



Universidad Nacional Autónoma de México  
Facultad de Contaduría y Administración  
Sistema Universidad Abierta y Educación a Distancia

Licenciatura en Informática

# Informática VI. Programación e Implementación de Sistemas

**Cuaderno de actividades**

# COLABORADORES

## **DIRECTOR DE LA FCA**

Dr. Juan Alberto Adam Siade

## **SECRETARIO GENERAL**

L.C. y E.F. Leonel Sebastián Chavarría

-----

## **COORDINACIÓN GENERAL**

Mtra. Gabriela Montero Montiel  
Jefe de la División SUAyED-FCA-UNAM

## **COORDINACIÓN ACADÉMICA**

Mtro. Francisco Hernández Mendoza  
FCA-UNAM

---

## **AUTOR**

Lic. Edith Tapia Rangel

## **DISEÑO INSTRUCCIONAL**

Lic. Guadalupe Montserrat Vázquez Carmona

## **CORRECCIÓN DE ESTILO**

Lic. José Antonio Medina Carranza

## **DISEÑO DE PORTADAS**

L.CG. Ricardo Alberto Báez Caballero  
Mtra. Marlene Olga Ramírez Chavero  
L.DP. Ethel Alejandra Butrón Gutiérrez

## **DISEÑO EDITORIAL**

Mtra. Marlene Olga Ramírez Chavero

## Contenido

Datos de identificación	5
Sugerencias de apoyo	6
Instrucciones para trabajar con el cuaderno de actividades	7
Objetivo general de la asignatura y temario oficial	9
<b>Unidad 1. introducción</b>	<b>10</b>
Objetivo particular y temario detallado	11
Actividad diagnóstica	12
Actividades de aprendizaje	13
Actividad Integradora	14
Cuestionario de reforzamiento	15
Examen parcial de autoevaluación	16
Respuestas	19
<b>Unidad 2. Modelo de implementación</b>	<b>20</b>
Objetivo particular y temario detallado	21
Actividad diagnóstica	22
Actividades de aprendizaje	23
Actividad Integradora	25
Cuestionario de reforzamiento	26
Examen parcial de autoevaluación	27
Respuestas	30
<b>Unidad 3. Plan de implementación</b>	<b>31</b>
Objetivo particular y temario detallado	32
Actividad diagnóstica	33
Actividades de aprendizaje	34
Actividad Integradora	37
Cuestionario de reforzamiento	38
Examen parcial de autoevaluación	39
Respuestas	42

<b>Unidad 4. Implementación de componentes</b>	<b>43</b>
Objetivo particular y temario detallado	44
Actividad diagnóstica	45
Actividades de aprendizaje	46
Actividad Integradora	49
Cuestionario de reforzamiento	50
Examen parcial de autoevaluación	51
Respuestas	54
<b>Unidad 5. Integración de subsistemas y sistemas</b>	<b>55</b>
Objetivo particular y temario detallado	56
Actividad diagnóstica	57
Actividades de aprendizaje	58
Actividad Integradora	60
Cuestionario de reforzamiento	61
Examen parcial de autoevaluación	62
Respuestas	65

## DATOS DE IDENTIFICACIÓN

<b>Informática VI – Programación e Implementación de Sistemas</b>	<b>Clave: 1656</b>
Plan: 2012	Créditos: 8
Licenciatura: Informática	Semestre: 6°
Área o campo de conocimiento: Tecnologías de la Información	Horas por semana: 4
Duración del programa: Semestral	Requisitos: ninguno
Tipo: Teórica    Teoría: 4    Práctica: 0	
Carácter:        Obligatoria ( x )        Optativa ( )	
Seriación: Si ( )    No ( x )        Obligatoria ( )        Indicativa ( x )	
Asignatura con seriación antecedente: Informática III (Análisis y Diseño de Sistemas Estructurado)	
Asignatura con seriación subsecuente: Informática VII (Ingeniería del software)	

## SUGERENCIAS DE APOYO

- Trata de compartir tus experiencias y comentarios sobre la asignatura con tus compañeros, a fin de formar grupos de estudio presenciales o a distancia (comunidades virtuales de aprendizaje, a través de foros de discusión y correo electrónico, etcétera), y puedan apoyarse entre sí.
- Programa un horario propicio para estudiar, en el que te encuentres menos cansado, ello facilitará tu aprendizaje.
- Dispón de periodos extensos para al estudio, con tiempos breves de descanso por lo menos entre cada hora si lo consideras necesario.
- Busca espacios adecuados donde puedas concentrarte y aprovechar al máximo el tiempo de estudio.

## Instrucciones para trabajar con el cuaderno de actividades

El programa de la asignatura consta de 5 unidades. Por cada unidad encontrarás una serie de actividades, el número de las mismas varía de acuerdo a la extensión de la unidad.

Notarás que casi todas las unidades comienzan con la elaboración de un mapa conceptual o mental, esto es con el fin de que tu primera actividad sea esquematizar el contenido total de la unidad para que tengan una mejor comprensión, y dominio total de los temas.

Te recomendamos que leas detenidamente cada actividad a fin de que te quede claro que es lo que tienes que realizar. Si al momento de hacerlo algo no queda claro, no dudes en solicitar el apoyo de tu asesor quien te indicará la mejor forma de realizar tu actividad en asesorías semipresenciales o por correo electrónico para los alumnos de la modalidad abierta, o bien para la modalidad a distancia a través de los medios proporcionados por la plataforma.

Te sugerimos (salvo la mejor opinión de tu asesor), seguir el orden de las unidades y actividades, pues ambas están organizadas para que tu aprendizaje sea gradual. En el caso de los alumnos de la modalidad a distancia, la entrega de actividades está sujeta al plan de trabajo establecido por cada asesor por lo que todo será resuelto directamente en plataforma educativa:

<http://fcaenlinea1.unam.mx/licenciaturas/>

La forma en que deberás responder a cada actividad dependerá de la instrucción dada (número de cuartillas, formatos, si hay que esquematizar etcétera).

Una vez que hayas concluido las actividades entrégalas a tu asesor si así él te lo solicita. Los alumnos de la modalidad a distancia, deberán realizar la actividad directamente en la plataforma educativa de acuerdo a la instrucción dada.

Te invitamos a que trabajes estas actividades con el mayor entusiasmo, pues fueron elaboradas considerando apoyarte en tu aprendizaje de ésta asignatura.



### Indicaciones:

Notarás que tanto los cuestionarios de reforzamiento como las actividades de aprendizaje, contienen instrucciones tales como “adjuntar archivo”, “trabajo en foro”, “texto en línea”, “trabajo en wiki o en Blog”, indicaciones que aplican específicamente para los estudiantes del SUAYED de la modalidad a distancia. Los alumnos de la modalidad abierta, trabajarán las actividades de acuerdo a lo establecido por el asesor de la asignatura en su plan de trabajo, incluyendo lo que sé y lo que aprendí.



### Biblioteca Digital:

Para tener acceso a otros materiales como libros electrónicos, es necesario que te des de alta a la Biblioteca Digital de la UNAM (BIDI). Puedes hacerlo desde la página principal de la FCA <http://www.fca.unam.mx/> **Alumnos >Biblioteca >Biblioteca digital >Clave para acceso remoto >Solicita tu cuenta.** Elige la opción de “Alumno” y llena los campos solicitados. Desde este sitio, también puedes tener acceso a los libros electrónicos.

## OBJETIVO GENERAL

Al finalizar el curso, el alumno aplicará el proceso de implementación y las pruebas para la construcción de sistemas de información.

## TEMARIO OFICIAL (64 horas)

	Horas
1. Introducción	10
2. Modelo de implementación	10
3. Plan de implementación	10
4. Implementación de componentes	24
5. Integración de subsistemas y sistemas	10
<b>Total</b>	<b>64</b>

# Introducción



## OBJETIVO PARTICULAR

Identificar los diferentes estilos y estándares de programación para la construcción de sistemas.

## TEMARIO DETALLADO (10 horas)

### 1. Introducción

1.1. Paradigmas de programación

1.2. Principios de programación

---

# ACTIVIDAD DIAGNÓSTICA

## LO QUE SÉ



*Actividad en foro.*

Con base en tu experiencia, y sin realizar consulta alguna, responde en el foro “Paradigmas y principios” las siguientes preguntas:

1. ¿Qué entiendes por paradigma?
2. ¿Qué entiendes por principio?

Recuerda que debes comentar al menos tres participaciones de tus compañeros.

Si tu asignatura la trabajas fuera de plataforma educativa, entonces realiza la misma actividad en no más de una cuartilla y entrégala a tu asesor.

# ACTIVIDADES DE APRENDIZAJE



**Unidad 1, actividad inicial. *Adjuntar archivo.*** A partir del estudio de la bibliografía específica sugerida, elabora un mapa conceptual u [organizador gráfico](#) con los temas de la unidad. Puedes auxiliarte de algunos programas como Mindjet [MindManager](#).

1. **Unidad 1, actividad 1. *Adjuntar archivo.*** Revisa el documento ["Operational versus definitional: a perspective on programming paradigms"](#),

Ambler, Burnett & Zimmerman. (1992).  
Operational versus definitional: a perspective on programming paradigms.  
Computer, 29 - 41

Posteriormente, extrae de dos paradigmas el ejemplo del algoritmo de ordenamiento. Transcribe los dos ejemplos y explica la diferencia entre ellos en un documento de máximo dos cuartillas.

2. **Unidad 1, actividad 2. *Adjuntar archivo.*** De acuerdo con tu propio criterio, en un procesador de textos ordena por relevancia los principios de programación (descritos en el tema "Principios de programación" del documento base de la unidad). Agrega la justificación de dicho orden.
3. **Unidad 1, actividad complementaria. *Adjuntar archivo.*** A partir del estudio de la unidad, realiza la siguiente actividad, misma que encontraras en el foro de la asignatura. Cabe señalar que esta será colocada en el foro por tu asesor.

# ACTIVIDAD INTEGRADORA

## LO QUE APRENDÍ



*Adjuntar archivo.*

Elabora un mapa conceptual con los conceptos abordados en la unidad. Incluye al final una reflexión del tema que haya sido más relevante para ti y justifica tu elección.

# CUESTIONARIO DE REFORZAMIENTO



*Adjuntar archivo. complementra*

1. ¿Qué es un paradigma de programación?
2. ¿Cuáles son las características de los paradigmas operacionales?
3. ¿Cuáles son los dos tipos de paradigmas operacionales?
4. ¿En qué consiste el paradigma imperativo?
5. ¿Cómo operan los paradigmas funcionales operacionales?
6. ¿Cuáles son los tres tipos de paradigmas pseudodeclarativos?
7. ¿Cuáles son las características del paradigma lógico?
8. ¿Cuáles son los tres tipos de paradigmas propiamente declarativos?
9. Describe en qué consiste el principio que señala “evita trucos”.
10. ¿Cuáles son las implicaciones del principio “escribe el código para leerlo de arriba hacia abajo”?
11. ¿Cómo se interpreta el principio “haz uso de estructuras de datos óptimas”?
12. ¿Por qué es importante considerar la aplicación del principio “ejecuta de manera aislada cada componente”?

# EXAMEN PARCIAL

## (de autoevaluación)



### I. *Selecciona la respuesta correcta.*

1. Un paradigma de programación es:

- |   |   |
|---|---|
| <input type="radio"/> a) Un conjunto de teorías que definen un estilo | <input type="radio"/> b) Una colección de patrones conceptuales |
| <input type="radio"/> c) Un conjunto de métodos de programación       |   |

2. Las tres grandes categorías en las que se agrupan los paradigmas de programación son:

- |  |   |
|--|---|
| <input type="radio"/> a) Operacionales, declarativos y demostrativos.                | <input type="radio"/> b) Estructurados, modulares y orientados a objetos. |
| <input type="radio"/> c) Orientados al usuario, orientados al servicio y por eventos |   |

3. Es el tipo de paradigma operacional al que pertenece el paradigma imperativo.

- |   |   |
|---|---|
| <input type="radio"/> a) Los que continuamente crean nuevos datos | <input type="radio"/> b) Los que modifican la representación de datos |
| <input type="radio"/> c) Los que intercambian datos               |   |

4. Paradigma donde los procedimientos operan sobre valores abstractos denominados “objetos”, en lugar de representaciones almacenadas.

<input type="radio"/> a) Paradigma imperativo	<input type="radio"/> b) Paradigma demostrativo
<input type="radio"/> c) Paradigma orientado a objetos	

5. ¿Cuáles son las dos propuestas que existen para los paradigmas definicionales?

<input type="radio"/> a) Imperativo y orientado a objetos	<input type="radio"/> b) Pseudodeclarativos y propiamente declarativos
<input type="radio"/> c) Inferenciales y no inferenciales	

6. Paradigma que trata de asemejarse a un modelo matemático mediante la expresión de funciones como lo hacen los matemáticos.

<input type="radio"/> a) Paradigma orientado a objetos	<input type="radio"/> b) Paradigma funcional
<input type="radio"/> c) Paradigma pseudodeclarativo	

7. Tipos de paradigmas demostrativos.

<input type="radio"/> a) Transformacional y lógico	<input type="radio"/> b) Basado en la forma y flujo de datos
<input type="radio"/> c) Inferenciales y no inferenciales	

8. La razón de ser del principio “Evite variables globales” es:

<input type="radio"/> a) Evitar errores lógicos difíciles de rastrear	<input type="radio"/> b) Lograr seguridad en el trabajo
<input type="radio"/> c) Hacer más comprensible el código	

9. La importancia de aplicar el principio “usa nombres significativos” radica en:

<input type="radio"/> a) Que facilita la lectura FSK	<input type="radio"/> b) Que evita un aumento de costos por mantenimiento al disminuir el tiempo para descifrar el código
<input type="radio"/> c) Que incrementa el uso de comentarios	

10. El beneficio derivado de aplicar el principio “comenta antes de finalizar la programación” es:

<input type="radio"/> a) Mayor facilidad para depurar el <i>software</i>	<input type="radio"/> b) Lograr programas más estructurados
<input type="radio"/> c) Favorecer la ejecución de la aplicación	

11. Es una directriz que apoya el principio “utiliza lenguajes de programación apropiados”.

<input type="radio"/> a) Inspecciona el código	<input type="radio"/> b) Ejecuta de manera aislada cada componente
<input type="radio"/> c) Si buscas aplicaciones que requieran poco mantenimiento, utiliza un lenguaje con muchas características de calidad incorporadas	

12. Es una directriz que apoya el principio “utiliza lenguajes de programación apropiados”.

<input type="radio"/> a) Los programadores deben dominar todos los lenguajes de programación	<input type="radio"/> b) Los buenos programadores son capaces de aprender cualquier nuevo lenguaje fácilmente
<input type="radio"/> c) Los malos programadores deben aprender la sintaxis y semántica de todos los lenguajes de programación	

# RESPUESTAS

## EXAMEN DE AUTOEVALUACIÓN



En este apartado encontrarás las respuestas al examen por unidad.

UNIDAD 1
I. Solución
1. <b>b</b>
2. <b>a</b>
3. <b>b</b>
4. <b>c</b>
5. <b>b</b>
6. <b>b</b>
7. <b>c</b>
8. <b>a</b>
9. <b>b</b>
10. <b>a</b>
11. <b>c</b>
12. <b>b</b>

# Modelo de Implementación

## OBJETIVO PARTICULAR

Diseñar un modelo de implementación de los paquetes de subsistemas que conforman el sistema.

## TEMARIO DETALLADO (10 horas)

### 2. Modelo de implementación

2.1. Patrones de diseño

2.2. Marcos de trabajo

2.3. Programación extrema

2.4. Frases del desarrollo rápido de aplicaciones (RAD)

---

# ACTIVIDAD DIAGNÓSTICA

## LO QUE SÉ



*Adjuntar archivo.*

Sin consultar algún tipo de información, responde a las siguientes preguntas:

1. ¿Qué entiendes por patrón de diseño?
2. ¿Cuál es tu definición para un marco de trabajo?
3. ¿A que nos referimos con el término “programación extrema”?
4. ¿Qué es RAD?

# ACTIVIDADES DE APRENDIZAJE



**Unidad 2, actividad inicial. *Adjuntar archivo.*** A partir del estudio de la bibliografía específica sugerida, elabora un mapa conceptual u [organizador gráfico](#) con los temas de la unidad. Puedes auxiliarte de algunos programas como Mindjet [MindManager](#).

1. **Unidad 2, actividad 1. *Adjuntar archivo.*** Revisa el texto de Gamma "[Design patterns](#)" y elije un patrón de diseño de los que ahí se presentan.

Gamma, Helm, Johnson, Vlissides. (1994).  
Design Patterns, 431  
USA: Addison-Wesley.

Posteriormente elabora un cuadro con todas las características del patrón elegido.

2. **Unidad 2, actividad 2. *Adjuntar archivo.*** Con base en el tema "Marcos de trabajo" del documento base de la unidad, elabora un cuadro sinóptico que describa las tres dimensiones características de un marco de trabajo.
3. **Unidad 2, actividad 3. *Adjuntar archivo.*** Investiga acerca de algunas de estas estructuras de los marcos de trabajo presentadas en el documento base de la unidad:
  - a) The layered architectural framework
  - b) The pipes filters architectural framework
  - c) The model view controller architectural framework
  - d) The presentation-abstraction-controller architectural framework
  - e) The reflective architectural framework
  - f) The microkernel architectural framework
  - g) The blackboard architectural framework
  - h) The broker architectural framework

Posteriormente redacta un análisis de máximo tres cuartillas, donde reportes tus hallazgos. No olvides incluir las referencias.

4. **Unidad 2, actividad 4. *Adjuntar archivo.*** En un procesador de textos o presentador de diapositivas, elabora un mapa mental sobre los elementos que componen la programación extrema, los cuales fueron descritos en el tema "Programación extrema" del documento base de la unidad.
5. **Unidad 2, actividad 5. *Adjuntar archivo.*** Revisa el siguiente "[Ejemplo de desarrollo de software](#)" para que conozcas el desarrollo de software utilizando la metodología Extreme Programming (programación extrema) y, posteriormente, en un procesador de textos describe una historia de usuario con su respectiva prueba.

Ejemplo de desarrollo software utilizando la metodología XP  
[http://users.dsic.upv.es/assignaturas/facultad/lsi/ejemploxp/Gestion\\_Proyecto.html#planificacion](http://users.dsic.upv.es/assignaturas/facultad/lsi/ejemploxp/Gestion_Proyecto.html#planificacion)  
06/Enero/2017

6. **Unidad 2, actividad 6. *Adjuntar archivo.*** En un procesador de textos o en un presentador de diapositivas, elabora un diagrama sobre cómo se organizan las fases y tareas de la metodología RAD, con base en lo descrito en el tema "Fases del desarrollo rápido de aplicaciones (RAD)" del documento base de la unidad.
7. **Unidad 2, actividad complementaria. *Adjuntar archivo.*** A partir del estudio de la unidad, realiza la siguiente actividad, misma que encontraras en el foro de la asignatura. Cabe señalar que esta será colocada en el foro por tu asesor.

# ACTIVIDAD INTEGRADORA

## LO QUE APRENDÍ



*Adjuntar archivo.*

1. Revisa el documento “[Ejercicio guiado de análisis y diseño orientado a objetos](#)”.

Diana Marcela Sánchez Fúquene  
Marzo, 2013

Pon atención a la página 25 y establece en qué patrón de diseño se comprenden las clases definidas en dicha página.

2. Considera uno de los marcos de trabajo presentados en la unidad (Microsoft Foundation Classes, Java’s Abstract, Windows Toolkit u otro) y define los componentes a crear dentro del marco para realizar la aplicación del cajero automático.

# CUESTIONARIO DE REFORZAMIENTO



***Adjuntar archivo. Responde las siguientes preguntas.***

1. ¿Quién hizo la primera referencia al término “patrón de diseño”?
2. ¿Qué es un patrón de diseño?
3. ¿Cuáles son los pasos para seleccionar un patrón de diseño?
4. ¿Qué son los marcos de trabajo?
5. ¿Cuáles son las dimensiones que caracterizan a los marcos de trabajo?
6. ¿Cuáles son las dos formas en que se pueden utilizar los marcos de trabajo?
7. ¿Cuáles son las dos formas en que se puede entender la programación extrema?
8. ¿Cuáles son los cuatro valores que establece la programación extrema apreciables en sus reglas y práctica?
9. ¿Cuáles son los pasos que sigue el plan de juego de la programación extrema?
10. ¿Qué es el desarrollo rápido de aplicaciones (RAD)?
11. ¿Qué actividades se realizan en la tarea “finalizar requerimientos” de la etapa “plan de requerimientos”?
12. ¿En qué consiste la etapa de “construcción rápida”?

# EXAMEN PARCIAL

## (de autoevaluación)



### I. *Selecciona la respuesta correcta.*

1. ¿Para qué disciplina fue propuesto por primera vez el término “patrón de diseño”?

- |                                       |                                      |
|---------------------------------------|--------------------------------------|
| <input type="radio"/> a) Arquitectura | <input type="radio"/> b) Informática |
| <input type="radio"/> c) Ingeniería   |                                      |

2. Elemento de los patrones de diseño que describe cuándo aplicar el patrón y explica el problema y su contexto.

- |                                   |                                   |
|-----------------------------------|-----------------------------------|
| <input type="radio"/> a) Nombre   | <input type="radio"/> b) Problema |
| <input type="radio"/> c) Solución |                                   |

3. Patrón de diseño cuya intención es definir el esqueleto de un algoritmo en una operación, aplazando algunos pasos a las subclasses.

- |  |                                  |
|--|----------------------------------|
| <input type="radio"/> a) Método de plantilla | <input type="radio"/> b) Momento |
| <input type="radio"/> c) Mediador            |                                  |

4. Tipo de marco de trabajo donde un cliente puede utilizar el marco, instanciando clases y realizando composiciones de instancias que adapta a sus necesidades.

- |  |  |
|--|--|
| <input type="radio"/> a) De caja blanca                          | <input type="radio"/> b) De caja negra |
| <input type="radio"/> c) Combinación de caja blanca y caja negra |  |

5. Dimensión del marco de trabajo que resuelve los aspectos de aplicación, dominio y soporte.

- |  |  |
|--|--|
| <input type="radio"/> a) Dominio del problema que resuelve el marco de trabajo | <input type="radio"/> b) Estructura interna del marco de trabajo |
| <input type="radio"/> c) Cómo se utilizará el marco de trabajo                 |  |

6. Es un marco de trabajo popular en el mundo del desarrollo de software.

- |   |                                      |
|---|--------------------------------------|
| <input type="radio"/> a) C++                    | <input type="radio"/> b) Java Script |
| <input type="radio"/> c) Reusable Objects (ORO) |                                      |

7. Variable de la programación extrema que establece la exactitud que requiere el sistema y cómo se evaluará.

- |                                  |                                  |
|----------------------------------|----------------------------------|
| <input type="radio"/> a) Tiempo  | <input type="radio"/> b) Calidad |
| <input type="radio"/> c) Alcance |                                  |

8. Valor de la programación extrema que establece que una buena comunicación es un factor clave necesario para lograr un proyecto de software exitoso.

- |                                       |                                      |
|---------------------------------------|--------------------------------------|
| <input type="radio"/> a) Comunicación | <input type="radio"/> b) Simplicidad |
| <input type="radio"/> c) Valentía     |                                      |

9. Fase del plan de juego de liberación en la que el desarrollador acepta la responsabilidad de una tarea asignada voluntariamente.

- |                                      |                                     |
|--------------------------------------|-------------------------------------|
| <input type="radio"/> a) Exploración | <input type="radio"/> b) Planeación |
| <input type="radio"/> c) Dirección   |                                     |

10. Etapa del desarrollo rápido de aplicaciones que, entre otras actividades, realiza una revisión de las áreas más relacionadas con el sistema propuesto.

- |   |  |
|---|--|
| <input type="radio"/> a) Construcción rápida    | <input type="radio"/> b) Diseño de usuario |
| <input type="radio"/> c) Plan de requerimientos |  |

11. Tarea de la etapa “diseño del usuario” donde se identifican los procedimientos requeridos por el sistema y se desarrollan esquemas tentativos para pantallas y reportes.

- |   |   |
|---|---|
| <input type="radio"/> a) Producir un modelo de área del sistema detallado | <input type="radio"/> b) Desarrollar el diseño exterior del sistema |
| <input type="radio"/> c) Refinar el diseño del sistema                    |   |

12. Tarea de la etapa “transición” que consiste en convertir la información necesaria para la operación del nuevo sistema, a partir de las fuentes de datos existentes.

- |   |  |
|---|--|
| <input type="radio"/> a) Conducir la capacitación del usuario | <input type="radio"/> b) Ejecutar la conversión de datos |
| <input type="radio"/> c) Instalar el sistema de producción    |  |

# RESPUESTAS

## EXAMEN DE AUTOEVALUACIÓN



En este apartado encontrarás las respuestas al examen por unidad.

UNIDAD 2
I. Solución
1. <b>a</b>
2. <b>b</b>
3. <b>a</b>
4. <b>b</b>
5. <b>a</b>
6. <b>c</b>
7. <b>b</b>
8. <b>a</b>
9. <b>b</b>
10. <b>c</b>
11. <b>b</b>
12. <b>b</b>

# Plan de Implementación

## OBJETIVO PARTICULAR

Planear la implementación de subsistemas.

## TEMARIO DETALLADO (10 horas)

### 3. Plan de implementación

- 3.1. Definición de objetivos
- 3.2. Estimación de tareas y tiempos
- 3.3. administración de la configuración
- 3.4. Administración de cambios
- 3.5. Modelo de la arquitectura propuesta

# ACTIVIDAD DIAGNÓSTICA

## LO QUE SÉ



*Adjuntar archivo.*

Responde las siguientes preguntas sin consultar fuente alguna o documento.

1. ¿Qué es un objetivo? ¿Por qué son importantes los objetivos?
2. ¿Por qué es valioso realizar el proceso de estimación en el desarrollo de un proyecto?
3. ¿Qué entiendes por administración de la configuración?
4. ¿Qué significa para ti el término “administración de cambios”?
5. ¿Cuál es tu concepto para “arquitectura de software”?

## ACTIVIDADES DE APRENDIZAJE



**Unidad 3, actividad inicial. *Adjuntar archivo.*** A partir del estudio de la bibliografía específica sugerida, elabora un mapa conceptual u [organizador gráfico](#) con los temas de la unidad. Puedes auxiliarte de algunos programas como Mindjet [MindManager](#).

1. **Unidad 3, actividad 1. *Adjuntar archivo.*** Elabora un mapa mental con todos los aspectos descritos en el tema “Definición de objetivos” del documento base de la unidad.
2. **Unidad 3, actividad 2. *Adjuntar archivo.*** Identifica un problema que requiera solución con un sistema, aplicando la herramienta “Es/No es” para describirlo (el uso de la herramienta se describe en los apuntes de la unidad).  
Describe el problema y la solución al mismo en un máximo de tres cuartillas.
3. **Unidad 3, actividad 3. *Adjuntar archivo.*** Establece los objetivos y la definición de un proyecto que resuelva el problema identificado en la actividad 2 (de esta unidad 3), mediante la herramienta SMART (presentada en el documento base).
4. **Unidad 3, actividad 4. *Adjuntar archivo.***
  - a) Describe las actividades necesarias que realizaste para desarrollar la actividad 1 (de esta unidad 3) y los tiempos de realización.
  - b) En el mismo documento y considerando las actividades y tiempos anteriores, establece una estimación de las actividades y tiempos requeridos para realizar un mapa mental que contemple todos los aspectos que conciernen al tema “Estimación de tareas y tiempos”. Acompaña tu estimación con una reflexión acerca del proceso desarrollado.

**5. Unidad 3, actividad 5. *Adjuntar archivo.***

- a) En un documento define las consideraciones de estimación de tamaño, lenguaje de desarrollo, herramientas de desarrollo, fiabilidad del software, almacenamiento del software, complejidad del software y factores de plataforma del proyecto establecido en la actividad 3 (de esta unidad 3).
- b) Asimismo, incluye una estimación no algorítmica del tiempo y costo del proyecto, mediante la elaboración en el mismo documento de un cronograma y de un presupuesto.

**6. Unidad 3, actividad 6. *Adjuntar archivo.*** Con base en lo descrito en el tema “Administración de la configuración”, del documento base de la unidad, elabora un documento de no más de dos cuartillas, en donde reflexiones acerca de la importancia de establecer las líneas base para la administración de la configuración.

**7. Unidad 3, actividad 7. *Adjuntar archivo.***

- a) Con base en lo descrito en el tema “Administración de la configuración”, del documento base de la unidad, establece los siguientes elementos de configuración de software de tu proyecto (el proyecto fue el desarrollado en la actividad 3), descrito en la actividad 5 (de esta unidad 3):
  - i. Programas necesarios.
  - ii. Documentación a generar.
  - iii. Estructuras de datos requeridas para el proyecto.
- b) Asimismo, define el proceso que llevarás a cabo para el control de cambios en el proyecto propuesto.

**8. Unidad 3, actividad 8. *Adjuntar archivo.*** Realiza una investigación en Internet sobre dos herramientas que soporten la gestión de la configuración, y con tus hallazgos elabora un reporte que indique las herramientas investigadas y sus características más importantes en no más de dos cuartillas. No olvides incluir referencias.

9. **Unidad 3, actividad 9. *Adjuntar archivo.*** Elabora un mapa mental que considere todos los temas abordados en el tema “Administración de cambios”.
10. **Unidad 3, actividad 10. *Adjuntar archivo.*** Busca en Internet la representación gráfica para uno de los estilos arquitectónicos descritos en el tema “Modelo de la arquitectura propuesta”, y elabora un reporte que incluya el gráfico y las principales características del estilo seleccionado. No olvides incluir las referencias.
11. **Unidad 3, actividad 11. *Adjuntar archivo.***
- Considerando tu proyecto propuesto en la actividad 7 (de esta unidad 3), establece qué patrón o patrones de diseño se adecuan al mismo y justifica tu elección.
  - Considerando el patrón de diseño seleccionado, señala qué patrón arquitectónico sería el más adecuado utilizar y justifica tu elección.
  - Finalmente, considerando los patrones de diseño definidos y el patrón arquitectónico seleccionado, elige el estilo arquitectónico que más se adecue a los mismos y justifica tu elección.
12. **Unidad 3, actividad 12. *Adjuntar archivo.*** Considerando el patrón de diseño seleccionado en la actividad 11 (de esta unidad 3), señala qué patrón arquitectónico sería el más adecuado utilizar y justifica tu elección.
13. **Unidad 3, actividad 13. *Adjuntar archivo.*** Finalmente, considerando los patrones de diseño definidos en la actividad 11 y el patrón arquitectónico seleccionado en la actividad 12 (de esta unidad 3), elige el estilo arquitectónico que más se adecue a los mismos y justifica tu elección.
14. **Unidad 3, actividad complementaria. *Adjuntar archivo.*** A partir del estudio de la unidad, realiza la siguiente actividad, misma que encontraras en el foro de la asignatura. Cabe señalar que esta será colocada en el foro por tu asesor.

# ACTIVIDAD INTEGRADORA

## LO QUE APRENDÍ



*Adjuntar archivo.*

1. Integra en un documento que denomines “Plan de implementación”, los productos de tus actividades 2, 3, 5, 7, 11, 12 y 13 (de esta unidad), y a partir de ello genera un índice común de los mismos.
2. Agrega un apartado en donde aportes tu reflexión sobre el proceso desarrollado.

# CUESTIONARIO DE REFORZAMIENTO



**Adjuntar archivo. Responde las siguientes preguntas.**

1. ¿Qué es un proyecto?
2. ¿Por qué es importante establecer los objetivos?
3. SMART es sigla de...
4. ¿Cómo se denomina a las variables en las que se basan los modelos de estimación del costo de desarrollo del software?
5. ¿Qué caracteriza a los métodos de estimación no algorítmicos?
6. ¿En qué consiste el método de estimación conocido como “uso de analogías”?
7. ¿Cómo se caracterizan los métodos de estimación algorítmicos?
8. ¿Cuáles son los grupos en los que se dividen los factores de costo de COCOMO?
9. ¿Cuáles son las principales características del método de estimación algorítmico analítico SOFTCOST?
10. ¿Qué es la administración de la configuración?
11. ¿Qué son los elementos de configuración del software?
12. ¿Cuáles son los elementos que considera la “identificación de objetos en la configuración del software”?
13. ¿Qué es la arquitectura de software?
14. ¿Qué establece la recomendación IEEE Std 1471-2000?
15. ¿Qué son los estilos arquitectónicos?

# EXAMEN PARCIAL

## (de autoevaluación)



### I. *Selecciona la respuesta correcta.*

1. Es la expresión de un fin que se quiere conseguir y que debe permitir la articulación de una serie de acciones encaminadas a su consecución.

a) Proyecto

b) Objetivo

c) Programa

2. Herramienta que permite describir un problema de manera más integral, integrando todas las comprensiones parciales del mismo.

a) Herramienta SMART

b) Herramienta PERT

c) Herramienta Es/No es

3. Es el proceso que proporciona un valor a un conjunto de variables para la realización de un trabajo, dentro de un rango aceptable de tolerancia.

a) Definición de objetivos

b) Estimación

c) Modelado de la arquitectura

4. Es el proceso de predecir el esfuerzo requerido para el desarrollo de un sistema de software.

a) Estimación del costo

b) Planeación temporal

c) Diseño del modelo

5. Método de estimación no algorítmico que se basa en la experiencia del experto y su conocimiento de las normas dominantes de la industria, como base para la estimación del costo del software.

<input type="radio"/> a) Uso de analogías	<input type="radio"/> b) <i>Price-to-win</i>
<input type="radio"/> c) Método de estimación experta	

6. Método de estimación no algorítmico que determina el costo del proyecto en función del precio que está dispuesto a pagar el cliente.

<input type="radio"/> a) El principio de Parkinson	<input type="radio"/> b) <i>Price-to-win</i>
<input type="radio"/> c) Enfoque <i>botton-up</i>	

7. Es un método de estimación algorítmico empírico.

<input type="radio"/> a) COCOMO	<input type="radio"/> b) SOFTCOST
<input type="radio"/> c) SLIM	

8. Son aquellos métodos de estimación que se basan en una comprensión del problema mediante la descomposición, para así entender mejor su comportamiento y a partir de ello desarrollar ecuaciones matemáticas que modelen el problema de estimar el esfuerzo de desarrollo del software.

<input type="radio"/> a) Métodos de estimación algorítmicos empíricos	<input type="radio"/> b) Métodos de estimación experta
<input type="radio"/> c) Métodos de estimación algorítmicos analíticos	

9. Método de estimación que se apoya en el modelo teórico de Norden-Rayleigh.

<input type="radio"/> a) SOFTCOST	<input type="radio"/> b) COCOMO
<input type="radio"/> c) SLIM	

10. Factor que consideran los métodos de estimación que se mide en las variables líneas de código (LOC), miles de líneas de código (KLOC) o líneas de código fuente (SLOC).

- |   |                                 |
|---|---------------------------------|
| <input type="radio"/> a) Lenguaje de desarrollo     | <input type="radio"/> b) Tamaño |
| <input type="radio"/> c) Herramientas de desarrollo |                                 |

11. Factor que consideran los métodos de estimación que atiende la probabilidad de ejecutar una aplicación y que no se presente ninguna falla en el sistema durante un tiempo y condiciones específicas.

- |  |  |
|--|--|
| <input type="radio"/> a) Almacenamiento del software | <input type="radio"/> b) Fiabilidad del software |
| <input type="radio"/> c) Factores de plataforma      |  |

12. Es un concepto de gestión de configuraciones del software que sirve como la base para continuar en el desarrollo.

- |  |   |
|--|---|
| <input type="radio"/> a) Elementos de configuración del software | <input type="radio"/> b) Administración de la configuración |
| <input type="radio"/> c) Línea base                              |   |

13. Tarea del proceso de “gestión de la configuración del software”, que inicia con una petición de cambio.

- |   |  |
|---|--|
| <input type="radio"/> a) Generación de informes | <input type="radio"/> b) Auditorías de configuración |
| <input type="radio"/> c) Control de cambios     |  |

14. Estilo de arquitectura de software que se aplica cuando los datos de entrada son transformados a través de una serie de componentes computacionales en los datos de salida.

- |  |   |
|--|---|
| <input type="radio"/> a) Arquitectura de llamada y retorno | <input type="radio"/> b) Arquitectura orientada a objetivos |
| <input type="radio"/> c) Arquitectura de flujo de datos    |   |

# RESPUESTAS

## EXAMEN DE AUTOEVALUACIÓN



En este apartado encontrarás las respuestas al examen por unidad.

Unidad 3	
I. Solución	
1.	<b>b</b>
2.	<b>c</b>
3.	<b>b</b>
4.	<b>a</b>
5.	<b>c</b>
6.	<b>b</b>
7.	<b>a</b>
8.	<b>c</b>
9.	<b>c</b>
10.	<b>b</b>
11.	<b>b</b>
12.	<b>c</b>
13.	<b>c</b>
14.	<b>c</b>

# Implementación de componentes

## OBJETIVO PARTICULAR

Codificar en un lenguaje de programación los componentes del sistema.

## TEMARIO DETALLADO (24 horas)

### 4. Implementación de componentes

- 4.1. Definición de componentes
- 4.2. Estándares y buenas prácticas de implementación
- 4.3. Diseño y modelo de componentes
- 4.4. Técnicas de implementación
- 4.5. Depuración y métodos para revisar el código

# ACTIVIDAD DIAGNÓSTICA

## LO QUE SÉ



*Adjuntar archivo.*

Responde las siguientes preguntas sin consultar ninguna referencia.

1. ¿Qué entiendes por componente?
2. ¿Qué es una buena práctica?
3. ¿Qué debe lograr la implementación de un sistema?
4. ¿Qué entiendes por modelo de componentes?
5. ¿Qué entiendes por “técnica de implementación”?
6. ¿Qué son las pruebas?
7. ¿Qué es la depuración?
8. ¿Cuál es el valor de las pruebas y la depuración en un proceso de desarrollo de software?

# ACTIVIDADES DE APRENDIZAJE



**Unidad 4, actividad inicial. *Adjuntar archivo.*** A partir del estudio de la bibliografía específica sugerida, elabora un mapa conceptual u [organizador gráfico](#) con los temas de la unidad. Puedes auxiliarte de algunos programas como Mindjet [MindManager](#).

1. **Unidad 4, actividad 1. *Adjuntar archivo.*** Elabora un cuadro sinóptico que contemple todos los elementos abordados en el tema “Definición de componente” del documento base de la unidad, e incluye tu propia definición de componente.

2. **Unidad 4, actividad 2. *Adjuntar archivo.***

a) Accede a [ComponentSource](#) y regístrate.

Component Source  
<https://www.componentsource.com/>  
1996 - 2017

b) Después, revisa y elige algún componente que pudiera soportar el proyecto que propusiste en el plan de implementación.

c) Posteriormente, evalúa el componente traduciendo y elaborando la siguiente [tabla](#).

Evaluation of the Fuzzy Kitten software's technical quality  
Campinas, SP, Brazil 2010

3. **Unidad 4, actividad 3. *Adjuntar archivo.*** Elabora un esquema en donde se visualicen todos los elementos que comprenden las actividades de implementación del estándar IEEE 1074-1997 SDLC, descrito en el documento base de la unidad. Acompáñalo con una reflexión acerca de la importancia de estas actividades para la implementación de componentes.

4. **Unidad 4, actividad 4. Adjuntar archivo.** Elabora una tabla con cada uno de los estándares de codificación, como se muestra a continuación:

Estándar	Valoración (adecuado, inadecuado)
Claridad	
Consistencia	
Uso de tabuladores	
...	

Una vez elaborada la tabla, elige el código fuente de algún programa que hayas escrito y califica tu código fuente de acuerdo con los estándares.

5. **Unidad 4, actividad 5. Adjuntar archivo.** Elabora un mapa mental de los aspectos abordados en el tema “Diseño y modelo de componentes”.
6. **Unidad 4, actividad 6. Adjuntar archivo.** De acuerdo con el componente seleccionado en la actividad 2 (de esta unidad 4), en una cuartilla señala el *framework* que lo soporte; presenta sus características y justifica tu elección.
7. **Unidad 4, actividad 7. Adjuntar archivo.** Elabora una tabla comparativa con tres columnas para los siguientes tipos de programación: programación modular, programación estructurada y programación orientada a objetos, y enlista en cada columna las características principales de cada tipo de programación, descritas en el tema “Técnicas de implementación”, del documento base de la unidad.

Programación modular	Programación estructurada	Programación orientada a objetos

8. **Unidad 4, actividad 8. Adjuntar archivo.** Elabora un cuadro sinóptico del subtema “Pruebas”, del tema “Depuración y métodos para revisar el código” del documento base de la unidad.
9. **Unidad 4, actividad 9. Adjuntar archivo.** Elabora dos diagramas de flujo, uno para el proceso de pruebas de unidad y otro para el proceso de

depuración, descritos en el tema “Depuración y métodos para revisar el código” del documento base de la unidad.

10. **Unidad 4, actividad 10.** *Adjuntar archivo.* Define un caso de prueba de caja blanca para el código fuente que seleccionaste en la actividad 4 (de esta unidad).
11. **Unidad 4, actividad complementaria.** *Adjuntar archivo.* A partir del estudio de la unidad, realiza la siguiente actividad, misma que encontraras en el foro de la asignatura. Cabe señalar que esta será colocada en el foro por tu asesor.

# ACTIVIDAD INTEGRADORA

## LO QUE APRENDÍ



*Adjuntar archivo.*

Elabora un documento a modo de reflexión, de no más de cuatro cuartillas, sobre la relevancia que tienen los temas tratados para el licenciado en informática, así como la dificultad que tuviste para comprenderlos y cuál consideras que puedes aplicar en el corto plazo para desarrollar tus actividades laborales, en qué situación y cómo lo aplicarías.

# CUESTIONARIO DE REFORZAMIENTO



**Adjuntar archivo. Responde las siguientes preguntas.**

1. De acuerdo con lo revisado en la unidad, ¿qué es un componente?
2. ¿Cuáles son las características clave de los componentes?
3. ¿A qué se refiere la característica de los componentes “sus servicios no varían”?
4. ¿Cuáles son los tres ámbitos a partir de los cuales se pueden describir los componentes?
5. ¿Cuáles son las variables que permiten clasificar los componentes?
6. ¿Qué son las métricas de complejidad?
7. Describe en qué consiste la variable reusabilidad para clasificar los componentes.
8. ¿Cuáles son las características que deben cumplir las buenas prácticas de codificación?
9. ¿Cuáles son los estándares generales de codificación que se pueden aplicar a todos los lenguajes?
10. ¿Qué señala el estándar general de codificación para las enumeraciones?

# EXAMEN PARCIAL

## (de autoevaluación)



### I. *Selecciona la respuesta correcta.*

1. Es un paquete dinámicamente vinculado con uno o varios programas manejados como una unidad, y que son accedidos mediante interfaces bien documentadas que pueden ser descubiertas en tiempo de ejecución.

a) Sistemas

b) Componente

c) Función

2. Es una característica de los componentes que señala que no se debe requerir la utilización de otros para finalizar la función para la cual fue diseñado.

a) Identificable

b) Autocontenido

c) Genérico

3. Característica de los componentes que señala que sus servicios deben servir para varias aplicaciones.

a) Bien documentado

b) Independiente

c) Genérico

4. Ámbito que describe al componente en el cual se debe adquirir el conocimiento acerca de las entidades de diseño y comportamiento, así como el conocimiento del flujo de información y tipos de algoritmos.

a) Programa

b) Dominio

c) Situación

5. Variable para clasificar los componentes que en algunas ocasiones utiliza métricas de tamaño para evaluar la complejidad.

<input type="radio"/> a) Complejidad	<input type="radio"/> b) Tamaño
<input type="radio"/> c) Mantenibilidad	

6. Métrica para medir la complejidad de los componentes que se centra en la estructura interna del componente, utilizando variables como la relación entre clases y el peso de cada relación.

<input type="radio"/> a) <i>Component Dynamic Complexity</i>	<input type="radio"/> b) <i>Component Plain Complexity</i>
<input type="radio"/> c) <i>Component Static Complexity</i>	

7. Variable para clasificar los componentes que considera la probabilidad de fallo en el funcionamiento del componente dentro de cierto escenario operacional.

<input type="radio"/> a) Confiabilidad	<input type="radio"/> b) Frecuencia reuso
<input type="radio"/> c) Mantenibilidad	

8. Característica que debe cumplir una buena práctica de codificación, que señala que la codificación debe seguir un estilo y organización de codificación que haga el código más fácil de leer.

<input type="radio"/> a) Correcto	<input type="radio"/> b) Consistente
<input type="radio"/> c) Moderno	

9. Estándar general de codificación que señala que no se debe hacer uso de tabuladores.

<input type="radio"/> a) Claridad y consistencia	<input type="radio"/> b) Utilizando librerías
<input type="radio"/> c) Formato y estilo	



10. Estándar general de codificación que señala que todas las funciones públicas y no públicas que no sean triviales se deben comentar en el banner.

a) Comentarios entre líneas

b) Comentarios en clase

c) Comentarios en funciones

# RESPUESTAS

## EXAMEN DE AUTOEVALUACIÓN



En este apartado encontrarás las respuestas al examen por unidad.

Unidad 4	
I. Solución	
1.	<b>b</b>
2.	<b>b</b>
3.	<b>c</b>
4.	<b>c</b>
5.	<b>a</b>
6.	<b>c</b>
7.	<b>a</b>
8.	<b>b</b>
9.	<b>c</b>
10.	<b>c</b>

# **Integración de subsistemas y sistemas**

## OBJETIVO PARTICULAR

Integrar los subsistemas de un sistema y validar su buen funcionamiento.

## TEMARIO DETALLADO (10 horas)

### **5. integración de subsistemas y sistemas**

- 5.1. Tipos de integración de sistemas
  - 5.2. Pruebas de integración
  - 5.3. Métricas para medir la calidad de sistemas
  - 5.4. Generación de documentación
-

# ACTIVIDAD DIAGNÓSTICA

## LO QUE SÉ



*Adjuntar archivo.*

Responde las siguientes preguntas sin apoyarte de ningún material.

1. ¿Qué entiendes por integración de sistemas?
2. ¿Cuál es tu concepto de pruebas de integración?
3. ¿Cómo medirías la calidad del software?
4. ¿Qué función tiene la documentación de un sistema?

## ACTIVIDADES DE APRENDIZAJE



**Unidad 5, actividad inicial. *Adjuntar archivo.*** A partir del estudio de la bibliografía específica sugerida, elabora un mapa conceptual u [organizador gráfico](#) con los temas de la unidad. Puedes auxiliarte de algunos programas como Mindjet [MindManager](#).

- 1. Unidad 5, actividad 1. *Adjuntar archivo.*** Elabora un cuadro sinóptico que considere los tipos de integración, las estrategias de integración y las técnicas de integración descritas en el tema "Tipos de integración de sistemas" del documento base de la unidad.
- 2. Unidad 5, actividad 2. *Adjuntar archivo.*** Elabora dos diagramas de flujo, uno para el proceso de la prueba ascendente y otro para el proceso de la prueba de humo descritos en el tema "Pruebas de integración" del documento base de la unidad. Así mismo, redacta una reflexión en torno a la importancia de seguir adecuadamente estos procesos y qué sucedería si se falla o salta alguna parte del proceso.
- 3. Unidad 5, actividad 3. *Adjuntar archivo.*** Investiga en internet todo lo referente a la norma ISO 9126 y en un procesador de textos elabora un resumen de la norma en no más de tres cuartillas. Así mismo, redacta una reflexión en torno a la importancia de esta norma para la integración de subsistemas y sistemas. No olvides incluir referencias.
- 4. Unidad 5, actividad 4. *Adjuntar archivo.*** Elabora un mapa mental que abarque todos los elementos considerados en el tema "Generación de documentación" del documento base de la Unidad.



5. **Unidad 5, actividad complementaria.** *Adjuntar archivo.* A partir del estudio de la unidad, realiza la siguiente actividad, misma que encontraras en el foro de la asignatura. Cabe señalar que esta será colocada en el foro por tu asesor.

# ACTIVIDAD INTEGRADORA

## LO QUE APRENDÍ



*Adjuntar archivo.*

Elige una [implementación de CORBA](#), instálala, y revisa sus características (generalmente vienen en el archivo readme.txt.) Elabora un resumen de las mismas y anexa una reflexión del proceso realizado y del valor que tiene CORBA en la integración de sistemas.

Corba<sup>®</sup>  
<http://www.corba.org/corbadownloads.htm>  
1997 - 2016

# CUESTIONARIO DE REFORZAMIENTO



**Adjuntar archivo. Responde las siguientes preguntas.**

1. Describe brevemente en qué consiste DCOM
2. ¿Cuáles son los conflictos que pueden aparecer cuando se implementa una aplicación distribuida en una red?
3. Describe en qué consiste la independencia de lenguaje de DCOM
4. ¿Qué es CORBA?
5. ¿Cuáles son las tres partes que constituyen a CORBA?
6. ¿Qué es Java?
7. ¿Cuáles son los objetivos de las pruebas?
8. ¿Cuáles son las características del software fácil de probar?
9. ¿Qué es la prueba de integración?
10. ¿Qué son las métricas?
11. ¿Cuáles son los atributos clave de calidad del estándar ISO 9126?
12. Menciona la clasificación de la documentación

# EXAMEN PARCIAL

## (de autoevaluación)



### I. *Selecciona la respuesta correcta.*

1. Mecanismo de integración de Microsoft donde se pueden utilizar los componentes creados en aplicaciones basadas en COM y trasladarlos a entornos distribuidos,

<input type="radio"/> a) COM	<input type="radio"/> b) DCOM
<input type="radio"/> c) COM+	

2. Con esta característica de DCOM la aplicación puede combinar componentes relacionados en máquinas cercanas entre sí, en una sola máquina o incluso en el mismo proceso.

<input type="radio"/> a) Independencia de la localización	<input type="radio"/> b) Localización del componente
<input type="radio"/> c) Control de componentes	

3. ¿Es necesario un lenguaje en específico para crear componentes COM?

<input type="radio"/> a) Sí	<input type="radio"/> b) No
-----------------------------	-----------------------------

4. Una de sus características es que se pueden ejecutar en múltiples plataformas,

<input type="radio"/> a) Componente	<input type="radio"/> b) Middleware
<input type="radio"/> c) Módulo	

5. Es una parte de CORBA.

- |   |   |
|---|---|
| <input type="radio"/> a) Conjunto de interfaces de invocación | <input type="radio"/> b) Objetos estandarizados |
| <input type="radio"/> c) Funciones interoperables             |   |

6. Elemento de Java que proporciona la capacidad para llamadas a métodos sobre objetos remotos.

- |  |                                   |
|--|-----------------------------------|
| <input type="radio"/> a) Java Remote Method Invocation (RMI) | <input type="radio"/> b) Java IDL |
| <input type="radio"/> c) JavaBean                            |                                   |

7. Principio de la prueba que señala que establece que la definición detallada de los casos de prueba puede empezar tan pronto como el modelo de diseño se ha consolidado.

- |   |   |
|---|---|
| <input type="radio"/> a) Seguimiento hasta los requisitos del cliente       | <input type="radio"/> b) Planificación de pruebas |
| <input type="radio"/> c) Empezar por lo pequeño y progresar hacia lo grande |   |

8. Característica del software fácil de probar que señala “Lo que ves es lo que pruebas”.

- |  |   |
|--|---|
| <input type="radio"/> a) Operatividad    | <input type="radio"/> b) Observabilidad |
| <input type="radio"/> c) Controlabilidad |   |

9. Prueba de integración que consiste de un planteamiento incremental hacia la construcción de la estructura del programa.

- |  |   |
|--|---|
| <input type="radio"/> a) Integración descendente | <input type="radio"/> b) Integración ascendente |
| <input type="radio"/> c) De regresión            |   |

10. Factor que afecta la calidad del software que mide el esfuerzo necesario para modificar un programa que ya está en funcionamiento.

- |  |                                     |
|--|-------------------------------------|
| <input type="radio"/> a) Facilidad de prueba | <input type="radio"/> b) Usabilidad |
| <input type="radio"/> c) Flexibilidad        |                                     |

11. Atributo de calidad del estándar ISP 9126 que mide el grado en que el software satisface las necesidades de idoneidad, corrección, interoperatividad, conformidad y seguridad.

- |                                       |  |
|---------------------------------------|--|
| <input type="radio"/> a) Usabilidad   | <input type="radio"/> b) Funcionalidad |
| <input type="radio"/> c) Flexibilidad |  |

12. Es la documentación que se utiliza para el propio desarrollo del producto y para su mantenimiento futuro.

- |   |  |
|---|--|
| <input type="radio"/> a) Documentación para los desarrolladores                   | <input type="radio"/> b) Documentación para los usuarios |
| <input type="radio"/> c) Documentación para los administradores o soporte técnico |  |

# RESPUESTAS

## EXAMEN DE AUTOEVALUACIÓN



En este apartado encontrarás las respuestas al examen por unidad.

Unidad 5	
I. Solución	
1.	<b>b</b>
2.	<b>a</b>
3.	<b>b</b>
4.	<b>b</b>
5.	<b>a</b>
6.	<b>a</b>
7.	<b>b</b>
8.	<b>b</b>
9.	<b>a</b>
10.	<b>c</b>
11.	<b>b</b>
12.	<b>a</b>



**Facultad de Contaduría y Administración**  
**Sistema Universidad Abierta y Educación a Distancia**