Programación III - Guía de Actividades

El siguiente documento incluye un compendio de las actividades principales requeridas a los alumnos durante el cursado de la materia, estas pueden ser solicitadas de manera sincrónica durante la clase (en aula informática) o de manera asincrónica a través del aula virtual. Dado a que el espíritu de la asignatura es desarrollar los saberes vinculados con la Programación Orientada a Objetos desde un punto de vista teórico y práctico, las actividades en general se dividen en preguntas orientadoras y en ejercicios prácticos, aunque para estos últimos se necesita el complemento de la asignatura Laboratorio de Computación III que se dicta de manera concurrente.

Actividad 1: Repaso de conceptos del Paradigma Orientado a Objetos (POO)

Utilizando la bibliografía indicada en clase conteste las siguientes preguntas orientadoras

1. ¿Qué es un objeto?
   a. Describa sus características
   b. ¿Qué puede ser un objeto?
2. Explique los conceptos de:
   a. Mensaje
   b. Colaboración
   c. Responsabilidad
3. ¿Qué es una clase?
   a. ¿Cómo se compone?

Actividad 2: Modelado inicial de una clase

Proponer atributos y métodos para las clases:

1. Ventilador
2. Candado

Actividad 3: Modelado de clases y relaciones

Continuando con la última actividad ¿Podría plantear dos clases diferentes y relacionarlas?
Utilice UML para realizar el diagrama de clases.

Actividad 4: Pilares del POO

Explique los pilares del POO

1. Abstracción
2. Encapsulamiento
   a. ¿Cómo se vincula con la abstracción?

Actividad 5: Modelado de clases y relaciones II

Modele un sistema para consulta de viajes de ómnibus simple, identifique clases, atributos y asociaciones.

1. Los viajes tienen una ciudad de origen y otra de destino, se realizan en determinadas fechas y poseen cierta duración
2. El precio del boleto depende de la categoría
3. Cada viaje lo realiza una empresa en particular
4. Utilice UML para realizar el diagrama de clases.

Actividad 6: Pilares del POO II

Explique los pilares del POO

1. Herencia
   a. ¿Qué tipos existen?
2. Polimorfismo
   a. ¿Cómo se logra?
   b. Ejemplifique

Actividad 7: Modelado de clases y relaciones III

Modele un sistema para consulta de viajes de ómnibus simple, identifique clases, atributos y asociaciones.

1. Los viajes tienen una ciudad de origen y otra de destino, se realizan en determinadas fechas y poseen cierta duración
2. El precio del boleto depende de la categoría
3. Cada viaje lo realiza una empresa en particular
4. Utilice UML para realizar el diagrama de clases.

Actividad 8: Prehistoria de .NET

Explique brevemente de que se trata cada una de las tecnologías que precedieron a .NET. Ventajas y Desventajas.

1. C/Windows API
2. C++/MFC
3. Visual Basic 6.0
4. Java
5. COM

Actividad 9: La filosofía de .NET

Responda las siguientes preguntas

1. ¿Qué diferencia existe entre una plataforma y un framework?
2. En particular, ¿qué es el .NET Framework y qué tipo de aplicaciones se pueden desarrollar?

Actividad 10: .NET

Responda las siguientes preguntas acerca de .NET Framework/.NET Core/.NET

1. ¿Cuáles son sus dos partes principales?
   a. Descríbalas brevemente
2. ¿Qué es el CTS y cuál es su importancia?
   a. ¿Cómo se vincula con la CLS?
3. ¿Qué versiones del framework fueron liberadas en el tiempo?
   a. Mencione 2-3 características de cada una de ellas

Actividad 11: Lenguajes de .NET

Responda las siguientes preguntas acerca de .NET Framework/.NET Core/.NET

1. Mencione algunos lenguajes que funcionan sobre .NET
   a. ¿Por qué cree que .NET es multilenguaje?
2. ¿Qué es un ensamblado?
3. Caracterice a C# y sus versiones

Actividad 12: IDE

1. ¿Qué es Visual Studio?
2. Ábralo y recórralo. Luego enumere las características encontradas
3. ¿Cuál es la estructura básica de soluciones y proyectos?

Actividad 13: Primera Aplicación de Consola en C#

Con el objetivo de comprender el uso de variables, asignaciones y operaciones de forma simple en C#, creará una aplicación de consola con las siguientes características:

1. Solicite el ingreso de dos valores por pantalla
2. Al final muestre en forma descriptiva los resultados de aplicar las cuatro operaciones básicas

Actividad 14: Conceptos básicos en C#

Considerando lo realizado en la actividad anterior y mediante la búsqueda en las fuentes bibliográficas indicadas:

1. Defina y caracterice las variables en C#
2. Mencione 4 tipos de datos y explíquelos
3. ¿Qué es el ámbito de una variable?
4. ¿De qué formas puedo realizar comentarios en el código?

Actividad 15: Estructuras de Decisión

Con el objetivo de comprender la aplicación de las estructuras de decisión en C#, creará una aplicación de consola con las siguientes características:

1. Solicite el ingreso de un texto y controle que no esté vacío.
2. Despliegue un menú que muestre 3 posibilidades (Texto en mayúscula, Texto en minúscula y Texto Original).
3. Capture la opción seleccionada con Console.Readkey y realice la acción solicitada.

Actividad 16: Estructuras de Decisión II

Considerando lo realizado en la actividad anterior y mediante la búsqueda en las fuentes bibliográficas indicadas conteste:

1. ¿De qué formas se puede utilizar if?
2. ¿Cuáles son los tipos de operadores lógicos y de evaluación más comunes?
    a. ¿A qué se refiere la evaluación por cortocircuito?
3. ¿Qué ventajas tiene el operador “?”?
4. ¿En qué se diferencian switch e if?
    a. ¿En qué casos utilizaría el primero?

Actividad 17: Estructuras Repetitivas

Con el objetivo de comprender la aplicación de las estructuras repetitivas en C#, creará una aplicación de consola con las siguientes características:

1. Calcule la media y la desviación estándar de un conjunto de 10 personas.
2. Tome por teclado la altura en cm de cada persona y cárguela en un arreglo.
3. Presente los resultados obtenidos.
4. Muestre que alturas se encuentran por encima de la media y por debajo de ella.
5. Muestre que alturas se encuentran dentro del rango definido por la desviación estándar.

Actividad 18: Estructuras Repetitivas II

Considerando lo realizado en la actividad anterior y mediante la búsqueda en las fuentes bibliográficas indicadas, conteste:

1. ¿Qué estructuras repetitivas presenta C#?
2. ¿En qué se diferencian?
3. ¿Para que usan break y continue?

Actividad 19: Ensamblados

Responda las siguientes preguntas acerca de .NET Framework/.NET Core/.NET

1. ¿Qué es un ensamblado?
2. ¿Qué funciones realiza?
3. Describa su estructura

Actividad 20: Ensamblados II

Responda las siguientes preguntas acerca de .NET Framework/.NET Core/.NET

1. ¿Según su disponibilidad entre aplicaciones como se clasifican los ensamblados?
2. ¿Qué es la GAC?
3. ¿Cuál es el proceso de ejecución de del código administrado?
4. ¿Cuáles son los principales tipos de ensamblados compilados?

Actividad 21: Ensamblados III

Escoja uno de los proyectos realizados y descompílelo.

1. ¿Qué herramienta usaría?
2. Identifique las partes del ensamblado.

Actividad 22: Clases

Con el objetivo de comprender los conceptos de OO mediante C#, creará una aplicación de consola con las siguientes características:

1. Permita apostar el resultado de arrojar un dado.
2. Toma el valor de 1 a 6.
3. Arroja el resultado aleatorio entre 1 y 6 indicando si ganó.
4. Respete el siguiente diagrama.

Actividad 22: Clases II

Responda las siguientes preguntas acerca de clases en C#

1. ¿Cómo se denominan a los atributos de una clase?
2. ¿Qué diferencias sintácticas existen entre funciones y procedimientos?
3. Mencione las características más importantes de los constructores.

Actividad 23: Métodos y sus modificadores

Cree una aplicación de consola con las siguientes características:

1. Implementar una función Moda, que reciba como valores una cantidad indeterminada de enteros y devuelva la moda (estadística), el valor mínimo y el valor máximo.
2. Invocar este método y mostrar los resultados por consola.
3. Elija los modificadores más adecuados

Luego responda:

4. ¿Qué tipos de modificadores de parámetros permite usar C#?
5. ¿En qué se diferencian?

Actividad 24: Clases III

Modifique el juego anterior implementando las siguientes características:

1. Dos jugadores, Dos dados y Apuestas
2. Tres modos de apuesta {conservador -1/2, arriesgado -2/5, desesperado -4/15}
3. Cada jugador cuenta con $100 iniciales y un pozo de $10000.
4. El juego termina cuando el pozo o el saldo de algún jugador llega a cero.
5. Use clases (hasta 5), propiedades, enumeraciones o contantes donde necesite, implemente asociaciones entre clases.

Luego responda:

6. ¿En qué se diferencian las propiedades, de los campos y métodos? ¿Cuál es su estructura básica?
7. ¿Cómo se declara y usa una constante?
8. ¿Qué son las enumeraciones? ¿Qué características presentan?

Actividad 25: Pasaje de arreglos por parámetros

Ejecute y analice el siguiente código:

```
static void Main()
{
    int[] arr = {1, 4, 5};
    Console.WriteLine("Dentro de Main, antes de llamar al
    método, el primer elemento es: {0}", arr [0]);
    Cambiar(arr);
    Console.WriteLine("Dentro de Main, luego de llamar al
    método, el primer elemento es: {0}", arr [0]);
}
static void Cambiar(int[] pArray)
{
    pArray[0] = 888;
    pArray = new int[5] {-3, -1, -2, -3, -4};
    Console.WriteLine("Dentro del método,
     el primer elemento es: {0}", pArray[0]);
}
```

Luego responda:

1. ¿Qué valores obtendría en cada salida?
2. ¿Qué tipos de arreglos soporta C#?
3. Explique cómo funciona el pasaje por valor y por referencia de un arreglo.

Actividad 26: Cadenas

Cree una aplicación de consola con las siguientes características:

1. Dado un texto de entrada, permita cifrarlo o descifrarlo
2. Utilizando el cifrado de César
3. Alfabeto de base: "aábcdeéfghiíjklmnñoópqrstuúüvwxyz"

Luego responda:

4. ¿Por qué se dice que las cadenas (string) son inmutables?
5. ¿Para qué sirven las funciones Split y Replace?
6. ¿Qué diferencia existe entre la concatenación y Join?
7. ¿Qué son los caracteres de escape? ¿Y para que se usa el verbatim?
8. ¿Cuáles son las ventajas y desventajas de StringBuilder?

Actividad 27: Manejo de fechas y tiempo

Realice una clase utilitaria de manejo de tiempo y fechas que cumpla con los siguiente:

1. Que contegan los métodos:
    a. ObtenerDiasCalendario() obtiene los días entre dos fechas
    b. ObtenerDiasLaborables() obtiene los días laborables entre dos fechas
    c. SumarDiasLaborables() obtiene una fecha sumando una cantidad de días a una fecha inicial
2. Considere fines de semanas y feriados
3. Puede guardar los feriados en un arreglo

Luego responda:

4. ¿Para qué se utiliza el tipo System.DateTime?
5. ¿Para qué se utiliza el tipo System.TimeSpan?
6. ¿Cómo se relacionan ambos?
7. ¿Qué características presentan?

Actividad 28: Depuración y errores

Tome el último proyecto realizado, compílelo y luego

1. Coloque puntos de interrupción
2. Ejecútelos paso a paso
3. Visualice el contenido de las variables
4. Realice modificaciones durante la ejecución
5. Haga que la ejecución falle
6. Utilice Try Catch para capturar los errores en tiempo de ejecución

Luego responda:

7. Defina depuración (debugging)
8. ¿Qué perfiles de compilación existen por defecto?
9. Enumere y describa los tipos de errores que pueden presentarse en una aplicación
10. ¿Qué herramientas posee VS para ayudar a descubrir y solucionar errores?
11. ¿Para qué sirve Try .. Catch, de que formas se puede utilizar?