

## UF1. [PAC01] SOLUCIÓN

Para responder a las siguientes cuestiones deberéis ayudarte del libro de texto, capítulo 1, y consultar, si lo creéis necesario, internet.

Debéis subir un único archivo comprimido que contenga **UN ÚNICO DOCUMENTO PDF** con las respuestas de la parte teórica, con los pseudocódigos e imágenes de los diagramas de flujo de la parte práctica.

Todas las preguntas valen lo mismo, 1 punto cada una, por lo que cada pregunta del tipo test tiene un valor de 0,33 puntos.

Las preguntas tipo test se corregirán sólo en la caja que veréis después de las preguntas.

Recordad que la fecha límite para la entrega de esta PAC es el día 17 de OCTUBRE.

**Información:** Cualquier PAC copiada y/o en la que se haya utilizado “copy-paste” de código ya escrito será puntuada con un 0.

## Actividades

### Parte teórica

1. Responde a este test en la matriz de respuestas que se encuentra al final de las preguntas:
  - 1.1. ¿Cuál de las siguientes afirmaciones es correcta?
    - a) En los modelos evolutivos no se necesita conocer todos los requisitos al comienzo.
    - b) Es muy común en el modelo en cascada el uso de prototipos.
    - c) El análisis de riesgos se lleva a cabo en cada incremento del modelo iterativo incremental.
    - d) El modelo en cascada es apropiado cuando se necesita una versión inicial del software a desarrollar.
  - 1.2. ¿Cuál de estas afirmaciones sobre la fase de diseño en el desarrollo de una aplicación es correcta?
    - a) En esta fase se especifica qué hay que hacer.

- b) En esta fase se especifica cómo hacerlo.
- c) En esta fase se realiza el proceso de programación.
- d) En esta fase se realizan las pruebas.

1.3. Cuáles de las siguientes afirmaciones sobre la máquina virtual de Java es cierta:

- a) Un fichero .class contiene código en un lenguaje máquina.
- b) La máquina virtual de Java toma y traduce el bytecode en código binario.
- c) Los ficheros .class sólo pueden ser ejecutados en Microsoft Windows y Linux.
- d) A la hora de instalar el entorno de ejecución de la máquina virtual Java necesitamos saber en qué sistema operativo se va a instalar.

#### **MATRIZ DE RESPUESTAS**

1.1	1.2	1.3
A	B	B, D

2. ¿Qué diferencias hay entre el freeware y el software libre?

El software libre es aquel en el que el autor cede una serie de libertades básicas al usuario en el marco de una licencia y establece la libertad para utilizar el programa en varios ordenadores, estudiar cómo funciona y adaptar su código, libertad para distribuir copias a otros usuarios y libertad para mejorar el programa. El software libre deja acceder al código fuente para poder realizar todo lo anterior.

El freeware sin embargo acceder se distribuye sin cargo y normalmente no incluye el código fuente. Suele incluir una licencia de uso que lo restringe. Normalmente no se puede modificar, ni venderla y siempre hay que dar cuenta a su autor. Los programas de software libre no necesariamente son freeware.

3. Califica las siguientes metodologías según el modelo de ciclo de vida.

- Scrum
- Xtreme Programming
- DSDM

Scrum: Se denominan así los marcos de desarrollo ágiles (o metodologías ágiles). Se caracterizan por:

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos autoorganizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o de cascada. Estaría basado en un modelo iterativo incremental.

Xtreme Programming: Pertenece también al modelo iterativo incremental. Se caracteriza por:

- Pruebas unitarias continuas: frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto.
- Frecuente integración del equipo de programación con el cliente o usuario.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Retrofactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto.
- Simplicidad en el código.

DSDM: El método de desarrollo de sistemas dinámicos (en inglés Dynamic Systems Development Method) es un método apoyado por su continua implicación del usuario en un desarrollo iterativo y creciente que sea sensible a los requerimientos cambiantes, para desarrollar un sistema que reúna las necesidades de la empresa en tiempo y presupuesto. Tiene semejanzas en el proceso y el concepto con Scrum. Sus principales características son:

- Involucrar al cliente.
- El equipo del proyecto tiene el poder para tomar decisiones.
- Entrega frecuente de los productos.
- Entregar sistemas que satisfacen las necesidades actuales del negocio.
- Desarrollo interactivo e incremental.
- Cambios reversibles.
- Las pruebas son realizadas durante todo el ciclo vital del proyecto.

4. Para evitar sorpresas al hacer un proyecto, ¿En qué fases del desarrollo nos tenemos que centrar más? Justifica tu respuesta.

Para evitar sorpresas y obtener un proyecto satisfactorio la fase en la que nos tendremos que centrar más es la fase de Análisis. Es la fase más importante para la obtención de éxito en el proyecto. En ella nos centraremos en entender y comprender el problema que se necesita resolver para después darle una solución. En esta fase se analizan y especifican los requisitos que el programa o sistema deben resolver. La obtención de estos requisitos no es fácil, pero existen numerosas técnicas como las entrevistas, el desarrollo conjunto de aplicaciones, el brainstorming, etc.

5. ¿El lenguaje Java es un lenguaje compilado o interpretado (u otra opción)? Justifica tu respuesta.

A diferencia de otros lenguajes, Java es compilado e interpretado. Java es compilado cuando su código fuente es traducido a un lenguaje objeto llamado código de máquina (binario, bytecodes) y es interpretado debido a que el código máquina puede ser ejecutado sobre cualquier plataforma la cual debe tener un intérprete ejecutándolo en tiempo real.

6. ¿Qué diferencias hay entre un compilador y un intérprete?

Un compilador es un programa que puede leer un programa escrito en un determinado lenguaje (lenguaje fuente) y traducirlo en un programa equivalente en otro lenguaje (lenguaje destino).

Un intérprete son otra alternativa para traducir los programas escritos en lenguaje de alto nivel, pero en vez de producir un programa destino nos da la apariencia de ejecutar directamente las operaciones especificadas en el programa fuente con las entradas proporcionadas por el usuario.

Los compiladores difieren de los intérpretes en varios aspectos:

- Un programa que ha sido compilado puede correr por sí solo, pues en el proceso de compilación se transformó en otro lenguaje (lenguaje máquina).
- Un intérprete traduce el programa cuando lo lee, convirtiendo el código del programa directamente en acciones.
- La ventaja del intérprete es que dado cualquier programa se puede interpretarlo en cualquier plataforma (sistema operativo), en cambio el archivo

generado por el compilador solo funciona en la plataforma en donde se lo ha creado.

- Pero por otro lado, un archivo compilado puede ser distribuido fácilmente conociendo la plataforma, mientras que en un archivo interpretado no funciona si no se tiene el intérprete.
- Hablando de la velocidad de ejecución un archivo compilado es de 10 a 20 veces más rápido que un archivo interpretado.

#### 7. ¿De qué tres elementos consta un lenguaje de programación?

Todo lenguaje de programación consta de un alfabeto o vocabulario, también llamado léxico, una sintaxis y una semántica.

- El léxico está formado por el conjunto de símbolos permitidos.
- La sintaxis son las reglas que indican cómo realizar las construcciones con los símbolos del lenguaje.
- La semántica son las reglas que determinan el significado de cualquier construcción del lenguaje.

#### 8. Cita alguna ventaja de usar lenguajes de programación ejecutados por máquinas virtuales.

Los lenguajes de programación ejecutados por máquinas virtuales se pueden ejecutar en cualquier plataforma, es decir, pueden ejecutarse en entornos UNIX, Mac, Windows, etc. Esto es una gran ventaja porque nos permite generar un solo programa que pueda ejecutarse en varios sistemas operativos ya que el encargado de ejecutar el código es la máquina virtual. Un ejemplo sería la JVM (Java Virtual Machine) que es la máquina virtual de Java.

### Parte práctica

#### 9. Realiza el pseudocódigo y diagrama de flujo de un programa que diferencie los números pares de los impares.

##### PSEUDOCÓDIGO

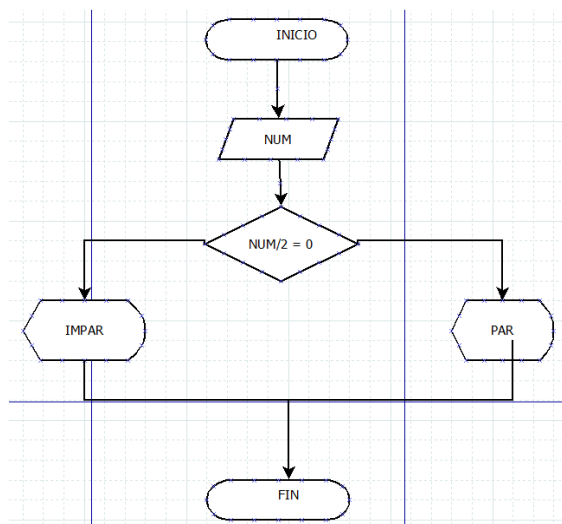
Inicio

```
Leer NUM
Si NUM%2 = 0 Entonces
  Visualizar "PAR"
Si no
  Visualizar "IMPAR"
```

Fin si

Fin

## DIAGRAMA DE FLUJO



10. Escriba un programa en pseudocódigo y el diagrama de flujo para la gestión de un call center de soporte de una empresa de telefonía. Esta empresa tiene 3 niveles de helpdesk para solucionar las incidencias de los clientes. El operador debe atender la llamada, comprobar que sea cliente de su compañía y que esté autorizado en el caso que el cliente sea una empresa y no el mismo. Si el Operador de nivel 1 no puede resolver el incidente lo escalará a nivel 2 y este a nivel hasta que se resuelva.

## PSEUDOCÓDIGO

Declarar solución = false

## Declarar N Llamada

## Declarar N Cliente

Visualizar “¿Con quién hablo porfavor?”

## Leer N Llamada

## Abrir Fichero Clientes

Leer Registro(Nombre, Empresa, Autorizado)

Si N Llamada = Nombre

Si Empresa = false Entonces

Si no

Visualizar “¿Su nombre completo?”

Leer N Cliente

Si N Cliente=Autorizado Entonces

Si no

Visualizar “Sentimos no poder ayudarle”

Fin si

Fin si

Visualizar “N1: ¿Se solucionó el incidente?”

Leer solución

Si solución=false Entonces

## Visualizar “Transferimos a N2”

Visualizar “N2: ¿Se solucionó el incidente?”

Leer solución

Si solución=false Entonces

### Visualizar “Transferimos a N3”

## Repetir

### Visualizar “N3: ¿Solucionó la incidencia?”

[Leer solución](#)

Hasta solución=true

Fin si

Fin si

Si no

### Visualizar “Sentimos no poder ayudarle”

Fin si

## Visualizar “Gracias por utilizar nuestro servicio”

## Cerrar Fichero Clientes

Fin

## DIAGRAMA DE FLUJO

