

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

IDE Bhike, herramienta para el desarrollo de software sobre el Framework Ksike

Autores:

- ✓ Antonio Membrides Espinosa - Ingeniero en Ciencias Informáticas
- ✓ Yadir Hernández Batista – Estudiante UCI
- ✓ Jorge Fuentes Rodríguez – Estudiante UCI
- ✓ Javier Rodríguez Ramírez – Estudiante UCI

Contactos: amembrides@uci.cu, yhbatista@estudiantes.uci.cu,
jfuentes@estudiantes.uci.cu, jrramirez@estudiantes.uci.cu

Temática: La innovación en el desarrollo de la industria cubana del sector de las Telecomunicaciones, la Electrónica, la Automática, la Informática y la Gestión Postal.

FORDES: Oncena Edición de la Semana Tecnológica,

Ciudad de la Habana, 25 de septiembre del 2011.

RESUMEN:

El Framework Ksike se muestra como una variante ideal para el desarrollo de aplicaciones para todo tipo de entornos y plataformas ya que gracias a las abstracciones de conceptos que presenta, brinda una solución a los problemas comunes que resultan propios de la construcción de software. Posibilita eliminar la brecha existente entre la forma de desarrollo para los distintos entornos. Aunque presenta grandes ventajas, Ksike no cuenta de una manera sencilla para efectuar su aprendizaje o uso, y sobre todas las cosas no cuenta con una herramienta que facilite el empleo de sus capacidades. Por tanto, se hace necesaria la creación de una aplicación informática que solucione las deficiencias mencionadas anteriormente.

En este punto se concibe Bhike como solución informática que pretende solventar las deficiencias de la plataforma Ksike, y se presenta en calidad de IDE conteniendo un conjunto de funcionalidades que le permitan a los desarrolladores obtener, de una manera más intuitiva, la materialización de un proyecto. Auxiliándose del mencionado Framework y logrando integrar un conjunto de principios que deban converger en el marco de trabajo Ksike-Bhike, el desarrollador tendrá en sus manos una potente herramienta.

Palabras claves: Bhike, Framework, IDE, Ksike, Software.

ABSTRACT:

Ksike framework shows up as an ideal variant for developing applications directed to any kind of environments and platforms thanks to the concept abstractions that it present, it features solutions to the most common problems in software developing. It makes possible to break the existing line between the way of building Web and desktop software. Although Ksike has great advantages, it does not count with an easy way to be learned and used, and among all other things, Ksike does not count with a tool that simplifies the use of its capabilities. That is why it becomes necessary the creation of a tool that solves the deficiencies previously mentioned.

At this point Bhike is conceived as an informatics solution that pretends to solve the deficiencies of Ksike platform, and it presents itself as an IDE containing a set of functionalities that allows the developers to achieve, in a more intuitive way, the materialization of a project. By using the mentioned framework integrated to Bhike application, the software developer should have in his hands a powerful tool.

Keywords: Bhike, Framework, IDE, Ksike, Platform, Software.

Tabla de contenidos:

INTRODUCCIÓN:	1
PANORAMA TECNOLÓGICO:	2
Entornos Integrados de Desarrollo (IDE):	2
Entornos Integrados de Desarrollo Web (WIDE):.....	5
ALTERNATIVA DE DESARROLLO ÁGIL:	6
Plataforma Ksike:	6
Concepción de Bhike:	7
Arquitectura:	9
Interfaz:.....	10
Evolución y desarrollo:	11
Módulos disponibles desde la versión 1.0 alpha:.....	12
Módulos disponibles desde la versión 1.0 betha:	13
Módulos disponibles desde la versión 1.0 estable:.....	16
Módulos disponibles desde la versión 2.0 estable:.....	17
Planes de migración:.....	19
CONCLUSIONES:.....	20
BIBLIOGRAFÍA:	20

INTRODUCCIÓN:

El empleo de herramientas ha sido la clave del éxito de la especie humana a través de los tiempos. Desde el uso de rocas, lanzas, flechas, Os, hasta las más modernas utilizadas en las industrias de producción de energía con fuentes nucleares o las versátilmente empleadas en la medicina para la realización de operaciones complejas, todas fueron ideadas para la satisfacción de necesidades del hombre ante problemas de su cotidianidad.

La industria del software genera problemas en la cotidianidad de los desarrolladores y estos no quedan absueltos al empleo de sus propias herramientas, que sobre todo agilizan el trabajo y mejoran la calidad y la eficiencia en el desarrollo de los productos informáticos. La gama de dichas herramientas es muy amplia: procesadores de textos, editores de imágenes, manejadores de multimedia, compiladores, intérpretes, herramientas de modelado, análisis y diseño de productos de software, estas últimas muy cercanas a la concepción e implementación de nuevos softwares o para el mantenimiento de los existentes. Lo cierto es que, sin lugar a dudas las herramientas informáticas son necesarias para aumentar la productividad y aprendizaje sobre el uso de las nuevas tecnologías.

Los Entornos Integrados de Desarrollo o IDE acrónimo del inglés (Intragrated Development Enviroment) son aplicaciones que proveen un marco de trabajo amigable para la concepción y desarrollo de software de alta calidad con elevada eficiencia. Están diseñados para maximizar la productividad de los programadores mediante la abstracción del uso de complicados conceptos y una interfaz que permite el empleo de funcionalidades, reduciendo la curva de aprendizaje y limitando los costes en tiempo para la construcción de sistemas informáticos. Están compuestos por: editor de texto, compilador, intérprete, depurador, sistema de control de versiones, factibilidad para la ayuda en la construcción de interfaces gráficas de usuario (GUI), y demás herramientas que contribuyan a la eficacia de las labores informáticas de desarrollo. También es posible que un mismo entorno provea integración con varios lenguajes de programación lo que permite brindar la

misma filosofía de trabajo y evitando costos en tiempo de capacitación sobre nuevas tecnologías o entornos de desarrollo.

PANORAMA TECNOLÓGICO:

De acuerdo con lo antes mencionado, se han desarrollado diversos IDE orientados a la construcción de aplicaciones para todo tipo de arquitecturas, plataformas o sistemas informáticos que responden a necesidades de los desarrolladores para los ambientes web, desktop o Mobile. Algunos de estos con el objetivo de lograr preferencia sobre el uso de una tecnología en particular, pero todos con el objetivo de garantizar rapidez y eficiencia en el desarrollo de software. A continuación, se exponen algunos de los IDE más conocidos por las filosofías que presentan o por su nivel de aceptación por parte de las distintas comunidades de desarrollo en las que son empleados, con motivo de tomar criterios y requisitos de las herramientas existentes.

Entornos Integrados de Desarrollo (IDE):

NetBeans IDE: Entorno integrado de desarrollo libre y gratuito sin restricciones de uso desarrollado completamente en el lenguaje Java usando la plataforma NetBeans. Soporta el desarrollo de todo tipo de aplicación Java (J2SE, web, EJB y aplicaciones de Mobile). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring. Su modularidad le ha permitido la potenciación de extensiones para el desarrollo de aplicaciones sobre los lenguajes de programación (Java, C/C++, Python, PHP, JavaScript) así como para las aplicaciones orientadas a servicios (SOA), incluyendo herramientas de esquemas XML, un editor WSDL, y un editor BPEL para web services. Es una aplicación multiplataforma y su última versión estable 7.0 es de 20 de abril del 2011. Es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo.

Eclipse: Es un entorno integrado de desarrollo de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar IDE como Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse. Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent o Azureus. Adicionalmente a permitirle a Eclipse extenderse usando otros lenguajes de programación como son C/C++ y Python, permite a Eclipse trabajar con lenguajes para procesamiento de texto como LaTeX, aplicaciones en red como Telnet y Sistema de gestión de base de datos. La arquitectura plugin permite escribir cualquier extensión deseada en el ambiente, como sería Gestión de la configuración. Se provee soporte para Java y CVS en el SDK de Eclipse.

Microsoft Visual Studio: Es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros. Visual Studio permite a los desarrolladores implementar todo tipo de aplicaciones, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Es un Software Privativo, aunque a partir de la versión 2005 Microsoft ofrece gratuitamente las Express Editions, que son varias ediciones básicas separadas por lenguajes de programación o plataforma enfocadas para novatos y entusiastas. Estas ediciones son iguales al entorno de desarrollo comercial, pero sin características avanzadas.

Qt Creator: Es un IDE multiplataforma para C++ que forma parte de Qt SDK. Incluye un debugger visual y un editor de Interfaces Gráficas de Usuario y de formularios. Contiene varias facilidades entre las que incluyen el sobresaltado de sintaxis y el auto completamiento. Qt Creator usa el compilador de C++ de la Colección de Compiladores de GNU en Linux y BSD. En Windows puede usar los compiladores MinGW o MSVC que forman parte de la propia instalación del

producto. Qt Creator provee soporte para construir y ejecutar aplicaciones de Qt para entornos de escritorio (Windows, Linux, FreeBSD y MacOS) y para dispositivos móviles (Symbian, Maemo y Meego). Las opciones de ensamblado permiten cambiar los objetivos del ensamblado. Cuando se construye una aplicación para un dispositivo Mobile conectado a una Computadora, Qt Creator genera un paquete de instalación, lo instala en el dispositivo y lo ejecuta. Los paquetes de instalación pueden ser publicados en Ovi Store. Para los dispositivos con Symbian, los paquetes deben ser firmados.

Code::Blocks: Es un IDE multiplataforma libre y de código abierto que soporta múltiples compiladores incluyendo GCC y MSVC. Está desarrollado en C++ usando wxWidgets como herramienta para el desarrollo de Interfaces Gráficas de Usuario. Usando una arquitectura de plugins, sus capacidades y facilidades están definidas por dichos plugins. Actualmente Code::Block está orientado hacia C y C++. También puede ser usado para la creación de programas y aplicaciones en ARM, AVR, D, DirectX, FLTK, Fortran, GLFW, GLUT, GTK+, Irrlicht, Lightfeather, MATLAB, OGRE, OpenGL, Qt, SDL, SFML, STL y SmartWin. Actualmente está siendo desarrollado para Windows, Linux y MacOS.

KDevelop: Es un IDE libre para los entornos KDE en sistemas operativos Unix, aunque también funciona en otros entornos como Gnome. KDE no incluye compilador por lo que utiliza compiladores externos como el gcc para producir código ejecutable. La versión actual, la 4.2, fue oficialmente liberada el 27 de enero de 2011. Está basado en la tecnología KDE 4 y soporta el desarrollo de aplicaciones en C, C++ y PHP. Otros lenguajes como Java, Ada, SQL, Python, Perl y Pascal, así como guiones (scripts) para el intérprete de comandos Bash no han sido portados a KDevelop4 todavía, aunque es posible que se soporten en un futuro. Kdevelop utiliza un editor de textos embebido a través del Framework KPart.

SharpDevelop: Es un entorno de desarrollo libre para los lenguajes de programación C#, Visual Basic .NET y Boo. Hay disponible un puerto para Mono/Gtk#, denominado MonoDevelop, el cual funciona en otros sistemas

operativos. Para el completado automático de código, la aplicación incorpora sus propios analizadores sintácticos. La versión 1.1 de la aplicación puede importar proyectos de Visual Studio .NET. La versión 2.0 es capaz de editarlos directamente. La versión 3.0 integra soporte para los lenguajes de programación Python y F#.

Aptana Studio: IDE gratuito basado en eclipse y desarrollado por Aptana, Inc., que puede funcionar bajo Windows, Mac y Linux. Provee soporte para lenguajes como: Php, Python, Ruby, CSS, Ajax, HTML y Adobe AIR. Tiene la posibilidad de incluir complementos para nuevos lenguajes y funcionalidades. Presenta un asistente de para HTML y JavaScript, librerías de Ajax, conexión vía ftp, sftp, ftps, y Aptana Cloud. Contiene además herramientas para el trabajo con bases de datos y es compatible con extensiones para Eclipse.

Zend Studio o Zend Development Environment: Es un completo entorno integrado de desarrollo para el lenguaje de programación PHP. Está escrito en Java, disponible para las plataformas Microsoft Windows, Mac OS X, GNU/Linux, etc. Junto con su contraparte Zend Server (antes llamado Zend Platform), son la propuesta de Zend Technologies para el desarrollo de aplicaciones Web utilizando PHP, actuando Zend Studio como la parte cliente y Zend Server como la parte servidora. Se trata en ambos casos de software comercial, lo cual contrasta con el hecho de que PHP es software libre. A partir de la versión 6, Zend Studio fue hecho tomando como base el entorno de desarrollo Eclipse. No requiere instalación previa del entorno de ejecución de Java, soportando PHP4 y PHP5.

Entornos Integrados de Desarrollo Web (WIDE):

WebDevStudio: Es un entorno de desarrollo basado en tecnologías AJAX/PHP accesible a través de un navegador web. Como cualquier otro entorno permite mantener proyectos informáticos implementados en diferentes lenguajes de programación, en este caso en C/C++ y Java, así como realizar una serie de operaciones básicas sobre ellos. WebDevStudio permite a los usuarios conectados la capacidad de tener un control absoluto sobre los proyectos que previamente hayan creado. Entre dicho control se destacan los procesos de

compilación y construcción del proyecto puesto que la herramienta permite abstraerse totalmente tanto de dichos procesos como del lenguaje de programación.

IDE Caxtor: Entorno de desarrollo conceptualizado en la Universidad de las Ciencias Informáticas dirigido al desarrollo de aplicaciones web. Se compone de dos productos, una plataforma y un IDE, en la plataforma están definidos la mayoría de los componentes, acciones y patrones que usualmente se utilizan en la capa de presentación de una aplicación Web. El IDE Castor está construido sobre la propia plataforma, y cuenta con un diseñador de interfaces gráficas que permite la creación de GUI utilizando las ventajas de la librería ExtJS, pero permitiendo una abstracción al código necesario para generarlas. Provee además un editor de código JS con completamiento de código, chequeo de sintaxis, un editor de eventos y propiedades. Actualmente el proyecto está desintegrado y solo quedan sus ideas que se reutilizaran a la medida que sea posible.

ALTERNATIVA DE DESARROLLO ÁGIL:

Plataforma Ksike:

Como parte de la filosofía de agilizar el proceso de desarrollo de software se han creado además numerosos frameworks que permiten al programador concentrarse en actividades propias del negocio de su aplicación. Este es el caso de Ksike, quien además se propone eliminar la brecha entre el desarrollo de aplicaciones web y de escritorio.

Entre las principales ventajas que provee se encuentra un API sobre el lenguaje javascript que plantea una simulación de los recursos de la Programación Orientada a Objetos permitiendo aprovechar las ventajas que este provee. Además, brinda una abstracción a la comunicación cliente servidor, utilizando tecnología Ajax y HtmlRPC, muy cómoda para el programador, aunque no es obligatorio el uso de toda la API que implementa.

La escalabilidad es otra de las metas que se plantea como lema en Ksike y por tanto brinda mecanismos que reducen la complejidad arquitectónica en función de

las necesidades reales del proyecto. En este sentido permite no utilizar el API javascript comentado anteriormente, además plantea el concepto de Package, los que son bibliotecas que se pueden incluir o no en el proyecto a desarrollar, pero además con características especiales como la posibilidad de redefinirlas físicamente. Introduce también el concepto de Filter cuyo objetivo es el de restringir las bibliotecas que serán cargadas en función de los módulos que se ejecuten.

Las potencialidades del uso del Framework Ksike son verdaderamente amplias, debido en lo fundamental a su arquitectura escalable y a las facilidades que provee. Se podrán desarrollar con él aplicaciones de variada complejidad, aprovechando las ventajas de librerías JavaScript para la generación de las interfaces de usuario. Se prevé en su desarrollo el soporte a la construcción de productos orientados a entornos de escritorio, con lo que se tendrá un elemento crucial que permite el desarrollo de aplicaciones para cualquier entorno como se muestra en la figura a continuación.

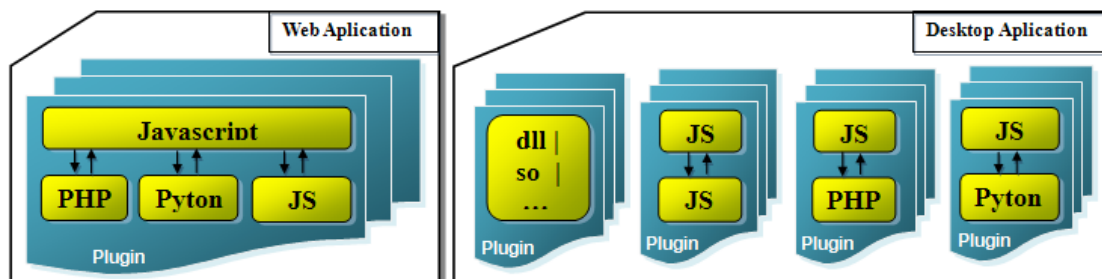


Figura 1: Esquema conceptual de las aplicaciones basadas en Ksike a partir de su versión 2

Concepción de Bhike:

Después de analizar las bondades que brinda el Framework Ksike y denotar el elevado impacto que presentan las diversas filosofías de programación, arquitectura y diseño de aplicaciones que implementa, este se muestra como una opción favorable para el desarrollo de productos informáticos tanto en entornos Web, como en Desktop y Mobile, debido a la gran flexibilidad y adaptabilidad que presenta Ksike, lo cual implica una mejora en rendimiento para negocios o desarrolladores que dediquen sus esfuerzos a realizar software para cubrir

grandes extensiones de mercado, o que cubran determinados ambientes, permitiendo reutilizar los recursos y componentes implementados, solo ajustando pequeñas detalles que marcan particularidad de acuerdo a los entornos para los que genere el producto final.

El inconveniente para el uso del Framework Ksike es que no existe actualmente una herramienta que facilite, tanto el aprendizaje de las características de este, como el uso factible de sus mencionados recursos y permita sobre todas las cosas la difusión de la tecnología para su empleo a gran escala.

Debido a la problemática que presenta el empleo de la plataforma en la cotidianidad de sus consumidores, surge la necesidad de construir un entorno de herramientas que faciliten el aprendizaje y uso del citado Framework y que traduzca el total de las funcionalidades implementadas por dicha plataforma a interfaces de usuario que reduzcan a la máxima escala posible su comprensión. Es por esta causa que se concibe al conjunto Ksike-Bhike como una alternativa para concepción óptima de proyectos informáticos y a la herramienta Bhike en sí como un puntero clave en la generalización para el uso de la plataforma Ksike.

Este proyecto se inició a finales del mes de marzo de 2011 y constituye un ejemplo de la consistencia de la plataforma que emplea como estructura funcional para su sistema y a la cual pretende solventar sus deficiencias. La idea prima fue dirigida al aprovechamiento de las tecnologías web y sus ventajas; accesibilidad que permite el uso de recursos desde cualquier lugar del mundo haciendo uso de un navegador y una conexión a Internet, sin necesidad de descarga de pesados instaladores ni acumulación de información en el disco duro del cliente. Otra de las ventajas que posee es la capacidad de ser multiplataforma, ya que solo requiere de una característica comparten todos los sistemas operativos destinados al usuario, los navegadores web. Basándose en esos preceptos de brindar un servicio aplicativo al desarrollo de software de calidad orientado a los desarrolladores como usuarios finales del sistema sin importar donde esté, ni que sistema operativo use, ni las prestaciones de las que disponga en su PC pueda utilizar el IDE Bhike y así hacer magia sobre las reglas de Ksike.

Arquitectura:

El IDE presenta la arquitectura cliente-servidor orientada a componentes como un esquema conceptual y organizativo para la concepción de proyectos, que generalmente suelen resultar soluciones informáticas para la automatización de ciertas tareas complicadas. Se basa en la creación de módulos que permitan mejorar tanto el rendimiento de las aplicaciones al margen de su desarrollo como la realización de labores de mantenimiento en el sistema, evitando en cierta medida la total caída funcional del mismo.

De esta forma implementa un módulo principal que constituye la aplicación en sí y que se encuentra ubicado en el directorio “**app**” eliminando la dependencia sobre otros plugins para su publicación. Sin embargo, la separación por módulos que posee la aplicación, potencia la reutilización de mismos así como la escalabilidad del producto final. Este módulo principal además de dar una vista panorámica sobre el funcionamiento del sistema, es el encargado de gestionar el flujo de información, así como la integración de los demás plugins al entorno Bhike, resultando el enclave fundamental para la definición tanto de configuraciones del funcionamiento de la aplicación como de los enlaces con el núcleo de la plataforma Ksike y lógica del negocio que se está implementando. Más detalles en la figura 2.

La versión alpha de Bhike consiste en una aplicación para los entornos Web, que debe contar con algunas extensiones que permitan utilizar algunos recursos de la plataforma Ksike, dichos componentes se especifican más adelante en la sección Evolución y desarrollo.

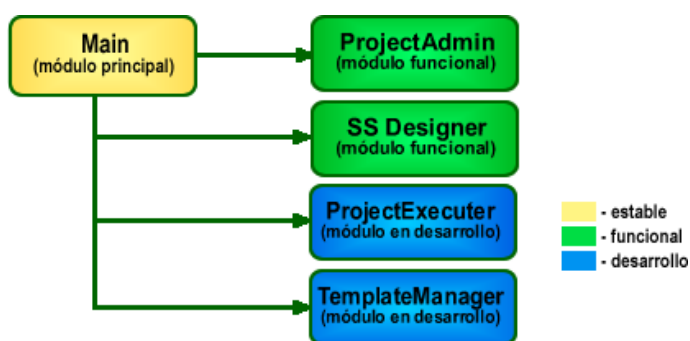


Figura 2: Componentes de la versión alpha de Bhike

Interfaz:

El diseño de la interfaz de la versión actual 1.0 alpha está implementada en JavaScript mediante el uso de las API propuestas por la versión 3.0 de ExtJS, conjunto de librerías libre para la creación de aplicaciones web estilo escritorio.

La propuesta de interfaz está basada en algunas de las GUI de los IDEs mencionados anteriormente. De esta forma se integra la sencillez y presentación intuitiva de Visual Studio como la robustez y flexibilidad del NetBeans. Por otra parte, se basa en ideas presentes en el Qt Creator, específicamente las empleadas para manipular el recurso de Signals and Slots provistos como uno de los avances y funcionalidades claves de la plataforma Qt. Refiérase a las figuras 3 y 4 para más detalles.

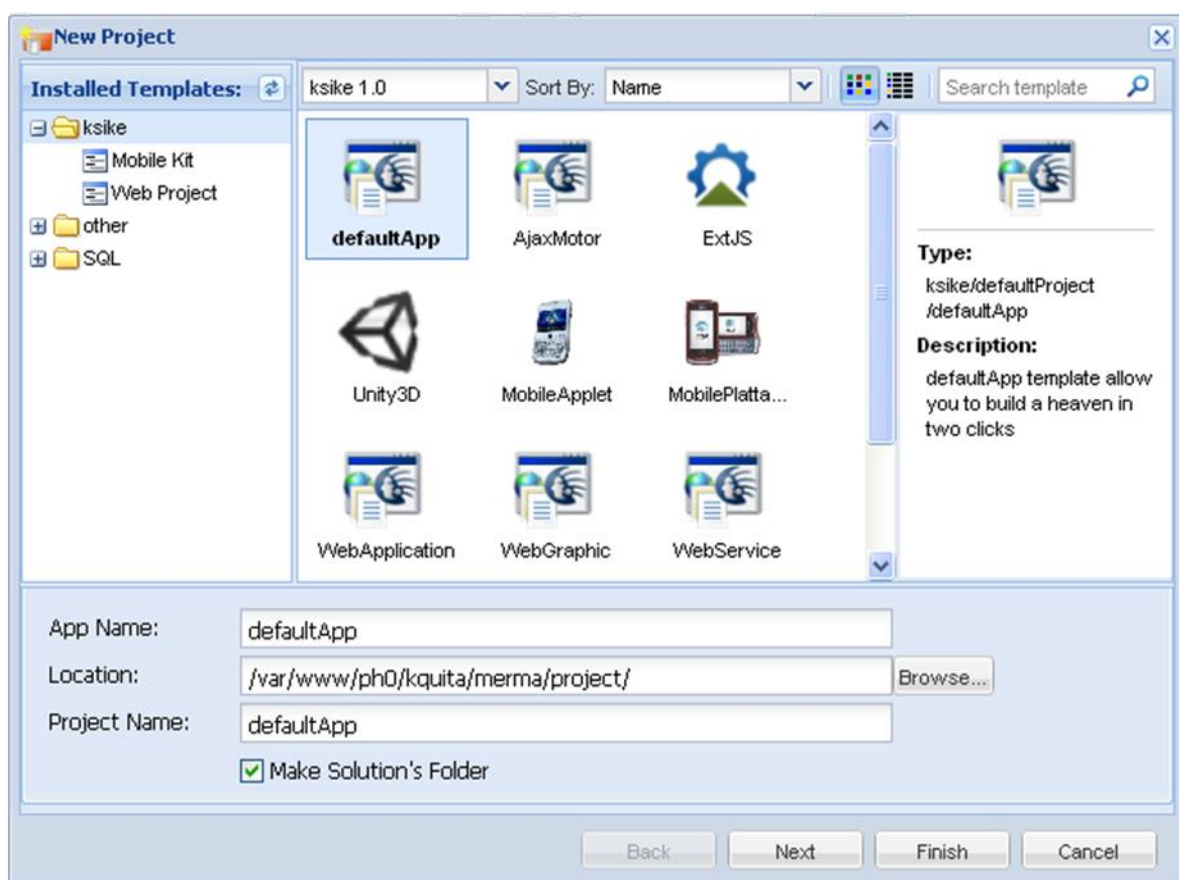


Figura 3: Interfaz de creación de proyectos basada en la interfaz de Visual Studio 2005.

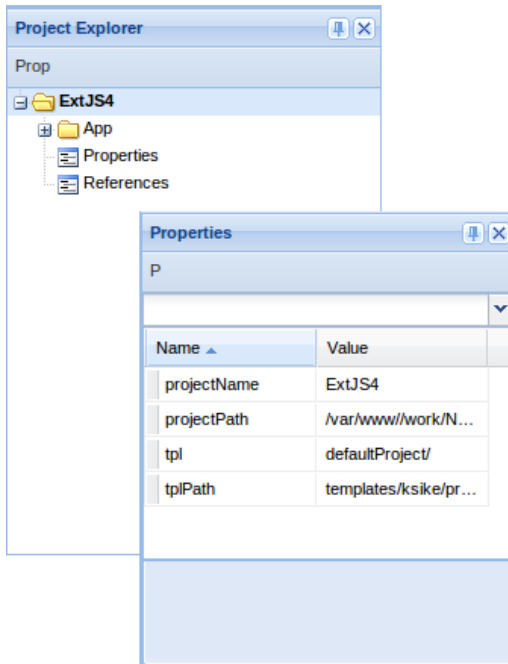


Figura 4: Estructura de un proyecto abierto en Bhike. La imagen representa la visualización de la estructura del proyecto “ExtJS4” en el Explorador de Proyecto de la aplicación y algunas de las propiedades asociadas al mismo. (Conceptos e interfaces basados en Visual Studio 2005).

Evolución y desarrollo:

Luego de quedar planteada la idea de desarrollar un IDE que permita llevar a cabo la implementación de la filosofía Ksike y quedar definida las características de este tipo de herramientas informáticas se ha llevado a cabo la identificación de las funcionalidades principales que debe proveer el sistema en una versión ligera. Como parte de las necesidades a cubrir sobre el consumo de los recursos que brinda la plataforma y de acuerdo a las planificaciones que existen para el versionado de la aplicación se han definido las siguientes, en forma de cronograma de nivel de desarrollo o madurez de módulos contra versiones del sistema Bhike. Destacar en el caso particular que el valor “5” denota el margen máximo de estabilidad de los módulos.

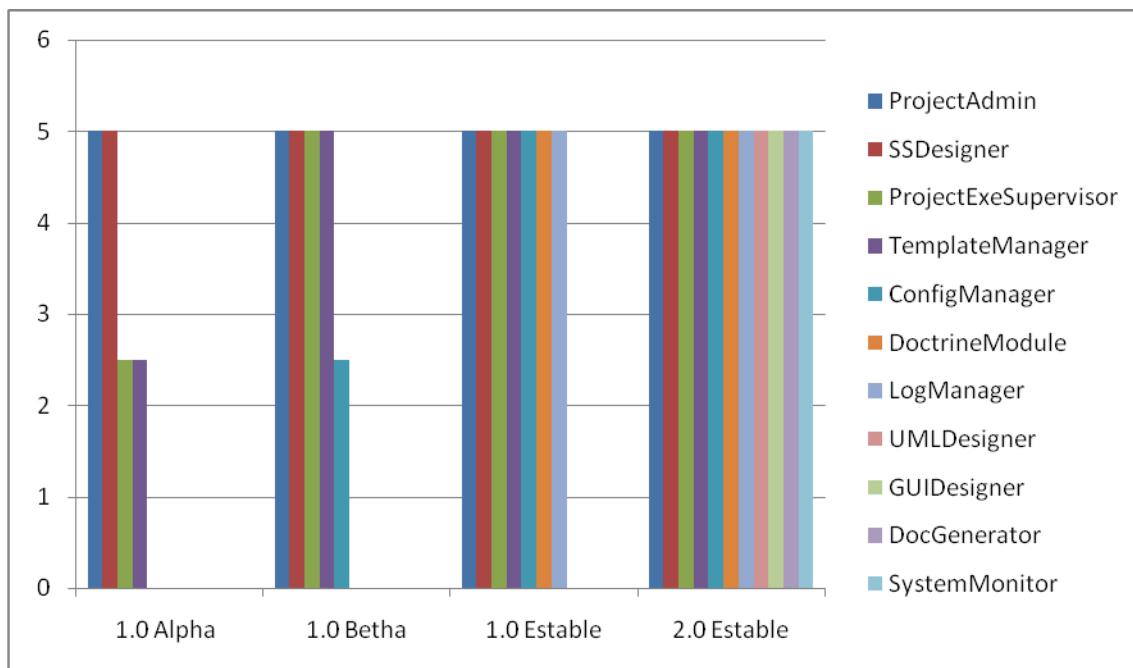


Figura 5 Cronograma de desarrollo por módulos de Bhike.

A continuación, se procede a explicar en qué consiste cada uno de los módulos que se presentan en el cronograma.

Módulos disponibles desde la versión 1.0 alpha:

ProjectAdmin: Es la principal funcionalidad que posee actualmente el sistema y que se basa en el manejo de la información asociada al concepto proyecto, información que consiste en tipo de aplicación, librerías, extensiones y recursos a emplear para su desarrollo. Actualmente permite:

- **Crear proyectos mediante plantilla:** Un Template o Plantilla es un componente que proporciona una separación entre la forma o estructura y su contenido. Es un medio que permite guiar, portar o construir un diseño o esquema predefinido. Agiliza el trabajo de reproducción de muchas copias idénticas o casi idénticas que no tiene que ser tan elaborado, sofisticado o personal. Si se quiere un trabajo más refinado, más creativo, la plantilla no es sino un punto de partida, un ejemplo, una idea aproximada de lo que se quiere hacer, o partes comunes de una diversidad de copias. A partir de la plantilla

pueden asimismo diseñarse y fabricarse nuevas plantillas. Con relación con los sistemas computacionales, utilizan en la actualidad un sistema de plantillas para separar la lógica del programa del formato visualizado. Típicamente, estas plantillas incluirán variables, y posiblemente unos pocos operadores lógicos para permitir una mejor adaptabilidad de la plantilla.

- El aditamento de gestión de proyecto permite la modificación de los datos asociados a los proyectos y la eliminación de los mismos. También permite importar proyectos existentes al IDE y exportar los productos finales.

SSDesigner: Es una herramienta que permite diseñar desde la interfaz gráfica de usuario el funcionamiento del recurso “Señales y Ranuras” que posee la plataforma Ksike en su API con orientación similar a las aplicaciones escritorio, proveyendo al desarrollador de la posibilidad de establecer comportamientos de comunicación y aviso entre módulos con mayor rapidez y claridad. En este tipo de aplicaciones es importante proveer de un mecanismo claro de comunicación entre objetos para minimizar el acoplamiento entre los diferentes módulos. Al igual que otros Framework como Qt, se proporciona un mecanismo basado en "Signals" y "Slots".

Módulos disponibles desde la versión 1.0 betha:

ProjectExecutionSupervisor: El módulo consiste en realizar un mecanismo para supervisar el desempeño de los proyectos y poder estimar desde el propio entorno Bhike, la generación de logs o fallos en el funcionamiento de los mismos, permitiendo darle seguimiento directo a la línea de programación que propiciaron los errores y otorgar la posibilidad de corrección a dichos desperfectos. Esta es una funcionalidad muy común en cada uno de los IDE analizados como parte del panorama actual en las tecnologías de entornos de desarrollo.

TemplateManager: Permite a los desarrolladores gestionar sus propias plantillas para la creación de proyectos, en cada caso seleccionado el tipo de aplicaciones a utilizar, así como el comportamiento y componentes bases que estimen conveniente. También permitirá crear nuevas categorías de proyectos y editar la información asociada a cada uno de los elementos anteriores, descripción librerías

a utilizar y demás. La creación de plantillas debe posibilitar la utilización de plantillas creadas anteriormente permitiendo extender comportamientos de las mismas, especializando comportamientos en las nuevas según sea la destinación de su empleo.

ConfigManager: Aunque la plataforma Ksike permite el trabajo con diferentes formatos de configuración sobre archivos (php, ini, json, xml), se dispone como una de las adiciones para la versión beta de Bhike, la incorporación del control sobre dichas extensiones para almacenar los datos generados como parte de la configuración del mismo, incluyéndose la posibilidad de manejar tanto la configuración de la herramienta Bhike en sí, como la configuración de los proyectos en desarrollo, plantillas de proyecto, plugins, drivers o el funcionamiento de Ksike. Para esta versión debe establecerse un enfoque principalmente dirigido al uso del formato XML siglas de eXtensible Markup Language (Lenguaje de Marcas Extensible). Este último posibilita la extensibilidad, lo que significa que después de diseñado y puesto en producción, se puede extender XML con la adición de nuevas etiquetas de forma que se pueda continuar usando sin complicaciones. El analizador de XML es un componente estándar, o sea que no es necesario utilizar uno nuevo para cada versión del lenguaje, lo que posibilita el empleo de cualquier analizador disponible y evita la existencia de bugs lo que acelera el desarrollo de aplicaciones. Se pueden comunicar aplicaciones de distintas plataformas, sin que importe el origen de los datos, por lo que se hace indispensable su uso.

DoctrineManager: Doctrine es un manipulador de objetos relacionales para PHP que provee la persistencia para los objetos de dicho lenguaje y que se encuentra en la parte superior de una capa de abstracción de bases de datos.

Doctrine 1.0 sigue el patrón de registros activos para el trabajo con información, donde una clase se corresponde con una tabla en una base de datos. En Doctrine 2.0, la última versión lanzada, las entidades no tienen que heredar de la clase base Active Record, sino que uno puede hacerlas persistentes pasándolas al Manejador de Entidades. Permite generar clases de objetos a partir de tablas en

las bases de datos, y el programador puede entonces especificar relaciones y añadir funcionalidades a las clases generadas. No es necesario generar o mantener complejos esquemas de bases de datos XML.

Una de las características principales de Doctrine es la opción de escribir consultas a la base de datos en un dialecto de objetos orientado a SQL denominado Doctrine Query Language (DQL). Además, no siempre es necesario escribir explícitamente consultas a la base de datos teniendo en cuenta que Doctrine puede realizar uniones y búsquedas automáticamente.

Otras características de Doctrine son:

- 1- Soporta información jerárquica.
- 2- Soporta ganchos y listeners para estructurar la lógica de negocio.
- 3- Uso de varios resguardos como memcached, SQLite o APC.
- 4- Una función compilar para combinar muchos archivos PHP del Framework en uno solo, para así evitar la pérdida de rendimiento que ocurre al incluir muchos archivos PHP del Framework.

Ksike consta de un plugin para el trabajo con Doctrine que permite lograr un manejo más rápido y sencillo de este último. Por lo tanto, es objetivo para Bhike brindar una interfaz gráfica para facilitar la utilización del plugin de Doctrine para hacer aún más ágil su uso.

LogManager: La forma efectiva de conocer lo sucedido ante una falla en el sistema cuando se ejecuta una petición, consiste en chequear la traza generada por el proceso que se ejecuta. La posibilidad de contar con un sistema de reportes de logs que implementa el Framework Ksike, debe ser aprovechada por la herramienta Bhike en aras de facilitar el proceso de desarrollo a los programadores, por lo se prevé la construcción de un módulo visual para la notificación y tratamiento de fallos.

Módulos disponibles desde la versión 1.0 estable:

UMLDesigner: Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. La programación orientada a objetos OOP resulta ser un complemento perfecto para la representación y análisis de problemas modelados por conceptos de clases.

La totalidad de los sistemas que pueden diseñarse sobre la plataforma Ksike presentan una orientación a objetos. Cada uno de los conceptos como: clase, objeto o instancia, método, atributo, propiedades, eventos, están descritos dentro de las filosofías de los lenguajes que emplea Ksike (PHP, Python), en el caso de Javascript que no se define como un lenguaje basado en el paradigma de programación orientado a objetos si existen algunos recursos que permiten simularlo y aprovechar de esta forma las potencialidades que este brinda.

Basándose en las enormes ventajas que brindan estos recursos se hace indispensable la construcción de una herramienta que permitirá diseñar clases UML con todas sus relaciones como se muestra en la figura 6 y efectuará generación instantánea de código según las filosofías que implementa la plataforma. Resultando en mejora de la comprensión sobre sistemas complicados y con estructuras de clases complejas.

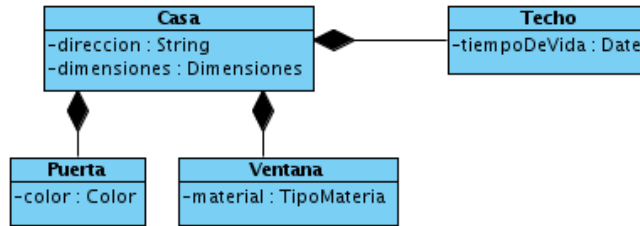


Figura 6: Problema de partes de una casa representado mediante UML.

Módulos disponibles desde la versión 2.0 estable:

GUIDesigner: El diseño de interfaces de usuario o ingeniería de interfaces de usuario, es el diseño de computadoras, maquinarias, dispositivos de comunicación Mobile, aplicaciones de software y sitios web con la concentración en la interactividad, así como la experiencia de los usuarios. El fin del diseño de interfaces de usuario es hacer de las mismas tan simples y eficientes como sean posibles, en términos de cumplir las metas de los usuarios.

La gran mayoría de los entornos de desarrollo de aplicaciones, poseen capacidades para el desarrollo de aplicaciones visuales, estos en sí poseen herramientas para el diseño de interfaces de presentación orientadas a los usuarios. Bhike contará con un módulo para el diseño de GUI, que deberá garantizar un diseño estándar de sus interfaces para cuales quiera que sean las tecnologías que se empleen a la hora de construirlas, permitiendo la reutilización de dichas interfaces en aplicaciones orientadas a todo tipo de entornos y plataformas. Con motivo de crear esta garantía, deberá establecerse un sistema que permita almacenar la información asociada a los diseños de interfaces de forma estándar (XML) y que permita generar mediante el uso de patrones de inversión de control (IOC) implementado en la plataforma Ksike, interfaces orientadas a componentes utilizando API como ExtJS, JQuery, en el caso de aplicaciones web, así como API para el caso de aplicaciones orientadas a Desktop o Mobile según muestra la figura.

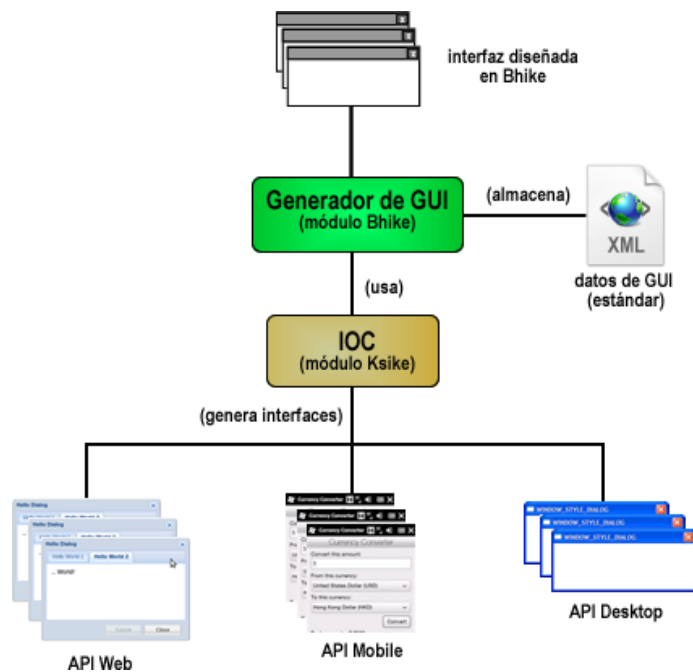


Figura 7: Esquema funcional de Generador de GUI.

DocGenerator: Como manifiesta José A. Mañas en su artículo Documentación de código, “Construir un módulo para la gestión de documentación de código fuente debido a la gran necesidad de generar la documentación de las fuentes de los proyectos. Documentar el código de un programa es añadir suficiente información como para explicar lo que hace, punto por punto, de forma que no solo los ordenadores sepan que hacer, sino que además los humanos entiendan que están haciendo y por qué. Es además una necesidad que solo se aprecia en su debida magnitud cuando hay errores que reparar o hay que extender el programa con nuevas capacidades o adaptarlo a un nuevo escenario.” (Mañas, 2003)

Por una u otra razón, todo programa que tenga éxito será modificado en el futuro, bien por el programador original, bien por otro programador que le sustituya. Pensando en esta revisión de código es por lo que es importante que el programa se entienda para poder repararlo y modificarlo. (Mañas, 2003)

Actualmente existen aplicaciones para la generación de documentación en base al código implementadas en lenguaje PHP, PHPDocumentor, PHPDoc, que presentan un estándar para la generación de documentación. Existen soluciones

con mayor escalabilidad como Doxygen, muy conocido en el ámbito de Java/C++ y PHP. Entre las funcionalidades de este último se encuentran: la creación de documentación en lista y jerarquía de clases, por archivos, módulos o estructuras de datos. Además, provee la funcionalidad de generar archivos HTML, CHM, RTF, PDF, LaTeX, PostScript o Man Pages.

Estos sistemas constituyen guías para el trabajo que se debe desarrollar para tratar de cumplir con estas especificaciones y poder incluir algunas nuevas que surjan como necesidades potenciales.

SystemMonitor: JMeter es un proyecto de Apache Jakarta que puede ser utilizado como una herramienta de pruebas y medir el desempeño de una gran variedad de servicios, orientados principalmente a la web. Puede ser usado como una herramienta de pruebas para conexiones con bases de datos con JDBC, FTP, LDAP, Web Services, JMS, HTTP y conexiones TCP genéricas. Basándose en estas funcionalidades se prevé la implementación de una herramienta de monitoreo sobre los proyectos en desarrollo en el entorno Bhike. Debe brindar además información sobre consumo de recursos de memoria, así como tiempo de respuestas a peticiones, como implementan otras plataformas como Symfony. Contribuyendo al desarrollo confiable sobre la herramienta y basado principalmente a mejorar el rendimiento de las aplicaciones.

Planes de migración:

Hasta el presente punto se han analizado un enorme conjunto de requerimientos funcionales que debe presentar la aplicación Bhike y que en el primer momento de su desarrollo va a estar orientado a entornos Web, pero con motivo de aprovechar las capacidades y recursos que presentan entornos como NetBeans, Eclipse, Zend Studio, como son depuradores, analizadores sintácticos y semánticos implementados en los mismos, se prevé realizar una extensión a los alguno de los mencionados entornos para garantizar mayor robustez, eficiencia y aceptación por parte de los usuarios familiarizados con dichos entornos.

CONCLUSIONES:

Luego de haber realizado una evaluación de las posibilidades que posee el desarrollo de la tecnología Ksike, queda resaltar el conjunto de mecanismos empleados por el sistema Bhike que potencian sus atributos de calidad, el empleo de una arquitectura orientada a componentes posibilita la reusabilidad, permitiendo que los módulos diseñados sean utilizados en diferentes escenarios y por diferentes aplicaciones. De igual forma se garantiza la extensibilidad, escalabilidad, usabilidad y flexibilidad de todas sus partes.

La versión web emplea la API de ExtJS con el objetivo de otorgarle al usuario la sensación de aplicaciones de escritorio buscando mejorar la experiencia de los mismos con la aplicación, como lo realizan las aplicaciones enriquecidas de internet (RIA del inglés Rich Internet Applications) con el objetivo de que no constituya un choque cognitivo para los desarrolladores que tienen mayor preferencia y práctica en entornos de escritorio.

Entre sus módulos más relevantes se encuentra el de gestión de proyectos o ProjectAdmin, pues este constituye la piedra principal para el uso de la plataforma Ksike. Presenta como proyecto la idea de eliminar la barrera existente entre el desarrollo de aplicaciones web y escritorio, permitiendo gran usabilidad de dicha tecnología. Otra de sus funcionalidades claves es el diseñador de señales y ranuras, que permite la comunicación entre las partes servidoras de la lógica de los negocios aplicadas a los módulos de los sistemas.

De esta manera el empleo de la tecnología Ksike-Bhike servirá para desarrollar soluciones a cualquier escala, para los diferentes entornos y plataformas, logrando marcar una pauta en el desarrollo de proyectos informáticos.

BIBLIOGRAFÍA:

Espinosa, I. A. (2011). *Tribe Larva - Definición ontológica de Ksike 1.0 Larva*. Habana, Cuba.

Farangos, S. (s.f.). *W3C Tackles RIA and AJAX Accessibility*. Obtenido de <http://www.cmswire.com/cms/industry-news/w3c-tackles-ria-and-ajax-accessibility-000760.php>

Mañas, J. A. (8 de Mayo de 2003). *Documentación de código*. Obtenido de <http://www.lab.dit.upm.es/~lprg/material/apuntes/doc/doc.htm>

NetBeans Release Roadmap. (s.f.). Obtenido de <http://netbeans.org/community/releases/roadmap.html>

Nokia. (s.f.). *Qt-Cross-platform application and UI framework*. Obtenido de <http://qt.nokia.com/>

The Code::Blocks Team. (s.f.). *The open source, cross platform, free C++ IDE*. Obtenido de <http://www.codeblocks.org/>