

# Apéndice A

## Código Fuente

### A.1 Código Fuente perteneciente al procedimiento *Timer*

```
procedure TPPrincipal.TiempoEjecucionBnTimer(Sender:
TObject);
var
  Terminado : Boolean;
begin
  TiempoEjecucionBn.Interval := Intervalo +
                                trunc(TiempoPerdido);
  if TiempoPerdido >= 1.0 then TiempoPerdido :=
                                TiempoPerdido - 1.0;
  TiempoPerdido := TiempoPerdido + Fraccion;

  Terminado := False;

  //Trata de apagar
  ApagarNotaTR;

  if (CabezaActivas = 0) and (il - 1 = fl) then
  //Si ya no hay notas activas y es fin de compás
    if (TActual >= Instruccion[fl].Delta) or
      (Instruccion[fl].Delta = 0) then
      if (Instruccion[fl].Delta > TActual) then
        NvoCompas := False
      else
        NvoCompas := True;

  if NvoCompas then
  //Si terminó con el compás actual...
    begin
      TActual := 0;
      if IndiceLista <= InstLista then
      //... y hay mas compases, se toman los valores
        begin
          il := ListaATocar[IndiceLista, 1];
          fl := ListaATocar[IndiceLista, 2];
          pc := pc + 30;
          If IndiceLista <= 16 then Punto1Click(Sender)
          else Punto2Click(Sender);
          if IndiceLista = 16 then pc := -30;
          if (IndiceLista = 8) or (IndiceLista = 24) then
            pc := pc + 95;
          Inc(IndiceLista);
```

```

        NvoCompas := False;
    end
else
    //Apaga el timer no hay nada mas
    Terminado := True;
end;

//Prende notas
if not Terminado then
    PrendeNotaTR;

TActual := TActual + 1;
//Va afuera debido a que tiene que incrementar siempre

//Si ya no hay nada que ejecutar, ya terminó con todo...
reinicia y da formato
if (IndiceLista-1 = InstLista) and
    (CabezaActivas = 0) and
    (il - 1 = fl) then
begin
    TiempoEjecucionBn.Enabled := False;
    NvoCompas := True;
    BGenera.Enabled := True;
    BEjecuta.Visible := True;
    Stop.Visible := False;
    LimpiaPuntos;
end;
end; //Fin del procedure

```

## **A.2 Código Fuente perteneciente al método *Trata de Apagar***

```

Procedure ApagarNotaTR;
var
    apt, k, ant : Byte; //Apuntadores que recorre Tabla
begin
    apt := CabezaActivas;
    ant := CabezaActivas;
    While apt > 0 do //Hay notas activas
        with Tabla[apt] do
            begin
                dur := dur - 1;
                if dur = 0 then
                    begin
                        ApagarNota(ch, note);
                        k := apt;
                        if apt = ant then
                            begin
                                CabezaActivas := link;

```

```

        ant := link;
    end
    else
        Tabla[ant].link := link;
        apt := link;
        LiberaReg(k);
    end
else
    begin
        ant := apt;
        apt := link;
    end;
end;
end;
end;

```

### **A.3 Código Fuente perteneciente al método *Prende Notas***

```

Procedure PrendeNotaTR;
begin
    if TActual = Instruccion[il].Delta then
        begin
            TActual := 0;
            //ultsecuencia := Instruccion[il].Delta;
            repeat
                apt := SolicitaReg;
                with Tabla[apt] do
                    begin
                        ch := Instruccion[il].Canal;
                        note := Instruccion[il].Nota;
                        dur := Instruccion[il].Duracion;
                        link := CabezaActivas;
                        CabezaActivas := apt;
                        if Despliega then
                            Form1Editor.Entrada.Row := ren;
                            ActivarNota(ch, note);
                            Inc(il);
                            Inc(ren);
                        end;
                    end;
                until (il - 1 = fl ) or (Instruccion[il].Delta >
0);
            end;
        end;
    end;
end;

```