

Universidad Tecnológica Nacional
Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	14/12/2023			
Nombre:		Docente ⁽²⁾ :				
División:	2°C	Nota ⁽²⁾ :				
Legajo:		Firma ⁽²⁾ :				
Instancia ⁽¹⁾ :	PP	RPP	SP	RSP	X	FIN

(1) Las instancias validas son: 1^{er} Parcial (PP), Recuperatorio 1^{er} Parcial (RPP), 2^{do} Parcial (SP), Recuperatorio 2^{do} Parcial (RSP), Final (FIN). Marque con una cruz.

(2) Campos a ser completados por el docente.

IMPORTANTE:

- **2 (dos) errores en el mismo tema anulan su puntaje.**
- La correcta documentación y reglas de estilo de la cátedra serán evaluadas.
- El proyecto debe ser creado en .Net 6.
- Colocar sus datos personales en el nombre de la carpeta principal y la solución: Apellido.Nombre.Div. Ej: Pérez.Juan.2C. No se corregirán proyectos que no sea identificable su autor.
- No se corregirán exámenes que no compilen.
- **Reutilizar** tanto código como crean necesario.
- Colocar nombre de la clase (en estáticos), **this** o **base** en todos los casos que corresponda.
- Aplicar los principios de los 4 pilares de la POO.

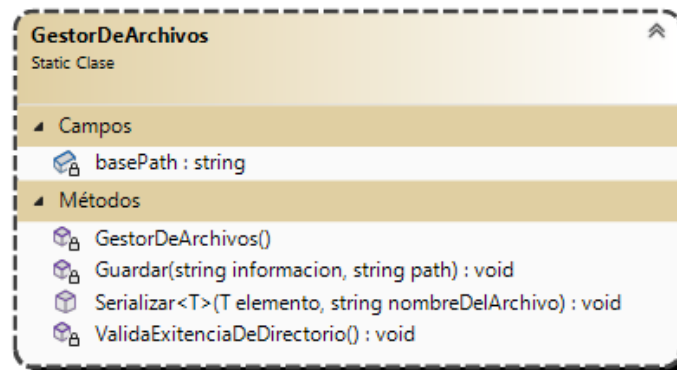
TIEMPO MÁXIMO PARA RESOLVER EL EXAMEN 120 MINUTOS.

1. Partir de la solución entregada. Modificar su nombre con el siguiente formato: [APELLIDO].[NOMBRE].
2. Crear la base de datos 14122023-rsp y correr el siguiente script:

```
USE [14122023-rsp]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[log](
[id] [int] IDENTITY(1,1) NOT NULL,
[pacientes_atendidos] [int] NOT NULL,
[alumno] [varchar](60) NOT NULL
) ON [PRIMARY]
GO
```

Files

3. Dentro del proyecto se deberá respetar el siguiente esquema:



4. GestorDeArchivos será estática.
 - a. En el constructor:
 - i. En el atributo basePath (privado) se almacenará la referencia al escritorio de la pc. Y se le concatenará un el nombre de la carpeta del parcial: ej {path escritorio}+\\14122023_Alumno\\
 - ii. Llamar al método ValidaExistenciaDeDirectorio.
 - b. ValidaExistenciaDeDirectorio:
 - i. Será privado.
 - ii. Si no existe el directorio almacenado en path, se creará.
 - c. Guardar:
 - i. Será privado.
 - ii. Será el método para poder generar archivos de texto.
 - d. Serializar:
 - i. Será genérico y solo aceptará tipos por referencia.
 - ii. Será el método encargado de serializar en json.
 - e. Reutilizar código donde crea necesario.

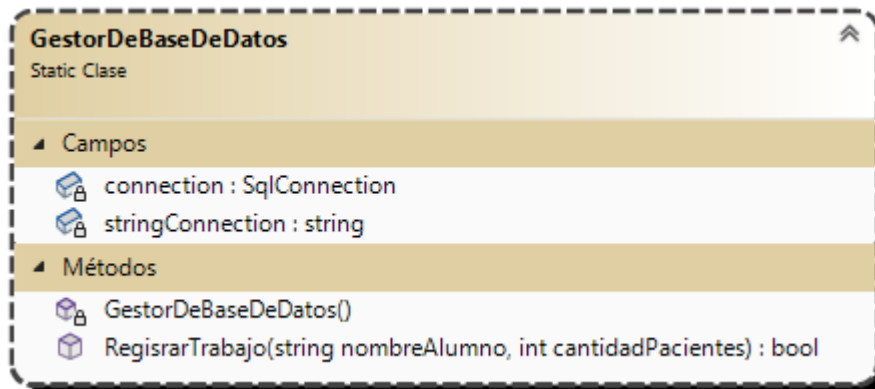
Excepciones

5. Dentro del proyecto se deberá respetar el siguiente esquema:



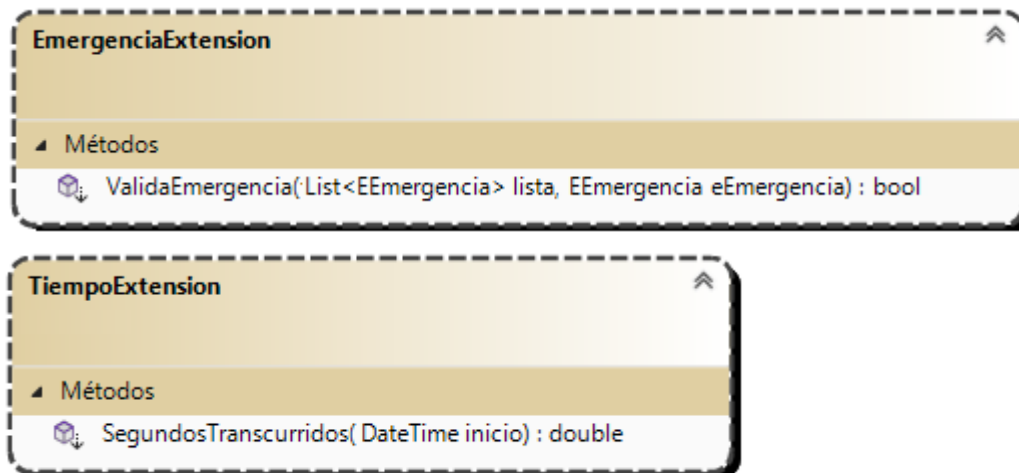
Bases de datos

6. GestorDeBaseDeDatos será estática:



- En el constructor de clase inicializar el string connection.
- RegistrarTrabajo**, recibirá el un string y un entero. Se usara para registrar en la base de datos el nombre del alumno y la cantidad de emergencias gestionadas.

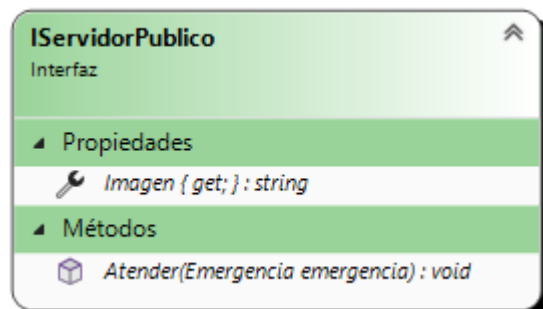
Métodos de extensión



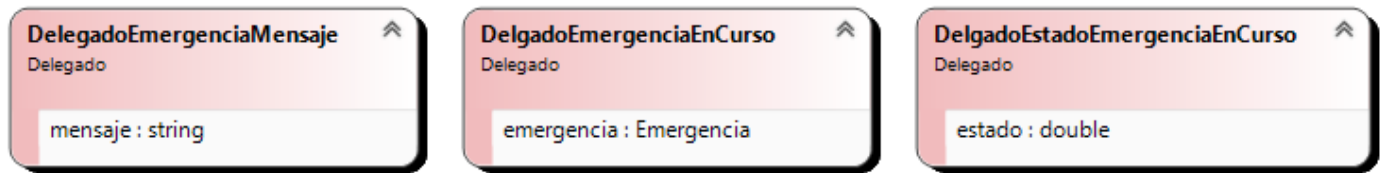
- ValidarEmergencia** extenderá la clase `List<EEmergencia>` la cual, adicionalmente recibirá un valor como parámetro. Este último se utilizará para validar si dentro de la lista existe el valor recibido. Retornara un bool para reflejar el resultado. **Usar el método Any de listas para lograr esto.**
- SegundosTrasncurridos** extenderá la clase `DateTime`. Retornará la diferencia en segundos entre la hora actual y la de inicio.

Interfaces

- Dentro del proyecto se deberá respetar el siguiente esquema:

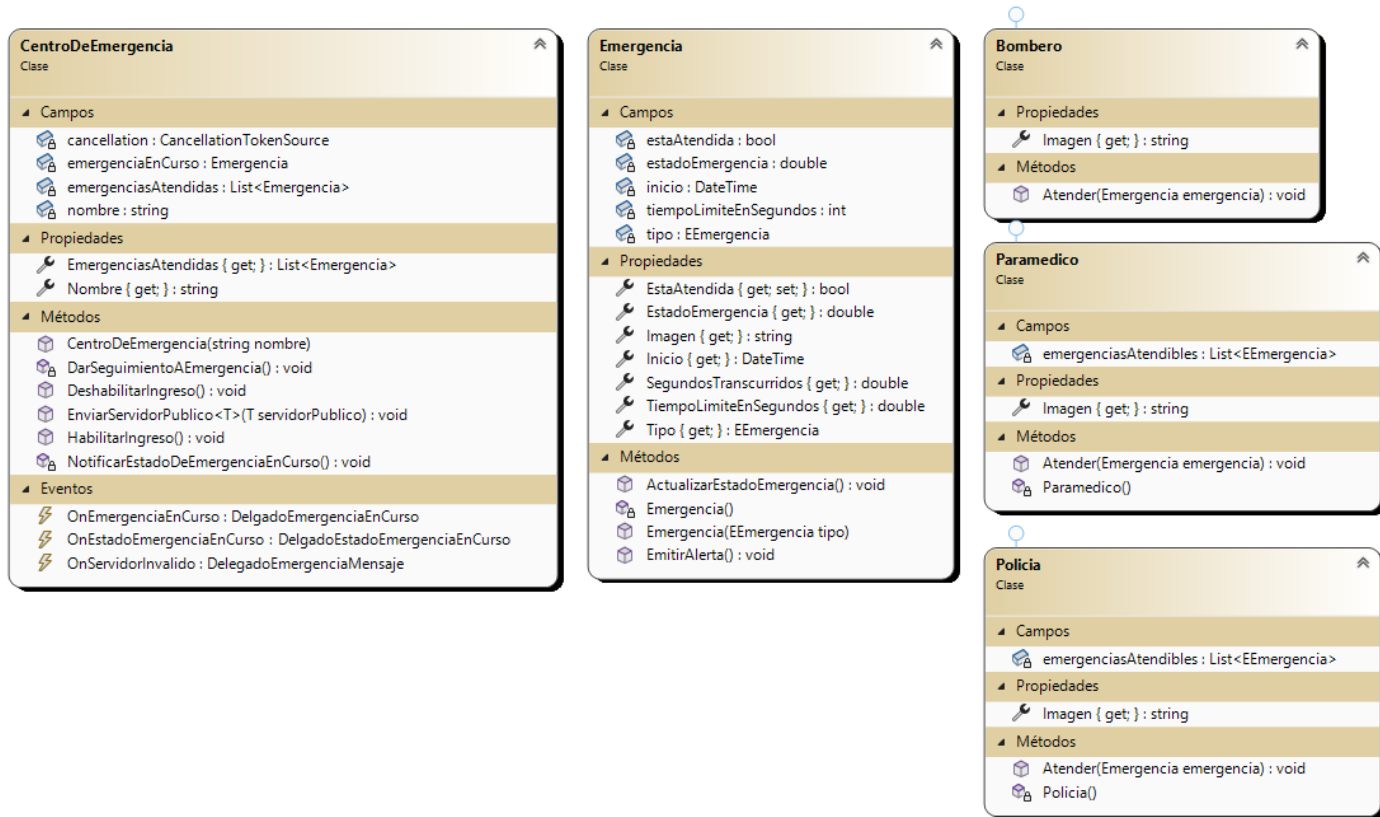


Delegados



Entidades

10. Dentro del proyecto se deberá respetar el siguiente esquema:



11. Bombero:

- Implementará la interfaz **IServidorPublico**
- La propiedad **Imagen** retornara un stirng con el siguiente valor => `$/./assets/{this.GetType().Name}.gif`
- El método **Atender**, tendrá la responsabilidad de atender una **Emergencia**, esto se llevará adelante modificando la propiedad **EstaAtendida** de la **Emergencia**. Por defecto un bombero podrá atender cualquier **Emergencia**.

12. Policía y Paramedico:

- Implementará la interfaz **IServidorPublico**
- La propiedad **Imagen** retornara un stirng con el siguiente valor => `$/./assets/{this.GetType().Name}.gif`
- El método **Atender**, tendrá la responsabilidad de atender una **Emergencia**, esto se llevará adelante modificando la propiedad **EstaAtendida** de la **Emergencia**. Un policía solo podrá atender las emergencias del listado de la clase. Para ello validara la emergencia recibida. Y en caso de que no la pueda atender lanzara una excepción de tipo **ServidorPublicoInvalidoException** indicando "El servidor público no puede atender este tipo de emergencias".

13. Emergencia:

- La propiedad **SegundosTranscurridos** deberá retornar los segundos transcurridos en base al inicio de la emergencia.

14. CentroDeEmergencia:

- Métodos:

- i. **HabilitarIngreso:** Tendrá la responsabilidad de que mientras que no se requiera cancelación notificar emergencias en curso. Para ello generara un tipo del enumerado **EEmergencia** de manera aleatoria, instanciara en el atributo **emergenciaEnCurso** una **Emergencia** con dicho valor aleatorio, notificara la emergencia generada a través del evento **OnEmergenciaEnCurso** y llamara **DarSeguimientoAEmergencia**.
- ii. **DarSeguimientoAEmergencia:** Tendrá la responsabilidad de emitir la alerta de la emergenciaEnCurso. Luego, mientras que:
 1. No se requiera cancelación y.
 2. Los **SegundosTrascurridos** de la **emergenciaEnCurso** sean menores que el **TiempoLimiteEnSegundos** de la Emergencia y.
 3. La **emergenciaEnCurso** no este atendida.
 4. Dormirá el hilo 1 segundo.
 5. Llamara a **ActualizarEstadoEmergencia** de la **emergenciaEnCurso**.
 6. Llamará a **NotificarEstadoEmergenciaEnCurso**.
- iii. **NotificarEstadoDeEmergenciaEnCurso:** Tendrá la responsabilidad de notificar a través del evento **OnEstadoEmergenciaEnCurso** el estado de la emergencia en curso.
- iv. **DeshabilitarIngreso:** Tendrá la responsabilidad de interrumpir la ejecución realizada en un hilo secundario.
- v. **EnviarServidorPublico:** Sera un método genético el cual solo podrá recibir tipos que implementen la interfaz **IServidorPublico**. A su vez tendrá la responsabilidad de ejecutar en un hilo secundario:
 1. Dormirá el hilo durante 3 segundos.
 2. Hará que el servidor público recibido por parámetro atienda la **emergenciaEnCurso**.
 3. Agregara la emergencia en curso al listado de emergencias atendidas.
 4. Si el servidor publico no puede atender dicha emergencia, notificara el error a través del evento **OnServidorInvalido**.

Formulario

15. Desarrollar todo lo indicado con comentario //Alumno:

Test Unitarios

16. Darle un nombre claro al proyecto, sus clases y sus métodos

17. Agregar 2 test unitarios:

- a. Forzar, mediante el código la ejecución de **ServidorPublicoInvalidoException**, validar que suceda de forma correcta.
- b. Al instanciar un nuevo **CentroDeEmergencia**, la cantidad de **emergenciasAtendidas** debe ser igual a 0 (cero).