

TRABAJO DE DIPLOMA CÓDIGO DE MATERIA 29

Titular: Dario Guillermo Cardacci

FACULTAD DE TECNOLOGÍA INFORMÁTICA

UNIVERSIDAD ABIERTA INTERAMERICANA

2017

ÍNDICE

GUÍA DE TRABAJOS REVISIÓN CONCEPTUAL	3
GUÍA DE TRABAJOS PRÁCTICOS	13
GUÍA DE ABORDAJE BIBLIOGRÁFICO.....	18

GUÍA DE TRABAJOS REVISIÓN CONCEPTUAL

UNIDAD I

1. ¿Por qué decimos que el software es complejo?
2. Enumere y explique los factores que hacen que el software sea complejo
3. ¿Cómo gestionamos y administramos la complejidad?
4. ¿Cuáles son los aspectos más relevantes a tener en cuenta cuando deseamos obtener beneficios económicos con el desarrollo de software?
5. ¿Qué significa migrar un software?
6. ¿Qué diferencia existe entre migrar y actualizar un software?
7. ¿Qué aspectos hay que considerar al migrar un software?
8. ¿Qué beneficios e inconvenientes trae aparejado migrar un software?
9. ¿Enumere los principales elementos que tendría en cuenta al plantear en ciclo de desarrollo de un proyecto de software?
10. ¿Qué rol juega el recurso humano en el desarrollo de un proyecto de software?
11. ¿Cuáles son las cosas que consideraría si debe evaluar entre actualizar o comprar un nuevo software?
12. ¿Cuáles son los aspectos distintivos que Ud. Consideraría si debe decidir entre comprar o desarrollar un software?
13. ¿Qué rol juegan los factores organizacionales en las decisiones de compra o desarrollo de software?
14. ¿Qué cosas no haría en el caso de adquirir software?
15. ¿Qué cosas haría en el caso de adquirir software?
16. ¿Qué cosas no haría en el caso de desarrollar software?
17. ¿Qué cosas haría en el caso de desarrollar software?
18. Si la situación actual de su organización le permite decidir libremente y sin restricciones entre comprar un producto o desarrollarlo ¿Qué elegiría y porqué?

19. ¿Qué otros factores tendría en consideración además del humano en la adquisición de software?
20. ¿Qué otros factores tendría en consideración además del humano en el desarrollo de software?

UNIDAD II

21. ¿Cuáles son los elementos que debería tener en cuenta para desarrollar un módulo de seguridad en un sistema de información?
22. ¿Qué elementos controlaría al hacer log-in en un sistema?
23. ¿Qué procesos relacionados con el módulo de seguridad pondría a funcionar si el log-in es correcto?
24. ¿Qué efectos causa en el sistema hacer un log-out?
25. ¿A qué denominamos perfil de usuario en un sistema?
26. ¿Qué es un rol de usuario?
27. ¿A qué llamamos bitácora del sistema?
28. ¿Para qué se puede utilizar la bitácora del sistema?
29. ¿Qué elementos debería tener la bitácora del sistema?
30. ¿A qué nos referimos cuando planteamos que se debe poder parametrizar el sistema?
31. ¿Qué es la criptografía?
32. ¿Qué datos de un sistema se deben encriptar?
33. ¿Qué algoritmos de encriptación son los más populares?
34. ¿A qué nos referimos cuando hablamos de la internacionalización de un sistema?
35. ¿Qué es la cultura de un sistema de información?
36. ¿Qué aspectos debo considerar para que mi sistema acepte trabajar con múltiples idiomas simultáneamente?
37. ¿Qué es una política de backup?
38. ¿Qué tipos de backup existen si consideramos que información se resguarda?

- 39. ¿Qué se debe considerar para realizar una gestión de backup eficiente?
- 40. ¿A qué nos referimos cuando hablamos de la ejecución de código seguro en la aplicación?

UNIDAD III

- 41. ¿Qué es un patrón en el contexto de desarrollo de software?
- 42. ¿Cómo nace el concepto de patrón?
- 43. ¿Qué es un patrón de diseño?
- 44. ¿De qué manera un patrón de diseño brinda una solución a un problema?
- 45. ¿Qué debería tener en cuenta para seleccionar un patrón de diseño?
- 46. ¿Cuáles son los pasos que seguiría para usar correctamente un patrón?
- 47. ¿Cuáles son los cuatro elementos iniciales de todo patrón?
- 48. ¿Qué elementos utilizaría en la descripción de un patrón?
- 49. ¿Cómo cataloga Gamma a los patrones de diseño?
- 50. ¿A qué denominamos anti patrón?
- 51. ¿Por qué en la utilización de patrones se alienta el uso de herencia de interfaces?
- 52. ¿En qué casos preferiría componer a heredar?
- 53. ¿A qué hace referencia el concepto de diseñar para el cambio?
- 54. ¿Cuáles son los patrones de diseño considerados de creación?
- 55. ¿Cuál es el propósito del patrón Builder?
- 56. ¿Se conoce con algún otro nombre al patrón Builder?
- 57. ¿En qué casos es aplicable el patrón se conoce Builder?
- 58. ¿Cuáles son las clases participantes en el patrón Builder?
- 59. ¿Posee colaboradores el patrón Builder?
- 60. ¿Qué consecuencias trae aparejadas utilizar el patrón Builder?
- 61. ¿Con qué patrones de relaciona el patrón Builder?

62. ¿Cuál es el propósito del patrón Prototype?
63. ¿Se conoce con algún otro nombre al patrón Prototype?
64. ¿En qué casos es aplicable el patrón se conoce Prototype?
65. ¿Cuáles son las clases participantes en el patrón Prototype?
66. ¿Posee colaboradores el patrón Prototype?
67. ¿Qué consecuencias trae aparejadas utilizar el patrón Prototype?
68. ¿Con qué patrones de relaciona el patrón Prototype?
69. ¿Cuál es el propósito del patrón Bridge?
70. ¿Se conoce con algún otro nombre al patrón Bridge?
71. ¿En qué casos es aplicable el patrón se conoce Bridge?
72. ¿Cuáles son las clases participantes en el patrón Bridge?
73. ¿Posee colaboradores el patrón Bridge?
74. ¿Qué consecuencias trae aparejadas utilizar el patrón Bridge?
75. ¿Con qué patrones de relaciona el patrón Bridge?
76. ¿Cuál es el propósito del patrón Facade?
77. ¿Se conoce con algún otro nombre al patrón Facade?
78. ¿En qué casos es aplicable el patrón se conoce Facade?
79. ¿Cuáles son las clases participantes en el patrón Facade?
80. ¿Posee colaboradores el patrón Facade?
81. ¿Qué consecuencias trae aparejadas utilizar el patrón Facade?
82. ¿Con qué patrones de relaciona el patrón Facade?
83. ¿Cuál es el propósito del patrón Flyweight?
84. ¿Se conoce con algún otro nombre al patrón Flyweight?
85. ¿En qué casos es aplicable el patrón se conoce Flyweight?

86. ¿Cuáles son las clases participantes en el patrón Flyweight?
87. ¿Posee colaboradores el patrón Flyweight?
88. ¿Qué consecuencias trae aparejadas utilizar el patrón Flyweight?
89. ¿Con qué patrones de relaciona el patrón Flyweight?
90. ¿Cuál es el propósito del patrón Proxy?
91. ¿Se conoce con algún otro nombre al patrón Proxy?
92. ¿En qué casos es aplicable el patrón se conoce Proxy?
93. ¿Cuáles son las clases participantes en el patrón Proxy?
94. ¿Posee colaboradores el patrón Proxy?
95. ¿Qué consecuencias trae aparejadas utilizar el patrón Proxy?
96. ¿Con qué patrones de relaciona el patrón Proxy?
97. ¿Cuál es el propósito del patrón Chain of Responsibility?
98. ¿Se conoce con algún otro nombre al patrón Chain of Responsibility?
99. ¿En qué casos es aplicable el patrón se conoce Chain of Responsibility?
100. ¿Cuáles son las clases participantes en el patrón Chain of Responsibility?
101. ¿Posee colaboradores el patrón Chain of Responsibility?
102. ¿Qué consecuencias trae aparejadas utilizar el patrón Chain of Responsibility?
103. ¿Con qué patrones de relaciona el patrón Chain of Responsibility?
104. ¿Cuál es el propósito del patrón Command?
105. ¿Se conoce con algún otro nombre al patrón Command?
106. ¿En qué casos es aplicable el patrón se conoce Command?
107. ¿Cuáles son las clases participantes en el patrón Command?
108. ¿Posee colaboradores el patrón Command?
109. ¿Qué consecuencias trae aparejadas utilizar el patrón Command?

- 110. ¿Con qué patrones de relaciona el patrón Command?
- 111. ¿Cuál es el propósito del patrón State?
- 112. ¿Se conoce con algún otro nombre al patrón State?
- 113. ¿En qué casos es aplicable el patrón se conoce State?
- 114. ¿Cuáles son las clases participantes en el patrón State?
- 115. ¿Posee colaboradores el patrón State?
- 116. ¿Qué consecuencias trae aparejadas utilizar el patrón State?
- 117. ¿Con qué patrones de relaciona el patrón State?
- 118. ¿Cuál es el propósito del patrón Strategy?
- 119. ¿Se conoce con algún otro nombre al patrón Strategy?
- 120. ¿En qué casos es aplicable el patrón se conoce Strategy?
- 121. ¿Cuáles son las clases participantes en el patrón Strategy?
- 122. ¿Posee colaboradores el patrón Strategy?
- 123. ¿Qué consecuencias trae aparejadas utilizar el patrón Strategy?
- 124. ¿Con qué patrones de relaciona el patrón Strategy?
- 125. ¿Cuál es el propósito del patrón Template Method?
- 126. ¿Se conoce con algún otro nombre al patrón Template Method?
- 127. ¿En qué casos es aplicable el patrón se conoce Template Method?
- 128. ¿Cuáles son las clases participantes en el patrón Template Method?
- 129. ¿Posee colaboradores el patrón Template Method?
- 130. ¿Qué consecuencias trae aparejadas utilizar el patrón Template Method?
- 131. ¿Con qué patrones de relaciona el patrón Template Method?
- 132. ¿Cuál es el propósito del patrón Visitor?
- 133. ¿Se conoce con algún otro nombre al patrón Visitor?

- 134. ¿En qué casos es aplicable el patrón se conoce Visitor?
- 135. ¿Cuáles son las clases participantes en el patrón Visitor?
- 136. ¿Posee colaboradores el patrón Visitor?
- 137. ¿Qué consecuencias trae aparejadas utilizar el patrón Visitor?
- 138. ¿Con qué patrones de relaciona el patrón Visitor?

UNIDAD IV

- 139. Defina serialización.
- 140. ¿Qué namespaces intervienen en la serialización?
- 141. ¿Para qué se utilizan los objetos formatter?
- 142. ¿Qué características posee la serialización binaria?
- 143. ¿Qué características posee la serialización SOAP?
- 144. ¿Para qué se utilizan los atributos <Serializable()> y <NonSerializable()>?
- 145. ¿Por regla general que cosas no deben serializarse?
- 146. ¿Para qué sirve el objeto ObjectIdGenerator?
- 147. ¿Para qué sirve el objeto ObjectManager?
- 148. ¿A qué se denomina gráfico de objetos?
- 149. ¿Qué ocurre con el proceso de serialización cuando en un gráfico de objetos existe referencias circulares?
- 150. ¿Qué es la serialización personalizada?
- 151. ¿En qué casos se debe utilizar la serialización personalizada?
- 152. ¿Qué interfaz se debe implementar para realizar una serialización personalizada?
- 153. ¿Qué elementos posee la interfaz ISerializable?
- 154. ¿Para qué se utiliza la estructura StreamingContext?
- 155. ¿Para qué se utiliza la interfaz IDeserializationCallback?

156. ¿Es lo mismo hablar de serialización XML y serialización SOAP?
157. ¿En qué se diferencia la serialización XML y serialización SOAP?
158. ¿Qué función cumple la clase XmlSerializer?
159. ¿En qué namespace se encuentra la clase XmlSerializar?
160. ¿Cuáles son los atributos de serialización utilizados en una serialización XML?
161. ¿Para qué se utiliza el atributo XmlRoot?
162. ¿Para qué se utiliza el atributo XmlElement?
163. ¿Para qué se utiliza el atributo XmlAttributeAttribute?
164. ¿Para qué se utiliza el atributo XmlText?
165. ¿Para qué se utiliza el atributo XmlIgnore?
166. ¿Para qué se utiliza el objeto XmlSerializerNamespaces?
167. ¿Cuáles son los sucesos de deserealización?
168. ¿Qué es un esquema XSD?

UNIDAD V

1. ¿Qué es la reflexión?
2. ¿En qué namespace encontramos las clases que utilizamos en la reflexión?
3. ¿Para qué utilizamos la clase Assembly?
4. ¿Para qué utilizamos la clase AssemblyName?
5. ¿Para qué utilizamos la clase Module?
6. ¿Qué es un objeto Type?
7. ¿Qué namespace necesito importar para poder utilizar los objetos de tipo Type?
8. ¿Cuáles son las propiedades de instancia de un objeto Type cuya categoría es Identidad?

9. ¿Cuáles son las propiedades de instancia de un objeto Type cuya categoría es Clasificación?
10. ¿Cuáles son las propiedades de instancia de un objeto Type cuya categoría es Ámbito?
11. ¿Cuáles son las propiedades de instancia de un objeto Type cuya categoría es Codificación?
12. ¿Cuáles son las propiedades de instancia de un objeto Type cuya categoría es Diseño de Campo?
13. ¿Cuáles son las propiedades de instancia de un objeto Type cuya categoría es Referencias a otros tipos?
14. ¿Cuáles son las propiedades de instancia de un objeto Type cuya categoría es Herencia?
15. ¿Cuáles son las propiedades de instancia de un objeto Type cuya categoría es Otros atributos?
16. ¿Cuáles son las propiedades de instancia de un objeto Type cuya categoría es Miscellaneous?
17. ¿Qué métodos posee el objeto Type?
18. ¿Se pueden crear objetos dinámicamente con la reflexión?
19. ¿Qué aspectos de seguridad tendría en cuenta al utilizar reflexión?

UNIDAD VI

20. ¿Qué es un proceso en el contexto de subprocesos de programación?
21. ¿Qué namespace es necesario para utilizar las ventajas de los subprocesos?
22. Describa un contexto donde se haga evidente la necesidad de utilizar subprocesos.
23. ¿Qué objeto se debe utilizar para crear un subproceso?
24. ¿Para qué se utiliza el atributo ThreadStatic?
25. ¿Cuáles son los posibles valores de la propiedad ThreadState?
26. ¿Cómo se pueden pasar y obtener datos de un subproceso?
27. ¿Qué significa sincronizar procesos?

28. ¿Para qué utilizamos la instrucción SyncLock?
29. ¿Qué característica debe tener la variable utilizada en el bloque SyncLock?
30. ¿Por qué podemos tener la necesidad de sincronizar objetos?
31. ¿Para qué se utiliza el atributo <Synchronization>?
32. ¿De dónde deben heredar las clases sincronizadas?
33. ¿En qué namespace se encuentra la clase ContextBoundObject?
34. ¿Cómo podemos evitar tener que sincronizar toda la clase completa?
35. ¿Para qué se utiliza el atributo <MethodImpl>?
36. ¿Para qué se utiliza la clase Monitor?
37. ¿Para qué se utiliza la clase Interlocked?
38. ¿Para qué se utiliza la clase Mutex?
39. ¿Para qué se utiliza la clase ReaderWriterLock?
40. ¿Para qué se utiliza la clase ManualResetEvent?
41. ¿Para qué se utiliza la clase AutoResetEvent?
42. ¿Para qué se utiliza la clase ThreadPool?
43. ¿Cómo puedo manejar temporizadores?
44. ¿Qué diferencia podemos observar entre el uso de System.Timers.Timer y System.Threading.Timer?
45. ¿Qué es una operación asíncrona?
46. ¿Qué permite realizar y con qué característica una llamada asíncrona?
47. ¿Qué es un procedimiento asíncrono de devolución de llamada?

GUÍA DE TRABAJOS PRÁCTICOS

UNIDAD I

1. Seleccione un sistema de información de gran magnitud. Describalo identificando las gestiones que se encuentran en su dominio. Identifique que gestiones poseen mayor complejidad y realice una lista detallada de los factores que determinan a las gestiones por Ud. seleccionadas como de gran complejidad.
2. Confeccione una lista con todas las tareas a tener en cuenta para desarrollar un sistema de información. Ordene esa lista en forma descendente de acuerdo al riesgo que representa ejecutar cada actividad en un escenario propuesto por Ud. Luego reordene la lista también en forma descendente de acuerdo al impacto económico que generaría si el elemento de riesgo se materializa en una pérdida. Finalmente constituya la lista en forma descendente considerando la exposición al riesgo.
3. Desarrolle una planilla de cotejo que permita considerar y ponderar todos los aspectos a tener en cuenta a fin de poder comparar productos de software antes de la adquisición de uno de ellos.
4. Enuncie un escenario organizacional que posea una necesidad concreta sobre un sistema de información. Construya un cuestionario con las preguntas que Uds. le realizaría a los proveedores de este tipo de sistemas al momento que sea citado para la demo del producto.
5. Dada la situación en dónde Ud. trabaja en el departamento de sistemas de una organización, lo convocan, ya que el departamento de ventas desea informatizar los procesos llevados adelante allí. Estos procesos son verdaderamente complejos y Ud. Debe reunirse con el gerente del área y dos empleados que conocen en profundidad las gestiones llevadas a cabo. Debe confeccionar un documento que contenga la estrategia que va a emplear para obtener toda la información necesaria que le permita determinar si los sistemas se desarrollaran en la empresa o se adquirirán a un proveedor externo.

UNIDAD II

6. Desarrolle un módulo que administre el log-in y log-out de un sistema de información. Lograr que el componente sea lo suficientemente funcional como para que pueda ser reutilizado en distintos sistemas de información. Aplíquelo en el proyecto práctico de la asignatura.

7. Desarrolle un módulo que administre la encriptación de un sistema de información. Lograr que el componente sea lo suficientemente funcional como para que pueda ser reutilizado en distintos sistemas de información. Aplíquelo en el proyecto práctico de la asignatura.
8. Desarrolle un módulo que administre el manejo de perfiles de usuarios de un sistema de información. Lograr que el componente sea lo suficientemente funcional como para que pueda ser reutilizado en distintos sistemas de información. Aplíquelo en el proyecto práctico de la asignatura.
9. Desarrolle un módulo que administre los backup de un sistema de información. Lograr que el componente sea lo suficientemente funcional como para que pueda ser reutilizado en distintos sistemas de información. Aplíquelo en el proyecto práctico de la asignatura.
10. Desarrolle un módulo que administre los múltiples idiomas dinámicos con los que se puede trabajar en un sistema de información. Lograr que el componente sea lo suficientemente funcional como para que pueda ser reutilizado en distintos sistemas de información. Aplíquelo en el proyecto práctico de la asignatura.
11. Desarrolle un módulo que administre el log-in y log-out de un sistema de información. Lograr que el componente sea lo suficientemente funcional como para que pueda ser reutilizado en distintos sistemas de información. Aplíquelo en el proyecto práctico de la asignatura.
12. Desarrolle un módulo que administre la bitácora de un sistema de información. Lograr que el componente sea lo suficientemente funcional como para que pueda ser reutilizado en distintos sistemas de información. Aplíquelo en el proyecto práctico de la asignatura.

UNIDAD III

13. Confeccione una plantilla que contenga todos los elementos necesarios para catalogar un patrón de diseño.
14. Confeccione una plantilla que contenga todos los elementos necesarios para catalogar un anti patrón de diseño.
15. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón Builder. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.
16. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón Factory Method. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.

17. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón Bridge. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.
18. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón Facade. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.
19. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón Flyweight. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.
20. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón Proxy. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.
21. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón Chain of Responsibility. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.
22. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón Command. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.
23. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón State. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.
24. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón Strategy. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.
25. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón Template Method. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.
26. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón Visitor. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.

UNIDAD IV

27. Construya un programa que posea un objeto que se pueda serializar. Serialícelo de forma binaria. Después aplíquelo deserialización y demuestre que obtuvo lo deseado.

28. Construya un programa que posea un objeto que se pueda serializar. Serialícelo en formato SOAP. Después aplíquelo deserialización y demuestre que obtuvo lo deseado.
29. Construya un programa que posea un objeto que se pueda serializar. Seleccione solo un subconjunto de miembros a serializar y proceda a hacerlo. Verifique que solo se serializó lo deseado. Proceda a demostrar que se puede deserializar.
30. Construya un programa que posea un grafo de objetos (con referencias circulares). serialícelo. Verifique que logró el objetivo. Proceda a demostrar que se puede deserializar.
31. Construya un programa donde se pueda observar claramente la aplicación de la interfaz ISerializable aplicada.
32. Construya un programa que posea un objeto que se pueda serializar. Serialícela utilizando XMLSerializer. Después aplíquelo deserialización y demuestre que obtuvo lo deseado.

UNIDAD V

33. Construya un programa que utilizando las características de la reflexión permita conocer los datos más importantes de un assembly.
34. Construya un programa que permita capturar en variables de tipo Type capturar distintos tipos y en base a ellos instanciar objetos dinámicamente.
35. Construya un programa que permita recuperar las propiedades y métodos de un objeto y mostrarlos en dos combos.
36. Construya un programa basándose en el ejercicio del punto anterior y permita que a una propiedad seleccionada del combo se le pueda cargar un valor y leerlo.
37. Construya un programa basándose en el ejercicio del punto anterior y permita ejecutar un método seleccionado del combo.

UNIDAD VI

38. Construya un programa donde se pueda observar contadores que muestren sus resultados en pantalla. Cada contador debe operar en un subproceso distinto.
39. Construya un programa donde se pueda observar al menos dos subprocesos ejecutándose y que ambos compartan un dato.
40. Construya un programa donde se pueda observar subprocesos trabajando de manera sincronizada.

41. Construya un programa donde se pueda observar timers funcionando simultáneamente en distintos procesos. Cada Timer de servir para que el programa realice una operación distintiva.
42. Construya un programa donde se pueda observar claramente cómo se puede implementar el uso de subprocesos con llamado asincrónico a operaciones.

GUÍA DE ABORDAJE BIBLIOGRÁFICO

UNIDAD I

Pressman, Roger S. **Ingeniería del software: un enfoque práctico**. 6a.ed.-- México, DF: McGraw-Hill, c2005. Capítulo 21.

Complementaria - Ampliatoria:

Piattini Velthuis, Mario G.; García Rubio, Félix O. **Calidad en el desarrollo y mantenimiento del software**-- México, DF: Alfaomega, c2003. XXV, 310 páginas

Schach, Stephen R.; Fernández, Esther (Traductor); Guerrero, Ekaterina (Traductor) y otros. **Ingeniería de software clásica y orientada a objetos**. 6a. ed.-- México, D.F.: McGraw-Hill Interamericana de México, 2006. XXI, 581 páginas

UNIDAD II

MSDN. Microsoft developer Network. <http://msdn.microsoft.com/es-ar/default.aspx>

- Seguridad.
- Gestión de accesos y perfiles.
- Encriptado de datos.
- Administración de múltiples idiomas.
- Gestión de Backup.
- Código seguro en las aplicaciones.

UNIDAD III

Gamma, Erich; Helm, Richard; Johnson, Ralph y otros. **Patrones de diseño: elementos de software orientado a objetos reutilizable**-- Madrid: Pearson Educación. c2003. xvii, 364 páginas

UNIDAD IV

Balena, Francesco. **Programación avanzada con Microsoft Visual Basic.Net**-- México, DF: McGraw-Hill, c2003. Capítulo 11.

UNIDAD V

Balena, Francesco. **Programación avanzada con Microsoft Visual Basic.Net.**-- México, DF: McGraw-Hill, c2003. Capítulo 15.

UNIDAD VI

Balena, Francesco. **Programación avanzada con Microsoft Visual Basic.Net.**-- México, DF: McGraw-Hill, c2003. Capítulo 13.