



UNIVERSIDAD ESTATAL A DISTANCIA  
VICERRECTORÍA ACADÉMICA  
ESCUELA DE CIENCIAS EXACTAS Y NATURALES  
Cátedra Tecnología de Sistemas



Programación Avanzada con C#

Código: 00830

**Proyecto 1. Valor 1%**

## **Temas de Estudio**

1. Tema 1 Introducción al lenguaje de programación
2. Tema 2 Particularidades del lenguaje C#
3. Tema 3 Manejo de excepciones en C#
4. Tema 4 Conceptos de Interfaz Gráfica

## **Objetivo**

Poner en práctica los conceptos de interfaz gráfica, así como el manejo de excepciones.

## **Software de Desarrollo**

Visual Studio Community 2022 (C# Net 8.0)

## **Desarrollo**

Una empresa de entregas de pedido ENTREGAS S.A. le ha contratado para que usted desarrolle una aplicación de escritorio, con el fin de poder administrar los pedidos y las entregas de los artículos registrados en el sistema. Esta aplicación debe ser creada en C# y se requiere que contenga las siguientes funcionalidades:

- Registrar y consultar Tipos de Artículos.
- Registrar y consultar Artículos.
- Registrar y consultar Clientes.
- Registrar y consultar Repartidores.
- Registrar y consultar Pedidos.

### **Registrar Tipos de Artículos**

Esta opción le permitirá a la persona usuaria registrar o agregar los datos de los tipos a los que puede pertenecer un artículo, por ejemplo: Abarrotes, Oficina, Comida, etc. Para ello se debe crear la clase correspondiente.

Se debe poder registrar la siguiente información:

- Id (int), identificador del tipo de artículo. Este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos.
- Nombre (string), nombre del tipo de artículo.
- Descripción (string), breve descripción del tipo de artículo.

Todos los datos son requeridos. Una vez capturada la información se debe agregar a un arreglo de objetos de tipo de la clase creada. El arreglo debe de ser de 10 posiciones.

### **Registrar Artículos**

Esta opción le permitirá a la persona usuaria registrar o agregar los artículos que pueden ser solicitados en el pedido. Para ello se debe crear la clase correspondiente.

Se debe poder registrar la siguiente información:

- Id (int), identificador del artículo. Este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos.
- Nombre (string), nombre del artículo.
- TipoArticulo (Objeto de Tipo de Articulo), debe realizarse por medio de un Combobox, el cual desplegará los tipos de artículos registrados previamente.
- Valor (double), monto del artículo. Se debe validar que solo acepte valores numéricos.
- Inventario(int), cantidad de artículos que se encuentra en el inventario
- Activo (bool), estado del artículo, si se encuentra activo o no. Se debe utilizar estrictamente un Combobox con los textos "Sí" y "No", el cual se debe utilizar el "Sí" como true y el "No" como false.

Todos los datos son requeridos. Una vez capturada la información se debe agregar a un arreglo de objetos de tipo de la clase creada. El arreglo debe de ser de 20 posiciones.

### **Registrar Clientes**

Esta opción le permitirá a la persona usuaria registrar o agregar los datos de una persona cliente. Para ello se debe crear la clase correspondiente.

Se debe poder registrar la siguiente información:

- Identificación (int), identificador del cliente. Este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos.
- Nombre (string), nombre del cliente.
- PrimerApellido (string), primer apellido del cliente.
- SegundoApellido (string), segundo apellido del cliente.
- FechaNacimiento (datetime), fecha de nacimiento del cliente. Se debe utilizar el componente de DateTimePicker. Debe mostrar solo la fecha y no el tiempo.
- Activo (bool), estado del artículo, si se encuentra activo o no. Se debe utilizar estrictamente un Combobox con los textos "Sí" y "No", el cual se debe utilizar el "Sí" como true y el "No" como false.

Todos los datos son requeridos. Una vez capturada la información se debe agregar a un arreglo de objetos de tipo de la clase creada. El arreglo debe de ser de 20 posiciones.

### **Registrar Repartidores**

Esta opción le permitirá a la persona usuaria registrar o agregar los datos de los repartidores de los pedidos. Para ello se debe crear la clase correspondiente.

Se debe poder registrar la siguiente información:

- Identificación (int), identificador del repartidor. Este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos.
- Nombre (string), nombre del repartidor.
- PrimerApellido (string), primer apellido del repartidor.
- SegundoApellido (string), segundo apellido del repartidor.
- FechaNacimiento (datetime), fecha de nacimiento del administrador. Se debe utilizar el componente de DateTimePicker y asegurarse que el repartidor es mayor de edad. Debe mostrar solo la fecha, sin el tiempo.
- FechaContratación (datetime), fecha de la contratación del repartidor. Se debe utilizar el componente de DateTimePicker y no puede ser mayor al día en que se está registrando el repartidor en la aplicación (o sea, tiene que ser <= hoy). Debe mostrar solo la fecha, sin el tiempo.

- Activo (bool), estado del artículo, si se encuentra activo o no. Se debe utilizar estrictamente un Combobox con los textos “Sí” y “No”, el cual se debe utilizar el “Sí” como true y el “No” como false.

Todos los datos son requeridos. Una vez capturada la información se debe agregar a un arreglo de objetos de tipo de la clase creada. El arreglo debe de ser de 20 posiciones.

### **Registrar Pedidos**

Funcionalidad que debe permitir al usuario registrar los pedidos de los artículos solicitados por un cliente, así como, asignarse el repartidor y la dirección de la ubicación donde se entregará el pedido. Esta pantalla de contar con la funcionalidad de registra el encabezado del pedido, para esta opción se debe poder registrar la siguiente información:

- Número de pedido (int), identificador del pedido. Este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos.
- FechaPedido (datetime), se registra la fecha del pedido. Se debe utilizar el componente de DateTimePicker, se debe validar que la fecha registrada por usuario **no** sea inferior al día que se realiza el registro (fecha del sistema). Debe mostrar solo la fecha, sin el tiempo.
- Cliente (Objeto Cliente), debe permitir al usuario seleccionar un cliente ya registrado en el arreglo de clientes.
- Repartidor (Objeto Repartidor), debe permitir al usuario seleccionar un repartidor ya registrado en el arreglo de repartidores.
- Dirección (string), dirección de la entrega del pedido.

Todos los datos son requeridos. La información del encabezado del pedido se debe almacenar en un arreglo de objetos de tipo Pedidos de 40 posiciones.

Así mismo, la pantalla tendrá la funcionalidad de registrar el detalle de los artículos asignados en el pedido, el usuario tendrá una opción para seleccionar el artículo que desea registrar y la cantidad, importante mencionar que se registrar más de un artículo en el pedido, para esta opción se debe poder registrar la siguiente información:

- Número de Pedido(int), corresponde al número de pedido del encabezado.
- Artículo (Objeto Artículo), debe permitir al usuario seleccionar un artículo ya registrado en el arreglo de clientes.
- Cantidad (int), corresponde a la cantidad de artículo del pedido, no puede ser superior a la cantidad registrada en el inventario del artículo.

- Monto, se debe almacenar el valor del artículo, más el 12% correspondiente del envío del artículo.

Todos los datos son requeridos. La información del detalle del pedido de cada artículo se debe almacenar en un arreglo de objetos de tipo Detalle de Pedidos de 500 posiciones.

- Mostrar en un control “DataGridView” la información de los artículos que se van registrando en detalle.
  - Se debe mostrar en el control el número de pedido, la cantidad y el monto.
  - Para cada artículo de la lista se debe mostrar la siguiente información del objeto: id, nombre, tipo de articulo
- No se puede registrar el mismo artículo en el pedido.
- Una vez realizado el pedido por cada artículo registrado en el arreglo correspondiente se debe actualizar el valor de inventario, restando la cantidad con el valor del inventario.
  - Se debe validar que la cantidad ingresada sea mayor a cero.
  - En caso de que el nuevo valor del artículo en el atributo inventario es cero se debe actualizar el estado del articulo a inactivo (false).

### **Consultar Tipos de Artículos**

Esta opción le permitirá a la persona usuaria visualizar en pantalla todos los tipos de artículos registrados, para tal fin debe hacer uso del control “DataGridView”, se debe mostrar todas las propiedades de la clase. (3 columnas)

### **Consultar Artículos**

Esta opción le permitirá a la persona usuaria visualizar en pantalla todos los artículos registrados, para tal fin debe hacer uso del control “DataGridView”, se debe mostrar todas las propiedades de la clase. Para la propiedad Tipo de artículo se debe mostrar nombre del tipo de artículo al que pertenece. Para la columna Activo debe mostrar “Sí” cuando es true y “No” cuando es false (6 columnas)

### **Consultar Cliente**

Esta opción le permitirá a la persona usuaria mostrar en pantalla todos los clientes registrados, para tal fin debe hacer uso del control “DataGridView”, se debe mostrar todas las propiedades de la clase. La fecha de nacimiento no debe mostrar el tiempo, solo la fecha (“dd/MM/yyyy”). Para la columna Activo debe mostrar “Sí” cuando es true y “No” cuando es false. (6 columnas)

**Consultar Repartidores**

Esta opción le permitirá a la persona usuaria visualizar en pantalla a todos los repartidos registrados, para tal fin debe hacer uso del control “DataGridView”, se debe mostrar todas las propiedades de la clase. La fecha de nacimiento y la fecha de contratación no deben mostrar el tiempo, solo la fecha (“dd/MM/yyyy”). (6 columnas)

**Consultar Pedidos**

Esta opción le permitirá a la persona usuaria visualizar en pantalla todos los pedidos registrados, para el cual se debe visualizar los grids, en el primero se deben mostrar todos los Encabezados de los pedidos con la siguiente información (11 columnas):

- Número de Pedido
- Fecha de Pedido
- Del Cliente:
  - Id
  - Nombre
  - Primer Apellido
  - Segundo Apellido
- Del Repartidor:
  - Id
  - Nombre
  - Primer Apellido
  - Segundo Apellido
- Dirección:

Cuando se seleccione una fila del encabezado del pedido, en el segundo control “DataGridView”, se debe visualizar los detalles asociados a ese pedido, el cual mostrará la siguiente información (5 columnas):

- Del Artículo:
  - Id
  - Nombre
  - Tipo de artículo, se debe mostrar nombre del tipo de artículo al que pertenece
- Cantidad
- Monto

## Consideraciones técnicas

### Uso de comentarios

- Se deberá implementar el uso de comentarios cuando se realiza la codificación del programa, en la parte superior indicar la universidad, el cuatrimestre, el nombre y la descripción del proyecto, el nombre del estudiante y la fecha en que está realizando la programación. Lo anterior debe estar en cada archivo cs (clase, program, formulario, etc) que se realiza el código de programación; como se puede observar en el siguiente ejemplo:

```
1  /**
2   * UNED X Cuatrimestre
3   * Proyecto X: Descripción del proyecto
4   * Estudiante:
5   * Fecha:
6   */
7  namespace CapaEntidades
8  {
```

Debe utilizar POO (Programación orientada a objetos) para resolver el problema

- Las clases de objetos que usted considere como entidades correspondientes a Tipo de Artículo, Artículo, Cliente, Repartidor, Pedidos, Detalle Pedido no deben contener lógica para solicitar información a la persona usuaria, solo debe tener la estructura de la clase y sus propiedades. El nombre del objeto y las propiedades no deben de llevar tilde ni ñ.
- Implementar en la solución proyectos independientes con el fin de tener las distintas capas recomendadas: Capa de Entidades, Capa de Acceso a Datos, Capa de Lógica Negocio, Capa de Presentación (Interfaz del Usuario), etc.
- Cada clase debe ser creada en archivo por separado.
- Las clases no deben tener métodos vacíos y recuerde utilizar nombres significativos a sus variables.
- Las entidades de Cliente y Repartidor tienen varias propiedades iguales, por lo cual se recomienda aplicar herencia (no obligatorio).

### Propiedades Id e Identificación

- No se debe crear una variable que se incremente con cada inserción. Debe permitir al usuario digitar estos valores.

### Uso de combobox

- Los combobox no se deben poder editar o escribir en él, solo debe permitir seleccionar una opción de la lista.

#### Uso de arreglos

- No deben ser arreglos de tipo "int" ni "string", deben ser arreglos de objetos de las clases entidades definidas.
- Cada vez que se agrega un objeto al arreglo se deben limpiar los campos de la pantalla y mostrar el siguiente mensaje: "El registrado se ha ingresado correctamente". Esto con el fin de que el usuario pueda ingresar otro registro. En caso de que el arreglo llegue a su límite y no se puedan ingresar nuevos registros, se debe visualizar un mensaje "No se pueden ingresar más registros".
- **No debe utilizar colecciones, solo se permite el uso de arreglos para esta tarea.**
- Los arreglos deben estar en la capa de acceso a datos.

#### Debe implementar el manejo de excepciones

- Si ocurre una excepción, el sistema no debe cerrarse, se debe mostrar un mensaje al usuario y manejar la excepción de forma apropiada.

#### Interfaz de usuario

- Debe usar interfaz de usuario con formularios GUI (Proyecto Windows Form).
- Se debe cuidar con el orden de los formularios.
- Se debe cuidar con la ortografía.
- Se debe crear un menú para acceder a las opciones solicitadas.

#### Uso de DataGridView

- Los títulos de las columnas deben de estar previamente establecidos para que se muestren de forma correcta, y no se debe dejar que tomen el nombre de la propiedad de la clase.
- Los DataGridView debe de ser de solo lectura, no se pueden modificar los datos que se muestren allí. Tampoco debe permitir agregar o borrar nuevos registros.
- Cuando se seleccione un dato del DataGridView se debe seleccionar toda la fila.



## Honestidad Académica



<https://audiovisuales.uned.ac.cr/play/player/23048>

### Nota Importante

Cada estudiante es responsable del contenido que entrega, si no es el archivo correcto, no podrá entregarlo posterior a la fecha establecida.

Si el contenido del archivo coincide con algún otro estudiante, o se comprueba que no es de su autoría, se aplicaría lo indicado en la plataforma en el documento [Lineamientos ante casos de plagio](#)

## Indicaciones Importantes

- Es obligatorio que incluya todo el directorio donde se encuentra el proyecto 1.
- El **proyecto 1** debe estar desarrollado en **Visual Studio Community 2022** que es la herramienta oficial del curso.
- Los trabajos deben realizarse en forma individual. Dentro del código del programa debe de indicar la documentación que explique cómo fue realizado el programa.
- Si utiliza código de algún ejemplo del libro, del algún profesor o de otra fuente que no sea de su autoría, debe de indicarlo.
- Comprima todos los archivos en un solo archivo .zip o .rar.
- **Nombre del archivo que envía:** debe ser nombre y primer apellido del estudiante, y nombre de la tarea. **Ejemplo: JuanRojas-Proyecto1.**
- La entrega del proyecto debe ser en las fechas establecidas en la plataforma de aprendizaje en línea Moodle, en el apartado que se indique.
- Si no concluyó a tiempo el proyecto, debe entregar lo que pudo hacer e incluir una carta explicando las razones por las cuales no finalizó.

**Rúbrica de Evaluación**

<b>Criterio</b>	<b>Cumple a satisfacción lo indicado en la evaluación</b>	<b>Cumple medianamente en lo indicado en la evaluación</b>	<b>Cumple en contenido y formato, pero los aportes no son significantes</b>	<b>No cumple o no presenta lo solicitado</b>
Presentación del código: Correcto uso de variables, nombres de métodos y variables significativos, no presenta métodos vacíos.	5	3	2	0
Entrega el archivo del programa con todo el directorio del programa desarrollado. Indica dentro del programa en los comentarios, si se utiliza código de algún ejemplo del libro o algún profesor o de otra fuente que no sea de autoría del estudiante. El nombre del archivo es el solicitado. Agrega los comentarios de información del proyecto en las clases.	5	3	2	0
POO y creación de las clases (Tipo de Artículo, Artículo, Cliente, Repartidor, Pedidos, Detalle Pedido) Utiliza propiedades y no atributos. No agrega constructores ni métodos adicionales. El proyecto está dividido en capas.	5	3	2	0
Registra Tipos de Artículo.	5	3	2	0
Registra Artículos.	5	3	2	0
Registra Clientes	5	3	2	0
Registra Repartidores	5	3	2	0

Registra Pedidos	15	10	5	0
Consultar Tipos de Artículos.	5	3	2	0
Consultar Artículos.	5	3	2	0
Consultar Cliente.	5	3	2	0
Consultar Repartidores	5	3	2	0
Consultar Pedidos	15	10	5	0
Correcto uso del manejo de excepciones con try catch.	5	3	2	0
Interfaz de usuario es fácil de usarse y presenta buen diseño. Presenta correctamente los DataGridView, los Combobox y los DateTimePicker. Muestra mensajes de error y de satisfacción de forma amigable. Presenta una correcta forma de navegación entre los formularios.	10	7	3	0
<b>TOTAL</b>	<b>100</b>			