



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

La más grande y famosa cadena de tiendas de videojuegos **45GAMES4U** le ha contratado para que usted desarrolle una aplicación de escritorio, con el fin de poder administrar el inventario de los videojuegos de cada una de las tiendas de la cadena. Esta aplicación debe ser creada en C# y se requiere que contenga las siguientes funcionalidades:

- Registrar y consultar Tipos de Videojuegos
- Registrar y consultar Videojuegos
- Registrar y consultar Administradores de las Tiendas
- Registrar y consultar Tiendas
- Registrar y consultar Inventario (Existencias de los Videojuegos por Tiendas)
- Registrar y consultar Clientes

Registrar Tipos de Videojuegos

Esta opción le permitirá a la persona usuaria registrar o agregar los datos de los tipos a los que puede pertenecer un videojuego, por ejemplo: Acción, Aventura, Deportivo, Estrategia, etc. Para ello se debe crear una clase llamada "TipoVideojuegoEntidad".

Se debe poder registrar la siguiente información:

- Id (int), identificador del tipo de videojuego. 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 videojuego. Por ejemplo: Acción, Aventura, Deportivo, Estrategia, etc.
- Descripción (string), breve descripción del tipo de videojuego.

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

Registrar Videojuegos

Esta opción le permitirá a la persona usuaria registrar o agregar los datos de los videojuegos. Para ello se debe crear una clase llamada "VideojuegoEntidad".

Se debe poder registrar la siguiente información:

- Id (int), identificador del videojuego. Este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos.
- Nombre (string), nombre del videojuego.
- TipoVideojuego (Objeto TipoVideojuegoEntidad), debe realizarse por medio de un Combobox, el cual desplegará los tipos de videojuegos registrados previamente.
- Desarrollador (string), nombre de la empresa desarrolladora del videojuego.
- Lanzamiento (int), año del lanzamiento del videojuego.
- Físico (bool), indica si el videojuego es físico (true) o virtual (false). Se debe utilizar estrictamente un combobox con los textos Físico y Virtual, el cual se debe utilizar el Físico como true y el Virtual como false.

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

Registrar Administradores de las Tiendas

Esta opción le permitirá a la persona usuaria registrar o agregar los datos de los administradores de las tiendas. Para ello se debe crear una clase llamada "AdministradorEntidad".

Se debe poder registrar la siguiente información:

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

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

Registrar Tiendas

Esta opción le permitirá a la persona usuaria registrar o agregar los datos de las tiendas de la cadena. Para ello se debe crear una clase llamada "TiendaEntidad".

Se debe poder registrar la siguiente información:

- Id (int), identificador de la tienda. Este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos.
- Nombre (string), nombre de la tienda.
- Administrador (Objeto AdministradorEntidad), debe realizarse por medio de un Combobox, el cual desplegará a todos los administradores registrados previamente. El combobox debe mostrar el nombre y los dos apellidos del administrador.
- Dirección (string), dirección de la tienda.
- Teléfono (string), teléfono de la tienda.

- Activa (bool), estado de la tienda, si se encuentra activa o no. Se debe utilizar estrictamente un combobox con los textos Si y No, el cual se debe utilizar el Si 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 "TiendaEntidad". El arreglo debe de ser de 5 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 una clase llamada "ClienteEntidad".

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.
- JugadorEnLinea (bool), almacena si el jugador juega en línea o solo local. Se debe utilizar estrictamente un combobox con los textos Si y No, el cual se debe utilizar el Si 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 "ClienteEntidad". El arreglo debe de ser de 20 posiciones.

Registrar Inventario

Funcionalidad que debe permitir al usuario registrar (asociar) los videojuegos que estarán disponibles en cada una de las Tiendas. Para ello se debe crear una clase llamada "VideojuegosXTiendaEntidad".

Para esta opción se debe poder registrar la siguiente información:

- Tienda (Objeto TiendaEntidad), este valor, en combinación con Videojuego, debe de ser único por lo que se debe validar en el sistema que no existan valores repetidos.
- Videojuego (Objeto VideojuegoEntidad), este valor, en combinación con Tienda, debe de ser único por lo que se debe validar en el sistema que no existan valores repetidos.
- Existencias (int), cantidad disponible del videojuego ubicado en la tienda.

Todos los datos son requeridos. La asociación de videojuegos a una tienda se debe almacenar en un arreglo de objetos de tipo VideojuegosXTiendaEntidad de 100 posiciones.

Para esto se debe realizar lo siguiente:

- Mostrar en pantalla en un “ComboBox” las tiendas activas.
 - Se debe validar que al menos exista una tienda registrada activa para realizar asociación.
 - Solo debe mostrar las tiendas que se encuentran activas.
 - Se debe visualizar el nombre de la tienda en el ComboBox.
- Mostrar en un control “DataGridView” los videojuegos disponibles para asignar a la tienda.
 - Se debe validar que al menos exista un videojuego registrado para ser asociado.
 - Para cada videojuego de la lista se debe mostrar la siguiente información del objeto: id, nombre, nombre del tipo de videojuego, nombre del desarrollador y año de lanzamiento.
 - Solamente se puede desplegar los videojuegos que se encuentren en su versión física.
- La persona usuaria puede seleccionar uno o más videojuegos del control “DataGridView” para asociar a la tienda.
- La persona usuaria debe contar con un control de entrada de datos “TextBox”, donde podrá definir la cantidad de existencias que contará cada videojuego.
 - Se debe validar que la cantidad ingresada sea mayor a cero.
 - Si selecciona varios videojuegos para asociar, se les asignará la misma cantidad de existencias a todos.

Consultar Tipos de Videojuegos

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

Consultar Videojuegos

Esta opción le permitirá a la persona usuaria visualizar en pantalla todos los videojuegos registrados, para tal fin debe hacer uso del control “DataGridView”, se debe mostrar todas las propiedades de la clase. Para la propiedad Tipo de videojuego se debe mostrar nombre del tipo de videojuego al que pertenece. (6 columnas)

Consultar Administradores de las Tiendas

Esta opción le permitirá a la persona usuaria visualizar en pantalla a todos los administradores 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. (6 columnas)

Consultar Tienda

Esta opción le permitirá a la persona usuaria visualizar en pantalla todas las tiendas registradas, para tal fin debe hacer uso del control “DataGridView”, se debe mostrar todas las propiedades de la clase. Para la propiedad administrador se debe mostrar el nombre completo del administrador y la fecha de contratación. La fecha de contratación no debe mostrar el tiempo, solo la fecha. Para la columna Activa debe mostrar Si cuando es true y No cuando es false. (7 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. Para la columna Jugador en línea debe mostrar Si cuando es true y No cuando es false. (6 columnas)

Consultar Inventario

Esta opción le permitirá a la persona usuaria visualizar en pantalla todas las asociaciones registradas, para tal fin debe hacer uso del control “DataGridView”, se debe mostrar las siguientes propiedades (7 columnas):

- De la Tienda:
 - Id
 - Nombre
 - Dirección
- Del Videojuego:
 - Id
 - Nombre
 - Nombre del tipo de videojuego
- Existencias

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 (TipoVideojuegoEntidad, VideojuegoEntidad, AdministradorEntidad, TiendaEntidad, ClienteEntidad, VideojuegosXTiendaEntidad) 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 AdministradorEntidad y ClienteEntidad tienen varias propiedades iguales, por lo cual se puede crear una entidad nueva llamada PersonaEntidad, se debe colocar las propiedades en común en esta nueva entidad y aplicar herencia.

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

Criterio	Cumple a satisfacción lo indicado en la evaluación	Cumple medianamente en lo indicado en la evaluación	Cumple en contenido y formato, pero los aportes no son significantes	No cumple o no presenta lo solicitado
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 (TipoVideojuegoEntidad, VideojuegoEntidad, AdministradorEntidad, TiendaEntidad, ClienteEntidad, VideojuegosXTiendaEntidad). Utiliza propiedades y no atributos. No agrega constructores ni métodos adicionales. El proyecto está dividido en capas.	5	3	2	0
Registra Tipos de Videojuegos.	5	3	2	0
Registra Videojuegos.	5	3	2	0

Registra Administradores de las Tiendas.	5	3	2	0
Registra Tiendas.	5	3	2	0
Registra Clientes.	5	3	2	0
Registra Inventario (Videojuegos X Tiendas).	10	7	3	0
Consultar Tipos de Videojuegos.	5	3	2	0
Consultar Videojuegos.	5	3	2	0
Consultar Administradores de las Tiendas.	5	3	2	0
Consultar Tienda.	5	3	2	0
Consultar Cliente.	5	3	2	0
Consultar Inventario (Videojuegos X Tiendas).	10	7	3	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
TOTAL	100			