



Programación orientada a objetos con Visual C# (2015) y ADO.NET 4.6

Autor: Manuel Torres Remon

© Derechos de autor registrados:

Empresa Editora Macro EIRL

© Derechos de edición, arte gráfico y diagramación reservados:

Empresa Editora Macro EIRL

Coordinación de edición:

Magaly Ramon Quiroz

Diseño de portada:

Rudy Herrera Torres

Corrección de estilo:

Sonia Obregon Dionisio

Diagramación:

Julissa Ventocilla Fernández

Edición a cargo de:

© Empresa Editora Macro EIRL

Av. Paseo de la República N.º 5613, Miraflores, Lima, Perú

☎ Teléfono: (511) 748 0560

✉ E-mail: proyectoeditorial@editorialmacro.com

🌐 Página web: www.editorialmacro.com

Primera edición: abril 2016

Tiraje: 1500 ejemplares

Impresión

Talleres gráficos de la Empresa Editora Macro EIRL

Jr. San Agustín N.º 612-624, Surquillo, Lima, Perú

ISBN N.º 978-612-304-372-8

Hecho el depósito legal en la Biblioteca Nacional del Perú N.º 2016-05066

Prohibida la reproducción parcial o total, por cualquier medio o método, de este libro sin previa autorización de la Empresa Editora Macro EIRL.

Índice

Introducción	13
CAPÍTULO 1: Programación orientada a objetos	15
1.1 Introducción a la programación orientada a objetos	17
1.2 Características de la programación orientada a objetos	18
1.3 La clase	21
1.4 Nivel de visibilidad	25
1.5 Atributos de clase	26
1.6 Métodos definidos por el usuario	27
<i>Caso desarrollado 1: Visibilidad pública - Control de promedio de notas</i>	<i>28</i>
1.7 Métodos get y set	32
<i>Caso desarrollado 2: Métodos get y set - Venta de productos</i>	<i>35</i>
1.8 Métodos de clase	41
<i>Caso desarrollado 3: Método de clase - Control de pago de empleados</i>	<i>42</i>
1.9 Método constructor	46
1.10 Los objetos	49
1.10.1 Enviando información a los atributos de la clase por medio del objeto	50
1.10.2 Usando los valores de clase	51
<i>Caso desarrollado 4: Método constructor - Control de facturas de vendedores</i>	<i>54</i>
1.11 Herencia de clases	58
<i>Caso desarrollado 5: Herencia - Venta de productos al contado y crédito</i>	<i>59</i>
1.12 Métodos polifórmicos	71
1.12.1 Modificadores del polimorfismo	71
<i>Caso desarrollado 6: Polimorfismo - Control de evaluaciones</i>	<i>72</i>
CAPÍTULO 2: Colecciones	79
2.1 Las colecciones en .Net Framework	81
2.2 Estructura foreach	81
2.3 Clases de colecciones	84
2.3.1 Espacio de nombre System.Collection	84
2.3.2 Espacio de nombre System.Collections.Specialized	84
2.4 Clase ArrayList	85
2.5 Clase List<>	88
2.6 Clase Stack	91

2.7	Clase Queue.....	93
	<i>Caso desarrollado 1: ArrayList - Control de registro de personal</i>	96
	<i>Caso desarrollado 2: List - Control de registro de productos</i>	102
	<i>Caso desarrollado 3: Stack - Control de registro de alumnos</i>	109
	<i>Caso desarrollado 4: Queue - Control de registro de libros.....</i>	114
CAPÍTULO 3: Serialización		123
3.1	Serialización.....	125
3.2	Ventajas de la serialización	126
3.3	Clase SaveFileDialog.....	126
3.4	Clase OpenFileDialog	128
3.5	Clase FileStream.....	129
	<i>Caso desarrollado 1: Serialización - Manejo de cadena</i>	130
	<i>Caso desarrollado 2: Serialización de datos - Registro de cliente.....</i>	133
	<i>Caso desarrollado 3: Serialización de datos e imagen - Registro de cliente con imagen...</i>	136
3.6	Clase XMLSerializer	141
	<i>Caso desarrollado 4: Serialización XML - Registro de producto.....</i>	142
	<i>Caso desarrollado 5: Serialización XML - Registro de productos usando arreglos</i>	145
	<i>Caso desarrollado 6: Serialización XML - Obtener registros de un archivo XML</i>	150
CAPÍTULO 4: ADO.NET 4.6		155
4.1	Script de la base de datos de origen	157
4.2	Diagrama de base de datos Contrato	169
4.3	Introducción al ADO.NET 4.6	170
4.4	Novedades de ADO.NET 4.6.....	171
4.5	Arquitectura de ADO.NET	172
4.6	Componentes de ADO.NET	173
4.7	Proveedores de datos	173
4.8	Objetos principales de los proveedores de datos	174
4.9	Espacio de nombre System.Data.SqlClient	175
4.10	Espacio de nombre System.Data	176
4.11	Clase SqlConnection	176
4.12	Definición de la cadena de conexión	179
4.13	Clase SqlDataAdapter	180
4.14	ConfigurationManager.....	184
4.15	Clase DataSet	186
4.16	Clase DataTable	188
4.17	Clase DataView.....	190
4.18	Instrucción Using.....	192
4.19	Clase SqlCommand.....	193

4.20 Casos desarrollados	196
<i>Caso desarrollado 1: Uso de asistente - Listado de clientes</i>	196
<i>Caso desarrollado 2: Cadena de conexión directa - Listado de clientes</i>	204
<i>Caso desarrollado 3: Usando clase - Listado de clientes</i>	207
<i>Caso desarrollado 4: Manejo de dataset - Listado de contratistas</i>	211
<i>Caso desarrollado 5: Manejo de datatable - Listado de fichas de devolución</i>	215
<i>Caso desarrollado 6: Manejo de dataview - Listado de equipos</i>	219
<i>Caso desarrollado 7: Listado de fichas de devolución por contratista</i>	222
<i>Caso desarrollado 8: Listado de equipos por estado y tipo</i>	227
<i>Caso desarrollado 9: Filtrado de contratos por cliente</i>	232
<i>Caso desarrollado 10: Búsqueda de datos de cliente</i>	236
<i>Caso desarrollado 11: Búsqueda de fichas de reclamo por año</i>	241
<i>Caso desarrollado 12: Mantenimiento de registros de equipo</i>	245
<i>Caso desarrollado 13: Mantenimiento de registro de contratista</i>	256
CAPÍTULO 5: TableAdapter - LinQ to SQL	273
5.1 TableAdapter	275
<i>Caso desarrollado 1: Usando tableadapter y consulta simple - Listado de contratistas</i>	276
<i>Caso desarrollado 2: Usando tableadapter y procedimiento almacenado - Listado de equipos</i>	285
<i>Caso desarrollado 3: Usando tableadapter y procedimiento almacenado - Listado de equipos por estado y tipo</i>	293
5.2 LinQ.....	298
5.3 Implementación de una consulta con LINQ	300
5.4 Operaciones básicas de una consulta LinQ to SQL.....	300
5.5 Clase DataContext	302
<i>Caso desarrollado 4: LinQ to SQL - Listado de contratista</i>	303
<i>Caso desarrollado 5: LinQ to SQL con procedimientos almacenados - Listado de equipos</i>	306
<i>Caso desarrollado 6: LinQ to SQL - Listado de fichas de reclamo por cliente</i>	308
<i>Caso desarrollado 7: LinQ to SQL - Listado de fichas de reclamo por año</i>	311
<i>Caso desarrollado 8: LinQ to SQL - Registro del nuevo equipo</i>	314
CAPÍTULO 6: Transacciones - Ado Entity	319
6.1 Transacciones	321
6.2 Clase SqlTransaction	321
6.3 Método beginTransaction.....	322
<i>Caso desarrollado 1: Mantenimiento de fichas de devolución</i>	323
6.4 Ado Entity	333
<i>Caso desarrollado 2: Ado Entity - Listado de equipo</i>	334
<i>Caso desarrollado 3: Ado Entity - Listado de equipo por estado y tipo</i>	338
<i>Caso desarrollado 4: Ado Entity - Mantenimiento de equipos</i>	341

CAPÍTULO 7: Reportes	347
7.1 Reportes con Crystal	349
7.2 Instalación de Crystal Report para Visual Studio 2015	349
7.3 Diseño de un informe con Crystal Report.....	351
7.4 Secciones del reporte	355
7.5 Modificar el diseño del reporte	356
<i>Caso desarrollado 1: Crystal Básico - Reporte de contratos</i>	<i>357</i>
<i>Caso desarrollado 2: Crystal Report - Reporte de contratos por cliente</i>	<i>361</i>
<i>Caso desarrollado 3: Crystal Report - Reporte de contratos por rango de años.....</i>	<i>366</i>
 CAPÍTULO 8: Servicios WCF	 373
8.1 WCF.....	375
8.2 Características del WCF	375
8.3 Implementación de un servicio WCF	378
<i>Caso desarrollado 1: WCF - Listado de contratistas</i>	<i>384</i>
<i>Caso desarrollado 2: WCF - Listado de fichas de devolución por contratista</i>	<i>388</i>
<i>Caso desarrollado 3: WCF - Mantenimiento de equipos.....</i>	<i>394</i>
 CAPÍTULO 9: Programación en N-Capas	 407
9.1 Arquitectura en capas.....	409
9.2 Aplicaciones distribuidas	410
9.3 Creación de una solución en N-Capas.....	412
<i>Caso desarrollado 1: Mantenimiento de contratistas en N-Capas</i>	<i>418</i>
<i>Caso desarrollado 2: Mantenimiento de clientes con imágenes en N-Capas</i>	<i>441</i>
<i>Caso desarrollado 3: Reporte de equipos con Crystal Report en N-Capas.....</i>	<i>476</i>
<i>Caso desarrollado 4: Reporte de equipos por estado y tipo con Crystal Report en N-Capas</i>	<i>482</i>
<i>Caso desarrollado 5: Reporte total de contratos por años con gráfico en N-Capas.....</i>	<i>490</i>
 Bibliografía.....	 495

1

Capítulo

Programación orientada a objetos

Capacidad terminal:

Implementar aplicaciones de plataforma que permitan el uso de clase, objetos y todos los elementos que componen a una clase.

Contenido

- Introducción a la programación orientada a objetos
- Características de la programación orientada a objetos
- La clase
- Nivel de visibilidad
- Atributos de clase
- Métodos definidos por el usuario
Caso desarrollado 1: Visibilidad pública - Control de promedio de notas
- Métodos get y set
Caso desarrollado 2: Métodos get y set - Venta de productos
- Métodos de clase
Caso desarrollado 3: Métodos de clase - Control de pago de empleados
- Método constructor
Caso desarrollado 4: Método constructor - Control de facturas de vendedores
- Herencia de clases
Caso desarrollado 5: Herencia - Venta de productos al contado y crédito
- Métodos polifórmicos
Caso desarrollado 6: Polimorfismo - Control de evaluaciones

1.1 Introducción a la programación orientada a objetos

La programación orientada a objetos es conocida como un modelo de programación. Esto lleva a pensar que no representa a un lenguaje específico, sino a una forma de implementar los procesos dentro de una organización.

La característica de la programación orientada a objetos es tener la misión de llevar elementos del mundo real a un nivel de código que represente lo mismo. Así, por ejemplo, los empleados de una empresa por características comunes tienen un código que los identifica: nombres y apellidos, categoría de empleado, entre otras propias del empleado.



Analizando un poco más allá de los empleados, se puede encontrar que ellos se distinguen por las distintas labores que realizan o los horarios de trabajo. Es así que se comenzará a pasar los elementos del mundo real en un código orientado a objetos. Sin embargo, resulta importante especificar que existen diferencias entre las palabras «programación orientada a objetos» y un «lenguaje de programación orientado a objetos».

Mientras que la programación orientada a objetos es reconocida como un modelo de programación que convierte entidades del mundo real en objetos, el lenguaje orientado a objetos es considerado como un lenguaje de programación que permite desarrollar aplicaciones orientadas a objetos. En virtud de este contraste, en estos primeros capítulos buscará usar correctamente la metodología de la programación orientada a objetos, tomando como entorno de trabajo a Visual Studio 2015.

Este tipo de programación surge como una alternativa moderna para el desarrollo de aplicaciones, muy distinta a los usos anteriores de una programación estructurada o de una programación modular que permitía descomponer un código completo.

Por otro lado, se trata de una programación que define una serie de conceptos y técnicas que permiten representar eventos o entidades del mundo real basados en objetos —ahora vistos en clases, objetos, métodos, propiedades, herencia, encapsulación y polimorfismo— generando así una aplicación duradera en el tiempo y sujeta a cambios que no perjudican el accionar de la misma.