

**Ejercicios de la Unidad 01. Bloque II - Fases de Desarrollo - Soluciones****Análisis**

1. ¿Qué es fundamental para poder resolver un problema?

*Entenderlo y comprenderlo*

2. ¿Qué se especifica en la fase del análisis?

*Los requisitos*

3. ¿Qué dificultades nos encontramos a la hora de realizar el análisis?

*El cliente no sabe exactamente qué quiere*

*No sabe expresarse en términos informáticos*

*Cliente no tiene conocimientos*

*Pueden surgir nuevos requisitos*

4. ¿Qué técnicas se pueden utilizar para facilitar la comunicación con el cliente?

*Entrevistas, Brainstorming, prototipos, casos de uso.*

5. ¿Qué técnica podemos utilizar para clarificar dudas iniciales del cliente y además puede utilizarse como primera versión del programa?

*El prototipo*

6. ¿De qué técnicas disponemos para representar los requisitos?

*DFD*

*DFC*


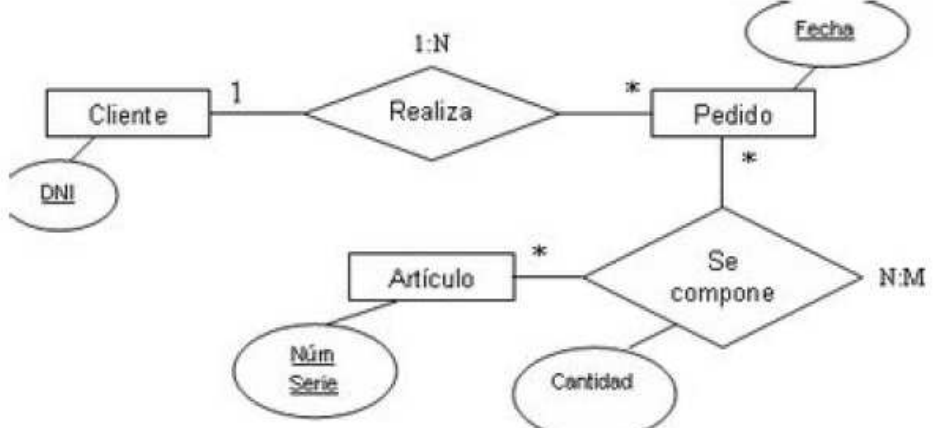
*Diagrama EntidadRelación*

*Diccionario de datos*

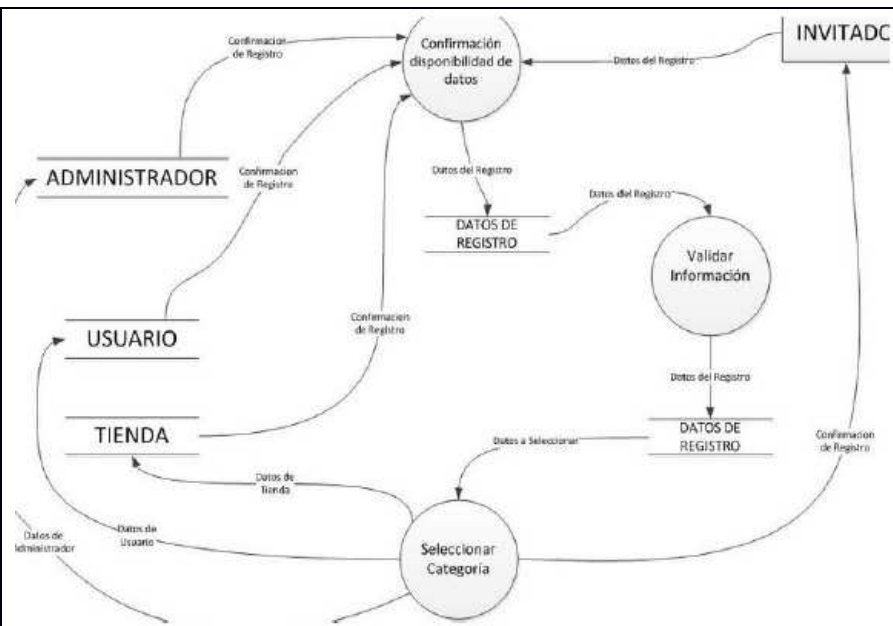
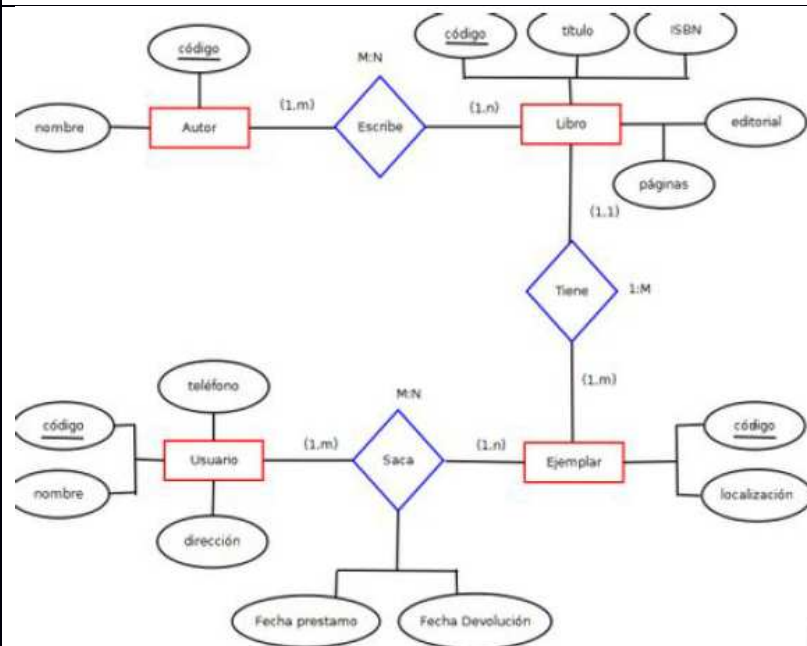
7. ¿Toda la documentación generada en el análisis en qué documento se deberá recoger?

*Especificación de requisitos software*

8. Rellena la tabla

Técnica	Representación Gráfica
<p style="color: red; text-align: center;">DFD</p>	
<p style="color: red;">Diagrama Entidad Relación</p>	

## DFD

Diagrama  
Entidad  
Relación

## Diseño

1. ¿Qué establece el diseño?

*La forma en la que se solucionará el problema*

2. ¿Cuáles son los dos principales tipos de diseño?

**Diseño estructurado:** *se basa en el flujo de datos a través del sistema.*

**Diseño orientado a objetos:** *se entiende el sistema como un conjunto de objetos formados por propiedades y métodos que interactúan entre sí.*

3. ¿Qué tres construcciones están en los fundamentos del diseño estructurado?

*Secuencial, condicional y repetitiva*

4. ¿Qué es un algoritmo?

*Serie de pasos organizados que describen el proceso que se debe seguir para dar solución a un problema.*

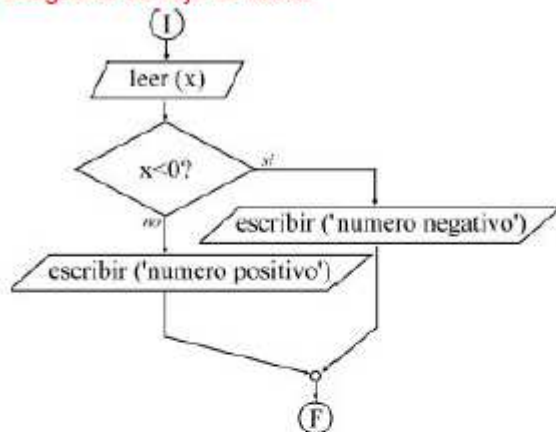
5. ¿Qué dos técnicas se utilizan en el diseño de algoritmos?

*Diagramas de flujos y pseudocódigo*

6. Realiza el diseño del algoritmo mediante diagramas de flujo y pseudocódigo para el siguiente problema:

*Dado un número decir si es positivo o negativo.*

## Diagrama de flujo de datos



## Algoritmo

Pseudocódigo

Algoritmo Positivo\_Negativo

Variables

x: Entero

Inicio

Leer (x)

Si (x &lt; 0) entonces

Escribir ('Numero negativo')

sino

Escribir ('Numero positivo')

Fin\_si

Fin

**Codificación y pruebas**

1. ¿Qué realiza el programador en la fase de codificación?

*Transforma las especificaciones del diseño en un conjunto de instrucciones escritas en un lenguaje de programación, almacenadas dentro de un programa.*

2. ¿Cómo se llama el conjunto de instrucciones generado en la fase de codificación?

*Código fuente*

3. Una vez obtenemos el código fuente, ¿lo podemos ejecutar?

*No, debemos traducirlo a lenguaje máquina y posteriormente enlazarlo para obtener el ejecutable.*

Lee el documento ConvencionesJava.pdf y contesta

4. ¿Por qué Convenciones de código?

*Las convenciones de código son importantes para los programadores por un gran número de razones:*

- *El 80% del coste del código de un programa va a su mantenimiento.*
- *Casi ningún software lo mantiene toda su vida el autor original.*
- *Las convenciones de código mejoran la lectura del software, permitiendo entender código nuevo mucho más rápidamente y más a fondo.*
- *Si distribuyes tu código fuente como un producto, necesitas asegurarte de que está bien hecho y presentado como cualquier otro producto.*
- *Para que funcionen las convenciones, cada persona que escribe software debe seguir la convención. Todos.*

5.- Comenta 5 convenciones que te llamen la atención

*Algunas de las reglas tienen que ver con contenidos de Java que todavía no has estudiado. Céntrate en las que ya conoces*

6. ¿Cuál es el objetivo de la fase de pruebas?

*Planificar y diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores*

7. ¿Cuándo consideraremos que ha tenido éxito una prueba?

*Cuando se localice un error.*

8. ¿Qué es un caso de prueba?

*Un caso de prueba es un documento que especifica valores de entrada, salida esperada y condiciones de la ejecución de la prueba.*

9. Marca como verdadero falso las siguientes afirmaciones.

Afirmación	V o F
Cada prueba debe definir unos resultados esperados	V
Una prueba tiene éxito si no hay errores ya que demuestra que el software funciona correctamente	F
El mejor probador es el propio programador ya que conoce donde pueden estar los errores en su código	F
Las pruebas deben incluir únicamente datos de entrada válidos y esperados	F
Se debe evitar hacer pruebas sin documentar	V
Uno de los dos objetivos en los que nos debemos centrar es comprobar que el software no hace lo que debe hacer	V
Uno de los dos objetivos en los que nos debemos centrar es comprobar que el software hace lo que no debe hacer	V
Es importante comprobar que el software no hace lo que no debe hacer	V
Un caso de prueba es un documento que especifica valores de entrada, salida esperada y condiciones de la ejecución de la prueba	V

## Mantenimiento

1. Explica en qué consiste el mantenimiento.

*La modificación de un producto de software después de la entrega al cliente para corregir fallos, mejorar el producto, o adaptar a un entorno modificado.*

2. ¿Cuándo termina el mantenimiento?

*Cuando el software deja de utilizarse. Es la última fase del ciclo de vida.*

3. ¿Qué cuatro tipos de mantenimiento existen?

- *Adaptativo.*
- *Correctivo*
- *Perfectivo*
- *Preventivo*

4. Rellena la tabla

Hechos	Tipo de mantenimiento
La empresa migra de Windows a Linux sus equipos	Adaptativo
El cliente piensa en añadir un nuevo gráfico que le ayudará en la toma de decisiones	Perfectivo
Tras entrega el software el cliente se da cuenta que nuestro software muestra la contraseña sin codificar al introducirla.	Correctivo
Añadimos comentarios en la entrada de cada programa con la fecha y desarrollador	Preventivo