



GUÍA DE APRENDIZAJE N° 19

1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

Programa de Formación: Técnico en programación de software	Código: Versión:	228120 102		
Nombre del Proyecto: Sistema de información para la gestión empresarial v 1.3	Código:	704330		
Fase del proyecto:Fase 2. Diseño o Formulación del Proyecto (Desarrollo)				
Actividad (es) del Proyecto: Desarrollo de interfaces y Validación de campos	Actividad (es) de Aprendizaje: Realizar pruebas de navegación, prueba de conexión, coherencia de información y manejo de errores en el proyecto	Ambiente de formación	MATERIALES DE FORMACIÓN	
			DEVOLUTIVO	CONSUMIBLE
Resultados de Aprendizaje: 22050100703 Construir el mapa de navegación de acuerdo con el diseño entregado para orientar al usuario en el uso del aplicativo	Competencia: 220501007 Desarrollar el sistema que cumpla con los requerimientos de la solución informática	Aula de Informática con excelente iluminación y ventilación. Conexión a internet ADSL, estable y con mínimo 1.5 megas de velocidad por equipo	Portátil o PC por aprendiz (Mínimo 25 equipos.) Equipos con software Específico instalado. (Visual Studio 2008 o superior y SQLServer2005 o superior) Video Beam y/o Smart Board	Marcadores Tinta para marcador Resma de papel
Duración de la guía (en horas):	40			



2. INTRODUCCIÓN

MANEJO DE EXCEPCIONES

En el lenguaje C#, los errores del programa se difunden en tiempo de ejecución a través del programamediante un mecanismo denominado excepciones. Las excepciones se producen cuando el código encuentra un error y se detectan mediante el código que puede corregir el error. Las excepciones se pueden producir mediante el CommonLanguageRuntime (CLR) de .NET Framework o mediante código de un programa. Una vez que se produce una excepción, ésta se difunde a la pila de llamadas hasta que se encuentra una instrucción catch para la excepción. Las excepciones no detectadas se identifican a través de un controlador de excepciones genérico proporcionado por el sistema que muestra un cuadro de diálogo. Las clases derivadas de Exception representan estas excepciones. Esta clase identifica el tipo de excepción y contiene las propiedades con detalles acerca de ésta. Producir una excepción implica crear una instancia de una clase derivada de la excepción, configurar opcionalmente las propiedades de la excepción y a continuación, producir el objeto con la palabra clave throw. Las características de control de excepciones del lenguaje C# proporcionan una manera de afrontar cualquier situación inesperada o excepcional que se presente mientras se ejecuta un programa. El control de excepciones utiliza las palabras clave try, catch y finally para intentar acciones que podrían no realizarse correctamente, controlar errores y limpiar los recursos después. CommonLanguageRuntime (CLR), las bibliotecas de otro fabricante o el código de aplicación que utiliza la palabra clave throw pueden generar excepciones.

Las excepciones tienen las propiedades siguientes:

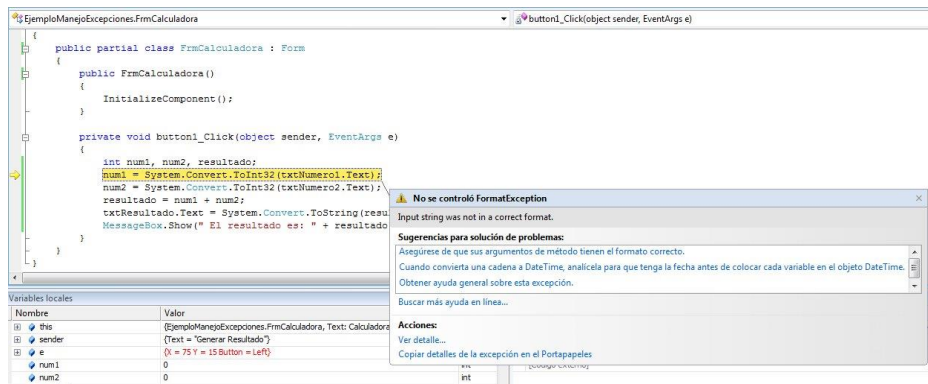
- Cuando la aplicación encuentra una circunstancia excepcional, como una división por cero o una advertencia de que no hay suficiente memoria, se genera una excepción.
- Utilice un bloque try alrededor de las instrucciones que puedan generar excepciones.
- Cuando se produce una excepción dentro del bloque try, el flujo de control salta inmediatamente a un controlador de excepciones asociado, si existe alguno.
- Si no hay un controlador de excepciones para una excepción determinada, el programa deja de ejecutarse y presenta un mensaje de error.
- Si un bloque catch define una variable de excepción, puede utilizar dicho bloque para obtener más información sobre el tipo de excepción que se ha producido.
- Las acciones que pueden producir una excepción se ejecutan con la palabra clave try.
- Un controlador de excepciones es un bloque de código que se ejecuta cuando se produce una excepción. En C#, la palabra clave catch se utiliza para definir un controlador de excepciones.
- Un programa que utiliza la palabra clave throw puede generar explícitamente excepciones.
- Los objetos de excepción contienen información detallada sobre el error que incluye el estado de la pila de llamadas y una descripción de texto del error.



- El código se ejecuta en un bloque finally aunque se produzca una excepción, permitiendo así que el programa libere recursos.

Generalmente las Excepciones son “Errores” que suceden en tiempo de ejecución del programa, deteniendo dicha ejecución cuando el algoritmo realizado detecta dicho error y este interrumpe el programa por completo, siempre se debe evitar y realizar un algoritmo que permita controlar de forma muy técnica dichos errores causados que pueden pasar por varias situaciones descritas anteriormente.

Ejemplo- No se controla una excepción:



PUNTOS DE INTERRUPCIÓN

Los puntos de interrupción nos permiten verificar hasta la línea de código marcado como está funcionando lógicamente paso a paso nuestro código.

Los puntos de interrupción o símbolos son los siguientes:

Glifo

Descripción



Punto de interrupción normal. El glifo relleno indica que el punto de interrupción está habilitado. El glifo vacío indica que está deshabilitado.



Punto de interrupción avanzado. Habilitado/deshabilitado. El signo + indica que el punto de interrupción tiene por lo menos una característica avanzada asociada a él, como por ejemplo una condición, número de llamadas o filtro.



Punto de interrupción asignado. Habilitado/deshabilitado. El punto de interrupción se establece en código de ASP/ASP.NET y se asigna a un punto de interrupción en la página HTML correspondiente, o se establece en un archivo de script de servidor y se asigna al archivo de script de cliente correspondiente.



Punto de seguimiento. Habilitado/deshabilitado. Cuando se llega a este punto, se realiza una acción específica pero no se interrumpe la ejecución del programa.



Punto de seguimiento avanzado. Habilitado/deshabilitado. El signo + indica que el punto de seguimiento tiene por lo menos una característica avanzada asociada a él, como una



condición, número de llamadas o filtro.



Punto de seguimiento asignado. Habilitado/deshabilitado. El punto de seguimiento se establece en código de ASP/ASP.NET y se asigna a un punto de seguimiento en la página HTML correspondiente.



Error de punto de interrupción o punto de seguimiento. La X indica que no se pudo establecer el punto de interrupción o punto de seguimiento debido a una condición de error.



Advertencia de punto de interrupción o de punto de seguimiento. El signo de admiración indica que no se pudo establecer un punto de interrupción o punto de seguimiento debido a una condición temporal. Normalmente, esto significa que no se ha cargado el código en la ubicación del punto de interrupción o punto de seguimiento. También podrá verlo si asocia un proceso y los símbolos del proceso no se cargan. Cuando se cargue el código o los símbolos, se habilitará el punto de interrupción y el glifo cambiará.



EJECUCIÓN PASO A PASO

Para ejecutar y verificar línea a línea puede dar clic en F10, si queremos retornar o ir a una función para evaluar los valores daremos F11.

3. ESTRUCTURACION DIDACTICA DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 Actividades de Reflexión inicial.

Después de realizar la lectura de la introducción de la guía, y la revisión del documento del material de apoyo participe en el **Foro Temático 3.1. Manejo de excepciones y puntos de interrupción**, dando su aporte a la preguntas planteadas a continuación:

¿Por qué es importante el manejo de excepciones en el desarrollo de un aplicativo de software?

¿Qué ventajas tiene el uso de puntos de interrupción en la ejecución de un aplicativo?

Nota: Argumente sus preguntas y realice un comentario al planteamiento de un compañero.

3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje.

En un documento en Word desarrolle las siguientes preguntas, dando las respuestas con sus propias palabras, desarrolle con gráficos explicativos.

- ¿Qué es depuración?
Defina en sus propias palabras que es un punto de interrupción.
- Defina ¿Qué es una Excepción?



- Consulte los tipos específicos de Excepciones más comunes.
- Consulte las palabras claves y la estructura para el manejo de las excepciones.

3.3 Actividades de apropiación del conocimiento (Conceptualización y Teorización).

- Después de leído el documento anexo en el material Cursos_csharp_4.- Instrucciones y excepciones.pdf, realice un documento que describa la metodología del manejo de excepciones e investigue y haga un cuadro comparativo de 10 tipos de errores que pueden surgir en la programación, que importancia tienen y las consecuencias que se tendrían al no tratarlos.
- Genere el listado de excepciones importantes para controlar los errores generados en la Aplicación que está elaborando.

3.4 Actividades de transferencia del conocimiento.

- Aplicar los puntos de interrupción y depuración paso a paso en su proyecto de formación.
- Realice el código necesario en C# para el control de las excepciones según el listado realizado en el literal 3.3.

3.5 Actividades de evaluación.

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento :	Identifica los conceptos de aplicación del tema.	Guía de aprendizaje Cuestionario
Evidencias de Desempeño:	A partir del proyecto formativo aplica de forma práctica.	Lista de chequeo
Evidencias de Producto:	Aplica en su proyecto el manejo de excepciones y punto de interrupción.	Lista de chequeo



4. RECURSOS PARA EL APRENDIZAJE

ACTIVIDADES DEL PROYECTO	DURACIÓN (Horas)	Materiales de formación devolutivos: (Equipos/Herramientas)		Materiales de formación (consumibles)		Talento Humano (Instructores)		AMBIENTES DE APRENDIZAJE TIPIFICADOS
		Descripción	Cantidad	Descripción	Cantidad	Especialidad	Cantidad	ESCENARIO (Aula, Laboratorio, taller, unidad productiva) y elementos y condiciones de seguridad industrial, salud ocupacional y medio ambiente
		Equipos de computo	25	Marcadores	4	Programación de Software	1	<ul style="list-style-type: none">• Aula de Informática• Computadores• Conexión a Internet• Ventilación e iluminación Adecuada
		Video Beam y/o Smart Board	1	Tinta para marcadores	4			
				Resma de Papel	1			



5. GLOSARIO DE TERMINOS

Breakpoint o puntos de interrupción: es una pausa intencional y controlada durante la ejecución de un programa.

Manejo de excepciones: es una característica de algunos lenguajes de programación, que les permite manejar o controlar los errores en tiempo de ejecución. Proveen una forma estructurada de atrapar las situaciones completamente inesperadas, como también errores predecibles o resultados inusuales. Todas esas situaciones son llamadas "excepciones".

6. REFERENTES BIBLIOGRÁFICOS

Microsoft Developer Network. Puntos de interrupción. Consultado el 9 de Marzo de 2015, en <http://msdn.microsoft.com/es-es/library/4607yxb0.aspx>

Microsoft Developer Network. Puntos de interrupción. Consultado el 9 de Marzo de 2015, en [http://msdn.microsoft.com/es-es/library/ms173160\(v=vs.80\).aspx](http://msdn.microsoft.com/es-es/library/ms173160(v=vs.80).aspx)

Microsoft Developer Network. Control de errores y excepciones. Consultado el 9 de Marzo de 2015, en [https://msdn.microsoft.com/es-es/library/384webt3\(v=vs.90\)](https://msdn.microsoft.com/es-es/library/384webt3(v=vs.90))

Eric Gunnerson and Nick Wienholt, *A programmer's Guide to C# 5.0*, 2012, www.it-ebooks.info

7. CONTROL DEL DOCUMENTO (ELABORADA POR)

ELABORADA POR: LUCY VANEGAS

ACTUALIZADA POR: EDWIN CAÑON – Octubre de 2014 – FERNANDO TARAZONA Marzo 2015