

ISI-BSI-11

Programación IV



Kevin Montes Varela



kevin.montes@uamcr.net

kevin.montes@kmontes.com



Ingeniería de Sistemas



Tarea 2



UNIVERSIDAD AMERICANA

Enunciado Tarea 2

Consiste en una tarea desarrollada de forma individual en donde debe crear el funcionamiento de los atributos y procesos de diferentes atracciones de un parque de diversiones.

El ejercicio debe realizarse en una solución de Visual Studio bajo la estructura de capas (BLL, DAL, PL).

El estudiante puede hacer uso del proyecto base que proporciona la herramienta.

```
class Program
{
    #region VARIABLES PRIVADAS
    private float _fNumA, _fNumB, _fResult;
    private char _cSimbolo;
    private string _sMsjError;
    #endregion

    #region OBJETOS O CONSTRUCTORES O VARIABLES PUBLICAS
    /// <summary>
    /// fNumA --> variable utilizada para obtener el primer numero de
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public float fNumA
    /// <summary>
    /// fNumB -->
    /// la calculadora
    /// </summary>
    5 references
    public float fNumB
    /// <summary>
    /// fResult -->
    /// la calculadora
    /// </summary>
    5 references
    public float fResult
    /// <summary>
    /// cSimbolo -->
    /// el simbolo de la operacion
    /// </summary>
    5 references
    public char cSimbolo
    /// <summary>
    /// sMsjError -->
    /// el mensaje de error
    /// </summary>
    5 references
    public string sMsjError
    #endregion
}
```



Enunciado Tarea 1 (Semana 3)

Selección de Atracción:

El usuario podrá seleccionar una atracción visible en la pantalla. Al seleccionar la atracción el sistema debe cargar en pantalla la información asociada a dicho objeto en una tabla.

La información a mostrar es:

- Nombre de la atracción
- Tipo de atracción (montaña rusa, carrusel, rueda de la fortuna)
- Capacidad de personas
- Duración del viaje (1 minuto 2 minutos)
- Disponibilidad (operativa/fuera de servicio)
- Horario de operación
- Estado (Encendida/Apagada/Abierta/Cerrada/En Mantenimiento)

***Esto corresponde al enunciado de la tarea 1 ya finalizada en semana 3. Es información para repasar la primera parte de la tarea. No será evaluado en esta tarea 2.

```
class JS_Calculos_DMC
{
    #region VARIABLES PRIVADAS
    private float _fNumA, _fNumB, _fResult;
    private char _cSimbolo;
    private string _sMsjError;
    #endregion

    #region OBJETOS O CONSTRUCTORES O VARIABLES PUBLICAS
    /// <summary>
    /// fNumA --> variable utilizada para obtener el primer numero de
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public float fNumA
    /// <summary>
    /// fNumB -->
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public float fNumB
    /// <summary>
    /// fResult -->
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public float fResult
    /// <summary>
    /// cSimbolo -->
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public char cSimbolo
    /// <summary>
    /// sMsjError -->
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public string sMsjError
    #endregion
}
```



Enunciado Tarea 2

Acciones de Atracción:

El usuario podrá seleccionar una acción a realizar de la atracción por medio de botones visibles en la pantalla.

Las acciones disponibles son:

- Iniciar atracción
- Detener atracción
- Realizar mantenimiento
- Abrir atracción
- Cerrar atracción
- Consultar disponibilidad (Mensaje en pantalla)

La información a mostrar en la tabla debe ser:

- Acción seleccionada, Tipo de Atracción, Nombre de Atracción, Estado

```
class cls_calculos_BMC
#region VARIABLES PRIVADAS
    private float _fNumA, _fNumB, _fResult;
    private char _cSimbolo;
    private string _sMsjError;
#endregion

#region OBJETOS O CONSTRUCTORES O VARIABLES PUBLICAS
    /// <summary>
    /// fNumA --> variable utilizada para obtener el primer numero de
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public float fNumA
    /// <summary>
    /// fNumB -->
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public float fNumB
    /// <summary>
    /// fResult -->
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public float fResult
    /// <summary>
    /// cSimbolo -->
    /// el símbolo de la operación matemática que se desea realizar
    /// </summary>
    5 references
    public char cSimbolo
    /// <summary>
    /// sMsjError -->
    /// el mensaje de error que se muestra cuando se intenta realizar una operación
    /// que no es válida
    /// </summary>
    5 references
    public string sMsjError
#endregion
```



Enunciado Tarea 2

Restricciones:

El usuario no podrá realizar ninguna acción si no ha seleccionado una atracción.

El usuario no podrá realizar ninguna acción si la disponibilidad de la atracción es fuera de servicio.

Para realizar las acciones de iniciar o detener atracción esta debe estar abierta.

No se puede realizar mantenimiento a la atracción si esta se encuentra abierta.

No se puede abrir, iniciar o detener la atracción si esta se encuentra en mantenimiento.

```
class cls_calculos_BMC
#region VARIABLES PRIVADAS
    private float _fNumA, _fNumB, _fResult;
    private char _cSimbolo;
    private string _sMsjError;
#endregion

#region OBJETOS O CONSTRUCTORES O VARIABLES PUBLICAS
    /// <summary>
    /// fNumA --> variable utilizada para obtener el primer numero de
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public float fNumA
    /// <summary>
    /// fNumB -->
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public float fNumB
    /// <summary>
    /// fResult -->
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public float fResult
    /// <summary>
    /// cSimbolo -->
    /// el símbolo de la operación matemática que se desea realizar
    /// </summary>
    5 references
    public char cSimbolo
    /// <summary>
    /// sMsjError -->
    /// el mensaje de error que se muestra cuando se intenta realizar una
    /// operación matemática que no es válida
    /// </summary>
    5 references
    public string sMsjError
#endregion
```



Enunciado Tarea 2

Consideraciones:

El usuario debe tener un insumo visual que le indique que tipo de atracción está seleccionando.

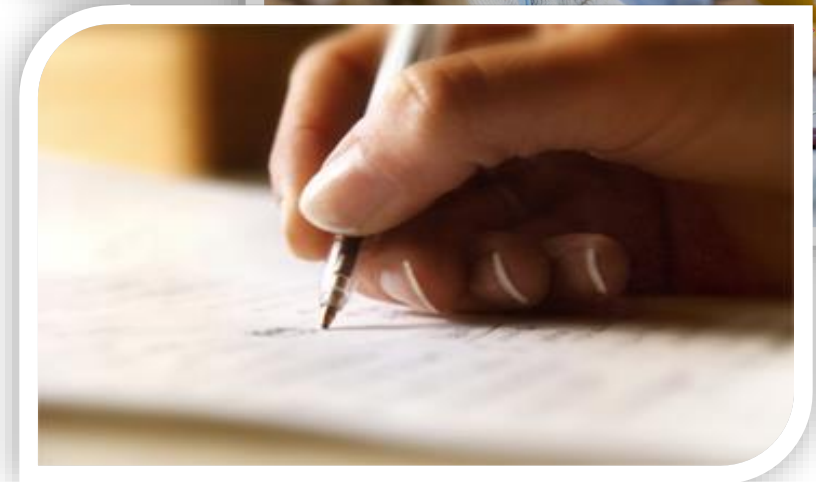
Como mínimo debe existir 6 atracciones.

Debe hacer uso del paradigma de programación orientada a objetos y la estructura en capas de la solución.

Puede hacer uso de hojas de estilo personalizadas y cualquier otro recurso que considere necesario para mejorar la estética de la interface de usuario.

```
class fcs_calculos_DMC
#region VARIABLES PRIVADAS
    private float _fNumA, _fNumB, _fResult;
    private char _cSimbolo;
    private string _sMsjError;
#endregion

#region OBJETOS O CONSTRUCTORES O VARIABLES PUBLICAS
    /// <summary>
    /// fNumA --> variable utilizada para obtener el primer numero de
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public float fNumA
    /// <summary>
    /// fNumB -->
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public float fNumB
    /// <summary>
    /// fResult -->
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public float fResult
    /// <summary>
    /// cSimbolo -->
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public char cSimbolo
    /// <summary>
    /// sMsjError -->
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public string sMsjError
#endregion
```



Enunciado Tarea 2

La Solución debe ser entregada en el campus virtual en un único archivo comprimido (.rar | .zip).

Fecha límite de Entrega 13/02/2026 – 11:59 pm

Entrega de forma individual. Las que sean exactamente iguales serán anuladas así como aquellas en las que no se respeta la nomenclatura del archivo a entregar.

Valor de Tarea : 2%

Tarea 2 de 10

```
class fcs_calculos_DMC
#region VARIABLES PRIVADAS
    private float _fNumA, _fNumB, _fResult;
    private char _cSimbolo;
    private string _sMsjError;
#endregion

#region OBJETOS O CONSTRUCTORES O VARIABLES PUBLICAS
    /// <summary>
    /// fNumA --> variable utilizada para obtener el primer numero de
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public float fNumA
    /// <summary>
    /// fNumB -->
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public float fNumB
    /// <summary>
    /// fResult -->
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public float fResult
    /// <summary>
    /// cSimbolo -->
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public char cSimbolo
    /// <summary>
    /// sMsjError -->
    /// la calculadora para las operaciones matemáticas del usuario
    /// </summary>
    5 references
    public string sMsjError
#endregion
```



Enunciado Tarea 2

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Progra4_Tarea2
{
    class Calculadora
    {
        #region VARIABLES PRIVADAS
        private float _fNumA, _fNumB, _fResult;
        private char _eSimbolo;
        private string _sMsjError;
        #endregion

        #region OBJETOS O CONSTRUCTORES O VARIABLES PUBLICAS
        /// <summary>
        /// fNumA --> variable utilizada para obtener el primer numero de
        /// la calculadora para las operaciones matemáticas del usuario
        /// </summary>
        public float fNumA { get => _fNumA; set => _fNumA = value; }
        /// <summary>
        /// fNumB --> variable utilizada para obtener el segundo numero de
        /// la calculadora para las operaciones matemáticas del usuario
        /// </summary>
        public float fNumB { get => _fNumB; set => _fNumB = value; }
        /// <summary>
        /// fResult --> variable utilizada para obtener el resultado de
        /// la calculadora para las operaciones matemáticas del usuario
        /// </summary>
        public float fResult { get => _fResult; set => _fResult = value; }
        #endregion

        #region METODOS
        /// <summary>
        /// Metodo para realizar la suma de dos numeros
        /// </summary>
        public float Suma()
        {
            return fNumA + fNumB;
        }

        /// <summary>
        /// Metodo para realizar la resta de dos numeros
        /// </summary>
        public float Resta()
        {
            return fNumA - fNumB;
        }

        /// <summary>
        /// Metodo para realizar la multiplicación de dos numeros
        /// </summary>
        public float Multiplicacion()
        {
            return fNumA * fNumB;
        }

        /// <summary>
        /// Metodo para realizar la división de dos numeros
        /// </summary>
        public float Division()
        {
            return fNumA / fNumB;
        }

        /// <summary>
        /// Metodo para mostrar el mensaje de error
        /// </summary>
        public void MostrarMensaje()
        {
            Console.WriteLine(sMsjError);
        }
        #endregion
    }
}
```

Solución de Visual Studio



Nombre del Estudiante



Progra4_Tarea2_KevinMontesVarela.rar



Cierre
