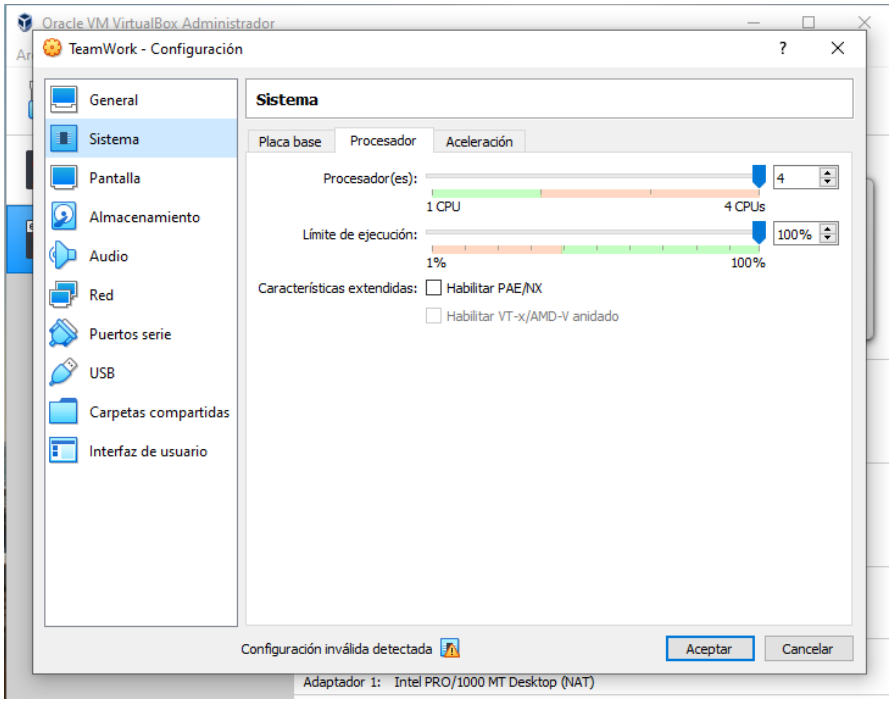
Fase 1: Medición de tiempo en la conversión de una imagen a escala de grises.

Configuración inicial de la máquina virtual para la realización del trabajo:



Una vez configurada, comenzamos las modificaciones en el fichero *main.cpp*:

Primero, almacenando las componentes de los píxeles de la imagen usando el tipo float**.**

```
// Data type for image components
typedef float data_t;
```

Posteriormente, creamos los punteros a la imagen a modificar y a la imagen modificada. Estas fueron, la imagen *uniovi_2.bmp* y, a partir de ella, la imagen *uniovi_2_final.bmp*.

```
// Pointers to the source image and the destination image
const char* SOURCE_IMG      = "/home/student/Pictures/normal/uniovi_2.bmp";
const char* DESTINATION_IMG = "/home/student/Pictures/uniovi_2_final.bmp";
```

Método *main()* de la clase:

Inicialización de las variables necesarias, obteniendo así la información de los componentes de la imagen original, para realizar la modificación de la imagen en nuestro algoritmo.

Realizado por: Alejandro Álvarez Solís, Carlos Concheso Cubillas, María Teresa Fernández Coro y Pablo Alonso Alonso.

```
/*************************************************
 * Variables initialization.
 *    - Prepare variables for the algorithm
 *    - This is not included in the benchmark time
 */
srcImage.display(); // Displays the source image
width  = srcImage.width(); // Getting information from the source image
height = srcImage.height();
nComp  = srcImage.spectrum();
            // Common values for spectrum (number of image components):
            //  B&W images = 1
            //  Normal color images = 3 (RGB)
            //  Special color images = 4 (RGB and alpha/transparency channel)
```

Comienzo de la medición de tiempo:

Tomamos el valor del tiempo al comienzo de la ejecución del algoritmo.

```
// Strat time
if(clock_gettime(CLOCK_REALTIME, &tStart) == -1) {
    printf("ERROR: clock_gettime: %d, \n", errno);
    exit(EXIT_FAILURE);
}
```

Algoritmo de la medición de tiempo. Es necesario realizar un bucle para poder obtener resultados significativos.

El resultado deseado es la modificación de una imagen a color a escala de grises. La imagen final, por tanto, debería tener solo el componente de la luminosidad (L).

```
/*************************************************
 * In this example, the algorithm is a components swap
 *
 * TO BE REPLACED BY YOUR ALGORITHM
 */

// Using nComp = 1 for B/W images
nComp = 1;

// Time measurement
// Making a loop so we obtain significant results
for(int j = 0; j < 300; j++) {
    for (long i =0; i < width * height; i++) {
        L = 0.3 * pRsrc[i] + 0.59 * pGsrc[i] + 0.11 * pBsrc[i];
        L = 255 - L;
        pLdest[i] = L;
    }
}
```

Por último, tomamos el valor del tiempo al finalizar la ejecución del algoritmo y calculamos el tiempo transcurrido en la ejecución del mismo.

```
// End time
if(clock_gettime(CLOCK_REALTIME, &tEnd) == -1) {
    printf("ERROR: clock_gettime: %d, \n", errno);
    exit(EXIT_FAILURE);
}

// Calculate the elapsed time in the execution of the algorithm
dElapsedTimeS = (tEnd.tv_sec - tStart.tv_sec);
dElapsedTimeS += (tEnd.tv_nsec - tStart.tv_nsec) / 1e+9;
```

Realizado por: Alejandro Álvarez Solís, Carlos Concheso Cubillas, María Teresa Fernández Coro y Pablo Alonso Alonso.

Tabla con mediciones realizadas y estadísticas:

Elapsed time											
6,497057	6,084999	6,200696	6,185552	6,204259	6,203025	6,381512	6,433590	6,566542	6,596798	6,296360	6,478689
Media		Desviación típica		Intervalo de confianza (95%) inferior				Intervalo de confianza (95%) superior			
6,339908388		0,170247722		5,999412945				6,680403832			

Estimación del porcentaje del proyecto que representa el trabajo realizado por cada miembro: Alejandro Álvarez 40%, Carlos Concheso 20%, María Teresa Fernández 20% y Pablo Alonso 20%.