

## CICLO DE VIDA

### **PARTE 1. Metodologías de desarrollo de software.**

#### **1. Sistemas de información.**

- a). ¿Cuál es el objetivo último de un programa informático?
- b). ¿Qué entendemos en informática como *empresa*?
- c). ¿A qué llamamos sistema de información?
- d). ¿Cuáles son los elementos o componentes principales de un sistema de información?

#### **2. Ingeniería del software. Etapas.**

- a). ¿De qué se hace necesario disponer?
- b). ¿De qué se encarga la ingeniería industrial?
- c). En ingeniería del software ¿Qué cuatro etapas o generaciones se consideran?

#### **3. Ciclo de vida del desarrollo del software.**

- a). ¿Qué entendemos como ciclo de vida de una aplicación o proyecto informático?
- b). ¿Cuáles son sus principales etapas?
- c). ¿Cuáles son los distintos tipos de ciclo de vida que existen?

#### **4. Metodología de desarrollo de software.**

- a). ¿Qué tipo de CV podemos elegir según las características del proyecto?
- b). ¿Qué metodologías oficiales existen?
- c). ¿Cuáles son los distintos tipos de ciclo de vida que existen?

### **PARTE 2. Ciclo de vida de desarrollo de software.**

#### **1. Especificación de requisitos.**

- a). ¿Cuál es el objetivo de esta etapa?
- b). Explica brevemente las fases de esta etapa.

#### **2. Análisis.**

a). ¿Qué pretendemos con el análisis? ¿Cuál es su objetivo? ¿Cuáles son sus fases?

b). ¿Qué entendemos como proceso y datos? Explica sus modelos.

### **2.1. Análisis de procesos.**

a). ¿Qué describe el modelo de procesos?

b). ¿Qué elementos forman el diagrama de flujo?

c). ¿Para qué se utiliza el diccionario de datos?

d). ¿Cómo se clasifican los tipos de entradas?

### **2.2. Análisis de datos.**

a). ¿Qué entendemos como datos?

b). ¿Qué refleja el modelo de datos?

c). ¿Qué elementos forman el modelo E/R?

## **3. Diseño.**

a). ¿Cuáles son los objetivos del diseño?

b). ¿Cuáles son sus principales etapas?

### **3.1. Diseño de procesos.**

a). ¿Qué es un módulo?

b). ¿Qué representa una conexión?

### **3.1. Diseño de datos.**

a). ¿Cuál es la idea en el diseño de datos?

## **4. Implementación.**

a). ¿Qué se pretende con la implementación?

b). ¿En qué consiste la implementación funcional del sistema?

c). ¿Qué implica la implementación de los datos?

## **PARTE 3. Depuración. Herramientas CASE.**

### **1. Depuración.**

a). ¿En qué consiste depurar un programa? ¿Cuándo se dice que un programa está depurado?

b). En el proceso de depuración, ¿qué es hacer la traza de un programa?

c). ¿Qué permite el *debugger*?

### **1.1. Puntos de interrupción (breakpoints).**

a). ¿Qué son? ¿Qué permite esta detección? ¿Qué permitirá después al programador?

b). ¿Cómo se simbolizan?

### **1.2. Ejecución paso a paso.**

a). ¿Qué otra opción poseen los depuradores?

b). ¿Qué permitirán estas opciones?

### **1.3. Ventanas de depuración: traza de un programa.**

a). ¿Qué ofrece el depurador al programador? ¿Para qué?

b). ¿Qué tipos de ventanas tenemos?

## **2. Herramientas CASE.**

a). ¿Qué son estas herramientas? ¿Qué permiten? ¿Cuándo se pueden aplicar?

b). ¿De qué consta un entorno CASE?

c). ¿Cuáles son los objetivos esenciales?

### **2.1. Tipos de herramientas CASE.**

a). ¿En qué categorías se dividen? Realiza un esquema.

b). ¿Qué denominador común tienen estas herramientas?

### **2.2. Características.**

a). ¿En qué ayudan y qué permiten?

b). ¿Cuál es la característica y la propiedad más importante?

c). ¿Qué no suele faltar y qué suelen aportar?

d). ¿Qué entendemos como la gestión del control de versiones?

### **3. Lenguaje Unificado y Modelado.**

- a). ¿Qué es el *UML*? ¿A qué ayuda?
- b). Enumera los elementos por los que están formados.

## **PARTE 4. Trabajo en equipo y control de versiones.**

- a). ¿En qué consiste el trabajo en equipo? ¿Qué ocurre cuando se trabaja en equipo? ¿Qué se necesita?
- b). ¿De qué se encarga el equipo de mejora continua?
- c). ¿En qué consiste el control de versiones?

### **1. Trabajo en equipo.**

- a). Explica los roles del trabajo en equipo.

#### **1.1. Divide y vencerás.**

- a). ¿En qué se basa esta idea?
- b). ¿Qué es la programación modular? ¿Qué propicia?

#### **1.2. Mejora o morirás.**

- a). ¿Qué es lo habitual que debemos hacer al trabajar en equipo?

#### **1.3. Y ahora...; Cómo lo hacemos?**

- a). ¿De qué dependerá el éxito o fracaso en la realización de un proyecto?

### **2. Control de versiones.**

- a). ¿Qué tendremos en el repositorio? ¿Dónde suele estar? ¿Qué se almacena en ellos?

#### **2.1. Tipos de repositorios.**

- a). Explica que tipos existen.
- b). Indica en qué consisten los programas distribuidos y centralizados.
- c). ¿Qué entendemos como repositorios en Institucionales, temáticos o de datos?

#### **2.2. Vocabulario y conceptos esenciales.**

a). ¿Qué entendemos como *Módulo*? ¿En qué consiste rotularlo? ¿Qué significa congelar?

b). ¿Qué función realiza un *Despliegue*?

c). ¿Cuándo se crea un *Commit*?

d). ¿Cuándo se produce un *Conflicto*?

e). ¿En qué consiste una *Importación* y una *Exportación*?

f). ¿Qué será una *Fusión* y una *Actualización*?