

```

namespace Entidades
{
    public delegate void FinAtencionPaciente(Paciente p, Medico m);

    public abstract class Medico : Persona
    {
        private Paciente pacienteActual;
        protected static Random tiempoAleatorio;

        public virtual string EscaAtendiendoA { get { return this.pacienteActual.ToString(); } }
        public Paciente AtenderA { set { this.pacienteActual = value; } }

        public event FinAtencionPaciente AtencionFinalizada;

        static Medico()
        {
            tiempoAleatorio = new Random();
        }
        public Medico(string nombre, string apellido)
            :base(nombre, apellido)
        { }

        protected abstract void Atender();
        protected void FinalizarAtencion()
        {
            this.AtencionFinalizada(this.pacienteActual, this);
            this.pacienteActual = null;
        }
    }
}
=====

public class MGeneral : Medico, IMedico
{
    public MGeneral(string nombre, string apellido)
        :base(nombre, apellido)
    { }

    protected override void Atender()
    {
        Thread.Sleep(tiempoAleatorio.Next(1500, 2200));
        base.FinalizarAtencion();
    }

    public void IniciarAtencion(Paciente p)
    {
        this.AtenderA = p;
        Thread h1 = new Thread(this.Atender);
        h1.Start();
    }
}
=====

public class Paciente : Persona
{
    private int turno;
    private static int ultimoTurnoDado;
    public int Turno { get { return this.turno; } }

    static Paciente()
    {
        ultimoTurnoDado = 0;
    }
    public Paciente(string nombre, string apellido)
        :base(nombre, apellido)
    {
        ultimoTurnoDado++;
        this.turno = ultimoTurnoDado;
    }

    public Paciente(string nombre, string apellido, int turno)
        :base(nombre, apellido)
    {
        ultimoTurnoDado = turno;
    }
    public override string ToString()
    {...} =====

```

```

namespace sistemaGestionSanatorio
{
    public partial class frmFinal : Form
    {
        private MEspecialista medicoEspecialista;
        private MGeneral medicoGeneral;
        private Thread mocker;
        private Queue<Paciente> pacientesEnEspera;

        public frmFinal()
        {
            InitializeComponent();

            this.medicoGeneral = new MGeneral("Luis", "Salinas");
            this.medicoEspecialista = new MEspecialista("Jorge",
"Iglesias", EEspecialidad.Traumatologo);

            this.pacientesEnEspera = new Queue<Paciente>();

            this.medicoGeneral.AtencionFinalizada += this.FinAtencionPaciente;
            this.medicoEspecialista.AtencionFinalizada += this.FinAtencionPaciente;
        }

        private void MockPacientes()
        {
            this.pacientesEnEspera.Enqueue(new Paciente("Joey", "Ramone"));
            this.pacientesEnEspera.Enqueue(new Paciente("johnny", "Ramone"));
            Thread.Sleep(5000);
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            this.mocker = new Thread(this.MockPacientes);
            this.mocker.Start();
        }

        private void frmFinal_FormClosing(object sender, EventArgs e)
        {
            if (this.mocker.IsAlive)
                this.mocker.Abort();
        }

        private void AtenderPacientes(IMedico imedico)
        {
            if (this.pacientesEnEspera.Count == 0)
                MessageBox.Show("no hay mas pacientes");
            if (this.pacientesEnEspera.Count > 0 && imedico is MEspecialista)
                ((MEspecialista)imedico).IniciarAtencion(this.pacientesEnEspera.Dequeue());
            if (this.pacientesEnEspera.Count > 0 && imedico is MGeneral)
                ((MGeneral)imedico).IniciarAtencion(this.pacientesEnEspera.Dequeue());
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.AtenderPacientes(this.medicoGeneral);
        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.AtenderPacientes(this.medicoEspecialista);
        }

        private void FinAtencionPaciente(Paciente p, Medico m)
        {
            MessageBox.Show(string.Format("Finalizó la atención de {0} por {1}", p.ToString(),
m.ToString()));
        }
    }
}

```