesario utilizar la plantilla en LATEX (**únicamente usando LATEX***) localizada en el siguiente enlace:
<https://www.overleaf.com/read/dgkhvfwnygvc>

Reporte Técnico de Desarrollo de Práctica

- Bajo ninguna circunstancia deben incluir **CÓDIGO FUENTE**. Si pueden incluir diagrama de flujo, Pseudocódigo, Diagrama E-R, Diagrama de Clases, de Casos de USO, etc. De incluir código fuente, solo tendrá un 50% del valor en la calificación.
- En caso de trabajos individuales o en EQUIPO, deben emplear la plantilla LaTeX que se provee. En caso de utilizar algo diferente a LaTeX u otra plantilla de LaTeX, la calificación proporcional del informe será **DESESTIMADA**.
- En caso de trabajos en equipo, se debe agregar los integrantes al inicio del INFORME. **El trabajo solo cuenta para aquellos integrantes mencionados en el informe (y que dicho nombre se encuentre registrado tal cual en la lista). Una vez ENTREGADO, si hay OMISIONES de los integrantes, no se realizará CORRECCION alguna, se debe asumir la consecuencias que esto conlleva.**

Ponderación del Informe en la Calificación del Proyecto

- Informe: 34 Puntos
 - Uso adecuado de Latex: 5 Puntos
 - Organización y Redacción: 6 Puntos
 - Referencias en formato adecuado: 8 Puntos
 - Evidencia del trabajo realizado: 8 Puntos
 - Sin faltas de ortografía ni errores de dedo: 7 Puntos
- Proyecto: 66 Puntos
 - Ejecución y Funcionalidad: 45 Puntos
 - Modularidad: 13 Puntos
 - Documentación: 8 Puntos

Entregables de proyecto individual (1)

- Crear un archivo ZIP con el siguiente formato de nombre:
 - **iti-271229_uX_nuno_maganda_marco_aurelio**
- Dentro, debe contener lo siguiente:
 - **iti-271229_uX_nuno_maganda_marco_aurelio_source** (Carpeta con código fuente de la aplicación)
 - **iti-271229_uX_nuno_maganda_marco_aurelio_latex** (Carpeta con código fuente del informe)
 - **iti-271229_uX_nuno_maganda_marco_aurelio.apk** (Instalable (solo si se trata de una aplicación móvil))
 - **iti-271229_uX_nuno_maganda_marco_aurelio.pdf** (Informe)
- Donde:
 - **X** es el número de unidad a un dígito (1, 2, etc)
 - **Sustituir con sus apellidos y nombres de manera apropiada**
- NO DEBE HABER otros archivos .ZIP dentro del ZIP PRINCIPAL

Entregables de proyecto individual (2)

- En el caso que un proyecto individual sea asignado en equipo a varios estudiantes, el archivo entregable DEBE MANEJARSE como la de un proyecto individual
 - Solo un integrante del equipo carga en la plataforma el entregable individual.
 - El informe debe llevar los nombres de los integrantes del equipo que trabajaron (Si se omite a alguien, se asume que no trabajo en el proyecto).
 - NO ES NECESARIO que los otros integrantes marquen en el sistema la tarea como entregada, ya que se conoce su situación desde que se asigna el proyecto. El profesor ya sabe que ustedes van en equipo con el estudiante que hizo la entrega, y por eso deben asegurarse que en el informe entregado, vayan anotados sus nombres.

Entregables de proyectos en equipo

- Crear un archivo ZIP con el siguiente formato de nombre:
 - **iti-271229_eq_NN_uX**
- Dentro, debe contener lo siguiente:
 - **iti-271229_eq_NN_uX_source** (Carpeta con código fuente de la aplicación)
 - **iti-271229_eq_NN_uX_latex** (Carpeta con código fuente del informe)
 - **iti-271229_eq_NN_uX.apk** (Instalable - Solo aplicaciones móviles)
 - **iti-271229_eq_NN_uX.pdf** (Informe)

Donde:

- **NN** es el número de equipo a dos dígitos (01, 02, etc)
 - **X** es el número de unidad a un dígito (1, 2, etc)
- En cada entrega, **UN SOLO INTEGRANTE DEL EQUIPO** deberá cargar los archivos en el classroom.
- NO DEBE HABER otros archivos .ZIP dentro del ZIP PRINCIPAL

Entregables de asignaciones especiales

- Crear un archivo ZIP con el siguiente formato de nombre:
 - **iti-271229_aeX_uY_nuno_maganda_marco_aurelio**
- Dentro, debe contener lo siguiente:
 - **iti-271229_aeX_uY_nuno_maganda_marco_aurelio_source** (Carpeta con código fuente de la aplicación - Cuando aplique)
 - **iti-271229_aeX_uY_nuno_maganda_marco_aurelio_latex** (Carpeta con código fuente del informe o diapositivas)
 - **iti-271229_aeX_uY_nuno_maganda_marco_aurelio.apk** (Instalable - Solo aplicaciones móviles, Cuando aplique)
 - **iti-271229_aeX_uY_nuno_maganda_marco_aurelio.pdf** (Informe)
- Donde:
 - **X** es el número de asignación dentro de la unidad a un dígito (1, 2, etc)
 - **Y** es el número de unidad a un dígito (1, 2, etc)
 - **Sustituir con sus apellidos y nombres de manera apropiada**
- NO DEBE HABER otros archivos .ZIP dentro del ZIP PRINCIPAL

Nombres de Archivos Entregables

En el caso de nombres y apellidos acentuados, con diéresis o con virgulilla (~), sustituir de acuerdo con las siguientes reglas:

- Sustituir N/n por Ñ/ñ
- Sustituir A/a por Á/á
- Sustituir E/e por É/é
- Sustituir I/i por Í/í
- Sustituir O/o por Ó/ó
- Sustituir U/u por Ú/ú
- Sustituir U/u por Ü/ü

Penalizaciones por Entregas Incompletas

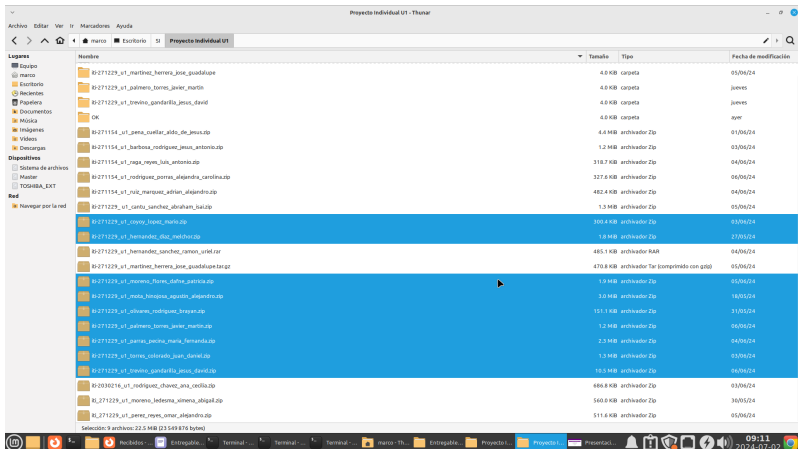
- Proyecto que no este entregado de acuerdo con las especificaciones, será penalizado. Dos escenarios posibles:
 - El proyecto puede revisarse (completo o con faltas al formato).
 - El proyecto NO puede revisarse (falta codigo fuente, informe, APK, no se compila por alguna falla, etc). En automático el proyecto queda descartado.

Se recomienda LEER con cuidado la sección de entregables de esta presentación. Las penalizaciones son acumulables.




Detalle	Puntos de penalización sobre calificación final
Nombre Archivo	8
Tipo de Archivo	8
Estructura de Directorios	8
Falta o Error en Script	8
Poner ZIPs dentro del ZIP	8
Incluir ejecutables (aplica solo cuando el lenguaje es C++)	8




Premio a la Compresión Lectora 2024




A pesar de las instrucciones en esta presentación, alguien va a hacer las cosas mal









Salon de la Fama de Entregas Completas e Incompletas



Nombre
 iti-271229_u1_trevino_gandarilla_jesus_david_source
 iti-271229_u1_trevino_gandarilla_jesus_david_latex
 iti-271229_u1_trevino_gandarilla_jesus_david.pdf

Nombre
 iti-271229_u1_coyoy_lopez_mario_source
 iti-271229_u1_coyoy_lopez_mario_latex
 iti_271229_u1_coyoy_lopez_mario.pdf

Nombre
 iti-271229_u1_olivares_rodriguez_brayan_source
 iti-271229_u1_olivares_rodriguez_brayan_latex
 iti-271229_u1_olivares_rodriguez_brayan.pdf

Nombre
 iti-271229_u1_martinez_herrera_jose_guadalupe_source
 REPORTE INDIVIDUAL U2.zip
 REPORTE_INDIVIDUAL_U2.pdf

Nombre
 iti-271154_u1_rodriguez_porras_alejandra_carolina_latex.zip
 iti-271154_u1_rodriguez_porras_alejandra_carolina_source.zip
 iti_271154_u1_rodriguez_porras_alejandra_carolina_latex .pdf

Nombre
 iti-271229_u1_parras_pecina_maria_fernanda_latex
 iti-271229_u1_parras_pecina_maria_fernanda_source

Respecto a las Asignaciones Especiales

Importante

Si en algún momento del curso, la asignación especial consiste en desarrollar una presentación de un tema explicado en un BLOG, capítulo de libro, tutorial, etc., el LENGUAJE en el que deben hacerse las DIAPOSITIVAS es el MISMO que en el que está explicado dicho BLOG (a menos que se establezca un lenguaje diferente de manera explícita en el momento de asignar dicha tarea).

Fechas importantes de entrega de proyectos (1)

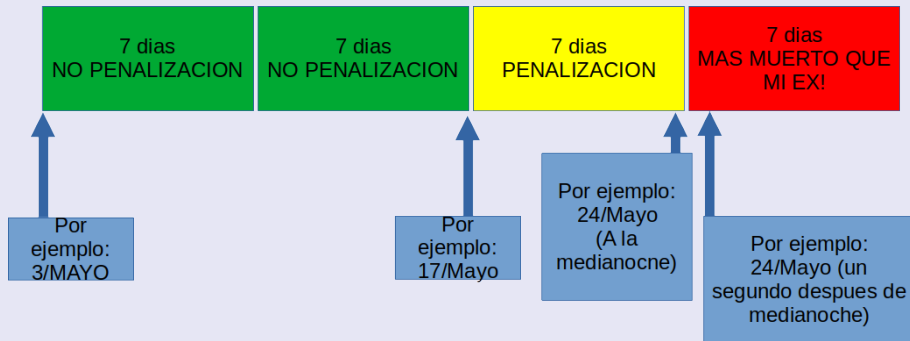
- Fecha de asignación: fecha en que se da a conocer al grupo el trabajo a elaborar
- Fecha de entrega sin penalización: 14 días naturales después de la fecha de asignación
- Proyecto entregado después de la fecha de penalización se le aplica una penalización de 25 PUNTOS
- Fecha de cierre: 21 días naturales después de la fecha de asignación.

Regla “CANTU”

- Ningún proyecto será revisado después de la fecha de cierre. Se programarán las entregas para cerrar y no permitir entregas tardías.

Fechas importantes de entrega de proyectos (2)

Grafo "DAFNE"



- En el momento de publicar la tarea, se incluirá la fecha para no penalización y fecha de cierre.

¿Es posible obtener una calificación negativa?

SI

Calificación asignada después de revisión	70
(-) Debería entregarse 17/Mayo, lo entregó 24/Mayo (1 minuto antes de la medianoche)	25
(-) Nombre Archivo MAL	8
(-) Tipo de Archivo MAL	8
(-) Estructura de Directorios INCORRECTA	8
(-) Faltas o Errores en Script	8
(-) ZIPs dentro del ZIP	8
(-) Incluir ejecutables (aplica solo cuando el lenguaje es C++)	8
(-) Penalización por Fragmentación de Equipo	25
Calificación Final	-28

LINUX

En orden de dificultad

- Linux instalado de manera emulada usando VirtualBox o VMWare.
- Crear una USB o HD booteable (con persistencia) y bootear desde su laptop solo para las clases y los proyectos.
- Linux instalado de manera nativa. Distribuciones recomendadas: **Mint, Ubuntu, Lubuntu, Xubuntu, Debian**

**** Tienen la opción de no INSTALAR LINUX, pero la evaluación será realiza en una PC con Linux instalado**

Software Utilizado

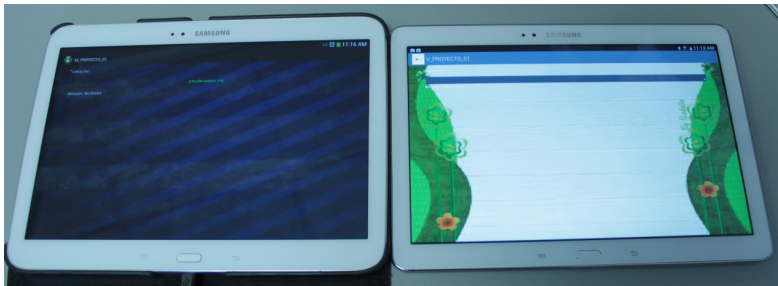
Sobre una instalación de Linux, se debe instalar lo siguiente:

- Navegador Chrome/Firefox actualizado
- LaTeX para edición de reportes
- Python3
- Otras librerías (se especificarán conforme se vayan utilizando)

Se buscan integrantes para ingresar al
Salon de la fama del PLAGIO

Plagio

- Reprobación automática a quien reproduzca códigos de otros compañeros y los reporte como suyos, además de una nota en su expediente con copia para el consejo de calidad



- Reprobación automática a quien copie códigos de Internet y los reporte como suyos, además de una nota en su expediente con copia para el consejo de calidad

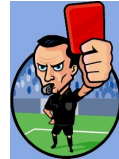


The screenshot shows a development environment with two windows. The left window is a C++ code editor containing the following code:

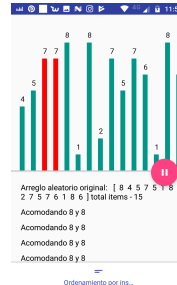
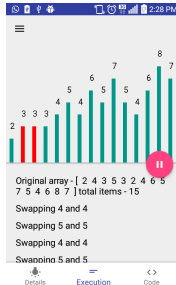
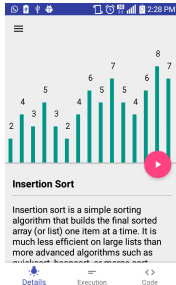
```

1 //Histogram equalization using C++ Image Processing
2
3 #include <iostream>
4 #include <opencv2/opencv.hpp>
5 #include <opencv2/imgproc/imgproc.hpp>
6
7 using std::cout;
8 using std::endl;
9
10 using namespace cv;
11 using namespace std;
12
13 //Funcion que inicializa todos los valores a 0
14 void initHistImage (cv::Histogram*)
15 {
16     for(int i = 0; i < 256; i++)
17     {
18         histImage[i] = 0;
19     }
20 }
21
22 //Calculando el numero de pixeles de cada valor de intensidad
23 for(int i = 0; i < image.rows; i++)
24 {
25     for(int j = 0; j < image.cols; j++)
26     {
27         histImage[histImage.at<uchar>(i,j)]++;
28     }
29 }
30
31 //Funcion para calcular el histograma
32 void calcHistImage (cv::Histogram*, int const histImage[])
33 {
34     calcHistImage[0] = histImage[0];
35     for(int i = 1; i < 256; i++)
36     {
37         calcHistImage[i] = calcHistImage[i-1] + histImage[i];
38     }
39 }
40
41 //Funcion que muestra los histogramas
42 void showHistImage (cv::Histogram*, const char* name)
43 {
44     cv::Mat hist[256];
45     for(int i = 0; i < 256; i++)
46     {
47         hist[i] = histImage[i];
48     }
49 }
50
51 //Calculando los histogramas
52 for(int i = 0; i < 256; i++)
53 {
54     for(int j = 0; j < 256; j++)
55     {
56         hist[i] = histImage[i];
57     }
58 }
59
60 //Normalizando la imagen intensidad del histograma
61 for(int i = 0; i < 256; i++)
62 {
63     for(int j = 0; j < 256; j++)
64     {
65         hist[i] = histImage[i];
66     }
67 }
68
69 //Normalizando la imagen intensidad del histograma
70 for(int i = 0; i < 256; i++)
71 {
72     for(int j = 0; j < 256; j++)
73     {
74         hist[i] = histImage[i];
75     }
76 }
77
78 //Normalizando la imagen intensidad del histograma
79 for(int i = 0; i < 256; i++)
80 {
81     for(int j = 0; j < 256; j++)
82     {
83         hist[i] = histImage[i];
84     }
85 }
86
87 //Normalizando la imagen intensidad del histograma
88 for(int i = 0; i < 256; i++)
89 {
90     for(int j = 0; j < 256; j++)
91     {
92         hist[i] = histImage[i];
93     }
94 }
95
96 //Normalizando la imagen intensidad del histograma
97 for(int i = 0; i < 256; i++)
98 {
99     for(int j = 0; j < 256; j++)
100    {
101        hist[i] = histImage[i];
102    }
103 }
104
105 //Normalizando la imagen intensidad del histograma
106 for(int i = 0; i < 256; i++)
107 {
108     for(int j = 0; j < 256; j++)
109     {
110         hist[i] = histImage[i];
111     }
112 }
113
114 //Normalizando la imagen intensidad del histograma
115 for(int i = 0; i < 256; i++)
116 {
117     for(int j = 0; j < 256; j++)
118     {
119         hist[i] = histImage[i];
120     }
121 }
122
123 //Normalizando la imagen intensidad del histograma
124 for(int i = 0; i < 256; i++)
125 {
126     for(int j = 0; j < 256; j++)
127     {
128         hist[i] = histImage[i];
129     }
130 }
131
132 //Normalizando la imagen intensidad del histograma
133 for(int i = 0; i < 256; i++)
134 {
135     for(int j = 0; j < 256; j++)
136     {
137         hist[i] = histImage[i];
138     }
139 }
140
141 //Normalizando la imagen intensidad del histograma
142 for(int i = 0; i < 256; i++)
143 {
144     for(int j = 0; j < 256; j++)
145     {
146         hist[i] = histImage[i];
147     }
148 }
149
150 //Normalizando la imagen intensidad del histograma
151 for(int i = 0; i < 256; i++)
152 {
153     for(int j = 0; j < 256; j++)
154     {
155         hist[i] = histImage[i];
156     }
157 }
158
159 //Normalizando la imagen intensidad del histograma
160 for(int i = 0; i < 256; i++)
161 {
162     for(int j = 0; j < 256; j++)
163     {
164         hist[i] = histImage[i];
165     }
166 }
167
168 //Normalizando la imagen intensidad del histograma
169 for(int i = 0; i < 256; i++)
170 {
171     for(int j = 0; j < 256; j++)
172     {
173         hist[i] = histImage[i];
174     }
175 }
176
177 //Normalizando la imagen intensidad del histograma
178 for(int i = 0; i < 256; i++)
179 {
180     for(int j = 0; j < 256; j++)
181     {
182         hist[i] = histImage[i];
183     }
184 }
185
186 //Normalizando la imagen intensidad del histograma
187 for(int i = 0; i < 256; i++)
188 {
189     for(int j = 0; j < 256; j++)
190     {
191         hist[i] = histImage[i];
192     }
193 }
194
195 //Normalizando la imagen intensidad del histograma
196 for(int i = 0; i < 256; i++)
197 {
198     for(int j = 0; j < 256; j++)
199     {
200         hist[i] = histImage[i];
201     }
202 }
203
204 //Normalizando la imagen intensidad del histograma
205 for(int i = 0; i < 256; i++)
206 {
207     for(int j = 0; j < 256; j++)
208     {
209         hist[i] = histImage[i];
210     }
211 }
212
213 //Normalizando la imagen intensidad del histograma
214 for(int i = 0; i < 256; i++)
215 {
216     for(int j = 0; j < 256; j++)
217     {
218         hist[i] = histImage[i];
219     }
220 }
221
222 //Normalizando la imagen intensidad del histograma
223 for(int i = 0; i < 256; i++)
224 {
225     for(int j = 0; j < 256; j++)
226     {
227         hist[i] = histImage[i];
228     }
229 }
230
231 //Normalizando la imagen intensidad del histograma
232 for(int i = 0; i < 256; i++)
233 {
234     for(int j = 0; j < 256; j++)
235     {
236         hist[i] = histImage[i];
237     }
238 }
239
240 //Normalizando la imagen intensidad del histograma
241 for(int i = 0; i < 256; i++)
242 {
243     for(int j = 0; j < 256; j++)
244     {
245         hist[i] = histImage[i];
246     }
247 }
248
249 //Normalizando la imagen intensidad del histograma
250 for(int i = 0; i < 256; i++)
251 {
252     for(int j = 0; j < 256; j++)
253     {
254         hist[i] = histImage[i];
255     }
256 }
257
258 //Normalizando la imagen intensidad del histograma
259 for(int i = 0; i < 256; i++)
260 {
261     for(int j = 0; j < 256; j++)
262     {
263         hist[i] = histImage[i];
264     }
265 }
266
267 //Normalizando la imagen intensidad del histograma
268 for(int i = 0; i < 256; i++)
269 {
270     for(int j = 0; j < 256; j++)
271     {
272         hist[i] = histImage[i];
273     }
274 }
275
276 //Normalizando la imagen intensidad del histograma
277 for(int i = 0; i < 256; i++)
278 {
279     for(int j = 0; j < 256; j++)
280     {
281         hist[i] = histImage[i];
282     }
283 }
284
285 //Normalizando la imagen intensidad del histograma
286 for(int i = 0; i < 256; i++)
287 {
288     for(int j = 0; j < 256; j++)
289     {
290         hist[i] = histImage[i];
291     }
292 }
293
294 //Normalizando la imagen intensidad del histograma
295 for(int i = 0; i < 256; i++)
296 {
297     for(int j = 0; j < 256; j++)
298     {
299         hist[i] = histImage[i];
300     }
301 }
302
303 //Normalizando la imagen intensidad del histograma
304 for(int i = 0; i < 256; i++)
305 {
306     for(int j = 0; j < 256; j++)
307     {
308         hist[i] = histImage[i];
309     }
310 }
311
312 //Normalizando la imagen intensidad del histograma
313 for(int i = 0; i < 256; i++)
314 {
315     for(int j = 0; j < 256; j++)
316     {
317         hist[i] = histImage[i];
318     }
319 }
320
321 //Normalizando la imagen intensidad del histograma
322 for(int i = 0; i < 256; i++)
323 {
324     for(int j = 0; j < 256; j++)
325     {
326         hist[i] = histImage[i];
327     }
328 }
329
330 //Normalizando la imagen intensidad del histograma
331 for(int i = 0; i < 256; i++)
332 {
333     for(int j = 0; j < 256; j++)
334     {
335         hist[i] = histImage[i];
336     }
337 }
338
339 //Normalizando la imagen intensidad del histograma
340 for(int i = 0; i < 256; i++)
341 {
342     for(int j = 0; j < 256; j++)
343     {
344         hist[i] = histImage[i];
345     }
346 }
347
348 //Normalizando la imagen intensidad del histograma
349 for(int i = 0; i < 256; i++)
350 {
351     for(int j = 0; j < 256; j++)
352     {
353         hist[i] = histImage[i];
354     }
355 }
356
357 //Normalizando la imagen intensidad del histograma
358 for(int i = 0; i < 256; i++)
359 {
360     for(int j = 0; j < 256; j++)
361     {
362         hist[i] = histImage[i];
363     }
364 }
365
366 //Normalizando la imagen intensidad del histograma
367 for(int i = 0; i < 256; i++)
368 {
369     for(int j = 0; j < 256; j++)
370     {
371         hist[i] = histImage[i];
372     }
373 }
374
375 //Normalizando la imagen intensidad del histograma
376 for(int i = 0; i < 256; i++)
377 {
378     for(int j = 0; j < 256; j++)
379     {
380         hist[i] = histImage[i];
381     }
382 }
383
384 //Normalizando la imagen intensidad del histograma
385 for(int i = 0; i < 256; i++)
386 {
387     for(int j = 0; j < 256; j++)
388     {
389         hist[i] = histImage[i];
390     }
391 }
392
393 //Normalizando la imagen intensidad del histograma
394 for(int i = 0; i < 256; i++)
395 {
396     for(int j = 0; j < 256; j++)
397     {
398         hist[i] = histImage[i];
399     }
400 }
401
402 //Normalizando la imagen intensidad del histograma
403 for(int i = 0; i < 256; i++)
404 {
405     for(int j = 0; j < 256; j++)
406     {
407         hist[i] = histImage[i];
408     }
409 }
410
411 //Normalizando la imagen intensidad del histograma
412 for(int i = 0; i < 256; i++)
413 {
414     for(int j = 0; j < 256; j++)
415     {
416         hist[i] = histImage[i];
417     }
418 }
419
420 //Normalizando la imagen intensidad del histograma
421 for(int i = 0; i < 256; i++)
422 {
423     for(int j = 0; j < 256; j++)
424     {
425         hist[i] = histImage[i];
426     }
427 }
428
429 //Normalizando la imagen intensidad del histograma
430 for(int i = 0; i < 256; i++)
431 {
432     for(int j = 0; j < 256; j++)
433     {
434         hist[i] = histImage[i];
435     }
436 }
437
438 //Normalizando la imagen intensidad del histograma
439 for(int i = 0; i < 256; i++)
440 {
441     for(int j = 0; j < 256; j++)
442     {
443         hist[i] = histImage[i];
444     }
445 }
446
447 //Normalizando la imagen intensidad del histograma
448 for(int i = 0; i < 256; i++)
449 {
450     for(int j = 0; j < 256; j++)
451     {
452         hist[i] = histImage[i];
453     }
454 }
455
456 //Normalizando la imagen intensidad del histograma
457 for(int i = 0; i < 256; i++)
458 {
459     for(int j = 0; j < 256; j++)
460     {
461         hist[i] = histImage[i];
462     }
463 }
464
465 //Normalizando la imagen intensidad del histograma
466 for(int i = 0; i < 256; i++)
467 {
468     for(int j = 0; j < 256; j++)
469     {
470         hist[i] = histImage[i];
471     }
472 }
473
474 //Normalizando la imagen intensidad del histograma
475 for(int i = 0; i < 256; i++)
476 {
477     for(int j = 0; j < 256; j++)
478     {
479         hist[i] = histImage[i];
480     }
481 }
482
483 //Normalizando la imagen intensidad del histograma
484 for(int i = 0; i < 256; i++)
485 {
486     for(int j = 0; j < 256; j++)
487     {
488         hist[i] = histImage[i];
489     }
490 }
491
492 //Normalizando la imagen intensidad del histograma
493 for(int i = 0; i < 256; i++)
494 {
495     for(int j = 0; j < 256; j++)
496     {
497         hist[i] = histImage[i];
498     }
499 }
500
501 //Normalizando la imagen intensidad del histograma
502 for(int i = 0; i < 256; i++)
503 {
504     for(int j = 0; j < 256; j++)
505     {
506         hist[i] = histImage[i];
507     }
508 }
509
510 //Normalizando la imagen intensidad del histograma
511 for(int i = 0; i < 256; i++)
512 {
513     for(int j = 0; j < 256; j++)
514     {
515         hist[i] = histImage[i];
516     }
517 }
518
519 //Normalizando la imagen intensidad del histograma
520 for(int i = 0; i < 256; i++)
521 {
522     for(int j = 0; j < 256; j++)
523     {
524         hist[i] = histImage[i];
525     }
526 }
527
528 //Normalizando la imagen intensidad del histograma
529 for(int i = 0; i < 256; i++)
530 {
531     for(int j = 0; j < 256; j++)
532     {
533         hist[i] = histImage[i];
534     }
535 }
536
537 //Normalizando la imagen intensidad del histograma
538 for(int i = 0; i < 256; i++)
539 {
540     for(int j = 0; j < 256; j++)
541     {
542         hist[i] = histImage[i];
543     }
544 }
545
546 //Normalizando la imagen intensidad del histograma
547 for(int i = 0; i < 256; i++)
548 {
549     for(int j = 0; j < 256; j++)
550     {
551         hist[i] = histImage[i];
552     }
553 }
554
555 //Normalizando la imagen intensidad del histograma
556 for(int i = 0; i < 256; i++)
557 {
558     for(int j = 0; j < 256; j++)
559     {
560         hist[i] = histImage[i];
561     }
562 }
563
564 //Normalizando la imagen intensidad del histograma
565 for(int i = 0; i < 256; i++)
566 {
567     for(int j = 0; j < 256; j++)
568     {
569         hist[i] = histImage[i];
570     }
571 }
572
573 //Normalizando la imagen intensidad del histograma
574 for(int i = 0; i < 256; i++)
575 {
576     for(int j = 0; j < 256; j++)
577     {
578         hist[i] = histImage[i];
579     }
580 }
581
582 //Normalizando la imagen intensidad del histograma
583 for(int i = 0; i < 256; i++)
584 {
585     for(int j = 0; j < 256; j++)
586     {
587         hist[i] = histImage[i];
588     }
589 }
590
591 //Normalizando la imagen intensidad del histograma
592 for(int i = 0; i < 256; i++)
593 {
594     for(int j = 0; j < 256; j++)
595     {
596         hist[i] = histImage[i];
597     }
598 }
599
600 //Normalizando la imagen intensidad del histograma
601 for(int i = 0; i < 256; i++)
602 {
603     for(int j = 0; j < 256; j++)
604     {
605         hist[i] = histImage[i];
606     }
607 }
608
609 //Normalizando la imagen intensidad del histograma
610 for(int i = 0; i < 256; i++)
611 {
612     for(int j = 0; j < 256; j++)
613     {
614         hist[i] = histImage[i];
615     }
616 }
617
618 //Normalizando la imagen intensidad del histograma
619 for(int i = 0; i < 256; i++)
620 {
621     for(int j = 0; j < 256; j++)
622     {
623         hist[i] = histImage[i];
624     }
625 }
626
627 //Normalizando la imagen intensidad del histograma
628 for(int i = 0; i < 256; i++)
629 {
630     for(int j = 0; j < 256; j++)
631     {
632         hist[i] = histImage[i];
633     }
634 }
635
636 //Normalizando la imagen intensidad del histograma
637 for(int i = 0; i < 256; i++)
638 {
639     for(int j = 0; j < 256; j++)
640     {
641         hist[i] = histImage[i];
642     }
643 }
644
645 //Normalizando la imagen intensidad del histograma
646 for(int i = 0; i < 256; i++)
647 {
648     for(int j = 0; j < 256; j++)
649     {
650         hist[i] = histImage[i];
651     }
652 }
653
654 //Normalizando la imagen intensidad del histograma
655 for(int i = 0; i < 256; i++)
656 {
657     for(int j = 0; j < 256; j++)
658     {
659         hist[i] = histImage[i];
660     }
661 }
662
663 //Normalizando la imagen intensidad del histograma
664 for(int i = 0; i < 256; i++)
665 {
666     for(int j = 0; j < 256; j++)
667     {
668         hist[i] = histImage[i];
669     }
670 }
671
672 //Normalizando la imagen intensidad del histograma
673 for(int i = 0; i < 256; i++)
674 {
675     for(int j = 0; j < 256; j++)
676     {
677         hist[i] = histImage[i];
678     }
679 }
680
681 //Normalizando la imagen intensidad del histograma
682 for(int i = 0; i < 256; i++)
683 {
684     for(int j = 0; j < 256; j++)
685     {
686         hist[i] = histImage[i];
687     }
688 }
689
690 //Normalizando la imagen intensidad del histograma
691 for(int i = 0; i < 256; i++)
692 {
693     for(int j = 0; j < 256; j++)
694     {
695         hist[i] = histImage[i];
696     }
697 }
698
699 //Normalizando la imagen intensidad del histograma
700 for(int i = 0; i < 256; i++)
701 {
702     for(int j = 0; j < 256; j++)
703     {
704         hist[i] = histImage[i];
705     }
706 }
707
708 //Normalizando la imagen intensidad del histograma
709 for(int i = 0; i < 256; i++)
710 {
711     for(int j = 0; j < 256; j++)
712     {
713         hist[i] = histImage[i];
714     }
715 }
716
717 //Normalizando la imagen intensidad del histograma
718 for(int i = 0; i < 256; i++)
719 {
720     for(int j = 0; j < 256; j++)
721     {
722         hist[i] = histImage[i];
723     }
724 }
725
726 //Normalizando la imagen intensidad del histograma
727 for(int i = 0; i < 256; i++)
728 {
729     for(int j = 0; j < 256; j++)
730     {
731         hist[i] = histImage[i];
732     }
733 }
734
735 //Normalizando la imagen intensidad del histograma
736 for(int i = 0; i < 256; i++)
737 {
738     for(int j = 0; j < 256; j++)
739     {
740         hist[i] = histImage[i];
741     }
742 }
743
744 //Normalizando la imagen intensidad del histograma
745 for(int i = 0; i < 256; i++)
746 {
747     for(int j = 0; j < 256; j++)
748     {
749         hist[i] = histImage[i];
750     }
751 }
752
753 //Normalizando la imagen intensidad del histograma
754 for(int i = 0; i < 256; i++)
755 {
756     for(int j = 0; j < 256; j++)
757     {
758         hist[i] = histImage[i];
759     }
760 }
761
762 //Normalizando la imagen intensidad del histograma
763 for(int i = 0; i < 256; i++)
764 {
765     for(int j = 0; j < 256; j++)
766     {
767         hist[i] = histImage[i];
768     }
769 }
770
771 //Normalizando la imagen intensidad del histograma
772 for(int i = 0; i < 256; i++)
773 {
774     for(int j = 0; j < 256; j++)
775     {
776         hist[i] = histImage[i];
777     }
778 }
779
780 //Normalizando la imagen intensidad del histograma
781 for(int i = 0; i < 256; i++)
782 {
783     for(int j = 0; j < 256; j++)
784     {
785         hist[i] = histImage[i];
786     }
787 }
788
789 //Normalizando la imagen intensidad del histograma
790 for(int i = 0; i < 256; i++)
791 {
792     for(int j = 0; j < 256; j++)
793     {
794         hist[i] = histImage[i];
795     }
796 }
797
798 //Normalizando la imagen intensidad del histograma
799 for(int i = 0; i < 256; i++)
800 {
801     for(int j = 0; j < 256; j++)
802     {
803         hist[i] = histImage[i];
804     }
805 }
806
807 //Normalizando la imagen intensidad del histograma
808 for(int i = 0; i < 256; i++)
809 {
810     for(int j = 0; j < 256; j++)
811     {
812         hist[i] = histImage[i];
813     }
814 }
815
816 //Normalizando la imagen intensidad del histograma
817 for(int i = 0; i < 256; i++)
818 {
819     for(int j = 0; j < 256; j++)
820     {
821         hist[i] = histImage[i];
822     }
823 }
824
825 //Normalizando la imagen intensidad del histograma
826 for(int i = 0; i < 256; i++)
827 {
828     for(int j = 0; j < 256; j++)
829     {
830         hist[i] = histImage[i];
831     }
832 }
833
834 //Normalizando la imagen intensidad del histograma
835 for(int i = 0; i < 256; i++)
836 {
837     for(int j = 0; j < 256; j++)
838     {
839         hist[i] = histImage[i];
840     }
841 }
842
843 //Normalizando la imagen intensidad del histograma
844 for(int i = 0; i < 256; i++)
845 {
846     for(int j = 0; j < 256; j++)
847     {
848         hist[i] = histImage[i];
849     }
850 }
851
852 //Normalizando la imagen intensidad del histograma
853 for(int i = 0; i < 256; i++)
854 {
855     for(int j = 0; j < 256; j++)
856     {
857         hist[i] = histImage[i];
858     }
859 }
860
861 //Normalizando la imagen intensidad del histograma
862 for(int i = 0; i < 256; i++)
863 {
864     for(int j = 0; j < 256; j++)
865     {
866         hist[i] = histImage[i];
867     }
868 }
869
870 //Normalizando la imagen intensidad del histograma
871 for(int i = 0; i < 256; i++)
872 {
873     for(int j = 0; j < 256; j++)
874     {
875         hist[i] = histImage[i];
876     }
877 }
878
879 //Normalizando la imagen intensidad del histograma
880 for(int i = 0; i < 256; i++)
881 {
882     for(int j = 0; j < 256; j++)
883     {
884         hist[i] = histImage[i];
885     }
886 }
887
888 //Normalizando la imagen intensidad del histograma
889 for(int i = 0; i < 256; i++)
890 {
891     for(int j = 0; j < 256; j++)
892     {
893         hist[i] = histImage[i];
894     }
895 }
896
897 //Normalizando la imagen intensidad del histograma
898 for(int i = 0; i < 256; i++)
899 {
900     for(int j = 0; j < 256; j++)
901     {
902         hist[i] = histImage[i];
903     }
904 }
905
906 //Normalizando la imagen intensidad del histograma
907 for(int i = 0; i < 256; i++)
908 {
909     for(int j = 0; j < 256; j++)
910     {
911         hist[i] = histImage[i];
912     }
913 }
914
915 //Normalizando la imagen intensidad del histograma
916 for(int i = 0; i < 256; i++)
917 {
918     for(int j = 0; j < 256; j++)
919     {
920         hist[i] = histImage[i];
921     }
922 }
923
924 //Normalizando la imagen intensidad del histograma
925 for(int i = 0; i < 256; i++)
926 {
927     for(int j = 0; j < 256; j++)
928     {
929         hist[i] = histImage[i];
930     }
931 }
932
933 //Normalizando la imagen intensidad del histograma
934 for(int i = 0; i < 256; i++)
935 {
936     for(int j = 0; j < 256; j++)
937     {
938         hist[i] = histImage[i];
939     }
940 }
941
942 //Normalizando la imagen intensidad del histograma
943 for(int i = 0; i < 256; i++)
944 {
945     for(int j = 0; j < 256; j++)
946     {
947         hist[i] = histImage[i];
948     }
949 }
950
951 //Normalizando la imagen intensidad del histograma
952 for(int i = 0; i < 256; i++)
953 {
954     for(int j = 0; j < 256; j++)
955     {
956         hist[i] = histImage[i];
957     }
958 }
959
960 //Normalizando la imagen intensidad del histograma
961 for(int i = 0; i < 256; i++)
962 {
963     for(int j = 0; j < 256; j++)
964     {
965         hist[i] = histImage[i];
966     }
967 }
968
969 //Normalizando la imagen intensidad del histograma
970 for(int i = 0; i < 256; i++)
971 {
972     for(int j = 0; j < 256; j++)
973     {
974         hist[i] = histImage[i];
975     }
976 }
977
978 //Normalizando la imagen intensidad del histograma
979 for(int i = 0; i < 256; i++)
980 {
981     for(int j = 0; j < 256; j++)
982     {
983         hist[i] = histImage[i];
984     }
985 }
986
987 //Normalizando la imagen intensidad del histograma
988 for(int i = 0; i < 256; i++)
989 {
990     for(int j = 0; j < 256; j++)
991     {
992         hist[i] = histImage[i];
993     }
994 }
995
996 //Normalizando la imagen intensidad del histograma
997 for(int i = 0; i < 256; i++)
998 {
999     for(int j = 0; j < 256; j++)
1000    {
1001        hist[i] = histImage[i];
1002    }
1003 }
1004
1005 //Normalizando la imagen intensidad del histograma
1006 for(int i = 0; i < 256; i++)
1007 {
1008     for(int j = 0; j < 256; j++)
1009     {
1010         hist[i] = histImage[i];
1011     }
1012 }
1013
1014 //Normalizando la imagen intensidad del histograma
1015 for(int i = 0; i < 256; i++)
1016 {
1017     for(int j = 0; j < 256; j++)
1018     {
1019         hist[i] = histImage[i];
1020     }
1021 }
1022
1023 //Normalizando la imagen intensidad del histograma
1024 for(int i = 0; i < 256; i++)
1025 {
1026     for(int j = 0; j < 256; j++)
1027     {
1028         hist[i] = histImage[i];
1029     }
1030 }
1031
1032 //Normalizando la imagen intensidad del histograma
1033 for(int i = 0; i < 256; i++)
1034 {
1035     for(int j = 0; j < 256; j++)
1036     {
1037         hist[i] = histImage[i];
1038     }
1039 }
1040
1041 //Normalizando la imagen intensidad del histograma
1042 for(int i = 0; i < 256; i++)
1043 {
1044     for(int j = 0; j < 256; j++)
1045     {
1046         hist[i] = histImage[i];
1047     }
1048 }
1049
1050 //Normalizando la imagen intensidad del histograma
1051 for(int i = 0; i < 256; i++)
1052 {
1053     for(int j = 0; j < 256; j++)
1054     {
1055         hist[i] = histImage[i];
1056     }
1057 }
1058
1059 //Normalizando la imagen intensidad del histograma
1060 for(int i = 0; i < 256; i++)
1061 {
1062     for(int j = 0; j < 256; j++)
1063     {
1064         hist[i] = histImage[i];
1065     }
1066 }
1067
1068 //Normalizando la imagen intensidad del histograma
1069 for(int i = 0; i < 256; i++)
1070 {
1071     for(int j = 0; j < 256; j++)
1072     {
1073         hist[i] = histImage[i];
1074     }
1075 }
1076
1077 //Normalizando la imagen intensidad del histograma
1078 for(int i = 0; i < 256; i++)
1079 {
1080     for(int j = 0; j < 256; j++)
1081     {
1082         hist[i] = histImage[i];
1083     }
1084 }
1085
1086 //Normalizando la imagen intensidad del histograma
1087 for(int i = 0; i < 256; i++)
1088 {
1089     for(int j = 0; j < 256; j++)
1090     {
1091         hist[i] = histImage[i];
1092     }
1093 }
1094
1095 //Normalizando la imagen intensidad del histograma
1096 for(int i = 0; i < 256; i++)
1097 {
1098     for(int j = 0; j < 256; j++)
1099     {
1100         hist[i] = histImage[i];
1101     }
1102 }
1103
1104 //Normalizando la imagen intensidad del histograma
1105 for(int i = 0; i < 256; i++)
1106 {
1107     for(int j = 0; j < 256; j++)
1108     {
1109         hist[i] = histImage[i];
1110     }
1111 }
1112
1113 //Normalizando la imagen intensidad del histograma
1114 for(int i = 0; i < 256; i++)
1115 {
1116     for(int j = 0; j < 256; j++)
1117     {
1118         hist[i] = histImage[i];
1119     }
1120 }
1121
1122 //Normalizando la imagen intensidad del histograma
1123 for(int i = 0; i < 256;
```

- Reprobación automática a quien copie códigos de Internet y los reporte como suyos, además de una nota en su expediente con copia para el consejo de calidad



<https://github.com/naman14/AlgorithmVisualizer-Android>



“Finalmente son jóvenes que están en la preparatoria y que deben de leer su convocatoria con toda claridad, si no cumplen con los requisitos, si no pueden leer una convocatoria que dice tienes que traer número uno esto, número dos esto, número tres esto, no están listos para ser **estudiantes de educación superior**, así lo digo con toda claridad”.

Sara Ladrón de Guevara.

Rectora de la Universidad Veracruzana (2013-2017 y 2017-2021).

CONCLUSIÓN

