```vhdl
library ieee;
use ieee.std_logic_1164.all;

package archivo_componentes  is
   Component CLOCK_DIV_10_MHz  is
    Port
    (  CLOCK_10MHz     :IN   STD_LOGIC;
       CLOCK_1MHz     :OUT  STD_LOGIC;
       CLOCK_100KHz   :OUT  STD_LOGIC;
       CLOCK_10KHz    :OUT  STD_LOGIC;
       CLOCK_1KHz     :OUT  STD_LOGIC;
       CLOCK_100Hz    :OUT  STD_LOGIC;
       CLOCK_10Hz     :OUT  STD_LOGIC;
       CLOCK_1Hz      :OUT  STD_LOGIC);
   end Component;

   Component encoder_Dec_BCD  is
      Port ( Ent: in STD_LOGIC_VECTOR(9 downto 0);
             Salida : OUT STD_LOGIC_VECTOR(3 downto 0));
   end Component;

   Component comparador_4bits  is
      port(A,B: in std_logic_vector(3 downto 0);
           igual: out std_logic);
   end Component;

   Component comparador_24bits  is
      port(A,B: in std_logic_vector(23 downto 0);
      igual: out std_logic);
   end Component;

   component mux_3s
      Port (num1: IN std_logic_vector (3 downto 0);
            num2: IN std_logic_vector (3 downto 0);
            num3: IN std_logic_vector (3 downto 0);
            num4: IN std_logic_vector (3 downto 0);
            num5: IN std_logic_vector (3 downto 0);
            num6: IN std_logic_vector (3 downto 0);
            selector: IN std_logic_vector (2 downto 0);
            s: OUT std_logic_vector (3 downto 0));
   end component;

   Component mux_8s is
      Port (num1: IN std_logic_vector (23 downto 0);
            num2: IN std_logic_vector (23 downto 0);
            num3: IN std_logic_vector (23 downto 0);
            num4: IN std_logic_vector (23 downto 0);
            num5: IN std_logic_vector (23 downto 0);
            num6: IN std_logic_vector (23 downto 0);
            num7: IN std_logic_vector (23 downto 0);
            num8: IN std_logic_vector (23 downto 0);
            selector: IN std_logic_vector (7 downto 0);
            s: OUT std_logic_vector (23 downto 0));
   end Component;

   Component mux_2selector_restador  is
      Port (num1: IN std_logic_vector (3 downto 0);
            num2: IN std_logic_vector (3 downto 0);
            num3: IN std_logic_vector (3 downto 0);
            selector: IN std_logic_vector (1 downto 0);
            s: OUT std_logic_vector (3 downto 0));
   end Component;

   Component mux_1selector  is
      Port (num1: IN std_logic_vector (3 downto 0);
            num2: IN std_logic_vector (3 downto 0);
            selector: IN std_logic;
            s: OUT std_logic_vector (3 downto 0));
   end Component;

   Component contador_1_4bits  is
       PORT (
           clk_cont,reset_cont  : IN  STD_LOGIC;
           suma: OUT STD_LOGIC_VECTOR(3 DOWNTO 0)
       );
   end Component;

   Component flip_flop_D  is
      port (clk, reset: in std_logic;
             d: in std_logic;
             q: out std_logic);
   end Component;

   component RAM is
     port (clock   : in  std_logic;
           we      : in  std_logic;
           address : in  std_logic_vector (3 downto 0);
           datain  : in  std_logic_vector (23 downto 0);
           dataout : out std_logic_vector (23 downto 0));
   end component;

   Component registro_sostenimiento is
      PORT(clock,reset,enable: IN STD_LOGIC;
          Ent : IN STD_LOGIC_VECTOR(3 downto 0);
          Q : OUT STD_LOGIC_VECTOR (3 downto 0));
   end Component;

   Component MSS_Call_Center is
   Port (clock,resetn,start,boton,BORRAR,Fu,HISTORIAL_LLAMADAS,cont6,Re_oc,Re_des,
igual_oc,dir_hist_igual,igual_hist1,
         REALIZAR_LLAMADA,FINALIZAR_LLAMADA,icuenta0,icuenta1,icuenta2,icuenta3,icuenta4,
icuenta5,u1,u2,u3,u4,u5,u6,u7,u8,
         Fu_ad1,Fu_ad2,Fu_ad3,Fu_ad4,Fu_ad5,Fu_ad6,Fu_ad7,Fu_ad8,iusuario1,iusuario2,
iusuario3,iusuario4,iusuario5,iusuario6,
         iusuario7,iusuario8,iusuario_ant1,iusuario_ant2,iusuario_ant3,iusuario_ant4,
iusuario_ant5,iusuario_ant6,iusuario_ant7,
         iusuario_ant8: IN std_logic;

         incrementa,reset_reg,cuenta_a_0,incremento_u1,incremento_u2,incremento_u3,
incremento_u4,incremento_u5,incremento_u6,
         incremento_u7,incremento_u8,we_u1,we_u2,we_u3,we_u4,we_u5,we_u6,we_u7,we_u8,
reset_oc,OCUPADO,reset_hist,disp_select,rhist_u1,
         rhist_u2,rhist_u3,rhist_u4,rhist_u5,rhist_u6,rhist_u7,rhist_u8,ihist_u1,ihist_u2
,ihist_u3,ihist_u4,ihist_u5,ihist_u6,ihist_u7,
         ihist_u8,led1,led2,led3,led4,led5,led6,led7,led8,clock_ff_d: OUT std_logic;

         selec_u1,selec_u2,selec_u3,selec_u4,selec_u5,selec_u6,selec_u7,selec_u8: OUT
std_logic_vector (1 downto 0);

         enable_reg: OUT std_logic_vector (5 downto 0));

   end Component;


end archivo_componentes;
```