

Oposiciones TIC

Oposiciones de Informática y Telecomunicaciones para ser funcionario TIC

Temarios y Exámenes de oposiciones gratis.

[Página principal](#) | [Preguntas de test para oposiciones de Informática ...](#) | [Examen de Inglés](#) | [Como ayudar a Oposiciones TIC](#) | [Sueldos de Funcionarios](#)

Herramientas CASE. Temario de las Oposiciones de Informática y Telecomunicaciones.

En esta entrada trataremos el tema "Herramientas CASE. Tipos. Estructura. Prestaciones" del temario de las oposiciones de Informática y Telecomunicaciones (múltiples administraciones).

Introducción

CASE es el acrónimo de Computer Aided Software Engineering, es decir, ingeniería del software asistida por ordenador.

Este término en su día tuvo sentido, puesto que se diseñaba software de forma manual: los ordenadores no tenían las capacidades necesarias para ayudar en los procesos de ingeniería. Hoy en día, todos los procesos de desarrollo se realizan ayudándose en herramientas informáticas.

En este tema veremos una descripción e historia de dichas herramientas, una clasificación de las mismas, y expondré la estructura general de las herramientas CASE actuales. Por último, veremos algunos ejemplos de herramientas y de sus prestaciones.

Herramientas CASE

En los años 60-70, cuando comenzaba a desarrollarse software, los sistemas se definían basándose en un conjunto de información textual.

A finales de los años 70 surgieron las técnicas de desarrollo estructurado, que permitían especificar el software de forma más precisa y concreta, basándose en técnicas gráficas. Sin embargo, para proyectos grandes, se generaban inconsistencias que eran difíciles de detectar. Además el volumen de documentación generado resultaba complicado de tratar de forma manual.

Estos problemas, junto con el aumento de las capacidades de los ordenadores, propicia el surgimiento de las herramientas CASE que se desarrollaron para afrontar los problemas de la calidad del software y de la documentación inadecuada. La idea de las herramientas CASE es automatizar los métodos existentes de la ingeniería del software, para mejorar la eficiencia y la calidad del proceso.

A principios de los 80, las herramientas CASE se utilizaban para la gestión de la documentación y para la creación de diagramas. A mediados de los 80 empiezan a integrarse herramientas que chequean las inconsistencias de forma automática, y se generalizan los repositorios centralizados con información del sistema.

A finales de los 80 se comienza a generar código a partir del diseño. En los años 90 se generalizan las interfaces gráficas y se fomenta la reutilización de código como metodología de desarrollo.

Hoy en día la tendencia es a utilizar un IDE o entorno integrado de desarrollo, como pueden ser Eclipse o Visual Studio, sobre el que se instalan plugins o complementos que cubren las diferentes necesidades del proceso: integración con gestión de proyectos, con repositorios de código, documentación y diseño de diagramas, etc..

Tipos

En la clasificación de herramientas CASE pueden utilizarse muchos criterios diferentes, según el punto de vista desde el que las examinemos. Veremos aquí tres clasificaciones que consideramos son las más relevantes.

Según tipo de integración

Juegos de herramientas (Toolkits)

Los toolkits consisten en una serie de herramientas independientes entre sí, donde cada una de ellas sirve para realizar una función. Por ejemplo podría haber un compilador/depurador específico (gcc) un IDE para proporcionar una interface gráfica (Eclipse) y un sistema para gestionar las versiones (SVN). Dichos productos se integran pero son intercambiables.

Bancos de trabajo (Workbenches)

Se trata de un conjunto de herramientas integradas que comparten una base de datos de soporte y una interface única. Están orientados a una metodología de desarrollo, y ayudan al usuario a seguir la misma. La salida de una de las fases metodológicas se puede utilizar directamente de entrada para la fase siguiente. Un ejemplo sería la herramienta Power Designer de Sybase

Según metodología utilizada

Metodología estructurada

Categorías

- [examen de test](#) (26)
- [Trabajo del funcionario TIC](#) (11)
- [Contratación TIC](#) (8)
- [ejercicios de examen](#) (8)
- [estadísticas oposiciones](#) (5)
- [funcionamiento del proceso selectivo](#) (5)
- [derechos de los funcionarios](#) (4)
- [examen de desarrollo](#) (4)
- [sueldo](#) (4)
- [examen de ingles](#) (3)

Entradas antiguas

- ▶ [2023](#) (1)
- ▶ [2019](#) (1)
- ▶ [2018](#) (1)
- ▶ [2017](#) (8)
- ▼ [2016](#) (4)
 - ▼ [octubre](#) (3)
 - Qué es la tasa de reposición de efectivos.
 - Herramientas CASE. Temario de las Oposiciones de I...
 - Arquitectura cliente servidor
- ▶ [septiembre](#) (1)
- ▶ [2015](#) (3)
- ▶ [2014](#) (2)
- ▶ [2013](#) (20)
- ▶ [2012](#) (11)
- ▶ [2011](#) (27)
- ▶ [2010](#) (77)
- ▶ [2009](#) (66)
- ▶ [2008](#) (5)

La metodología estructurada fue la primera que se usó de forma extendida, y a la que las primeras herramientas CASE daban soporte. Las herramientas de este tipo permiten gestionar proyectos de desarrollo mediante la construcción y gestión de DFDs, modelos Entidad/Relación, Diagramas de módulos, etc..

TODO: Poner un ejemplo

Metodología orientada a objetos

La metodología orientada a objetos surgió con posterioridad, cuando ya las herramientas CASE estaban más maduras. Las técnicas de la MOO son muy diferentes, y están basadas sobre todo en el lenguaje UML y en el desarrollo iterativo basado en casos de uso. Hoy en día es más común desarrollar con esta metodología y con herramientas que la soportan. También existen herramientas mixtas, que permiten soportar ambas metodologías.

Según etapa del ciclo de vida en que se usan

Otra forma de clasificar las herramientas es según en qué fase del ciclo de vida resultan de utilidad, aunque hay que tener en cuenta que muchas de ellas están integradas en suites y se aplican a más de una tarea (por ejemplo, creación de E/R y generación del SQL)

Gestión de proyectos y requisitos

Permiten realizar diagramas de Gantt y PERT, obtener el camino crítico de un proyecto, y gestionar los recursos del mismo. Además permiten realizar un seguimiento del trabajo realizado para controlar si deben realizarse ajustes en el plan.

Un ejemplo de estas herramientas serían MS Project u OpenProj.

Las herramientas de gestión de requisitos comienzan a cobrar fuerza en los últimos tiempos. Permiten recoger los requisitos, asegurar la coherencia de los mismos, y realizar el seguimiento de los mismos a lo largo del proceso de desarrollo (qué requisito afecta a qué artefacto: DFD, clase, etc)

<http://www.navegapolis.net/content/view/51/49/>

Análisis y diseño

Las herramientas de análisis y diseño son muy diferentes según su metodología.

En metodología estructurada permiten realizar DFDs, descomposición modular de programas y modelos entidad/relación. Las herramientas más clásicas siguen esta metodología, aunque están adaptándose para permitir técnicas tanto estructuradas como orientadas a objetos, ya que cada vez la metodología estructurada se utiliza menos. Un ejemplo sería PowerDesigner, de Sybase, o Designer, de Oracle (herramienta por excelencia para hacer Diagramas ED y E/R.

Las herramientas de metodología orientada a objetos, debido a que el desarrollo es iterativo, suele haber una fuerte integración entre el diseño y la generación de código.

Se puede generar a partir de un modelo de clases el código que le da soporte, normalmente en varios lenguajes de programación. Las herramientas más modernas como SDE permiten además que el modelo se actualice conforme se modifica el código. Un ejemplo de este tipo de herramientas sería el ya mencionado Smart Development Environment, Rational Rose o Enterprise Architect.

Programación

Las herramientas de programación facilitan las tareas de edición, compilación y depuración de código. Suelen estar integradas en un entorno de desarrollo o IDE (de Integrated Development Environment). Se incluyen en esta categoría, aparte de las mencionadas, herramientas como los editores de código con chequeo de sintaxis, los generadores de interface de usuario (editores de ventanas), generadores de consultas a bases de datos, etc.

Ejemplos de este tipo de herramientas serían Microsoft Visual Studio o Eclipse.

Integración y pruebas

TODO: ejemplo

TestLink, JUnit, JMeter, etc.

Soporte y mantenimiento

TODO: ejemplo

Otra posible clasificación según el ciclo de vida las divide en Upper-CASE (apoyo a las fases iniciales: análisis de requisitos, gestión de proyectos), Medium-CASE (análisis y diseño) y Lower-CASE (implementación, pruebas y mantenimiento).

Estructura

ver herramientas case-tema2 en los materiales

Prestaciones

Dado que las herramientas CASE están en continua evolución y cada empresa utiliza las que considera más adecuadas, no tendría sentido centrar este apartado en el uso profesional de las mismas. En vez de eso voy a mencionar algunas herramientas CASE que resultarían de utilidad en la práctica docente, mencionando dónde podrían resultar más apropiadas.

- Open Proj
- Project.NET
- Ganttproject
- Eclipse
 - Clay
 - Papyrus
- Netbeans
- BoUML
- phpMyAdmin
- MySQL Navigator
- Microsoft Visual Studio
- SVN
- <http://www.moskitt.org/cas/moskitt0/>

Modeling Software KIT (MOSKitt)

Es una herramienta CASE LIBRE, basada en Eclipse que está siendo desarrollada por la Conselleria de Infraestructuras y Transporte (CIT) para dar soporte a la metodología gvMétrica (una adaptación de Métrica III a sus propias necesidades). gvMétrica utiliza técnicas basadas en el lenguaje de modelado UML.

MOSKitt se desarrolla en el marco del proyecto gvCASE, uno de los proyectos integrados en gvPontis, el proyecto global de la CIT para la migración de todo su entorno tecnológico a Software Libre.

GvPontis:

<http://www.gvpontis.gva.es/index.php?id=gvpontis&L=0>

Gestión de proyectos y requisitos

Análisis y diseño

Programación

Integración y pruebas

Soporte y mantenimiento

También pueden resultar de utilidad entornos web para el desarrollo de código. Por ejemplo, Google Code permite generar de forma sencilla un proyecto de desarrollo. Proporciona un servidor SVN para guardar el código, un Wiki para la documentación, y gestión de permisos para el acceso. Mediante Google Code puede generarse en unos minutos un entorno que permita a los alumnos trabajar de forma colaborativa en un proyecto tanto desde el centro como desde sus casas. El problema lo tendríamos con el ancho de banda, que suele ser limitado en los centros educativos.

Enlaces de interés.

http://www.taringa.net/posts/downloads/944801/Sybase-PowerDesigner-v12_5,-Modelamiento-Empresarial.html

<http://sourceforge.net/>

<http://code.google.com/>

Bibliografía

<http://www.visual-paradigm.com/product/sde/ec/>

<http://alts.homelinux.net/task.php?task=devel&view=alt>

<http://www.sybase.com/products/modelingmetadata/powerdesigner>

<http://zarza.usal.es/~fgarcia/docencia/isoftware/case/casetools.html>

<http://www.casestudio.com/enu/default.aspx>

<http://argouml.tigris.org/>

<http://www.fabforce.net/dbdesigner4/>

<http://www.embarcadero.com/>

<http://www.visible.com/>

<http://www.genexus.es/>

Ejemplo

- Smart Development Environment <http://www.visual-paradigm.com/product/sde/>
- ArgoUML, Herramienta de modelado UML escrito en Java (enlace externo)
- BOUML, Ligera herramienta de modelado UML y generación de código C++, Java e IDL. Disponible para Windows, Unix/Linux y Mac OS X (Sitio Oficial)
- Fujaba, No solo sirve para modelar sino que puede generar código Java automáticamente. También es capaz de hacer ingeniería inversa y crear los diagramas a partir del código Java [1].
- Dia Puede ser usado para modelar varios tipos de diagramas UML (enlace externo)
- gModeler Herramienta para modelado de UML basada en Flash (utilizable desde el navegador), que permite generar código Action Script 2.0 Compatible (enlace externo)
- MonoUML Herramienta CASE para la plataforma mono (Sitio Oficial)
- Papyrus, Herramienta gráfica basada en Eclipse para el modelado con UML2, es de código abierto y se ofrece bajo licencia EPL (Sitio Oficial)
- StarUML Herramienta de modelado para Windows desarrollada en Delphi. Bastante estable y utilizable (enlace externo)
- TCM, Toolkit for Conceptual Modeling, herramienta para crear diversos tipos de diagramas incluidos UML [<http://wwwhome.cs.utwente.nl/~tcm/> Web oficial]
- Umbrello Herramienta para modelado UML para el entorno KDE (enlace externo)
- UMLet Herramienta para modelado rápido de UML también escrita en Java (enlace externo)
- Netbeans módulo UML -> no acepta UML2.0
- Existen varios programas libres para hacer diagramas de Gantt:
- GanttProject: quizás uno de los más extendidos, está programado en Java por lo que es multiplataforma. Me parece algo lento y además la navegación por el timeline es poco intuitiva, pero viene bien para salir del paso.
- TaskJuggler: cuenta con una interfaz CLI y una GUI, pero parece algo complejo si sólo se quieren hacer cosas básicas. Además para cambiar una tarea el editor gráfico muestra el fichero de texto para editarlo directamente.
- KPlato: forma parte de la suite KOffice de KDE. Parece algo verde, además no he podido ni cambiar la duración de una tarea de manera fácil.
- OpenWorkbench: la alternativa a Microsoft Project, está programado en Java pero de momento sólo funciona en entornos windows así que no lo he podido probar, pero tiene buena pinta. Port para GNU/Linux en camino.
- OpenProj: también desarrollado en Java y multiplataforma, es el programa que más me ha convencido. No tiene unas barras espectaculares pero es muy sencillo y intuitivo crear tareas y modificar su duración y fechas de inicio/fin y además cuenta con varias vistas del proyecto interesantes. Es un programa open-source pero no libre, pero tiene hasta paquetes debian para su descarga.
- Planner: un gestor de proyectos para el entorno GNOME desarrollado con las librerías GTK+, simple pero efectivo.