

```

program ejemplo_4_1;
  const valoralto="ZZZ";
  type
    empleado = record;
      Nombre      : string[50];
      Direccion   : string[50];
      Documento   : string[12];
      Edad        : integer;
      Observaciones: string[200];
    end;
  archivoemple = file of Empleado;
var
  reg: empleado;
  archivo, archivonuevo: archivoemple;
procedure leer (var archivo: archivoemple; var dato:empleado);
begin
  if (not eof(archivo))
  then
    read (archivo,dato)
  else
    dato.Nombre := valoralto;
  end;
end;

{programa principal}
begin
  assign (archivo, 'archemple');
  assign (archivonuevo, 'archnuevo');
  reset(archivo);
  rewrite(archivonuevo);
  leer(archivo,reg)

  {se copian los registros previos a Carlos Garcia}
  while (reg.Nombre<>"Carlos Garcia")
  begin
    write(archivonuevo,reg)
    leer(archivo,reg)
  end

  {se descarta a Carlos Garcia}
  leer(archivo,reg)

  {se copian los registros restantes}
  while (reg.Nombre<>valoralto) do
  begin
    write(archivonuevo,reg)
    leer(archivo,reg)
  end

  close(archivonuevo);
  close(archivo);
end.

```

```

program ejemplo_4_2;
  const valoralto="ZZZ";
  type
    empleado = record;
      Nombre      : string[50];
      Direccion   : string[50];
      Documento   : string[12];
      Edad        : integer;
      Observaciones: string[200];
    end;
  archivoemple = file of Empleado;
var
  reg: empleado;
  archivo: archivoemple;

procedure leer (var archivo: archivoemple; var dato:empleado);
begin
  if (not eof(archivo))
  then
    read (archivo,dato)
  else
    dato.Nombre := valoralto;
  end;
end;

{programa principal}
begin
  assign (archivo, 'archemple');
  reset(archivo);
  leer(archivo,reg);

  {se avanza hasta Carlos Garcia}
  while (reg.Nombre<>"Carlos Garcia")
  begin
    leer(archivo,reg);
  end;

  {se avanza hasta el siguiente a Carlos Garcia}
  leer(archivo,reg);

  {se copian los registros restantes}
  While (reg.Nombre<>valoralto) do
  begin
    Seek( archivo, filepos(archivo)-2 );
    write(archivo,reg);
    Seek( archivo, filepos(archivo)+1 );
    leer(archivo,reg);
  end;

  truncate(archivo);
end.

```

```

program ejemplo_4_3;

    const valoralto="ZZZ";

    type
        empleado = record;
            Nombre          : string[50];
            Direccion       : string[50];
            Documento       : string[12];
            Edad            : integer;
            Observaciones: string[200];
        end;
        archivoemple = file of Empleado;
var
    reg: empleado;
    archivo: archivoemple;

procedure leer (var archivo: archivoemple; var dato:empleado);
begin
    if (not eof(archivo)) then
        read (archivo,dato)
    else
        dato.Nombre := valoralto;
end;

{programa principal}
begin
    assign (archivo, 'archemple');
    reset(archivo);
    leer(archivo,reg);

    {se avanza hasta Carlos Garcia}
    while (reg.Nombre<>"Carlos Garcia")do
        begin
            leer(archivo,reg);
        end;

    {se genera una marca de borrado }
    reg.nombre := "****"

    {se borra lógicamente a Carlos Garcia}
    Seek( archivo, filepos(archivo)-1 );
    write(archivo,reg);

    close(archivo);
end.

```

```

program ejemplo_4_4;
  const valoralto="ZZZ";
  type
    empleado = record;
      Nombre      : string[50];
      Direccion   : string[50];
      Documento   : string[12];
      Edad        : integer;
      Observaciones: string[200];
    end;
  archivoemple = file of Empleado;
var
  reg: empleado;
  archivo: archivoemple;
procedure leer (var archivo: archivoemple; var dato:empleado);
begin
  if (not eof(archivo)) then
    read (archivo,dato)
  else
    dato.Nombre := valoralto;
end;

{programa principal}
begin
  assign (archivo, 'archemple');
  reset(archivo);
  leer(archivo,reg);
  { se recorre el archivo mostrando en pantalla solo los registros válidos}
  while (reg.Nombre<>valoralto) do
    begin
      if (reg.Nombre<>'***') then
        write('El nombre es:'+reg.Nombre)
        leer(archivo,reg);
      end;
    close(archivo);
  end.

```