
GLOSARIO

En este apartado se recogen algunos de los términos que pudieran necesitar alguna aclaración precisa acerca de su significado y que, por no formar parte del propio trabajo desarrollado en el proyecto o no estar directamente relacionados con sus objetivos, no se hayan explicado en profundidad en el resto de documentos.

API: del inglés *Application Program Interface*. Corresponde con la interfaz que un sistema operativo, una biblioteca o un servicio proporciona para poder dar soporte a las peticiones realizadas por otros programas o otros componentes. Se implementa a través de un conjunto de funciones y procedimientos que proporcionan una capa de abstracción para acceder a la funcionalidad subyacente.

Doclet: es un programa escrito en Java que utiliza el *API* específica de Java para la especificación del contenido y el formato de la salida generada por la herramienta JavaDoc de generación de documentación. El *doclet* que la herramienta utiliza por defecto genera documentación `HTML` en un formato más o menos estándar para la documentación técnica de *APIs*. Sin embargo, existen muchos otros disponibles para obtener otros tipos de salida.

DTD: del inglés *Document Type Definition*. Es uno de los lenguajes de definición de esquemas de `SGML` y `XML`. Se utiliza para definir un esquema mediante el uso de un conjunto de declaraciones que especifiquen las marcas que van a conformar con dicho esquema. Permite así declarar el conjunto de restricciones que debe cumplir un documento cuyo esquema se defina mediante este lenguaje. Los documentos `XML` se describen mediante un subconjunto del lenguaje *DTD* que impone un cierto número de restricciones a la estructura del documento.

GUI: del inglés *Graphical User Interface*. Se trata de un tipo de interfaz de usuario que permite al este interaccionar con una máquina o aplicación. Su particularidad reside en el hecho de estar basada en una representación visual del sistema, sus procesos y las actuaciones del usuario. Para ello, utiliza iconos, etiquetas y todo tipo de indicadores visuales o elementos gráficos que permiten manipulación directa por parte del usuario.

IDE: del inglés *Integrated Development Environment*. Es un aplicación software que proporciona diferentes tipos de funcionalidad orientadas a la mejora y a la facilitación

del proceso de programación y desarrollo. Normalmente incluye, como mínimo, un editor de código fuente, un compilador o intérprete, herramientas de automatización de procesos de construcción y, habitualmente, además un depurador. Dependiendo del paradigma o el lenguaje al que esté orientado, puede incluir también de forma típica funcionalidad específica del entorno, como puede ser un explorador de objetos, diagramas *UML*, etc.

Locale: conjunto de parámetros que definen el idioma del usuario, el país, y otro conjunto variable de preferencias que el usuario pueda haber querido modificar en relación con la interfaz de usuario de un sistema y, posiblemente, también con el idioma que utiliza. La *locale* utilizada puede afectar al idioma en que se muestra la interfaz, el formato de fecha y hora, la zona horaria, o los formatos utilizados para representar números (separador de miles y decimales, por ejemplo).

OSGI: del inglés *Open Services Gateway initiative framework*. Es un sistema de módulos y una plataforma de servicios para Java. Facilita la modularización de aplicaciones y la separación de responsabilidades por paquetes llamados *bundles*. Los paquetes pueden definir servicios y las implementaciones de dichos servicios pueden ser sustituidas en tiempo de ejecución de forma transparente para los clientes de dichos servicios. Es la plataforma para la modularización que utiliza Eclipse.

Parser: programa o componente, a menudo parte de un compilador o un intérprete, y que recibe una entrada en forma de una secuencia de instrucciones de código, comandos interactivos en línea, etiquetas de marcado, o algún otro tipo de contenido de interfaz definida, y la descompone en partes menores (unidades gramaticales) que puedan ser utilizadas en el procesamiento por parte de algún otro componente.

SDK: del inglés *Software Development Kit*. Se trata de un conjunto de herramientas de desarrollo que permiten crear aplicaciones para un cierto paquete software, para un *framework* determinado, para una plataforma, etc. A menudo se puede reducir simplemente a una *API* con la que acceder a un set de bibliotecas, pero puede incluir también complejas estructuras de acceso a hardware y otros sistemas.

UML: del inglés *Unified/Universal Modeling Language*. Se trata de un lenguaje estándar de especificación visual para el modelado de objetos. El lenguaje *UML* es una herramienta genérica que sirve de cierta notación gráfica para crear modelos abstractos de un sistema, que se denominan *modelos UML*. Abarca diagramas de diversos tipos, entre los que se encuentran los diagramas de clases, de paquetes, de

actividad y de casos de uso.

XML: del inglés *Extensible Markup Language*. Es una especificación de propósito general para la creación de lenguajes de marcado adecuados a dominios específicos. Permite definir elementos propios de cada entorno, de ahí que lleve el calificativo de *extensible*. Su objetivo principal es facilitar la transmisión de datos estructurados entre diferentes sistemas de información.