

```

using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

namespace fronton
{
    /// <summary>
    /// Descripción breve de Form1.
    /// </summary>
    public class Form1 : System.Windows.Forms.Form
    {
        private System.Windows.Forms.PictureBox pelota;
        private System.Windows.Forms.Timer timer1;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.PictureBox raqueta;
        private System.Windows.Forms.Label lblvidas;
        private System.ComponentModel.IContainer components;

        public Form1()...
        /**/
        protected override void Dispose( bool disposing )...
        Código generado por el Diseñador de Windows Forms

        /**/ [...]
        static void Main() ...
        // direccion_vertical, direccion_horizontal,posicion_del_cursor,tamaño de raqueta,
        limite para golpear, vidas
        int velocidad,dv,dh,pc,tr=200,limite,vidas=3;
        private void Form1_Load(object sender, System.EventArgs e)
        {
            raqueta.Top=Form1.ActiveForm.Height-60; // Posicionar la raqueta en la parte
inferior
            limite=raqueta.Top; // Saber el límite maximo de pelota valida
            velocidad=6; // Velocidad inicial (depende del procesador)
            timer1.Interval=2;// Cada 2 milisegundos se ejecutará el timer1
            Cursor.Hide(); // Ocultar el cursor
            dv=velocidad; // Dirección vertical
            dh=velocidad; // Direccion horizontal
            timer1.Enabled=true; // Activar el timer1
        }

        private void timer1_Tick(object sender, System.EventArgs e)
        {
            pelota.Top=pelota.Top+dv; // Dibujar la pelota en una posicion vertical
            pelota.Left=pelota.Left+dh; // Dibujar la pelota en una posicion horizontal

            if (pelota.Top+26>=limite) // checar si pasó el limite de la parte inferior
            { // checando si pegó en la raqueta
                if ((pelota.Left>=raqueta.Left) && (pelota.Left+40<raqueta.Left+tr))
                {
                    dv=-velocidad; // que el desplazamiento vertical se haga negativo
                }else
                {
                    vidas--; // quitarle una vida por cada pelota que pasa el limite
inferior
                    // sin golpear en la raqueta.
                    lblvidas.Text=vidas.ToString(); // Mostrar las vidas restantes
                    pelota.Top=8;pelota.Left=8; //posicionar una nueva pelota
                }
            }else
            {
                if(pelota.Top<=1) // checar si hubo rebote superior
                {
                    dv=velocidad; //hacer desplazamiento vertical positivo
                }

                if (pelota.Left+40>=Form1.ActiveForm.Width) // checar si hubo rebote derecho
                {
                    dh=-velocidad;//hacer desplazamiento horizontal negativo
                }else
                {
                    if (pelota.Left<=1) // checar si hubo rebote izquierdo

```

```
    {
        dh=velocidad; //hacer desplazamiento horizontal positivo
    }
}

private void Form1_MouseMove(object sender, System.Windows.Forms.MouseEventArgs e)
{
    pc=Cursor.Position.X; // Obtener la posición del cursor
    // Eliminar vibración de raqueta en puntos críticos
    if (pc>=946) // si la posición del cursor es mayor o igual al borde derecho
    { pc=946; }
    if (pc<=85) // si la posición del cursor es menor o igual a 85
    { pc=1; }
    raqueta.Left=pc; // dibujar la raqueta
}

private void Form1_KeyPress(object sender, System.Windows.Forms.KeyPressEventArgs e)
{
    if (e.KeyChar==27) Application.Exit(); // Salir del juego con Escape
}
}
```