

The background of the entire page is a photograph of two men in business suits. They are both smiling and looking at a tablet computer. The man on the right is holding a pen and pointing at the screen. The lighting is warm and professional.

Diplomado. Desarrollo de aplicaciones empresariales con Java SE Java EE y Frameworks.



Agente capacitador externo.
Registro ante la Secretaria del Trabajo y Previsión Social
No. TAI100831FM2-0013.

Revisión 2015B

Curso:	Diplomado. Desarrollo de Aplicaciones Empresariales con Java SE, Java EE y Frameworks.
Duración:	192 horas. Divididas en 4 módulos, cada uno con una duración de 48 horas.
Introducción:	Java es un lenguaje de programación de alto nivel orientado a objetos, desarrollado por James Gosling en 1995. El lenguaje en sí mismo toma mucha de su sintaxis de C, Cobol y Visual Basic, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. La memoria es gestionada mediante un recolector de basura. Java Platform, Enterprise Edition o Java EE, es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. La plataforma Java EE está definida por una especificación.
Destinatarios:	Profesionales y estudiantes de las siguientes áreas de la ingeniería o afines: Informática Sistemas Computacionales Electrónica Mecatrónica Biónica Telecomunicaciones Computación Robótica Control Automático Matemáticas Física. Y entusiastas de la tecnología.
Objetivos:	<ol style="list-style-type: none"> 1. Desarrollar software con el lenguaje de programación Java y bajo los estándares de la especificación JEE y Frameworks como Spring, Struts2 y Hibernate principalmente. 2. Diseñar y definir soluciones programables a diferentes problemas de negocio. 3. Analizar requerimientos y procesos de negocio. 4. Definir y construir arquitecturas base.
Conocimientos previos:	Es deseable que el participante tenga conocimientos básicos de programación en C, C++ u otro lenguaje o plataforma de programación aunque no es imprescindible. Ganas de aprender, disposición y compromiso.
Software:	La instalación y uso del software se hará durante el diplomado; todas las herramientas serán open source y/o evaluaciones trial de algunos productos propietarios.
Modalidad:	Sabatino. Nuestra metodología es bastante personalizada, ya que nuestros grupos de aprendizaje son pequeños, el Diplomado Java en su mayor porcentaje está enfocado a la práctica, con problemas reales y transmitiéndote la experiencia de cada uno de los instructores, sin embargo, hay teoría que es necesaria estudiar, entender y comprender.
Horario(s):	Los publicados en la url www.grupoxita.com.mx/#java
Precios por módulo:	<p>Precios para profesionistas que NO asistan con laptop. Pago en efectivo, por depósito bancario, transferencia bancaria o tarjeta de crédito: \$ 7,200 MXN.</p> <p>Precio para profesionistas que SI asistan con laptop. Pago en efectivo, por depósito bancario, transferencia bancaria o tarjeta de crédito: \$ 6,500 MXN.</p> <p>Precio para estudiantes que SI asistan con laptop. Pago en efectivo, por depósito bancario, transferencia bancaria o tarjeta de crédito: \$ 4,850 MXN.</p> <p>Estos precios no incluyen IVA, para la emisión de su factura electrónica deberá cubrir este monto y enviar sus datos fiscales a training@grupoxita.com.mx.</p>
Facilidades de pago:	<ol style="list-style-type: none"> 1. El participante puede cubrir el costo de cada módulo en dos partes del 50% cada una; si lo realiza en efectivo, por depósito bancario o transferencia bancaria: <ul style="list-style-type: none"> - Primer pago. Hasta tres días hábiles antes de que de inicio el módulo. - Segundo pago. Durante la tercera semana después de iniciado el módulo. 2. El participante puede realizar el pago de cada uno de los cuatro módulos con tarjeta de crédito a tres meses sin intereses usando nuestra forma de pago por PayPal. Esta forma de pago no incluye IVA, para la emisión de su factura electrónica deberá cubrir este monto en efectivo, por depósito bancario o transferencia bancaria y enviar sus datos fiscales a training@grupoxita.com.mx

Datos de cuentas:	BBVA Bancomer. Tecnologías Avanzadas y de la Información, Grupo XITA BROS S.A. de C.V. No. de Cuenta: 0184920570 CLABE: 012 180 00184920570 8 RFC: TAI100831FM2
Reposición de clase:	Tendrá un costo de \$300.00 MXN + IVA con duración de dos horas y se programará de acuerdo a la disponibilidad de los instructores, entre semana.
Inscripciones:	La inscripción se podrá realizar hasta 3 días hábiles antes del inicio del curso. Para inscribirse el interesado deberá enviar un mail a training@grupoxita.com.mx con su nombre completo, teléfono de contacto, nombre del curso, horario de elección, y haber realizado el pago. Si el pago decide hacerlo en efectivo puede pasar a nuestra oficina previa cita de lunes a viernes, en un horario de 11 a.m. a 7 p.m.
Junta informativa:	Puede solicitar una junta informativa en donde se expondrá con mayor amplitud el contenido del Diplomado y se dará respuesta a sus preguntas. Al solicitar esta junta usted no adquiere ningún compromiso de inscripción.
Aplazamiento o cancelación del curso de parte de XITA:	Grupo XITA se reserva el derecho de poder cancelar o aplazar la fecha de inicio de un curso cuando no se haya cubierto el cupo mínimo. En caso de aplazamiento XITA dará a conocer a los participantes el estado del curso y la nueva fecha de inicio lo antes posible. El participante tendrá la opción de elegir si desea emplear su pago para la nueva fecha ó si desea la devolución de éste.
Cancelación del participante:	El participante podrá cancelar su inscripción hasta antes de la fecha de inicio del curso; si lo hace dentro de los dos días previos al inicio se le cobrará un 20% del costo total por gastos administrativos.
Instructor:	Grupo XITA le garantiza que usted contará con un instructor profesional con sólida trayectoria industrial y/o académica.

TEMARIO.

Módulo.	Tema.
1. Programación.	<ol style="list-style-type: none"> 1. ¿Qué es un programa? 2. Sistema operativo. <ul style="list-style-type: none"> > ¿Qué es un proceso? 3. ¿Qué es un lenguaje de programación? <ul style="list-style-type: none"> > ¿Qué es JAVA? 4. Editores, Compiladores, e Intérpretes. <ul style="list-style-type: none"> > Lenguaje interpretado Vs. lenguaje compilado. > Lenguaje fuertemente tipados Vs. lenguajes no tipados. 5. Lenguaje C++ Vs. lenguaje JAVA. 6. Características de lenguaje de programación JAVA. <ul style="list-style-type: none"> > Orientado a objetos. >Distribuido. > Interpretado y compilado a la vez. > Robusto. > Seguro. > Indiferente a la arquitectura. > Portable. > Alto rendimiento. > Multihilo. > Dinámico. 7. Pasado, presente y futuro de JAVA. 8. Los padres de la criatura. <ul style="list-style-type: none"> > James Gosling. > Bill Murray. 9. JAVA está en todas partes. 10. ¿Por qué aprender JAVA? 11. ¿JAVA, es fácil de aprender? 12. JAVA en el mundo empresarial. 13. ¿JAVA standard edition Vs. JAVA enterprise edition? 14. Arquitectura JAVA Standard Edition. 15. ¿Que es un API en JAVA? <ul style="list-style-type: none"> > Principales APIs en JAVA. 16. Introducción a las interfaces gráficas de usuario. <ul style="list-style-type: none"> > ¿Es el fin de los applets? > Bailando con Swing. <ul style="list-style-type: none"> >> Alternativa swing con griffon. >> Look and feel en aplicaciones de escritorio. >> Front-end Vs. Back-end. >> ¿y JAVA FX? 17. Versiones de JAVA. <ul style="list-style-type: none"> > Viejitas pero NO tan bonitas. >> Cronología de versiones del lenguaje. 18. La versión JAVA 1.7 <ul style="list-style-type: none"> > Novedades en la versión 7. > Novedades en la versión 8. 19. Plataforma JAVA Vs. Plataforma .NET. 20. Arquitectura JAVA Enterprise Edition. 21. Prontuario de términos. JAVA, API, Applet, JavaScript, JQuery, ExtJS, JEE, JSE, JVM, JRE, JDK, JAR, WAR, EAR, JSP, EJB, JavaBean, Framework, Patrón de diseño, DDT, IDE, Contenedor, Servidor de aplicaciones, UML, RUP, Prueba unitaria, JCP, JNLP, JNDI, Socket, RMI, Thread, Classloader, Closure, etc. 22. Instalando y configurando ambiente de desarrollo básico. <ul style="list-style-type: none"> > JDK 1.7 y JDK 8. <ul style="list-style-type: none"> >> Instalando y configurando el JDK 1.7. / JDK 8 >>> Variables de entorno. >>>> PATH.

1
Programación.

```

>>>> CLASSPATH.
>>>> JAVA_HOME.
>>> Desmenuzando el JDK.
>>>> Herramientas y directorios.
>>>>> Herramientas principales.
>>>>>> javac.
>>>>>> java.
>>>>>> jdb.
>>>>>> jar.
>>>>>> javap.
>>>>>> javah.
>>>>>> javadoc.
>>>>> Directorios.
>>>>>> bin.
>>>>>> demo.
>>>>>> include.
>>>>>> sample.
>>>>>> jre.
>>>>>> lib.
> Eclipse / NetBeans.
>> Configurando características de arranque y variables.
>> Ant y Maven.
>> MySQL.
>> Web Logic o Glassfish Application Server.
>> Tomcat.
>> JUnit.
>>> Pruebas Unitarias y TDD.
>>> Escalabilidad y desempeño.
>>> ¿Por qué es difícil escribir código de calidad?
>>> Test-driven Development (TDD).
>>>> Beneficios.
>>>> Argumentos en contra TDD.
>>>>> Testing.
>>>>>> Objetos Mock.
>>>>>> Mockito.
>>>>>> httpUnit.
> Control de Versiones.
>> CVS.
>> SVN.
>> Diferencias entre CVS y SVN.
>> ¿Qué es GitHub?
23. Proceso de compilación y ejecución de un programa JAVA.
24. ¿Qué es la JAVA Virtual Machine?
> Tiempo de vida de una Java Virtual Machine.
25. Entendiendo y comprendiendo la JAVA Virtual Machine.
26. Arquitectura de la JAVA Virtual Machine.
> Cargador de Clases - Classloader.
>> ¿Cómo se cargan las clases?
>> Haciendo nuestro cargador de clases.
>> API Reflection.
> Verificador de Bytecode.
> Garbage Collector.
>> System.gc().
> Security Manager.
> Execute Engine.
27. Organización de la JAVA Virtual Machine.

```


1
Programación.

- 27. Organización de la JAVA Virtual Machine.
 - > Class area.
 - >> Method area.
 - >> Constantes.
 - >> Runtime Constant Pool.
 - > JAVA Stack.
 - >> Frame.
 - >> Stack.
 - >> PC.
 - > Heap.
 - >> Parámetros de configuración.
 - >>> -Xms.
 - >>> -Xmx.
 - >>> -Xmn.
 - >> Jconsole.
 - >> Java VisualVM.
 - >> Eden Space.
 - >> Survivor Space.
 - > PermGen Space.
 - > Native method stacks.
 - >> Native Area.
 - > Que es un Bytecode.
 - >> Bytecode Vs. MSIL de Microsoft.
 - > JAVA Runtime Environment.
 - > Cambios en la JVM Java 8.
 - > Administración de memoria en JAVA.
 - >> Goteos de memoria.
 - >> Heap.
 - >> Garbage Collection.
 - > Lenguajes “alternativos” para la JAVA Virtual Machine.
 - >> Groovy.
 - >> Scala.
 - >> Clojure.
 - >> JRuby / Jython.
 - > Otras máquinas virtuales.
 - >> JRocket.
 - >> OpenJDK.
 - >> et al.
- 28. Aprendiendo a conocer el lenguaje.
- 29. Ejecución y compilación.
- 30. Entendiendo y comprendiendo el proceso de ejecución de un programa en JAVA.
- 31. Entendiendo y comprendiendo el proceso de compilación de un programa en JAVA.
- 32. Opciones de compilación.
- 33. Opciones de compilación cruzada.
- 34. Visualizando proceso de ejecución de un programa en JAVA mediante software.
- 35. Program Animation/Visualization tool – Jeliot.
- 36. Compilador JIT.
- 37. Mi primera taza de café.
 - > Estructura: Partes de un programa en JAVA.
 - > Creación de archivo de código.
 - > Archivo punto JAVA.
 - > Buenas prácticas de programación.
- 38. Técnicas de abstracción de datos.
- 39. Diagramas de Flujo.
- 40. Mi primera taza de café.
 - > Para no perder la costumbre. Hola Mundo.

1
Programación.

41. ¿Aprender a programar?
42. Algoritmos.
43. Seudo código.
44. Debugando una aplicación con el IDE Eclipse.
 - > Puntos de ruptura.
45. Performance Vs diseño de la aplicación.
46. El concepto de paquete en JAVA.
 - > La sentencia package.
 - > Estructura de directorios en la definición de paquetes.
 - > Como diseñar paquetes.
 - > La sentencia import.
 - > import implícito Vs. import explícito.
 - > La palabra clave static.
 - > La sentencia static import.
47. Corriendo nuestra primera aplicación JAVA
 - > El método main().
48. Entrada y salidas
 - > La poderosa línea de comando.
 - > La clase java.io.Console.
 - > La clase java.util.Scanner.
 - > El paquete java.util.
49. Elementos del lenguaje de programación en JAVA.
 - > ¿Qué es una gramática?
 - > ¿Qué es una sintaxis?
 - > Letras, dígitos y otros.
 - > Espacios en blanco.
 - > Caracteres especiales y signos de puntuación.
 - > Secuencias de escape.
 - > Comentarios.
 - > Simple línea de comentario.
 - > Comentarios multilinea.
50. Un código bien documentado es un programador feliz.
51. La API JAVA doc.
 - > Aprendiendo a comentar correctamente nuestro código.
 - > Buenas prácticas.
 - > Notaciones y estándares.
 - > Tags.
 - >> @author.
 - >> @author.
 - >> @deprecated.
 - >> @param.
 - >> @see.
 - >> @since.
 - >> @return.
52. Generando documentación.
 - > Comando Javadoc.
 - > El paquete com.sun.javadoc.
 - > La Interface Taglet.
53. Variables.
54. ¿Qué es una variable?
55. ¿Por qué se llaman variables?
56. Nombrando variables.
 - > Unicode.
57. Identificadores en JAVA.
 - > Es necesario mencionarlo: palabras reservadas en JAVA.
58. Declarando variables correctamente.

1
Programación.

59. Inicializando variables: ¿Dónde? ¿Cómo? ¿Cuándo?
60. Inicialización dinámica.
61. Ciclo de vida de una variable en JAVA.
62. Las variables y la JAVA Virtual Machine.
63. Ámbito de una variable.
64. Modificadores de acceso de variables.
 - > Modificador private.
 - > Modificador protected.
 - > Modificador public.
 - > Modificador default.
65. Constantes en JAVA.
 - > La palabra reservada: final.
 - > ¿Qué es una constante?
66. Variables tipo static.
67. Literales en JAVA.
68. ¿Qué es un literal?
 - > Literales enteros.
 - > Literales reales.
 - > Literales de un solo carácter.
 - > Literales de cadenas de caracteres.
 - > Literal Vs. Constante.
 - > Evaluando expresiones en JAVA
69. Operadores.
 - > Operadores de incremento y decremento.
 - > Operadores de asignación.
 - > Operadores de igualdad y comparación.
 - > Operadores lógicos.
 - > Operador de negación.
 - > Operadores de precedencia.
70. Datos en JAVA.
71. Tipos de datos simples.
72. Tipos de datos primitivos en JAVA.
 - > double.
 - > int.
 - > byte.
 - > short.
 - > long.
 - > char.
 - > float.
73. Tipo de dato boolean.
74. Operadores de comparación.
75. Clases envoltorio.
76. Clases envoltorio Vs. datos primitivos.
 - > Double Vs. double.
 - > Integer Vs. int.
 - > Long Vs. long.
 - > Float Vs. float.
 - > BigDecimal Vs. Float Vs. Double.
 - > BigInteger.
 - > ¿Cuándo usarlos?
77. Autoboxing y Autocasting.
78. Tipos primitivos Vs. tipos referenciados.
79. Expresiones regulares.
80. Estructuras de control.
 - > Sentencia if.
 - > Sentencia if-else.

1
Programación.

- > Sentencia if anidada.
- > Sentencia switch.
- > if Vs. switch. 81. Loops.
- > for.
- > collection for.
- > while.
- > do-while.
- 82. La sentencia: continue.
- 83. La sentencia: break.
- 84. Funciones matemáticas y constantes.
 - > El paquete java.math.
- 85. Definiendo métodos correctamente en JAVA.
 - > Modificadores de acceso en métodos.
 - > ¿Qué son los closures?
- 86. Declarando métodos correctamente en JAVA.
- 87. Sobreescritura de métodos.
- 88. Sobrecarga de métodos.
- 89. Constructores en Java,
 - > El operador this.
 - > El operador super.
- 90. Entendiendo los métodos estáticos.
- 91. El bloque static.
- 92. API Introspección en JAVA.
- 93. Paso por valor Vs. paso por referencia.
- 94. ¿Cómo pasan exactamente los valores en JAVA?
- 95. Arreglos y Strings.
- 96. Arreglos unidimensionales.
- 97. Arreglos multidimensionales.
- 98. Ordenamiento.
 - > Ordenamiento bubble.
 - > Ordenamiento merge.
 - > Ordenamiento quick.
 - > Ordenamiento heap.
- 99. El paquete java.lang.
- 100. La clase java.lang.String.
 - > ¿Qué es un String?
 - > ¿Dónde viven los Strings?
 - > Concatenando Strings.
 - > El método toString().
 - > Sobrescribiendo el método toString().
 - > La clase StringBuilder.
 - > La clase StringBuffer.
 - > La clase StringWriter.
 - > Operaciones.
 - > String Vs. StringBuilder Vs. StringBuffer vs StringWriter.
 - > La Clase ArrayList.
- 101. Expresiones regulares.
 - > Clase Pattern.
 - > Clase Matcher.
- 102. Objetos inmutables.
 - > Patrón inmutable.
 - > Características.
- 103. La Interface CharSequence.
- 104. Fechas y horas en Java.
- 105. Manipulación de Bits.
 - > Operaciones a nivel de bits.

1
Programación.

- > La clase BitSet.
- > Operadores:
 - >> &, |, ^, <<, >>, >>>, ~
- 106. Programación Orientada a Objetos.
- 107. ¿Qué es un paradigma?
- 108. ¿Qué es la programación orientada a objetos?
- 109. Características de la programación orientada a objetos.
- 110. Objetos y el mundo real.
- 111. Programación estructurada Vs. programación orientada a objetos.
- 112. Otros paradigmas.
- 113. Java será funcional y Orientado a Objetos.
- 114. Introducción: Patrones de diseño.
- 115. ¿Qué es un patrón?
 - > ¿Qué es un anti patrón?
 - > Patrones GRASP.
 - > Patrones GOF.
- 116. Entendiendo y comprendiendo el concepto de clase como patrón de diseño.
- 117. Técnicas de abstracción de datos.
- 118. Estructura de una clase en JAVA.
- 119. Implementación de concepto de clase en JAVA.
- 120. Diagramación UML.
 - > Integrando UML en Eclipse IDE.
 - >> Diagramas de clases.
 - >> Diagrama de secuencia.
 - >> Diagrama de colaboración.
- 121. Concepto de alta cohesión
- 122. Concepto de bajo acoplamiento.
- 123. Composición.
 - > Diagrama de composición en UML.
- 124. Agregación.
 - > Diagrama de agregación en UML.
- 125. Composición Vs. agregación.
- 126. Clases y Objetos en Java.
- 127. Tipos de datos de referencia.
 - > Objetos.
 - >> La Clase Object.
 - >>> El método toString().
 - >>> El método equals().
 - >>> El método hashCode().
 - >>> El método clone().
 - > La clase Class.
 - > Encapsulamiento.
 - > Definiendo clases en JAVA.
 - >> Anatomía de una clase.
 - > ¿Qué son los objetos en JAVA?
 - > Inicialización y constructores.
 - > Construcción de Objetos.
 - >> El operador new en JAVA.
 - >> Midiendo el tamaño de nuestros objetos.
 - >> API Instrumentation.
 - >> memoryagent.jar
 - > Comparando objetos.
 - > Relaciones.
 - >> "Is a".
 - >> "Has a".
 - > Constructores y sobrecarga de constructores.

1
Programación.

- > Clases internas.
 - >> Clases estáticas.
 - >> Clases anónimas
 - >> Clases abstractas.
 - >> Clase finales.
 - >> Clases mutables e inmutables.
- 128. El patrón de diseño Singleton.
 - > ¿Existen los singletons en Java?
- 129. Herencia entre clases.
 - > Super clases.
 - > Sub clases.
- 130. Generalización de clases.
- 131. Especialización de clases.
- 132. Composición Vs. herencia.
- 133. El método isInstance().
- 134. La palabra reservada instanceof.
- 135. El método getClass().
- 136. Reutilización de código.
- 137. Interfaces.
- 138. ¿Qué es una interface?
 - > El concepto de interface como patrón de diseño.
 - > Herencia múltiple.
 - > Problemas asociados con al herencia múltiple.
- 139. Herencia múltiple usando interfaces.
 - > Atributos en una interface
 - > Métodos en una interface.
 - > Heredando interfaces.
 - > Limitaciones de las interfaces.
- 140. ¿Cuándo usar interfaces?
- 141. ¿Cuándo usar clases abstractas?
- 142. Interfaces Vs. Clases abstractas.
- 143. El patrón Strategy.
- 144. El patrón Observer.
- 145. Polimorfismo.
 - > ¿Qué es el polimorfismo?
 - >> Enlace dinámico.
 - >> Enlace estático.
- 146. Acoplamiento.
- 147. Métodos polimórficos.
- 148. Tipos de retorno covariantes.
- 149. Introducción a la arquitectura JAVA Beans.
 - > El patrón DTO.
 - > El patrón VO.
 - > El patrón TO.
 - > ¿Son lo mismo?
- 150. Manejo de archivos en JAVA.
 - > Entendiendo los Streams.
 - > Flujos de datos en JAVA.
 - > Archivos y directorios.
 - >> Lectura y escritura de Archivos con Java.
 - >> Archivos binarios Vs. archivos de texto.
 - >> La Clase PrintWriter.
 - >> Clases para manejo de Archivos Binarios.
 - >>> La Clase InputStream.
 - >>>> La Clase FileInputStream.
 - >>>> La Clase ObjectInputStream.

1
Programación.

```

>>>> La Clase ObjectInputStream.
>>>> La Clase FilterInputStream.
>>>>> La Clase FilterInputStream.
>>>>> La Clase DataInputStream.
>>>>> La Clase BufferedInputStream.
>>> La Clase OutputStream.
>>>> La Clase FileOutputStream.
>>>> La Clase ObjectOutputStream.
>>>> La Clase FilterOutputStream.
>>>>> La Clase DataOutputStream.
>>>>> La Clase BufferedOutputStream.
>> Abriendo archivos para acceso secuencial.
>>> Flujo de bytes.
>>>> La Clase FileOutputStream.
>>>> La Clase FileInputStream.
>>> La Clase File.
>>> Flujo de caracteres.
>>>> La Clase FileWriter.
>>>> La Clase FileReader.
>> PrintWriter Vs. FileWriter.
>> PrintWriter Vs. BufferedWriter.
>> Acceso aleatorio de archivos.
151. Recursión en Java.
> Recursión Vs. Iteración.
> Recursión directa Vs. Recursión indirecta.
> El clásico: Torres de hanoi.
152. ¿Qué es la serialización?
> Serialización de objetos en Java.
> ¿Para qué usamos la serialización en JAVA?
> ¿Serializar Clases u Objetos?
> La Interface Serializable.
>> Excepción NotSerializableException.
>> Tipo de dato transient.
>> El famoso serialVersionUID.
> Concepto de deserialización en JAVA.
> La interface Externalizable.
> Serializando arreglos.
153. Excepciones
> ¿Qué es una excepción?
> ¿Para qué sirve una excepción?
>> Manejo de excepciones en Java.
>>> La Clase Throwable.
>>> La Clase Error.
>>> La Clase Exception.
> Tipos de Excepciones.
> La clase Exception y el operador instanceof.
> Detección y manejo de una excepción.
>> El bloque try.
>> El bloque catch.
>> El bloque finally.
>> La traza de la pila.
>>> El método printStackTrace.
>>> Regeneración y encadenamiento de Excepciones.
> La sentencia throw.
> La cláusula throws.
> Precondiciones y post condiciones.

```

1 Programación.	<ul style="list-style-type: none">> Excepciones y Aserciones.> Errores en tiempo de ejecución.> Creando nuestras propias excepciones.> Errores lógicos.> Técnicas en el manejo de excepciones.<ul style="list-style-type: none">>> Terminar el programa.>> Corregir y continuar.>> Logs.>> Cuando usar y cuando no usar excepciones.>> Excepciones comunes en el manejo de archivos.>> Técnicas de debugeo.154. Técnicas de programación.155. Buenas Prácticas de Programación en JAVA.156. Introducción patrón MVC.<ul style="list-style-type: none">> Modelo Vista Controlador.> Arquitectura cliente servidor.157. Patrón Service Locator.158. Proyecto. Construcción de un framework basado en patrones de diseño.159. Evitando ser un programador COPY/PASTE.<ul style="list-style-type: none">> Saber usar la documentación de JAVA.160. Reingeniería Vs. Refactoring.161. ¿Por qué son importantes las arquitecturas?
--------------------	--

2
Desarrollo.

1. Anotaciones en Java.
2. ¿Qué son las anotaciones?
 - > Tipos de anotaciones.
 - >> @Override.
 - >> @Deprecated.
 - >> @SuppressWarnings.
 - >> @SafeVarargs.
 - >> @Inherited.
 - >> @Documented.
 - > Meta Anotaciones.
 - >> @SuppressWarnings.
 - >>> @Target.
 - >>> @Retention.
 - >> Definiendo nuestras anotaciones.
 - >>> @Interface.
 - >>> @Stub.
3. Internacionalización.
 - > I18n: Ejemplo básico.
 - > Aspectos a internacionalizar.
 - > Texto y datos intercalados.
 - > Ejemplo con MessageFormat.
 - > Formato de fechas y horas.
 - > Fechas y horas. Ejemplo.
 - > Patrones para fechas y horas.
 - >> La Clase Locale.
 - >> La Clase TimeZone.
 - >> La Clase DateFormat.
 - >> La Clase DateFormatSymbols.
 - > Ejemplo de formato.
 - > Formato de números.
 - > Patrones para números.
 - > Reglas de ordenación.
4. Ejecución de programas no Java.
 - > Ejecución de programas no Java
 - > Interfaz Nativa de Java.
 - > Compilación del programa C.
 - > Ejecución de programas en C Sharp.
 - >> Paso de parámetros.
 - > Acceso al objeto principal. Campos
 - > Acceso al objeto principal. Métodos
 - > Manejo de excepciones
5. Generics en JAVA.
 - > Downcasting Vs. Generics.
 - > La excepción ClassCastException.
 - > ¿Qué son y para qué sirven?
 - > Implementación y tiempo de compilación.
 - > Definiendo Generics en Clases e Interfaces.
 - > Nomenclatura.
 - >> E,T,K,V.
 - > Tipos de argumentos en Generics.
 - >> "Concrete type".
 - >> "Concrete parameterized".
 - >> "Array".
 - >> "Type parameter".
 - >> "Wildcard".
 - > Definiendo Generics en métodos.

2
Desarrollo.

- > Parametrizando métodos.
 - >> Sobrecargando métodos genéricos.
 - >> Arreglos y Generics.
- 6. Tipos Enum.
- 7. Clases tipo Enum.
- 8. ¿Qué es un Framework?
- 9. El Framework de colecciones JAVA.
- 10. Patrón de diseño Iterator.
- 11. Patrón de diseño Factoría.
- 12. Patrón de diseño Factoría Abstracta.
- 13. Entendiendo el framework de colecciones.
 - > Arquitectura.
 - >> Core de interfaces.
 - >> Clases.
 - >> Utilerías.
- 14. Listas, Pilas y Colas.
- 15. Interface Cloneable y Serializable.
- 16. El paquete java.util.
- 17. Colecciones principales de objetos.
 - > Sets.
 - > Maps.
 - > Sequences.
- 18. El loop for-each.
- 19. Iterator.
 - > La interface Iterator.
 - >> Métodos principales.
 - >>> hasNext().
 - >>> next().
- 20. Collection.
 - > La interface Collection.
 - > Clase AbstractCollection.
- 21. Tipos de colecciones principales de objetos en JAVA.
 - > Sets.
 - >> Interface TreeSet.
 - >>> AbstractSet.
 - >>> HashSet.
 - >>> LinkedHashSet.
 - > Lists.
 - >> Las clases ArrayList y LinkedList.
 - >>> Ordenamiento de listas.
 - >>>> Métodos estáticos.
 - >> Desempeño de entre Sets y Lists.
 - >> Maps.
 - >>> HashMap.
 - >>> LinkedHashMap.
 - >>> TreeMap.
- 22. Comparable Vs. Comparator.
- 23. Comparación entre Collection e Iterator.
- 24. Algoritmos eficientes.
- 25. Árboles binarios en Java.
- 26. Patrón de diseño Fachada.
- 27. Patrón de diseño Proxy.
- 28. Patrón de diseño Bridge.
- 29. Patrón de diseño Adapter.
- 30. Patrón de diseño Command.
- 31. Continuación de nuestro framework con patrones de diseño.

2
Desarrollo.

32. Threads en JAVA
33. Concurrencia.
 - > Los Threads no son objetos.
 - > Entendiendo y comprendiendo los hilos en JAVA.
 - > Procesos.
 - > Semáforos.
34. Hilos y concurrencia.
35. Clásico: Productor consumidor.
36. ¿Por qué usar hilos?
37. Creando hilos en JAVA.
38. Modelo de memoria en Java.
39. Thread Safety.
 - > Operaciones atómicas.
 - > La palabra reservada synchronized.
40. Retomando Concepto de inmutabilidad.
41. La Clase Thread.
 - > Elementos de la clase Thread.
42. Ciclo de vida de un Thread en JAVA.
 - > Estados y prioridades.
43. La clase Object,
 - > El método wait().
 - > El método notify().
 - > El método notifyAll().
44. Iniciando Threads en JAVA.
 - > Método run().
 - > Método start().
 - > Método isAlive().
45. Terminando Threads en JAVA.
46. Pausando, suspendiendo y resumiendo threads en JAVA.
47. La Interface Runnable.
48. Interface Runnable Vs. Clase Thread.
49. ¿Cuándo usar la Clase Thread?
50. ¿Cuándo usar la Interface Runnable?
51. Sincronización de hilos en JAVA.
52. Operaciones atómicas.
53. El identificador volatile.
54. Condición race.
 - > ¿Cuándo se presenta?
55. Locks.
 - > Ámbito.
 - > La Interface Lock.
56. Sincronización de métodos Vs. Sincronización de bloques.
57. Deadlock.
58. Método wait().
59. Método notify().
60. Método notifyAll().
61. El paquete java.util.Concurrent.
62. Threads Pools.
 - > La Clase ThreadLocal.
 - > La Clase ThreadPoolExecutor.
63. La interface Callable.
64. La Interface Future.
65. El fork-join framework.
66. ThreadSafe y Colecciones.
 - > Colecciones sincronizadas.
67. Anotaciones.
 - > @GuardedBy.

2
Desarrollo.

- > @Immutable.
- > @ThreadSafe.
- > @NotThreadSafe.
- 68. JDBC.
- 69. Herramientas para Hibernate.
- 70. Spring Roo.
- 71. Java Beans y eventos.
- 72. Repositorios de datos.
- 73. Modelo relacional de datos.
- 74. Modelo entidad relación.
- 75. Identificando llaves.
- 76. Normalización de bases de datos.
- 77. Arquitectura y componentes JDBC.
- 78. JDBC DataSource.
- 79. Como trabaja JDBC.
 - > Estableciendo la conexión.
 - > Drivers.
 - > Tipos de Driver.
 - >> Tipo 1. "JDBC-ODBC Bridge"
 - >> Tipo 2. "Native-API-Partly-Java"
 - >> Tipo 3. "JDBC-Net-All-Java"
 - >> Tipo 4. "Native-Protocol-All-Java"
 - > Abriendo conexiones.
 - >> DriverManager.
 - >> DataSource.
 - >> DriverManager Vs. DataSource.
 - >> JDBC URL.
- 80. Connector MySQL.
- 81. SQL.
 - > Ejecutando sentencia SQL.
 - > SQL Básico.
 - >> Sentencia INSERT.
 - >> Sentencia SELECT.
 - >> Sentencia DELETE.
 - >> Sentencia UPDATE.
- 82. Ejecutando sentencias.
 - > Usando el objeto Statement.
 - > Usando el objeto PreparedStatement.
 - > Usando el objeto CallableStatement.
 - > Un poco más de métodos de ejecución.
- 83. Procesando resultados.
 - > La Interface ResultSet.
 - >> ResultSet Object.
 - > Usando la Interfaces JDBC metaData.
 - >> Interface ResultSetMetaData.
 - >> Interface DatabaseMetaData.
 - >> Interface ParameterMetaData.
- 84. Cerrando conexiones y sentencias.
- 85. Arquitectura de dos capas.
- 86. Arquitectura de tres capas.
- 87. Introducción a ORM.
- 88. Object Databases (ODBMS).
- 89. Java Data Objects (JDO).
- 90. Transacciones. > Spring / JPA / Hibernate.
 - > Rollback.
 - > Commit.

2 Desarrollo.	91. Introducción a JNDI. > Ventajas y desventajas. > javax.sql: Soporte JNDI. 92. DataSource. 93. Pool de Conexiones. 94. Struts 2 / Tiles. 95. Struts2. 96. Generando un aplicación web con Struts2. 97. Jquery para Struts2. 98. Modelo. 99. Vista. 100. Controlador.
------------------	--

3
Web.

1. ¿Qué hace un desarrollador Web Java?
 - > Roles.
2. ¿Qué es JEE?
3. Arquitectura JEE.
 - > Componentes.
 - > Versiones de JEE.
 - >> Java EE 5.
 - >> Java EE 6.
 - >> Java EE 7.
 - >> ¿Cuál es la diferencia?
 - >> ¿Vale la pena aprender todas?
 - > Arquitectura JEE: Beneficios.
4. Arquitectura cliente-servidor.
5. Stack de protocolos de red.
6. Arquitectura MVC.
 - > Ventajas y desventajas.
7. ¿Qué es un protocolo?
 - > Protocolo http.
 - > URL, URI, etc.
8. Entendiendo el protocolo HTTP.
 - > Protocolo HTTP.
 - >> Head.
 - >> Get.
 - >> Post.
 - >> Put.
 - >> Delete.
 - >> Trace.
 - >> Options.
 - >> Connect.
9. Servidor de aplicaciones.
 - > BEA Weblogic y Glassfish.
 - >> Configurando BEA Weblogic.
 - >> Configurando Glassfish.
10. Contenedor Web.
 - > Apache Tomcat
 - >> Configurando Apache Tomcat.
11. Entendiendo y comprendiendo los “Web Containers”.
12. Entendiendo y comprendiendo los “Application Servers”.
13. Aplicaciones Front-end.
14. Patrones para el desarrollo de aplicaciones Web.
15. Capa de presentación.
 - > Patrón Front Controller.
 - > Patrón View Helper.
 - > Patrón Composite View.
 - > Patrón Service to Worker.
 - > Patrón Dispatcher View.
16. Aplicaciones Web.
 - > Componentes de una aplicación Web.
 - > Ciclo de vida de una aplicación Web.
 - > Tipos de aplicaciones WEB.
 - > ¿Programación o diseño?
 - > Propósito de las aplicaciones Web.
 - > Organización de una aplicación Web en Java.
 - >> Estructura de directorios y archivos.
 - >> Tipos de archivos.
 - > Portabilidad.

3
Web.

- > Separación de capas.
- > Soporte a Base de Datos.
- > Implementado la lógica de negocio.
- > Implementado la capa de presentación.
- 17. Empaquetando nuestra aplicación.
- 18. Ant, Maven. Maven 2.
 - > Archivos JAR. Anatomía.
 - > Archivos WAR. Anatomía.
 - > Archivos EAR. Anatomía
- 19. Deployando nuestra aplicación.
 - > Hot Deploy con IDE Eclipse.
- 20. Herramientas para el desarrollo Web.
- 21. Construyendo una aplicación Web.
 - > Debugueando remotamente nuestra aplicación.
- 22. El archivo WEB.xml.
 - > Estructura.
 - > Mapeando Servlets.
 - > Definiendo filtros.
 - > Especificando páginas de inicio.
 - > Manejo de páginas de error.
 - > Seguridad.
 - >> Autenticación
 - >> Configurando Weblogic para usar SSL.
 - >> Certificados.
- 23. Utilerías.
 - > Log4j.
 - > Las Apache Commons.
 - > Frameworks javascript.
 - >> JQuery.
 - >> MooTools.
 - >> Ext JS.
 - >> Java Script del lado del cliente.
 - >> node.js
 - >> Java Script del lado del servidor.
 - >> Rhino. Interprete de JS para Java.
- 24. ¿Qué es un Servlet?
- 25. ¿Están muertos los CGIs?
- 26. Especificación Servlets.
 - > Especificación 2.5.
 - > Especificación 3.0.
- 27. Arquitectura de los Servlets.
- 28. El paquete javax.servlet.
- 29. Peticiones y Respuestas.
- 30. Estructura de un Servlet.
 - > La Clase GenericServlet.
 - > La Clase HttpServlet.
 - > La Interface HttpServletRequest.
 - > La Interface HttpServletResponse.
- 31. Ventajas y desventajas de los servlets.
- 32. Ciclo de vida de un Servlet.
 - > El Método service().
 - > El Método doGet().
 - > El Método doPost().
 - > El Método init().
 - > Tipo de inicialización.
 - > Método destroy().

3
Web.

33. ¿Cuándo y por qué usar Servlets?
34. Configuración de Servlets.
 - > La interface ServletConfig.
 - > La interface ServletContext.
35. Compilando, instalando e invocando un Servlet.
36. Generando HTML y CSS.
 - > Fundamentos de HTML y CSS
 - > Corrigiendo vicios.
 - > ¿XHTML y HTML 5?
37. La interface SingleThreadModel.
38. La interface RequestDispatcher.
39. La carpeta WEB-INF
 - > La carpeta classes.
 - > La carpeta lib.
 - > Descriptor o Web.xml.
 - >> Etiquetas y elementos.
40. Filtros, Eventos y Listeners.
41. Manejo de peticiones desde formulario.
 - > Repoblado de datos.
42. Encabezados HTTP Request.
43. Formato HTTP Response.
 - > Códigos de estado.
 - > Encabezados.
44. Session Tracking.
45. Manejo de cookies.
 - > Galletas mal horneadas.
46. Thread Safe Servlets.
47. Desarrollando aplicaciones Web seguras.
48. Administración de sesiones.
49. Java Server Pages.
50. Java Server Pages y XML.
51. Arquitectura de las JSPs.
 - > Modelo 1 – Arquitectura.
 - > Modelo 2 – Arquitectura.
52. Ventajas y desventajas.
 - > Ciclo de vida de una JSP.
53. Ámbito de las JSPs.
54. Modelos de programación.
 - > “Simple model”.
 - > “Forward delegation model”.
 - > “Include delegation model”.
 - > “Decoupled model”.
55. Delegando funciones:
 - > El modelo.
 - > La vista.
 - > El controlador.
56. ¿Cuándo y por qué usar JSPs?
57. Diferencias entre Servlets y JSPs.
58. Sintaxis y Semántica.
 - > Declaraciones.
 - > Directivas.
 - > Expresiones.
 - > Comentarios.
 - > Los JSP scriptlets.
59. Atributos.
 - > Atributo import.

3
Web.

- > Atributo contentType.
- > Atributo isThreadSafe.
- > Atributo session.
- > Atributo buffer.
- > Atributo autoflush.
- > Atributo extends.
- > Atributo info.
- > Atributo errorPage.
- > Atributo isErrorPage.
- > Atributo language.
- 60. Lenguaje de expresión unificado.
- 61. Arquitectura java beans.
 - >Enfrijoladas.
 - >¿Qué son los java beans?
 - > Propiedades de un bean.
 - > Creando un java bean.
 - > Usos de los java beans.
 - > java beans Vs EJBs.
- 62. Patrón de diseño.
 - > Transfer Object.
- 63. Patrón de diseño.
 - >Factory.
 - > Bean Factory
- 64. Contexto de una aplicación Web.
- 65. Errores y excepciones
 - > Servlets y JSPs.
 - > Manipulando Errores en las JAVA Server Pages.
 - >> “Forwarding Errors”.
 - > Incluyendo otros archivos en la generación de Java Server Pages.
 - > Incluyendo archivos en tiempo de compilación.
 - > Incluyendo archivos en tiempo de ejecución.
 - > Procesando formularios de datos.
 - > Control de navegación con JAVA Server Pages.
 - > Estado de mantenimiento.
 - > Uso de mejores prácticas en el desarrollo de JAVA Server Pages.
- 66. Administrador de sesiones y subprocesos.
 - > Seguimiento de sesiones.
 - > La API de sesión.
 - > Administración de subprocesos.
 - > Modelos de subproceso de Servlets.
 - > Aplicaciones multihilo.
 - > HTTP y estados de sesión.
 - > Estado de persistencia.
- 67. ¿Qué es JSTL?
 - > c:out
 - > c:forEach
 - > c:forTokens
 - > c:if
 - > c:choose
 - > c:set
 - > c:remove
 - > c:import
 - > c:url
 - > c:param
 - > c:redirect
 - > c:catch

3
Web.

- 68. JSTL y EL.
- 69. JSTL vs Scriptlets.
- 70. Tags personalizados.
- 71. JSP Vs. JSE
- 72. Introducción JAVA Server Faces.
 - > JSF Vs. JSP
 - > JSTL Vs. JSP
- 73. Conectividad con base de datos
- 74. Inyección de recursos.
- 75. Patrones de diseño para JDBC.
- 76. El Patrón de diseño DAO Factory.
- 77. El Patrón DAO.
 - > BOs y DAOs
 - > Persistencia transparente.
 - > Pequeñas cuestiones.
- 78. SQL Scripts.
- 79. Transacciones.
 - > Métodos de transacciones.
 - > Control de concurrencia.
- 80. DataSource y JNDI API.
- 81. Pool de Conexiones
- 82. Problemas con JDBC.
- 83. JSPs y XML
- 84. ¿Qué es XML?
 - > sax.
 - > stax.
 - > jdom4j.
- 85. ¿Es el fin del XML?
- 86. Ventajas y desventajas del uso del XML.
- 87. JSON y XML.
- 88. Definición de documentos XML.
 - > XML bien formado.
 - > XML Sintácticamente correcto.
 - > Descriptores de datos.
 - > Schemas.
- 89. JSON y AJAX.
- 90. Validaciones.
- 91. XSL y SOAP
- 92. XSLT.
- 93. Otros Frameworks.
- 94. Lógica de negocio.
- 95. Patrones de diseño.
 - > Capa de negocio.
 - >> ServiceLocator.
 - >> Business Delegate.
 - >> Session Facade.
- 96. Spring.
 - > Inversión de Control.
 - > Inyección de dependencias.
 - >> ¿Una noche de primavera?
 - >> ¿Qué es Spring y para qué sirve?
 - >>> Filosofía.
 - >>> Tecnología no invasiva.
 - >>> Configuración.
 - > Cronología.
 - >> Versiones de Spring.

3
Web.

- >> Lo nuevo en Spring 3.0. y 4.0.
- >> Spring Expression Language (SpEL).
- >> Componentes basados en anotaciones.
- >> Características de Spring.
- >> ¿Qué es un POJO?
- >> Programación orientada a aspectos.
- >> POA con Spring.
- >> AspectJ.
- >> ¿Alternativas?
 - >>> EJBs Vs. Spring.
 - >>> Frameworks.
 - >>> Patrones de diseño.
 - >>> Lenguajes dinámicos.
- 97. Arquitectura de aplicaciones con Spring.
 - > Arquitectura Spring
 - >> Spring Core.
 - >> Spring DAO.
 - >> Spring ORM.
 - >> Spring MVC.
 - >>> Beneficios Spring Web MVC.
 - >>> Spring Expression Language (SpEL).
- 98. Configuración de Contenedor.
 - > Configurando Spring en Eclipse IDE.
 - > Configuración de Archivos Spring.
 - >> applicationContext.xml.
- 99. Creación y Configuración de Beans.
- 100. Bean Factory.
 - > Ciclo de vida.
- 101. Instanciación.
- 102. Spring Bean Factory.
 - > XmlBeanFactory.
- 103. ¿Cuándo usar Bean Factory?
- 104. ¿Cableando frijoles?
- 105. Patrón de Inyección de dependencia en Spring.
- 106. Dependencias.
 - > Tipos de inyección.
 - >> Inyección por constructor.
 - >> Inyección por "setter".
 - >> Inyección por constructor Vs "setter".
 - >> Inyección de referencias de objetos.
 - > Declarando beans.
 - > Propiedades de los beans.
 - > Configuración.
 - >> Referencia a otros beans.
 - >> Colecciones.
 - >>> Listas y Arreglos.
 - >>> Sets.
 - >>> Maps.
 - >>> Properties.
 - >> Beans internos.
 - >> Valores nulos.
 - > Configuración por anotaciones.
 - >> @Autowired.
 - >> @Inject.
 - >> @Resource.
 - >> @PostConstruct.

3
Web.

- >> @PreDestroy.
- 107. Auto cableado en Spring.
 - > Tipos.
 - >> byName.
 - >> byType.
 - >> constructor.
 - >> autodetect.
 - >> por defecto.
- 108. Hasta donde llegan los frijoles (Scope).
 - > Singleton.
 - > Prototype.
 - > Request.
 - > Session.
 - > Global-Session.
 - > Retrollamadas (Callbacks).
 - >> Ciclo de vida.
 - > Inicialización.
 - > Destrucción.
- 109. Inicialización de beans.
 - > Interfaces.
 - > ID.
- 110. Beans heredados.
- 111. Beans abstractos.
- 112. Inyección de métodos.
 - > Interfaz MethodReplacer.
 - > Inyección "getter".
- 113. Propiedades personalizadas.
 - > Interfaz PropertyEditor.
- 114. El contenedor ApplicationContext.
 - > Ciclo de vida.
 - > ClassPathXmlApplicationContext.
 - > FileSystemXmlApplicationContext.
 - > XmlWebApplicationContext.
 - >> Eventos. (Application Events).
 - >> Genericos.
- 115. ApplicationContext Vs. Bean Factory.
- 116. Patrones de Diseño y Spring.
 - > Capa de presentación.
 - >> Front Controller.
 - >> Application Controller.
 - >> Page Controller.
 - >> Context Object.
- 117. Spring Tag Library.
- 118. Spring MVC.
 - > Introducción.
 - > Arquitectura General.
 - > Paradigma MVC.
 - >> Modelo.
 - >> Vista.
 - >> Controlador.
 - > AbstractController.
 - > DispatcherServlet.
 - > FrontController.
 - >> WebApplicationContext.
 - > Controladores y Vistas.
 - > Handler Mappings.

3
Web.

- >> BeanNameUrlHandlerMapping.
- >> SimpleUrlHandlerMapping.
- >> CommonsPathMapUrlHandlerMapping.
- > Handling Requests.
 - >> HandlerExecutionChain.
 - >> Interceptores.
 - >> WebContentInterceptor.
- 119. Interceptores.
- 120. Transacciones con Spring.
 - > Tipos de transacciones.
 - > Anotación @Transactional.
 - > Propagación de transacciones.
- 121. Introducción a Spring JDBC Framework.
- 122. Hibernate y Spring.
- 123. ¿Qué es Hibernate?
- 124. Shampoo para POJOs.
- 125. ¿Por qué Hibernate?
- 126. Herramientas, instalación y configuración.
 - > Archivo de configuración Hibernate.
 - > JPA configuración.
- 127. Entendiendo la persistencia de objeto/relación.
- 128. ¿Qué significa persistir?
- 129. Capas de persistencia.
- 130. Persistiendo múltiples objetos.
- 131. ORM: "Object Relational Mapping".
- 132. Mapeo de documentos.
- 133. Mi primera clase persistente.
- 134. Modelos de domino y "metadata".
- 135. Anotaciones con Hibernate.
 - > @Entity.
 - > @Table.
 - > @Id.
 - > @GeneratedValue.
 - > @Basic.
 - > @Lob.
 - > @Transient.
 - > @Column.
- 136. Mapeo o "Mapping".
 - > Asociaciones.
 - >> one-to-one.
 - >> many to one.
 - >> one-to-many.
 - >> many-to-many.
- 137. Hibernate y Spring.
- 138. Web Services: SOA / REST. Implementando Web Services.
- 139. Validaciones y pruebas unitarias.
- 140. Spring Security.
- 141. Monitoreo y Controladores.
- 142. JSFs vs JSPs.

4
Arquitectura.

1. Arquitectura de software.
 - > Buena y mala arquitectura.
 - > Alta disponibilidad.
 - > ¿Licuados de metodologías de desarrollo?
 - > Análisis de requerimientos.
 - > Escalabilidad.
 - > Análisis de los procesos de negocio.
 - > Diseño y definición de soluciones.
 - > Construcción de arquitecturas base.
 - > Implementación de casos de uso.
 - > Diseño y ejecución de pruebas (unitarias, integración, prueba de stress, etc.)
 - > Administración de la configuración de un proyecto.
 - > ¿Cómo ser un buen líder de proyecto?
2. Latencia.
3. Throughput.
4. Alta disponibilidad.
5. Balanceo de carga.
6. Patrones de arquitectura.
7. ¿Por qué las arquitecturas son importantes?
8. Atributos de calidad.
9. Java Management Extensions (JMX).
10. Hibernate 3 y 4.
 - > Cache de Segundo Nivel con Hibernate.
 - > Patrones de sesión con Hibernate.
 - > ORM Avanzado.
 - >> Objetos embebidos.
 - >> Llaves compuestas.
 - >> Elementos de Mapeo Avanzados.
 - >> Tablas Múltiples.
11. JMS.
 - > Estilos de mensaje soportados por JMS.
 - > Entendiendo y comprendiendo JMS.
 - > Programas punto a punto.
 - > Publicador suscriptor.
 - > Productores.
 - > Consumidores.
 - > Transacciones Distribuidas.
 - > Mensajería.
 - >> Síncrona.
 - >> Asíncrona.
 - >> Concurrente.
 - > JMS y Multi-threading.
 - > Seguridad y JMS.
 - > La interface Message.
12. Spring Vs. EJBs.
13. Spring y EJBs.
14. Enterprise Java Beans.
15. Los EJBs. ¿Semillas poderosas o engorrosas?
 - > EJB 3.
 - >> Consideraciones.
 - > Versiones de EJBs y estado de uso.
16. Arquitectura de los Enterprise Java Beans.
 - > Componentes de una arquitectura EJB.
 - > Características de los EJBs.
17. Tipos de EJBs.
 - > Session beans.

4
Arquitectura.

- >> Stateful Session Beans.
- >> Stateless Session Beans.
- >> Singleton Session Beans.
- > Message-driven beans.
- > Entities.
- > Java Persistence API.
- 18. Roles y responsabilidades: ¿Quién se encarga de qué?
 - > El Enterprise Bean Provider.
 - > El Application Assembler.
 - > El EJB Server Provider.
 - > El EJB Container Provider.
 - > El EJB Deployer.
 - > El System Administrator.
- 19. JNDI.
 - > Tipos de Clientes JEE.
 - >> Clientes Web.
 - >> Clientes EJB.
 - >> Clientes EIS.
- 20. ¿Cuándo usar JNDI?
- 21. JNDI Tree.
- 22. JNDI Initial Context.
- 23. Spring HTTP Invoker, Hessian, Burlap.
- 24. WebSockets.
- 25. Anotaciones Vs. Descriptores.
 - > Ventajas y desventajas.
- 26. Nuestro primer EJB.
 - > La anotación @EJB.
 - > Desarrollando un Session Bean: Encapsulando la lógica de negocio.
 - >> Anatomía de un Session Bean.
 - >>> Tipos de Session Bean.
 - >>>> ¿Cuándo usar session bean?
 - >>> Tiempo de vida de un session bean.
- 27. Creando un cliente Session Bean.
 - > Stateless Session Bean.
 - >> Anotación @Stateless.
 - >> Creación de un Stateless Session Bean.
 - >> Interfaces locales e interfaces remotas.
 - >> Anotaciones: Interface de negocio:
 - >>> @Remote.
 - >>> @Local.
 - >> ¿Y El archivo ejb-jar.xml?
 - >> Inyección de dependencias.
 - >> Métodos CallBack.
 - >>> @PostConstruct.
 - >>> @PreDestroy.
 - >> Interceptores en EJBs.
 - >>> Conceptos AOP.
 - >>> @AroundInvoke.
 - >>> @Interceptors.
 - >>> @Interceptor.
 - > Stateful Session Bean.
 - >> ¿Qué es un Stateful Session Bean?
 - >> Creación de un Stateful Session Bean.
 - >> Anotación @Stateful.
 - >> Anotaciones: Interface de negocio.
 - >>> @Remote.

4
Arquitectura.

- >>> @Local.
- >> Deployment de un "Stateful Session Bean".
- >> Mecanismos de "Swapping"
- >> Consideraciones de desempeño.
- >> Métodos CallBack.
- >>> @PostConstruct.
- >>> @PreDestroy.
- >>> @PrePassivate.
- >>> @PostActivate.
- >> Destruyendo un stateful session bean.
- >> Diferencias entre stateless y stateful session beans.
- >> Deployando los session beans.
- >> Compilando y ejecutando nuestros clientes.
- >> Entities o entities beans.
 - >>> EntityManager Interface.
- >> ¿Qué es un Entity?
- >> Entities Vs. Session Beans.
- >> La anotación @Entity.
- >> El archivo persistence.xml.
- >> ORM.
 - >>> Anotaciones de persistencia.
 - >>> Estados.
 - >>> Mapeos.
 - >>>> Tipos.
 - >>>> Relaciones.
 - >>>> Entity Manager.
 - >>>> Hibernate y EJBs.
- >> Creando un singleton bean.
- >> Usando múltiples singleton beans.
- >> Administración de concurrencias.
- >> La anotación @ConcurrencyManagement.
- >> Accesando un EJB desde una JSP.
- 28. Invocación de EJBs usando JNDI.
- 29. Transacciones en JEE.
- 30. ¿Qué es una transacción?
 - > API JTA.
 - > Transacciones distribuidas.
 - > Core ACID.
 - >> Atomicidad.
 - >> Consistencia.
 - >> Aislamiento.
 - >> Durabilidad.
- 31. Actores de la transacción en el contenedor.
- 32. Diferenciadores de los EJBs.
- 33. Propagación de Transacciones.
- 34. Propiedades Transaccionales.
- 35. Manejo de la Transacción de manera sencilla.
- 36. Anotaciones:
 - > @TransactionManagement.
 - > @TransactionAttribute.
 - > Atributos de interacción:
 - >> REQUIRED
 - >> REQUIRES_NEW
 - >> SUPPORTS
 - >> MANDATORY
 - >> NOT_SUPPORTED

4
Arquitectura.

- >> NEVER
- 37. Manejo de errores.
 - > Excepciones de sistema.
 - > Excepciones de aplicación.
 - > @ApplicationException.
- 38. Interface SessionSynchronization con session beans.
- 39. Tipos de arquitectura.
- 40. ¿Por qué es importante la arquitectura?
- 41. Arquitecturas orientadas a atributos de calidad.
- 42. El futuro de la comunicación Cliente Servidor.
- 43. El rol del arquitecto de software.
 - > Delegando funciones.
- 44. Metodologías en el desarrollo de Software.
- 45. Más patrones de Diseño.
- 46. Java Testing y diseño.
 - > Criterios para el buen desempeño de una aplicación WEB.
- 47. Integración de aplicaciones.
 - > ESBs.
 - > Web Services.
 - > Message Driver bean y JMS.
- 48. Refactoring de aplicaciones.
- 49. Aplicaciones legadas.
- 50. Catálogo de Patrones JEE.
 - > Malas prácticas en las capa de presentación.
 - > Malas prácticas en la capa de negocio.
- 51. Seguridad en JEE y Weblogic Server.
- 52. Amenazas de Seguridad.
- 53. Soluciones de JEE.
- 54. Autenticación.
 - > SSO.
- 55. Escenarios de Autenticación.
- 56. Autenticación de la Capa Web.
- 57. Autenticación de EJB.
- 58. Autorización.
- 59. Mecanismos de Integridad.
- 60. Mecanismos de Encriptamiento.
- 61. Mecanismos de Auditoria.
- 62. WLS Security Realms.
- 63. Técnicas de performance.
- 64. Diseño y desempeño de las aplicaciones empresariales.
- 65. Pools en bases de datos
 - > Performance.
- 66. Herramientas para medir el desempeño de una aplicación
- 67. Análisis
- 68. Caché y de Servidores de aplicaciones.
- 69. Spring y EJBs.
- 70. Arquitectura EJBs.
- 71. Contexto e Inyección de Dependencias en JEE.
- 72. Spring y EJBs.
- 73. Implementando EJBs con Spring.
- 74. Web Services.
- 75. ¿Qué es un Web Service?
- 76. Web Services.
 - > Arquitectura de los Web Services.
 - > Servlets y JSPs en Web Services.
 - > Esquema A2A – Application to application.

4
Arquitectura.

- > Esquema B2B – Business to business.
- > Web Services APIs.
- 77. Tipos de Web Services.
- 78. Ventajas y desventajas de los Web Services.
- 79. RESTful Services.
- 80. JAX-WS 2.0 Web Services.
- 81. Alternativas Web Services.
 - > Axis 2.
 - > Spring WS.
- 82. Creación, Publicación y Testing.
- 83. Session Tracking en Web Services.
- 84. Web Services y bases de datos.
- 85. Arquitectura orientada a servicios (SOA).
 - >SOA con XML.
- 86. SOAP y WSDL.
 - > Anotación @WebResult.
 - > Estructura de SOAP.
 - >> SOAP Namespaces.
 - >> SOAP Headers.
 - >> SOAP Body.
 - > Apache SOAP
 - > Estructura de WSDL.
 - >> Elementos: “definitions”, “types” y “import”.
 - >> WSDL y JAVA.
 - >> WSDL Abstract Interface.
 - >> Patrones de intercambio de mensajes.
- 87. UDDI.
 - > Estructura: “businessEntity”.
 - > Estructura: “businessService”.
 - > Estructura: “binding Template”.
- 88. Web Services con Spring.
- 89. Spring-WS.
- 90. Servidor y Cliente.
- 91. Web.xml.
- 92. Accesando un EJB desde un Web Service (JAX-WS)
- 93. Accesando un EJB desde un Web Service (JAX-RS)
- 94. Administración de aplicaciones.
- 95. Guía de seguridad para aplicaciones Web.
- 96. Comida de fin de diplomado.