



Universidad Tecnológica Metropolitana

Código	<b>INSTRUMENTO DE EVALUACIÓN</b>	Revisión:
<b>F-SGC-033</b>		<b>00</b>

### DATOS GENERALES DEL INSTRUMENTO.

División: **TIC**  
FDC\*/Carrera: **Desarrollo de Software Multiplataforma**  
Asignatura: **Estructuras de Datos Aplicadas**  
Cuat.-Gpo(s): **4A, 4B, 4C, 4D, 4E, 4F** Fecha de aplicación: **Octubre/2024**  
Unidad(es) de aprendizaje y/o tema(s) a evaluar.

**III. Listas.**  
**IV. Pilas y Colas.**

Especificar con una "X" el tipo de instrumento de evaluación a utilizar (señalar sólo uno).

Tec. evaluación para el SABER			Tec. evaluación para el SABER HACER + SER		
<input type="checkbox"/>	Prueba oral (entrevista)	<input type="checkbox"/>	<input type="checkbox"/>	Otro (Especificar):	<input type="checkbox"/>
<input type="checkbox"/>	Prueba escrita	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Proyectos	<input type="checkbox"/>
<input type="checkbox"/>	Trabajo investigación	<input type="checkbox"/>	<input type="checkbox"/>	Prácticas, ejercicios, demostraciones	<input type="checkbox"/>
<input type="checkbox"/>	Ensayo, informe	<input type="checkbox"/>	<input type="checkbox"/>	Rúbrica	<input type="checkbox"/>
<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Lista de cotejo	<input type="checkbox"/>
<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Guía de observación	<input type="checkbox"/>

Profesor(es) de la asignatura: **ISC. Ruth Betsaida Martínez Domínguez, MGTI.**  
**ISC. Mirian Magaly Canche Caamal, MGTI.**

Nombre del alumno: Hacer referencia a la lista de asistencia Octubre - Noviembre de 2024 Calificación (puntaje): 25%

### CONTENIDO DEL INSTRUMENTO DE EVALUACIÓN

#### SEGUNDO PARCIAL

**INSTRUCCIONES:** Dado los siguientes problemas, analiza y desarrolla el programa en el lenguaje de programación C#

**Práctica 1.** Productos disponibles y retirados Un supermercado nos pide que hagamos una pequeña aplicación que almacene los productos pasados por el escáner. La aplicación debe almacenar Productos (clase), cada producto al crearse contiene una cantidad, un precio (estos dos generados aleatoriamente). El nombre del producto será básico (producto1, producto2, producto3, etc.), se solicita de igual forma que la misma aplicación permita retirar elementos del mismo almacén de productos al igual que mostrar la cantidad de productos disponibles y retirados (Para resolver este problema utiliza una lista).

**Práctica 2.** Pares e Impares Crea una aplicación que genere una lista de números aleatorios, posterior a esto la aplicación debe de mostrar todos los números pares en una lista y todos los números impares en otra lista.

**Práctica 3.** Aprobados y Reprobados Crea una aplicación que solicite nombres de un alumno y sus respectivas calificaciones y los almacene en una lista, dicha aplicación tendrá que sacar a los alumnos que reprobaron con una calificación menor que siete y almacenarlos nuevamente en una lista de reprobados, la aplicación debe de mostrar a los alumnos que aprobaron y los alumnos que aprobaron.

\*FDC: familia de carreras

Código	INSTRUMENTO DE EVALUACIÓN	Revisión:
F-SGC-033		00

**Práctica 4.** Eliminar y ordenar productos Crea una aplicación que permita agregar productos a una lista, los productos deben de tener un precio, la aplicación debe de eliminar los productos de acuerdo a una clave de producto solicitada al usuario, además de eso la aplicación debe de mostrar los elementos ordenados por nombres y mostrar el costo total de compra (crea una lista para almacenar los datos).

**Práctica 5.** Listar palabras por letra Crea una aplicación para almacenar palabras (solicitadas al usuario), la aplicación debe de separar las palabras y dividir las en listas clasificadas por la primera letra de la palabra es decir todas las palabras que coincidan en la primera letra, se muestra un ejemplo:

Lista1: Fresa, Fresca, Orilla, Oso, Peso, Pez

Lista2: Fresa, Fresca

Lista3: Orilla, Oso

Lista4: Peso, Pez

**Nota:** En caso de que el usuario agregue más palabras con letras diferentes la aplicación tiene que crear nuevas listas en caso de no existir las listas con la letra inicial.

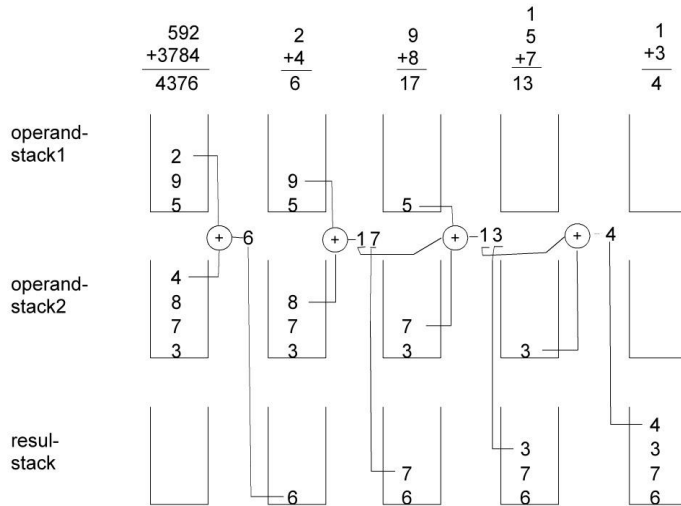
**Práctica 6.** Realiza una aplicación que lea una palabra e invierta el orden, por ejemplo, si recibes la palabra UTM el programa debe de invertirla y mostrar MTU.

**Práctica 7.** Un palíndromo es una palabra, número o frase que se lee igual hacia adelante que hacia atrás, algunos ejemplos de palíndromos son las palabras “ana”, arenera, arepera, anilina, Malayalam, Oruro, oso, radar, reconocer, rotor, salas, seres, somos, sometemos, entre otras. Realiza un programa que lea una palabra e indique si se trata de un palíndromo o no.

**Práctica 8.** Realiza una aplicación que mediante pilas sume números enteros muy grandes, puedes leer los números como string, ingresarlos cada uno de ellos a una pila y realizar la suma.

## Pilas

Otra aplicación de las pilas es un algoritmo para sumar números enteros relativamente grandes que no pueden ser representados mediante algún tipo de datos, por ejemplo sumar 592+3784



**Práctica 9.** Reemplazar valor Escribir una función Reemplazar que tenga como argumentos una pila con tipo de elemento int y dos valores int: nuevo y viejo de forma que si el segundo valor aparece en algún lugar de la pila, sea reemplazado por el segundo.

### Práctica 10. VENTANILLA DE UN BANCO

Desarrolle un programa que administre una Cola de una ventanilla de un banco que tenga la siguiente forma:

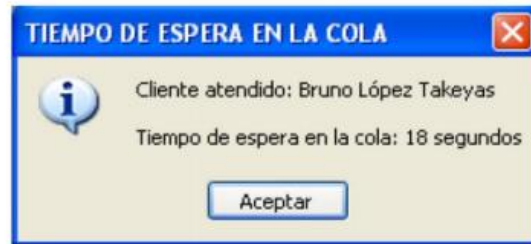
#### Requerimientos generales:

- El programa debe almacenar los siguientes datos de cada cliente: o Número de turno (entero) o Nombre del cliente (string) o Tipo de movimiento que desea realizar (string) o Hora de llegada del cliente (DateTime)
- Cuando llegue un cliente al banco, debe formarse al final de la cola y se almacenan los datos antes mencionados.
- Seleccione el tipo de movimiento de un comboBox:

- Al oprimir el botón “Atender en ventanilla”, se elimina el primer cliente de la cola y se debe calcular el tiempo en espera de ser atendido.
- **Inserción de clientes en la Cola (InsertaCola):**
  - Validar que no exista sobreflujo; es decir, que no esté llena la estructura cuando se intente insertar un nuevo dato. Utilice el método ColaLlena() que devuelve el valor true si la cola está llena y false en caso contrario.
  - Cuando inserte un cliente en la cola, actualice el dataGridView1 y muestre una ventana como la sig:

- **Eliminación de datos (bajas):**
  - Validar que la estructura no esté vacía antes de solicitar el dato que se desea eliminar. Utilice el método ColaVacia() que devuelve el valor true si la cola está vacía y false en caso contrario.
  - Cuando se atiende un cliente en ventanilla (EliminaCola( )), obtenga la hora del sistema para calcular el tiempo de espera en la cola de acuerdo a su hora de llegada.

Código	INSTRUMENTO DE EVALUACIÓN	Revisión:
F-SGC-033		00

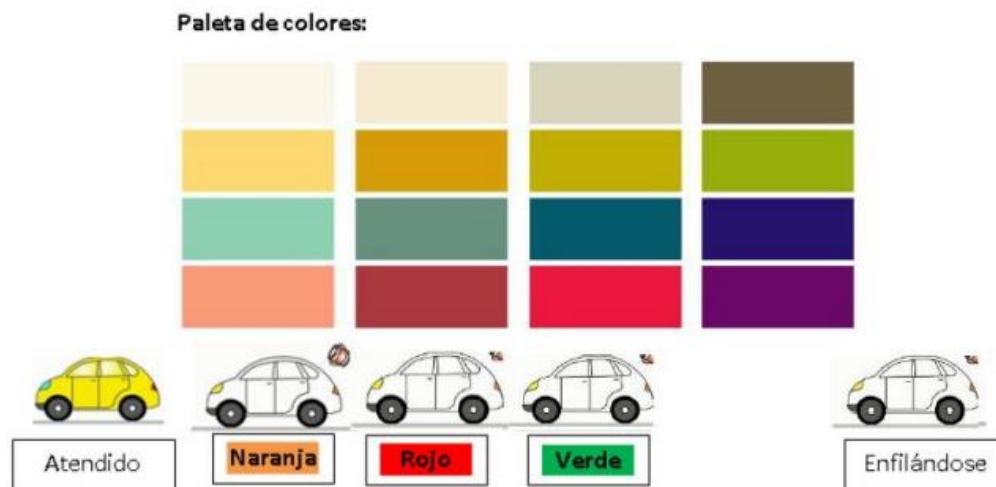


- **Mostrar la estructura:**
  - Después de cada operación, actualice el dataGridView1
  - Muestre el valor de frente y final.

### Práctica 11. JUEGO PINTAR COCHES

Realiza el siguiente juego: El juego consiste en pintar coches, conforme lleguen (cola de atención), como jugador tienes que atender el primer coche que llega y pintarlo de acuerdo con el color que se solicita en la etiqueta de debajo de cada carro, tienes que seleccionar el color de una paleta de colores para poder pintar dicho carro, el juego se acaba cuando la cola de atención llegue a 5 carros.

A continuación, se representa la explicación:



**NOTA:** Puedes usar las imágenes que consideres adecuado para esta práctica, a continuación, se enumera los puntos a cubrir en este juego:

1. Un carro se enfila cada determinado tiempo (20 segundos), la velocidad de enfilamiento aumenta (conforme el usuario pinta 3 carros).
2. Solo se puede pintar el primer carro respetando la cola.
3. Los colores que se tienen que pintar se generan de forma aleatoria (Etiquetas de color debajo de cada carro).
4. El juego termina cuando hay 5 carros enfilados.
5. Se tiene que mostrar el record obtenido por el jugador (Cantidad de carros pintados y tiempo total).

Código	<b>INSTRUMENTO DE EVALUACIÓN</b>	Revisión:
<b>F-SGC-033</b>		<b>00</b>

## Práctica 12. ESTACIONAMIENTO PARA AUTOS

Usted administra un estacionamiento de autos en forma de “callejón” que tiene una puerta para ingresar los vehículos y otra para sacarlos, dado que es un callejón imaginario no tiene límites en la cantidad de autos que tiene como cupo. Se cobra a \$2.00 pesos el segundo

Se diseña una clase de Cola Circular Doblemente Ligada que contiene una cola de objetos de autos donde cada auto se considera un nodo y se desea controlar los siguientes datos:

- Placas del auto (cadena)
- Propietario (cadena)
- Hora de entrada (DateTime)

Diseñar un programa que contenga las siguientes opciones en el menú:

1. Entrada de Auto
2. Salida de Autos

Consideraciones:

1. **Entrada de Autos:** Cuando se ingrese un auto debe considerarse lo siguiente:
  - Capturar las placas y el nombre del propietario del auto.
  - Registrar la hora de entrada tomándola del sistema mediante DateTime.Now.
  - Insertar en la cola.
  - Confirmar la entrada mediante un mensaje.
2. **Salida de Autos:**
  - Validar que el estacionamiento no esté vacío.
  - Presentar la información del vehículo (placas y nombre propietario), la hora de entrada, hora de salida y el cálculo del costo de estacionamiento.

## CRITERIOS DE EVALUACION

Las 12 prácticas tienen un total de 120 puntos que se reflejará con un valor con escala de 1-10 de calificación del parcial equivalente al **25%** para el **HACER + SER** del alumno.

## VALIDACION DE LA ACADEMIA\*

Nombre de los integrantes de la academia	Firma
<b>ISC. Ruth Betsaida Martínez Domínguez, MGTI.</b>	
<b>ISC. Mirian Magaly Canche Caamal, MGTI.</b>	
ELABORÓ: Academia	

\* Este apartado solo se llenará para la entrega de este instrumento a la División correspondiente.