



DEPARTAMENTO DE INFORMÁTICA Y COMPUTACIÓN

---

## **TRABAJO N°1**

---

### **PROYECTO IMÁGENES**

### **COMPUTACIÓN PARALELA Y DISTRIBUIDA**

**1° SEMESTRE-2020**

---

**PROFESOR: SEBASTIÁN SALAZAR MOLINA**

**N° DE GRUPO: 9**

**RICARDO ABDÓN ALISTE GONZÁLEZ**

**DANIEL IGNACIO CAJAS ULLOA**

**RODRIGO ANTONIO CARMONA RAMÍREZ**

---

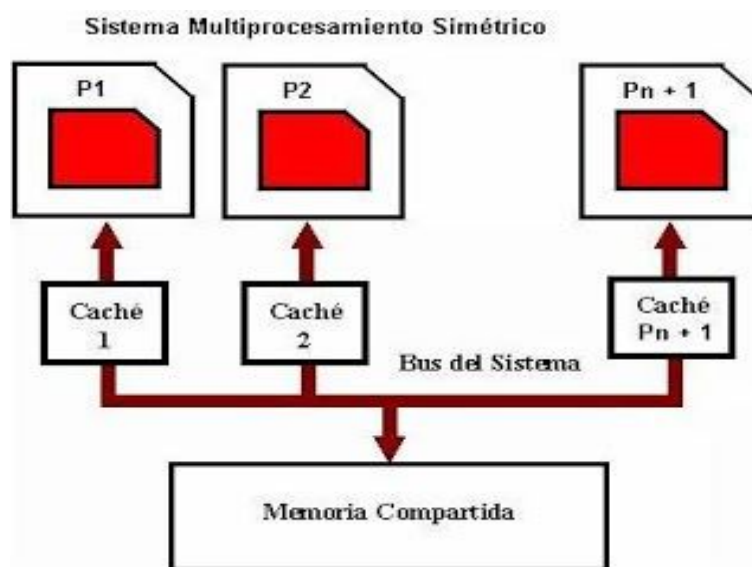
**12/8/2020**

## INTRODUCCIÓN

El **traspaso de información mediante mensajes** es un modelo de comunicación ampliamente usado en la computación paralela. En años recientes se han logrado desarrollar aplicaciones importantes basadas en este paradigma. Dichas aplicaciones han demostrado que es posible implementar sistemas **basados en el uso de mensajes** de una manera eficiente y portable. El crecimiento en el volumen y diversidad de tales aplicaciones originaron la necesidad de crear un estándar, es así como surge MPI o Message Passing Interface [1].

El contexto donde aparece este término es que el aumento exponencial de la capacidad de los procesadores no es suficiente para el estudio de muchos sistemas, es decir, el salto tecnológico no ha sido tan significativo para evitar casos donde es requerido usar varios procesadores para realizar una sola tarea. Es por ello que aparece un factor importante, la comunicación entre procesos. De esta forma, la computación paralela es capaz de conectar varios procesos bajo un mismo ecosistema.

Uno de estos métodos se le conoce como **"Sistemas de Memoria Compartida"**, los cuales están compuestos de procesadores y módulos de memoria interconectados, con un direccionamiento de memoria común para todos los procesadores. Por otro lado, se encuentran los **"Sistemas de Memoria Distribuida"**, donde cada procesador tiene su propia memoria y tiene un sistema de interconexión que permite acceder la memoria de los otros procesadores [2].



**Figura 1.1: Ilustración sobre las características del sistema de memoria compartida**

MPI es un estándar para la implementación de sistemas de uso de mensajes, diseñado por un grupo de investigadores de la industria y la academia para funcionar en una amplia variedad de computadores paralelos, de forma tal que los códigos sean portables. Su diseño está inspirado en máquinas con una arquitectura de memoria distribuida, sin embargo, hoy en día también se pueden encontrar implementaciones de MPI en máquinas de memoria compartida [1].

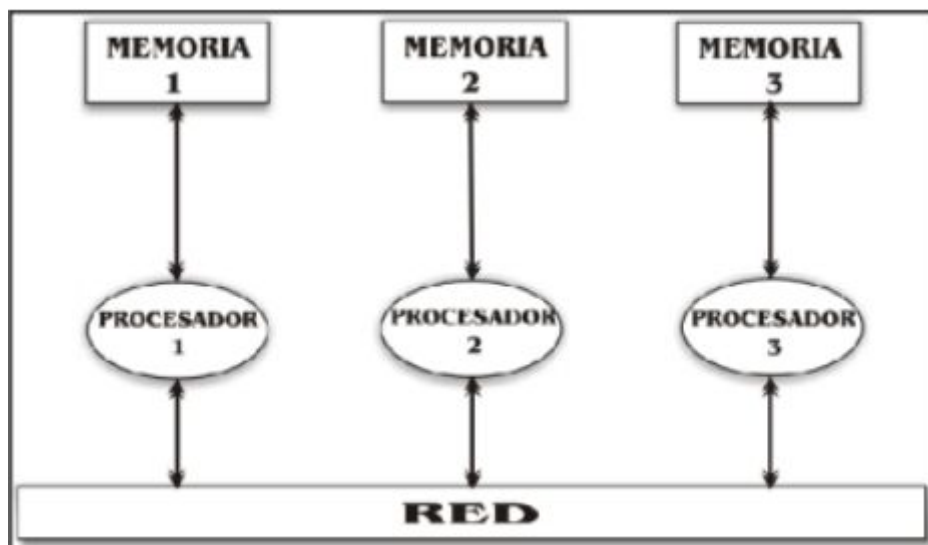


Figura 1.2: Ilustración sobre las características del sistema de memoria distribuida

El objetivo de MPI es desarrollar un estándar para ser ampliamente usado, **el cual permite escribir programas usando traspaso de mensajes**. Por lo tanto, la interfaz debería establecer un estándar práctico, portable, eficiente y flexible.

A continuación se presentan los objetivos de MPI:

- Diseñar una API (Application Programming Interface).
- Hacer eficiente la comunicación.
- Permitir que las aplicaciones puedan usarse en ambientes heterogéneos.
- Soportar conexiones de la interfaz con C y Fortran, con una semántica independiente del lenguaje.
- Proveer una interfaz de comunicación confiable.
- Definir una interfaz implementable en plataformas de diferentes proveedores sin tener que hacer cambios significativos.

Por otro lado, este proyecto se centra en el procesamiento de imágenes, cuyo objetivo es mejorar el aspecto de las imágenes y hacer más evidentes en ellas ciertos detalles que se desean hacer notar. El procesamiento de las imágenes se puede en general hacer por medio de métodos ópticos, como en este caso, en una computadora, debido a la complejidad y el número de cálculos necesarios para realizarlo.

En términos históricos, la utilización de imágenes radiográficas para diagnóstico clínico data prácticamente desde el descubrimiento de los rayos X en 1895 (Röntgen) [3], incluso las imágenes funcionales a partir de la emisión de fotones (rayos  $\gamma$ ) por parte de radionucleidos

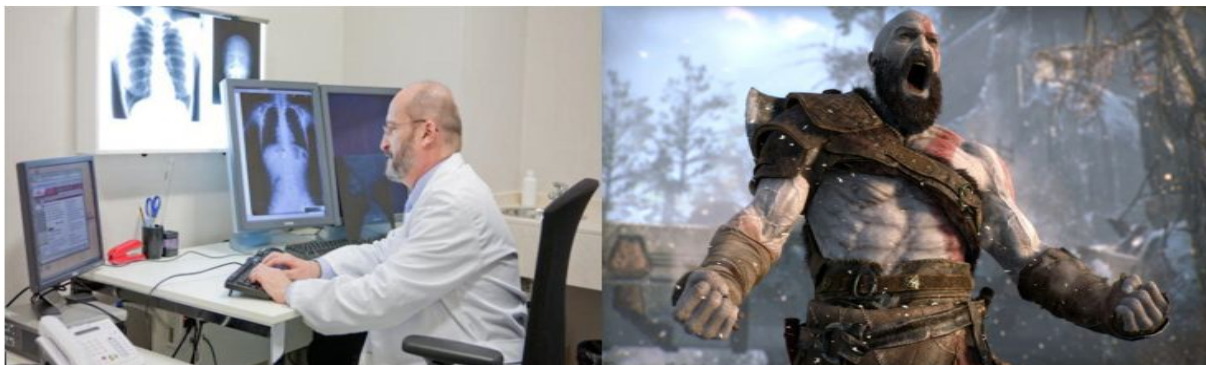
ya cuenta con más de 90 años de antigüedad (Heavesy & Seaborg, 1924) [3]. Sin embargo, en ese tiempo las imágenes eran adquiridas sobre films radiográficos, por lo que su correcto procesamiento no ha explotado su real potencialidad sino hasta la incorporación de la tecnología que permitió digitalizarlas.



**Figura 1.3: Registro de las primeras imágenes digitales existentes (1957).**

El motivo principal de esta “aparición tardía” del procesamiento de imágenes se debe a las limitaciones de los recursos que se utilizaban para estudiar esta clase de fenómenos, los requerimientos de hardware tanto para el procesamiento de las mismas como para la representación de estas en sistemas gráficos era demasiado alto en anteriores épocas. De forma paralela a este desarrollo, la formulación de algoritmos para el procesamiento ha seguido los avances tecnológicos logrando un alto grado de sofisticación y manipulación de imágenes en tiempo real.

En años recientes, el procesamiento digital de imágenes ha adquirido un papel importante en las tecnologías de la información, así como en cómputo. Actualmente, es la base de una creciente variedad de aplicaciones que incluyen análisis y diagnósticos médicos, percepción remota, exploración espacial, visión por computadora, reconocimiento de patrones, visión robótica/artificial, mejora de imágenes, filtros, animación, etc [4].



**Figura 1.4: Ejemplos de aplicaciones actuales del procesamiento de imágenes**

Existe un sinfín de elementos que podemos incorporar y utilizar en una imagen, ya sea para que se vea mejor o bien por la necesidad de añadir un filtro que nos permita utilizar esa

fotografía sin inconvenientes. Dentro de este proyecto, se concentrará en las siguientes tareas de edición de imágenes:

- Difuminación: Es un efecto que elimina el énfasis de los detalles concretos y, en cambio, transmite el matiz general de una foto. En otras palabras, la imagen pierde claridad, nitidez o intensidad. Por ejemplo, se puede ocultar un objeto o una persona que aparece en la fotografía, desvanecer las líneas o los colores de un dibujo, etc [5].

Un ejemplo de esto, es el difuminado Gaussiano, el cual se basa en la aplicando de una máscara circunstancial, donde los píxeles más cercanos al píxel actual son considerados mediante su peso en la imagen, alterando así el valor del píxel actual [11].



**Figura 1.5: Ejemplo de una difuminación al fondo de una imagen**

- Escala de grises: Es la representación de una imagen en la que cada píxel se dibuja usando un valor numérico individual que representa su luminancia, en una escala que se extiende entre blanco y negro [6].



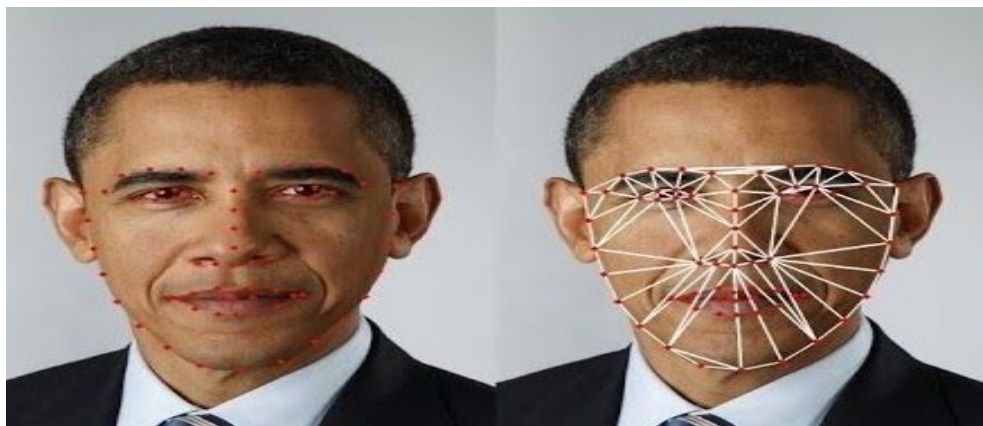
**Figura 1.6: Ejemplo de aplicación de una escala de grises a una fotografía**

- Escalar una imagen: Esto permite agrandar o reducir el tamaño físico de la imagen cambiando el número de píxeles que contiene. Cambia el tamaño del contenido de la imagen y redimensiona el lienzo en consonancia [7].

## RESOLUCIÓN DEL PROBLEMA

El proyecto fue programado en el lenguaje C++, utilizando la librería OpenCV. Esta proporciona un marco de trabajo de alto nivel para el desarrollo de aplicaciones de visión por computador en tiempo real: estructuras de datos, procesamiento y análisis de imágenes, análisis estructural. Para ello dispone de varios algoritmos que permiten clasificar objetos, identificar rostros, entre otros. Se decidió trabajar con OpenCV al ser compatible con una buena cantidad de lenguajes y una librería multiplataforma para los sistemas operativos de Windows, Mac, Linux y Android [8].

Entre las ventajas de C++ se destaca la robustez en la creación de sistemas complejos y el alto rendimiento que ofrece. Es un lenguaje de programación orientado a objetos, posee gran variedad de parámetros de optimización y se integra de forma directa con el lenguaje ensamblador. Por otro lado, OpenCV es una librería utilizada en el reconocimiento de objetos, como redes neuronales, soporte de máquinas vectoriales y tratamiento de imágenes digitales, es decir, se puede trabajar en múltiples aplicaciones de diversos sectores, esto le otorga a esta biblioteca altos niveles de versatilidad.



**Figura 2.1: Ejemplo de aplicación de la librería OpenCV**

La estrategia para abordar el problema es el siguiente:

- 1.- Investigar sobre el procesamiento de imágenes, y observar algunos algoritmos encontrados en la red para tratar de llevar a cabo la implementación del código del proyecto.
- 2.- Se desarrolla una versión del proyecto donde se toma en cuenta los argumentos/parámetros de cada función, y se definen las opciones sobre las acciones que realiza el archivo, con una ejecución secuencial.



3.- Posteriormente, se realiza la implementación de procesos MPI. Tras analizar el código escrito en ese entonces, había un problema donde era necesario fragmentar la imagen, donde lo más sencillo fue cortar la foto en sectores. Debido a la situación, se decidió trabajar con la cantidad de procesadores activos para ejecutar la fragmentación de la imagen ingresada al sistema.

4.- Crear una función de apoyo que permita agrupar las partes separadas de la fotografía, cuyo propósito es contrastar la división del punto anterior, esto se puede interpretar como la parte donde se manejan las hebras o hilos para llevar a cabo la transformación de la imagen seleccionada (la unidad de procesamiento ejecuta una misma tarea en procesos más pequeños).

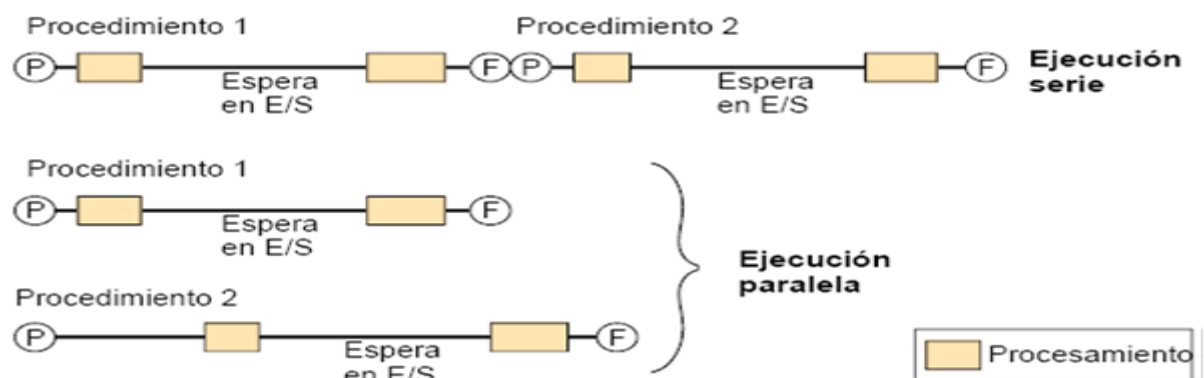


Figura 2.2: Proceso de Ejecución en serie vs Ejecución paralela

5.- En búsqueda de información complementaria, se realiza una indagación por internet sobre el difuminado gaussiano (un efecto gráfico de borrosidad para mapas de bits generado por computadora), un algoritmo para la escala de grises, donde se calcula una media entre los colores que componen la imagen [9], y la interpolación bilineal, que consiste en realizar una interpolación lineal, a 2 interpolaciones lineales, obtenidas en función de las coordenadas de bit [10]. Estos datos sirvieron para el algoritmo de escalamiento de la fotografía.



Figura 2.3: Ejemplo de aplicación del difuminado o desenfoque gaussiano

6.- Dado que se desarrollaron varias versiones del proyecto, se realizó una fase de pruebas donde el programa definitivo es aquel que presente el rendimiento más convincente (en términos de la calidad de la imagen obtenida) dentro del grupo de trabajo. Cabe recordar

que el archivo de fotografías procesadas debe correr nativamente en el sistema operativo Ubuntu 20.04 LTS.

7.- Hacer pruebas y testeos del código escrito en cada una de las versiones creadas durante el desarrollo del proyecto, corrigiendo los errores que se presenten durante la ejecución del código (hipotéticamente hablando).

8.- Se implementa la versión definitiva de las funciones que efectúan el escalado, la difuminación y la escala de grises de cualquier imagen.

9.- Escritura de la función principal (main) del sistema, donde se establecen todas las variables involucradas, el menú de opciones y las restricciones generales de este software.

10.- Mejorar el algoritmo de escalamiento y los hilos que se utilizan en la operación de procesamiento de imágenes.



**Figura 2.3: Ejemplo de una Imagen comparada con su versión escalada**

11.- Ejecutar una modificación del algoritmo de difuminación con tal de hacer mejoras en la transformación de la imagen al momento de ser procesada por el programa.

11.- Hacer pruebas y testeos de la versión definitiva del proyecto, corrigiendo los errores que se presenten durante la ejecución del programa (hipotéticamente hablando), teniendo en mente que el procesamiento y posterior transformación de imágenes se cumpla de forma efectiva.

12.- Realizar la documentación correspondiente de la versión definitiva del código escrito en el archivo del lenguaje C++.



## CONCLUSIÓN

Los avances que se han descubierto con el procesamiento de una imagen son muy amplios, como el uso médico mediante radiografías y resonancias magnéticas, campañas de publicidad, en la realización de películas cinematográficas, entre otras actividades. Dadas las libertades de software, es posible realizar varios tipos de procesamiento que van más allá de los tres casos presentados en el informe. Esta evolución ha sido posible gracias a que la imagen digital es, por naturaleza, económica, ya que es de fácil duplicación, y puede ser de acceso simultáneo, es decir, no gasta materiales en su producción y no hay que crear nuevos recursos de hardware que los que generalmente existen, Su gran valor desde el punto de vista académico está dado por la capacidad de resaltar un gran número de detalles, lo que hace que su estudio sea bastante interesante desde una perspectiva educacional.

El sistema desarrollado se basa en la transformación de una imagen mediante una serie de operaciones que están relacionadas particularmente en el estudio de fenómenos físicos y tecnológicos como la luz, la visión artificial, la manipulación de datos (en este caso píxeles) o la interpolación de parámetros. Se debe tener en cuenta cómo se construye la disposición e intensidad de los filtros de suavización, el aumento de resolución o escalado, entre otros factores, pues estos influyen en el procesamiento de la imagen.

Es importante destacar el gran soporte que ofrece la librería MPI para las comunicaciones entre los ordenadores del clúster, dado que es complicado no pensar en llevar a cabo la implementación en serie de todo lo que conforma el sistema, porque no es lo mismo realizar una aplicación pensando desde el primer momento en paralelo, que intentar paralelizar una aplicación que resulta más “sencillo” hacer de forma secuencial.

La utilización de OpenCV también ha sido muy importante, aunque se necesitó dedicar una gran cantidad de tiempo para analizar las características que ofrece la librería y poder interiorizar los temas necesarios para implementar las funciones de procesamiento de imágenes, dadas las amplias funcionalidades que proporciona. Se destaca su aporte en la extracción de características de la imagen, el proceso de filtrado, los retoques fotográficos que se realizaron, etc, donde se abarcaron diferentes conceptos y definiciones.

## BIBLIOGRAFÍA

[1] F. Hidrobo y H. Hoeger, 12 Marzo 2015. [En línea]. Available: [http://132.248.182.159/Herramientas/ComputoParalelo/MPI/Introduccion\\_MPI.pdf](http://132.248.182.159/Herramientas/ComputoParalelo/MPI/Introduccion_MPI.pdf). [Último acceso: 17 Julio 2020].

[2] X. Andrade, «Grupo de Geofísica Computacional UNAM,» [En línea]. Available: <http://www.mmc.geofisica.unam.mx/cursos/femp/HPC/MPI/MpiBasico1.pdf> [Último acceso: 17 Julio 2020].

[3] P. Pérez, «FANAF, Universidad de Córdoba,» 2018. [En línea]. Available: <https://www.famaf.unc.edu.ar/~pperez1/manuales/cim/cap2.html>. [Último acceso: 17 Julio 2020].

[4] A. C. Nieto, «Repositorio Institucional UAEM,» 7 Junio 2018. [En línea]. Available: <http://ri.uaemex.mx/handle/20.500.11799/94519>. [Último acceso: 17 Julio 2020].

[5] The Free Dictionary, [En línea]. Available: <https://es.thefreedictionary.com/difuminar>. [Último acceso: 17 Julio 2020].

[6] FotoNostra, [En línea]. Available: <https://www.fotonostra.com/glosario/escalagrisesscale.htm>. [Último acceso: 17 Julio 2020].

[7] The GIMP Documentation, [En línea]. Available: <https://docs.gimp.org/es/gimp-image-scale.html>. [Último acceso: 17 Julio 2020].

[8] Robologs, [En línea]. Available: <https://robologs.net/tutoriales/tutoriales-opencv/>. [Último acceso: 17 Julio 2020].

[9] J. Bekios, «<http://jbekios.blogspot.com/>,» 2 Septiembre 2011. [En línea]. Available: <http://jbekios.blogspot.com/2011/09/tips-tres-algoritmos-para-convertir-una.html>. [Último acceso: 17 Julio 2020].

[10] A. Roncero, «Repositori UJI,» Octubre 2019. [En línea]. Available: [http://repositori.uji.es/xmlui/bitstream/handle/10234/185159/TFG\\_AlbaRonceroGarcia\\_%20%281%29.pdf?sequence=1&isAllowed=y](http://repositori.uji.es/xmlui/bitstream/handle/10234/185159/TFG_AlbaRonceroGarcia_%20%281%29.pdf?sequence=1&isAllowed=y). [Último acceso: 17 Julio 2020].

[11] N. Aguirre, «Biblioteca de Ingeniería Universidad de Sevilla,» [En línea]. Available: [http://bibing.us.es/proyectos/abreproy/12112/fichero/Documento\\_por\\_capitulos%252F3\\_Cap%C3%ADtulo\\_3.pdf](http://bibing.us.es/proyectos/abreproy/12112/fichero/Documento_por_capitulos%252F3_Cap%C3%ADtulo_3.pdf). [Último acceso: 17 Julio 2020].

#### INFORMACIÓN ADICIONAL

- [https://es.qwe.wiki/wiki/Gaussian\\_blur](https://es.qwe.wiki/wiki/Gaussian_blur)
- [https://en.wikipedia.org/wiki/Gaussian\\_blur](https://en.wikipedia.org/wiki/Gaussian_blur)