

## Diplomado Web Programming with C# 2010 Using OOP, ASP.NET MVC and AJAX

**Duración:** 105 horas (3 Módulos de 35 horas cada uno)

**Modalidades:**

|                               |         |               |
|-------------------------------|---------|---------------|
| Lunes a Viernes               | Mañanas | 09:00 a 16:00 |
| Martes y Jueves               | Tardes  | 17:00 a 21:00 |
| Lunes y Miércoles             | Tardes  | 17:00 a 21:00 |
| <b>¡Nuevo!</b> Lunes a Jueves | Tardes  | 17:00 a 20:00 |
| Sábados                       | Mañanas | 09:00 a 15:00 |

**Objetivo:** Mostrar las técnicas necesarias para un programador .NET con especialidad en desarrollo Web utilizando programación orientada a objetos, el patrón de diseño MVC en ASP.NET y AJAX.

**Descripción:** Con la creciente demanda de aplicaciones Web, el tipo de tecnologías, técnicas y herramientas usadas para el desarrollo estas aplicaciones han ido evolucionando considerablemente. Dentro de estas evoluciones han surgidos conceptos tales como la programación orientada a objetos, el patrón de diseño MVC (Modelo – Vista – Controlador) y AJAX. Este diplomado pretende mostrar al participante tales conceptos a fin de poder intervenir el desarrollo y mantenimiento de este tipo de aplicaciones.

El diplomado está estructurado de manera tal que el participante conozca de manera gradual como cada uno de estos componentes (Programación Orientada a Objetos, ASP.NET MVC y AJAX) intervienen en la arquitectura de este tipo de aplicaciones.

**Requisitos:**

- ✓ Conocimientos de un lenguaje de programación ya sea estructurado u orientado a objetos. Si bien no es requerido que el participante conozca C# si es altamente recomendable que conozca conceptos básicos de algún lenguaje de programación, no necesariamente orientada a objetos. En otras palabras este diplomado está orientado a programadores ya formados. Para personas que inician en un primer lenguaje de programación no es recomendable.
- ✓ Este curso no es introductorio al desarrollo Web, asume Conocimientos de tecnologías fundamentales para el desarrollo Web tales como HTML, CSS, JavaScript y DHTML.
- ✓ Nociones (no dominio) de conceptos generales de un entorno Web tales como, peticiones HTTP, cookies, programación del lado servidor, programación del lado cliente
- ✓ Conocimientos de elementos fundamentales de bases de datos: tablas, llaves primarias y llaves foráneas.

**Audiencia:** Para programadores con los siguientes perfiles:

- ✓ Que deseen iniciarse en el desarrollo de aplicaciones Web bajo la plataforma .NET con C#, técnicas de programación orientada a objetos, el patrón de diseño MVC y AJAX.
- ✓ Programadores de otras tecnologías orientadas a Web que tengan interés en conocer la plataforma de desarrollo .NET.
- ✓ Desarrolladores de aplicaciones ASP y ASP.NET (en versiones anteriores) que deseen conocer las nuevas herramientas y tecnologías ofrecidas por ASP.NET 4.0 así como patrones de diseño tales como MVC orientados a formar una arquitectura más robusta que la ofrecida en versiones anteriores.
- ✓ Desarrolladores de ASP.NET con Web Forms y que deseen conocer el nuevo marco de trabajo MVC, presentado en la nueva versión de Visual Studio, la 2010.

# Módulo 1: Object Oriented Programming with C# 2010

**Objetivo:** Presentar el alcance de la plataforma .NET, los componentes que la integran, los tipos de aplicaciones que se pueden desarrollar y sobre todo la función de C# como lenguaje base para la integración de dichos componentes.

**Descripción:** Es un curso adecuado para toda aquella persona que se inicia en el desarrollo de aplicaciones .NET. Su principal objetivo es presentar la sintaxis del lenguaje, a fin que el alumno posteriormente pueda integrar los componentes que formaran una aplicación completa. Es un curso elemental y altamente recomendado para manejar con soltura cualquier curso o temas especializados de .NET tales como ASP.NET, ADO.NET, LINQ, AJAX, WCF, WPF entre muchos más.

**Audiencia:** Programadores o desarrolladores en cualquiera de los siguientes perfiles:

- ✓ Tienen la necesidad de actualizar sus conocimientos de lenguajes de programación tales como VB 6.0, Cobol, C++, Delphi.
- ✓ Conocen otras lenguajes orientados a objetos tales como Java, VB.NET y desean conocer la sintaxis y filosofía de C#.
- ✓ Tienen experiencia en versiones anteriores del lenguaje y desean conocer las nuevas características que éste ofrece.
- ✓ Requieren crear objetos de negocios para implementar una solución basada en componentes y por lo tanto es fundamental conocer la teoría e implementación de técnicas de programación orientada a objetos.
- ✓ Requieren dar mantenimiento a aplicaciones escritas en C#.

**Requisitos:** Dado que no es un curso de introducción a la programación se requiere que el alumno:

- ✓ Maneje conceptos esenciales de programación (ya sea estructurada u orientada a objetos).
- ✓ Este familiarizado con el desarrollo de algoritmos y la implementación de estos en un lenguaje en particular.
- ✓ En general que ya tenga experiencia en otros lenguajes de programación.

# CONTENIDO

## 1. NET: What You Need to Know

- Getting Started
- .NET: What is Really Happening
- .NET Programming in a Nutshell
- Viewing the Assembly
- Viewing Intermediate Language
- Understanding .NET
- Visual Studio 2010
- Creating a Console Application
- Adding a C# file
- Using the Visual Studio Text Editor
- IntelliSense
- Build and Run the Project
- Pausing the Output
- Visual C# and GUI Programs
- .NET Documentation

## 2. First C# Programs

- Hello, World
- Compiling, Running (Command Line)
- Program Structure
- Namepaces
- Variables
- Expressions
- Assignment
- Calculations Using C#
- More About Output in C#
- Input in C#
- More About Classes
- InputWrapper Class
- Echo Program
- Using InputWrapper
- Compiling Multiple Files
- Multiple Files in Visual Studio
- The .NET Framework

## 3. Data Types in C#

- Strong Typing
- Typing in C#
- Typing in C++
- Typing in Visual Basic 6
- C# Types
- Integer Types
- Integer Type Range
- Integer Literals
- Floating Point Types
- Floating Point Literals
- IEEE Standard for Floating Point

- Decimal Type
- Decimal Literals
- Character Type
- Character Literals
- string
- Escape Characters
- Boolean Type
- Implicit Conversions
- Explicit Conversions
- Nullable Types

## 4. Operators and Expressions

- Operator Cardinality
- Arithmetic Operators
- Multiplication
- Division
- Additive Operators
- Increment and Decrement
- Relational Operators
- Conditional Logical Operators
- Short-Circuit Evaluation
- Ternary Conditional Operators
- Bitwise Operators
- Bitwise Logical Operators
- Bitwise Shift Operators
- Assignment Operators
- Expressions
- Precedence
- Associativity
- Checking

## 5. Control Structures

- If Test
- Blocks
- Loops
- while Loop
- do while Loops
- for Loops
- Arrays
- foreach Loop
- break
- continue
- goto
- Structure Programming
- Multiple Methods
- switch
- switch in C# and C/C++

## **6. Object-Oriented Programming**

- Objects
- Objects in the Real World
- Object Models
- Reusable Software Components
- Objects in Software
- State and Behavior
- Abstraction
- Encapsulation
- Classes
- Inheritance Concepts
- Relationships among Classes
- Polymorphism
- Object-Oriented Analysis and Design
- Use Cases
- CRC Cards and UML

## **7. Classes**

- Classes as Structure Data
- Classes and Objects
- References
- Instantiating and Using an Object
- Assigning Object References
- Garbage Collection
- Methods
- Public and Private
- Abstraction
- Encapsulation
- Initialization
- Initialization with Constructors
- Default Constructor
- this
- Static Fields and Methods
- Static Methods
- Static Constructor
- Constant and Readonly Fields

## **8. More about Types**

- Overview of Types in C#
- Structures
- Uninitialized Variables
- Coping a Structure
- Hotel.cs
- HotelCopy.cs
- Classes and Structs
- Enumeration Types
- Reference Types
- Class Types
- object
- string
- Arrays

- Default Values
- Boxing and Unboxing
- Implicitly Types Variables

## **9. Methods, Properties, and Operators**

- Static and Instance Methods
- Method Parameters
- No “Freestanding” Functions in C#
- Classes with All Static Methods
- Parameter Passing
- Parameter Terminology
- Value Parameters
- Reference Parameters
- Output Parameters
- Structure Parameters
- Class parameters
- Method Overloading
- Modifiers as Part of the Signature
- Variable Length Parameter Lists
- Properties
- Auto-Implemented Properties
- Operator Overloading
- Operator Overloading in the Class Library

## **10. Characters and Strings**

- Characters
- Character Codes
- ASCII and Unicode
- Escape Sequences
- Strings
- String Class
- String Literals and Initialization
- Concatenation
- Index
- Relational Operators
- String Equality
- String Comparisons
- String Input
- String Methods and Properties
- StringBuilder Class
- StringBuilder Equality
- Command Line Arguments
- Command Line Arguments in the IDE
- Command Loops
- Splitting a String
- 11. Arrays and Indexers**
- Arrays
- One Dimensional Arrays
- System.Array
- Random Number Generation
- Next Methods

Jagged Arrays  
Rectangular Arrays  
Arrays as Collections  
Bank Case Study: Step 1  
Account Class  
Bank .Class  
TestBank Class  
Atm Class  
Running the Case Study  
Indexers  
Using the Indexer

## **12. Inheritance**

Inheritance Fundamentals  
Inheritance in C#  
Root Class – object  
Access Control  
Public Class Accessibility  
Internal Class Accessibility  
Member Accessibility  
Member Accessibility Qualifiers  
Method Hiding  
Method Hiding and Overriding  
Initialization  
Initialization Fundamentals  
Default Constructor  
Overloaded Constructors  
Invoking Base Class Constructors  
Bank Case Study: Step 2  
Bank Case Study Analysis  
Account  
CheckingAccount  
SavingsAccount  
TestAccount  
Running the Case Study

## **13. Virtual Methods and Polymorphism**

Introduction to Polymorphism  
Abstract and Sealed Classes  
Virtual Methods and Dynamic Binding  
Type Conversions in Inheritance  
Converting Down the Hierarchy  
Converting Up the Hierarchy  
Virtual Methods  
Virtual Method Cost  
Method overriding  
The Fragile Base Class Problem  
override Keyword  
Polymorphism  
Polymorphism Using “Type Tags”  
Polymorphism Using Virtual  
Abstract Classes

Sealed Classes  
Heterogeneous Collections  
Bank Case Study: Step 3  
Case Study Classes  
Run the Case Study  
Account  
CheckingAccount, SavingsAccount  
Bank and Atm  
TestBank

## **14. Formatting and Conversion**

Introduction to Formatting  
ToString  
ToString in Your Own Class  
Using Placeholders  
Format String  
Controlling Width  
Format String  
Currency  
String.Format  
PadLeft and PadRight  
Bank Case Study: Step 4  
Type Conversions  
Conversion of Built-In Types

## **15. Exceptions**

Introduction to Exceptions  
Exception Fundamentals  
.NET Exception Handling  
Exception Flow of Control  
Context and Stack Unwinding  
System.Exception  
User-Defined Exception Classes  
Structure Exception Handling  
Family Block  
Bank Case Study: Step 5  
Inner Exceptions  
Checked Integer Arithmetic

## **16. Interfaces**

Interfaces in C#  
Interface Inheritance  
Programming with Interfaces  
Implementing Interfaces  
Using an Interface  
Dynamic Use of Interfaces  
is Operator  
as Operator  
Bank Case Study: Step 6  
Common Interfaces in Case Study – IAccount  
Apparent Redundancy  
IStatement

IStatement Methods  
IChecking  
ISavings  
The Implementation  
SavingsAccount  
The Client  
Resolving Ambiguity  
Access Modifier

## **17. .NET Interfaces and Collections**

Collections  
Count and Capacity  
foreach Loop  
Array Notation  
Adding to the List  
Remove Method  
RemoveAt Method  
Collection Interfaces  
IEnumerable and IEnumerator  
ICollection  
IList  
A Collection of User-Defined Objects  
Duplicate Objects  
A Correction to AccountList (Step 1)  
Bank Case Study: Step 7  
Copy Semantics and ICloneable  
Copy Semantics in C#  
Shallow Copy and Deep Copy  
Reference Copy  
Memberwise Clone  
Using ICloneable  
Comparing Objects  
Sorting an Array  
Anatomy of Array.Sort  
Using the is Operator  
The Use of Dynamic Type Checking  
Implementing IComparable  
Running the Program  
Complete Solution  
Writing Generic Code  
Using a Class of object  
Generic Types  
Generic Syntax in C#  
Generic Client Code  
System.Collections.Generic

Object Initializers  
Collection Initializers  
Anonymous Types

## **18. Delegates and Events**

Overview of Delegates and Events  
Callbacks and Delegates  
Usage of Delegates  
Declaring a Delegate  
Defining a Method  
Creating a Delegate Object  
Calling a Delegate  
A Random Array  
Anonymous Methods  
Combining Delegate Objects  
Account.cs  
DelegateAccount.cs  
Lambda Expressions  
Named Method  
Events  
Events in C# and .NET

## **19. Introduction to Windows Forms**

Creating a Windows Forms App  
Partial Classes  
Windows Forms Event Handling  
Add Events for a Control  
Events Documentation  
Closing a Form  
ListBox Control

## **20. New Features in C#**

dynamic Type  
dynamic versus object  
Behavior of object  
Behavior of dynamic  
Names Arguments  
Optional Arguments  
Book Class  
Using Optional Arguments  
Automating Office with C# 4.0  
Automating Excel  
Automating Word  
Variance in Generic Interfaces  
Variance with IComparer<T>  
Interfaces with Variance Support

## Módulo 2: ASP.NET MVC Using C# (VS 2010)

**Objetivo:** Presentar la implementación del patrón de diseño MVC para aplicaciones ASP.NET con C#.

**Descripción:** Este curso proporciona una introducción práctica al desarrollo de aplicaciones Web usando ASP.NET MVC 2 con C#. Este es un nuevo framework de desarrollo para aplicaciones Web que difiere de manera importante con respecto anteriores opciones como Formas Web el cual se sigue soportando. Este curso cubre los fundamentos del patrón de diseño Modelo-Vista-Controlador y su implementación en ASP.NET MVC. Después de presentar los fundamentos de esta tecnología con varios ejemplos, se cubren a detalle los principales componentes de Modelo, Contralor y Vista. La discusión del modelo abarca tecnologías modernas de acceso a datos, incluyendo LINQ y el framework ADO.NET Entity 4.0. También se cubre el mecanismo de ruteo de ASP.NET MVC. El curso también presenta el concepto de pruebas unitarias automatizadas para aplicaciones Web, una de las ventajas notorias de esta nueva tecnología. Otro aspecto importante considerado es la seguridad en las aplicaciones a través de ejemplos de ataques y defensas. También aprenderás a usar las tecnologías de formas Web y MVC juntas en una misma aplicación. Y por último el curso concluye con el tema de distribución de tu aplicación.

**Audiencia:** Desarrolladores de aplicaciones experimentados o arquitectos responsables de crear aplicaciones Web usando la tecnología ASP.NET MVC con C#.

**Requisitos:** Dado que no es un curso de introducción al desarrollo Web se requiere que el alumno:

- ✓ Tecnologías propias del desarrollo de aplicaciones Web: HTML, CSS, JavaScript y DHTML.
- ✓ Dominio de la sintaxis de C# (Objetos, métodos, eventos, constructores, sobre carga, herencia, etc.).
- ✓ Uso de librerías del framework 4.0 de .NET.

También es altamente conveniente que el alumno tenga conocimientos básicos de Formas Web de ASP.NET.

# Contenido

## 1. INTRODUCTION TO ASP.NET MVC

- Review of ASP.NET Web Forms
- Advantages of ASP.NET Web Forms
- Disadvantages of ASP.NET Web Forms
- Model-View-Controller Pattern
- ASP.NET MVC
- Advantages of ASP.NET MVC
- Disadvantages of ASP.NET MVC
- Goals of ASP.NET MVC
- Unit Testing

## 2. GETTING STARTED WITH ASP.NET MVC

- Installing ASP.NET MVC
- Visual Studio ASP.NET MVC Project
- Starter Application
- Simple App with Controller Only
- Action Methods and Routing
- Action Method Return Type
- Rendering a View
- Creating a View in Visual Studio
- The View Web Page
- Dynamic Output
- Embedded Scripts
- Using a Model
- Controller Using the Model
- View Using the Model
- Strongly-Typed View
- Strongly-Typed View in Visual Studio

## 3. ASP.NET MVC ARCHITECTURE

- The Controller in ASP.NET MVC
- The View in ASP.NET MVC
- The Model in ASP.NET MVC
- How MVC Works
- Using Forms
- HTML Helper Functions
- Displaying Sign In Form
- Handling Form Submission
- Model Binding
- Greet View
- Input Validation
- Nullable Type
- Checking Model Validity
- Validation Summary

## 4. THE MODEL

- Microsoft Technologies for the Model
- SmallPub Database
- ADO.NET
- Language Integrated Query (LINQ)
- ADO.NET Entity Framework 4.0

- EDM Example
- SmallPub Tables
- SmallPub Entity Data Model
- Entity Data Model Concepts
- Querying the EDM
- Class Diagram
- Context Class
- List of Categories
- List of Books
- LINQ to Entities Demo
- IntelliSense
- Modifying a Data Source
- Entity Framework in a Class Library
- Data Access Class Library
- Client Code
- ASP.NET MVC Database Clients

## 5. THE CONTROLLER

- IController Interface
- A Low Level Controller
- Displaying Context
- Add and Subtract
- Controller Base Class
- Action Methods
- HomeController
- Math Controller
- Invoking MathController
- Receiving Input
- Non-Nullable Parameters
- Nullable Parameters
- Using a Model
- Action Results
- JavaScript Object Notation
- Action Method Attributes
- HTML Output
- Filters
- Asynchronous Controllers

## 6. THE VIEW

- View Responsibility
- A Program without a View
- A Program with a View
- View Page
- Passing Data to the View
- Passing Lists to the View
- Strongly-Typed Views
- Strongly-Typed View in Visual Studio
- Passing Data with Dynamic Objects
- More about Dynamic Objects
- HTML Helper Methods
- Link-Building Helpers
- Form Helpers
- Validation Helpers

- Form Helpers
- Templated Helpers

## 7. ROUTING

- Routing in ASP.NET MVC
- Properties of Routes
- Registering Routes
- Route Debugging Utility
- Multiple Matches
- Fixing the Bug
- Debugging Routes
- Areas

## 8. UNIT TESTING

- Visual Studio Unit Testing Framework
- Unit Testing Framework Namespace
- Assert Class
- Assert.AreEqual()
- More Assert Methods
- CollectionAssert Class
- StringAssert Class
- Test Case
- Test Methods
- Test Class
- Test Runner
- Command Line Test Runner
- Ignoring Tests
- Using the Ignore Attribute
- Grouping Test Results
- Inversion of Control
- Testing with Mocking
- ASP.NET MVC Test Projects
- Refactoring
- Test-Driven Development

## 9. SECURITY

- Web Security
- Input Forgery
- Cross-Site Scripting
- XSS Example

- Entering JavaScript
- The Attack
- Defending Against the Attack
- Session Hijacking
- Cross-Site Request Forgery
- Controller and Model Code
- View Code
- Attacker Code
- The Defense
- SQL Injection
- Using the MVC Framework Securely
- Authorize Attribute
- RequireHttps Attribute
- ValidateInput Attribute

## 10. COMBINING ASP.NET MVC AND WEB FORMS

- Using Web Forms in an MVC Application
- Linking and Redirecting from Web Forms Pages to MVC Actions
- Default.aspx
- Web Forms in MVC Example
- Sharing Data Between ASP.Net MVC and ASP.NET Web Forms
- Using Web Form Controls in MVC Views
- Using MVC in a Web Forms Application
- Steps to Create Hybrid MVC – Web Forms Application

## 11. DEPLOYMENT

- Server Requirements
- Continuous Integration
- Environment Configurations
- Configuration File Transformation
- MvcBuildViews for View Compile Checking
- Release Build View Compile Checking
- XCOPY
- Web Deploy

# Módulo 3: ASP.NET AJAX Using C# 2010

**Objetivo:** Mostrar los beneficios que se pueden obtener al aplicar AJAX en nuestros sistemas ASP.NET.

**Descripción:** Con la evolución que han sufrido el uso y desarrollo de aplicaciones Web han ido surgiendo nuevas tecnologías, herramientas y conceptos que buscan que usuario final de dichas aplicaciones las aproveche al máximo. Y aunque se pueden hacer muchas mejoras en las diferentes partes de la aplicación; sin duda alguna uno de los aspectos más importantes es la capa de presentación, que por su naturaleza es la parte de la aplicación con la cual el usuario final tiene un contacto directo.

En ese sentido ha surgido el concepto de aplicaciones RIA (Rich Internet Applications), este concepto busca entregar al usuario final una interfase gráfica que sea amigable, estética y funcional, aplicando diferentes tecnologías y soluciones. Una de esas soluciones es AJAX y particularmente ASP.NET AJAX.

En este curso aprenderás las herramientas que ASP.NET AJAX pone a tu alcance como desarrollador Web para explotar al máximo las capacidades del navegador y lograr aplicaciones RIA con un gran nivel de productividad. Aprenderás la arquitectura de ASP.NET AJAX y la función de dichos elementos para lograr beneficios tales como: codificación de lógica orientado a objetos del lado cliente, procesamiento de procesos asíncronos (una aplicación ASP.NET ya no tendrá que recargar nuevamente toda la página) y la utilización de varios controles que permitan páginas más estéticas y funcionales.

**Audiencia:** Desarrolladores de aplicaciones Web con ASP.NET 4.0 que busquen darle un plus a sus aplicaciones.

**Requisitos:** Dado que es un curso para enriquecer una tecnología, este caso ASP.NET se asume que el participante ya tiene amplio dominio en dicha tecnología así como en elementos generales de la plataforma .NET. En general se recomienda:

- Amplio dominio de la sintaxis general (declaración de variables, estructuras de controles, ciclos, etc.) y sintaxis orientada a objetos (clases, objetos, constructores, herencia, etc.) en C#. En caso de no conocer los temas anteriores te sugerimos primero considerar el curso Object Oriented Programming Using C# 4.0.
- Este curso aunque está orientado al desarrollo de aplicaciones Web, no es un curso introductorio al respecto; por lo mismo se asume que el participante tiene conocimientos de tecnologías fundamentales del desarrollo Web, tales como, HTML, JavaScript, CSS y DHTML. En caso de no conocer los temas anteriores te sugerimos primero considerar el curso Introducción al Desarrollo Web.
- Amplio dominio en los conceptos y conocimientos necesarios para el desarrollo de aplicaciones ASP.NET 4.0. En caso de no contar con dichos elementos te recomendamos el curso ASP.NET Using C# 2010 (ASP.NET 4.0). Dado que no es un curso de introducción al desarrollo Web se requiere que el alumno:

# Contenido

## **RICH INTERNET APPLICATIONS AND AJAX**

- Desktop Applications
- Web Applications
- Plug-Ins
- Client-Side Scripting
- JavaScript Example
- Script Code
- JavaScript in ASP.NET
- Dynamic Pages
- Efficient Page Redraws
- AJAX
- Google Maps
- ASP.NET AJAX
- ASP.NET Enhancements
- ASP.NET AJAX Control Toolkit
- Demo – Accessing a Database
- An ASP.NET AJAX-Enabled Website
- Summary

## **USING JAVASCRIPT**

- History and Standards
- Uses of JavaScript
- Example: JavaScript in WSH
- JavaScript in the Browser
- Embedded JavaScript
- .js Files
- JavaScript Language
- Variables
- Variables Example
- Reserved Words
- Types
- Strings
- Numbers
- Boolean
- Operators in JavaScript
- Logical Operators
- Bitwise and Assignment Operators
- Functions
- Scope
- Output from Scope Example
- Arrays
- Loops
- for/in Loops in Arrays
- Selection Structures
- Exceptions
- Error Object
- Objects in JavaScript
- Creating Your Own Objects
- Using Your Objects
- Dynamically Adding a Method
- for/in Loops with Objects

## **DHTML, DOM AND CSS**

- What is DHTML ?
- DHTML Components
- XHTML
- Document Object Model
- Browser Objects
- Accessing DOM Nodes
- Using jQuery
- Manipulating Elements: Properties
- Creating and Inserting Nodes
- CSS (Cascading Style Sheets)
- CSS Syntax
- Style Sheets
- Output without the Style Sheet
- Output with the Style Sheet
- StyleSheet1.css
- Output with a Different Style Sheet
- StyleSheet2.css
- CSS Class Example
- Using a CSS Class
- jQuery Library
- jQuery and Visual Studio
- Selectors
- jQuery Wrapper
- CSS Manipulation
- Adding and Removing Class Name
- More about Selectors
- Animation
- Animation Effects
- Modifying the Document Structure
- Remove and Append Code
- Microsoft Content Delivery Network
- Summary

## **MICROSOFT AJAX CLIENT LIBRARY**

- Microsoft AJAX Components
- AJAX Client Library
- Using the Client Library
- ScriptManager Control
- Embedded JavaScript Files
- Client Library Namespaces
- Sys.Debug Tracing
- AJAX Client Life Cycle Events
- Enable Script Debugging
- TraceConsole
- Extending JavaScript Objects
- Test Program for Array Extension
- Test Program Code
- Array Extensions in Client Library
- Object-Oriented JavaScript Class
- Person Class
- Account Class
- Test Code

- Namespaces
- Inheritance
- Book Class
- Initialization
- Other Functions
- User Interface
- Top-Level Function
- Shortcut Methods
- Populating a List Box
- JavaScript in Assemblies
- Providing a ScriptReference
- Enhanced AcmeClass Application

## **PARTIAL PAGE RENDERING**

- Partial Page Rendering
- UpdatePanel Control
- AJAX Extensions Controls
- Controlling Updates
- Triggers
- Types of Triggers
- Server Initiated Updates
- A Challenge
- Triggers Again
- Timer Control
- PageRequestManager Class
- Customization Scenarios
- PageRequestManger Event Handlers
- UpdateProgress Control
- Factors Code
- Canceling the Async Postback
- Limitations and Performance Issues

## **REMOTE METHOD CALLS**

- Why Remote Methods
- Designing Remote Methods
- A Web Service in an .asmx File
- Registering AJAX Web Services
- Calling AJAX Web Services
- Running the Application
- Handling Errors
- Context
- Modified Web Service
- Using Context
- Method Parameter
- Using Method Parameter
- Page Methods
- JavaScript Object Notation
- JSON Data Types
- Comparing JSON and SOAP
- SOAP Serialization
- ScriptMethod Attribute

## **AJAX CONTROL TOOLKIT**

- AJAX Control Toolkit
- Installing ACT in Visual Studio
- ACT Controls in Visual Studio
- Hello ACT
- Intellisense for Properties
- AjaxControlToolkit.dll
- Registering AjaxControlToolkit.dll
- Extender Controls
- NumericUpDownExtender Control
- Rating Control
- Using a Style Sheet
- MaskedEdit Control
- MaskedEditValidator Control
- PasswordStrength Control
- Page Layout Controls
- Tab Controls
- Accordion Control
- CollapsiblePanel Control
- CollapsiblePanel Markup
- Popup Controls
- Modal Popup Visual Effects
- ACT Controls and Web Services
- DynamicPopulate

## **APPLICATION SERVICES**

- Using ASP.NET Application Services
- Forms Authentication
- Authorization
- Built-in Authentication Service
- ASP.NET Configuration Tool
- Login Page
- Members Page
- Configuration Files
- JavaScript on Login Page
- JavaScript on Home Page
- Profile Properties
- Using ASP.NET Profile Properties
- Using Client-Side Profile Service
- Profile JavaScript Code
- Storing Profile Information

**Duración:**

105 horas

**Incluye:**

- ❖ Manuales importados de cada uno de los tres módulos, diploma de participación, estacionamiento y servicio de cafetería.

**Nota:**

- ❖ El material está en inglés técnico.

**Formas y condiciones de pago:**

El pago debe hacerse de alguna de las siguientes maneras, pagando el total antes de la fecha de inicio del curso:

- ❖ Depósito Banamex cuenta 4923239 Suc. 575 a nombre de Desarrollo y Capacitación en Internet, S. A. de C. V. o transferencia bancaria CLABE 002180057549232394
- ❖ Cheque a nombre de Desarrollo y Capacitación en Internet, S. A. de C. V.