

Diseñar el siguiente formulario:

The screenshot shows a Windows application window with the title bar 'Final 27-07-2017'. Inside the window, there are two buttons stacked vertically. The top button is labeled 'Atender Médico General' and the bottom button is labeled 'Atender Médico Especialista'.

```

frmFinal
Class
+ Form

Fields
medicoEspecialista : MEspecialista
medicoGeneral : MGeneral
mock : Thread
pacientesEnEspera : Queue<Paciente>

Methods
AtenderPacientes(IMedico iMedico) : void
btnEspecialista_Click(object sender, EventArgs e) : void
btnGeneral_Click(object sender, EventArgs e) : void
FinAtencion(Paciente p, Medico m) : void
frmFinal()
frmFinal_FormClosing(object sender, FormClosingEventArgs e) : void
frmFinal_Load(object sender, EventArgs e) : void
MockPacientes() : void
    
```

11) (1 punto) Crear los botones y las modificaciones necesarias para poder serializar y desSerializar una lista de pacientes auxiliar, tomando los datos de pacientesEnEspera

12) (3 puntos) Modelar un sistema de atención para un Sanatorio.

Siendo:

- frmFinal() el constructor del formulario, dónde se instanciarán los atributos del formulario, siendo medicoEspecialista y medicoGeneral:
`this.medicoGeneral = new MGeneral("Luis", "Salinas");`
`this.medicoEspecialista = new MEspecialista("Jorge", "Iglesias", MEspecialista.Especialidad.Traumatologo);`
- frmFinal_Load() el evento de carga del formulario, dónde inicializaremos el hilo mock.
- frmFinal_FormClosing() el evento de cierre del formulario, dónde, si el hilo mock aun está activo, se abortará.
- MockPacientes() dónde se agreguen pacientes a la cola pacientesEnEspera, haciendo un Sleep de 5000 (`Thread.Sleep(5000)`).
- AtenderPacientes(IMedico) será invocado por los eventos click de los botones (btnEspecialista_Click y btnGeneral_Click) pasandole el médico que corresponda (medicoEspecialista o medicoGeneral, respectivamente). En el caso de haber pacientes en espera, se deberá iniciar la atención del primer elemento de la cola.