

FABRICA DE JUEGOS: PART 2

[Retoques que se hicieron con respecto al TP°3:

-Se agregó una excepción nueva “ProductoExistenteException” que se lanza cuando se quiere insertar un producto ya existente dentro de la base de datos.

-A la clase abstract Producto se le agregó el atributo “controlado” que es del tipo bool.

Utilizado para saber si al producto ya se le ha controlado su calidad.

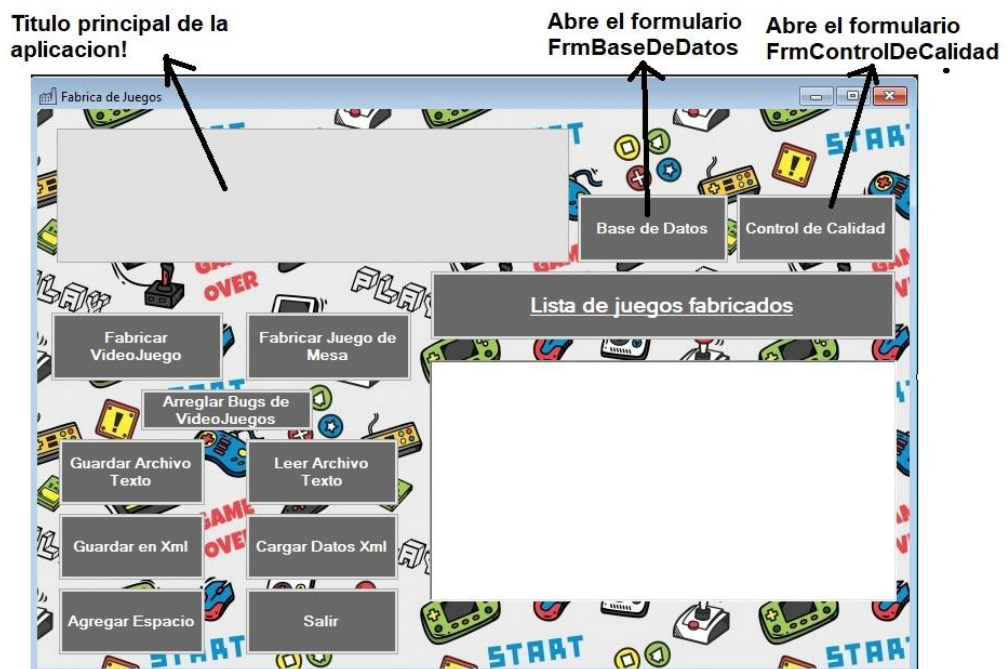
-Se agregó la clase estática FabricaDAO en entidades.

-Se agregó dos formularios nuevos: FrmBaseDeDatos y FrmControlDeCalidad.

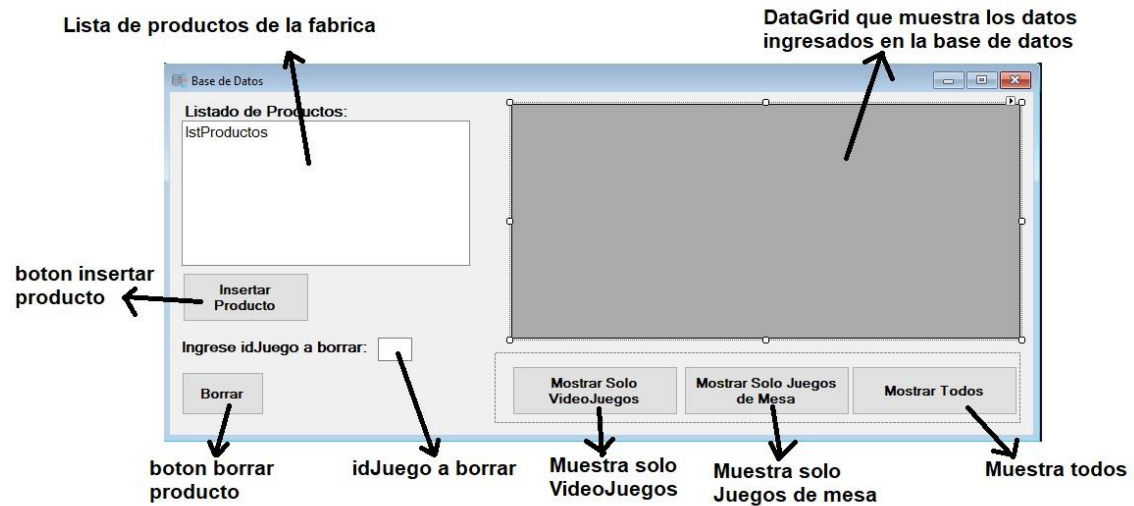
-Al FrmPrincipal y FrmBugs se le agregó el atributo “flagGuardado” bool para tener en cuenta los cambios que se generaron en la fabrica, ya sea un nuevo producto o el arreglo de bugs y notificar al usuario que si intenta salir del formulario si esta seguro de que desea salir sin guardar.

(detalle: Aunque el usuario no tenga obligación de hacerlo, se ruega guardar los datos xml para que haya consistencia de datos tanto de los archivos xml, texto, base de datos y archivos de control de calidad).]

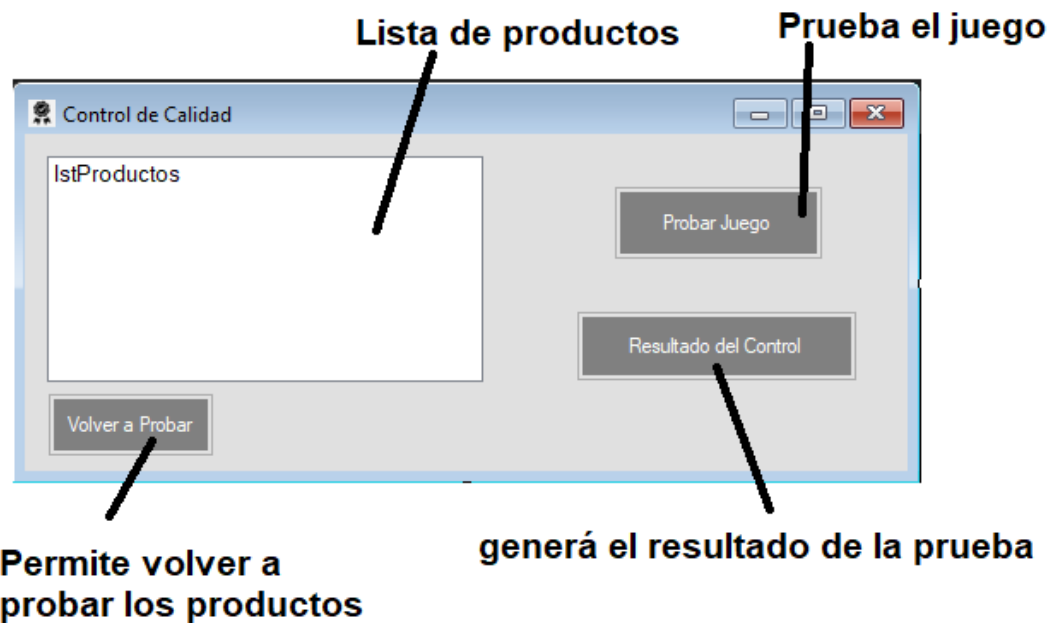
*FrmPrincipal actualizado:



*FrmBaseDeDatos:



FrmControlDeCalidad:



*FrmPrincipal:

Nuevos métodos creados por mi:

(Linea 305)Private void MostrarTitulo(): Muestra el titulo del formulario principal: "Fabrica de Juegos"

(Linea 336)Private void CambiarImagen(int numero): Cambia la imagen del titulo según el numero que se paso por parámetro.

*FrmBaseDeDatos : Formulario con el que se trabajo base de datos.

Metodos creados por mi:

(Linea 79)Public void ActualizarDatosDeTabla() : Actualiza los datos de la tabla DataGridView

(Linea 94)Public void CargarDatosALaTabla(): Carga los datos a la tabla DataGridView.

(Linea 148)Private void LimpiarTabla(): Limpia la tabla dataGridView.

(Linea 159)Private void MostrarPorTipo(): Muestra los productos en la tabla según su tipo.

*FrmControlDeCalidad: Formulario donde se controla la calidad de un producto.

Metodos creados por mi:

(Linea 103)Private void Probar() : método que simula el control de un juego. (El control dura 5 segundos)

(Linea 119)private void ProbarCalidadDelJuego(int tiempo) : método invocado por el evento eventoControlador que prueba la calidad de un juego.

(Linea 252)private void BorrarArchivoControl() : Borra el archivo de control, si es que ya se ha generado uno, para poder generar uno nuevo.

(Nota: Los productos cargados no están controlados, ni hay un archivo generado para poder probarlos y ver como se prueba y se genera el archivo.)

-SQL/Base de datos:

Clase estática FabricaDAO:

*Atributos:

Private static SqlConnection conexionSql;

Private static SqlCommand comandoSql;

Private static DataTable tabla;

*Constructor:

Static FabricaDAO(): Inicializa la conexionSql, comandoSql y la tabla. Tambien le setea la conexión instanciada al comandoSql y se le setea al tipo de comando el tipo de comando texto.

*Metodos:

-Public static DataTable CargarDatosALaTabla():

Carga los datos que se encuentran en la base de datos a la tabla.

-Public static DataTable CargarDatosALaTablaPorTipo(string tipoProducto):

Carga los datos en la tabla según el tipo de producto que se desee visualizar.

-public static bool BuscarProductoExistente(Producto producto):

Busca si el producto pasado por parámetro ya existe dentro de la base de datos.

Retorna false si no existe, de lo contrario, lanza la exception "ProductoExistenteException"

-public static int InsertarJuego(Producto producto)

Inserta un producto dentro de la base de datos.

-public static int BorrarJuegoPorId(int idJuego):

Borra un juego de la base de datos teniendo en cuenta el idJuego pasado por parámetro.

-Hilos:

Hilo utilizado en el FrmPrincipal:

Private Thread hiloTitulo : instanciado en el constructor del FrmPrincipal y se le cargo el método MostrarTitulo().

Se le da start() en el método load del formulario principal. (Linea 293).

Hilo utilizado en el FrmBaseDeDatos:

Private Thread hiloBaseDeDatos : instanciado en el constructor del FrmBaseDeDatos y se le cargo el método ActualizarDatosDeTabla(). (Linea 26).

Se le da start() en el metodo load del formulario BaseDeDatos.

Hilo utilizado en el FrmControlDeCalidad:

Private Thread hiloControlDeCalidad :

instanciado en el método btnProbarJuego_Click(Linea 62) y se inicia en el mismo

-Delegados:

Public delegate void DelegadoTitulo(int numero) : La firma se encuentra en el FrmPrincipal.

Utilizado para crear el evento private evento DelegadoTitulo eventoTitulo.

Public delegate void TablaDelegado() : La firma se encuentra en el FrmBaseDeDatos.

Utilizado para crear el evento private evento TablaDelegado eventoTabla.

Public delegate void CallBack(int tiempo): La firma se encuentra en el FrmControlDeCalidad.

Utilizado para crear el evento public evento Callback eventoControlador.

-Eventos:

Evento del FrmPrincipal:

Private event DelegadoTitulo eventoTitulo: Se le suscribe el método CambiarImagen(Linea 31) y es invocado en el método MostraTitulo() (Linea 311).

Evento del FrmBaseDeDatos:

Private event TablaDelegado eventoTabla: se le suscribe el método CargarDatosALaTabla() y

Es invocado desde el metodo ActualizarDatosDeTabla() (Lineas 41 y 85 respectivamente).

Evento del FrmControlDeCalidad:

Public evento CallBack eventoControlador: se le suscribe el método

ProbarCalidadDelJuego(int tiempo) y se lo invoca en el método Probar() (Lineas 59 y 109).

-Métodos de Extensión:

*Public static class Control:

utilizado principalmente en el FrmControlDeCalidad.

(línea 18)

Public static bool GuardarArchivoControl(this Producto producto, string textoCalidad):

Guarda un archivo de texto de los juegos controlados. (**Nota:** El archivo se guarda en la carpeta \Trabajo Practico N°4\FrmFabrica\bin\Debug)

(Linea 55)Public static string ControlarCalidadDeJuego(this Producto producto):

A partir de un numero random, se le asigna un texto con detalles genéricos sobre la calidad del juego y retorna ese texto.

Métodos privados dentro de la clase de extension Control:

(Linea 55)Private static string ControlVideoJuego(VideoJuego videojuego,int calidadNumerica):

Otorga mas detalles sobre la calidad siendo el producto un VideoJuego.

(Linea 129)Private static string ControlJuegoDeMesa(JuegoDeMesa juegoDeMesa, int calidadNumerica):

Otorga mas detalles sobre la calidad siendo el producto un JuegoDeMesa.

*public static class TipoDeJuego:

Public static string ObtenerTipoDeJuego(this Producto producto): utilizado en el método GuardarArchivoControl perteneciente a la extensión Control (Linea 40 de la clase static Control)

Retorna un string con el nombre del tipo de Producto.