

Guía para el desarrollo del componente práctico y rubrica de evaluación -Unidad 2 - Fase 3 - Prácticas simuladas.

Anexo 1 Problemas a desarrollar

Unidad 2: Fase 3 Practicas Simuladas

Estimado Estudiantes debe desarrollar los tres problemas planteados de la practica Simulada de forma individual.

Desarrollar una aplicación codificada en el lenguaje de programación propuesto como fundamento a las estructuras de datos lineales representados con clases y objetos.

La actividad consiste en:

Las actividades que va a desarrollar en el componente práctico son las siguientes:

Es una actividad práctica que se desarrolla de forma individual al interior del aula virtual haciendo uso de los escenarios con apoyo tecnológico; para su desarrollo contará con los contenidos temáticos de la unidad 2, videos de apoyo que se compartirán en el foro colaborativo y la grabación de las webconferencia de la unidad 2.

Así mismo se requiere que profundice en la temática de la unidad 2 relacionada con las estructuras de datos lineales (Pilas, Colas y Listas). Implementadas a través de las colecciones genéricas **Stack<T>**, **Queue<T>** y **List<T>**, apropiando conceptos básicos de la programación orientada a objetos, como clases y métodos, aplicando la interfaz gráfica de usuario GUI de Windows Forms, utilizando el lenguaje de programación Visual C#.

Cada estudiante desarrollará una única aplicación que a través de la implementación de clases y formularios maneje los datos de las tres estructuras lineales (pila, cola y lista) de acuerdo con los planteamientos de problemas que se indican a continuación.

Tabla 1. Planteamientos de problemas para la pila, cola y lista

Planteamiento del problema 1, para ser resuelto aplicando el concepto de PILA utilizando la colección genérica Stack <T>

La empresa de telecomunicaciones Tekcom S.A. desea implementar una aplicación que se comporte como una estructura de datos tipo **Pila**, es decir que el ultimo registro ingresado sea el primero en salir.

Que cuente con un menú de opciones para registrar, eliminar datos y el reporte del total de los usuarios. Para lo cual requiere que el usuario del sistema registre a través del formulario de la pila, los datos de cada usuario y se almacenen en una clase llamada **PilaClientes**, la cual tendrá como atributos o propiedades: los datos del clientes, (número de identificación, nombre, dirección, estrato (1, 2, 3, 4, 5, 6) para seleccionar de una lista desplegable, mes de afiliación a la empresa de comunicaciones el cual será mediante el uso del control DateTimePicker, al igual que la categoría (Urbano o Rural) del cliente, es importante que se muestre en otra caja de texto el total a pagar por el servicio de mes, sabiendo que el valor a pagar depende de la siguiente tabla

Según el canal contratado

Tabla 1. Valor Canal Contratado

1. Canal	2. Valor mes	
	Urbano	Rural
5 megas	\$ 20000	\$ 30000
10 megas	\$ 30000	\$ 40000
20 megas	\$ 45000	\$ 55000
50 megas	\$ 60000	\$ 70000

se tienen los siguientes descuentos

Tabla 2. Descuentos por Estrato

1. Estrato	2. Descuento
1	15%
2	15%
3	15%
4	10%
5	10%
6	10%

El valor será, Valor mes teniendo en cuenta el descuento según el estrato socioeconómico, esta caja de texto tendrá la propiedad no editable, para evitar que el usuario ingrese ese valor manualmente. Se requiere que en la medida que se vayan realizando los registros se muestren en el mismo formulario en un control DataGridView. Realizar el cálculo matemático para ir acumulando y al elegir la opción de reportes se muestre en una caja de texto en el formulario la sumatoria del total de los pagos de los clientes registrados.

Planteamiento del problema 2, para ser resuelto aplicando el concepto de COLA a través de la colección genérica Queue <T>

La secretaría de movilidad de Medellín desea implementar una aplicación que se comporte como una estructura de datos tipo **Cola**, es decir, que el primer registro ingresado sea el primero en salir. Que cuente con un menú de opciones para registrar, eliminar datos y reportes de los infractores que se acercan a la secretaria de movilidad a realizar el pago de sus comparendos.

Para lo cual requiere que el usuario del sistema registre a través del formulario de la cola, los datos de cada infractor y se almacenen en una clase llamada **ColaInfraccion**, la cual tendrá como atributos o propiedades: la identificación y el nombre completo del Infractor, dirección de residencia, placa del automotor, tipo de automotor (Automóvil, motocicleta, taxi, camión o bus) en un control ComboBox para seleccionarlos de una lista desplegable, el año de matrícula (desde 2010 hasta el año actual), la fecha del comparendo para lo cual hacer uso del control DateTimePicker, el número de días desde la expedición del comparendo en cuyo caso si los días son menores a 10 días, tendrá un descuento del 50%, si los días son menores a 20 días, tendrá un descuento del 25%, si los días son menores a 30 días, tendría un descuento del 10% y más de 30 días no tendrá ninguno descuento, tomando como valor base del comparando \$450.000 pesos. Se requiere que en la medida que se vayan realizando los registros se muestren en el mismo formulario en un control DataGridView. Al elegir la opción del menú reportes mostrar en una caja de texto en el formulario el total de registros ingresados, para lo cual sugiere utilizar la propiedad Count() de la cola.

Planteamiento del problema 3, para ser resuelto aplicando el concepto de LISTA a través de la colección genérica List<T>

La institución educativa técnico de la alcaldía de Neiva requiere que a través de una aplicación que se comporte como una estructura de datos tipo **Lista**, es decir que permita realizar el registro y eliminar cualquier registro deseado previamente consultado. Que cuente con un menú de opciones para registrar, consultar y eliminar datos de los estudiantes. Se requiere que el usuario del sistema registre a través del formulario de la lista, los datos de cada Estudiante y se almacenen en una clase llamada **ListaEstudiante**, la cual tendrá como atributos o propiedades los datos básicos de los Estudiantes inscritos así:

Tipo de identificación para seleccionar de una lista desplegable (CC, TI, CE, NIP, NIT), identificación del Estudiante, el nombre completo, el estrato socioeconómico (1, 2, 3, 4, 5 y 6) para seleccionar de una lista desplegable, la fecha de nacimiento hacer uso del control DateTimePicker, el grado que hace de 6 a 11 en una lista desplegable, los estudiantes de 6 a 9 tienen derecho a desayuno, los

estudiantes de 10 y 11 tienen derecho a almuerzo, en radios la opción si utilizo o no el servicio del PAE. Se requiere que en la medida que se vayan realizando los registros se muestren en el mismo formulario en un control DataGridView. Se debe tener en cuenta que antes de eliminar un registro se debe consultar a través de la identificación que el estudiante esté registrado, si es así se procede a eliminar el registro de la lista, una vez confirme el usuario que realmente desea eliminarlo, por ende, el botón Eliminar debe de deshabilitarse después del eliminar el Registro para que se obligue a realizar nuevamente la consulta para borrar cada registro.

Una vez que haya leído y comprendido los tres planteamientos, procede a realizar el análisis y la abstracción de los datos suministrados en cada planteamiento, con el fin de identificar los requerimientos funcionales de la aplicación.

Con el propósito de estandarizar y facilitar este proceso de abstracción se pone a disposición una plantilla que se indica a continuación en la **Tabla 3**, la cual se diligencia y se comparte en el foro para recibir comentarios.

Tabla 3. Plantilla para la abstracción

Nombre del estudiante	(Ingrese el nombre del estudiante)		
Planteamiento del Problema:	(síntesis resumida de los tres planteamientos)		
Nombre de la Clase y su Ámbito de visibilidad ya sea (Public o Private)	Nombre de las propiedades y/o atributos de la clase con sus tipos de datos.	Tipo de estructura (pila, cola o lista)	Opciones del menú que tendrá la aplicación (Métodos)
<u>Ejemplo de cómo diligenciar</u> Nombre de la clase: Persona Ámbito de visibilidad: Publica Public Class Pila	<u>Ejemplo de atributos con sus tipos de datos</u> String Identificación String Nombre Int Edad	Pila	Registrar Eliminar Salir
Ámbito y nombre de la clase		Cola	
Ámbito y nombre de la clase		Listas	

Autoría Propia

Diligenciada la tabla de la abstracción después de eliminar los datos del ejemplo. Proceda a crear el nuevo proyecto en el IDE Visual Studio, utilizando el lenguaje C#, debe darle como nombre **"Fase3NombreEstudiante"**, por ejemplo: **Fase3HernandoRobles**. Se requiere hacer uso de las buenas prácticas de programación

especialmente para el nombre de los atributos y propiedades de la clase y los métodos para que sean nemotécnicos, es decir que el nombre que le asigne indique la información que contendrá.

Requerimientos funcionales de la aplicación

La aplicación tendrá una interfaz inicial que al ejecutarla muestre su nombre completo como autor de la aplicación, el nombre de la aplicación, y el nivel de seguridad básico con la contraseña de acceso genérica que será: **123** que estará enmascarada, **No incluir Nombre de usuario**, solo la contraseña, de tal manera que si digita la contraseña correcta lo lleve a una nueva pantalla o interfaz en la que se muestra el formulario con el menú de opciones que hará el llamado al formulario para la captura de los datos de la Clase Pila, de la Clase Cola y La Clase Lista.

El siguiente video 1 denominado Navegando entre formularios se explica cómo realizar el menú de opciones de la en la arquitectura de navegación de la aplicación a realizar. Ver video. https://youtu.be/GL_tEbmHNFo

De acuerdo con lo anterior, lo primero que se debe hacer es la estructura de navegación tal como se explica en el video anterior y luego para cada estructura representada en los tres planteamientos (pila, cola, lista) se requiere crear una clase distinta a la clase formulario que contenga los atributos y/o las propiedades auto implementadas con los tipos de datos apropiados según el planteamiento indicado, con su respectivo formulario personalizado para el manejo de los datos. Utilice color de fondo diferente para cada formulario para que diferencie por el color si se trata de la pila, la cola y la lista.

Aplicar los filtros o validaciones de datos para controlar la entrada de datos en el formulario, es decir que todos los campos son obligatorios y si es un dato numérico que no permita ingresar letras.

Evite utilizar imágenes para el fondo del formulario, sea creativo en el diseño.

Crear los supuestos que considere necesarios para mejorar la aplicación.

Tenga en cuenta que en ningún caso se hará uso de Bases de datos, o manejos de archivos, los datos se guardan en memoria principal a través de las clases tal como lo identificó en la tabla de abstracción.

Entrega de la actividad:

Un archivo comprimido, que contiene el código completo de la aplicación desarrollada de forma individual incluyendo el documento de la abstracción como resultado de la práctica.

Esta carpeta se entrega en el entorno de evaluación Unidad 2 - Fase 3 - Práctica Estructuras lineales - Rúbrica de evaluación y entrega de la actividad.

Nota: Si el archivo comprimido es muy pesado se puede cargar a un Google Drive y el enlace generado será publicado dentro del archivo a entregar.