

Registros de Longitud Fija

Profesor: Heider Sanchez

ACLs: Mariana Capuñay, Sebastián Loza, Jouse Arroyo

P1 (10 pts):

Dada la siguiente estructura del registro, en donde se encuentran dos atributos no textuales: ciclo y costo de mensualidad.

Listing 1: Estructura Alumno

```
struct Alumno {  
    char codigo[5];  
    char nombre[11];  
    char apellidos[20];  
    char carrera[15];  
    int ciclo;  
    float mensualidad;  
};
```

Se le pide encapsular las siguientes operaciones de **manipulación de archivo binario** con dos estrategias de eliminación en una clase llamada **FixedRecord**:

- a) El constructor reciba el nombre del archivo y el modo de eliminación:
 1. **MOVE_THE_LAST**: mueve el último registro a la posición del registro eliminado
 2. **FREE_LIST**: mantiene una lista de espacios libres, para ser usado por nuevas inserciones.
- b) Implemente la función `vector<Alumno> load()` para leer todos los registros del archivo.
- c) Implemente la función `void add(Alumno record)` para agregar un nuevo registro al archivo.
- d) Implemente la función `Alumno readRecord(int pos)` para obtener el registro de la posición "pos".
- e) Implemente la función `bool delete(int pos)` para eliminar el registro de la posición "pos".
- f) Realice pruebas de cada función en el programa principal (`p1.cpp`).

P2 (4 pts):

Asuma que tiene un **archivo de texto de registros de longitud variable**. En donde, cada registro contiene cuatro campos de tamaño variable, los campos están delimitados por el caracter |, y los registros están delimitados por el salto de línea \n. La primera línea del archivo indica los nombres de cada campo por lo tanto ignore esta línea en la lectura.

Listing 2: Estructura Alumno

```
struct Alumno{  
    string Nombre;  
    string Apellidos;  
    string Carrera;  
    float mensualidad;  
};
```

Ejemplo de registros para la estructura Alumno

Nombre	Apellidos	Carrera	Mensualidad
Howard	Paredes Zegarra	Computación	1500.00
Penny	Vargas Cordero	Industrial	2550.50
Sheldon	Cooper Quizpe	Mecatrónica	1850.00

Encapsular las siguientes operaciones de manipulación del archivo en una clase llamada **VariableRecord**:

- El constructor reciba el nombre del archivo.
- Implemente la función `vector<Alumno> load()` para leer todos los registros del archivo.
- Implemente la función `void add(Alumno record)` para agregar un nuevo registro al archivo.
- Implemente la función `Alumno readRecord(int pos)` para obtener el registro de la posición "pos".
- Realice pruebas de cada función en el programa principal (`p2.cpp`).

P3 (6 pts):

Implemente otro programa para leer y escribir registros de longitud variable en un **archivo binario usando el tamaño del dato como separador**. El registro debe tener los campos que se indican en la siguiente figura.

Listing 3: Estructura Matrícula

```
struct Matricula{  
    string codigo;  
    int ciclo;  
    float mensualidad;  
    string observaciones;  
};
```

Se le pide implementar la estructura adecuada para acceder directamente a los registros considerando lo siguiente:

- a) Maneje un archivo adicional (metadata) para indicar la posición inicial de cada registro. Evalúe si es necesario guardar también el tamaño del registro.
- b) Implemente adecuadamente la función `vector<Matricula> load()` para leer todos los registros del archivo.
- c) Implemente adecuadamente la función `void add(Matricula record)` para agregar un nuevo registro al archivo.
- d) Implemente la función `Matricula readRecord(int pos)` para obtener el registro de la posición "`pos`".
- e) Realice pruebas de cada función en el programa principal (`p3.cpp`).

Entregable:

- Subir al Canvas el código fuente de la solución compilable (`p1.cpp`, `p2.cpp`, `p3.cpp`).
- Los cuatro programas deben contener pruebas de funcionalidad en el `main` (debe correr).
- Compile con C++ versión 2017.