



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



Programación Avanzada

Código: 00830

## Tarea 2. Valor 2%

### Temas de Estudio

1. Tema 3 Manejo de excepciones en C#
2. 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 Framework 4.8 o Net 6.0)

### Desarrollo

Debido al éxito que se ha obtenido con la primera versión del sistema de información entregada para la empresa **REST-UNED**, se le ha solicitado el realizar algunas mejoras a la primera versión entregada con el fin de incorporar nuevas funcionalidades, así como, modificar la interfaz de la aplicación, porque se solicita que desarrolle el sistema de información para que utilice formularios de Windows (Interfaz Gráfica). El sistema debe disponer de las siguientes funcionalidades:

- **Registrar Restaurante**
- **Registrar Categoría Plato**
- **Registrar Plato**
- **Registrar Clientes**
- **Registrar Platos por Restaurante**
- **Registrar Extras**
- **Consultar Restaurante**
- **Consultar Categoría Plato**
- **Consultar Plato**
- **Consultar Clientes**
- **Consultar Platos por Restaurante**

- **Consultar Extras**

### **Registrar Restaurante**

Esta opción del menú le permite al usuario registrar o agregar los datos relacionados con los restaurantes con que cuenta la empresa. Se debe poder registrar la siguiente información:

- Id de restaurante (int), este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos
- Nombre de restaurante (string)
- Dirección (string)
- Estado (bool), activo o inactivo, (mostrar en Combobox)
- Teléfono (string)

Una vez capturada la información se debe agregar a un arreglo de objetos de tipo restaurante. El arreglo debe ser de 20 posiciones

### **Registrar Categoría de Plato**

Debe permitir al usuario registrar las categorías de platos con que cuenta la empresa, pudiendo registrar la siguiente información:

- Id categoría (int), este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos
- Descripción (string)
- Estado (bool), activo o inactivo, (mostrar en Combobox)

Una vez capturada la información se debe agregar a un arreglo de objetos de categoría. El arreglo debe ser de 20 posiciones.

### **Registrar Platos**

Debe permitir al usuario registrar los platos con que cuenta la empresa, pudiendo registrar la siguiente información:

- Id Plato (int), este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos
- Nombre (string)
- Precio (int)

- Id Categoría (Objeto Categoría) debe almacenar un objeto de tipo Categoría, por lo tanto, debe validar que exista en el arreglo de Categorías.

Una vez capturada la información se debe agregar a un arreglo de objetos de tipo Plato. El arreglo debe ser de 20 posiciones

### **Registrar Clientes**

Debe permitir al usuario registrar los clientes del restaurante, pudiendo registrar la siguiente información:

- Identificación (string) se debe validar que no existan identificaciones repetidas
- Nombre (string)
- Primer apellido (string)
- Segundo apellido (string)
- Fecha de nacimiento (DateTime)
- Género (char)

Una vez capturada la información se debe agregar a un arreglo de objetos de tipo Cliente. El arreglo debe ser de 20 posiciones

### **Registrar Platos por Restaurante**

Funcionalidad que debe permitir al usuario registrar (asociar) los platos que están disponibles en cada restaurante. Para ello se debe crear una clase "PlatoRestaurante", la cual debe contener además de los atributos de id asignación, fecha de afiliación, restaurante, debe contener un arreglo de los platos asignados a un restaurante. Para esto se debe realizar lo siguiente:

- Mostrar en pantalla en un ComboBox los restaurantes disponibles.
  - Se debe validar que al menos exista un restaurante registrado para realizarle asignación
  - Para cada restaurante registrado debe mostrar los siguientes datos: Id restaurante, Nombre, Dirección
  - Solo debe mostrar los restaurantes activos
- Permitir al usuario seleccionar desde ComboBox el restaurante
- Mostrar en un control DataGridView los platos disponibles para asignar al restaurante
  - Se debe validar que al menos exista un plato registrado para ser asociado
  - Para cada plato de la lista se debe mostrar la siguiente información: id plato, nombre, precio

- El usuario puede seleccionar uno o más platos del control DataGridView para asociar al restaurante.
- Se debe preguntar al usuario si desea agregar más platos al restaurante. Si la respuesta es afirmativa, entonces se vuelve a mostrar la lista de platos para que el usuario nuevamente seleccione e indique un id.
- La clase PlatoRestaurante debe tener un arreglo de 10 posiciones donde se agreguen los platos
- En la parte inferior de esta pantalla, se debe mostrar en un control DataGridView los platos (id plato, nombre, precio) que han sido asociados al restaurante seleccionado inicialmente, es decir, cada vez que se asocia un plato al restaurante, estos deben ser mostrada en la parte inferior.

Para esta opción se debe poder registrar la siguiente información para PlatoRestaurante:

- Id asignación (int)
- Fecha (DateTime)
- Restaurante (Objeto Sucursal)
- Platos (Arreglo de objeto Platos con 10 posiciones)

La asociación de platos a restaurante se debe almacenar en un arreglo de objetivos tipo PlatoRestaurante de 20 posiciones.

### **Registrar Extras**

Funcionalidad que permite registrar las extras disponibles por categoría de plato, pudiendo registrar la siguiente información:

- Id extra (int), se debe validar que no existan id repetidos
- Descripción (string)
- Id categoría plato (Objeto Categoría)
- Estado (bool), activo o inactivo (mostrar en Combobox)
- Precio (int)

### **Consultar Restaurantes**

En esta opción debe mostrar en pantalla en un control DataGridView todos los restaurantes registrados, para cada restaurante se debe mostrar todos los atributos

**Consultar Clientes**

En esta opción debe mostrar en pantalla en un control DataGridView los clientes registrados, para cada cliente se debe mostrar todos los atributos

**Consultar Categoría de Plato**

Esta opción permite al usuario mostrar en pantalla en un control DataGridView las categorías de platos registrados, se debe mostrar todos los atributos

**Consultar Platos**

Esta opción permite al usuario mostrar en pantalla en un control DataGridView todos los platos registrados, para cada plato se debe mostrar todos los atributos

**Consultar Platos por Restaurante**

Esta opción permite consultar los platos que tiene asociados un restaurante, es decir, consulta los platos que están asignados a un restaurante. Debe imprimir en pantalla la información de cada asociación, es decir, id asignación, fecha afiliación, datos de restaurante (id restaurante, nombre, dirección), la información de los platos (id plato, nombre, precio) que están asignados al restaurante. Puede usar un o más DataGridView o el control de interfaz gráfica que permita mostrar la información de la asociación de restaurante y el listado de platos asociados.

**Consultar Extras**

Esta opción permite al usuario mostrar en pantalla en un control DataGridView las extras disponibles por categoría de plato.

**Consideraciones técnicas**

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

- Las clases de objetos (Restaurante, Cliente, Categoría Plato, Plato, Plato Restaurante) no deben contener lógica para solicitar información al usuario, solo debe tener la estructura de la clase y sus propiedades.
- Las clases no deben tener métodos vacíos y recuerde utilizar nombres significativos a sus variables.

### Uso de arreglos

- No deben ser arreglos de tipo “string” ni “int”, deben ser arreglos de objetos de las clases definidas.
- Cada vez que se agrega un objeto al arreglo se debe preguntar al usuario si desea agregar otro y no obligar al usuario a ingresar los 20 registros de una vez.
- No debe utilizar colecciones, solo se permite el uso de arreglos para esta tarea.

### 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.

# 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 expone a las sanciones indicadas en la plataforma en el documento [Lineamientos ante casos de plagio](#)

## Indicaciones Importantes

- Es obligatorio que incluya todo el directorio donde se encuentra < tarea2>.
- La **Tarea 2** debe estar desarrollado en **Visual Studio Community 2019 o superior** que es la herramienta oficial del curso.
- El programa debe ser modular, utilizando de la mejor manera funciones definidas por usted.
- 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-tarea2.**
- La entrega de la <**Tarea2**> en las fechas establecidas en la plataforma de aprendizaje en línea Moodle en el apartado que se indique.
- Si no concluyó a tiempo la tarea, 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 significativos, no presenta métodos vacíos.	5	3	1	0.5
POO y creación de las clases Restaurante, Cliente, Categoría Plato, Plato, Plato Restaurante	10	5	2	0.5
Registrar Restaurante	5	3	1	0.5
Registrar Categoría Plato	5	3	1	0.5
Registrar Plato	5	3	1	0.5
Registrar Clientes	5	3	1	0.5
Registrar Plato por Restaurante	10	5	2	0.5
Registrar Extras	5	3	1	0.5
Consultar Restaurante	5	3	1	0.5
Consultar Clientes	5	3	1	0.5
Consultar Categoría Plato	5	3	1	0.5
Consultar Plato	5	3	1	0.5
Consultar Platos por Restaurante	10	5	2	0.5
Consultar Extras	5	3	1	0.5
Correcto uso del manejo de excepciones	5	3	1	0.5
Interfaz de usuario es fácil de usarse y presenta buen diseño.	10	5	2	0.5
<b>TOTAL</b>	<b>100</b>			



