

# Universidad Tecnológica Nacional Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	07/05/2019
Nombre:		Docente <sup>(2)</sup> :	F. Dávila / D. Boullon
División:	2ºC	Nota <sup>(2)</sup> :	
Legajo:		Firma <sup>(2)</sup> :	
Instancia <sup>(1)</sup> :	<div style="display: flex; justify-content: space-around;"> <span>PP</span> <span>X</span> <span>RPP</span> </div>	<div style="display: flex; justify-content: space-around;"> <span>SP</span> <span></span> <span>RSP</span> </div>	<div style="display: flex; justify-content: space-around;"> <span>FIN</span> <span></span> </div>

(1) Las instancias validas son: 1º Parcial (PP), Recuperatorio 1º Parcial (RPP), 2º Parcial (SP), Recuperatorio 2º Parcial (RSP), Final (FIN). Marque con una cruz.

(2) Campos a ser completados por el docente.

### IMPORTANTE:

- Guardar el proyecto en el **disco D:**. Ante un corte de energía o problema con el archivo de corrección, el alumno será responsable de que el proyecto sea recuperable.
- **2 (dos) errores en el mismo tema anulan su puntaje.**
- **Errores de conceptos de POO anulan el punto.**
- **Cada tema vale 1 (un) punto (Herencia, Generics, Test Unitarios, etc.). La correcta documentación también será evaluada.**
- **Se deberán tener al menos el 60% bien de los temas a evaluar según la instancia para lograr la aprobación.**
- Colocar sus datos personales en el nombre del proyecto principal, colocando: Apellido.Nombre.AñoCursada. Ej: Pérez.Juan.2018. No se corregirán proyectos que no sea identificable su autor.
- **Salvo que se indique lo contrario, TODAS** las clases deberán ir en una Biblioteca de Clases llamada Entidades.
- No se corregirán exámenes que no compilen.
- **Reutilizar** tanto código como crean necesario.

Al finalizar, colocar la carpeta de la Solución completa en un archivo ZIP que deberá tener como nombre

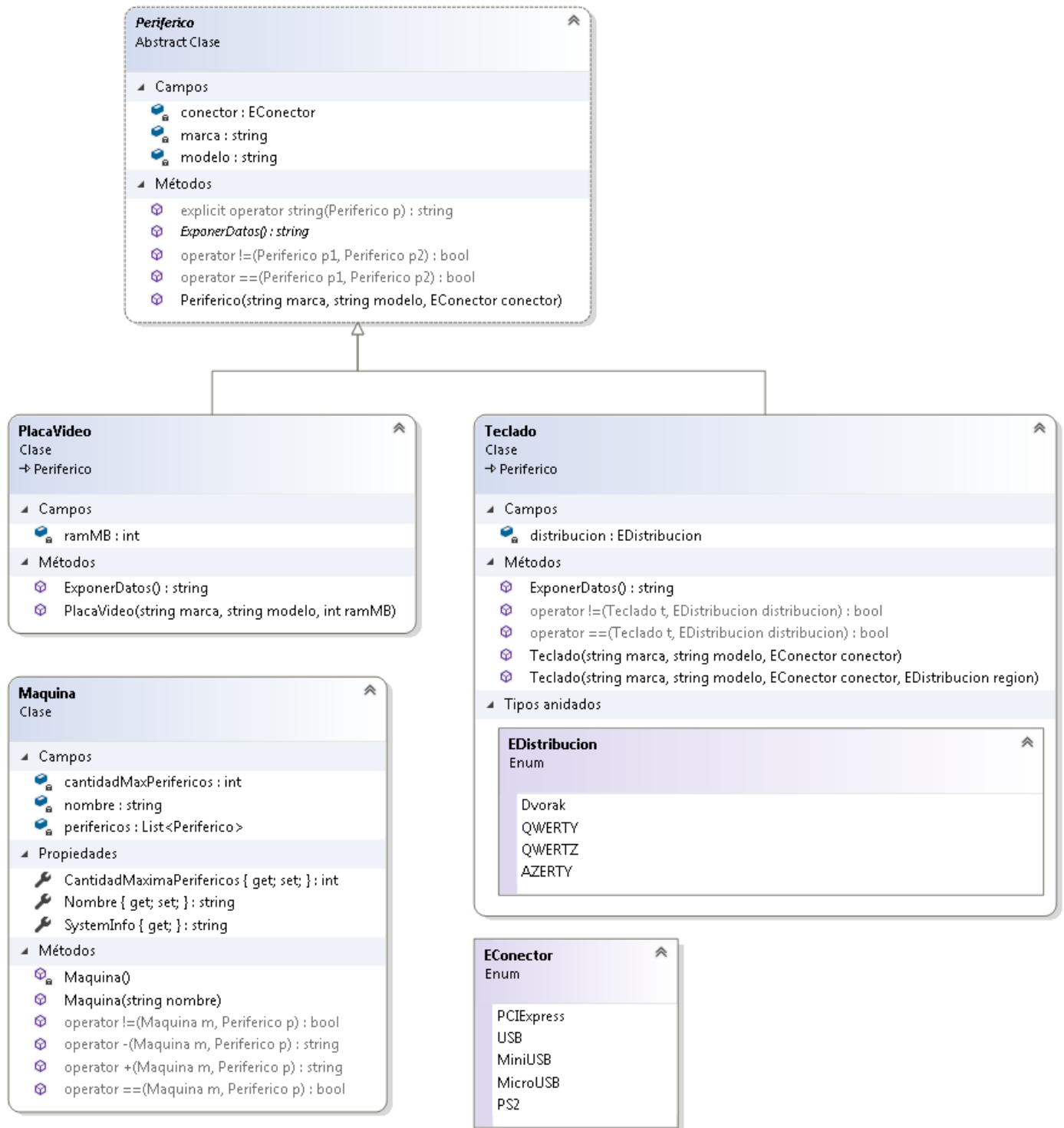
Apellido.Nombre.AñoCursada.zip y dejar este último en el Escritorio de la máquina. Luego presionar el botón de la barra superior, **colocar un mensaje** y presionar *Aceptar*. **Aguardar a que el profesor indique que el examen fue copiado de forma correcta.** Luego retirarse del aula.

---

*TIEMPO MÁXIMO PARA RESOLVER EL EXAMEN 90 MINUTOS.*

---

1. Crear una Solución llamada "20190509 – [Apellido].[Nombre]" siendo el Apellido y Nombre del alumno que lo crea.
2. Dentro crear 3 proyectos: uno del tipo *Consola* llamado **Test**, otro de *Formulario* llamado **GUI** y un último de *Biblioteca de Clases* con el nombre de **Entidades**.
3. Modelar los elementos necesarios para cumplir con el siguiente diagrama de clases:



4. Periféricos: clase abstracta con 3 atributos.
  - a. Dos Periféricos serán **iguales** siempre y cuando tengan la misma **marca** y **modelo**.
  - b. El operador explícito retornará toda la información del periférico (utilizar `StringBuilder` y `string.Format`).
  - c. **ExponerDatos** será abstracto.
5. Teclado: hereda de Periférico y tiene 1 atributo propio.
  - a. De no asignarse ninguna distribución al constructor, esta será **Dvorak**.
  - b. **ExponerDatos** retornará toda la información del Teclado (ejemplo en Salida por pantalla).
6. PlacaVideo: hereda de Periférico y tiene 1 atributo propio.
  - a. El conector será siempre **PCIExpress**.
  - b. **ExponerDatos** retornará toda la información de la Placa de Video (ejemplo en Salida por pantalla).
7. Máquina: 3 atributos propios.
  - a. El constructor privado inicializará la lista y asignará 3 al atributo **cantidadMaxPerifericos**.

- b. La propiedad **CantidadMaximaPerifericos** validará que la cantidad no sea menor a 1 ni mayor a 4. Si está fuera de dicho rango, asignará el valor más bajo o más alto según corresponda al atributo; caso contrario asignará el valor recibido.
- c. La propiedad **SystemInfo** retornará toda la información de la máquina, incluida la de sus periféricos, según corresponda su tipo (Teclado o PlacaVideo).
- d. El operador == entre una Maquina y un Periférico será verdadero si el elemento ya forma parte de la máquina, según la comparación descrita anteriormente.
- e. El operador + agregará un nuevo periférico, siempre y cuando este no se encuentre con anterioridad en la lista ni se haya superado la cantidad máquina permitida.
  - i. Si lo logra, retornará: "Periferico conectado!".
  - ii. Caso contrario retornará: "No se puede conectar el dispositivo.".
- f. El operador - quitará el periférico indicado de la lista.
  - i. Si lo logra, retornará: "Periferico desconectado!".
  - ii. Caso contrario retornará: "No se puede desconectar el dispositivo.".

8. Agregar en el Main el siguiente código:

```
ConsoleColor aux = ConsoleColor.Green;

Maquina m = new Maquina("LAB01PC03");

// Agrega
Console.WriteLine(m + new PlacaVideo("ASUS", "P360-A", 64));
// Agrega
Console.WriteLine(m + new Teclado("GENIUS", "J961-P", EConector.PS2,
Teclado.EDistribucion.AZERTY));
// Agrega
Console.WriteLine(m + new Teclado("GENIUS", "JRR-10", EConector.USB));
// No agrega por cantidad superada
Console.WriteLine(m + new PlacaVideo("SENTEY", "P361-A", 32));

// Cambio Max Perisfericos y agrego
m.CantidadMaximaPerifericos = 5;
Console.WriteLine(m + new PlacaVideo("SENTEY", "P361-A", 32));

Console.ForegroundColor = ConsoleColor.Green;
Console.WriteLine("Cantidad Máxima Perisfericos: " + m.CantidadMaximaPerifericos);
Console.ForegroundColor = aux;

// Quito una
Console.WriteLine(m - new PlacaVideo("ASUS", "P360-A", 64));
// No agrega por repetida
Console.WriteLine(m + new Teclado("GENIUS", "J961-P", EConector.PS2,
Teclado.EDistribucion.AZERTY));
// Agrega
Console.WriteLine(m + new PlacaVideo("ASUS", "P360-A", 64));

Console.WriteLine("\n-----");
Console.ForegroundColor = ConsoleColor.Yellow;
Console.WriteLine(new PlacaVideo("ASUS", "P360-A", 64).ExponerDatos());
Console.WriteLine(new Teclado("GENIUS", "JRR-10", EConector.USB).ExponerDatos());
Console.ForegroundColor = aux;

Console.WriteLine("\n-----");
Console.ForegroundColor = ConsoleColor.Red;
Console.WriteLine(m.SystemInfo);
Console.ForegroundColor = aux;

Console.ReadKey();
```

Logrando la siguiente salida por pantalla:

```

Periferico conectado!
Periferico conectado!
Periferico conectado!
No se puede conectar el dispositivo.
Periferico conectado!
Cantidad Máxima Perisfericos: 4
Periferico desconectado!
No se puede conectar el dispositivo.
Periferico conectado!

-----
PLACA DE VIDEO ASUS P360-A <Puerto: PCIExpress>
Memoria 64MB

TECLADO GENIUS JRR-10 <Puerto: USB>
Distribución Dvorak

-----
LAB01PC03:
TECLADO GENIUS J961-P <Puerto: PS2>
Distribución AZERTY

TECLADO GENIUS JRR-10 <Puerto: USB>
Distribución Dvorak

PLACA DE VIDEO SENTEY P361-A <Puerto: PCIExpress>
Memoria 32MB

PLACA DE VIDEO ASUS P360-A <Puerto: PCIExpress>
Memoria 64MB

```

9. Generar el siguiente formulario, logrando la misma funcionalidad:
  - a. El título del formulario serán los datos del alumno.
  - b. Agregar ingresará un nuevo elemento a una máquina ya instanciada en el constructor.
  - c. Sacar buscará y quitará una máquina.
  - d. Mostrar el resultado de cada operación (agregar y sacar) en un MessageBox.
  - e. System Info mostrará en el RichTextBox toda la información de la máquina.

**FrmPC**  
 Clase  
 → Form

**Campos**

- btnAgregar : Button
- btnSacar : Button
- btnSystemInfo : Button
- components : IContainer
- groupBox1 : GroupBox
- lblMarca : Label
- lblMemoria : Label
- lblModelo : Label
- nudMemoria : NumericUpDown
- rbtSystemInfo : RichTextBox
- txtMarca : TextBox
- txtModelo : TextBox

**Métodos**

- Dispose(bool disposing) : void
- FrmPC()
- InitializeComponent() : void