



UNIVERSIDAD DE GRANADA

Administración de Sistemas y Seguridad

Máster en Ingeniería Informática

Curso 2019-2020

Lidia Sánchez Mérida

Índice

Implantación, administración y gestión de un Sistema de Información.	3
1.1. Introducción, planificación y viabilidad.	3
1.2. Análisis del Sistema de Información.	5
1.3. Diseño del Sistema de Información.	6
1.4. Construcción	7
1.5. Implantación	8
1.6. Mantenimiento	10
1.7. Configuración y seguridad	10
Rendimiento de un Sistema de Información.	11
2.1. Tipos de cargas	11
2.2. Evaluación del rendimiento	12

1. Implantación, administración y gestión de un Sistema de Información.

1.1. Introducción, planificación y viabilidad.

Un **sistema informático** solo puede considerarse como sistema si sus componentes se encuentran integrados. Esta característica les permite trabajar de forma coordinada y colaborativa para alcanzar el objetivo global. Los elementos que lo forma son de tipo *hardware*, *software*, redes, personal y de documentación del sistema.

Una de las principales acciones que se debe llevar a cabo en un sistema es la **administración**, conocida como la capacidad de organizar un conjunto de recursos para que funcione correctamente. Existen diversos tipos de administración de sistemas de información.

- **Administración de sistemas.** Su objetivo principal consiste en mantener disponibles los sistemas informáticos controlando los distintos aspectos que pueden afectar a su funcionamiento, como el rendimiento, los recursos y la evolución del sistema. Cada organización establece sus propias actividades de administración pero algunas de ellas son comunes, como la renovación de los componentes de un sistema. Cuando existen varios sistemas informáticos se suele asignar un administrador para cada uno, existiendo así administradores de servidores, bases de datos, entre otros.
- **Administración de sistemas operativos.** En este tipo se gestionan los recursos y componentes de un sistema operativo, como pueden ser los usuarios, su información, etc.
- **Administración de red.** En este ámbito se lleva a cabo el diseño y la manutención de las redes de comunicaciones así como la configuración y adaptación de los dispositivos involucrados.
- **Administración de base de datos.** Para este tipo se realizan tareas especializadas en sistemas de almacenamiento como el diseño de los modelos de datos, medidas de seguridad, rendimiento, copias de seguridad, gestión y evolución de la capacidad, entre otras.
- **Administración de documentos.** El objetivo reside en controlar las distintas fases por las cuales se desarrolla y mantiene la documentación de un sistema.

Para el desarrollo de un sistema o cualquier tipo de software, se debe aplicar alguna de las metodologías de desarrollo existentes. Entre ellas consideraremos la española denominada **MÉTRICA v.3** en la que se recopilan las siguientes fases.

1. **Planificación de Sistemas de Información.** En esta fase se definen los objetivos generales de la organización para diseñar un **modelo de**

información que puede estar repartido en varios subsistemas interconectados para funcionar como uno solo. El diseño de los mismos lo proporciona el **modelo de sistemas de información**. A continuación se especifica la **arquitectura tecnológica** en la que se analizan todos los requisitos para elegir los recursos más adecuados. En este momento se puede contratar una consultora puesto que son especialistas en realizar estudios temporales y monetarios de las diferentes alternativas tecnológicas. Finalmente se establece el **plan de proyecto** que se va a seguir así como el **plan de mantenimiento** que recopila las tareas y su temporización para cumplir la planificación, aunque puede estar sujeto a cambios.

La tarea del administrador es verificar que están disponibles los recursos de todo tipo, ya sean software, servicios como el suministro eléctrico, entre otros, para poder establecerlos contratos con los proveedores de los mismos..

2. **Desarrollo del sistema.** En esta fase se efectúan las siguientes etapas del ciclo de vida del software. Es recomendable que el administrador se encuentre presente para preparar los recursos necesarios antes de comenzar el desarrollo.

- **Estudio de Viabilidad del Sistema.** En esta fase se especifican y analizan los requisitos a través de las siguientes actividades.
 - **Alcance del sistema.** Se describe de forma general el sistema así como las características y relaciones de sus entidades, como los usuarios. A continuación se establece el dominio del sistema que proporcionará el rango de valores para cada atributo. Finalmente se agrupan aquellos requisitos que se encuentran relacionados.
 - **Estudio de mercado.** Se analiza el estado del arte para comprobar los sistemas existentes relacionados con el que se va a desarrollar. El objetivo es conocer la información asociada que esté disponible así como las opiniones de los usuarios.
 - **Catálogo de requisitos.** En esta fase se tratarán las políticas técnicas y de seguridad para verificar la viabilidad de la planificación establecida, la metodología de desarrollo, arquitectura del sistema y las medidas de seguridad tanto del sistema como para la protección de datos. Asimismo se establecen las medidas para gestionar el control de versiones y evaluar la calidad del sistema. Por último se clasifican los requisitos y se les asigna su respectiva prioridad.
 - **Análisis de las propuestas.** Para cada una se presentan los subsistemas a desarrollar, esquema de comunicación y los costes de la implantación así como el impacto en la organización. Si van a ser desarrollados además se incluyen los modelos de datos y de procesos así como los recursos tecnológicos. Si por el contrario ya existen, se deberá analizar

- sus características, su adecuación, adaptación y los costes tanto de las licencias de uso como de los riesgos que implica.
- **Selección de la alternativa.** Una vez disponemos de información suficiente se comparan todas las propuestas y se escoge la más adecuada en función de la planificación establecida, los costes, los riesgos que implica y el impacto en la organización.

1.2. Análisis del Sistema de Información.

En esta etapa se establecen las funcionalidades que incluirá el sistema para satisfacer los requisitos así como el estudio de los recursos necesarios para su desarrollo, validación y funcionamiento, considerando el entorno y el contexto en el que se implantará. Para ello se llevan a cabo las siguientes actividades.

1. **Descripción del sistema.** En esta fase se establece el alcance del sistema a través de los requisitos, sus funciones y el entorno tecnológico para estudiar los recursos y servicios que necesita. A continuación se analizan los estándares a seguir para la implementación que facilitará su adaptación y mantenimiento posterior. Finalmente se especifican los usuarios involucrados detallando sus roles y sus respectivas responsabilidades.
2. **Requisitos.** Se analizan y validan todos los requisitos recopilados relacionados tanto con el software y hardware, como con el rendimiento, seguridad, recursos, disponibilidad e **implantación**. Existen dos métodos para incluir un sistema en una organización: la implantación es de carácter obligatorio mientras que en la implementación se consideran las opiniones de los usuarios involucrados y sus necesidades.

A continuación se definen los **casos de uso** para describir las situaciones que se pueden originar en el sistema, detallando sus precondiciones, poscondiciones, participantes y las acciones tanto habituales como excepcionales. Por último se analizan los requisitos y casos de uso así como su forma de relacionarse para verificar que no surgen conflictos o ambigüedades, además de asignarles una determinada prioridad.

3. **Identificación de subsistemas.** Se establecen los subsistemas en los que se divide el sistema general para conseguir que trabajen de forma cooperativa y colaborativa.
4. **Modelado de la estructura de datos.** En esta fase se diseña la estructura de la información, los tipos de datos y su rango de valores. Asimismo se definen las relaciones y dependencias entre ellos. Como tarea de administración se planifica la obtención de datos para el testeo y funcionamiento del sistema. En este análisis se incluyen las técnicas para generarlos, en caso de que el sistema sea nuevo, o realizar una migración si ya existen en otro sistema. También se especifican los recursos correspondientes para su preprocesamiento y almacenamiento.

5. **Modelado del flujo de procesos.** En esta etapa se desglosan los subsistemas en procesos para conocer los recursos que necesitan. Los administradores deberán detallar los usuarios que pueden interactuar con ellos así como otros aspectos intrínsecos tales como el rendimiento, localización, entre otros.
6. **Interfaces de usuario.** Se establece qué roles de usuarios pueden acceder a las distintas operaciones del sistema de modo que solo accedan a los recursos que necesiten.
7. **Análisis de los modelos de datos y procesos.** Se analiza la viabilidad de los modelos y en caso de aprobarlos se comienza la recopilación de requisitos software para definir los recursos y funciones del sistema.
8. **Testeo del sistema.** En esta última fase se definen las pruebas de validación del sistema así como el entorno de testeo dentro de un marco de trabajo general para comprobar que cumple los requisitos del cliente. Para ello se especifica el alcance de las pruebas definiendo varios niveles para verificar el cumplimiento de todos los tipos de requisitos (hardware, software, rendimiento, etc.). Finalmente, el cliente define los criterios de aceptación para especificar las pruebas necesarias con el fin de aprobar o no el sistema.

1.3. Diseño del Sistema de Información.

El objetivo de esta tercera etapa consiste en definir y **diseñar la arquitectura del sistema**. Para ello recopilamos los requisitos de la misma, puesto que afectarán al diseño del sistema, y pasarán a formar parte del catálogo de requisitos general. A continuación se detallan tanto las situaciones normales como las excepcionales que se puedan originar, los estándares a seguir y el **entorno tecnológico**, especificando los componentes de la infraestructura tecnológica así como sus relaciones. Estos originarán una serie de restricciones, como por ejemplo de almacenamiento, que deberán considerar los administradores a la hora de proporcionar los recursos y servicios necesarios para el funcionamiento del sistema en el entorno.

De igual modo los diseñadores y administradores deberán llegar a un consenso para especificar medidas de **seguridad** únicas e independientes del contexto, como la protección de datos, así como los requisitos de operaciones, como sus dependencias o periodicidad de ejecución.

Una vez se termina el diseño se somete a una validación y aprobación para comprobar que no existen conflictos y se ajusta los requisitos del cliente. A continuación se especifican aspectos de programación tales como las clases de los módulos y sus relaciones, las estructuras de datos y el método para obtenerlos en función de su cometido. Por ejemplo, los **datos de prueba** son aquellos que se utilizan para la validación del sistema y pueden provocar situaciones anómalas para verificar el comportamiento del sistema. Mientras que la metodología para obtenerlos reside en si existen previamente, en cuyo caso se puede preparar un

entorno para la migración de datos con los procedimientos adecuados de obtención, tratamiento, almacenamiento y validación del proceso.

En relación con los datos, en esta etapa también se diseña el **sistema de almacenamiento** de los datos en el que se establece el modelo físico, los puntos de acceso y otra serie cualidades como el rendimiento y la seguridad. En esta tarea los administradores se encargan de ubicar los nodos que contienen los gestores de la base de datos de modo que cada uno almacene un subconjunto de datos.

Cuando ya se han realizado los últimos diseños se realiza una segunda verificación para comprobar que cumplen con los requisitos y no existen conflictos entre ellos. Una vez aprobados se concretan los requisitos del **entorno de construcción**, especificando los todos los tipos recursos y herramientas necesarios, teniendo en cuenta las restricciones del mismo. Del mismo modo se procede con el **entorno de testeo** en el cual se ejecutan el **plan de pruebas** establecido para validar el sistema. Lo más recomendable es establecer una jerarquía de tests que comprueben desde el elemento más sencillo hasta el sistema al completo.

Para finalizar se especifican las medidas de **implantación** relacionadas, comúnmente, con la documentación del sistema, la formación de los usuarios y el análisis de la infraestructura en la que se instalará el sistema. Por último se somete a la aprobación final del sistema a partir de toda esta información.

1.4. Construcción

En esta cuarta fase se desarrolla y testea el sistema a partir de la información anterior. Posteriormente, si se adquiere su propiedad se tendrá acceso al código fuente, procedimientos desarrollados, evaluaciones, mientras que si solo se adquiere una licencia de uso se proporciona el ejecutable y la documentación del sistema. Sin embargo, para la construcción del sistema de información se deben realizar las siguientes actividades:

1. **Entorno de construcción.** En esta primera tarea se comprueban que los recursos y herramientas necesarios para la construcción se encuentren disponibles. Asimismo, se crean y configuran los sistemas de almacenamiento requeridos para el funcionamiento del sistema.
2. **Implementación y documentación.** En esta segunda actividad se codifican los módulos del sistema a la vez que se van documentando para los distintos tipos de usuarios involucrados. En el ámbito de la administración se desarrollan los procedimientos de administración, el control de acceso y las medidas de seguridad establecidas en el diseño.
3. **Pruebas unitarias.** En esta fase se comprueba el correcto funcionamiento de los componentes del sistema de forma individual. En este caso los administradores deben preparar el entorno de validación para realizar las pruebas unitarias.

4. **Pruebas de integración.** En esta actividad se testean conjuntos de componentes para comprobar su comportamiento al ser integrados. De nuevo los administradores deben preparar el entorno para realizar las pruebas de integración.
5. **Pruebas del sistema.** En este caso las pruebas validan el comportamiento del sistema una vez se han integrado todos los componentes. Para ello los administradores deben preparar el entorno para realizarlas.
6. **Formación de los usuarios finales.** En esta actividad se prepara el contenido de la formación que recibirán los usuarios finales para utilizar el sistema. Para ello se deberán considerar los recursos necesarios y sus costes.
7. **Migración y carga inicial de datos.** Los administradores deben preparar los recursos necesarios para efectuar la migración de datos, si procede, así como la carga de datos inicial. A continuación los desarrolladores codifican estos procedimientos así como sus tests asociados para verificar su funcionamiento.
8. **Aprobación del sistema.** En última instancia se somete el sistema a la aprobación por parte de los jefes de proyecto.

1.5. Implantación

En esta fase se prepara el entorno en el que se integrará el sistema utilizando el software desarrollado y su documentación. Se tratarán diversos aspectos tales como la contratación de los servicios necesarios, la comprobación del cumplimiento de los requisitos, entre otros. Los pasos específicos se detallan a continuación.

1. **Plan de implantación.** El objetivo consiste en planificar los recursos tecnológicos, humanos y de información necesarios para implantar el sistema. Para ello se analizan los requisitos de infraestructura y componentes para cada subsistema. A continuación se definen las pruebas de implantación para verificar el comportamiento del sistema una vez instalado, así como las pruebas de aceptación cuyo objetivo es demostrar que satisface las necesidades del cliente.
2. **Formación del equipo de implantación.** Se proveen los recursos necesarios para la formación del equipo de implantación. Cada perfil recibirá una formación específica en función de sus responsabilidades.
3. **Integración del sistema y carga de datos inicial.** La primera tarea consiste en proporcionar los recursos necesarios, como los datos iniciales, las dependencias y bases de datos, para instalar y ubicar el sistema. A continuación se comprueba que el proceso de carga de datos efectuado ha sido satisfactorio. Por último se configuran los procedimientos asociados a las copias de seguridad, restauración, sincronización, entre otros.
4. **Pruebas de implantación.** El objetivo consiste en comprobar que el sistema implantado se comporta tal y como se recoge en los requisitos. Para ello se prepara el entorno de implantación, se realizan los tests, pudiéndose introducir más pruebas bajo diversas condiciones para comprobar aspectos

tales como la recuperación frente a fallos o el rendimiento. Por último se analizan los resultados obtenidos y se comparan con los esperados para tomar las decisiones correspondientes. Una de ellas consiste en decidir si efectuar de nuevo las pruebas de implantación de forma parcial o total para verificar las nuevas modificaciones introducidas, en su caso.

5. **Pruebas de aceptación.** Estas pruebas son desarrolladas por los usuarios finales para asegurar que el comportamiento del sistema es tal y como se esperaba. Se analizan los resultados obtenidos con los esperados para poder, de nuevo, resolver las incidencias que se hayan presentado y estimar su coste asociado. Finalmente los usuarios aprobarán o no el sistema.
6. **Mantenimiento del sistema.** El responsable de mantenimiento debe encontrarse dentro del equipo de implantación para que conozca los productos sujetos a mantenimiento. De este modo podrá valorar si dispone de las herramientas e información necesarias para el mantenimiento. Por último se establece un plan de mantenimiento para cada producto, incluyendo los tipos de recursos requeridos.
7. **Acuerdo de nivel de servicio.** En esta fase se especifican los servicios comunes y específicos de los sistemas que se ofertan a los usuarios. Para ello se establece un acuerdo de nivel de servicio en el que se especifican todos sus detalles como los recursos necesarios para que funcionen, sus costes, funcionamiento, limitaciones y condiciones de los servicios así como las medidas de calidad, entre otros. Se pueden ofertar servicios de atención al cliente, los cuales incluyen características del propio sistema, como el rendimiento, así como servicios de gestión de operaciones. Dependiendo de su naturaleza necesitarán unos recursos diferentes.
8. **Presentación y aprobación del sistema.** En esta etapa se presenta toda la información generada hasta el momento para presentar el sistema a la directiva y someterlo a la aprobación.
9. **Producción.** Una vez el sistema es aprobado se prepara el entorno de producción para implantar el sistema en él. Desde este momento se carga la información inicial para su funcionamiento, comienza el mantenimiento así como los servicios ofertados en los acuerdos de nivel de servicio.

1.6. Mantenimiento

En esta etapa se registran las peticiones de los usuarios para resolver las necesidades que detectan cuando utilizan el sistema. Para ello se configura un registro en común con los responsables de mantenimiento para gestionar las incidencias. Su información asociada recoge los pasos para replicarla, el fallo, los datos de entrada y la salida esperada. En caso de que se trate de una mejora, además deberá contener sus requisitos y el objetivo de la misma. A continuación, a cada uno de los registros se les asocia un identificador único, su origen y tipo, una prioridad que indica su importancia y una descripción breve sobre de qué se trata.

Cada petición se analiza para comprobar la veracidad de la información y clasificarla según es mantenimiento **correctivo**, si se trata de un error, o **evolutivo** en otro caso. Para el primer tipo se determina su criticidad de modo que si es elevada se elabore una solución inmediata aunque posteriormente se estudie una mejor. Mientras que en el segundo tipo se estudia la viabilidad de la nueva funcionalidad y el impacto en los productos involucrados, en los servicios y en el entorno tecnológico.

Asimismo, se analiza para todas las peticiones los recursos necesarios así como sus costes asociados. Si algunas se encuentran relacionadas se pueden agrupar para abordarlas en conjunto. Finalmente se selecciona junto al usuario la alternativa más adecuada.

A continuación se detallan los componentes de la infraestructura tecnológica y los subsistemas afectados para planificar el desarrollo de la modificación. Para ello se llevan a cabo todas las actividades estudiadas hasta el momento. De igual modo se especifican las denominadas **pruebas de regresión**, que verifican si los cambios han solventado el problema y no han generado otros nuevos tanto en los productos afectados como en su entorno. Cuando se obtienen los resultados se analizan para comprobar el correcto funcionamiento tras los cambios y se finaliza la petición. Finalmente se calculan los costes monetarios y temporales y se registra la información de la petición como el impacto, la solución aplicada, de modo que el equipo de mantenimiento disponga de cierto *feedback* para futuras peticiones.

1.7. Configuración y seguridad

En esta etapa el objetivo reside en controlar la configuración del sistema así como los cambios y las versiones de sus componentes. Para **establecer el plan de gestión de la configuración** es necesario, en primer lugar definir los requisitos generales y los procesos que se van a llevar a cabo para supervisar los productos. Entre estos se incluyen la asignación de tareas a un equipo o identificar la información relevante, entre otros. A continuación se debe decidir si seguir algún estándar para llevar a cabo la gestión de la configuración o desarrollar un modelo propio con el que cubrir las necesidades detalladas anteriormente. En función de esta decisión se deberá explicitar los recursos tecnológicos necesarios para preparar el entorno de la gestión de la configuración.

Cada producto deberá disponer de un identificador único, su versión y su estado inicial para comenzar su supervisión y su posterior mantenimiento. En este último caso se consigue ahorrar mucho tiempo cuando se dispone de un seguimiento y control sobre las incidencias detectadas en los productos. Para ello es necesario registrar todas las peticiones de mantenimiento especificando las versiones de los productos afectados incluyendo las distintas modificaciones hasta el producto final. Así se puede mantener un catálogo de los cambios realizados sobre los productos de modo que se tenga acceso a sus históricos y las versiones generadas. Dentro del registro de versiones también se deberán apuntar los nuevos productos

desarrollados a consecuencia de una petición, asignándoles su identificador, versión y estado inicial.

En cuanto a la **seguridad** de los sistemas, en MÉTRICA v3 solo se contempla aquella relacionada con la parte lógica, como ataques de terceros, infecciones por *malware*, entre otros. En este caso, como en el anterior, también se debe decidir entre aplicar un plan de seguridad existente o desarrollar uno propio. En él se incluyen dos tipos de actividades relacionadas con la seguridad: intrínsecas al propio sistema o al desarrollo del mismo. Las medidas de seguridad deberán de aplicarse de forma particular a cada sistema, considerando sus características y los recursos disponibles. Para ello será necesario un análisis de riesgos con el fin de concretar los riesgos que estamos dispuestos a asumir.

2. Rendimiento de un Sistema de Información.

El rendimiento de un sistema suele expresarse como el conjunto de prestaciones que este puede proporcionar en función de los programas que se ejecuten, la red de comunicaciones y sus componentes hardware. Su rendimiento debe encontrarse establecido en el **Acuerdo de Nivel de Servicio** para que tanto los usuarios finales como los responsables del sistema conozcan las prestaciones acordadas.

Para comprobarlo se suelen realizar estudios con objetivos específicos para establecer cuáles son los componentes a analizar. Para ello se utilizan **programas de carga**, que estresan el sistema con el fin de cuantificar unas prestaciones concretas en un tiempo determinado. En la siguiente sección se detallan los tipos de carga que podemos utilizar.

2.1. Tipos de cargas

Por lo general la carga que se aplica al sistema durante un período determinado se denomina **carga de prueba**. Esta debe ser la misma en los diferentes experimentos, aunque suele ser complicado de controlar. La carga de prueba, a su vez, se divide en:

- **Real.** Es la carga real que se está ejecutando en el sistema, incluyendo tanto los programas en ejecución como los usuarios activos. Se utiliza cuando se desea representar el estado real del sistema aunque es poco reproducible puesto que cada situación es única.
- **Sintética natural.** Son los denominados *benchmark*, programas que contienen una serie de operaciones reales que se llevan a cabo en el sistema. En ellos se pueden establecer algunos parámetros para la monitorización del sistema y establecer prioridades para las tareas.
- **Sintética híbrida.** Es una mezcla entre las actividades reales que se producen en el sistema y otras adicionales para incrementar el impacto total.

- **Carga artificial.** En este caso no se incluyen operaciones reales sino que se modela una determinada carga para aplicarla al sistema o para representarlo. Existen varios tipos.
 - **Ejecutable MIX de instrucciones.** Mide la frecuencia de aparición de ciertas instrucciones. Se suelen utilizar para comparar procesadores.
 - **Ejecutable MIX de sentencias.** El objetivo es el mismo que en el caso anterior solo que ahora las instrucciones se programan en un lenguaje de alto nivel y por ello son altamente dependientes del compilador.
 - **Ejecutable Kernel.** Representa una carga compuesta por programas similares a los de una carga real de los que se conoce su consumo de recursos.
 - **Ejecutable programas sintéticos.** Son programas orientados al consumo ineficiente de recursos. Su objetivo es saturar el sistema.
 - **Ejecutable secuencias conversacionales.** El objetivo es simular el envío de peticiones al sistema utilizando otros sistemas o a él mismo.
 - **Ejecutable benchmarks.** Son programas creados aplicando alguna de las técnicas anteriores cuyo objetivo es estresar al sistema.
 - **No ejecutables.** Se originan a partir de procedimientos estadísticos en los que se recogen las características asociadas a la carga real del sistema con el objetivo de generar una carga similar. Posteriormente se desarrolla un modelo matemático parametrizable con el que estimar las prestaciones obtenidas a partir de la carga artificial generada. Entre sus ventajas se encuentran la posibilidad de replicarlas, su adaptación por medio de los parámetros, entre otras.

2.2. Evaluación del rendimiento

La evaluación de las prestaciones de un sistema se realiza mediante la aplicación de una misma carga de prueba, que puede ser real si esta es estática o sintética, como ocurre en la mayoría de casos puesto que es ajustable y reproducible. El objetivo de estos estudios es generar una carga lo más parecida a la real del sistema, y para ello se pueden tomar algunas de las siguientes medidas.

- **Medidas para los componentes de carga.** Se puede medir el tiempo de CPU por tarea y entre operaciones E/S, memoria, prioridad de los procesos, etc.
- **Medidas para el conjunto.** Tiempo entre dos peticiones, su frecuencia, distribución de tareas, número de usuarios simultáneos, entre otros.
- **Para las prestaciones.** Cualidades del sistema, su estado durante el estudio, prestaciones que percibe el usuario (tiempos de respuesta) y el administrador (tiempo real de uso, accesos a un recurso, sobrecarga, frecuencias de fallos).
- **Otras medidas.** Como la fiabilidad del sistema en un período determinado, disponibilidad, seguridad, facilidad en el mantenimiento, entre otras.

Además de las medidas, también debemos decidir las herramientas que vamos a emplear para el estudio del rendimiento. Algunas de las más populares se explican a continuación.

2.2.1. Técnicas de evaluación del rendimiento

La primera de ellas se conoce como **monitorización**, en la que se utilizan programas para supervisar el sistema en tiempo real. Esta condición impide reproducir las cargas con exactitud por lo que estas herramientas recopilan la mayor cantidad de información posible acerca de su comportamiento. Dependiendo de los componentes que deseemos evaluar, existen monitores de software y hardware, monitores que se activan con eventos o en períodos de tiempo concretos, además de herramientas que presentan los resultados en tiempo real o al final del estudio. La visualización de estos suele representarse de forma gráfica.

Otra de las más utilizadas es el **benchmarking**, con la cual se pueden comparar las prestaciones obtenidas entre varios sistemas o configuraciones del mismo. Su popularidad reside en su sencillez, pero se debe tener cuidado con la carga generada de modo que esta sea lo más similar a la carga real para que los resultados obtenidos sean útiles.

Por último cabe destacar los **modelos analísticos** y los **modelos de simulación**. Los primeros se basan en teorías matemáticas y estadísticas para determinar el uso de los recursos. Para ello establecen la denominada *red de colas* con la que analizan el tiempo transcurrido entre que se manda una petición y se finaliza. Dependiendo de los análisis que se deseen realizar, se establecen una serie de parámetros y las relaciones entre ellos haciendo uso de las *leyes operacionales*, con las que se determinan los atributos a medir para evaluar el rendimiento del sistema. Mientras los **modelos de simulación** elaboran programas capaces de replicar el comportamiento real del sistema para realizar diferentes simulaciones y analizar sus posibles reacciones. Esta técnica se caracteriza por su alta complejidad tanto en el desarrollo como en la validación del modelo.

Una vez disponemos de los resultados de las evaluaciones realizadas, se analizan e identifican los problemas para poder aplicar las soluciones oportunas. Muchas de ellas consisten en mejorar la configuración del hardware, software, bases de datos y/o conexiones de red cambiando algunos de sus parámetros o aplicando una política diferente para redistribuir la carga de modo que se usen los recursos eficientemente.