

```
##### Apareo de Archivos #####
```

```
# Genera un maestro actualizado, en base al archivo maestro original y las  
# novedades que vienen informadas en el archivo de novedades.  
# En este caso, en el archivo de Novedades, pueden venir registros duplicados.
```

```
# Para esta solucion se utiliza el metodo de clave maxima, por el cual se  
# define un valor, que nunca debe ser alcanzado por las claves que se  
# encuentran en el archivo.
```

```
# La funcion leer_MaeNov utiliza el parametro "devolver", para que cuando sea  
# utilizada para leer el archivo Maestro, devuelva 3 valores cuando encuentre EOF;  
# y en el caso del archivo Novedades, devuelva 2 valores.
```

```
def leer_MaeNov(archivo, devolver):  
    linea = archivo.readline()  
    linea = linea.strip('\n')  
    if not linea:  
        linea = devolver  
    return linea.split(',')
```

```
def grabar_MaeActualizado(archivo, cod_art, desc_art, stock_actual):  
    archivo.write(cod_art + ',' + desc_art + ',' + stock_actual + '\n')
```

```
def grabar_error(archivo, cod_art, cantidad, desc_error):  
    archivo.write(cod_art + ',' + cantidad + ',' + desc_error + '\n')
```

```
def actualizar_Maestro(arMaestro, arNovedades, arMaeActualizado, arLogErrores):
```

```
    # Tener en cuenta que en esta funcion estamos recibiendo los archivos  
    # abiertos, si no estuviéramos seguros que nos encontramos al principio  
    # de los archivos, antes de hacer la primer lectura, deberíamos aplicar  
    # un seek(0) a cada uno de los archivos, para asegurar que procesaremos  
    # los datos desde el principio al final de cada archivo
```

```
    MAXIMO = "9999" # Se debe asignar un valor que nunca sea alcanzado por las claves
```

```
    cod_art_mae, desc_art, stock = leer_MaeNov(arMaestro, MAXIMO+",",")  
    cod_art_nov, cantidad_vendida = leer_MaeNov(arNovedades, MAXIMO+",",")
```

```
    while (cod_art_mae != MAXIMO or cod_art_nov != MAXIMO):  
        if (cod_art_mae < cod_art_nov):  
            # Va directo al nuevo archivo  
            grabar_MaeActualizado(arMaeActualizado, cod_art_mae, desc_art, str(stock))  
            # Vuelvo a leer Maestro  
            cod_art_mae, desc_art, stock = leer_MaeNov(arMaestro, MAXIMO+",",")  
  
        elif (cod_art_mae > cod_art_nov):  
            # Es un error, una novedad sin registro en maestro  
            grabar_error(arLogErrores, cod_art_nov, cantidad_vendida, "Codigo de Artículo  
            Inexistente en Maestro")  
  
            # Se vuelve a leer novedades  
            cod_art_nov, cantidad_vendida = leer_MaeNov(arNovedades, MAXIMO+",",")  
  
        else: # son iguales  
            # Controlar que la cantidad vendida sea menor o igual al stock  
            if (int(cantidad_vendida) <= int(stock)): # actualizo stock  
                stock = int(stock) - int(cantidad_vendida)  
            else: # va al Log de Errores
```

```
grabar_error(arLogErrores, cod_art_nov, cantidad_vendida, "Cantidad supera  
stock")
```

```
# Leo solo el archivo de Novedades para ver si viene otro para el mismo cod_art  
cod_art_nov, cantidad_vendida = leer_MaeNov(arNovedades, MAXIMO+",")
```

```
#####
```

```
arMaestro = open("art_Maestro.csv","r")  
arNovedades = open("art_Novedades.csv","r")  
arMaeActualizado = open('art_Maestro_Actualizado.csv','w')  
arLogErrores = open('art_LogErrores.txt','w')
```

```
print("Comienzo Proceso")
```

```
actualizar_Maestro(arMaestro, arNovedades, arMaeActualizado, arLogErrores)
```

```
print("Fin Proceso")
```

```
arMaestro.close()  
arNovedades.close()  
arMaeActualizado.close()  
arLogErrores.close()
```