# Universidad de Costa Rica

# Ciencias de Computación e Informática

# CI-1201 Programación II

# Profesor Adolfo Di Mare

# Listado de [fotos repetidas](http://soygcm.github.io/Programacion2/Tarea6/html/fotosRepetidas_8cpp_source.html) con List y Map

# [Gabriel Castañaza B31520](https://github.com/soygcm/)

# gcmelmejor@gmail.com

## Introducción

Tal y como cita en el enunciado de la tarea, tenemos que crear un programa, capaz de trabajar con dos carpetas, una donde estén las fotos de la cámara y otra donde guardamos una copia de esas fotos. Obtener dos listas con los archivos en cada carpeta y luego compararlas de tal forma que se “obtenga 3 grupos: los que aparecen en ambos, los que aparecen solo en la primera carpeta y los que solo aparecen en la segunda. Para obtener los nombres de archivos use la bilioteca **'readDir'** vista en clase y las listas **std::list<>** de la biblioteca estándar de C++” (Adolfo Di Mare)

Pero también hay que tomar en cuenta “que algunas fotos hayan sido **renombradas**, separe en 2 partes el nombre de cada foto. Por ejemplo, el nombre "dsc00200 Dele Beso.jpg" quedaría separado en [ 'dsc00200' | "Dele Beso.jpg" ] de manera que la comparación de nombres de carpeta se pueda hacer utilizanado 'dsc00200' y desechando "Dele Beso.jpg". Recuerde ignorar las diferencias entre **mayúsculas y minúsculas** en los nombres de archivo” (Adolfo Di Mare)

## Descripción del problema

Como hemos visto, es necesario usar una biblioteca para leer una carpeta. Que es la parte mas importante de la aplicación. Luego lo que hay que hacer es la comparación usando las listas de la biblioteca estándar.

## Librería readDir

Resulta que esta librería usa otra <io.h> la cual no existe en el sistema operativo Mac OS

Así que el primer problema, es buscar como leer una carpeta usando otra alternativa.

La alternativa encontrada es [<dirent.h>](http://pubs.opengroup.org/onlinepubs/7908799/xsh/dirent.h.html)

Que por medio de estos dos métodos, se puede acceder un directorio y luego a cada uno de sus elementos.

* DIR \*opendir(const char \*);
* struct dirent \*readdir(DIR \*);

Por supuesto, hay que tomar en cuenta que las carpetas son en realidad arboles, asi que deben ser recorridos como arboles, en este caso se escogió el algoritmo de profundidad primero. Que hace uso de la recursividad.

## Comparación de Listas

Luego de leer las carpetas y guardar sus archivos (solamente los que terminan en “jpg”)

Se procede a realizar una comparación entre las dos listas.

Esta tarea se hace por medio de iteradores, simplemente recorriendo ambas listas en un ciclo for anidado dentro de otro.

Esto hay que repetirlo dos veces, una vez con la primera lista en el ciclo externo y en la segunda vez con la segunda lista en el ciclo externo.

Además en el primer ciclo hay que identificar los archivos que están en ambas listas.

## Comparación de carpetas usando Mapas

Tanto como para guardar, como para comparar las carpetas también se uso Mapas, para ello es necesario recordar, que se debe definir un tipo par:

* typedef pair<T, T> name;

El cual sirve, para insertar un par de valores en el mapa. Por otro lado la iteración es similar a la lista. Con la diferencia que cada iterador tiene dos punteros uno llamado first y otro second, que apuntan a el primer valor del par y al segundo correspondientemente.

Todo esto puede usted verlo aquí en la documentación

[**Documentación**](http://soygcm.github.io/Programacion2/Tarea6/html/)

[**Aquí también esta con Mapas**](http://soygcm.github.io/Programacion2/Tarea6/html/fotosRepetidasMap_8cpp_source.html)