



Microsoft Visual C# 2008

Curso de desarrollo de aplicaciones utilizando la tecnología de programación Microsoft .NET. El lenguaje utilizado es Visual C# 2008, cuyas particularidades se estudian en la primera parte del curso. Después el curso se centra en la construcción de aplicaciones Windows y, finalmente, se estudia en profundidad el acceso a bases de datos utilizando la tecnología ADO .NET.

45 horas lectivas

40 horas prácticas

Contenido

—El lenguaje Visual C# 2008

Estudia la sintaxis y características específicas del lenguaje de programación Visual C# 2008, prestando especial interés a su enfoque orientado a objetos.

-Introducción a .NET

Se presenta la tecnología .NET Framework, que facilita el desarrollo de aplicaciones informáticas. Esta tecnología es la base sobre la que se trabaja al utilizar cualquier lenguaje perteneciente a Visual Studio 2008, como puede ser Visual C# 2008.

-Entorno Integrado de Desarrollo (IDE)

Se presenta el entorno de desarrollo de Visual Studio 2008, que consta de un completo editor de código y muchas herramientas que facilitan el desarrollo, programación y depuración de aplicaciones informáticas. También se determinan los conceptos de solución y proyecto, así como el estudio del sistema de ayuda o biblioteca MSDN.

-Fundamentos de programación (I)

Estudia conceptos básicos de programación, como el de variable, constante y tipo de datos, particularizándolos para el lenguaje Visual C# 2008.

-Fundamentos de programación (II)

Estudia conceptos básicos de programación, como los de operadores, desbordamiento, conversiones implícitas y explícitas, expresiones lógicas, etc. Se indica su uso en el lenguaje Visual C# 2008.

-Arrays y estructuras de control

Introduce el concepto de array como estructura de datos; y las estructuras de control, que permiten modificar el flujo de ejecución de un programa.

-Métodos

Introduce el concepto de método como mecanismo para dividir un proyecto de programación en unidades más pequeñas y fáciles de manejar. Se proporciona la sintaxis adecuada en el lenguaje de programación Visual C# 2008.

-Pensar en objetos (I)

Introduce el concepto de objeto como elemento fundamental en el desarrollo de aplicaciones con Visual C# 2008. Describe las diferencias entre la programación orientada a objetos y la programación procedural.

-Pensar en objetos (II)

Explica cómo crear objetos de una determinada clase, así como el concepto de visibilidad y las propiedades de sólo lectura y sólo escritura.

-Herencia

Explica el concepto de herencia en un lenguaje de programación orientado a objetos y cómo utilizarla en Visual C# 2008. Se destacan los beneficios de utilizar esta característica en un proyecto de cierta envergadura.

-Interfaces y espacios de nombres

Estudia algunos conceptos propios de la tecnología de orientación a objetos, como las clases abstractas, las interfaces y los espacios de nombres. Se presentan tanto teóricamente como su implementación utilizando el lenguaje de programación Visual C# 2008.

-Aspectos avanzados de los arrays

Describe características avanzadas en el uso de arrays, como la posibilidad de crear arrays cuyos elementos son objetos de una determinada clase; arrays con más de una dimensión; y métodos con un número variable de parámetros.

-Tipos de datos como clases y la clase String

Explica el proceso de "boxing", que consiste en la posibilidad de tratar cualquier tipo de datos de valor como una clase (tipo de datos por referencia) con las ventajas y el coste que ello implica. También se estudia detenidamente la clase String, que facilita enormemente la manipulación de cadenas de caracteres, así como StringBuilder, que puede ser más eficaz que String en ciertas situaciones.

-Colecciones

Introduce el concepto de colección como una estructura de datos alternativa a los arrays, ya que proporcionan mayor funcionalidad. Se estudian los tipos de colecciones ArrayList y HashTable, cuya principal característica es que pueden almacenar elementos de cualquier tipo.

-Colecciones genéricas

Presenta las colecciones genéricas, que son un tipo de colección más eficaz que las clásicas, ya que el programador determina el tipo de elementos que puede almacenar en ellas, lo que permite un código más seguro y eficaz.

—Windows Forms

Describe el desarrollo de aplicaciones Windows utilizando la tecnología Windows Forms y el lenguaje de programación Visual C# 2008. Se centra principalmente en la confección de la interfaz de usuario de este tipo de aplicaciones. También se introduce la tecnología alternativa, Windows Presentation Foundation.

-Introducción

Presenta las cualidades del entorno integrado de desarrollo Visual Studio 2008 para facilitar la confección de aplicaciones Windows. Se comprueban los beneficios de la programación visual en este tipo de proyectos de programación.

-Formularios

Las aplicaciones Windows se componen de una o varias ventanas que aparecen a medida que el usuario interactúa con ella. Las ventanas se conocen como formularios cuando se programa en Visual C# 2008 y en esta lección se estudian detalladamente.

-Trabajar con menús

Explica cómo crear un menú desplegable con el editor de menús de Visual Studio 2008. También se proporciona ejemplos de la creación de menús contextuales.

-Barras de herramientas y de estado

Explica cómo crear barras de herramientas en el entorno de desarrollo Visual Studio 2008. También se estudia el cuadro de diálogo Abrir (OpenFileDialog) y las barras de estado.

-Cuadros de diálogo

Explica las características que debe cumplir una ventana de una aplicación para actuar como cuadro de diálogo. Presenta algunos cuadros de diálogo "prefabricados", como MessageBox, el cuadro Fuente (FontDialog) y el de Color (ColorDialog).

-Controles básicos (I)

Introduce el concepto de control, que son aquellos elementos gráficos que aparecen en los formularios de un proyecto de programación en Visual Studio 2008 y que sirven para obtener datos y presentar la salida que produce la aplicación. Se estudian algunos, como las etiquetas (Label), cuadros de texto (TextBox), GroupBox y botones de comando (Button).

-Controles básicos (II)

Estudia el uso de los controles casilla de verificación (CheckBox), botón de opción (RadioButton), cuadro de lista (ListBox) y cuadro combinado (ComboBox) en un proyecto de Visual Studio 2008.

-Características gráficas (I)

Presenta las características gráficas de .NET Framework, utilizando para ello el lenguaje de programación Visual C# 2008. Estudia la clase Graphics y explica cómo dibujar líneas, rectángulos, polígonos, elipses y círculos, además de especificar colores.

-Características gráficas (II)

Se explica cómo utilizar las características gráficas de .NET Framework para dibujar arcos, curvas, trazados y rellenos. Se utiliza el lenguaje Visual C# 2008 para ello.

-Características gráficas (III)

Se estudian algunas características gráficas avanzadas de .NET Framework: aplicar transformaciones, dibujar texto, evitar el aliasing y dibujar imágenes. Se utiliza el lenguaje Visual C# 2008 para los ejemplos.

-Eventos de ratón y teclado

Se estudia cómo programar los eventos de ratón y teclado en una aplicación Windows. También se presenta una estrategia para conseguir la persistencia del dibujo en pantalla.

-Aplicaciones MDI

Se estudia el concepto de aplicación MDI, en la que encontramos una ventana principal y el resto se sitúan en el interior de ésta, manteniendo una relación principal-secundaria (padre-hijo). Se utiliza el lenguaje Visual C# 2008 para desarrollar una sencilla aplicación MDI.

-Excepciones

Explica el concepto de excepción como método que proporciona .NET Framework para informar de la ocurrencia de un error en tiempo de ejecución. Se estudia cómo se pueden manejar, de forma que la aplicación esté preparada para estas situaciones.

-Introducción a Windows Presentation Foundation

Presenta la nueva tecnología de Microsoft para crear aplicaciones que se caracterizan por tener una interfaz de usuario de gran riqueza y en las que se trabaja con distintos tipos de contenido. Esta tecnología es Windows Presentation Foundation o WPF, que coexiste con Windows Forms.

—ADO .NET

Describe la tecnología ADO .NET como medio fundamental de acceso a bases de datos cuando se está desarrollando aplicaciones con Visual C# 2008. Se estudian los dos enfoques de acceso a datos: conectado y desconectado.

-Bases de datos

Muestra las herramientas que proporciona Visual Studio 2008 para trabajar visualmente con bases de datos. Se utiliza el servidor de bases de datos SQL Server 2008 para crear una base de datos de ejemplo, con sus tablas, relaciones y otras propiedades.

-El lenguaje SQL

Repasa la sintaxis del lenguaje de consulta de bases de datos relacionales SQL, tanto para consultas de selección como de modificación de la base de datos.

-Introducción a ADO .NET

Presenta la arquitectura de ADO .NET, que permite desarrollar aplicaciones aplicando dos enfoques distintos de acceso a datos: conectado y desconectado. Estudia los objetos básicos de ADO .NET, como la conexión, el comando y el lector de datos. Utiliza el lenguaje de programación Visual C# 2008 para los ejemplos.

-Proveedor de datos

Se estudian las clases que componen un proveedor de datos, prestando especial atención a cómo utilizar parámetros en las consultas efectuadas contra la base de datos, así como el uso de transacciones para asegurar la consistencia de la misma.

-El objeto DataSet

Estudia el concepto de DataSet, como estructura que proporciona ADO .NET para el acceso desconectado a datos. Se comprueba que el DataSet está compuesto de una completa jerarquía de objetos y colecciones que se pueden manejar desde el código: tablas, relaciones, columnas, registros, etc. Es decir, que permite representar la misma estructura que tenemos en la base de datos subyacente.

-Programar el DataSet

Se estudia cómo actualizar el DataSet desde el código, es decir, se proporciona el código necesario para añadir nuevos registros al conjunto de datos, para modificarlos y para eliminarlos. Para ello, se utiliza el lenguaje Visual C# 2008.

-Acceso a datos con Visual Studio (I)

Presenta las herramientas que proporciona Visual Studio 2008 para desarrollar aplicaciones con acceso a datos. Explica el concepto de DataSet tipificado (typed DataSet) y las diferencias con los DataSet genéricos. También se presenta cada uno de los objetos que crea Visual Studio para facilitar la programación de este tipo de aplicaciones: TableAdapter, TableAdapterManager, BindingSource y BindingNavigator.

-Acceso a datos con Visual Studio (II)

Estudia los objetos y controles que crea Visual Studio al utilizar las herramientas de acceso a datos, como el Asistente para configurar un origen de datos y la ventana Data Sources. Se explica la razón de ser de estos objetos y cómo utilizarlos.

-Escenarios de enlace de datos (I)

Describe escenarios o situaciones que se presentan habitualmente durante el desarrollo de aplicaciones con acceso a datos desarrolladas mediante las herramientas de Visual Studio. Se explica cómo afrontar estas situaciones cotidianas.

-Escenarios de enlace de datos (II)

Describe escenarios o situaciones que se presentan habitualmente durante el desarrollo de aplicaciones con acceso a datos desarrolladas mediante las herramientas de Visual Studio. Se explica cómo afrontar estas situaciones cotidianas, como el uso de parámetros en las consultas para rellenar el conjunto de datos.

-Escenarios de enlace de datos (III)

Describe escenarios o situaciones que se presentan habitualmente durante el desarrollo de aplicaciones con acceso a datos desarrolladas mediante las herramientas de Visual Studio. Se describe el proceso de actualizar el conjunto de datos modificando, eliminando y añadiendo nuevos registros, así como actualizar la base de datos con esos cambios.

-Escribir código independiente de la base de datos

Introduce el concepto de la factoría de proveedores DbProviderFactory de ADO .NET, especialmente creada para el desarrollo de aplicaciones independientes de la base de datos que se utilice; esto es, que el código escrito sirva independientemente de que se utilice una base de datos SQL Server, Oracle, Access, etc.