

Desarrollo de una aplicación interactiva aplicando la metodología OpenUP/Basic como parte del Proyecto ESPE-GINGA

Ángel Quingaluisa, Jonathan Torres, Danilo Martínez y Santiago Salvador

Departamento de Ciencias de la computación, Escuela Politécnica del Ejército, Sangolquí, Ecuador, angel_quingaluisa@hotmail.com, idiiter@hotmail.com, danilo.martinez@espe.edu.ec, mssalvador@espe.edu.ec

RESUMEN: El proyecto ESPE-GINGA busca como objetivo difundir y motivar el desarrollo de contenidos interactivos para la TV digital en Latinoamérica, como parte de la Red Latinoamérica de Cooperación en Investigación, Desarrollo y Formación en el área de Software para TV Digital Interactiva. En este contexto, el Departamento de Ciencias de la Computación de la Escuela Politécnica del Ejército para apoyar al proyecto de implementación del Laboratorio de TV Digital Interactiva realizado por el Departamento de Eléctrica y Electrónica, decide iniciar la investigación del middleware Ginga-J. El presente trabajo propone la investigación y creación de una aplicación interactiva con el estándar Brasileño de TV Digital. Para llevar a cabo el proyecto se realizó la investigación de la arquitectura y funcionamiento del Middleware Ginga-J, además se utilizó el emulador OpenGinga, la arquitectura MVC, metodología OpenUP / Basic, lo que ha permitido obtener como resultado una aplicación interactiva digital cuyo contenido se alimenta de imágenes y archivos XML, demostrando la potencia que tiene el API Ginga-J.

Palabras claves: Middleware, Ginga J, OpenGinga, OpenUP / Basic, API.

ABSTRACT: The project ESPE-GINGA, as part of the Latin America Network of Cooperation on Research and Development of Software for digital TV, tries to introduce and spread the use of new interactive modules for digital TV in Latin America. The Computer Science Department of the Escuela Politécnica del Ejército launch with the research of the Ginga-J middleware to support the project and set up of the Digital TV laboratory managed by the Electric & Electronics' Department, decide to start the research of the middleware Ginga-J. This project creates a digital interactive application using the Brazilian model of digital TV. It was necessary a deep study of the operation of middleware Ginga-J architecture besides, the use of Open Ginga emulator, MVC architecture and OpenUP/Basic. The final product is an interactive digital application that is feed by images and XML files which shows the power of API Ginga-J.

Keywords: Middleware, Ginga J, Open Ginga, OpenUP / Basic, API.

1. INTRODUCCIÓN

En el Ecuador, se resolvió adoptar el estándar Japonés-Brasileño ISDB-T/SBTD como sistema de Televisión Digital Terrestre (TDT). Se oficializó el 24 de marzo del 2010 por parte de la Superintendencia de Telecomunicaciones (SUPTTEL). La interactividad es la parte fundamental del sistema TDT, permitiendo al usuario interactuar con el emisor a través del control remoto, estableciendo la comunicación entre ellos.

El Departamento de Ciencias de la Computación en conjunto con el Departamento de Eléctrica y Electrónica de la Escuela Politécnica del Ejército (ESPE) han decidido impulsar la investigación y desarrollo de aplicaciones interactivas para televisión digital utilizando el middleware GINGA y el API GINGA-J como parte de un proyecto de la Carrera de Ingeniería en Sistemas de la ESPE.

El proyecto ESPE–GINGA, actualmente está orientado a la investigación de aplicaciones interactivas desarrolladas con el middleware Ginga-NCL, pero todavía se desconoce el funcionamiento del recién creado Ginga-J. Por tal motivo se desarrolla una aplicación de tipo direccional bajo el Middleware Ginga-J, las pruebas se realizaron en la máquina virtual OpenGinga.

El presente trabajo propone la investigación y creación de una aplicación interactiva con el estándar Brasileño de TV Digital. Para llevar a cabo el proyecto se realizó la investigación de la arquitectura y funcionamiento del Middleware Ginga-J, además se utilizó el emulador OpenGinga, la arquitectura MVC, metodología OpenUP / Basic, lo que ha permitido obtener como resultado una aplicación interactiva digital cuyo contenido se alimenta de imágenes y archivos XML, demostrando la potencia que tiene el API Ginga-J.

Para el estudio, investigación y desarrollo de una aplicación con el Middleware Ginga-J se utilizó las siguientes metodologías: “Investigación Aplicada” y “OpenUp/Basic”, las cuales permitieron aprovechar los conocimientos ya desarrollados para aplicar en este proyecto de investigación y en la definición de roles, iteraciones y actividades que se dio a cada uno de los integrantes del proyecto para el cumplimiento de las fases de desarrollo.

El resto del artículo ha sido organizado como sigue: La sección 2 describe los fundamentos teóricos que sustentan esta investigación. La sección 3 explica los materiales. La sección 4 detalla el aspecto metodológico. En la sección 5 se evalúan los resultados. Finalmente en la sección 6 se exponen las conclusiones y trabajo futuro.

2. FUNDAMENTOS TEÓRICOS.

Ginga-J es un middleware desarrollado para el estándar Brasileño de TV digital el cual fue acogido por Ecuador. Ginga-J es una de las principales herramientas de desarrollo para aplicaciones interactivas que utilizan programación Java, el mismo que por su arquitectura logra una compatibilidad total sobre las mismas debido a sus APIs. Al momento el desarrollo de aplicaciones en Ginga-J es muy limitado ya que la versión de la máquina virtual de Java es muy básica

2.1 Arquitectura Ginga-J.

Las aplicaciones nativas se pueden ejecutar sin necesidad de contar con las funcionalidades del sistema operativo Ginga o por la aplicación Ginga. Sin embargo también se pueden utilizar los APIS estándares de Ginga-J. Las aplicaciones Xlets deben utilizar los Apis de Ginga-J, como se muestra en la Figura 1. Con el fin de mantener la compatibilidad con el API de GEM [1], Ginga-J se basa en tres grandes grupos de APIS:

- i) API Verde (APIs compatibles con GEM), aquí se encuentran incluyendo las APIs provenientes de los paquetes de Sun JavaTV, DAVIC [DAVIC, 1999] e HAVI [HAVI, 2001].
- ii) API Amarillo (compuesto por el JMF 2.1 API), lo cual es necesario para el desarrollo de aplicaciones, con captura de sonido)

iii) API azul permite al receptor de TV Digital comunicarse con cualquier dispositivo con una interfaz compatible (con conexión con cable, como Ethernet o PLC, de red o inalámbrica, como infrarrojos o Bluetooth), aquí se encuentra el API que permite el desarrollo de las aplicaciones Ginga-J que tengan Ginga-NCL (API puente); como se muestra en la Figura 2.

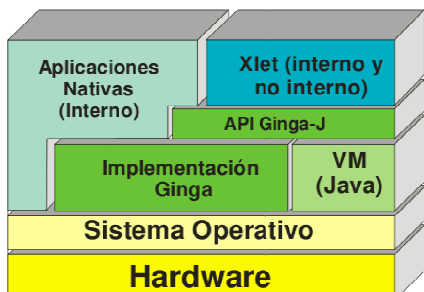


Figura 1. Arquitectura General Ginga-J

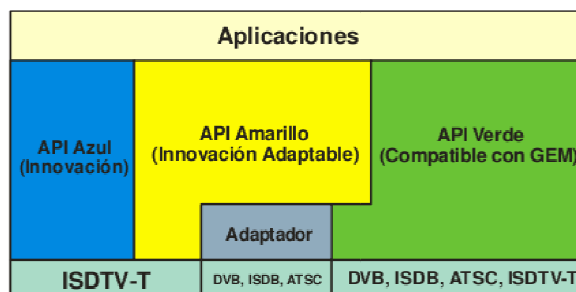


Figura 2. API's GINGA JAVA

2.2 Ginga-Ncl.

Ginga-NCL es el subsistema del Middelware Ginga el cual permite procesar documentos NCL. Un componente clave de Ginga-NCL es el motor decodificador de contenidos declarativos. La especificación de este subsistema se base en las normas ABNT NBR 15606-2 y ABNT NBR 15606-5.

2.3 Ginga-J.

Ginga-J es el subsistema del Middelware Ginga que proporciona una infraestructura para la ejecución de aplicaciones de TV Digital, el ambiente interactivo Ginga-J ofrece soporte a las aplicaciones desarrollas con el lenguaje de programación Java. La especificación de este subsistema se basa en la norma ABNT NBR 15606-4[2], que mantiene compatibilidad con la mayoría de *middlewares* de TV Digital actuales. [3]

2.4 OpenGinga.

OpenGinga es una máquina virtual creada con Virtual Box y se basa en el Sistema Operativo Ubuntu, con una interfaz gráfica que nos permite ejecutar aplicaciones Ginga-J y GINGANCL en ambiente de PC.

3. MATERIALES.

Para demostrar las diferentes ventajas del Middelware Ginga-J se hizo uso de las siguientes herramientas para el desarrollo de una aplicación interactiva de Televisión Digital Ver Tabla 1.

TABLA 1. Herramientas de desarrollo.

Herramienta	Utilidad
Eclipse	Herramienta para el desarrollo del código fuente de la aplicación.
OpenGinga	Máquina Virtual utilizada para desarrollar y ejecutar aplicaciones de TV Digital.
XML	Archivos de gestión de datos.
GINGA-J	Estas librerías se encuentran divididas de la siguiente manera: <ul style="list-style-type: none"> ✓ El núcleo también llamada API's verde, estas son responsables de mantener el sistema compatible lo máximo posible. ✓ El módulo responsable por el soporte son las llamadas APIs roja y amarilla, las API's amarillas pueden ser exportadas en otros sistemas, soporte a múltiples usuarios, dispositivos y redes, mientras que las API's rojas son el soporte a las necesidades específicas de aplicaciones para Brasil, en especial aplicaciones de inclusión social.

4. IMPLEMENTACION DE LA APLICACIÓN INTERACTIVA UTILIZANDO API GINGA-J.

Se ha desarrollado una aplicación interactiva de TV Digital utilizando un parser XML como canal de retorno, para la visualización de contenidos de los diferentes departamentos de la Escuela Politécnica del Ejército, a través de la navegación de imágenes que se encuentran definidas mediante el archivo de propiedades, basándonos en el paradigma de programación orientada objetos con la estructura Xlet.

Para el desarrollo de esta aplicación se utilizó eventos propios del API Ginga-J, las cuales en la pantalla de interactividad permite realizar una navegación de los diferentes departamentos. Esta aplicación usa como canal de retorno un parser XML mediante el consumo del componente *kxml2-min-2.3.0.jar*; donde el telespectador mediante el teclado del computador simula un control remoto; para ingresar a la pantalla de interactividad se pulsa la tecla M, como se muestra en la Figura 3.



Figura 3. Pantalla inicial de la aplicación interactiva.

A manera de ejemplo, en este aplicativo cuando el telespectador se desplace con las teclas de navegación por los diferentes “Departamentos”, se visualizará los contenidos que tiene almacenado el archivo XML con la utilización del parser. Como resultado se obtendrá la información de cada uno de los diferentes Departamentos, como se muestra en la Figura 4.



Figura 4. Resultado de la búsqueda de la Ingeniería de Software.

5. EVALUACION DE RESULTADOS.

Ginga-J es un middleware que al estar en desarrollo es un Api muy potente el cual nos permitió el desarrollo de nuestra aplicación mediante la construcción de la misma utilizando lenguaje de programación JAVA.

Para el desarrollo en Ginga-J se puede utilizar cualquier IDE de programación compatible con Java, la única regla es que las librerías de Ginga-J y GingaCore deben estar definidas de forma correcta para no tener problemas al momento de la construcción y la compilación.

Para realizar las pruebas de la aplicación se debe tomar la decisión de que emulador se utilizara, para ello se elaboro una matriz de comparación entre los dos principales emuladores de Ginga-J: XletView y Ginga-J.

La Tabla 2 se muestra los puntajes y la comparación que se realizó tomando en cuenta algunos factores importantes como son la portabilidad, la interacción con el usuario y la capacidad de entender y utilizar el mismo.

TABLA 2. Matriz de Comparación XletView vs OpenGinga.

Medidores:				
Se medirá con puntajes de 0 al 10 siendo el ultimo la máxima puntuación				
		Prioridad:	Alta	
			Normal	
			Baja	
No	FACTOR	EMULADORES		
		XletView	OpenGinga	General
1	Fácil de Instalar	4	1	10
2	Multiplataforma(Linux, Windows, MacOS)	3	3	10
3	Software de código abierto	10	10	10
4	Cantidad Documentación en ingles	4	2.5	10
5	Cantidad Documentación en español	2	0.5	10
6	Cantidad Documentación en portugués	4	7	10
7	Estructuración de los path del emulador	1	9	10
8	Facilidad de interacción y comprensión por el usuario	6	8	10
9	Facilidad de instalación de las aplicaciones	8	9	10
10	Facilidad de comprensión para el usuario	8	8	10
11	Facilidad de portabilidad (Código, Herramientas, Emulador).	3	10	10
12	Facilidad de compilación de código.	9	9	10
13	Facilidad de utilización de aplicación de terceros (terminal, eclipse).	10	10	10
14	Compatibilidad con JAVA TV.	10	10	10
15	Utiliza Herramientas libres para su uso, S.O, máquinas virtuales, etc.	5	10	10
16	TOTAL SUMA:	87	107	150
17	TOTAL PORCENTAJE:	58.00%	71.33%	100.00%
18	TOTAL PRIORIDAD ALTA:	31	38	60
19	TOTAL PRIORIDAD NORMAL:	17	20	30
20	TOTAL PRIORIDAD BAJA:	39	49	60
20	TOTAL PRIORIDAD:	87	107	150

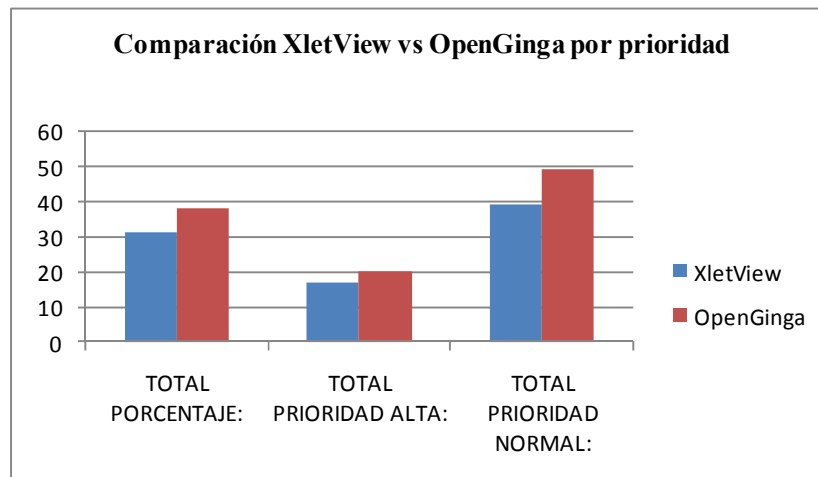


Figura 5. Comparación XletView vs OpenGinga por prioridad

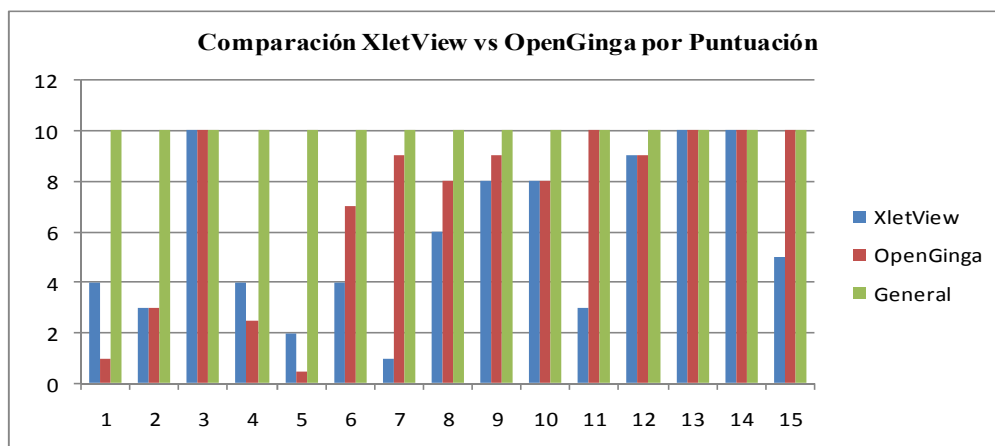


Figura 6. Comparación XletView vs OpenGinga por puntuación

La comparación entre las dos herramientas nos lleva a la conclusión que las dos tienen una funcionalidad semejante, que es la facilidad de interacción con el usuario.

El punto más importante por el cual se elige el emulador OpenGinga es su portabilidad ya que al ser una máquina virtual, posee todas las herramientas necesarias para el desarrollo compilación y ejecución de las aplicaciones interactivas construidas con Ginga-J.

Algo importante sobre la utilización del emulador es que no se necesita comprar licencias para utilizarlo ya que al momento todas las herramientas necesarias para su ejecución se distribuyen de manera libre.

6. CONCLUSIONES Y TRABAJO FUTURO

La metodología OpenUp / Basic permite el desarrollo de aplicaciones de corto alcance con un grupo pequeño de personas ya que las mismas pueden desempeñar varios roles, así mismo con el uso de las iteraciones en cada una de las fases de desarrollo se puede realizar correcciones preventivas de cada componente a desarrollar. La versión actual del emulador OpenGinga tiene limitaciones en la librería API HAVI, por tal razón los objetos gráficos con los que se puede trabajar en el desarrollo de aplicaciones para tv digital aun son limitados. El desarrollo de una aplicación en Ginga-J conlleva entender la estructura principal de un Xlet, el cual almacena los componentes gráficos, los mismos que para ser programados, el desarrollador debe tener un conocimiento amplio sobre programación orientada a objetos empleando el

lenguaje de programación JAVA. Los principales problemas que detectamos dentro del desarrollo de este tipo de aplicaciones son 2 que consideramos son de alta importancia, el primero es la restricción en el uso de memoria por lo cual la ejecución de las aplicaciones es limitada y las aplicaciones pueden colapsar; la segunda es la versión de la máquina virtual de java que utiliza el emulador aun se encuentra en una versión básica de tal manera que para realizar la programación orientada a objetos resulta más compleja. El middleware Ginga-J no se puede instalar fácilmente en una PC con Sistema Operativo Ubuntu, por esta razón se utiliza la máquina virtual con Ginga-J pre-instalado que nos permite levantar un ambiente rápido, siendo esta una opción que está disponible al momento y de esta manera desplegar aplicaciones Ginga J. Para el buen funcionamiento de la arquitectura del Middleware Ginga-J se debe usar las funcionalidades que tienen los API's estandarizadas Ginga-J, ya que los Xlets para su funcionamiento deben utilizar API's estandarizados provistos por Ginga-J.

Para el desarrollo adecuado de aplicaciones de corto alcance se recomienda el uso de la metodología de desarrollo OpenUP / Basic, ya que si se cumple los roles y las iteraciones que se definen en el tiempo estimado tendremos el control que se necesita en el desarrollo de sistemas informáticos. El Proyecto ESPE-GINGA debe realizar una investigación de la incompatibilidad de los componentes gráficos desarrollados con el API HAVI en la versión actual de OPENGINGA, para de esta manera poder realizar el desarrollo de librerías genéricas compatibles que mejoren el rendimiento de dichos componentes gráficos y compartir a la comunidad Ginga j. Para comenzar el desarrollo de aplicaciones de tv digital basadas en Ginga-J se recomienda analizar la estructura de un Xlet aplicando los conocimientos del lenguaje Ginga-J orientado a objetos. El proyecto ESPE-GINGA debe investigar las limitaciones de memoria en el emulador para de esta manera aportar en la investigación del proyecto OpenGinga de Lavid.

Referencias Bibliográficas

- [1] Políticas de asignación de buffers de memoria. URL: <http://dSPACE.ups.edu.ec/bitstream/123456789/144/4/Capitulo%203.pdf>
- [2] GingaEcuador Hpage: URL <http://www.ginga.org.ec/gingaec/index.php?article&id=45&Itemid=54>
- [3] http://www.dtv.org.br/download/es-es/ABNTNBR15606-4_2010Esp_2010.pdf
- [4] GingaPerú Home Page: URL: <http://www.gingaperu.org/tvd-y-ginga/ginga.html>
- [5] <http://wiki.solar.org.ar/publico:ginga>
- [6] http://www.pleiad.cl/_media/research/adi/tvd-desarrollo.pdf
- [7] http://aat.inictel-uni.edu.pe/files/SET_TOP_BOX%28Informe_de_Avance1%29.pdf

La importancia de la Gestión de la Configuración del Software, en una Empresa de Desarrollo.

Santiago Paredes, Cecilia Hinojosa, Jenny Ruiz

*Departamento de Ciencias de la Computación, Escuela Politécnica del Ejército, Ecuador,
ogaitnas6@hotmail.com, cmhinojosa@espe.edu.ec, jaruiz@espe.edu.ec*

RESUMEN: La Gestión de la Configuración del Software (GCS) es un proceso de ingeniería, que indica procedimientos técnicos para el control y mejora de la calidad del software. Los subprocesos o fases que lo conforman son: Identificación de la Configuración, Control de Cambios, Control de Versiones, Informes de Estado, Auditoría de la Configuración. Cada una de las fases tiene parámetros para procesos de control que buscan la integración de las actividades relacionadas con el desarrollo del software desde las primeras fases, asignando roles y responsabilidades al personal de trabajo, para llegar a obtener un producto software de calidad; de esta manera se puede realizar un cambio o mantenimiento de una manera eficiente. Como contribución, el presente trabajo muestra el diseño de una metodología personalizada de GCS para una empresa, la cual fue realizada en base a un análisis comparativo entre el estándar ISO/IEC 12207, la metodología Métrica Versión 3, y el proceso RUP. Como resultado se han obtenido procesos claramente definidos para la identificación de los elementos de configuración, control de cambios y el control de versiones mediante herramientas software.

Palabras Clave: Gestión de la Configuración del Software, Control de Versiones, Identificación de la Configuración, Control de Cambios.

ABSTRACT: Software configuration management (SCM) is a software engineering process, which gives technical procedures to control and increase software quality. The sub process or phases are: Configuration identification, change control, versioning control, status reporting, and configuration audit. Each one of these phases set parameters for procedures and control in order to integrate development activities since the beginning of software life cycle, establish roles and team responsibilities so we can get a software quality product so we could easily do a change or give maintenance in a efficient way. The main the contribution of this paper is to design a personalized approach to SCM for a company, which was carried out based on a comparative analysis of the ISO / IEC 12207, the methodology metric version 3, and RUP process. As a result we were obtained, a clearly defined process for the identification of configuration items, change control and version control using software tools.

KeyWords: Software Configuration Management, Versioning Control, Configuration Identification, Change Control