Unidad 5. Fases de desarrollo/ejecución del software

Contenido

[1. Análisis 2](#_Toc84940689)

[a. Mapa conceptual 2](#_Toc84940690)

[b. Preguntas y dudas 2](#_Toc84940691)

[2. Diseño 6](#_Toc84940692)

[a. Mapa conceptual 6](#_Toc84940693)

[b. Preguntas y dudas 6](#_Toc84940694)

[1) Diseño estructurado 6](#_Toc84940695)

[2) Diseño orientado a objetos 7](#_Toc84940696)

[3. Codificación 8](#_Toc84940697)

[a. Mapa conceptual 8](#_Toc84940698)

[b. Preguntas y dudas 8](#_Toc84940699)

[1) Fases en la obtención del código 8](#_Toc84940700)

[2) Máquinas virtuales 10](#_Toc84940701)

[3) SDG 10](#_Toc84940702)

[4) Entornos de ejecución 10](#_Toc84940703)

[5) Java RuntimeEnviroment 11](#_Toc84940704)

[4. Pruebas 12](#_Toc84940705)

[a. Mapa conceptual 12](#_Toc84940706)

[b. Preguntas y dudas 12](#_Toc84940707)

[5. Documentación 13](#_Toc84940708)

[a. Mapa conceptual 13](#_Toc84940709)

[b. Preguntas y dudas 13](#_Toc84940710)

[6. Explotación 14](#_Toc84940711)

[a. Mapa conceptual 14](#_Toc84940712)

[b. Preguntas y dudas 14](#_Toc84940713)

[7. Mantenimiento 14](#_Toc84940714)

[a. Mapa conceptual 14](#_Toc84940715)

[b. Preguntas y dudas 14](#_Toc84940716)

# Análisis

## Mapa conceptual

## Preguntas y dudas

1. **¿Qué dos tipos de requisitos establece el análisis?**

En el análisis se establecen los requisitos funcionales y no funcionales del sistema.

1. **¿Qué se establece en la fase de análisis?**

En esta fase, se establece los pasos a seguir para conseguir el producto final deseado por el cliente, a partir de los requisitos que este establezca.

1. **¿En qué técnica las entrevistas están dirigidas a la alta dirección?**

La técnica dirigida para la alta dirección es la planificación conjunta de requisitos.

1. **¿Qué se obtiene al terminar la fase de análisis?**

Al terminar esta, obtenemos el documente ERS, el cual es un informe de las especificaciones de requisitos del software.

1. **¿En qué técnica participan muchos tipos de miembros del proyecto y se apoya en la dinámica de grupos?**

En la técnica de “Desarrollo conjunto de aplicaciones”

1. **¿Qué es un requisito funcional?**

El requisito funcional es el conjunto de los servicios que el sistema debe proporcionar, es decir, las funciones que va a realizar la aplicación del cliente junto con su respuesta y comportamiento en situaciones inesperadas.

1. **¿Qué es un requisito no funcional?**

Consiste en el grupo de las restricciones que afectan al sistema.

1. **¿Un requisito que pide que el usuario pueda hacer una copia de seguridad de que tipo es?**

Este responde al tipo funcional, porque en ella, se le indica al sistema que debe hacer.

1. **Enumera las fases del desarrollo del software**

Las fases de desarrollo del software son los siguientes:

* Análisis
* Diseño
* Construcción
  + - * Codificación
      * Pruebas
      * Instalación
* Mantenimiento
* Validación y verificación

1. **¿Un requisito que pide que el usuario pueda visualizar en un mapa una ruta en coche de que tipo es?**

Este responde al tipo no funcional, ya que se le indica al sistema como debe hacer una tarea determinada ya previamente.

1. **¿Qué representan las entidades externas en un DFD?**

Las entidades externas representan a los componentes que no forman parte del sistema. Tienen forma de rectángulo.

1. **¿En qué técnica de representación de requisitos se representan con rectángulos las**

**entidades y con rombos la relaciones?**

Esta representación responde a la técnica “Diagrama Entidad/ Relación DER”.

1. **¿Qué representan los flujos de datos en un DFD?**

Los flujos de datos representan los movimientos de los datos del sistema, a través de flechas.

1. **¿Qué técnica consiste en hablar con el cliente?**

La entrevista, la técnica más tradicional, consiste en hablar con el cliente.

1. **¿Qué técnicas se utilizan en el análisis?**

Las técnicas que se emplean en el análisis son:

* Entrevista
* Desarrollo conjunto de aplicaciones
* Planificación conjunta de requisitos
* BrainStorming

1. **¿Qué técnica de representación de requisitos representa el flujo de los datos entre los**

**distintos procesos, entidades externas y almacenes que forman el sistema?**

Esa técnica de representación es denominada “Diagrama de flujo de datos DFD”

1. **¿Qué técnica de representación de requisitos describe detalladamente los datos utilizados por el sistema?**

La técnica que descripción con lujo de detalles los datos utilizados por el sistema es el “Diccionario de datos DD”

1. **¿Qué identifican los procesos en un DFD?**

Los procesos identifican las funciones dentro del sistema, en forma de burbujas ovaladas.

1. **¿Qué técnica de representación de requisitos utiliza escenarios para describir como se usa la aplicación?**

La técnica que se basa en escenarios que describen como se usa el software, es “Casos de uso”

1. **¿Un requisito que pide que la aplicación se pueda usar en dispositivos móviles de que tipo es?**

Se trata de un requisito funcional, al indicar que se debe hacer.

1. **¿En qué técnica se generan ideas desde diferentes puntos de vista?**

En la técnica del BrainStorming, los grupos de participantes generan ideas desde diferentes puntos de vista para la resolución de un problema.

1. **¿Qué representan los almacenes en un DFD?**

Los almacenes representan el lugar, desde el punto de vista estático, donde se guardan los datos y a su vez, donde se pueden recuperar.

1. **¿Qué técnicas se utilizan para representar los requisitos?**

Las técnicas empleadas para la representación de los requisitos son:

* Diagramas de flujo de datos (DFD)
* Diagrama Entidad/Relación (DER)
* Diccionario de Datos (DD)
* Prototipos
* Casos de uso
* Un modelo de dominio

1. **¿Qué especifica el documento de especificación de requisitos de software?**

En él, se especifican los siguientes puntos:

* Las reuniones que se van a realizar.
* Relación cliente-sistema, desde el punto de vista de los objetivos.
* Relación requisitos funcionales-no funcionales.
* Relación objetivos prioritarios-temporización.
* Reconocimiento de requisitos mal planteados o con contradicciones.

# 2. Diseño

## Mapa conceptual

## Preguntas y dudas

1. **¿Qué se realiza en esta fase?**

En esta fase, se traducen los requisitos funcionales y no funcionales, previamente definidos, en una representación del software.

1. **¿Qué dos tipos de diseños se pueden utilizar?**

Los dos tipos de diseño que se pueden emplear son el diseño estructurado y el orientado a objetos

### Diseño estructurado

1. **¿Qué cuatro niveles tiene el diseño clásico?**

Los cuatro niveles que dispone este diseño son los siguientes:

* Diseño de datos
* Diseño arquitectónico
* Diseño de la interfaz
* Diseño a nivel de componentes (procedimental)

1. **¿Qué fase del diseño de estructurado diseña las estructuras de datos?**

El diseño de datos, a partir de los diagramas DER y DD, transforma los datos y relaciones en estructuras de datos.

1. **¿Qué fase del diseño de estructurado detalla la estructura de cada componente?**

El diseño a nivel de componentes (procedimental) se encarga del detalle de la estructura de cada componente.

1. **¿Qué fase del diseño de estructurado divide el proyecto en módulos?**

Se trata del diseño arquitectónico.

1. **¿Qué fase del diseño estructurado crea los formatos de pantalla?**

El diseño de la interfaz crea los formatos de pantalla

1. **¿Qué fase del diseño de estructurado utiliza diagramas de flujo, pseudocódigo etc.?**

El diseño a nivel de componentes (procedimental) emplead estos elementos, junto con los diagramas de caja, tablas de decisión, etc.

1. **¿Qué fase del diseño de estructurado diseña la estructura de componentes de software?**

El diseño arquitectónico se encarga de la representación de las estructuras de los componentes del software, junto con sus propiedad e interacciones en base a los DFD.

### Diseño orientado a objetos

**¿Qué requisito debe cumplir la fase de análisis para poder realizar un diseño orientado a objetos?**

El análisis debe estar orientado a objetos (AOO) para que sea posible posteriormente realizar un diseño orientado a objetos.

1. **¿Qué cuatro capas define el diseño orientado a objetos?**

Los niveles que se encuentran dentro de este diseño son:

* Subsistema
* Clases y objetos
* Mensajes
* Responsabilidades

# 3. Codificación

## Mapa conceptual

## Preguntas y dudas

* 1. **¿Qué ventajas proporciona el escribir código fácil de leer?**

La principal ventaja que aporta que el código se escriba en una lectura fácil de leer, es que esto permite que su desarrollo y mantenimiento futuro sea más efectivo.

* 1. **¿Qué característica deseable del código requiere que el código haga lo especificado en los requisitos?**

La característica de la corrección es la que establece que el código realice lo que se pida realmente.

1. **¿Qué implica la programación eficiente?**

La programación eficiente supone el buen uso de los recursos.

1. **¿Qué permite un código portable?**

Esta característica permite que se pueda implementar el código en cualquier equipo.

1. **¿Qué característica deseable del código lo divide en trozos pequeños?**

La característica de modularidad establece que el código se divida en trozos pequeños.

### Fases en la obtención del código

1. **Define código fuente**

El código fuente es el conjunto de instrucciones que el programa deberá realizar, escritas por los programadores en un lenguaje de alto nivel, utilizando un editor de texto.

1. **Define algoritmo**

Un algoritmo es un conjunto de pasos a seguir para obtener la solución del problema.

1. **¿En qué lenguaje escribimos un código fuente y en que lenguaje debe recibirlo un ordenador?**

El código los escribimos en un inicio en nuestro lenguaje y después, lo traducimos al lenguaje máquina para que el ordenador pueda entenderlo.

1. **¿Cuáles son los pasos para obtener el código fuente de una aplicación?**

Los pasos necesarios para obtener el código fuente son:

* Las etapas de análisis y diseño.
* Diseño del algoritmo.
* Elección del “Lenguaje de Programación”.
* Codificación del algoritmo.

1. **¿Qué es el código objeto?**

El código objeto es un código intermedio, resultante del proceso de traducir el código fuente a un código binario que aún no puede ejecutarse en la computadora.

1. **¿Mediante que dos procesos se puede obtener un código objeto?**

El código objeto se pueden adquirir, a través de estos dos procesos:

* Compilación.
* Interpretación.

1. **¿En qué se diferencian la compilación y la interpretación?**

Estas se diferencian en los siguientes aspectos:

* Mientras que en el primero se traduce todo el código fuente en un solo paso, el segundo traducirá línea por línea.
* A consecuencia de la traducción masiva, el primero dispone de un código intermedio que deberá ser enlazado, a diferencia del segundo.
* El software responsable del primero es el compilador y del segundo, el intérprete.
* El segundo es recomendable para un programador inexperto, ya que, de esta forma, la detección de errores es más detallada.

1. **¿Cómo se denomina el proceso de unir todos los archivos de código objeto?**

El proceso de enlazar todos los archivos de código objeto, se denomina enlazar.

1. **¿Cómo se denomina el software que realiza el proceso de enlazado?**

El software empleado para este proceso es el “linker”.

1. **¿Se puede ejecutar el código obtenido del proceso de compilación?**

No se puede ejecutar este código, al ser uno de tipo objeto, que, en otras palabras, es código binario que no se puede ejecutar en el ordenador.

1. **¿Se puede ejecutar el código obtenido del proceso de enlazado?**

Este sí se puede ejecutar al haberse unido todos los códigos objetos, dando lugar a un código ejecutable.

### Máquinas virtuales

1. **¿Cuál es la función principal de las máquinas virtuales?**

La función principal de estos es separar el funcionamiento del ordenador de los componentes hardware instalados.

1. **¿Qué nos permite el uso de una máquina virtual?**

Con estas máquinas, podemos desarrollar y ejecutar una aplicación sobre cualquier equipo, independientemente de las características de los componentes físicos.

1. **¿Cuáles son las funciones principales de una máquina virtual?**

Las funciones principales de las máquinas virtuales son:

* Portabilidad de las aplicaciones.
* Reserva memoria empleada y liberación de memoria.
* Comunicación máquina-sistema para el control de los dispositivos hardware.
* Cumplimiento de las normas de seguridad.

### SDG

### Entornos de ejecución

1. **¿Qué es un entorno de ejecución?**

Un entorno de ejecución es un servicio de la máquina virtual que sirve como base software para la ejecución de programas.

1. **¿De qué partes consta el entorno de ejecución?**

Este consta de la máquina virtual y los API’s, que se suelen distribuir conjuntamente.

1. **¿Qué es el runtime?**

Se le denomina runtime al tiempo que el programa tarda en ejecutarse en el ordenador.

1. **¿Cuáles son las funciones de los entornos de ejecución?**

Las funciones que tienen los entornos de ejecución son:

* Configuración de la memoria principal del sistema
* Enlazamiento de los archivos del programa con las bibliotecas y subprogramas existentes y realizados.
* Depuración de los programas

1. **¿Por qué la máquina virtual y las API’s se suelen distribuir juntas?**

Estas precisan distribuirse conjuntamente porque estas deben ser compatibles entre sí.

1. **¿A quién pertenece el entorno de ejecución?**

Este elemento le pertenece, en general, al propio sistema operativo, aunque en ocasiones puede ser un software independiente que funciona por debajo de la aplicación.

### Java RuntimeEnviroment

1. **¿Qué es el Java RuntimeEnvironment (JRE)?**

El JRE es el conjunto de utilidades que permitirá la ejecución de programas Java sobre cualquier tipo de plataforma.

# 4. Pruebas

## Mapa conceptual

## Preguntas y dudas

**¿Las pruebas de caja negra y caja blanca son pruebas de verificación o de validación?**

Las pruebas de caja negra y blanca son del tipo verificación, ya que se encargan de validar los requisitos funcionales y la estructura interna del programa.

**¿En qué tipo de pruebas se intenta comprobar si el software implementa correctamente la función para la que está diseñado?**

Este enunciado hace referencia a las pruebas de verificación.

**¿Qué pruebas se centran en validar los requisitos funcionales sin fijarse en el funcionamiento interno del programa?**

Se trata de las pruebas de caja negra, que únicamente se centran en los requisitos funcionales.

**¿Qué dos tipos de pruebas se realizan para comprobar que hemos construido el software de forma correcta?**

Los dos tipos de pruebas que se realizan para comprobar la correcta construcción del software son las pruebas de caja negra y blanca.

**¿En qué tipo de pruebas se intenta comprobar si el software cumple los requisitos del cliente?**

En la prueba de validación, nos encargamos de comprobar que el producto cumple con los requisitos del cliente.

**¿Qué pruebas se centran en validar la estructura interna del programa?**

Las pruebas de caja blanca se centran en lo descrito en el enunciado.

# 5. Documentación

## Mapa conceptual

## Preguntas y dudas

* 1. **¿En qué documento queda reflejada la información para la puesta en marcha de la aplicación?**

La información para la puesta en marcha se encuentra en la guía de instalación.

* 1. **¿En qué documento quedan reflejadas las pruebas realizadas?**

Las pruebas realizadas se muestran en la guía técnica.

* 1. **¿Cuál es el objetivo de la guía de instalación?**

El objetivo de esta guía es dar toda la información necesaria para garantizar que la implantación de la aplicación se realice de forma correcta y segura.

* 1. **¿En qué documento quedan reflejados ejemplos de uso del programa?**

Los ejemplos de uso quedan reflejados en la guía de uso.

* 1. **¿Cuál es el objetivo de la guía técnica?**

EL objetivo de la guía técnica es facilitar el correcto desarrollo y realización de las correcciones en los programas, permitiendo un mantenimiento futuro.

* 1. **¿En qué documento quedan reflejados la solución de posibles problemas que puedan surgir al utilizar la aplicación?**

Las soluciones a los posibles problemas a futuro se reflejan en la guía de uso.

* 1. **¿En qué documento queda reflejada información sobre la seguridad del sistema?**

Esta información se encuentra en la guía de instalación.

* 1. **¿A quién va dirigida la guía de instalación?**

La guía de instalación va dirigida al personal informático responsable de la instalación junto con los clientes.

* 1. **¿En qué documento queda reflejado la codificación de los programas?**

La codificación se muestra en la guía técnica.

* 1. **¿A quién va dirigida la guía técnica?**

Esta guía está dirigida al personal técnico en informática.

* 1. **¿En qué documento queda reflejada la descripción de la funcionalidad de la aplicación?**

La descripción de la funcionalidad se encuentra dentro de la guía de uso.

* 1. **¿En qué documento queda reflejado el diseño de la aplicación?**

El diseño de la aplicación queda reflejado en la guía técnica.

* 1. **¿Qué fases de un proyecto es necesario documentar?**

Es preciso documentar todas las etapas de desarrollo del software para dar toda la información a los usuarios del software y acometer futuras revisiones.

* 1. **¿En qué documento quedan reflejados requisitos software de la aplicación?**

La guía de uso establece los requerimientos software de la aplicación.

* 1. **¿A quién va dirigida la guía de uso?**

La guía de uso va dirigida para los clientes, los usuarios que van a usar la aplicación.

* 1. **¿Cuál es el objetivo de la guía de uso?**

Su objetivo es el de dar a los clientes toda la información necesaria acerca del funcionamiento de la aplicación.

* 1. **¿En qué documento queda reflejada la forma de comenzar a ejecutar la aplicación?**

La guía de uso explica la forma de comenzar la ejecución de la aplicación.

# 6. Explotación

## Mapa conceptual

## Preguntas y dudas

1. **¿En qué consiste la fase de explotación?**

Esta fase consiste en el conocimiento de la aplicación y su utilización inicial por parte de los usuarios finales.

1. **¿En qué consisten las betas test?**

Estos test son las últimas pruebas que se realizan en los equipos de los clientes.

1. **Una vez se alcanza la fase de explotación del software, ¿es necesaria alguna labor por parte del equipo de desarrollo?**

Sí, es precioso la fase de configuración, en la cual asignamos los parámetros de funcionamiento normal de la empresa y se prueba que la aplicación es operativa.

1. **¿Quién se encarga de realizar las configuraciones en la fase de configuración?**

Esta fase se le puede otorgar a los propios clientes, siempre y cuando se les haya entregado la guía de instalación previamente.

1. **¿Quién debería realizar la configuración del software si es una aplicación hecha a medida del cliente?**

En este caso, la configuración debería realizarla el personal que lo haya fabricado.

# 7. Mantenimiento

## Mapa conceptual

## Preguntas y dudas

1. **¿Cómo se denomina a los cambios que se realizan en el software para adaptarse a las nuevas tendencias del mercado, a nuevos componentes hardware, etc.?**

A estos cambios acorde con las tendencias del mercado son los cambios adaptivos.

1. **¿Cómo se denomina a los cambios que se realizan en el software para satisfacer nuevas necesidades del cliente?**

Estos cambios que satisfacen las necesidades de los clientes son los cambios evolutivos.

1. **La duración de la etapa de mantenimiento es corta. ¿Verdadero o falso?**

Falso, la etapa de mantenimiento es la más larga de todo el ciclo, ya que el software por su naturaleza, precisa de actualizaciones con el tiempo.

1. **¿Cómo se denomina a los cambios que se realizan en el software para mejorar la funcionalidad del software?**

Los cambios que se realizan en el software para la mejora de su funcionamiento son los cambios perfectivos.

1. **¿Cómo se denomina a los cambios que se realizan en el software para eliminar errores?**

Los cambios correctivos son los encargados de solventar los errores en el futuro.