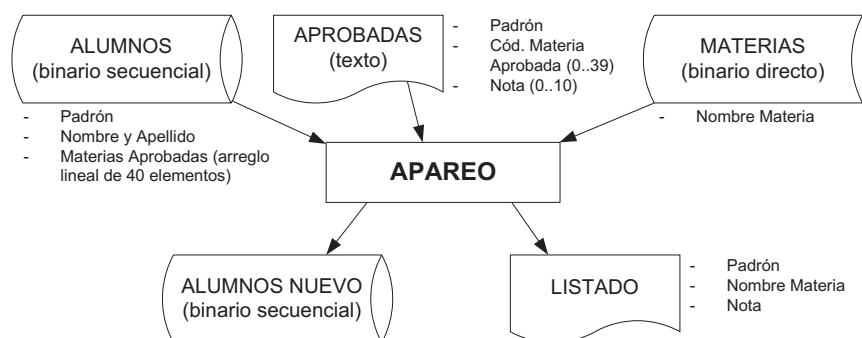


75.40 Algoritmos y Programación I – Segundo Parcial Integrador
Cátedra: Ing. Pablo Guarna
27 de febrero de 2007

Aclaración: Para la aprobación del examen, deberá demostrarse conocimientos de todos los ejercicios.

1. Archivos y Estructuras de Datos (7 pts)

Desarrollar un programa Pascal que realice el siguiente apareo:



El archivo APROBADAS no está ordenado, y sus líneas tienen la siguiente estructura (el caracter # es el separador de campos):

```
75925#3#8
75025#27#10
...
```

Para leer y parsear¹ las líneas del archivo APROBADAS, deberá construirse una función que respete el siguiente prototipo:

```
function leer_linea(var aprobadas: text): ^t_aprobada;
```

La función `leer_linea()` lee y parsea la siguiente línea, y retorna un puntero a un registro en **memoria dinámica**, o NIL si alcanzó el fin de archivo. Dicho registro tiene la siguiente estructura:

```
t_aprobada = record
    padron:    longint;
    cod_mat:   integer;
    nota:      byte;
end;
```

Considerar que, en el archivo MATERIAS, el *código de materia* coincide con la posición del registro en el archivo. Los códigos de materia son números enteros, entre 0 y 39 (inclusive).

Los archivos ALUMNOS y ALUMNOS NUEVO, y el listado final, están ordenados por *padrón*, ascendente-mente.

Aclaraciones: El programa desarrollado debe estar completo, incluyendo tipos de datos, constantes, cuerpo principal, etc. Utilizar nombres representativos, y comentar aquellos fragmentos que, por su complejidad, lo requieran. Modularizar adecuadamente. Tener en cuenta las precauciones que deben tomarse por el uso de memoria dinámica. Sea prolijo (*el código que no se entiende se considera incorrecto*).

¹“Parsear” significa extraer los campos de la línea de texto, y convertirlos a su formato numérico correspondiente.

2. Seguimiento (3 pts)

Explique con detalle qué tarea realiza el siguiente procedimiento. Dar un ejemplo de funcionamiento, para un conjunto de parámetros $p1$ y $p2$ dados.

```
procedure incognita(p1, p2: ^char);

var p3: ^char;
    a: char;

begin
  p3 := p1 + 1;
  while ( p3 <= p2 ) do
    begin
      if ( p1^ > p3^ ) then
        begin
          a := p1^;
          p1^ := p3^;
          p3^ := a;
        end;
      inc( p3 );
    end;

    if ( p1 <> p2 ) then
      incognita( p1+1 , p2 );
  end;
```