

Developing Microsoft .NET Applications for Windows (Visual Basic .NET) - N° 2565

ALCANCES Y OBJETIVOS:

After completing this course, students will be able to:

- Create and populate Windows forms.
- Organize controls on Windows forms.
- Create menus in a Windows Forms application.
- Add code to form and control event procedures in a Windows Forms application.
- Create multiple-document interface (MDI) applications.
- Use dialogs in Windows Forms applications.
- Validate user input in a Windows Forms application.
- Create and use user controls in a Windows Forms application.
- Create licenses for controls.
- Bind Windows Applications to various data sources using Microsoft ADO.NET.
- Consume Web services with Windows Forms applications.
- Use .NET and COM components in a Windows Forms application.
- Call Win32[®] application programming interfaces (APIs) from a Windows Forms application.
- Migrate Visual Basic 6.0 applications to Visual Basic .NET.
- Print documents in a Windows Forms application.
- Create and format reports using Crystal Reports.
- Create multithreaded Windows Forms application.
- Debug a Windows Forms application.
- Incorporate accessibility features in a Windows Forms application.
- Localize a Windows Forms application.
- Create help files in a Windows Forms application.
- Deploy Windows Forms application.
- Implement code-access and role-based security in a Windows Forms application.
- Add deployment flexibility to applications by using shared assemblies.

REQUISITOS:

The course assumes that students have the following skills:

- Understanding of Visual Basic .NET/C# programming
- Experience building applications

The following prerequisites are recommended for this course:

- Course 2559—[Introduction to Visual Basic .NET Programming](#)
- OR
- Equivalent knowledge

DURACION:

24 horas

CONTENIDO:

Module 1: Introducing Windows Forms

This module introduces Windows forms and controls in the .NET framework. Students will learn how to create and populate base forms and inherited forms using the Visual Studio .NET framework. They will also learn how to add controls to a form and how to display and hide forms.

Lessons

- Creating a Form
- Adding Controls to a Form
- Creating an Inherited Form
- Organizing Controls on a Form
- Creating MDI Applications

Lab 1: Creating Windows Forms

- Creating a New Windows Form
- Inheriting a New Form from an Existing Windows Form

After completing this module, students will be able to:

- Create a form and add controls to it.
- Create an inherited form using Visual Inheritance.
- Organize controls on a form.
- Create Multiple Document Interface (MDI) applications.

Module 2: Working With Controls

This module explains how to create event procedures (handlers) in your application that will run in response to user actions. Students will learn how to add programming logic to the event procedures of a control, how to use the Windows Forms intrinsic controls, dialog boxes, and menus, and how to validate the data entered by users of your application.

Lessons

- Creating an Event Handler for a Control
- Using Windows Forms Controls
- Using Dialogs
- Validating User Input
- Creating Controls at Run Time
- Creating Menus

Lab 2: Working with Controls

- Creating and Using Controls

After completing this module, students will be able to:

- Create an event handler for a control.
- Select and use the appropriate controls in a Windows Forms application.
- Use dialog boxes in a Windows Forms application.
- Validate user input in a Windows Forms application.
- Add controls to a form at run time.
- Create and use menus in a Windows Forms application.

Module 3: Building Controls

This module explains how to extend the functionality of an existing Windows control, combine multiple existing controls into a composite control, and build a new custom control. Students will also learn how to add design-time licensing support to a control.

Lessons
<ul style="list-style-type: none"> ▪ Options for Creating Controls ▪ Adding Functionality to Controls ▪ Adding Design-Time Support for Controls ▪ Licensing a Control
Lab 3: Building Controls
<ul style="list-style-type: none"> ▪ Declare an Event and Raising It from an Extended Control ▪ Creating a Composite Control ▪ Adding Design-Time Support

After completing this module, students will be able to:

- Create a composite control by combining functionality of several existing Windows Forms controls.
- Describe the design-time support options for components provided by Microsoft Visual Studio® .NET.
- Add Attributes that provide information to the Visual Designer.
- Create and validate licenses for controls.

Module 4: Using Data in Windows Forms Applications

This module explains how to bind Windows forms to various data sources using ADO .NET. Students will learn about binding Windows forms to data from sources such as databases and XML files. Students will also learn how to display data using the Visual Studio.NET data controls.

Lessons
<ul style="list-style-type: none"> ▪ Adding ADO.NET Objects to a Windows Forms Application ▪ Accessing and Modifying Data by Using DataSets ▪ Binding Data to Controls ▪ Using the DataGrid Control ▪ Overview of XML Web Services ▪ Creating a Simple XML Web Services Client

Lab 4.1: Accessing Data with ADO.NET
<ul style="list-style-type: none"> ▪ Generating, Populating, and Persisting DataSets
Lab 4.2: Calling an XML Web Service
<ul style="list-style-type: none"> ▪ Calling an XML Web Service

After completing this module, students will be able to:

- Describe the objects in the ADO.NET object model.
- Add and configure ADO.NET objects in a Windows Forms application.
- Access and modify data from a database by using DataSets.
- Bind data to controls.
- Use the **DataGrid** control.
- Describe the XML Web services model and the roles of HTML, SOAP, XML, and marshaling in the XML Web services model.
- Create and test a simple XML Web service client application.
- Persist data to and read data from files and isolated storage by using methods of the **BinaryReader** and **BinaryWriter** methods and the **TextReader** and **TextWriter** methods.

Module 5: Interoperating with COM Objects

This module explains how to use .NET and COM components in their Windows Forms application. They will also learn how to call Win32 APIs in their Windows Forms application.

Lessons
<ul style="list-style-type: none"> ▪ Using .NET and COM Components in a Windows Forms Application ▪ Calling Win32 APIs from Windows Forms Applications
Lab 5: Interoperating with COM and Calling Win32 APIs
<ul style="list-style-type: none"> ▪ Using a COM Component in a .NET Application

After completing this module, students will be able to:

- Use .NET and COM components in a Microsoft .NET Framework Windows Forms application.
- Call Microsoft Win32® application programming interfaces (APIs) from a Windows Forms application.

Module 6: Reporting and Printing in Windows Forms Applications

This module explains how to create reports in a Windows Forms application using Crystal Reports and how to implement printing in a Windows Forms application.

Lessons
<ul style="list-style-type: none"> ▪ Creating Reports Using Crystal Reports ▪ Printing From a Windows Forms Application

<ul style="list-style-type: none"> Using the Print Preview, Page Setup, and Print Dialogs Constructing Print Document Content Using GDI+
Lab 6: Printing Formatted Documents
<ul style="list-style-type: none"> Adding Print Support to an Application Creating Printed Output by Using GDI+

After completing this module, students will be able to:

- Create and format reports using Crystal Reports.
- Print documents in a Windows Forms application.
- Use the Visual Studio .NET printing dialogs in a Windows forms application.
- Use GDI+ to construct print document content.

Module 7: Asynchronous Programming

This modules explains how to use the techniques of asynchronous programming and multithreading to avoid blocking the user interface of an application.

Lessons
<ul style="list-style-type: none"> The .NET Asynchronous Programming Model The Asynchronous Programming Model Design Pattern How to Make Asynchronous Calls to Methods
Making Asynchronous Calls to an XML Web Service
<ul style="list-style-type: none"> Converting Synchronous Calls to Asynchronous Calls

After completing this module, students will be able to:

- Describe the .NET Framework asynchronous programming model.
- Modify a client application to use built-in .NET Framework support for asynchronous calls to methods.
- Describe how to add explicit support for asynchronous calls to any method.

Module 8: Enhancing the Usability of Applications

This module explains support that is available in the Microsoft® .NET Framework, which allows developers to more easily enhance the usability of applications they develop.

Lessons
<ul style="list-style-type: none"> Adding Accessibility Features Adding Help to an Application Localizing an Application
Lab 8: Enhancing the Usability of an Application
<ul style="list-style-type: none"> Adding Support for Accessibility Adding Help to an Application Adding ToolTips to an Application

--

After completing this module, students will be able to:

- Use .NET Framework features to add and enable accessibility features in an application.
- Add support for context-sensitive help, Help menus, and tool tips to an application.
- Use localization properties and resource files to create a localized version of a .NET Framework Windows Forms application.

Module 9: Deploying Windows Forms Applications

This module explains assemblies and how to use strong-named assemblies and the Global Assembly Cache (GAC) in the Microsoft® .NET Framework. Students will also learn how to configure and deploy your .NET applications.

Lessons
<ul style="list-style-type: none"> .NET Assemblies Deploying Windows Forms Applications
Lab 9: Deploying an Application
<ul style="list-style-type: none"> Building and Referencing a Strong-Named Assembly Installing a Strong-Named Assembly into the GAC Deploying a .NET Application Using an Application Configuration File

After completing this module, students will be able to:

- Use strong-named assemblies in .NET applications.
- Use application configuration files to configure and use Microsoft Windows® Installer 2.0 to package and deploy .NET applications

Module 10: Securing Windows Forms Applications

This module explains the .NET Framework security model. Students will learn how to use .NET Framework security features in applications.

Lessons
<ul style="list-style-type: none"> Security in the .NET Framework Using Code Access Security Using Role-Based Security
Lab 10: Adding and Testing Permission Requests
<ul style="list-style-type: none"> Adding and Testing Permission Requests

After completing this module, students will be able to:

- Define evidence and describe its role in the security system in the .NET Framework.

-
- Define Authentication and Authorization and describe their roles in the security system in the .NET Framework.
 - List the major characteristics of code access security and role-based security.
 - Describe the .NET Framework security model.