

Práctica 3 - Programación 2 - Sección 3 - B2013

Preparador: Donato Galo

10 de junio de 2014

1. Práctica sobre manejo de archivos

En esta práctica pondremos a prueba los conocimientos adquiridos sobre el manejo de archivos.

Los registros que vamos a guardar en los archivos tienen un *id*, un *nombre* y un *apellido*. Un registro se vería de la siguiente forma:

5,Maria,Perez

El *id* debe ser único, ya que nos servirá para realizar búsquedas de registros.

2. Código fuente

Adjunto a este documento está un código fuente en C++, a partir del cual puede desarrollar la práctica.

En el código se define la estructura *TRegister*, la cual representa cada registro para ser guardado en un archivo.

```
struct TRegister {  
    int id;  
    string name;  
    string lastName;  
};  
  
typedef struct TRegister Register;
```

A esta estructura le asignamos el nombre alternativo *Register*, para facilitar el uso de la estructura como un tipo de dato.

También está implementada la sobrecarga del operador << para un *Register*:

```
ostream & operator<<(ostream &out, const Register &reg) {  
    return out << reg.id << ", " //  
               << reg.name << ", " //  
               << reg.lastName;  
}
```

Puede compilar el programa utilizando el *Makefile* adjunto.

3. Requisitos de la Práctica

3.1. Generar registros aleatorios

A partir de los arreglos *names* y *lastNames* genere aleatoriamente tantos registros como se indique en la constante *NUM_REG*.

```
#define NUM_REG 50
```

```
string names[] = {"Luisa", "Jose", "Maria", "Juana", "Simon", "Manuel", "Carolina", "Ingrid", "Andres", "Daniela"};

string lastNames[] = {"Torres", "Perez", "Barrios", "Rodriguez", "Bolívar", "Salcedo", "Mora", "Lopez", "Blanco", "Araujo"};
```

Para crear los registros genere un entero para el *id*, seleccione aleatoriamente un elemento del arreglo *names*, otro de *lastNames*, guardelos en un *Register* e inserte el registro en un archivo.

Para generar números enteros aleatorios puede utilizar el siguiente código:

```
int r = lim_inf + rand() % (lim_sup + 1);
```

3.2. Buscar registro

Implemente la función:

```
Register searchRegister(istream &file, const int &id);
```

La cual devuelve el resultado de buscar en el archivo *file* el registro que tiene el *id* dado por parámetros.

3.3. Modificar registro

Implemente la función:

```
void modifyRegister(fstream &file, const int &id);
```

La cual modifica el registro que tiene el *id* dado por parámetros en el archivo *file*.