



Proyecto JavaDiKt
Dossier V Concurso Universitario de Software Libre
por Luis A. Arce González
(30-3-2011, versión 1.1.3beta)

Índice de contenido

Sobre los kanjis	3
¿Qué es un Kanji?	3
Sobre JavaDiKt	4
¿Qué es JavaDiKt?	4
¿Qué existe parecido a JavaDiKt?	4
¿Qué aporta de nuevo JavaDiKt?	5
Los tres pilares de JavaDiKt	5
También es interesante	6
Más sobre JavaDiKt	7
En que entorno se desarrolla JavaDiKt	7
Cómo usar JavaDiKt y hacer pruebas	7
La estructura de ficheros del repositorio SVN	7
Como compilar JavaDiKt	9
Recursos	9
Impacto y estrategia de difusión	10
Estado del proyecto: beta	11
Contacto	12

Sobre los kanjis

¿Qué es un Kanji?

La palabra “Kanji” es la denominación japonesa para los caracteres ideográficos chinos. Cada carácter tiene un significado en concreto y puede leerse de varias maneras. Existen más de 13000 que son usados entre varios idiomas, entre ellos el chino, el japonés, el vietnamita o el coreano, aunque en japonés se usan a lo sumo entre 4000 y 6000.

Existen kanjis muy fáciles como éste:



Existen también otros muy difíciles:



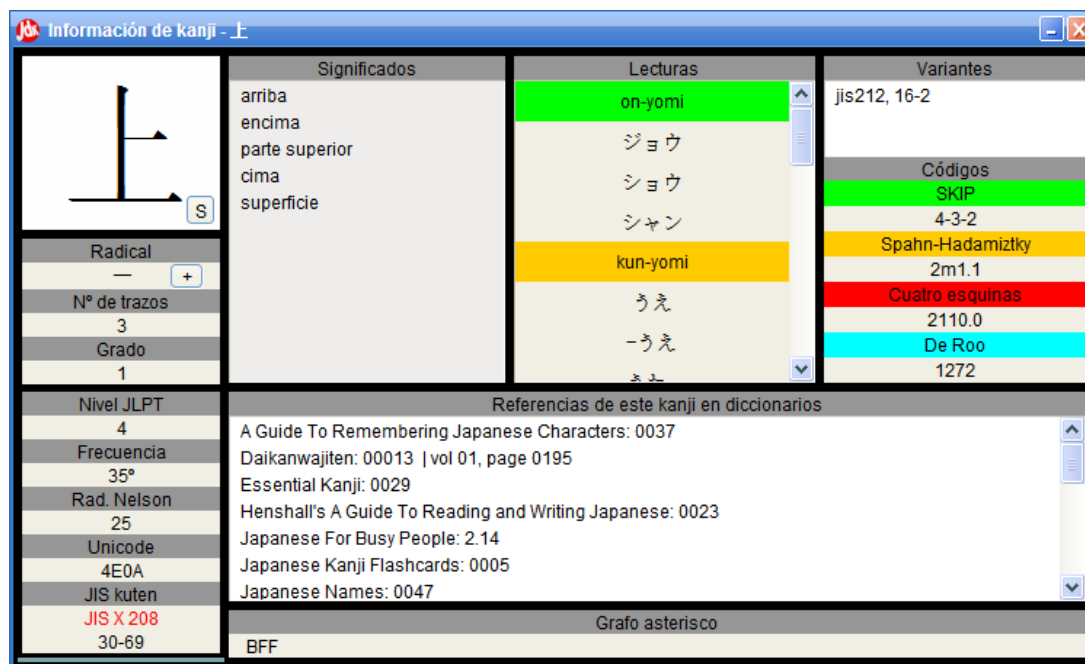
En consecuencia, los principales problemas a los que se enfrenta un estudiante de japonés cuando lee un texto es averiguar que **significan los kanjis y como se leen**. El problema se multiplica por mil aún teniendo el texto a leer un tamaño relativamente corto, tal y como se puede ver en el siguiente ejemplo de texto en japonés:

漢字（かんじ）は、古代中国に発祥を持つ文字。中国語を表記するための伝統的な文字である。また古代において中国から日本へ伝えられ、その形態・機能を利用して日本語の表記にも使われている（これについては日本における漢字を参照）。

人類史上、最も文字数が多い文字体系であり、その数は10万文字をはるかに超

え他の文字体系を圧倒している。近代以降、異体字を整理したり使用頻度の少ない漢字の利用を制限しようとする動きは何度もあったが、現在でもその数は増え続けている。

Para solucionar este problema existen los diccionarios de kanjis, que traen registrada toda esa información. Sin embargo organizar los kanjis en ese tipo de diccionarios no es trivial, dado que **lo que se busca es esencialmente un dibujo**. Es por eso que existen casi tantos métodos como diccionarios distintos para ordenar los kanjis y buscarlos (Nº de trazos, Radical, SKIP, 4 esquinas, ...) .Por todas estas razones la búsqueda de un Kanji desconocido puede resultar muy problemática.



Ejemplo de toda la información que puede estar relacionada a un Kanji en una pantalla de JavaDiKt

Puede consultar más sobre las propiedades de los kanjis en el apéndice 2 del [manual](#),

Sobre JavaDiKt

¿Qué es JavaDiKt?

JavaDiKt es un diccionario electrónico multiplataforma para PC de kanjis que aspira a servir de referencia a estudiantes de japonés de todos los niveles.

¿Qué existe parecido a JavaDiKt?

Se listan a continuación una serie de programas de escritorio libres y gratuitos que con funciones similares a JavaDiKt.

- [JWPCE](#): es un procesador de textos para Windows que permite sin instalar herramientas adicionales escribir en Japonés. Incluye también herramientas de búsqueda tanto de kanjis como de vocabulario.
- [Tagaini Jisho](#): es una herramienta multiplataforma completa de estudio de japonés, incluye búsqueda de kanjis y vocabulario y herramientas de estudio.
- [Kanji Gold](#): diccionario básico de japonés orientado más al estudio que a la referenciación.

La oferta de este tipo de programas no es especialmente abundante, especialmente en sistemas operativos Linux y Mac.

¿Qué aporta de nuevo JavaDiKt?

La mayoría de los diccionarios tanto en formato de escritorio como de página web usan la completísima base de datos colaborativa de Kanjis [KANJIDIC](#), de Jim Breen, licenciada bajo Creative Commons. JavaDiKt también usa esa base de datos, pero incorpora una serie de mejoras relativas a como maneja la inmensa cantidad de datos disponible.

Los tres pilares de JavaDiKt

1. **Sistema de búsquedas avanzado:** KANJIDIC incorpora para cada kanji gran número de índices que permiten buscar kanjis usando muchos criterios distintos. La mayoría de las aplicaciones diseñadas para la búsqueda de kanjis permiten al usuario usar estos criterios para localizar los kanjis, pero **sólo de manera individualizada**, es decir, no permiten combinar estos criterios para acotar búsquedas.

Sin embargo JavaDiKt...

permite usar un modelo similar al lenguaje natural para construir expresiones bastantes complicadas combinando multitud de criterios **que permiten al usuario usar toda la información inconcisa que ya conoce sobre ese kanji para localizar la información que no conoce.**

Lista de búsqueda				Lista de búsqueda			
Y/O	Campo	Cumple	Valor	Y/O	Campo	Cumple	Valor
-	Número de trazos	mayor que	11	-	Número de trazos	igual que	11
Y	Número de trazos	menor que	12	Y	Lectura	igual que	シ ャ ウ
Y	Lectura	igual que	シ ャ ウ	Ó	Número de trazos	igual que	12
				Y	Lectura	igual que	シ ャ ウ

Buscando lo mismo de dos formas distintas

2. **Búsqueda por dibujo:** JavaDiKt no es el primer programa que incorpora búsqueda por dibujo, varias aplicaciones de escritorio y web ofrecen esta funcionalidad.

Sin embargo JavaDiKt...

incorpora un panel de dibujo más simple e intuitivo utilizando únicamente líneas rectas. Los dibujos son más fáciles y rápidos de hacer y desaparece la necesidad de devanarse dibujando líneas con el ratón mientras se pulsa el botón izquierdo como otros sistemas como el [global IME](#). Y, por encima de todo, también pueden **combinarse búsquedas por dibujo con otros criterios.**

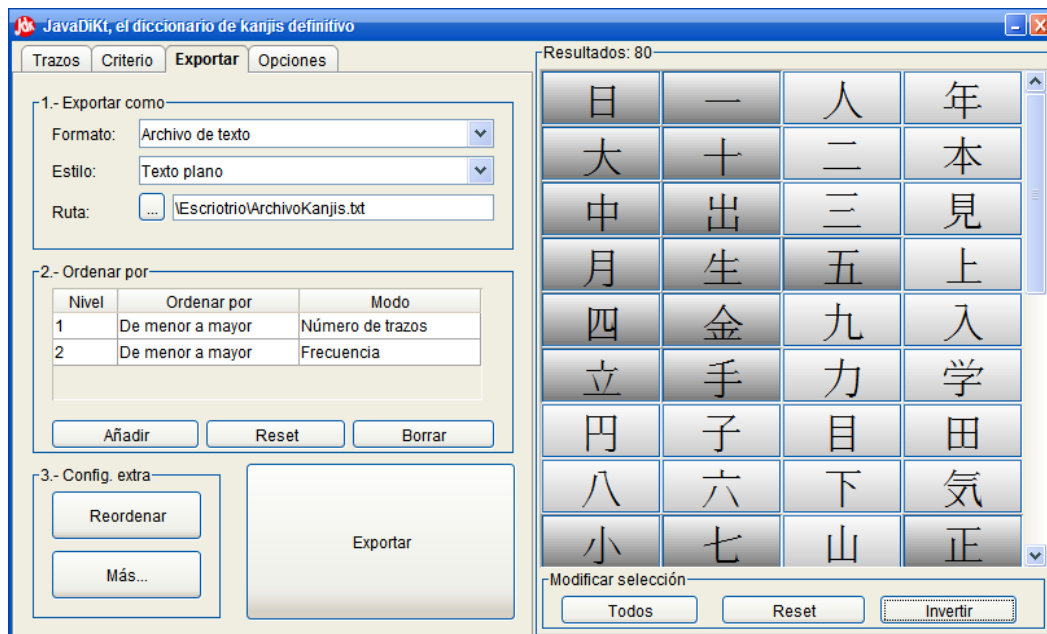


Combinando en una misma búsqueda un dibujo y su significado

3. **Exportación avanzada:** ningún programa de Japonés permite hoy en día de manera satisfactoria seleccionar aleatoriamente un grupo de kanjis y exportarlo a otros formatos que permitan portarlos e incluso usarlos en programas externos

Sin embargo JavaDiKt...

aunque de momento esta funcionalidad no está del todo implementada, JavaDiKt permitirá al usuario seleccionar entre los resultados de la búsqueda los kanjis y la información de ellos que quiere **exportar a diversidad de formatos**(PDF, html, png, odt, .anki, texto,) y estilos(tabla, tarjeta de estudio, información completa, ...) ordenándolos por los mismos criterios que sirven para buscarlos.



Exportando la información de los kanjis de manera ordenada

También es interesante

4. **Kanji API:** para hacer funcionar JavaDiKt se ha desarrollado una API no “Ad Hoc” fácilmente exportable. Es muy probable que en el futuro esta API sea

documentada y liberada, pudiendo usarse para propósito general en un gran número de sistemas gracias a estar desarrollada en Java.

5. **Multilinguaje:** desde su concepción JavaDiKt fue desarrollado con soporte multilinguaje y por defecto el usuario puede elegir entre el Español y el Inglés como idioma de la interfaz. La mayoría de los programas de escritorio de Japonés están en inglés y esto imposibilita a muchos usuarios a mejorar su idioma. Además, están en camino una página web y manuales en inglés para propiciar la internacionalización, así como una API para que los usuarios puedan crear nuevas traducciones.
6. **Funcionamiento 100% garantizado:** todas las dependencias y recursos necesarios(librerías, archivos de fuente, lanzadores) han sido añadidos directamente dentro de JavaDiKt. Por ejemplo, el usuario nunca tendrá que pelearse con las fuentes de texto y el error típico asociado a la representación de caracteres asiáticos de “falta de glyph” que ocasiona la aparición de cuadrados donde deberían aparecer caracteres. Todo preparado para funcionar solo con hacer doble click.
7. **Nueva fuente de texto:** para poder representar los 13108 kanjis de la base de datos de JavaDiKt y representar a la vez caracteres occidentales se ha construido una nueva fuente que contiene todos los caracteres tanto del plano CJK como del plano Latin-1. Esto se ha conseguido fusionando las dos fuentes libres [Sazanami Mincho](#) y [Hanazono](#) en una única fuente llamada, de momento, “Sazanami-Hanazono-Mincho”. Actualmente no existe(o no se ha encontrado) una fuente de estas características y será publicada en el futuro para su distribución una vez sea documentada.

Más sobre JavaDiKt

En que entorno se desarrolla JavaDiKt

JavaDiKt está siendo desarrollado en Java sobre una distribución de Linux Fedora, usando el entorno de desarrollo Eclipse Helios y las herramientas Java más libres disponibles, las del proyecto [IcedTea](#).

Cómo usar JavaDiKt y hacer pruebas.

Desde la primera versión de JavaDiKt todos los lanzamientos han sido “*bundled*”. En un mismo paquete zip se encuentran todas las dependencias necesarias para hacer funcionar el programa, por lo que no será necesario nada más que tener instalada una máquina virtual de Java para hacerlo correr. Se recomienda usar como mínimo la versión 1.5, pero técnicamente debería funcionar a partir de la 1.4.

Al estar desarrollado en Java, JavaDiKt puede ejecutarse casi en cualquier sistema operativo. Para mejorar el look&feel propio de cada sistema operativo se han añadido al paquete lanzadores exe, app y sh para los sistemas operativos Windows, mac y basados en Unix respectivamente.

Por otra parte, la mejor forma de toquetear JavaDiKt y obtener resultados tangibles es siguiendo [los ejemplos del manual](#), tanto de la sección de manejo como de la sección “[Significado de los campos de los kanjis](#)”.

La estructura de ficheros del repositorio SVN

En el directorio raíz del repositorio pueden distinguirse los siguientes archivos y

carpetas:

- **branches:** carpeta que contiene las distintas ramas de JavaDiKt separadas del código principal. Actualmente se está revisando a fondo el código de la API de datos de cara a mejorar la velocidad y rendimiento de las consultas en la próxima versión, la 1.1.4.
- **trunk:** directorio principal del código fuente, contiene el último código fuente disponible almacenado en una estructura de nombres de paquetes Java. Los paquetes principales son:
 - **net.ajaest.jdk:** código fuente específico de JavaDiKt
 - **net.ajaest.jdk.data:** API para el tratamiento y búsqueda de Kanjis independiente de JavaDiKt, nivel más bajo del programa. Será mejor documentada y convertida en librería en el futuro.
 - **net.ajaest.jdk.core:** núcleo de JavaDiKt y nivel intermedio entre la API y la interfaz gráfica, contiene el lanzador([Launcher.java](#)), la clase principal([Javadikt.java](#)), el sistema de gestión de mensajes ([Messages.java](#)) y los gestores de ventana ([net.ajaest.jdk.core.windHandlers](#)).
 - **net.ajaest.jdk.gui:** contiene todas las clases de definición de la interfaz de JavaDiKt y clases auxiliares específicas de SWING.
 - **net.ajaesj.jdk.tools:** contiene todas las clases de utilidad y de pruebas unitarias de JavaDiKt. Ninguna clase de este paquete será de utilidad para el usuario, enfocado a desarrolladores.
 - **net.ajaest.lib:** contiene librerías y útiles reutilizables de propósito general y que pueden ser usadas en otros programas.
- **resources:** contiene cualquier recurso o utilidad del programa que no sea código Java, como pueden ser imágenes, archivos de referencia, etc...
 - **data:** contiene principalmente los archivos y diccionarios que conforman la base la base de datos de JavaDiKt y que sirven para constuirla.
 - **doc:** contiene todos los archivos de documentación, ya sean de desarrollador o de usuario.
 - **checklists:** ficheros con los pasos que han de comprobarse antes de lanzar una nueva versión
 - **developing:** documentación de desarrollador y técnica.
 - **diffusion:** contiene todos los archivos multimedia usados para difundir JavaDiKt. Por ejemplo, este mismo archivo es guardado en esta carpeta.
 - **javadoc:** contiene el sistema de archivos HTML de la referenica Javadoc del programa.
 - **man:** contiene los binarios y archivos de origen de la documentación de usuario
 - **neodatis:** contiene documentación específica de la base de datos Neodatis.
 - **fonts:** contiene los archivos de fuente de texto usados en JavaDiKt

- **icon:** contiene los archivos planos y multicapa de los iconos usados en JavaDiKt
- **jar:** contiene los ficheros complementarios necesarios para construir el archivo .jar empaquetado del programa.
- **launcher:** contiene los lanzadores y los archivos que permiten construir los lanzadores del programa para los distintos sistemas operativos.
- **library:** contiene los binarios, códigos fuentes y licencias de las librerías usadas en JavaDiKt. Estas son:
 - **JGoodies:** librería de look&feel licenciada bajo licencia BSD.
 - **Neodatis:** librería de base datos objeto-relacional usada para el almacenamiento y búsqueda de kanjis en JavaDiKt. Licenciada bajo GNU LGPL 2.1.
 - **xom.nu:** librería de modelado de documentos mediante objetos (DOM) licenciada bajo GNU LGPL 2.1.
 - **junit 3:** herramientas de pruebas unitarias para Java, licenciado bajo Common Public License - v 1.0. Esta librería no está incluida porque no forma parte del código destinado a usuario y porque la mayoría de los IDEs lo incluyen de serie.
- **utils:** contiene programas externos de utilidad.
- **LICENSE.txt:** archivo con el contenido de la licencia GNU GPLv3
- **make.sh:** script que permite compilar el código fuente y construir una distribución de JavaDiKt
- **README.txt:** archivo con información completa destinada a desarrolladores.

Como compilar JavaDiKt

Lo primero de todo es descargar el código fuente, que puede hacerse desde la forja ya sea descargando una versión empaquetada(cada dos versiones, la última la 1.1.2beta) o accediendo al repositorio SVN(recomendado para obtener las últimas mejoras introducidas de cara al desarrollador en la versión 1.1.3beta):

<https://forja.rediris.es/svn/cusl5-javadikt>

JavaDiKt puede ser compilado desde sistemas Unix que tengan instalado el JDK de java lanzando el fichero de script “make.sh” desde la carpeta principal del código fuente de JavaDiKt(solo para versiones superiores a la 1.1.3beta). Esta acción generará dentro de la carpeta “out_javadikt” una distribución completamente lista para ser usada. Para más información, puede leer el fichero “README.txt” localizado en el fichero raíz del repositorio SVN.

Recursos

JavaDiKt pone a disposición del usuario los siguientes recursos:

- javadikt.net: es la página principal del proyecto, simple y eficiente, es la principal ventana que tiene el usuario para acceder a los recursos.
- **Manual de usuario:** el manual de usuario de JavaDiKt viene incluido *bundled* en formato PDF en el [archivo de descarga de JavaDiKt](#). En la página web puede

accederse además a su descarga directa en [PDF](#) o a una versión de navegador en [HTML](#). El manual es revisado y resubido cada versión.

Aparte de incluir información sobre el funcionamiento del programa, se ha realizado una labor de investigación para proporcionar al usuario el [significado de todas las propiedades de los kanjis](#) y sobre como aprovechar estas propiedades para realizar una mejor búsqueda.

- **Bitácora:** desde que empezó el proyecto, se han ido publicando entradas en [ajaest.net](#), mi blog personal.

Se han publicado un total de 26 entradas, lo que hace una media de más de 4 post al mes, es decir un post por semana. Algunas de ellas están comentadas. Entre los post más destacados en orden cronológico están:

[Introducción a JavaDiKt: la problemática de los Kanjis Japoneses I](#)

[Introducción a JavaDiKt: la problemática de los Kanjis Japoneses II](#)

[Introducción a JavaDiKt: la problemática de los Kanjis Japoneses III](#)

[Introducción a JavaDiKt: la problemática de los Kanjis Japoneses IV](#)

[Introducción a JavaDiKt: qué es JavaDiKt](#)

[Planificación: Estructura de JavaDiKt](#)

[Planificación: el paquete data](#)

[Planificación: el paquete data.kanji I](#)

[Planificación: el paquete data.kanji II, la base de datos Neodatis](#)

[Planificación: el paquete data.kanji III, decisiones de diseño sobre la base de datos Neodatis](#)

[Planificación: el paquete data.dict](#)

[Primera versión de JavaDiKt](#)

[Lo que el proyecto JavaDiKt supone para mí: razonamientos de un novato](#)

[JavaDiKt en marzo, nueva versión y noticias](#)

[Presentación de JavaDiKt](#)

[Entrega del premio local de Sevilla del V Concurso Universitario de Software Libre](#)

- **Forja Red-Iris:** se ha intentado aprovechar al máximo las posibilidades de la forja Red Iris.

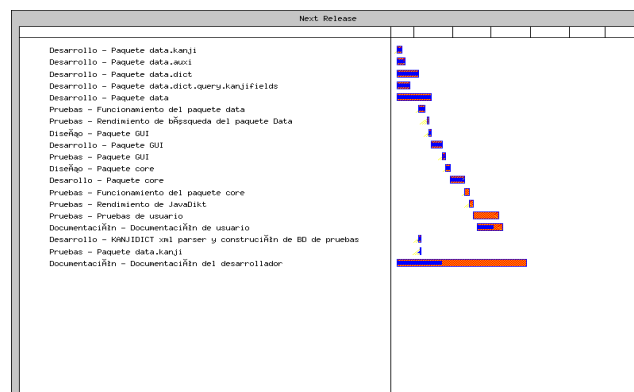


Diagrama de gant de la categoría "Next Release"

- **Registros:** a mitad de desarrollo se añadió en registros la opción de notificar bugs y solicitar nuevas funcionalidades.
- **Tareas:** casi desde el primer día se usó el sistema de tareas para definir en cierta manera el roadmap del programa. En *Next Release* se añadieron las

tareas necesarias para sacar la primera versión estable del programa, en “*Mirai*”(futuro en Japonés) se añadieron las características deseables para la primera familia de versiones y en la categoría “*Chishiki*”(sabiduría) las de las siguientes.

- [Documentación](#): en la sección de documentación se ha publicado todo aquel archivo o enlace relevante relacionado con JavaDiKt, ya sea en el ámbito del usuario([logo](#), [manual](#), [página del proyecto](#),...) o del desarrollador([diagrama de clases](#), [javadoc](#)...).
- [Repositorio SVN](#): con más de 1000 actualizaciones, el repositorio SVN se ha convertido en una herramienta fundamental a la hora de desarrollar JavaDiKt.
- [Ficheros](#): aquí se ha colgado tanto el binario como el empaquetado código fuente del programa, que es actualizado cada dos versiones.

Actualmente con 189 descargas JavaDiKt es **el segundo proyecto del VCUSL alojado en Red Iris más descargado** después de [IberOgre](#).

- [Foro de ayuda y soporte técnico](#): un pequeño foro de ayuda y soporte técnico está todavía en desarrollo en mi blog ajaest.net.

Impacto y estrategia de difusión

De momento y por distintos motivos la difusión del proyecto se ha limitado a la comunidad hispanohablante de estudiantes de japonés:

- Pese a que la interfaz del programa está traducida al inglés, aún no han sido traducidas tanto la página oficial como el manual. Este paso es fundamental para continuar la expansión a otras comunidades y está previsto en la estructura de la página.
- Las herramientas de localización por parte de la comunidad aún están en fase de planificación y una reforma en el sistema de mensajes de la aplicación ya está empezada para tal propósito.

Por estas razones la expansión de JavaDiKt va lentamente, sin prisa pero sin pausa.

El medio principal de difusión ha sido a través de foros especializados de Japonés y de descarga gratuita con buenos resultados y participación de usuarios. Los post en dichos foros son revisados todos los días en busca de respuestas de usuarios:

[Japanese for everyone](#)

[HablaJaponés.org](#)

[Athena-no-seinteo](#)

[Vagos.es](#)

[Taringa](#)

[Espacio Japón](#)

También se ha añadido a la lista de recursos en las páginas de “Japonés” y “Kanji” de la Wikipedia:

[Wikipedia - Japonés](#)

[Wikipedia - Kanji](#)

Por otra parte, un artículo sobre JavaDiKt ha sido publicado en el blog especializado de auto-aprendizaje “[Conquistando Japón](#)” y alguna noticia ha aparecido en twitter

publicada por su autora y por una persona desconocida:

[Conquistando Japón - Diccionario de kanjis JavaDiKt](#)

[twitter - @ConquistaJapon - 9 de marzo](#)

[twitter - @suguruni - 22 de marzo](#)

JavaDiKt cuenta con otros recursos. Desde mi cuenta de twitter se publican las últimas novedades del programa

[twitter - @ajaest](#)

También se han publicado en Flickr fotos sobre el concurso de Software Libre en general y del proyecto JavaDiKt en particular.

[Flickr – Presentaciones de los proyectos finalistas del premio local de Sevilla del CUSL](#)

[Flickr - Presentación de JavaDiKt en las Jornadas de Software Libre y Sociedad](#)

Por último, se realizó una presentación en japonés en el grupo de Japonés IV del instituto de idiomas de la universidad de Sevilla, la chuleta puede descargarse. A continuación, referencia de la entrada en la bitácora.

[Tecnovato - Presentación de JavaDiKt en el Instituto de Idioma para estudiantes de Japonés](#)

[Forja Red Iris - Presentación de JavaDiKt en el Instituto de Idiomas para estudiantes de Japonés\(PDF Japonés/Español\)](#)

[Tecnovato - Presentación de JavaDiKt para no estudiantes de Japones en las Jornadas de Software Libre y Sociedad](#)

[Tecnovato - Presentación de JavaDiKt para no estudiantes de Japones en las Jornadas de Software Libre y Sociedad\(PDF Español\)](#)

Estado del proyecto: beta

JavaDiKt es de momento un programa en fase beta.

En apariencia el programa es bastante estable, pero es imposible asegurar esto con 100% de seguridad, pues están pendiente de revisar desde hace tiempo una serie de pruebas unitarias que quedaron obsoletas con una actualización importante de una parte del programa. Hasta que no se implementen una serie de reformas implícitas en el roadmap de “*Mirai*”(definido principalmente en la sección “*tareas*” de la forja) no tiene sentido retomar las pruebas y el proyecto seguirá en fase beta.

Esta situación debería cambiar en las próximas actualizaciones, en las cuales se hará hincapié en la mejora del código en detrimento de la adición de nuevas funcionalidades.

Contacto

*Gracias por leer este dossier. Si tenes alguna duda o pregunta no dudes en escribir a la dirección de correo **ajaest[arroba]gmail[punto]com***