

APAREO - Actualización de Archivos

Actualizar significa incorporar, modificar o eliminar información de un archivo mayor y se utilizan como mínimo dos ficheros ,uno de entrada y uno de salida.

La actualización de archivos es un proceso que dependiendo del tipo de organización que tengan estos archivos, el algoritmo adoptará una estructura particular.

Este proceso es muy empleado para la **actualización del maestro** a través del archivo de novedades. Las novedades tendrán que ver con **altas, bajas o modificaciones** o algunas de ellas solamente.

Normalmente existen *dos archivos de entrada*.

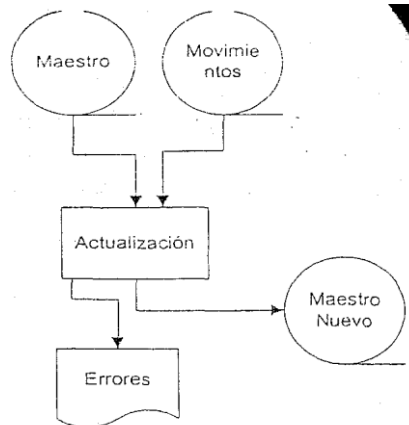
- **Maestro:** es el archivo *de estructura mayor* y es el que *va a sufrir la actualización*. Por su periodicidad *puede ser*
 - *Histórico:* contiene información desde *el nacimiento de la entidad*.
 - *Común:* es una partición *del histórico*, conteniendo información *de uso cotidiano*.
- **Movimiento o Novedades:** Es un archivo de estructura *menor o igual a la del maestro* y *debe contener como mínimo el campo clave de apareo*. para realizar la actualización.
 - **Altas:** incorporación *de un nuevo registro al fichero maestro*, manteniendo **la** *unicidad de claves*, en general.
 - **Modificaciones:** consisten en cambiar todo o parte *de los tipos contenidos del registro del maestro*, a excepción *del campo clave*
 - **Bajas:** consisten en la eliminación física o lógica *de un registro del maestro*.
Físicas: consisten en eliminar totalmente el contenido y *clave del registro maestro*.
Lógicas consisten en señalar al registro como eliminado, pero sin eliminarlo, para lo cual debe agregarse un campo al registro **del** maestro llamado código de estado

Si ambos archivos poseen una organización secuencial, entonces ambos archivos deben encontrarse ordenados –ascendente o descendente- por medio del valor de una clave en común. El resultado de este proceso será un nuevo archivo de salida con la misma estructura que el maestro a actualizar, siendo este archivo el maestro

actualizado. El archivo de novedades tiene la misma estructura que el maestro pero con un campo más, el cual indica el *código de movimiento*. Cada archivo trabajará con sus propios registros, es decir, un registro para el archivo maestro viejo, otro para el archivo de maestro nuevo y un registro para el archivo de novedades.

Las situaciones de errores por alta existente o bajas o modificaciones inexistentes, se emitirán por medio del dispositivo de la impresora.

Este proceso que genera un nuevo archivo –el maestro actualizado- se denomina **Maestro Nuevo**.



Tipos de Actualización

Por lotes: Implica que por cada registro del maestro existen varios registros en el archivo de movimiento.

Unitaria: Por cada registro del maestro hay un único registro de movimiento.

Algoritmo de Actualización Unitaria

```

Acción Act_Unit es
  Abrir_Archivos;
  Leer_Maestro;
  Leer Movimiento;
  Mientras (clave_mae <> High Value) ∨ (clave_mov <> High Value) hacer
    Si clave_mae = clave_mov entonces
      Procesos_iguales
    sino
      Si clave_mae < clave_mov entonces
        reg_sal := reg_mae;
        Escribir (Arch_sal, reg_sal);
        Leer_Maestro;
      sino
        Procesos_distintos;
      Fsi
    Fsi
  Fmientras
  Cerrar_Archivos;
Facción
  
```

```

Subacción Leer_Maestro es
  Leer (Arch_mae, reg_mae);
  Si FDA (Arch_mae) entonces
    Clave_mae := High value
  Fsi
Fsubacción

```

```

Subacción Leer_Movimiento es
  Leer (Arch_mov, reg_mov);
  Si FDA (Arch_mov) entonces
    Clave_mov := High value
  Fsi
Fsubacción

```

```

Subacción Procesos_iguales es
  Si cod_mov = alta entonces
    Escribir ('Error alta no existe');
    Reg_sal := reg_mae;
    Escribir (Arch_sal, reg_sal);
  sino
    Si cod_mov = modificación entonces
      Proceso_modificar_maestro;
      Reg_sal := reg_mae;
      Escribir (Arch_sal, reg_sal);
    sino
      Marcar_registro;
      Reg_sal := reg_mae;
      Escribir (Arch_sal, reg_sal);
    Fsi
  Fsi
  Leer_maestro;
  Leer_movimiento;
Fsubacción

```

```

Subacción Procesos_distintos es
  Si cod_mov = baja entonces
    Escribir ('Error baja no existe');
  sino
    Si cod_mov = modificación entonces
      Escribir ('Error modificación no existe')
    sino
      Reg_sal := reg_mov;
      Escribir (Arch_sal, reg_sal);
    Fsi
  Fsi
  Leer_movimiento;
Fsubacción

```

Algoritmo de Actualización por Lotes

