

```
unit QueuesArrayNC;
```

```
interface
Uses
  Tipos, stdctrls, SysUtils;

Const
  MIN = 1; // Frente de la cola
  MAX = 2000; // Final de la cola
  Nulo= 0; // Posicion NO Valida de frente o final

Type
  PosicionCola = LongInt;

  Cola = Object
  Private
    Elementos: Array Of TipoElemento;
    Inicio, Fin: PosicionCola;
    Q_Items: Integer;
    TDatoDeLaClave: TipoDatosClave;
    Size : LongInt;
    Function CantidadItems(): LongInt;
  Public
    Function Crear(avTipoClave: TipoDatosClave; alSize: LongInt): Resultado;
    Function EsVacía(): Boolean;
    Function EsLlena(): Boolean;
    Function Encolar(X:TipoElemento): Resultado;
    Function DesEncolar(): Resultado;
    Function Recuperar(): TipoElemento;
    Function RetornarClaves(): String;
    Function InterCambiar(Var CAux: Cola; bCrearVacía: Boolean): LongInt;
    Function LLenarClavesRandom(alSize: LongInt; RangoDesde, RangoHasta:LongInt): Resultado;
    Function Frente(): PosicionCola;
    Function Final(): PosicionCola;
    Function DatoDeLaClave: TipoDatosClave;
    Function SizeQueue(): LongInt;
    Function MaxSizeQueue(): LongInt;
  End;
```

```
implementation
```

```
// Creo la cola vacía
Function Cola.Crear(avTipoClave: TipoDatosClave; alSize: LongInt): Resultado;
Begin
  if alSize < Min then Crear:= CError;
  if alSize > Max then Crear:= CError;
  if (alSize >= Min) And (alSize <= Max) then Begin
    SetLength(Elementos, (alSize+1));
    Inicio := Nulo;
    Fin := Nulo;
    Q_Items := 0;
    TDatoDeLaClave := avTipoClave;
    Size := alSize;
    Crear := OK;
  End;
End;

// control de cola vacía
Function Cola.EsVacía(): Boolean;
Begin
  EsVacía := (Inicio = Nulo);
End;

// control de cola llena
Function Cola.EsLlena(): Boolean;
Begin
  EsLlena := (Fin = Size);
End;

// Agrega un elemento a la Cola al Final
Function Cola.Encolar(X:TipoElemento): Resultado;
Begin
```

```

Encolar := CError;
// Controla que el Tipo de Dato de la Clave sea Homogeneo a la Lista
if X.TipoDatoClave(X.Clave) <> TDatoDeLaClave then Begin
    Encolar := ClaveIncompatible;
    Exit;
End;
// Ahora Encolo
If EsLlena() Then Encolar := Llena
Else Begin
    Fin := Fin + 1;
    Elementos[Fin] := X;
    If EsVacía Then Inicio := Fin;
    Inc(Q_Items);
    Encolar := OK;
End;
End;

// Elimina un elemento de la Cola. Siempre del Frente
Function Cola.DesEncolar(): Resultado;
Var P, Q: PosicionCola;
Begin
    DesEncolar := CError;
    If EsVacía Then DesEncolar := Vacía
    Else Begin
        P := Inicio;
        For Q := P To (Fin - 1) Do
            Elementos[Q] := elementos[Q + 1]; // Aplasta el item a borrar
        Dec(Fin);
        If Fin < Inicio Then Crear(TDatoDeLaClave, Size); //Si borrar el unico que hay se crea vacía
        Dec(Q_Items);
        DesEncolar := OK;
    End;
End;

// retorna el elemento del frente de la cola
Function Cola.Recuperar(): TipoElemento;
Var X: TipoElemento;
Begin
    Recuperar := X.TipoElementoVacío;
    If Not EsVacía Then
        Begin
            Recuperar := Elementos[Inicio];
        End;
End;

// Pasa los elementos de una Cola Auxiliar a la Cola "Cola"
Function Cola.InterCambiar(Var CAux: Cola; bCrearVacía: Boolean): LongInt;
Var X: TipoElemento;
    I: Integer;
Begin
    I := 0;
    If bCrearVacía = true Then Crear(TDatoDeLaClave, CAux.Size);
    While Not CAux.EsVacía() Do Begin
        X := CAux.Recuperar();
        If Encolar(X) = OK Then Inc(I);
        CAux.DesEncolar;
    End;
    InterCambiar := I;
End;

// Retorna un string con todos los elementos de Cola
// Cada Item separado por Retorno de Carro + Final de Linea
Function Cola.RetornarClaves():String;
Var X: TipoElemento;
    S, SS: String;
    CAux: Cola;
Begin
    SS:= '';
    CAux.Crear(TDatoDeLaClave, Size);
    While Not EsVacía() Do Begin
        X := Recuperar();
        CAux.Encolar(X);
    End;
    S:= SS;
    For I:= 1 To Fin Do
        S:= S + Elementos[I].Clave + RetornoCarro + RetornoLinea;
    End;
    RetornarClaves:= S;
End;

```

```
S := X.ArmazString;
SS := SS + S + cCRLF;
DesEncolar();
End;
InterCambiar(Caux, True);
RetornarClaves := SS;
End;

// Llena la cola con valores aleatorios en el atributo DI
// desde <RangoDesde> hasta <RangoHasta>
Function Cola.LLenarClavesRandom(alSize: LongInt; RangoDesde, RangoHasta:LongInt): Resultado;
Var X: TipoElemento;
Begin
  X.Valor2 := NIL;
  TdatoDeLaClave := Numero;
  Crear(TdatoDeLaClave, alSize);
  X.Inicializar(TdatoDeLaClave, '');
  Randomize;
  While Not EsLlena Do Begin
    X.Clave := RangoDesde + Random(RangoHasta);
    Encolar(X);
  End;
End;

// retorno posicion del frente
Function Cola.Frente(): PosicionCola;
Begin
  Frente := Inicio;
End;

// retorno posicion del final
Function Cola.Final(): PosicionCola;
Begin
  Final := Fin;
End;

// Retorno la cantidad de elementos
Function Cola.CantidadItems(): LongInt;
Begin
  CantidadItems := Q_Items;
End;

Function Cola.DatoDeLaClave: TipoDatosClave;
Begin
  DatoDeLaClave := TdatoDeLaClave;
End;

Function Cola.SizeQueue(): LongInt;
Begin
  SizeQueue := Size;
End;

Function Cola.MaxSizeQueue(): LongInt;
Begin
  MaxSizeQueue := MAX;
End;

end.
```