

**Universidad Tecnológica Nacional**  
**Facultad Regional Avellaneda**



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

**Materia: Laboratorio de Programación II**

Apellido:		Fecha:	11-02-2021
Nombre:		Docente <sup>(2)</sup> :	Dávila/Oggioni
División:	2°	Nota <sup>(2)</sup> :	
Legajo:		Firma <sup>(2)</sup> :	
Instancia <sup>(1)</sup> :	<div style="display: flex; justify-content: space-around;"> <span><b>PP</b></span> <span></span> <span><b>RPP</b></span> <span></span> <span><b>SP</b></span> <span></span> <span><b>RSP</b></span> <span></span> <span><b>FIN</b></span> <span><b>X</b></span> </div>		

(1) Las instancias validas son: 1<sup>er</sup> Parcial (**PP**), Recuperatorio 1<sup>er</sup> Parcial (**RPP**), 2<sup>do</sup> Parcial (**SP**), Recuperatorio 2<sup>do</sup> Parcial (**RSP**), Final (**FIN**). Marque con una cruz.

(2) Campos a ser completados por el docente.

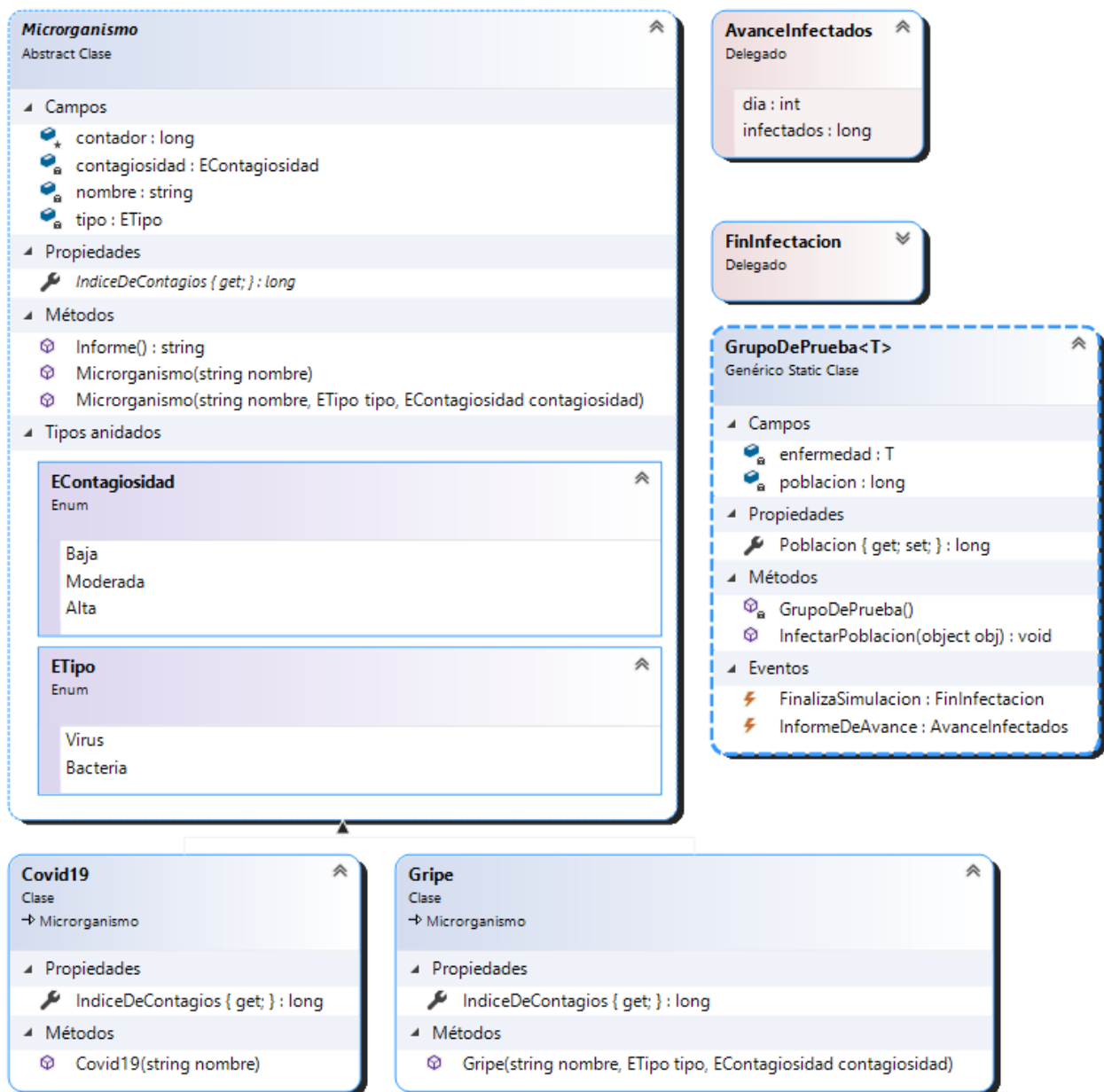
**IMPORTANTE:**

- **2 (dos) errores en el mismo tema anulan su puntaje.**
- La correcta documentación y reglas de estilo de la cátedra serán evaluadas.
- Colocar sus datos personales en el nombre de la carpeta principal y la solución: Apellido.Nombre. Ej: Pérez.Juan. 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.
- La entrega será en un archivo comprimido, el cual debe contar con Apellido y Nombre, al igual que la solución. Se entregará al finalizar, mediante Slack, por MD al docente a cargo.
- **No se corregirán exámenes que no compilen.**

Al finalizar, comprimir la carpeta de la Solución completa y enviar vía Slack a cualquiera de los profesores presentes en el examen.

Se creará un simulador del avance de una enfermedad, dado un cierto número de habitantes a evaluar y eligiendo una enfermedad. Dicho microorganismo tendrá un índice de contagio que indicará cada paciente a cuantos es capaz de contagiar.

Cumplir con el siguiente diagrama de clases:



### Microorganismos:

- El contador será definido en 1 en sus constructores. Esta variable contendrá la cantidad de gente contagiada por esta simulación, siendo alterada cada vez que se llame a la propiedad `IndiceDeContagios`.
- Informe retornará un texto con el siguiente formato: " El TIPO lleva el nombre de NOMBRE y tiene una contagiosidad CONTAGIOSIDAD. Su impacto de contagios se calcula en INDICE DE CONTAGIOS."

### Gripe:

- Su índice de contagios será de duplicación del contador.

### Covid-19:

- Su índice de contagios será de quintuplicación del contador.

### GrupoDePrueba:

- La clase es **estática**.
- Su tipo genérico deberá tener una restricción de tipo: deberá ser `Microorganismo` o uno de sus derivados.
- Por defecto, su población será de 10000000.
- `InfectarPoblacion`:
  - Controlará que su argumento sea del tipo `T`.
  - Establecerá la enfermedad a simular (cargar el atributo de clase), e iniciará el día en 1.
  - Calculará mediante la propiedad `IndiceDeContagios` cuantos infectados hay.

- Los infectados no podrán superar la cantidad de población, solo igualarla.
- Se informará mediante el evento InformeDeAvance el estado de la simulación.
- Se incrementará el día, se hará un sleep de 0.75 segundos y se iterará hasta completar la infección de toda la población.
- Una vez finalizado el proceso anterior, se deberá informar mediante FinalizaSimulacion que toda la población ha sido infectada.
- Tener en cuenta no llamar a eventos que no tienen suscriptores.

### Formulario **SimuladorForm**:

The image shows two side-by-side windows. The left window is titled 'Simulador de Pandemia' and contains a form with two input fields: 'Población a evaluar' with the value '100000' and 'Microorganismo' with a dropdown menu showing 'Covid-19'. Below these fields is a blue 'Ejecutar' button. At the bottom of the window, there is a text area displaying the simulation results: '¡Toda la población fue infectada!', followed by a list of daily infection counts from Day 1 to Day 7, and a concluding statement about the virus and its contagiousness. The right window is the Visual Studio code editor for 'SimuladorForm'. It shows the 'Campos' (Fields) section with a list of controls: btnEjecutar (Button), cmbMicroorganismo (ComboBox), components (IContainer), hilo (Thread), lblMicroorganismo (Label), lblPoblacion (Label), txtEvolucion (RichTextBox), and txtPoblacion (TextBox). The 'Métodos' (Methods) section lists several methods: btnEjecutar\_Click, Dispose, Ejecutar, Final, Informe, InitializeComponent, SimuladorForm, and SimuladorForm\_FormClosing.

- Tener en cuenta el formato de visualización: debajo de toda la información de la enfermedad, cada aviso de avance irá por encima, y arriba de todo el informe de finalización. Dicha información deberá ser impresa paso a paso, a medida que se avance en la simulación.
- Se cargará de forma opcional la población a evaluar y el microorganismo. Si no se carga nada, será la población precargada y la enfermedad Covid-19.
- Al finalizar la ejecución del programa, guardar un archivo en el escritorio con toda la información que se visualiza en el txtEjecutar.
- La simulación se iniciará por medio de un hilo al presionar el botón Ejecutar.
- Si se cierra el programa cuando aún se está ejecutando el hilo, este deberá ser cerrado en el acto para finalizar el programa de forma correcta.

### **Test Unitario:**

- Crear al menos un test unitario para evaluar alguna acción importante del sistema.