DIPLOMADO DE ESPECIALIZACIÓN PROFESIONAL EN

PROGRAMACIÓN CON MICROSOFT VISUAL STUDIO

MENCIÓN: DESARROLLO WEB

De 6 pm a 9 pm

184 HORAS

Martes y Jueves

CODIT

- *Instructores Certificados*
- *Clases Online en Vivo*
- *Preparación para Certificación*
- *Material Didáctico*
- *Evaluaciones y Certificado*

Dirigido a Certificación Microsoft

RESUMEN DEL DIPLOMADO

Tecnología

Visual Studio

A quien va dirigido

Desarrolladores profesionales con poca experiencia en programación, interesados en desarrollar aplicaciones utilizando HTML5 con JavaScript y CSS3.

Exámenes de Certificación

Dirigido a la certificación:

Microsoft CERTIFIED Solutions Developer Web Applications

Módulos

- ✓ **Moc 40375** HTML5 Application Development Fundamentals
  Duración 24 Horas
- ✓ **Moc 20480** Programming in HTML5 with JavaScript and CSS3
  Duración 40 Horas
- ✓ **Moc 20483** Programming in C#
  Duración 40 Horas
- ✓ **Moc 20486** Developing ASP.NET MVC 5 Web Applications
  Duración 40 Horas
- ✓ **Moc 20487** Developing Microsoft Azure and Web Services
  Duración 40 Horas

Pre-requisitos

- ✓ Mínima experiencia creando aplicaciones web, incluida la escritura de código JavaScript simple.
- ✓ Mínima experiencia creando aplicaciones cliente Windows.
- ✓ Mínima experiencia usando Visual Studio 2017

Objetivos

- ✓ Construir la interfaz de usuario usando HTML5 y formatearla usando CSS.
- ✓ Explicar cómo usar Visual Studio 2017 para crear y ejecutar una aplicación web.
- ✓ Crear y diseñar páginas y formularios con HTML5, que puedan adaptarse a diferentes dispositivos y factores de forma.
- ✓ Crear un código JavaScript
- ✓ Crear aplicaciones web que admitan operaciones sin conexión.
- ✓ Usar Web Sockets para transmitir datos entre una aplicación web y un servidor.
- ✓ Describir la sintaxis central y las características de Visual C #.
- ✓ Implementar la estructura básica y los elementos esenciales de una aplicación de escritorio típica.
- ✓ Integrar bibliotecas no administradas y componentes dinámicos en una aplicación de Visual C #.
- ✓ Diseñe la arquitectura y la implementación de una aplicación web que cumpla con un conjunto de requisitos funcionales, requisitos de interfaz de usuario y aborde los modelos de negocios.
- ✓ Desarrollar una aplicación web que utilice el motor de enrutamiento ASP.NET para presentar URL amigables y una jerarquía de navegación lógica para los usuarios.
- ✓ Describa la plataforma de nube de Microsoft Azure y sus ofertas de cómputo, datos y hospedaje de aplicaciones.
- ✓ Diseñe y desarrolle una aplicación centrada en datos usando Visual Studio 2017 y Entity Framework Core.

www.codit.us

Contenido Programático

Moc 40375 HTML5 Application Development Fundamentals

(24 Horas)

Module 1: Managing the Application Life Cycle

Module 2: Building the User Interface by Using HTML5: Text, Graphics, and Media

Module 3: Building the User Interface by Using HTML5: Organization, Input, and Validation

Module 4: Understanding CSS Essentials: Content Flow, Positioning, and Styling

Module 5: Understanding CSS Essentials: Layouts

Module 6: Managing Text Flow by Using CSS

Module 7: Managing the Graphical Interface by Using CSS

Module 8: Understanding JavaScript and Coding Essentials

Module 9: Creating Animations, Working with Graphics, and Accessing Data

Module 10: JavaScript Coding for the Touch Interface, Device and Operating System Resources, and More

Venezuela: Edificio Phelps frente a la fuente de Plaza Venezuela, Piso N°2, Ofic.1-12 / **Teléfonos: +58-212-750-7190 / 750-7191**
Colombia: Calle 34 B # 65 D – 02, Ofic. 201, Barrio Conquistadores - Medellin / **Teléfonos: +57-034-520-8855**

Moc 20480 Programming in HTML5 with JavaScript and CSS3

(40 Horas)

Module 1: Overview of HTML and CSS

Most modern web applications are built upon a foundation of HTML pages that describe the content that users read and interact with, style sheets to make that content visually pleasing, and JavaScript code to provide a level of interactivity between user and page, and page and server. The web browser uses the HTML markup and the style sheets to render this content, and runs the JavaScript code to implement the behavior of the application. This module reviews the basics of HTML and CSS, and introduces the tools that this course uses to create HTML pages and style sheets.

Lessons

1. Overview of HTML
2. Overview of CSS
3. Creating a Web Application by Using Visual Studio 2017

Lab: Exploring the Contoso Conference Application

4. Exploring the Contoso Conference Application
5. Examining and Modifying the Contoso Conference Application

After completing this module, you will be able to:

- Explain how to use HTML elements and attributes to lay out a web page.
- Explain how to use CSS to apply basic styling to a web page.
- Describe the tools that Microsoft Visual Studio provides for building web applications.

Module 2: Creating and Styling HTML Pages

The technologies forming the basis of all web applications—HTML, CSS, and JavaScript—have been available for many years, but the purpose and sophistication of web applications have changed significantly. HTML5 is the first major revision of HTML in 10 years, and it provides a highly suitable means of presenting content for traditional web applications, applications running on handheld mobile devices, and also on the Windows 10 platform. This module introduces HTML5, describes its new features, demonstrates how to present content by using the new features in HTML5, and how to style this content by using CSS.

Lessons

1. Creating an HTML5 Page
2. Styling an HTML5 Page

Lab: Creating and Styling HTML5 Pages

3. Creating HTML5 Pages
4. Styling HTML pages

After completing this module, students will be able to:

- Describe the purpose of and new features in HTML5, and explain how to use new HTML5 elements to lay out a web page.
- Explain how to use CSS to style the layout, text, and background of a web page

Module 3: Introduction to JavaScript

HTML and CSS provide the structural, semantic, and presentation information for a web page. However, these technologies do not describe how the user interacts with a page by using a browser. To implement this functionality, all modern browsers include a JavaScript engine to support the use of scripts in a page. They also implement Document Object Model (DOM), a W3C standard that defines how a browser should reflect a page in memory to enable scripting engines to access and alter the contents of that page. This module introduces JavaScript programming and DOM.

Lessons

1. Overview of JavaScript
2. Introduction to the Document Object Model

Lab: Displaying Data and Handling Events by Using JavaScript.

3. Displaying Data Programmatically
4. Handling Events

After completing this module, students will be able to:

- Describe basic JavaScript syntax.
- Write JavaScript code that uses the DOM to alter and retrieve info from a web page.

Module 4: Creating Forms to Collect and Validate User Input

Web applications frequently need to gather user input in order to perform their tasks. A web page needs to be clear and concise about the input expected from a user in order to minimize frustrating misunderstandings about the information that the user should provide. Additionally, all input must be validated to ensure that it conforms to the requirements of the application. In this module, you will learn how to define input forms by using the new input types available in HTML5. You will also see how to validate data by using HTML5 attributes. Finally, you will learn how to perform extended input validation by using JavaScript code, and how to provide feedback to users when their input is not valid or does not match the application's expectations.

Lessons

1. Creating HTML5 Forms
2. Validating User Input by Using HTML5 Attributes
3. Validating User Input by Using JavaScript

Lab: Creating a Form and Validating User Input

4. Creating a Form and Validating User Input by Using HTML5 Attributes
5. Validating User Input by Using JavaScript

After completing this module, students will be able to:

- Create input forms by using HTML5.
- Use HTML5 form attributes to validate data.
- Write JavaScript code to perform validation tasks that cannot easily be implemented by using HTML5 attributes.

Module 5: Communicating with a Remote Server

Many web applications require the use of data held by a remote site. In some cases, you can access this data simply by downloading it from a specified URL, but in other cases the data is encapsulated by the remote site and made accessible through a web service. In this module, you will learn how to access a web service by using JavaScript code and to incorporate remote data into your web applications. You will look at two technologies for achieving this: the XMLHttpRequest object, which acts as a programmatic wrapper around HTTP requests to remote web sites, and Fetch API, which simplifies many of the tasks involved in sending requests and receiving data. Because the Fetch API and the XMLHttpRequest object are asynchronous api You will first learn about how to handle asynchronous tasks with the Promise object, arrow functions and the new async/await syntax that lets you handle asynchronous request as if they were synchronous.

Lessons

1. Async programming in JavaScript
2. Sending and Receiving Data by Using the XMLHttpRequest Object
3. Sending and Receiving Data by Using the Fetch API

Lab: Communicating with a Remote Data Source

4. Retrieving Data
5. Serializing and Transmitting Data
6. Refactoring the Code by Using the jQuery ajax Method

After completing this module, students will be able to:

- Handle asynchronous JavaScript tasks using the new async programing technologies.
- Send data to a web service and receive data from a web service by using an XMLHttpRequest object.
- Send data to a web service and receive data from a web service by using the Fetch API.

Module 6: Styling HTML5 by Using CSS3

Styling the content displayed by a web page is an important aspect of making an application attractive and easy to use. CSS is the principal mechanism that web applications use to implement styling, and the features added to CSS3 support many of the new capabilities found in modern browsers. Where CSS1 and CSS2.1 were single documents, the World Wide Web Consortium has chosen to write CSS3 as a set of modules, each focusing on a single aspect of presentation such as color, text, box model, and animations. This allows the specifications to develop incrementally, along with their implementations. Each specification defines properties and values that already exist in CSS1 and CSS2, and also new properties and values. In this module, you will examine the properties and values defined in several of these modules, the new selectors defined in CSS3, and the use of pseudo-classes and pseudo-elements to refine those selections.

Lessons

1. Styling Text by Using CSS3
2. Styling Block Elements
3. Pseudo-Classes and Pseudo-Elements
4. Enhancing Graphical Effects by Using CSS3

Lab: Styling Text and Block Elements by Using CSS3

5. Styling the Navigation Bar
6. Styling the Register Link
7. Styling the About Page

After completing this module, students will be able to:

- Use the new features of CSS3 to style text elements.
- Use the new features of CSS3 to style block elements.
- Use CSS3 selectors, pseudo-classes, and pseudo-elements to refine the styling of elements.
- Enhance pages by using CSS3 graphical effects.

Module 7: Creating Objects and Methods by Using JavaScript

Code reuse and ease of maintenance are key objectives of writing well-structured applications. If you can meet these objectives, you will reduce the costs associated with writing and maintaining your code. This module describes how to write well-structured JavaScript code by using language features such as namespaces, objects, encapsulation, and inheritance. These concepts might seem familiar if you have experience in a language such as Java or C#, but the JavaScript approach is quite different and there are many subtleties that you must understand if you want to write maintainable code.

Lessons

1. Writing Well-Structured JavaScript Code
2. Creating Custom Objects
3. Extending Objects

Lab: Refining Code for Maintainability and Extensibility

4. Object Inheritance
5. Refactoring JavaScript Code to Use Objects

After this module, students will be able to:

- Write well-structured JavaScript code.
- Use JavaScript code to create custom objects.
- Implement object-oriented techniques by using JavaScript idioms.

Module 8: Creating Interactive Pages by Using HTML5 APIs

Interactivity is a key aspect of modern web applications, enabling you to build compelling web sites that can quickly respond to the actions of the user, and also adapt themselves to the user's location. This module describes how to create interactive HTML5 web applications that can access the local file system, enable the user to drag-and-drop data onto elements in a web page, play multimedia files, and obtain geolocation information.

Lessons

1. Interacting with Files
2. Incorporating Multimedia
3. Reacting to Browser Location and Context
4. Debugging and Profiling a Web Application

Lab: Creating Interactive Pages with HTML5 APIs

5. Dragging and Dropping Images
6. Incorporating Video
7. Using the Geolocation API to Report the User's Current Location

After completing this module, students will be able to:

- Access the local file system, and add drag-and-drop support to web pages.
- Play video and audio files in a web page, without the need for plugins.
- Obtain information about the current location of the user.
- Use the F12 Developer Tools in Microsoft Edge to debug and profile a web application.

Module 9: Adding Offline Support to Web Applications

Web applications have a dependency on being able to connect to a network to fetch web pages and data. However, in some environments a network connection may be intermittent. In these situations, it might be useful to enable the application to continue functioning by using data cached on the user's device. HTML5 provides a choice of new client-side storage options, including session storage and local storage, and a resource caching mechanism called the Application Cache. In this module, you will learn how to use these technologies to create robust web applications that can continue running even when a network connection is unavailable.

Lessons

1. Reading and Writing Data Locally
2. Adding Offline Support by Using the Application Cache

Lab: Adding Offline Support to Web Applications

3. Caching Offline Data by Using the Application Cache API
4. Persisting User Data by Using the Local Storage API

After completing this module, students will be able to:

- Save data locally on the user's device, and access this data from a web application.
- Configure a web application to support offline operations by using the Application Cache.

Module 10: Implementing an Adaptive User Interface

One of the most enduring features of the web is its temporary nature. For the first time, the monopoly of the keyboard and mouse is coming under challenge, and that means questioning how user interfaces are designed. You may develop a web application on a computer with a large, high-resolution monitor, a mouse, and a keyboard, but other users might view and interact with your application on a smartphone or a tablet without a mouse, or have a monitor with a different resolution. Users may also want to print pages of your application. In this module, you will learn how to build a website that adapts the layout and functionality of its pages to the capabilities and form factor of the device on which it is being viewed. You will see how to detect the type of device being used to view a page, and learn strategies for laying out content that effectively targets particular devices.

Lessons

1. Supporting Multiple Form Factors
2. Creating an Adaptive User Interface

Lab: Implementing an Adaptive User Interface

3. Creating a Print-Friendly Style Sheet
4. Adapting Page Layout to Fit Different Form Factors

After completing this module, students will be able to:

- Describe the requirements in a website for responding to different form factors.
- Create web pages that can adapt their layout to match the form factor of the device on which they are displayed.

Module 11: Creating Advanced Graphics

High-resolution, interactive graphics are a key part of most modern applications. Graphics can help to enhance the user's experience by providing a visual aspect to the content, making a website more attractive and easier to use. Interactivity enables the graphical elements in a website to adapt and respond to user input or changes to the environment, and is another important element in retaining the attention of the user and their interest in the content. This module describes how to create advanced graphics in HTML5 by using Scalable Vector Graphics (SVG) and the Microsoft Canvas API. You will learn how to use SVG-related elements such as , , and to display graphical content on a web page. You will also learn how to enable the user to interact with SVG elements through the use of events such as keyboard events and mouse events. The Canvas API is somewhat different than SVG.

Lessons

1. Creating Interactive Graphics by Using SVG
2. Drawing Graphics by Using the Canvas API

Lab: Creating Advanced Graphics

3. Creating an Interactive Venue Map by Using SVG
4. Creating a Speaker Badge by Using the Canvas API

After completing this module, students will be able to:

- Use SVG to create interactive graphical content.
- Use the Canvas API to generate graphical content programmatically.

Module 12: Animating the User Interface

Animations are a key element in maintaining the interest of a user in a website. Implemented carefully, animations improve the usability of a web page and provide useful visual feedback on user actions. This module describes how to enhance web pages by using CSS animations. You will learn how to apply transitions to property values. Transitions enable you to specify the timing of property changes. For example, you can specify that an element should change its width and height over a five-second period when the mouse pointer hovers over it. Next, you will learn how to apply 2D and 3D transformations to elements. Transformations enable you to scale, translate, rotate, and skew elements. You can also apply transitions to transformations, so that the transformation is applied gradually over a specified animation period. At the end of this module, you will learn how to apply keyframe animations to elements. Keyframe animations enable you to define a set of property values at specific moments during an animation.

For example, you can specify the color and position of an element at 0 percent, 33 percent, 66 percent, and 100 percent of the animation period.

Lessons

1. Applying CSS Transitions
2. Transforming Elements
3. Applying CSS Keyframe Animations

Lab: Animating the User Interface

4. Applying CSS Transitions
5. Applying Keyframe Animations

After completing this module, students will be able to:

- Apply transitions to animate property values to HTML elements.
- Apply 2D and 3D transformations to HTML elements.
- Apply keyframe animations to HTML elements.

Module 13: Implementing Real-time Communication by Using Web Sockets

Web pages request data on demand from a web server by submitting HTTP requests. This model is ideal for building interactive applications, where the functionality is driven by the actions of a user. However, in an application that needs to display constantly changing information, this mechanism is less suitable. For example, a financial stocks page is worthless if it shows prices that are even a few minutes old, and you cannot expect a user to constantly refresh the page displayed in the browser. This is where web sockets are useful. The Web Sockets API provides a mechanism for implementing real-time, two-way communication between web server and browser. This module introduces web sockets, describes how they work, and explains how to create a web socket connection that can be used to transmit data in real time between a web page and a web server.

Lessons

1. Introduction to Web Sockets
2. Using the WebSocket API

Lab: Performing Real-time Communication by Using Web Sockets

3. Receiving Messages from a Web Socket
4. Sending Messages to a Web Socket
5. Handling Different Web Socket Message Types

After completing this module, students will be able to:

- Describe how using web sockets helps to enable real-time communications between a web page and a web server.
- Use the Web Sockets API to connect to a web server from a web page, and exchange messages between the web page and the web server.

Module 14: Performing Background Processing by Using Web Workers

JavaScript code is a powerful tool for implementing functionality in a web page, but you need to remember that this code runs either when a web page loads or in response to user actions while the web page is being displayed.

The code is run by the browser, and if the code performs operations that take a significant time to complete, the browser can become unresponsive and degrade the user's experience. HTML5 introduces web workers, which enable you to offload processing to separate background threads and thus enable the browser to remain responsive. This module describes how web workers operate and how you can use them in your web applications.

Lessons

1. Understanding Web Workers
2. Performing Asynchronous Processing by Using Web Workers

   Lab: Creating a Web Worker Process
3. Improving Responsiveness by Using a Web Worker

After completing this module, students will be able to:

- Explain how web workers can be used to implement multithreading and improve the responsiveness of a web application.
- Perform processing by using a web worker, communicate with a web worker, and control a web worker.

Module 15: Packaging JavaScript for Production Deployment

Using models allows you to build large, complex applications. The progress of the language in the version of ECMAScript6 allows the build app with to simplify the application construction process. However, the use of ECMAScript6 modules and other features is not yet supported in all browsers. Tools such as Node.js, Webpack, and Babel enable the use of new language features along with support for different browsers in order to avoid harming the user experience. In this module we will introduce the theory behind these tools, when we need to used them, and the different options for use. At the end of the module we will see how to use these tools to write ECMAScript6 code supported in all browsers.

Lessons

1. Understanding Transpilers And Module bundling
2. Creating Separate Packages for Cross Browser Support

   Lab: Setting Up Webpack Bundle for Production
3. Creating and Deploying Packages using WebPack

Venezuela: Edificio Phelps frente a la fuente de Plaza Venezuela, Piso N°2, Ofic.1-12 / **Teléfonos: +58-212-750-7190 / 750-7191**
Colombia: Calle 34 B # 65 D – 02, Ofic. 201, Barrio Conquistadores - Medellin / **Teléfonos: +57-034-520-8855**

Moc 20483 Programming in C#

(40 Horas)

Module 1: Review of Visual C# Syntax

The Microsoft .NET Framework version 4.7 provides a comprehensive development platform that you can use to build, deploy, and manage applications and services. By using the .NET Framework, you can create visually compelling applications, enable seamless communication across technology boundaries, and provide support for a wide range of business processes. In this module, you will learn about some of the core features provided by the .NET Framework and Microsoft Visual Studio. You will also learn about some of the core Visual C# constructs that enable you to start developing .NET Framework applications.

Lessons

1. Overview of Writing Application by Using Visual C#
2. Data Types, Operators, and Expressions
3. Visual C# Programming Language Constructs

Lab: Implementing Edit Functionality for the Students List

4. Implementing Insert Functionality for the Students List
5. Implementing Delete Functionality for the Students List
6. Displaying a Student's Age

After completing this module, students will be able to:

- Describe the architecture of .NET Framework applications and the features that Visual Studio 2017 and Visual C# provide.
- Use basic Visual C# data types, operators, and expressions.
- Use standard Visual C# constructs.

Module 2: Creating Methods, Handling Exceptions, and Monitoring Applications

Applications often consist of logical units of functionality that perform specific functions, such as providing access to data or triggering some logical processing. Visual C# is an object-orientated language and uses the concept of methods to encapsulate logical units of functionality. A method can be as simple or as complex as you like, and therefore it is important to consider what happens to the state of your application when an exception occurs in a method. In this module, you will learn how to create and use methods and how to handle exceptions. You will also learn how to use logging and tracing to record the details of any exceptions that occur.

Lessons

1. Creating and Invoking Methods
2. Creating Overloaded Methods and Using Optional and Output Parameters
3. Handling Exceptions
4. Monitoring Applications

Lab: Extending the Class Enrolment Application Functionality

5. Refactoring the Enrolment Code
6. Validating Student Information
7. Saving Changes to the Class List

After completing this module, students will be able to:

- Create and invoke methods.
- Create overloaded methods and use optional parameters.
- Handle exceptions.
- Monitor applications by using logging, tracing, and profiling

Module 3: Basic types and constructs of Visual C#

To create effective applications by using Windows Presentation Foundation (WPF) or other .NET Framework platforms, you must first learn some basic Visual C# constructs. You need to know how to create simple structures to represent the data items you are working with. You need to know how to organize these structures into collections, so that you can add items, retrieve items, and iterate over your items. Finally, you need to know how to subscribe to events so that you can respond to the actions of your users. In this module, you will learn how to create and use structs and enums, organize data into collections, and create and subscribe to events.

Lessons

1. Implementing Structs and Enums
2. Organizing Data into Collections
3. Handling Events

Lab: Writing the Code for the Grades Prototype Application

4. Adding Navigation Logic to the Grades Prototype Application
5. Creating Data Types to Store User and Grade Information
6. Displaying User and Grade Information

After completing this module, students will be able to:

- Create and use structs and enums.
- Use collection classes to organize data.
- Create and subscribe to events.

Module 4: Creating Classes and Implementing Type-Safe Collections

In this module, you will learn how to use interfaces and classes to define and create your own custom, reusable types. You will also learn how to create and use enumerable, type-safe collections of any type.

Lessons

1. Creating Classes
2. Defining and Implementing Interfaces
3. Implementing Type-Safe Collections

Lab: Adding Data Validation and Type-Safety to the Application

4. Implementing the Teacher, Student, and Grade Structs as Classes
5. Adding Data Validation to the Grade Class
6. Displaying Students in Name Order
7. Enabling Teachers to Modify Class and Grade Data

After completing this module, you will be able to:

- Create and instantiate classes.
- Create and instantiate interfaces.
- Use generics to create type-safe collections.

Module 5: Creating a Class Hierarchy by Using Inheritance

In this module, you will learn how to use inheritance to create class hierarchies and to extend .NET Framework types.

Lessons

1. Creating Class Hierarchies
2. Extending .NET Framework Classes

Lab: Refactoring Common Functionality into the User Class

3. Refactoring Common Functionality into the User Class
4. Implementing Password Complexity by Using an Abstract Method
5. Creating the ClassFullException Custom Exception

After completing this module, you will be able to:

- Create base classes and derived classes by using inheritance.
- Create classes that inherit from .NET Framework classes.

Module 6: Reading and Writing Local Data

In this module, you will learn how to read and write data by using transactional file system I/O operations, how to serialize and deserialize data to the file system, and how to read and write data to the file system by using streams.

Lessons

1. Reading and Writing Files
2. Serializing and Deserializing Data
3. Performing I/O by Using Streams

Lab: Generating the Grades Report

4. Serializing Data for the Grades Report as XML
5. Previewing the Grades Report
6. Persisting the Serialized Grade Data to a File

After completing this module, you will be able to:

- Read and write data to and from the file system by using file I/O.
- Convert data into a format that can be written to or read from a file or other data source.
- Use streams to send and receive data to or from a file or data source.

Module 7: Accessing a Database

In this module, you will learn how to create and use entity data models (EDMs) and how to query many types of data by using Language-Integrated Query (LINQ).

Lessons

1. Creating and Using Entity Data Models
2. Querying Data by Using LINQ

Lab: Retrieving and Modifying Grade Data

3. Creating an Entity Data Model from The School of Fine Arts Database
4. Updating Student and Grade Data by Using the Entity Framework
5. Extending the Entity Data Model to Validate Data

After completing this module, you will be able to:

- Create, use, and customize an EDM.
- Query data by using LINQ.

Module 8: Accessing Remote Data

In this module, you will learn how to use the request and response classes in the System.Net namespace to directly manipulate remote data sources. You will also learn how to use Windows Communication Foundation (WCF) Data Services to expose and consume an entity data model (EDM) over the web.

Lessons

1. Accessing Data Across the Web
2. Accessing Data by Using OData Connected Services

Lab: Retrieving and Modifying Grade Data Remotely

3. Creating a WCF Data Service for the SchoolGrades Database
4. Integrating the Data Service into the Application
5. Retrieving Student Photographs Over the Web (If Time Permits)

After completing this module, you will be able to:

- Send data to and receive data from web services and other remote data sources.
- Access data by using WCF Data Services.

Module 9: Designing the User Interface for a Graphical Application

In this module, you will learn how to use Extensible Application Markup Language (XAML) and Windows Presentation Foundation (WPF) to create engaging UIs.

Lessons

1. Using XAML to Design a User Interface
2. Binding Controls to Data

Lab: Customizing Student Photographs and Styling the Application

3. Customizing the Appearance of Student Photographs
4. Styling the Logon View
5. Animating the StudentPhoto Control (If Time Permits)

After completing this module, you will be able to:

- Use XAML to design a UI.
- Bind a XAML control to data.
- Apply styles to a XAML UI.

Module 10: Improving Application Performance and Responsiveness

In this module, you will learn how to improve the performance of your applications by distributing your operations across multiple threads.

Lessons

1. Implementing Multitasking
2. Performing Operations Asynchronously
3. Synchronizing Concurrent Access to Data

Lab: Improving the Responsiveness and Performance of the Application

4. Ensuring That the UI Remains Responsive When Retrieving Teacher Data
5. Providing Visual Feedback During Long-Running Operations

After completing this module, you will be able to:

- Use the Task Parallel Library to implement multitasking.
- Perform long-running operations without blocking threads.
- Control how multiple threads can access resources concurrently.

Module 11: Integrating with Unmanaged Code

In this module, you will learn how to interoperate unmanaged code in your applications and how to ensure that your code releases any unmanaged resources.

Lessons

1. Creating and Using Dynamic Objects
2. Managing the Lifetime of Objects and Controlling Unmanaged Resources

Lab: Upgrading the Grades Report

3. Generating the Grades Report by Using Word
4. Controlling the Lifetime of Word Objects by Implementing the Dispose Pattern

After completing this module, you will be able to:

- Integrate unmanaged code into a Microsoft Visual C# application by using the Dynamic Language Runtime (DLR).
- Control the lifetime of unmanaged resources and ensure that your application releases resources.

Module 12: Creating Reusable Types and Assemblies

In this module, you will learn how to consume existing assemblies by using reflection and how to add additional metadata to types and type members by using attributes. You will also learn how to generate code at run time by using the Code Document Object Model (CodeDOM) and how to ensure that your assemblies are signed and versioned, and available to other applications, by using the global assembly cache (GAC).

Lessons

1. Examining Object Metadata
2. Creating and Using Custom Attributes
3. Generating Managed Code
4. Versioning, Signing, and Deploying Assemblies

Lab: Specifying the Data to Include in the Grades Report

5. Creating and Applying the IncludeInReport attribute
6. Updating the Report
7. Storing the Grades.Utilities Assembly Centrally (If Time Permits)

After completing this module, you will be able to:

- Use reflection to inspect and execute assemblies.
- Create and consume custom attributes.
- Generate managed code at run time by using CodeDOM.
- Version, sign, and deploy your assemblies to the GAC.

Module 13: Encrypting and Decrypting Data

In this module, you will learn how to implement symmetric and asymmetric encryption and how to use hashes to generate mathematical representations of your data. You will also learn how to create and manage X509 certificates and how to use them in the asymmetric encryption process.

Lessons

1. Implementing Symmetric Encryption
2. Implementing Asymmetric Encryption

Lab: Encrypting and Decrypting the Grades Report

3. Encrypting the Grades Report
4. Encrypting the Grades Report

After completing this module, you will be able to:

- Encrypt data by using symmetric encryption.
- Encrypt data by using asymmetric encryption.

Venezuela: Edificio Phelps frente a la fuente de Plaza Venezuela, Piso N°2, Ofic.1-12 / **Teléfonos: +58-212-750-7190 / 750-7191**
Colombia: Calle 34 B # 65 D – 02, Ofic. 201, Barrio Conquistadores - Medellin / **Teléfonos: +57-034-520-8855**

Moc 20486 Developing ASP.NET MVC 5 Web Applications

(40 Horas)

Module 1: Exploring ASP.NET MVC 5

The goal of this module is to outline to the students the components of the Microsoft Web Technologies stack, which can be used to host a completed web application. Students will also learn about ASP.NET and be introduced to the web forms, web pages, and MVC programming models. Finally they will see an overview of ASP.NET MVC 5, including new features and configuration.

Lessons

1. Overview of Microsoft Web Technologies
2. Overview of ASP.NET
3. Introduction to ASP.NET MVC 5

Lab: Exploring ASP.NET MVC 5

4. Exploring a Photo Sharing Application
5. Exploring a Web Pages Application
6. Exploring a Web Forms Application
7. Exploring an MVC Applicationn

After completing this module, you will be able to:

- Describe the Microsoft Web Technologies stack and select an appropriate technology to use to develop any given application.

Module 2: Designing ASP.NET MVC 5 Web Applications

The goal of this module is to introduce students to the typical design process that architects must complete when they plan an MVC 5 application. At this stage in the design process, MVC 5 has been selected as the most appropriate programming model, but the details of the application, such as the overall architecture, Controllers, Views, Models, and routes to create, have not been fixed. How to plan such details is shown during this module.

Lessons

1. Planning in the Project Design Phase
2. Designing Models, Controllers, and Views

Lab: Designing ASP.NET MVC 5 Web Applications

3. Planning Model Classes
4. Planning Controllers
5. Planning Views
6. Architecting an MVC Web Application

After completing this module, students will be able to:

- Design the architecture and implementation of a web application that will meet a set of functional requirements, user interface requirements, and address business models.

Venezuela: Edificio Phelps frente a la fuente de Plaza Venezuela, Piso N°2, Ofic.1-12 / **Teléfonos: +58-212-750-7190 / 750-7191**
Colombia: Calle 34 B # 65 D – 02, Ofic. 201, Barrio Conquistadores - Medellin / **Teléfonos: +57-034-520-8855**

Module 3: Developing ASP.NET MVC 5 Models

The goal of this module is to enable the students to create Models within an MVC application that implement the business logic necessary to satisfy business requirements. The module also describes how to implement a connection to a database, or alternative data store, using the Entity Framework and LINQ.

Lessons

1. Creating MVC Models
2. Working with Data

Lab: Developing ASP.NET MVC 5 Models

3. Creating an MVC Project and Adding a Model
4. Adding Properties to MVC Models
5. Using Data Annotations in MVC Models
6. Creating a New Microsoft Azure SQL Database

After completing this module, students will be able to:

- Create MVC Models and write code that implements business logic within Model methods, properties, and events.

Module 4: Developing ASP.NET MVC 5 Controllers

The goal of this module is to enable students to add Controllers to MVC applications and to implement actions that respond to user input and other events. The students will learn how Controllers relate to Models and how to implement Controller actions that define the View used to display or edit data. This module also covers how to write action filters that run code before or after multiple actions in the Controller. The students will learn about situations when action filters are useful.

Lessons

1. Writing Controllers and Actions
2. Writing Action Filters

Lab: Developing ASP.NET MVC 5 Controllers

3. Adding an MVC Controller and Writing the Actions
4. Optional—Writing the Action Filters in a Controller

After completing this module, students will be able to:

- Add Controllers to an MVC Application to manage user interaction, update models, and select and return Views.

Module 5: Developing ASP.NET MVC 5 Views

The goal of this module is to describe the role of Views in an MVC web application and enable users to create and code them. The syntax of a Razor View is of critical importance for students to understand because it defines both the layout and the functionality of the data display. HTML Helpers will also be discussed in detail and common Helpers, such as Html.ActionLink() and Html.EditorFor(), will be described. Reusing code by defining Partial Views and Razor Helpers will be discussed as well.

Venezuela: Edificio Phelps frente a la fuente de Plaza Venezuela, Piso N°2, Ofic.1-12 / **Teléfonos: +58-212-750-7190 / 750-7191**
Colombia: Calle 34 B # 65 D – 02, Ofic. 201, Barrio Conquistadores - Medellin / **Teléfonos: +57-034-520-8855**

Lessons

1. Creating Views with Razor Syntax
2. Using HTML Helpers
3. Re-using Code in Views

Lab: Developing ASP.NET MVC 5 Views

4. Adding a View for Photo Display
5. Adding a View for New Photos
6. Creating and Using a Partial View
7. Adding a Home View and Testing the Views

After completing this module, students will be able to:

- Create Views in an MVC application that display and edit data and interact with Models and Controllers

Module 6: Testing and Debugging ASP.NET MVC 5 Web Applications

The goal of this module is to enable students to increase the resilience and quality of an application by locating and correcting code errors, bugs, and other unexpected results. MVC applications are well suited to unit testing techniques and these techniques ensure a high quality of code by systematically testing the functionality of each small component. In addition the debugging tools and exception handling available in Visual Studio will be explained.

Lessons

1. Unit Testing MVC Components
2. Implementing an Exception Handling Strategy

Lab: Testing and Debugging ASP.NET MVC 5 Web Applications

3. Performing Unit Tests
4. Optional – Configuring Exception Handling

After completing this module, students will be able to:

- Run unit tests and debugging tools against a web application in Visual Studio and configure an application for troubleshooting.

Module 7: Structuring ASP.NET MVC 5 Web Applications

The goal of this module is to enable students to structure a web application in such a way that users can rapidly locate the information they need. Two aspects of the design are emphasized: the URLs presented in the browser address bar should be understandable and can be controlled by adding routes to the ASP.NET Routing Engine, and the navigation controls, such as menus and breadcrumb trails, should present the most relevant links to frequently read pages. Search Engine Optimization is important throughout this module.

Lessons

1. Analyzing Information Architecture
2. Configuring Routes
3. Creating a Navigation Structure

Lab: Structuring ASP.NET MVC 5 Web Applications

4. Using the Routing Engine
5. Optional—Building Navigation Controls

After completing this module, students will be able to:

- Develop a web application that uses the ASP.NET routing engine to present friendly URLs and a logical navigation hierarchy to users.

Module 8: Applying Styles to ASP.NET MVC 5 Web Applications

The goal of this module is to explore how students can impose a consistent look and feel to an MVC application and share other common components, such as headers and footers, between all Views. Besides describing CSS styles and template views, the module will discuss how to migrate a look and feel created by a web designer into an MVC application. Techniques for adapting the display of a site for small screens and mobile devices will also be introduced.

Lessons

1. Using Layouts
2. Applying CSS Styles to an MVC Applicaiton
3. Creating an Adaptive User Interface

Lab: Applying Styles to ASP.NET MVC 5 Web Applications

4. Creating and Applying Layouts
5. Applying Styles to an MVC Web Application
6. Optional—Adapting Webpages for Mobile Browsers

After completing this module, students will be able to:

- Implement a consistent look and feel, including corporate branding, across an entire MVC web application.

Module 9: Building Responsive Pages in ASP.NET MVC 5 Web Applications

The goal of this module is to describe to the students how partial page updates and caching can optimize the responsiveness of a web application. Students will see how to make use of AJAX helpers and partial views to update small portions of a page instead of refreshing the entire page. The module also covers the different caches developers can use to store rendered pages and discusses how to configure caching for maximum performance.

Lessons

1. Using AJAX and Partial Page Updates
2. Implementing a Caching Strategy

Lab: Building Responsive Pages in ASP.NET MVC 5 Web Applications

3. Using Partial Page Updates
4. Optional—Configuring the ASP.NET Caches

After completing this module, students will be able to:

- Use partial page updates and caching to reduce the network bandwidth used by an application and accelerate responses to user requests.

Module 10: Using JavaScript and jQuery for Responsive MVC 5 Web Applications

The goal of this module is to teach the students techniques that run code on the browser. This approach can increase the responsiveness of the application because a rendered page can respond to a user action without reloading the entire page from the server.

Students will learn about the jQuery script library and how to use it to call web services and update user interface components.

Lessons

1. Rendering and Executing JavaScript Code
2. Using jQuery and jQueryUI

Lab: Using JavaScript and jQuery for Responsive MVC 5 Web Applications

3. Creating and Animating the Slideshow View
4. Optional—Adding a jQueryUI ProgressBar Widget

After completing this module, students will be able to:

- Write JavaScript code that runs on the client-side and utilizes the jQuery script library to optimize the responsiveness of an MVC web application.

Module 11: Controlling Access to ASP.NET MVC 5 Web Applications

The goal of this module to ensure good security in terms of strong authentication and authorization for access. The lessons describe how to enable anonymous users to create their own user account and gain privileged access to content.

Lessons

1. Implementing Authentication and Authorization
2. Assigning Roles and Membership

Lab: Controlling Access to ASP.NET MVC 5 Web Applications

3. Configuring Authentication and Membership Providers
4. Building the Logon and Register Views
5. Authorizing Access to Resources
6. Optional—Building a Password Reset View

After completing this module, students will be able to:

- Implement a complete membership system in an MVC 5 web application.

Module 12: Building a Resilient ASP.NET MVC 5 Web Application

The goal of this module is to enable the students to build applications that are stable and reliable. Such applications are not vulnerable to common hacking techniques such as cross-site scripting and also store state information such as the contents of a shopping cart and user preferences. This state information is preserved when servers or browsers restart, connections are lost, and other connectivity issues occur.

Lessons

1. Developing Secure Sites
2. State Management

Lab: Building a Resilient ASP.NET MVC 5 Web Application

3. Creating Favorites Controller Actions
4. Implementing Favorites in Views

After completing this module, students will be able to:

- Build an MVC application that resists malicious attacks and persists information about users and preferences.

Module 13: Implementing Web APIs in ASP.NET MVC 5 Web Applications

The goal of the module is to introduce the concept of a Web API to students and to describe how to make an application's core functionality more broadly available for integration into other web and mobile applications. Students will learn about the new Web API feature of MVC 5 and see how to build a RESTful Web API and call it from other applications.

Lessons

1. Developing a Web API
2. Calling a Web API from Mobile and Web Applications

Lab: Implementing Web APIs in ASP.NET MVC 5 Web Applications

3. Adding a Web API to the Photo Sharing Application
4. Using the Web API for a Bing Maps Display

After completing this module, students will be able to:

- Describe what a Web API is and why developers might add a Web API to an application.

Module 14: Handling Requests in ASP.NET MVC 5 Web Applications

The goal of this module is to describe how to write components that intercept requests from browsers before they are received by MVC Controllers. These components include HTTP Modules, HTTP Handlers, and the Web Sockets protocol. The module describes scenarios in which developers use such components and shows how to add them to an MVC application.

Lessons

1. Using HTTP Modules and HTTP Handlers
2. Using Web Sockets

Lab: Handling Requests in ASP.NET MVC 5 Web Applications

3. Creating a SignalR Hub
4. Creating a Photo Chat View

After completing this module, students will be able to:

- Modify the way browser requests are handled by an MVC application.

Module 15: Deploying ASP.NET MVC 5 Web Applications

The goal for this module is to enable students to deploy a completed MVC application to a web server or Microsoft Azure. The module begins by describing testing, staging, and production deployments and the web server environments required for each. It also describes the advantages and disadvantages of using Microsoft Azure to host the application. Students also see all the available deployment options in Visual Studio.

Lessons

1. Deploying a Web Application
2. Deploying an ASP.NET MVC 5 Web Application

Lab: Deploying ASP.NET MVC 5 Web Applications

3. Deploying a Web Application to Microsoft Azure
4. Testing the Completed Application

After completing this module, students will be able to:

- Describe how to package and deploy an ASP.NET MVC 5 web application from a development computer to a web server for staging or production.

Moc 20487 Developing Microsoft Azure and Web Services

(40 Horas)

Module 1: Overview of service and cloud technologies

This module provides an overview of service and cloud technologies using the Microsoft .NET Framework and the Windows Azure cloud.

Lessons

1. Key Components of Distributed Applications
2. Data and Data Access Technologies
3. Service Technologies
4. Cloud Computing
5. Exploring the Blue Yonder Airlines' Travel Companion Application

Lab: Exploring the work environment

After completing this module, you will be able to:

- Describe the key components of distributed applications.
- Describe data and data access technologies.
- Explain service technologies.
- Describe the features and functionalities of cloud computing.
- Describe the architecture and working of the Blue Yonder Airlines Travel Companion application.

Module 2: Querying and Manipulating Data Using Entity Framework

This module Describes the Entity Framework data model, and how to create, read, update, and delete data.

Lessons

1. ADO.NET Overview
2. Creating an Entity Data Model
3. Querying Data
4. Manipulating Data

Lab: Creating a Data Access Layer by Using Entity Framework

After completing this module, students will be able to:

- Explain basic objects in ADO.NET and asynchronous operations.
- Create an Entity Framework data model.
- Query data by using Entity Framework.
- Insert, delete, and update entities by using Entity Framework.

Module 3: Creating and Consuming ASP.NET Web API Services

This module describes HTTP-based services that are developed, hosted, and consumed by using ASP.NET Web API.

Lessons

1. HTTP Services
2. Creating an ASP.NET Web API Service
3. Handling HTTP Requests and Responses
4. Hosting and Consuming ASP.NET Web API Services

Lab: Creating the Travel Reservation ASP.NET Web API Service

After completing this module, students will be able to:

- Design services by using the HTTP protocol.
- Create services by using ASP.NET Web API.
- Use the HttpRequestMessage/HttpResponseMessage classes to control HTTP messages.
- Host and consume ASP.NET Web API services.

Module 4: Extending and Securing ASP.NET Web API Services

This module describes in detail the ASP.NET Web API architecture and how you can extend and secure ASP.NET Web API services.

Lessons

1. The ASP.NET Web API Pipeline
2. Creating OData Services
3. Implementing Security in ASP.NET Web API Services
4. Injecting Dependencies into Controllers

Lab: Extending Travel Companion's ASP.NET Web API Services

After completing this module, students will be able to:

- Extend the ASP.NET Web API request and response pipeline.
- Create OData services using ASP.NET Web API.
- Secure ASP.NET Web API.
- Inject dependencies into ASP.NET Web API controllers.

Module 5: Creating WCF Services

This module introduces Windows Communication Foundation (WCF) and describes how to create, host, and consume a WCF service.

Lessons

1. Advantages of Creating Services with WCF
2. Creating and Implementing a Contract
3. Configuring and Hosting WCF Services
4. Consuming WCF Services

Lab: Creating and Consuming the WCF Booking Service

After completing this module, students will be able to:

- Describe why and when to use WCF to create services.
- Define a service contract and implement it.
- Host and configure a WCF service.
- Consume a WCF service from a client application.

Module 6: Hosting Services

This module describes how to host web services both on-premises and in Windows Azure. It explains various components of Windows Azure Cloud Services: Web Role, Worker Role, and Windows Azure Web Sites.

Lessons

1. Hosting Services On-Premises
2. Hosting Services in Windows Azure

Lab: Hosting Services

After completing this module, students will be able to:

- Host services on-premises by using Windows services and IIS
- Host services in the Windows Azure cloud environment by using Windows Azure Cloud Services and Web Sites

Module 7: Windows Azure Service Bus

This module describes web-scale messaging patterns, and the infrastructures provided by Windows Azure Service Bus.

Lessons

1. Windows Azure Service Bus Relays
2. Windows Azure Service Bus Queues
3. Windows Azure Service Bus Topics

Lab: Windows Azure Service Bus

After completing this module, students will be able to:

- Describe the purpose and functionality of relayed and buffered messaging.
- Provision, configure, and use the service bus queues.
- Enhance the effectiveness of queue-based communications using topics, subscriptions and filters.

Module 8: Deploying Services

This module describes different techniques for deploying web applications.

Lessons

1. Web Deployment with Visual Studio 2012
2. Creating and Deploying Web Application Packages
3. Command-Line Tools for Web Deploy
4. Deploying Web and Service Applications to Windows Azure
5. Continuous Delivery with TFS and Git
6. Best Practices for Production Deployment

Lab: Deploying Services

After completing this module, students will be able to:

- Deploy web applications with Visual Studio.
- Create and deploy web applications by using IIS Manager.
- Deploy web applications by using the command line.
- Deploy web applications to Windows Azure environments.
- Use continuous delivery with TFS and Git.
- Apply best practices for deploying web applications on-premises and to Windows Azure.

Module 9: Windows Azure Storage

This module Describes Windows Azure Storage, the services it provides, and the best way to use these services.

Lessons

1. Introduction to Windows Azure Storage
2. Windows Azure Blob Storage
3. Windows Azure Table Storage

4. Windows Azure Queue Storage
5. Restricting Access to Windows Azure Storage

Lab: Windows Azure Storage

After completing this module, students will be able to:

- Describe the architecture of Windows Azure Storage.
- Implement Blob Storage in your applications.
- Use Table Storage in your applications
- Describe how to use Windows Azure Queues as a communication mechanism between different parts of your application
- Control access to your storage items.

Module 10: Monitoring and Diagnostics

This module describes how to perform monitoring and diagnostics in Windows Azure services.

Lessons

1. Performing Diagnostics by Using Tracing
2. Configuring Service Diagnostics
3. Monitoring Services Using Windows Azure Diagnostics
4. Collecting Windows Azure Metrics

Lab: Monitoring and Diagnostics

After completing this module, students will be able to:

- Perform tracing in the .NET Framework with the System.Diagnostics namespace.
- Configure and explore web service and IIS tracing.
- Monitor services by using Windows Azure Diagnostics.
- View and collect Windows Azure metrics in the management portal.

Module 11: Identity Management and Access Control

This module describes the basic principles of modern identity handling and demonstrates how to use infrastructures such as Windows Azure Access Control Service (ACS) to implement authentication and authorization with claims-based identity in Windows Communication Foundation (WCF).

Lessons

1. Claims-based Identity Concepts
2. Using the Windows Azure Access Control Service
3. Configuring Services to Use Federated Identities

Lab: Identity Management and Access Control

After completing this module, students will be able to:

- Describe the basic principles of claims-based identity.
- Create a Security Token Service (STS) by using Windows Azure ACS.
- Configure WCF to use federated identity.

Module 12: Scaling Services

This module describes the ways in which you can ensure services can handle increasing workloads and user demand.

Lessons

1. Introduction to Scalability
2. Load Balancing
3. Scaling On-Premises Services with Distributed Cache
4. Windows Azure Caching
5. Scaling Globally

Lab: Scalability

After completing this module, students will be able to:

- Explain the need for scalability.
- Describe how to use load balancing for scaling services.
- Describe how to use distributed caching for on-premises as well as Windows Azure services.
- Describe how to use Windows Azure caching.
- Describe how to scale services globally.

Module 13: Appendix A: Designing and Extending WCF Services

This module covers designing Windows Communication Foundation (WCF) service contracts, creating services that support distributed transactions, and extending the WCF pipeline with custom runtime components and custom behaviors.

Lessons

1. Applying Design Principles to Service Contracts
2. Handling Distributed Transactions
3. Extending the WCF Pipeline

Lab: Designing and Extending WCF Services

After completing this module, students will be able to:

- Design and create services and clients to use different kinds of message patterns.
- Configure a service to support distributed transactions.
- Extend the WCF pipeline with runtime components, custom behaviors, and extensible objects.

Module 14: Appendix B: Implementing Security in WCF Services

This module deals with the various considerations you have to take into account when designing a secure web service, such as encryption, input validation, authentication, and authorization, and the techniques to use while applying these considerations to services developed with WCF.

Lessons

1. Introduction to Web Services Security
2. Transport Security
3. Message Security
4. Configuring Service Authentication and Authorization

Lab: Securing a WCF Service

After completing this module, students will be able to:

- Describe web application security.
- Configure a service for transport security.
- Configure a service for message security.
- Implement and configure authentication and authorization logic.

RESUMEN DEL DIPLOMADO

Información de Contacto

María Isabel Salas

CCO

E-mail: maria.salas@codit.us

Teléfonos:

+58-212-750-7190

+58-212-750-7191

+57-320-901-0878

Tipos de Pago

Financiado:

- ✓ **Inscripción:** 40%
- ✓ **Mensualmente:** 10% (6 meses)

De contado:

- ✓ **10% de descuento**

Formas de Pago: Transferencias

Pagos en DÓLARES

Banco Intermediario: *CITIBANK* de Nueva York

Código ABA: 021000089

SWIFT: CITIUS33

Cuenta: **N° 36006658** (*BANCOLOMBIA*)

Pagos en PESOS COLOMBIANOS (COP)

Banco *BANCOLOMBIA*

Cuenta de Ahorro: **N° 54080298971**

A nombre de: *Soluciones CODIT SAS*

NIT 901100997

OBSERVACIONES

1. Capacitación orientada a mayores de 18 años de edad.
2. Para la inscripción se requiere:
   a. Documento de Identidad
   b. Planilla de Datos
   c. Soporte de Pago (Particulares) O Carta de Compromiso de Pago (Empresas)
3. Al inscribirse, lo participantes recibirán el siguiente material:
   a. Cuaderno
   b. Bolígrafo
   c. Material de estudio digital
4. Una vez completados todos los módulos del diplomado y de haber aprobado satisfactoriamente todas las evaluaciones, los participantes recibirán:
   a. Certificado de participación
5. Las clases serán impartidas en la modalidad on line.
6. En el caso de los pagos a crédito: todos los participantes (empresas o particulares) deberán cancelar puntualmente los primeros cinco (5) días del mes a cursar, de lo contrario no podrán acceder a las aulas virtuales.
7. Son cien (100) cupos disponibles a nivel internacional.
8. No se hacen reembolsos de ningún tipo.