

TRABAJO DE CAMPO I

CÓDIGO DE MATERIA 26

2015

Titular: Claudia Pons

FACULTAD DE TECNOLOGÍA INFORMÁTICA

UNIVERSIDAD ABIERTA INTERAMERICANA

INDICE

GUÍA DE TRABAJOS REVISIÓN CONCEPTUAL	3
GUÍA DE TRABAJOS PRÁCTICOS	15
GUÍA DE ABORDAJE BIBLIOGRÁFICO.....	20

GUÍA DE TRABAJOS REVISIÓN CONCEPTUAL

UNIDAD I

1. Describa brevemente cómo ha sido el papel evolutivo del software.
2. Enumere y explique cuáles son las cosas que hacen que el software sea desde la óptica de los productos distinto a los demás.
3. Enumere las categorías más importantes de software de computadoras que existen en la actualidad y presentan a los desarrolladores retos continuos debido a sus continuos cambios.
4. ¿Qué es la computación ubicua?
5. Mencione al menos cuatro motivos por los cuales el software heredado debe evolucionar.
6. Enumere y explique al menos diez mitos relacionados con el desarrollo del software.
7. Dentro del marco de trabajo para el proceso de desarrollo de software enumere y explique cuáles son las acciones de ingeniería de software son altamente distintivas.
8. Dentro del marco de trabajo para el proceso de desarrollo de software a qué denominamos actividades sombrillas.
9. ¿Qué son los patrones de procesos?
10. ¿Qué son los modelos prescriptivos para el desarrollo de software?
11. ¿Qué caracteriza al modelo en cascada?
12. ¿Qué caracteriza los modelos incrementales?
13. Enumere y describa los modelos incrementales.
14. ¿Qué caracteriza los modelos evolutivos?
15. Enumere y describa los modelos evolutivos.
16. ¿Qué caracteriza los modelos especializados de proceso?
17. Enumere y describa los modelos especializados de proceso.
18. ¿Qué caracteriza al proceso unificado?
19. ¿Qué facetas posee el proceso unificado?

20. ¿Qué disciplinas posee el proceso unificado?
21. ¿Qué son y qué características poseen las metodologías ágiles?
22. Enumere y explique las características que debe tener el equipo de gente que participa en un desarrollo ágil.
23. ¿Qué es y qué características posee la programación extrema?
24. ¿Qué es y qué características posee el desarrollo adaptativo de software?
25. ¿Qué es y qué características posee el método de desarrollo de sistemas dinámicos?
26. ¿Qué es y qué características posee el desarrollo conducido por características?
27. ¿Qué se entiende al expresar modelo de desarrollo ágil?
28. ¿A qué procesos puede aplicarse el concepto de agilidad?
- ¿Cuáles son los beneficios de aplicar un proceso ágil en relación costo-cambio?
29. ¿Cómo hace para lograr una adaptación incremental?
30. Enumere y describa los principios de la agilidad.
31. Enumere y explique las características que debe tener el equipo de gente que participa en un desarrollo ágil.
32. Defina los valores para programación extrema (XP).
33. ¿Qué actividades posee el proceso de la programación extrema? Describir
34. ¿Cuáles son las prácticas que incorpora la XP industrial?
35. ¿Cuáles son las críticas de la XP?
36. ¿Qué caracteriza al desarrollo adaptativo de software (DAS)?
37. En qué hace énfasis el DAS.
38. ¿Qué caracteriza al SCRUM?
39. Explique el flujo de proceso del SCRUM y el grupo de acciones que define.
40. ¿Qué caracteriza el método de desarrollo basado en sistemas (MDSD)?
41. Describa el ciclo de vida MDSD.

42. ¿Qué características posee el desarrollo impulsado por las características (DIC)?

43. ¿Qué principios posee el desarrollo esbelto de software (DES)?

44. ¿Qué es el modelo ágil (MA)?

45. ¿Cuáles son los principios de modelado que sugiere el MA?

46. ¿Qué características posee el proceso unificado?

47. ¿Cuáles son las actividades que aborda cada iteración (PUA)?

UNIDAD II

48. ¿Qué consideraciones tendría al diseñar y desarrollar un módulo de seguridad?

49. ¿Qué estrategias se pueden adoptar para lograr un módulo de log in /Log out efectivo y eficiente?

50. ¿A qué se denomina gestión de perfiles?

51. Enumere os elementos y funcionalidades que debe contemplar un módulo para gestionar perfiles de usuarios.

52. ¿Qué es una bitácora de seguimiento?

53. ¿Qué diferencia existe entre una bitácora de seguimiento y una de auditoría?

54. ¿Para qué se utiliza el concepto de parametrización en un SI?

55. ¿Qué elementos parametrizaría en un SI?

56. ¿Cuál es el objetivo de la encriptación?

57. ¿Cuáles son los algoritmos más conocidos de encriptación?

58. ¿A qué nos referimos cuando hablemos de integridad de los datos en un SI?

59. Diseñe y prototipe una estrategia para lograr que un SI pueda administrar múltiples idiomas

60. ¿Qué es y para que sirve un backup?

61. ¿Qué es una política de backup?

- 62. Describa una política de backup que resulte útil para mantener datos de un SI que se estima imprescindible contar con los datos del último año.
- 63. ¿Qué consideraciones se deben tener para crear código seguro en las aplicaciones?
- 64. ¿Qué es un mapa de navegación?
- 65. ¿Qué consideraciones debería tener para generar interfaces de usuarios?
- 66. ¿A qué se denomina administración de errores?
- 67. Defina una estrategia para administrar los errores de un SI.

UNIDAD III

- 68. Defina medida, métrica e indicador.
- 69. ¿Cuáles son los retos de las métricas del producto?
- 70. Enumere y explique los principios de la medición
- 71. ¿Qué consideraciones tendría presentes al momento de medir software orientado a objetos?
- 72. ¿Qué atributos poseen las métricas efectivas del software?
- 73. Enumere las métricas para el modelo de análisis.
- 74. Explique cómo funcionan las métricas basadas en la función.
- 75. ¿Cómo puede medir la calidad de las especificaciones?
- 76. Enumere las métricas para el modelo de diseño.
- 77. ¿Cómo puede medir la calidad del diseño arquitectónico?
- 78. ¿Cómo puede medir la calidad del diseño orientado a objetos?
- 79. ¿Cuáles son las métricas orientadas a las clases y qué proponen?
- 80. ¿Cuáles son las métricas orientadas a los objetos y qué proponen?
- 81. ¿Cuáles son las métricas para el código fuente y qué proponen?
- 82. Enumere las métricas para la prueba.

- 83.¿Cuáles son las métricas tradicionales para la prueba y qué proponen?
- 84.¿Cuáles son las métricas para pruebas orientadas a objetos y qué proponen?
- 85.¿Cuáles son las métricas para el mantenimiento y qué proponen?
- 86.¿Cuáles son las métricas para el proceso de desarrollo de software y qué proponen?
- 87.¿Cuáles son las métricas del proyecto y qué proponen?
- 88.¿Cuáles son las métricas orientadas al tamaño y qué proponen?
- 89.¿Cuáles son las métricas orientadas a la función y qué proponen?
- 90.¿Qué observaría para medir la calidad?
- 91.¿Qué es la eficacia en la eliminación de defectos?
- 92.¿Qué significa establecer una línea referencia y qué beneficios proporciona?
- 93.¿Qué consideraciones tendría si desea aplicar métricas en organizaciones pequeñas?
- 94.¿Cuáles son las métricas orientada a Casos de Uso y qué proponen?
- 95. ¿Cuáles son los indicadores para la medición de Calidad?

UNIDAD IV

- 96.¿A qué denominamos gestión del riesgo?
- 97.¿Qué es la incertidumbre en el contexto de administración del riesgo?
- 98.¿Qué es la pérdida en el contexto de administración del riesgo?
- 99.¿Cómo se pueden identificar potenciales riesgos?
- 100. ¿Cómo se pueden realizar una evaluación del riesgo global del proyecto?
- 101. ¿Cómo se puede proyectar el riesgo?
- 102. ¿Qué elementos debe contener una tabla de evaluación de riesgos?
- 103. ¿Cómo se puede evaluar el impacto del riesgo?
- 104. ¿Qué es y qué contiene un plan de reducción, supervisión y gestión del riesgo?

105. ¿Qué significa gestionar la calidad?
106. Defina control de calidad
107. Defina garantía de calidad
108. ¿Qué es el costo de la calidad?
109. ¿Qué actividades se desarrollan en el contexto del aseguramiento de la calidad?
110. ¿Qué diferencia existe entre un error y un defecto?
111. ¿Qué son las revisiones técnicas formales?
112. ¿Cuáles son las directrices a considerar en las revisiones técnicas formales?
113. ¿Qué es para la ingeniería de software “Seis Sigma”?
114. ¿Qué expresa la medida de fiabilidad?
115. ¿Qué expresa la medida de disponibilidad?
116. ¿Qué es el modelo de madurez de las capacidades?
117. ¿Qué niveles comprende el modelo de madurez de las capacidades y qué implica cada uno de ellos?
118. ¿Cómo se puede definir a la calidad del software?
119. ¿Cuáles son los factores de calidad que propone Mc Call?
120. ¿Cuáles son los factores de calidad que propone la ISO 9126?

UNIDAD V

121. ¿Qué es un patrón en el contexto de desarrollo de software?
122. ¿Cómo nace el concepto de patrón?
123. ¿Qué es un patrón de diseño?
124. ¿De qué manera un patrón de diseño brinda una solución a un problema?
125. ¿Qué debería tener en cuenta para seleccionar un patrón de diseño?
126. ¿Cuáles son los pasos que seguiría para usar correctamente un patrón?

127. ¿Cuáles son los cuatro elementos iniciales de todo patrón?
128. ¿Qué elementos utilizaría en la descripción de un patrón?
129. ¿Cómo cataloga Gamma a los patrones de diseño?
130. ¿A qué denominamos anti patrón?
131. ¿Por qué en la utilización de patrones se alienta el uso de herencia de interfaces?
132. ¿En qué casos preferiría componer a heredar?
133. ¿A qué hace referencia el concepto de diseñar para el cambio?
134. ¿Cuáles son los patrones de diseño considerados de creación?
135. ¿Cuál es el propósito del patrón Abstract Factory?
136. ¿Se conoce con algún otro nombre al patrón Abstract Factory?
137. ¿En qué casos es aplicable el patrón se conoce Abstract Factory?
138. ¿Cuáles son las clases participantes en el patrón Abstract Factory?
139. ¿Posee colaboradores el patrón Abstract Factory?
140. ¿Qué consecuencias trae aparejadas utilizar el patrón Abstract Factory?
141. ¿Con qué patrones de relaciona el patrón Abstract Factory?
142. ¿Cuál es el propósito del patrón Factory Method?
143. ¿Se conoce con algún otro nombre al patrón Factory Method?
144. ¿En qué casos es aplicable el patrón se conoce Factory Method?
145. ¿Cuáles son las clases participantes en el patrón Factory Method?
146. ¿Posee colaboradores el patrón Factory Method?
147. ¿Qué consecuencias trae aparejadas utilizar el patrón Factory Method?
148. ¿Con qué patrones de relaciona el patrón Factory Method?
149. ¿Cuál es el propósito del patrón Singleton?
150. ¿Se conoce con algún otro nombre al patrón Singleton?
151. ¿En qué casos es aplicable el patrón se conoce Singleton?

152. ¿Cuáles son las clases participantes en el patrón Singleton?
153. ¿Posee colaboradores el patrón Singleton?
154. ¿Qué consecuencias trae aparejadas utilizar el patrón Singleton?
155. ¿Con qué patrones de relaciona el patrón Singleton?
156. ¿Cuáles son los patrones de diseño estructurales?
157. ¿Cuál es el propósito del patrón Adapter?
158. ¿Se conoce con algún otro nombre al patrón Adapter?
159. ¿En qué casos es aplicable el patrón se conoce Adapter?
160. ¿Cuáles son las clases participantes en el patrón Adapter?
161. ¿Posee colaboradores el patrón Adapter?
162. ¿Qué consecuencias trae aparejadas utilizar el patrón Adapter?
163. ¿Con qué patrones de relaciona el patrón Adapter?
164. ¿Cuál es el propósito del patrón Composite?
165. ¿Se conoce con algún otro nombre al patrón Composite?
166. ¿En qué casos es aplicable el patrón se conoce Composite?
167. ¿Cuáles son las clases participantes en el patrón Composite?
168. ¿Posee colaboradores el patrón Composite?
169. ¿Qué consecuencias trae aparejadas utilizar el patrón Composite?
170. ¿Con qué patrones de relaciona el patrón Composite?
171. ¿Cuál es el propósito del patrón Decorator?
172. ¿Se conoce con algún otro nombre al patrón Decorator?
173. ¿En qué casos es aplicable el patrón se conoce Decorator?
174. ¿Cuáles son las clases participantes en el patrón Decorator?
175. ¿Posee colaboradores el patrón Decorator?
176. ¿Qué consecuencias trae aparejadas utilizar el patrón Decorator?

177. ¿Con qué patrones de relaciona el patrón Decorator?
178. ¿Cuáles son los patrones de diseño de comportamiento?
179. ¿Cuál es el propósito del patrón Interpreter?
180. ¿Se conoce con algún otro nombre al patrón Interpreter?
181. ¿En qué casos es aplicable el patrón se conoce Interpreter?
182. ¿Cuáles son las clases participantes en el patrón Interpreter?
183. ¿Posee colaboradores el patrón Interpreter?
184. ¿Qué consecuencias trae aparejadas utilizar el patrón Interpreter?
185. ¿Con qué patrones de relaciona el patrón Interpreter?
186. ¿Cuál es el propósito del patrón Iterator?
187. ¿Se conoce con algún otro nombre al patrón Iterator?
188. ¿En qué casos es aplicable el patrón se conoce Iterator?
189. ¿Cuáles son las clases participantes en el patrón Iterator?
190. ¿Posee colaboradores el patrón Iterator?
191. ¿Qué consecuencias trae aparejadas utilizar el patrón Iterator?
192. ¿Con qué patrones de relaciona el patrón Iterator?
193. ¿Cuál es el propósito del patrón Memento?
194. ¿Se conoce con algún otro nombre al patrón Memento?
195. ¿En qué casos es aplicable el patrón se conoce Memento?
196. ¿Cuáles son las clases participantes en el patrón Memento?
197. ¿Posee colaboradores el patrón Memento?
198. ¿Qué consecuencias trae aparejadas utilizar el patrón Memento?
199. ¿Con qué patrones de relaciona el patrón Memento?
200. ¿Cuál es el propósito del patrón Observer?
201. ¿Se conoce con algún otro nombre al patrón Observer?

- 202. ¿En qué casos es aplicable el patrón se conoce Observer?
- 203. ¿Cuáles son las clases participantes en el patrón Observer?
- 204. ¿Posee colaboradores el patrón Observer?
- 205. ¿Qué consecuencias trae aparejadas utilizar el patrón Observer?
- 206. ¿Con qué patrones de relaciona el patrón Observer?

UNIDAD VI

- 207. ¿Qué significa probar un software?
- 208. ¿Cuál es el objetivo de la prueba?
- 209. ¿Qué consideraciones tendría al conformar un equipo de prueba?
- 210. ¿Qué son las estrategias de prueba?
- 211. ¿Cuáles son las estrategias de prueba?
- 212. ¿Qué proponen las pruebas de unidad?
- 213. ¿Qué proponen las pruebas de integración?
- 214. ¿Qué proponen las pruebas de integración ascendente?
- 215. ¿Qué proponen las pruebas de descendente?
- 216. ¿Qué proponen las pruebas de regresión?
- 217. ¿Qué proponen las pruebas de humo?
- 218. ¿Qué elementos se deben documentar en las pruebas de integración?
- 219. ¿Qué proponen las pruebas de validación?
- 220. ¿Qué proponen las pruebas del sistema?
- 221. ¿Qué significa revisar la configuración?
- 222. ¿Qué son las pruebas alfa y betas?
- 223. ¿Qué son las pruebas de recuperación?
- 224. ¿Qué son las pruebas de seguridad?

- 225. ¿Qué son las pruebas de resistencia?
- 226. ¿Qué son las pruebas de desempeño?
- 227. ¿Qué significa depurar?
- 228. ¿Cuáles son las estrategias de depuración?
- 229. ¿Qué son las técnicas de prueba de software?
- 230. ¿Cuáles son las técnicas de prueba de software más conocidas?
- 231. Enumere y explique los fundamentos de las pruebas de software
- 232. ¿Cuáles son las características de la prueba?
- 233. ¿Qué proponen las pruebas de caja blanca?
- 234. ¿Qué proponen las pruebas de caja negra?
- 235. ¿Qué proponen las pruebas de la ruta básica?
- 236. Defina complejidad ciclomática.
- 237. ¿Cómo se pueden probar las estructuras de control?
- 238. ¿Cómo se pueden desarrollar las pruebas de condición?
- 239. ¿Cómo se pueden desarrollar las pruebas de flujo de datos?
- 240. ¿Cómo se pueden desarrollar las pruebas de bucles?
- 241. ¿Qué proponen los métodos gráficos de prueba?
- 242. ¿Qué proponen el método de prueba de partición equivalente?
- 243. ¿Qué proponen el análisis de valores límites?
- 244. ¿Qué proponen el método de prueba de tabla ortogonal?
- 245. ¿Qué consideraciones tendría al desarrollar pruebas en entornos especializados?
- 246. ¿Qué consideraciones tendría al desarrollar pruebas para sistemas en tiempo real?
- 247. ¿Qué son los patrones de prueba?
- 248. ¿Cuáles son las estrategias de prueba para software orientado a objetos y qué proponen?
- 249. Enumere los métodos de pruebas orientadas a objetos.

250. ¿Qué propone el método de prueba aleatoria para clases orientadas a objetos?
251. ¿Qué proponen el método de prueba basado en fallas?
252. ¿Qué proponen el método de prueba denominado casos de prueba y jerarquía de clases?
253. ¿Qué proponen las pruebas basadas en escenarios?
254. ¿Qué son los métodos de pruebas aplicables a nivel de clase?
255. ¿Qué propone el método de prueba de partición a nivel de clase?
256. ¿Qué propone el método de prueba de diseño de caso de prueba de interclase?
257. ¿Qué propone el método de pruebas derivadas de modelos de comportamiento?

GUÍA DE TRABAJOS PRÁCTICOS

UNIDAD I

1. Seleccione un sistema real y redacte una especificación técnica donde se puedan observar claramente su alcance y qué gestiones conforman al mismo.
2. Sobre el sistema seleccionado en el punto 1 y suponiendo un escenario donde hay que realizar cambios masivos, elabore una lista de aspectos a considerar para decidir entre las opciones de reconvertir o mantener al sistema. Resalte ventajas y desventajas de cada posibilidad.
3. Proponga un sistema y establezca su objetivo y alcance. Luego elabore una lista de tareas a desarrollar para lograr el objetivo. Organícelas de acuerdo al modelo lineal secuencial. Grafique cómo quedarían organizadas.
4. Proponga un sistema y establezca su objetivo y alcance. Luego elabore una lista de tareas a desarrollar para lograr el objetivo. Organícelas de acuerdo al modelo de prototipos. Grafique cómo quedarían organizadas.
5. Proponga un sistema y establezca su objetivo y alcance. Luego elabore una lista de tareas a desarrollar para lograr el objetivo. Organícelas de acuerdo al modelo DRA. Grafique cómo quedarían organizadas.
6. Proponga un sistema y establezca su objetivo y alcance. Luego elabore una lista de tareas a desarrollar para lograr el objetivo. Organícelas de acuerdo a cada modelo evolutivo... Grafique cómo quedarían organizadas y compárelas. ¿Qué conclusiones obtiene?
7. Proponga un sistema y establezca su objetivo y alcance. Luego elabore una lista de tareas a desarrollar para lograr el objetivo. Organícelas de acuerdo al proceso unificado. Grafique cómo quedarían organizadas.

UNIDAD II

8. Seleccione un software ampliamente conocido y elabore el mapa de navegación del mismo.
9. Genere un documento sobre usabilidad de los sistemas y las GUI. Detecte y resalte en una lista de cotejos todos los aspectos a considerar referidos a estos puntos en el desarrollo de software. Organícelos de acuerdo a su relevancia.
10. Genere un documento sobre las distintas políticas de Backup que existen. Compárelas y detecte ventajas y desventajas de cada una.
11. Genere un documento sobre las distintas formas de manejar las políticas de acceso a un sistema, roles de usuarios y perfiles. Compare las distintas formas y elabore un cuadro de ventajas y desventajas de cada una.

12. Desarrolle un prototipo que demuestre la forma en que trabaja una bitácora de seguimiento y una de auditoría.

UNIDAD III

13. Defina una lista de atributos para ser utilizados en la construcción de una métrica de calidad para un SI.
14. Tomando como base el punto 13 pondere los atributos seleccionados y desarrolle la planilla que permita aplicar la métrica.
15. Seleccione una especificación de caso de uso de su TP y sométala a la evaluación de tres compañeros de clase con el objetivo de establecer el nivel de ambigüedad que posee. (previamente establezca que elementos intervendrán en la medición)
16. Seleccione la forma que considera más adecuada para medir la calidad de sus clases y aplique esto a las clases de su TP. ¿Qué resultados obtuvo?
17. Tome el diagrama de clases de su carpeta de proyecto y calcule cual es el nivel de acoplamiento entre las clases.

UNIDAD IV

18. Tomando como base el sistema que Ud. Está desarrollando como trabajo práctico de la asignatura, elabore una tabla de exposición al riesgo considerando las amenazas que entiende lo puedan afectar.
19. De acuerdo a lo elaborado en el punto anterior genere un plan de reducción, supervisión y gestión del riesgo que resulte útil y adecuado a su proyecto.
20. Calcular la EDE sabiendo que tenemos 50 errores y 100 Defectos.
21. Calcular la EED sabiendo que tenemos 50 errores y 100 Defectos. Compare el resultado con el ejercicio anterior. ¿A qué conclusión llega?

UNIDAD V

22. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón Abstract Factory. 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 Factory Method. 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 Singleton. 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 Adapter. 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 Composite. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.
27. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón Decorator. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.
28. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón Interpreter. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.
29. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón Iterator. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.
30. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón Memento. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.
31. Elabore o seleccione una situación real donde el problema que plantea se resuelve con el patrón Observer. Documente la situación y deje claramente explicado cómo el patrón da solución al problema. Desarrolle el código y demuestre su funcionamiento.

UNIDAD VI

32. Dentro de un esquema de prueba de software calcular la complejidad ciclomática (cantidad mínima de pruebas que debo realizar para asegurarme que recorrí al menos una vez cada camino básico). El algoritmo solicitado debe permitir ingresar un número y determinar si es primo. En caso afirmativo o negativo mostrar una leyenda que lo acredite.
33. Dentro de un esquema de prueba de software calcular la complejidad ciclomática (cantidad mínima de pruebas que debo realizar para asegurarme que recorrí al menos una vez cada camino básico). El algoritmo solicitado debe permitir ingresar N número y mostrar aquellos que son primos.
34. Nos solicitan calcular la complejidad ciclomática de acuerdo a las tres formas conocidas, armar el lote de prueba necesario para recorrer al menos una vez cada camino básico y

graficar el diagrama de flujo utilizando la simbología de clase. El algoritmo debe permitir: ingresar números enteros mayores que 3 y menores que 20 de a uno por vez. Se deberá determinar si el factorial (calcular el factorial considerando que para todo número natural n , se llama n factorial o factorial de n al producto de todos los naturales desde 1 hasta n) del número ingresado es menor que 55 y múltiplo de 18. En caso de serlo mostrar una leyenda que así lo exprese, en caso contrario permitir el ingreso de otro número.

35. Dentro de un esquema de prueba de software calcular la complejidad ciclomática (cantidad mínima de pruebas que debo realizar para asegurarme que recorrí al menos una vez cada camino básico). El algoritmo solicitado debe permitir ingresar 4 números y retornar la suma de todos ellos.
36. Dentro de un esquema de prueba de software calcular la complejidad ciclomática (cantidad mínima de pruebas que debo realizar para asegurarme que recorrí al menos una vez cada camino básico). El algoritmo solicitado debe permitir ingresar 4 números al principio, almacenarlos y luego retornar la suma de todos ellos.
37. Dentro de un esquema de prueba de software calcular la complejidad ciclomática (cantidad mínima de pruebas que debo realizar para asegurarme que recorrí al menos una vez cada camino básico). El algoritmo solicitado debe permitir ingresar 4 números al principio, almacenarlos y luego retornar la suma de todos ellos.
38. Dentro de un esquema de prueba de software calcular la complejidad ciclomática (cantidad mínima de pruebas que debo realizar para asegurarme que recorrí al menos una vez cada camino básico). El algoritmo solicitado debe permitir ingresar 4 números al principio, almacenarlos, luego sumar los pares y mostrar los resultados. Funciones: División Entera. Retorna el resultado entero de una división de $x/2$. $DV(x)$.
39. Dentro de un esquema de prueba de software calcular la complejidad ciclomática (cantidad mínima de pruebas que debo realizar para asegurarme que recorrí al menos una vez cada camino básico). El algoritmo solicitado debe permitir ingresar 4 números al principio, almacenarlos, luego sumar los pares y mostrar los resultados. Funciones: Módulo. Retorna el resto de una división. $MO(x)$.
40. Dentro de un esquema de prueba de software calcular la complejidad ciclomática (cantidad mínima de pruebas que debo realizar para asegurarme que recorrí al menos una vez cada camino básico). El algoritmo solicitado debe permitir ingresar 6 números al principio, almacenarlos, luego mostrar cuantos son pares y cuantos impares. Funciones: Módulo. Retorna el resto de una división. $MO(x)$.
41. Nos solicitan calcular la complejidad ciclomática de acuerdo a las tres formas conocidas, armar el lote de prueba necesario para recorrer al menos una vez cada camino básico y graficar el diagrama de flujo utilizando la simbología de clase, del algoritmo que permita: ingresar 10 números enteros mayores que cero y menores que cien de a uno por vez y luego finalizar. Por cada número ingresado se desea mostrar una leyenda que diga "múltiplo de 3 mayor que 50" cuando el número ingresado sea mayor que 50 y múltiplo de 3 a la vez, "múltiplo de 3 menor o igual que 50" cuando el número ingresado sea menor o igual que 50 y múltiplo de 3 a la vez, "NO múltiplo de 3 mayor que 50" cuando el

numero ingresado sea mayor que 50 y no sea múltiplo de 3 a la vez y “NO múltiplo de 3 menor o igual que 50” cuando el numero ingresado sea menor o igual que 50 y no sea múltiplo de 3 a la vez.

42. Tome el diagrama de clases de su carpeta de proyecto y genere algunos casos de prueba basado en escenarios.
43. Tome el diagrama de clases de su carpeta de proyecto y genere pruebas aleatorias aplicables a nivel de clase.
44. Tome las especificaciones técnicas de su carpeta de proyecto y realice las pruebas que permiten determinar la falta de ambigüedad en las especificaciones.

GUÍA DE ABORDAJE BIBLIOGRÁFICO

UNIDAD I

Pressman, Roger S. "INGENIERÍA DEL SOFTWARE. Un enfoque práctico". 7ª Ed. México. Mc. Graw Hill. 2010. Capítulos 1, 2, 3.

Complementaria – Alternativa

Pressman, Roger S. "INGENIERÍA DEL SOFTWARE. Un enfoque práctico". 6ª Ed. España. Mc. Graw Hill. 2005.

Piattini, Mario - Villalba, José - Ruiz, Francisco - Bastanchury, Teresa - Polo, Macario - Martínez, Miguel Ángel - Nistal, César. "MANTENIMIENTO DEL SOFTWARE. MODELOS, TÉCNICAS Y MÉTODOS PARA LA GESTIÓN DEL CAMBIO". 1ª Ed. Alfaomega. 2001.

Sommerville, Ian. "INGENIERÍA DEL SOFTWARE". 7 Ed. México. Addison Wesley. 2005.

Pfleeger, Shari Lawrence. "INGENIERÍA DEL SOFTWARE – Teoría y Práctica". 1eEd. Argentina. Prentice Hall. 2002.

Weitzenfeld, Alfredo. "INGENIERÍA DEL SOFTWARE ORIENTADA A OBJETOS CON UML". 1 Ed. México. Thompson. 2005.

Braude. "INGENIERÍA DE SOFTWARE. Una perspectiva orientada a objetos". 1ª Ed. México. Alfaomega. 2003.

UNIDAD II

Pressman, Roger S. "INGENIERÍA DEL SOFTWARE. Un enfoque práctico". 7ª Ed. México. Mc. Graw Hill. 2010. Capítulos 9, 10, 11, 12

Complementaria – Alternativa

Pressman, Roger S. "INGENIERÍA DEL SOFTWARE. Un enfoque práctico". 6ª Ed. España. Mc. Graw Hill. 2005.

Piattini, Mario G. - García, Felix O. "CALIDAD EN EL DESARROLLO Y MANTENIMIENTO DEL SOFTWARE". 1ª Ed. Alfaomega. 2003.

Piattini, Mario - Villalba, José - Ruiz, Francisco - Bastanchury, Teresa - Polo, Macario - Martínez, Miguel Ángel - Nistal, César. "MANTENIMIENTO DEL SOFTWARE. MODELOS, TÉCNICAS Y MÉTODOS PARA LA GESTIÓN DEL CAMBIO". 1ª Ed. Alfaomega. 2001.

Sommerville, Ian. "INGENIERÍA DEL SOFTWARE". 7 Ed. México. Addison Wesley. 2005.

UNIDAD III

Pressman, Roger S. "INGENIERÍA DEL SOFTWARE. Un enfoque práctico". 7ª Ed. México. Mc. Graw Hill. 2010. Capítulos 23, 25

Complementaria – Alternativa

Pressman, Roger S. "INGENIERÍA DEL SOFTWARE. Un enfoque práctico". 6ª Ed. España. Mc. Graw Hill. 2005.

Piattini, Mario G. - García, Felix O. "CALIDAD EN EL DESARROLLO Y MANTENIMIENTO DEL SOFTWARE". 1ª Ed. Alfaomega. 2003.

Sommerville, Ian. "INGENIERÍA DEL SOFTWARE". 7 Ed. México. Addison Wesley. 2005.

UNIDAD IV

Pressman, Roger S. "INGENIERÍA DEL SOFTWARE. Un enfoque práctico". 7ª Ed. México. Mc. Graw Hill. 2010. Capítulos 14, 15, 16, 28, 29,30.

Complementaria – Alternativa

Pressman, Roger S. "INGENIERÍA DEL SOFTWARE. Un enfoque práctico". 6ª Ed. España. Mc. Graw Hill. 2005.

Sommerville, Ian. "INGENIERÍA DEL SOFTWARE". 7 Ed. México. Addison Wesley. 2005.

Pfleeger, Shari Lawrence. "INGENIERÍA DEL SOFTWARE – Teoría y Práctica". 1eEd. Argentina. Prentice Hall. 2002.

Weitzenfeld, Alfredo. "INGENIERÍA DEL SOFTWARE ORIENTADA A OBJETOS CON UML". 1 Ed. México. Thompson. 2005.

Braude. "INGENIERÍA DE SOFTWARE. Una perspectiva orientada a objetos". 1ª Ed. México. Alfaomega. 2003.

UNIDAD V

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

Complementaria – Alternativa

Pressman, Roger S. "INGENIERÍA DEL SOFTWARE. Un enfoque práctico". 6ª Ed. España. Mc. Graw Hill. 2005.

UNIDAD VI

Pressman, Roger S. "INGENIERÍA DEL SOFTWARE. Un enfoque práctico". 7ª Ed. México. Mc. Graw Hill. 2010. Capítulos 17, 18, 19, 20.