

CONSTRUCCIÓN DEL ESTADO DEL ARTE DE LA PROGRAMACIÓN POR
COMPONENTES

ARTURO CARDONA LONDOÑO
MAURICIO MARÍN JIMÉNEZ

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA
2013

CONSTRUCCIÓN DEL ESTADO DEL ARTE DE LA PROGRAMACIÓN POR
COMPONENTES

ARTURO CARDONA LONDOÑO
MAURICIO MARÍN JIMÉNEZ

MONOGRAFIA PARA OPTAR AL TÍTULO DE INGENIEROS EN EL PROGRAMA
DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

ASESOR
ÁNGEL AUGUSTO AGUDELO ZAPATA
DOCENTE ACADÉMICO

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA
2013

Nota de aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Pereira, 23 de Septiembre del 2013

Dedicatoria

A mi Mamá María Cristina, por acompañarme y apoyarme a lo largo de toda la carrera.

A mi Papá Arturo, por ser un apoyo incondicional con el cual sé que cuento para toda la vida.

A mis hermanas Gina y Angela por todas nuestras experiencias de vida, las quiero mucho.

A mi novia Laura por acompañarme en los buenos y malos momentos, y animarme cada día a ser mejor.

A mis amigos Migue, Darwin, Yeicol, Mafe, Angela y Juli, por ser mis compañeros en el estudio y mis amigos para toda la vida.

Arturo Cardona Londoño

A mi familia por ser un gran apoyo y acompañarme a lo largo de toda la carrera.

Mauricio Marín Jiménez

Agradecimientos

- Agradecemos a la ingeniera Luz Stella Valencia Ayala por su colaboración en el proceso de la formulación del proyecto.
- Agradecemos a nuestro asesor Ángel Augusto Agudelo por el apoyo para la realización de este trabajo.
- Agradecemos a todos nuestros profesores por todas sus enseñanzas para nuestras vidas profesionales.

TABLA DE CONTENIDO

	pág.
LISTA DE FIGURAS	10
LISTA DE TABLAS	11
LISTA DE ANEXOS	12
1. INTRODUCCIÓN	13
2. FORMULACIÓN DEL PROBLEMA	14
3. JUSTIFICACIÓN	16
4. OBJETIVOS	17
4.1. OBJETIVO GENERAL	17
4.2. OBJETIVOS ESPECÍFICOS	17
5. MARCO DE REFERENCIA	18
5.1 MARCO CONCEPTUAL	18
5.2 MARCO HISTÓRICO O DE ANTECEDENTES	22
5.3 MARCO TEÓRICO	23
6. INSTRUMENTO DE MEDICIÓN PARA EVALUAR FRAMEWORKS ORIENTADOS A COMPONENTES	28
6.1 USABILIDAD	28
6.2 SOPORTE	31
6.3 COMPATIBILIDAD	32
6.4 INTEROPERABILIDAD	33
6.5 ACCESIBILIDAD	33
7. .NET	35
7.1 HISTORIA	35
7.2 CARACTERÍSTICAS TÉCNICAS	36
7.3 USABILIDAD	38
7.4 SOPORTE	40
7.5 COMPATIBILIDAD	40
7.6 INTEROPERABILIDAD	40
7.7 ACCESIBILIDAD	40
8. AFC – APPLICATION FOUNDATION CLASSES	41
8.1 HISTORIA	41

8.2 CARACTERÍSTICAS TÉCNICAS	41
8.3 USABILIDAD	43
8.4 SOPORTE	43
8.5 COMPATIBILIDAD	44
8.6 INTEROPERABILIDAD	44
8.7 ACCESIBILIDAD	44
9. BONOBO	45
9.1 HISTORIA	45
9.2 CARACTERÍSTICAS TÉCNICAS	45
9.3 USABILIDAD	46
9.4 SOPORTE	46
9.5 COMPATIBILIDAD	46
9.6 INTEROPERABILIDAD	46
9.7 ACCESIBILIDAD	47
10. COM – COMPONENT OBJECT MODEL	48
10.1 HISTORIA	48
10.2 CARACTERÍSTICAS TÉCNICAS DE COM (COM, DCOM, COM+ Y ACTIVE X)	48
10.3 USABILIDAD	49
10.4 SOPORTE	50
10.5 COMPATIBILIDAD	50
10.6 INTEROPERABILIDAD	50
10.7 ACCESIBILIDAD	50
11. CORBA – COMMON OBJECT REQUEST BROKER ARCHITECTURE	51
11.1 HISTORIA	51
11.2 CARACTERÍSTICAS TÉCNICAS.	51
11.3 USABILIDAD	53
11.4 SOPORTE	55
11.5 COMPATIBILIDAD	55
11.6 INTEROPERABILIDAD	56
11.7 ACCESIBILIDAD	56
12. EJB - ENTERPRISE JAVA BEANS	57
12.1 HISTORIA	57

12.2 CARACTERÍSTICAS TÉCNICAS	57
12.3 USABILIDAD	59
12.4 SOPORTE	61
12.5 COMPATIBILIDAD	61
12.6 INTEROPERABILIDAD	62
12.7 ACCESIBILIDAD	62
13. GCF - GENERIC COMPONENT FRAMEWORK	63
13.1 HISTORIA	63
13.2 CARACTERÍSTICAS TÉCNICAS	63
13.3 USABILIDAD	64
13.4 SOPORTE	65
13.5 COMPATIBILIDAD	65
13.6 INTEROPERABILIDAD	65
13.7 ACCESIBILIDAD	66
14. KDE	67
14.1 HISTORIA	67
14.2 CARACTERÍSTICAS TÉCNICAS	67
14.3 USABILIDAD	68
14.4 SOPORTE	69
14.5 COMPATIBILIDAD	69
14.6 INTEROPERABILIDAD	69
14.7 ACCESIBILIDAD	69
15. UNO - UNIVERSAL NETWORKING OBJECTS	70
15.1 CARACTERÍSTICAS TÉCNICAS	70
15.2 USABILIDAD	71
15.3 SOPORTE	71
15.4 COMPATIBILIDAD	72
15.5 INTEROPERABILIDAD	72
15.6 ACCESIBILIDAD	72
16. XPCOM - CROSS PLATFORM COMPONENT OBJECT MODEL	73
16.1 HISTORIA	73
16.2 CARACTERÍSTICAS TÉCNICAS	73
16.3 USABILIDAD	73

16.4 SOPORTE	74
16.5 COMPATIBILIDAD	75
16.6 INTEROPERABILIDAD	75
16.7 ACCESIBILIDAD.	75
17. SÍNTESIS DE LOS RESULTADOS	76
17.1 USABILIDAD	76
17.2 SOPORTE	77
17.3 COMPATIBILIDAD	79
17.4 INTEROPERABILIDAD	80
17.5 ACCESIBILIDAD	82
18. CONCLUSIONES	84
BIBLIOGRAFÍA	85
ANEXO A. PRUEBAS DE SOPORTE	89
A.1 .NET	90
A.2 AFC	90
A.3 BONOBO	91
A.4 COM	91
A.5 CORBA	92
A.6 GCF	93
A.7 EJB	93
A.8 KDE	95
A.9 UNO	95
A.10 XPCOM	95

LISTA DE FIGURAS

	pág.
Figura 1. Interfaces de un Componente de software	19
Figura 2. .Net	35
Figura 3. .Net framework en contexto	37
Figura 4. Calculadora.....	39
Figura 5. Calculadora sumando	39
Figura 6. Jerarquía de AFC	42
Figura 7. Jerarquía de clases de los elementos de la interfaz de usuario proporcionadas por las AFC.	42
Figura 8. Petición de un Cliente a un Objeto	52
Figura 9. Diagrama de una invocación remota	52
Figura 10. ORBD de Corba.....	54
Figura 11. Servidor en java	54
Figura 12. Consola donde se invoca el cliente en java	55
Figura 13. Aplicación Web de una calculadora en EJB	60
Figura 14. Operación "Restar"	60
Figura 15. Resultado de la resta	61
Figura 16. UNO.....	70

LISTA DE TABLAS

	pág.
Tabla 1. Usabilidad	76
Tabla 2. Puntuación usabilidad	77
Tabla 3. Soporte	78
Tabla 4. Puntuación Soporte	78
Tabla 5. Compatibilidad	79
Tabla 6. Puntuación Compatibilidad	80
Tabla 7. Interoperabilidad	81
Tabla 8. Puntuación Interoperabilidad	81
Tabla 9. Accesibilidad	82
Tabla 10. Puntuación Accesibilidad	82

LISTA DE ANEXOS

	pág.
ANEXO A. PRUEBAS DE SOPORTE.....	89

1. INTRODUCCIÓN

La programación por componentes es de gran importancia para la ingeniería del software, ya que busca disminuir el tiempo de desarrollo y aumentar la calidad de los proyectos de software; varios estudios mencionan la poca cantidad de proyectos que son considerados exitosos cuando se utilizan las metodologías convencionales en la construcción de proyectos desde cero, motivo por el cual el desarrollo de software basado en componentes toma cada día más fuerza.

En esta investigación se construirá un instrumento de medición para evaluar de forma objetiva todos los frameworks orientados a la programación por componentes, analizando sus características y comparándolos con los demás, con el fin de poder determinar cuál es la mejor opción a elegir para desarrollar componentes de software según las necesidades que se tengan.

2. FORMULACIÓN DEL PROBLEMA

Desde las últimas dos décadas, el uso de las Tecnologías de la Información y la Comunicación (TIC's) en la sociedad se ha convertido en algo muy común, y el desarrollo de software ha sido una herramienta fundamental de apoyo para muchas tareas de la vida cotidiana, desde el vuelo de un avión hasta la planeación de proyectos empresariales; se han creado e implementado diferentes tipos de aplicaciones para resolver las necesidades de la sociedad, esto generalmente tiene una duración y costos significativos para las empresas, sin contar el alto grado de fallos que tienen los proyectos de desarrollo de software, como se puede evidenciar en el reporte que realizó "The Standish Group" en 1994 donde se mostraba que el 31.1% de los proyectos fracasaron, el 52.7% de los proyectos representaron un desafío para los equipos de trabajo y tan solo el 16.2% tuvieron éxito.¹

En 2008 según un estudio de la Universidad de Manchester, solo el 15% de los proyectos eran considerados exitosos, el 35% un fracaso y el 50% de ellos un fracaso parcial, Considerando como exitoso los proyecto en que "Los principales stakeholders del proyecto dicen que se han logrado los principales objetivos y no han experimentado resultados no deseados significativos" mostrando una vez más las complicaciones en los proyectos de desarrollo de software.² En el año 2009 "The Standish Group" realizó un informe llamado "Chaos Report" en el que mostraba que el 24% de los proyectos fracasan, el 44% de los proyectos son un desafío para los equipos de trabajo y tan solo el 32% tienen éxito.³ Aunque se han mejorado los resultados y la tasa de éxito ha superado por poco la de fracaso, casi la mitad de los proyectos representan un reto, de manera análoga podemos decir que un gran porcentaje de los proyectos de la actualidad no finalizan en buenos términos.

Aparte de los resultados anteriores, hay una creciente demanda de software para la realización de actividades cotidianas, con mayor complejidad, en menor tiempo, con menor costo, seguro, confiable, con un alto rendimiento y libre de errores garantizando la integridad de las operaciones que se realicen; Esto no es una tarea sencilla, pero actualmente se han generado alternativas para lograr este tipo de retos, una de ellas es la programación por componentes, estos pueden ser utilizados para resolver diversas necesidades según los requerimientos de

¹ Reporte del Caos [en línea]. The Standish Group, 1994 [citado el 10 de Febrero del 2013]. Disponible en: <<http://www.materiabiz.com/mbz/ityoperaciones/nota.vsp?nid=29672> >

² Comportamiento de proyectos de TI [en línea]. [citado el 10 de Febrero de 2013]. Disponible en: <<http://www.alejandrobarrros.com/content/view/691759/Comportamiento-de-proyectos-TI-Estan-en-deuda.html> >

³ Reporte del Caos [en línea]. The Standish Group, 2009 [citado el 10 de Febrero de 2013]. Disponible en: <http://www1.standishgroup.com/newsroom/chaos_2009.php >

desarrollo de software que se tengan, y acoplándolos correctamente se puede lograr la satisfacción del cliente.

Existen muchas formas para trabajar en la programación por componentes, cada una de ellas trabaja con metodologías y lenguajes de programación determinados, que hacen de sus desarrollos de software característicos de otros dependiendo de las necesidades de desarrollo que se presenten y los requisitos funcionales y no funcionales que deba cumplir, lo cual no se logra en la mayoría de los proyectos. ¿Cuál será la metodología de programación por componentes más apropiada para solucionar las necesidades del desarrollo de software actuales según las características propias que presenten los proyectos?

3. JUSTIFICACIÓN

La programación por componentes es un paradigma de programación por el cual cada uno de sus elementos es reutilizable, dejando en claro que reutilizable no significa utilizarse muchas veces, sino que sus elementos pueden ser utilizados para construir estructuras con funciones bien definidas en diferentes problemas de la programación. Cada componente es parametrizable de diferentes maneras según el objetivo que se desea alcanzar, definiendo unas entradas y obteniendo unas salidas de tal forma que solo se necesite acoplarlo al desarrollo total.

En la facultad de ingenierías, en el programa ingeniería de sistemas y computación, El grupo de avanzada de desarrollo de software (GRANDE) en conjunto con los estudiantes de pregrado y maestría, busca desarrollar un proyecto en el cual se construya un repositorio de componentes, que ayudará con el aprendizaje de los estudiantes, dándoles la capacidad de realizar desarrollos más grandes y cercanos a las necesidades empresariales, también brindará la posibilidad de aportar al repositorio con sus propios componentes; y a satisfacer algunas necesidades de software de la universidad teniendo a disposición componentes seguros y confiables.

Para la construcción de cada componente es necesario saber cuál es la mejor forma de hacerlo, y por medio de este proyecto se realizará una investigación sobre el estado del arte de la programación por componentes y algunos frameworks de trabajo actuales, como Corba, JavaBeans, XpCom, Kparts, GCF y .Net entre otros que podrían surgir, dando un amplio espectro para el análisis y criterios de decisión para elegir con que metodología se podrán construir los componentes de acuerdo a los recursos disponibles de la universidad, requisitos funcionales y no funcionales, entre otros aspectos; también le brinda a la comunidad universitaria y empresarial de la región aspectos válidos para elegir de qué forma deben realizar sus proyectos de desarrollo de software con respecto a la metodología o framework a utilizar según los requerimientos que deban solucionar.

4. OBJETIVOS

4.1. OBJETIVO GENERAL

Elaborar el estado del arte de todos los frameworks orientados a la programación por componentes y sus metodologías.

4.2. OBJETIVOS ESPECÍFICOS

- Construir un instrumento de medición para Framework de POC (Programación Orientada a Componentes), usando los estándares actuales y conocimientos adquiridos a lo largo de la carrera.
- Aplicar el instrumento de medición sobre cada Framework de POC.
- Construir un documento formal donde se resuma la experiencia propuesta mediante un cuadro comparativo de los Framework analizados, en base a sus características.

5. MARCO DE REFERENCIA

5.1 MARCO CONCEPTUAL

- **Componente:** Es una unidad de composición de aplicaciones software que posee un conjunto de requisitos, y que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto con otros componentes, de forma independiente en tiempo y espacio.⁴
- **Stakeholders:** Cualquier grupo o individuo que pueda afectar o ser afectado por el logro de los propósitos de una corporación. Stakeholders incluye a empleados, clientes, proveedores, accionistas, bancos, ambientalistas, gobierno u otros grupos que puedan ayudar o dañar a la corporación.⁵
- **Framework:** Un framework es a menudo una estructura en capas que indica qué tipo de programas se puede o se debe construir y cómo se interrelacionan. Algunos frameworks del sistema de cómputo también incluyen programas actuales, especifican interfaces de programación u ofrecen herramientas de programación para el uso de los frameworks. Un framework puede ser un conjunto de funciones dentro de un sistema y cómo se relacionan entre sí, las capas de un sistema operativo, las capas de un subsistema de aplicación, cómo debe ser estandarizada a un cierto nivel de una red, y así sucesivamente.⁶
- **Paradigma de programación:** Un paradigma de programación provee (y determina) la visión y métodos de un programador en la construcción de un programa o subprograma. Diferentes paradigmas resultan en diferentes estilos de programación y en diferentes formas de pensar la solución de problemas (con la solución de múltiples “problemas” se construye una aplicación).⁷
- **Repositorio:** Un repositorio, depósito o archivo es un sitio web centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos. Pueden contener los archivos en su servidor

⁴ SZYPERSKI, Clemence. Component Software. Beyond Object -Oriented Programming, 1998

⁵ FREEMAN, Edwar .Strategic Management: A Stakeholder Approach. Boston, 1984.

⁶ Framework [en línea]. [citado el 21 de Julio de 2013]. Disponible en:

<<http://whatis.techtarget.com/definition/framework> >

⁷ Paradigma de Programación [en línea]. 2009 [citado el 21 de Julio de 2013]. Disponible en:

<<http://labsys.frc.utn.edu.ar/ppr-2009/Unidad%20I%20-%20POO%20Avanzada/Apunte/Unidad%20I%20-%20POO%20Avanzada.pdf>>

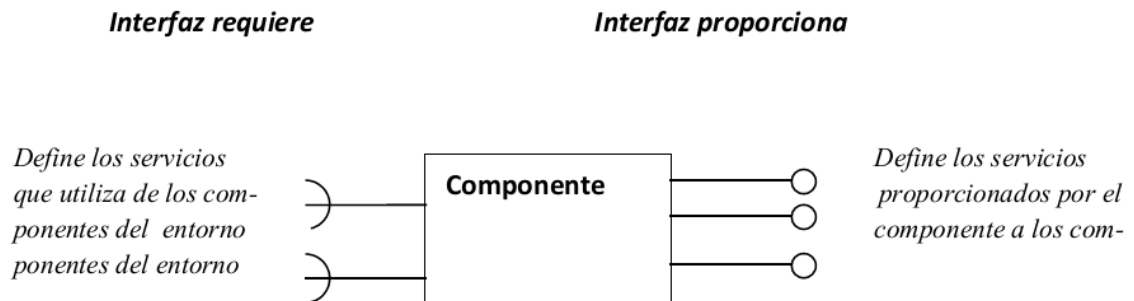
o referenciar desde su web al alojamiento originario. Pueden ser de acceso público, o pueden estar protegidos y necesitar de una autenticación previa, tienen por objetivo organizar, archivar, preservar y difundir información.⁸

- Interfaces: Son la forma en que se define un componente, comprende:

La Interfaz que proporciona, que define los servicios que proporciona el componente, lo que se conoce como el API del componente. Define los métodos que pueden ser llamados por el usuario del componente.

Interfaz que requiere: Especifica que servicios son necesarios para que el componente funcione, estos son requeridos de otros componentes del sistema.⁹

Figura 1. Interfaces de un Componente de software



Fuente: FONTÁN, Emmanuel. Metodología de desarrollo de software basado en componentes, 2010

- Eventos: Mecanismo de comunicación por el que se pueden propagar las situaciones que ocurren en un sistema de forma asíncrona. Emitidos para avisar a los componentes de su entorno de cambios en su estado.¹⁰
- Reutilización: Habilidad de un componente software de ser utilizado en contextos distintos a aquellos para los que fue diseñado.¹¹
- Contratos: Especificación que se añade a la interfaz de un componente y que establece las condiciones de uso e implementación que ligan a los

⁸ Repositorio [en línea]. [citado el 21 de Julio de 2013]. Disponible en: <http://bibliotecabiologia.usal.es/tutoriales/catalogos-repositorios-bibliosvirtuales/repositorios_digitales.html >

⁹ FONTÁN, Emmanuel. Metodología de desarrollo de software basado en componentes, 2010

¹⁰ Ibid

¹¹ Ibid

clientes y proveedores del componente. Los contratos cubren aspectos tanto funcionales (semántica de las operaciones de la interfaz) como no funcionales (calidad de servicio, prestaciones, fiabilidad o seguridad).¹²

- Widget: es un elemento de una interfaz (interfaz gráfica de usuario o GUI) que muestra información con la cual el usuario puede interactuar¹³
- IIOP: (Internet Inter-ORB Protocol) es un protocolo de CORBA (Common Object Request Broker Architecture) que hace posible que los programas distribuidos por escrito en diferentes lenguajes de programación puedan comunicarse a través de Internet.¹⁴
- IDL: (Interface Definition Language) es un término genérico para un lenguaje que permite que un programa o un objeto escrito en un lenguaje se comuniquen con otro programa escrito en un lenguaje diferente.¹⁵
- GUI: Una interfaz gráfica de usuario es un programa o entorno que gestiona la interacción con el usuario basándose en relaciones visuales como iconos, menús o un puntero.¹⁶
- IDE: Un Entorno Integrado de Desarrollo (IDE, Integrated Development Environment) es un sistema que facilita el trabajo del desarrollador de software, integrando sólidamente la edición orientada al lenguaje, la compilación o interpretación, la depuración, las medidas de rendimiento, la incorporación de los fuentes a un sistema de control de fuentes, etc., normalmente de forma modular.¹⁷
- ISO: (International Organization for Standardization) es el mayor desarrollador mundial de estándares internacionales.¹⁸
- IEC: (International Electrotechnical Commission) es la organización más importante del mundo para la preparación y publicación de normas

¹² FONTÁN, Emmanuel. Metodología de desarrollo de software basado en componentes, 2010

¹³ Widget [en línea]. [citado el 21 de Julio de 2013]. Disponible en:

<<http://www.alegsa.com.ar/Dic/widget.php> >

¹⁴ IIOP [en línea]. [citado el 21 de Julio de 2013]. Disponible en:

<<http://searchcio-midmarket.techtarget.com/definition/IIOP> >

¹⁵ IDL [en línea]. [citado el 21 de Julio de 2013]. Disponible en:

<<http://whatis.techtarget.com/definition/IDL-interface-definition-language>>

¹⁶ GUI [en línea]. [citado el 21 de Julio de 2013]. Disponible en:

<http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Interfaces/enlightment/definiciones_1.html>

¹⁷ IDE [en línea]. [citado el 21 de Julio de 2013]. Disponible en:

<<http://curso-sobre.berlios.de/introsobre/2.0.1/sobre.html/sec-ide.html> >

¹⁸ ISO [en línea]. [citado el 21 de Julio de 2013]. Disponible en: <<http://www.iso.org/iso/home.htm> >

internacionales para todas las tecnologías eléctricas, electrónicas y relacionadas.¹⁹

- Website: Se conoce como Website al conjunto de páginas web enlazadas que muestran diferentes partes del mismo. En general los sitios pueden ser de empresas, servicios, información, educación, tecnología, etc. Los sitios sobre una persona suelen llamarse Páginas personales.²⁰
- Copyleft: El software con copyleft es software libre cuyos términos de distribución garantizan que todas las copias de todas las versiones tengan aproximadamente los mismos términos de distribución. Esto significa, por ejemplo, que las licencias copyleft generalmente no permiten que terceros le agreguen requisitos adicionales al software y exigen que el código fuente esté disponible. Esto tutela el programa y sus versiones modificadas contra algunas de las formas más comunes de convertirlo en software privativo.²¹
- Copyright: Derechos de Autor, Está reconocido como uno de los derechos humanos fundamentales en la Declaración Universal de los Derechos Humanos.²²
- Api: (Application Programming Interface o Interfaz de Programación de Aplicaciones) es un conjunto de funciones que facilitan el intercambio de mensajes o datos entre dos aplicaciones. Es una forma de que dos aplicaciones que trabajan al mismo tiempo –como podría ser un procesador de texto y una hoja de cálculo– se comuniquen e intercambien datos.²³
- Applets: Un applet es una aplicación java que se ejecuta en el navegador del usuario. Para ello JRE proporciona un componente denominado plug-in que permite a los navegadores más populares ejecutar el código del applet.²⁴

¹⁹ IEC [en línea]. [citado el 21 de Julio de 2013]. Disponible en: <<http://www.iec.ch/index.htm>>

²⁰ Website [en línea]. [citado el 21 de Julio de 2013]. Disponible en:
< http://diseno.ciberaula.com/articulo/que_es_un_sitio_web/ >

²¹ Copyleft [en línea]. [citado el 21 de Julio de 2013]. Disponible en:
< <http://www.gnu.org/philosophy/categories.es.html#CopyleftedSoftware> >

²² Constitución Política de Colombia, artículo 61, 1991

²³ Api [en línea]. [citado el 21 de Julio de 2013]. Disponible en:

<<http://www.elwebmaster.com/referencia/api-interface-de-programacion-de-aplicaciones> >

²⁴ VÉLEZ SERRANO, José F. et al. Diseñar y programar, todo es empezar: Una introducción a la programación orientada a objetos usando UML y java. Madrid: DYKYNSON, 2011

- Web Service: Los servicios web son sistemas de software que permiten el intercambio de datos y funcionalidad entre aplicaciones sobre una red. Esta soportado en diferentes estándares que garantizan la interoperabilidad de los servicios. Los servicios web utilizan como su gran insumo el lenguaje extensible de marcado XML y se basa en una arquitectura en la que se define el servicio web a través de uno de los lenguajes estándar se publica en un directorio donde se halla la descripción anteriormente hecha y se utiliza de acuerdo a las expectativas de resolver una necesidad de acuerdo con la descripción provista.²⁵
- Ipc: (Inter-process Communication) es una función básica de los sistemas operativos. Los procesos pueden comunicarse entre sí a través de compartir espacios de memoria, ya sean variables compartidas o buffers, o a través de las herramientas provistas por las rutinas de IPC. La IPC provee un mecanismo que permite a los procesos comunicarse y sincronizarse entre sí, normalmente a través de un sistema de bajo nivel de paso de mensajes que ofrece la red subyacente.²⁶
- Plugin: La palabra plugin (del inglés plug-in) hace referencia a pequeñas aplicaciones funcionales creadas para ser añadidas a programas más grandes con el propósito de ampliar su capacidad en alguna actividad. El concepto de plugin implica que el objeto se "enchufa" en un programa y puede ser utilizado.²⁷

5.2 MARCO HISTÓRICO O DE ANTECEDENTES

En el año 1998 Clemens Szyperski publicó el libro “Component Software - Beyond Object-Oriented Programming”, En el cual define formalmente los conceptos relacionados con el desarrollo de software basado en componentes.²⁸

En el año 2004 Maribel Ariza Rojas y Juan Carlos Molina García en su artículo titulado: “INTRODUCCIÓN Y PRINCIPIOS BÁSICOS DEL DESARROLLO DE SOFTWARE BASADO EN COMPONENTES” realizan una investigación en la cual clasifican de los componentes a partir del tamaño, la complejidad y confiabilidad

²⁵ MORALES MACHUCA, Carlos. Andrés. (s.f.). Estado del Arte: Servicios Web, 2010

²⁶ Ipc [en línea]. [citado el 21 de Julio de 2013]. Disponible en:
< <http://carocolis.files.wordpress.com/2011/10/modos-de-transmisi3b3n.pdf> >

²⁷ Plugin [en línea]. [citado el 21 de Julio de 2013]. Disponible en:
< <http://informaticandotf.blogspot.com/2010/08/plugin-definicion.html> >

²⁸ SZYPERSKI, Clemence. Component Software. Beyond Object -Oriented Programming, 1998

que presenten.²⁹ Al año siguiente, ambos autores continuaron su investigación en su proyecto de grado titulado: “JERARQUÍA Y GRANULARIDAD DE COMPONENTES DE SOFTWARE PARA PYMES EN BOGOTÁ” en el cual hacen una profundización del tema y presentan una jerarquía de componentes: Componentes de negocio, de servicio, operacionales de persistencia, operacionales, de integración y GUI.³⁰

Más tarde, en el año 2008 Fredy Humberto Vera y Fernando Rojas publican su artículo: “Propuesta de un Proceso de Desarrollo de Componentes Software Reutilizables” mediante el cual establecieron los pasos necesarios para crear componentes software reutilizables en Java Edición Empresarial 5 (Java EE 5). También propusieron un modelo de selección de componentes software reutilizables y por último establecieron las alternativas de arquitectura que se pueden utilizar para implementar este tipo de componentes, dentro de estas arquitecturas se pueden definir: la arquitectura por capas, arquitectura modelo – vista – controlador y la arquitectura orientada a servicios.³¹

5.3 MARCO TEÓRICO

La **Programación Orientada a Objetos** es un paradigma que utiliza objetos como elementos fundamentales en la construcción de la solución. Un objeto es una abstracción de algún hecho o ente del mundo real que tiene atributos que representan sus características o propiedades y métodos que representan su comportamiento o acciones que realizan. Los conceptos más utilizados son:³²

Clase: Definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.

Objeto: Instancia de una clase. Entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos), los mismos que consecuentemente reaccionan a eventos. Se corresponden con los objetos reales

²⁹ Rojas, M. A., & García, J. C. INTRODUCCIÓN Y PRINCIPIOS BÁSICOS DEL DESARROLLO DE SOFTWARE BASADO EN COMPONENTES, 2004

³⁰ ROJAS, M. A., & GARCÍA, J. C. JERARQUÍA Y GRANULARIDAD DE COMPONENTES DE SOFTWARE PARA PYMES EN BOGOTÁ. Bogotá, 2005

³¹ Vera, Fredy Humberto & Rojas, Fernando. Propuesta de un Proceso de Desarrollo de Componentes Software Reutilizables. En: Gerencia Tecnológica Informática, 2008, vol. 7, no. 19. [en Línea]. [citado el 18 de Mayo de 2013]. Disponible en: < <http://revistas.uis.edu.co/index.php/revistagti/article/view/160/870> >

³² Aguilar, Luis Joyanes. Programación Orientada a Objetos, 1996

del mundo que nos rodea, o con objetos internos del sistema (del programa). Es una instancia a una clase.

Método: Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.

Evento: Es un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir como evento la reacción que puede desencadenar un objeto; es decir, la acción que genera.

Atributos: Características que tiene la clase.

Mensaje: Una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.

Propiedad o atributo: Contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto y esto se define como sus características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método.

Estado interno: Es una variable que se declara privada, que puede ser únicamente accedida y alterada por un método del objeto, y que se utiliza para indicar distintas situaciones posibles para el objeto (o clase de objetos). No es visible al programador que maneja una instancia de la clase.

Componentes de un objeto: Atributos, identidad, relaciones y métodos.

Identificación de un objeto: Un objeto se representa por medio de una tabla o entidad que esté compuesta por sus atributos y funciones correspondientes.

Las características más importantes de la programación orientada a objetos son las siguientes:

Abstracción: Denota las características esenciales de un objeto, donde se capturan sus comportamientos. Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el sistema sin revelar cómo se implementan estas características. Los procesos, las funciones o los métodos pueden también ser abstraídos, y, cuando lo están, una variedad de técnicas son requeridas para ampliar una abstracción. El proceso de abstracción permite seleccionar las características relevantes dentro de un conjunto e identificar comportamientos

comunes para definir nuevos tipos de entidades en el mundo real. La abstracción es clave en el proceso de análisis y diseño orientado a objetos, ya que mediante ella podemos llegar a armar un conjunto de clases que permitan modelar la realidad o el problema que se quiere atacar.

Encapsulamiento: Significa reunir todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema. Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.

Modularidad: Se denomina modularidad a la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes. Estos módulos se pueden compilar por separado, pero tienen conexiones con otros módulos. Al igual que la encapsulación, los lenguajes soportan la modularidad de diversas formas.

Principio de ocultación: Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una interfaz a otros objetos que especifica cómo pueden interactuar con los objetos de la clase. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas; solamente los propios métodos internos del objeto pueden acceder a su estado. Esto asegura que otros objetos no puedan cambiar el estado interno de un objeto de manera inesperada, eliminando efectos secundarios e interacciones inesperadas. Algunos lenguajes relajan esto, permitiendo un acceso directo a los datos internos del objeto de una manera controlada y limitando el grado de abstracción. La aplicación entera se reduce a un agregado o rompecabezas de objetos.

Polimorfismo: Comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre; al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O, dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en "tiempo de ejecución", esta última característica se llama asignación tardía o asignación dinámica. Algunos lenguajes proporcionan medios más estáticos (en "tiempo de compilación") de polimorfismo, tales como las plantillas y la sobrecarga de operadores de C++.

Herencia: Las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento, permitiendo a los objetos ser

definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que volver a implementarlo. Esto suele hacerse habitualmente agrupando los objetos en clases y estas en árboles o enrejados que reflejan un comportamiento común. Cuando un objeto hereda de más de una clase se dice que hay herencia múltiple.

Desarrollo de Software Basado en Componentes (DSBC): Hoy en día podemos decir que existe un consenso al considerar el desarrollo basado en componentes como un avance significativo hacia la construcción de sistemas mediante el ensamblado de componentes prefabricados. El DSBC trata de sentar las bases para el diseño y desarrollo de aplicaciones distribuidas basadas en componentes software reutilizables. Dicha disciplina cuenta actualmente con un creciente interés, tanto desde el punto de vista académico como desde la industria, en donde la demanda de mecanismos y herramientas de desarrollo basados en componentes es cada día mayor.

En general, el desarrollo de software basado en componentes puede verse como una extensión natural de la programación orientada a objetos dentro del ámbito de los sistemas abiertos y distribuidos. Este paradigma se basa en el uso de los componentes software como entidades básicas del modelo, entendiendo por componente “una unidad de composición de aplicaciones software que posee un conjunto de requisitos, y que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto con otros componentes, de forma independiente en tiempo y espacio”³³

Numerosas son las características que aporta la programación orientada a componentes frente a la programación orientada a objetos tradicional. Entre ellas, el desarrollo de los componentes de forma independiente del contexto en donde serán ejecutados, la reutilización por composición (frente a herencia, al tratarse de entidades “binarias” o “cajas negras”), la introspección (facilidad para interrogar al componente sobre sus propiedades y métodos de forma dinámica, normalmente mediante el uso de reflexión), nuevas formas de comunicación (como los “eventos” y las comunicaciones asíncronas frente a los rudimentarios mecanismos de los objetos), el enlazado dinámico y composición tardía, la extensión de las IDLs tradicionales, etc.³⁴

Una de las principales ventajas del desarrollo de software basado en componentes se basa en la “reutilización” de los mismos. De esta forma, los componentes se diseñan y desarrollan con el objetivo de poder ser reutilizados en otras aplicaciones, reduciendo el tiempo de desarrollo, la fiabilidad del producto final (al usar componentes ya probados previamente), y siendo más competitivos en costes. Aunque hasta ahora la reutilización suele suceder principalmente a nivel

³³ SZYPERSKI, Clemence. Component Software. Beyond Object -Oriented Programming, 1998

³⁴ Ibid

interno dentro de las organizaciones, el uso de los componentes comerciales comienza a extenderse. De esta forma se habla de componentes COTS (Commercial-Off-The-Shelf), que han sido definidos como una clase especial de componentes software, normalmente de grano grueso, que presentan las siguientes características³⁵:

- Son vendidos o licenciados al público en general.
- Los mantiene y actualiza el propio vendedor, quien conserva los derechos de la propiedad intelectual.
- Están disponibles en forma de múltiples copias, todas idénticas entre sí.
- Su código no puede ser modificado por el usuario.

³⁵ Bertoa, Manuel F., et al. Aspectos de Calidad en el Desarrollo de Software Basado en Componentes.

6. INSTRUMENTO DE MEDICIÓN PARA EVALUAR FRAMEWORKS ORIENTADOS A COMPONENTES

Para la construcción del instrumento de medición se tomaron como base las normas ISO/IEC 9126 “estándar internacional para la evaluación de la calidad del software” y ISO/IEC 14598 “estándar sobre la evaluación de productos de software” las cuales definen las características que se le pueden evaluar a un Framework.

De la norma ISO/IEC 14598 se tomaron los conceptos de las variables de medición para la evaluación de los frameworks³⁶. De la norma ISO/IEC 9126 se tomo como base la estructura de la métrica para la construcción de nuestros indicadores³⁷. La estructura del instrumento de medición se definió de la siguiente manera:

Variable: Es la característica que se pretende medir u observar dentro del framework. Se evaluarán cinco (5) variables, cada una tendrá una puntuación de veinte (20) puntos, para un total de cien (100) puntos.

Definición Operacional: Es la definición técnica valida dentro del proyecto sobre las variables a evaluar.

Indicador: Son puntos de referencia, que brindan información cualitativa o cuantitativa, conformada por uno o varios datos, constituidos por percepciones, números, hechos, opiniones o medidas.

Ítem: indica el nivel de medición del indicador.

Listado de las variables:

6.1 USABILIDAD

Definición Operacional: La capacidad del Framework de ser entendido, aprendido y atractivo para el usuario.

Cada una de las fases se dividirá en horas hombre, es decir la duración total será la suma del tiempo total dedicado por los autores.

³⁶ ISO/IEC 14598-1:1999 (E)

³⁷ ISO/IEC FDIS9126-1:2000(E)

La primera fase se basa en documentarse sobre el framework, para aprender sobre su funcionamiento, se define un tiempo máximo de 30 horas

La segunda fase consiste en realizar la instalación y configuración del framework, con un tiempo máximo de duración de 20 horas

La tercera fase consiste en buscar y ejecutar un ejemplo para el framework, para esta actividad se define un tiempo máximo de 30 horas.

Por último, después de completar el ejemplo, se plantea un ejercicio para realizar como prueba base para todos los Frameworks, el cual consiste en la construcción de una calculadora básica, es decir que pueda realizar las cuatro operaciones aritméticas básicas (suma, resta, multiplicación y división) a partir de 2 números. Para esta actividad se define un tiempo máximo de 20 horas.

El total de tiempo máximo dedicado para la evaluación de esta variable es de 100 horas, divididas entre 50 horas para cada uno de los autores del proyecto. La duración de cada fase se medirá sumando el tiempo que dedicaron ambos autores para completarlas.

Indicadores de Usabilidad:

Instalación: Cuál fue la dificultad para realizar la instalación y configuración del framework.

Ítems de Instalación:

- **Baja:** no se presentaron inconvenientes a la hora de instalar el framework, se pudo realizar sin instrucciones o búsqueda extra para ponerlo en funcionamiento y en menos de 6 horas.
- **Media:** Se presentaron algunos inconvenientes a la hora de instalar el framework, fue necesario buscar documentación extra para la instalación y/o instalar software extra para ponerlo en funcionamiento, con un tiempo de 6 a 10 horas.
- **Alta:** la instalación del framework presentó muchos inconvenientes, se debió recurrir a software extra, documentación externa y/o de diversas fuentes, y solución de problemas comunes para ponerlo en funcionamiento utilizando más de 10 horas, o No se pudo poner en funcionamiento después de utilizar el tiempo máximo.

Ejemplo: Cuál fue la dificultad para realizar un ejemplo en base a la documentación.

Ítems de Ejemplo:

- **Baja:** el ejemplo se pudo realizar siguiendo instrucciones, en menos de 6 horas.
- **Media:** Las instrucciones con las que se hizo el ejemplo no fueron suficientes, por lo que se debió recurrir a documentación extra, con un tiempo de 6 a 10 horas.
- **Alta:** Se debe invertir más de 10 horas investigando la forma de hacer funcionar el ejemplo o después de usar el máximo de horas el ejemplo no pudo ser realizado.

Actividad: Con qué dificultad se realizó la actividad planteada.

Ítems de Actividad:

- **Baja:** En Base a lo aprendido en la realización de ejemplo, los pasos a seguir para realizar la actividad no presentaron mayores dificultades, con un tiempo máximo de 6 horas de duración.
- **Media:** Después de seguir las mismas bases que en el ejemplo, se presentan dificultades técnicas en la realización de la actividad, por lo que se requiere de un estudio más profundo del framework para realizarla, utilizando de 6 a 10 horas para completarlo.
- **Alta:** Se debe invertir más de 10 horas investigando la forma de completar la actividad, o después de usar el máximo de horas no pudo ser realizada.

IDE: El framework ofrece una Interfaz Gráfica de Usuario para los desarrolladores de software.

Ítems de IDE: Si, No.

La puntuación para cada indicador se dividirá de la siguiente manera:

- Instalación siete (7) puntos:
 - Alta: 1 punto
 - Media: 3.5 puntos
 - Baja: 7 puntos
- Ejemplo seis (6) puntos:
 - Alta: 1 punto
 - Media: 3 puntos
 - Baja: 6 puntos
- Actividad cuatro (4) puntos:
 - Alta: 1 punto
 - Media: 2 puntos
 - Baja: 4 puntos
- IDE tres (3) puntos:
 - Si: 3 puntos

- No: 1 punto

6.2 SOPORTE

Definición Operacional: Grado de servicios de asistencia que proporcionan al Framework.

Indicadores de Soporte:

Cantidad de dominios en español dedicados: Por medio búsquedas avanzadas en Google se analizará la cantidad de dominios obtenidos en los primeros 100 resultados la búsqueda en español.

Ítems de Cantidad de dominios en español dedicados: 0, 1 – 5, 6 – 10, 11 – 15, 16 – 20, > 20.

Cantidad de dominios en inglés dedicados: Por medio búsquedas avanzadas en Google se analizará la cantidad de dominios obtenidos en los primeros 100 resultados la búsqueda en inglés.

Ítems de Cantidad de dominios en inglés dedicados: 0, 1 – 5, 6 – 10, 11 – 15, 16 – 20, > 20.

Tipo de Soporte: La clase soporte que ofrece el framework.

Ítems de Tipo de Soporte:

- **De pago:** Se debe realizar algún tipo de pago para tener acceso al soporte.
- **Libre:** El acceso a la documentación y soporte técnico es de libre acceso y distribución.
- **Sin Soporte:** No se le da oficialmente soporte al framework.

Comunidad dedicada: Foros de debate oficiales para el framework.

Ítems Comunidad dedicada: Si, No.

Website Oficial: Página oficial del sitio.

Ítems de Website Oficial: si, no.

La puntuación para cada indicador se dividirá de la siguiente manera:

- Cantidad de dominios en español dedicados (CDED) cinco (5) puntos:
 - 0 : 0 puntos

- 1 – 5 : 1 punto
- 6 – 10: 2 puntos
- 11 – 15: 3 puntos
- 16 - 20: 4 puntos
- Mayor a 20: 5 puntos
- Cantidad de dominios en Inglés dedicados (CDID) cinco (5) puntos:
 - 0 : 0 puntos
 - 1 – 5 : 1 punto
 - 6 – 10: 2 puntos
 - 11 – 15: 3 puntos
 - 16 - 20: 4 puntos
 - Mayor a 20: 5 puntos
- Tipo de soporte tres (3) puntos:
 - Sin Soporte: 0 puntos
 - De pago: 1.5 puntos
 - Libre: 3 puntos
- Web site oficial cuatro (4) puntos:
 - No: 0 puntos
 - Si: 4 puntos
- Comunidad dedicada tres (3) puntos:
 - No: 0 puntos
 - Si: 3 puntos

6.3 COMPATIBILIDAD

Definición Operacional: Es la capacidad del framework de ser usado con el mismo propósito en otras plataformas o ambientes de desarrollo.

Indicadores de Compatibilidad:

Sistemas operativos: Funciona en diferentes sistemas operativos.

Ítems de Sistemas operativos: Windows, GNU-Linux, Mac.

Lenguajes: Cantidad de lenguajes de programación con los que trabaja en framework.

Ítems de lenguaje: Se tomaran en cuenta veinte (20) lenguajes de programación, que corresponden a todos lenguajes que pueden ser usados por alguno de los Frameworks.

La puntuación para cada indicador se dividirá de la siguiente manera:

- Sistemas Operativos seis (6) puntos:
 - 2 Puntos por cada Sistema operativo compatible con el framework
- Lenguajes catorce (14) puntos:
 - 0.7 puntos por cada lenguaje compatible con el framework

6.4 INTEROPERABILIDAD

Definición Operacional: Es la capacidad del framework de interactuar con uno o más sistemas específicos.

Indicadores de Interoperabilidad:

Otros frameworks: Con qué otros frameworks pueden operar los componentes.

Ítems de Otros frameworks: .Net, AFC, Bonobo, COM, Corba, EJB, GCF, KDE, UNO y XPCOM.

La puntuación para cada indicador se dividirá de la siguiente manera:

- Otros Frameworks veinte (20) puntos:
 - 2 puntos por cada framework con el que pueda comunicarse el framework evaluado.

6.5 ACCESIBILIDAD

Definición Operacional: Es el grado en el que los usuarios pueden acceder y usar el framework

Indicadores de Accesibilidad:

Licencia: Tipo de licencia con la que trabaja el framework.

Ítems de licencia: Copyright, Copyleft.

Requisitos de software: Requisitos de software adicional.

Ítems de Requisitos de software: Los que indique cada framework.

La puntuación para cada indicador se dividirá de la siguiente manera:

- Licencia doce (12) puntos:

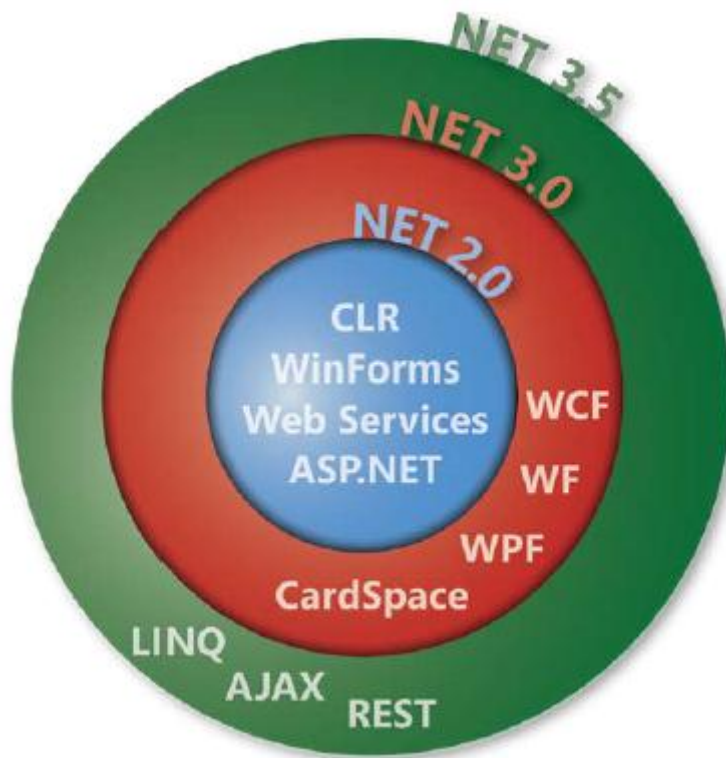
- Copyright 6 puntos
- Copyleft 12 puntos
- Requisitos del software 8 puntos
 - Más de 4 requisitos 1 punto
 - 3 requisitos 4 puntos
 - 2 requisitos 6 puntos
 - 1 requisito 8 puntos

7. .NET

7.1 HISTORIA

La primera versión de .NET Framework fue liberada el 13 de febrero de 2002, a partir de este punto Microsoft se ha encargado de ir lanzando nuevas versiones y mejoras con el paso de los años, actualmente se encuentra en la versión 4.0 y se mantiene vigente.³⁸

Figura 2. .Net



Additive versions of the .NET Framework

Fuente: <http://blogs.msdn.com/b/warnov/archive/2010/07/04/breve-historia-del-framework-net.aspx>

³⁸ Historia de .Net [en línea]. [citado el 15 de Junio de 2013] Disponible en :
<<http://blogs.msdn.com/b/warnov/archive/2010/07/04/breve-historia-del-framework-net.aspx> >

7.2 CARACTERÍSTICAS TÉCNICAS

.NET Framework es un entorno de ejecución administrado que proporciona diversos servicios a las aplicaciones en ejecución. Consta de dos componentes principales: CLR, que es el motor de ejecución que controla las aplicaciones en ejecución; y la biblioteca de clases de .NET Framework, que proporciona una biblioteca de código probado y reutilizable al que pueden llamar los desarrolladores desde sus propias aplicaciones.³⁹

El CLR (Common Language Runtime) es un entorno de ejecución que se encarga de administrar procesos, memoria, tipos de variables y demás de las aplicaciones de .Net, facilitando el trabajo del desarrollador ya que lo invita a no preocuparse por estas cosas.

Junto al CLR se encuentra complementándolo la Biblioteca de Clases de .NET, que proporciona una colección de tipos facilitando a CLR la creación de código intermedio y mejorando la interoperabilidad entre lenguajes.⁴⁰

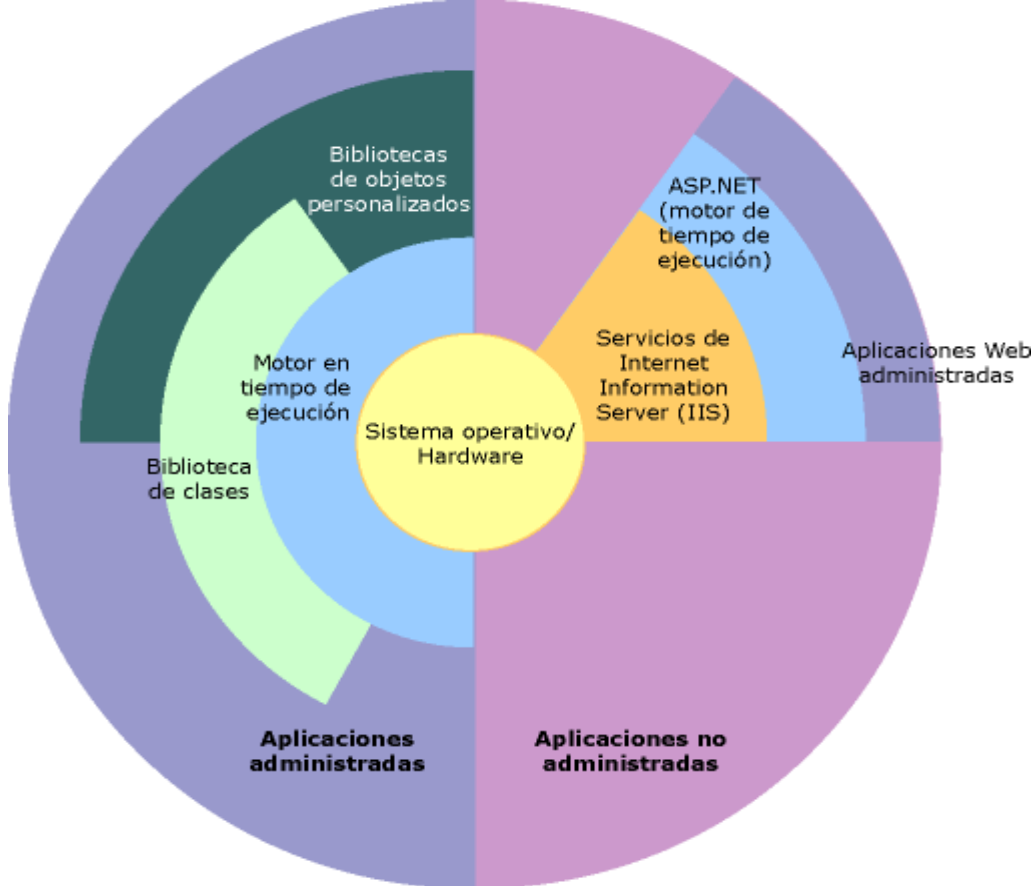
Con respecto a la seguridad, los componentes administrados reciben grados de confianza diferentes, en función de una serie de factores entre los que se incluye su origen (como Internet, red empresarial o equipo local). Esto significa que un componente administrado puede ser capaz o no de realizar operaciones de acceso a archivos, operaciones de acceso al Registro y otras funciones delicadas, incluso si se está utilizando en la misma aplicación activa.⁴¹

³⁹ Introducción a .NET Framework [en línea]. [citado el 15 de Junio de 2013]. Disponible en: < <http://msdn.microsoft.com/es-es/library/vstudio/hh425099> >

⁴⁰ Biblioteca de clases .Net [en línea]. [citado el 15 de Junio de 2013]. Disponible en: < <http://msdn.microsoft.com/es-es/library/vstudio/zw4w595w> >

⁴¹ Introducción a .NET Framework [en línea]. [citado el 15 de Junio de 2013]. Disponible en: < <http://msdn.microsoft.com/es-es/library/vstudio/hh425099> >

Figura 3. .Net framework en contexto



Fuente: <http://msdn.microsoft.com/es-es/library/vstudio/zw4w595w>

Los tipos de .NET Framework permiten realizar tareas de programación comunes, como son la conectividad de bases de datos, acceso a archivos, etc., además incluye tipos útiles para distintas necesidades de desarrollo especializados, por ejemplo, se puede utilizar .NET Framework para desarrollar los siguientes tipos de aplicaciones y servicios⁴²:

- Aplicaciones de consola.
- Aplicaciones GUI de Windows (Windows Forms).
- Aplicaciones de Windows Presentation Foundation (WPF).
- Aplicaciones de ADO.NET (ActiveX Data Objects) para el acceso a los datos.
- Aplicaciones de ASP.NET (Active Server Pages) para aplicaciones web.
- Aplicaciones orientadas a servicios utilizando Windows Communication Foundation (WCF).

⁴² Biblioteca de clases .Net [en línea]. [citado el 15 de Junio de 2013]. Disponible en: < <http://msdn.microsoft.com/es-es/library/vstudio/zw4w595w> >

- Aplicaciones habilitadas para el flujo de trabajo utilizando Windows Workflow Foundation (WF).
- Servicios de Windows.

7.3 USABILIDAD

Para la documentación sobre .NET se tomó un tiempo de 20 horas en total, leyendo sobre sus características y funcionalidades, de diferentes fuentes y autores, pero tomando como base principal la página oficial de dicho framework.

Las pruebas de usabilidad de este framework se realizaron en el sistema operativo Windows 7, para ello fue necesario instalar:

- Visual Studio 2010.

Lo cual tomó 5 horas en total por la descarga e instalación.

Se realizó un ejemplo de la página oficial:

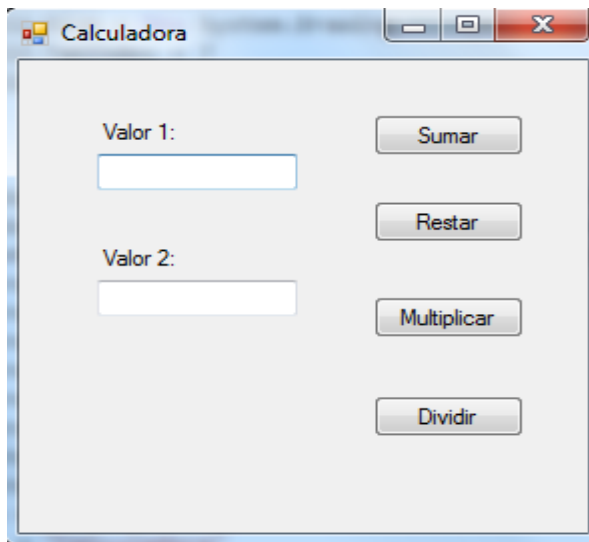
<http://msdn.microsoft.com/es-es/library/151w6x12%28v=vs.80%29.aspx>

Para completar este ejemplo se debieron realizar algunas modificaciones adicionales, ya que la documentación no fue suficientemente explícita, esto tomó una duración de 7 horas

Después de comprobar el funcionamiento de .Net, se procedió a realizar el ejercicio práctico planteado para evaluar la usabilidad desarrollando el componente de la calculadora básica, la cual se pudo realizar correctamente en un tiempo de 4 horas. Los resultados concretos son los siguientes:

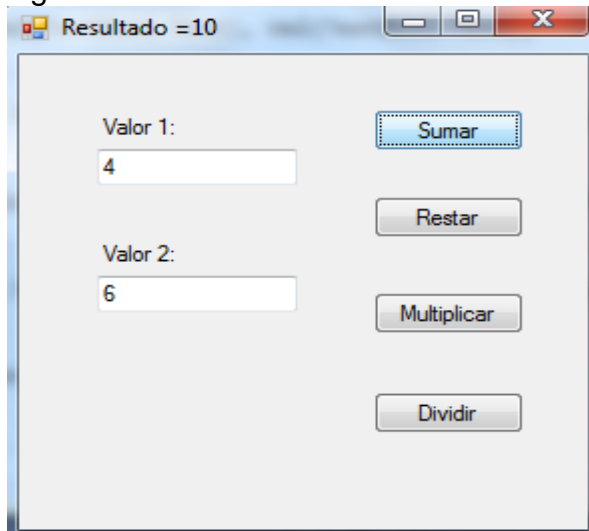
- Instalación: Baja.
- Ejemplo: Media.
- Actividad: Baja.
- IDE: Si.

Figura 4. Calculadora



Fuente: Los autores.

Figura 5. Calculadora sumando



Fuente: Los autores.

7.4 SOPORTE

Los resultados de la evaluación de las pruebas de soporte de .NET fueron los siguientes:

- Cantidad de dominios en español dedicados: se encuentra en el rango de 6 -10, se pueden encontrar en el anexo **A. Pruebas de Soporte.**
- Cantidad de dominios en ingles dedicados: Se encuentra en el rango 16 - 20, se pueden encontrar en el anexo **A. Pruebas de Soporte.**
- Tipo de Soporte: De pago.
- Comunidad dedicada: Si (<http://social.msdn.microsoft.com/Forums/vstudio/es-es/home>).
- Website Oficial: Si (<http://www.microsoft.com/net/>).

7.5 COMPATIBILIDAD

- Sistemas Operativos: Windows.
- Lenguajes: 10 - C#, Visual Basic .NET, Delphi (Object Pascal), C++, F#, J#, Perl, Python, Fortran, Prolog.

7.6 INTEROPERABILIDAD

Otros frameworks: COM y XpCom.

7.7 ACCESIBILIDAD

- Licencia: Privativo, comercial.
- Requisitos de software: Visual Studio.

8. AFC – APPLICATION FOUNDATION CLASSES

8.1 HISTORIA

El 2 de abril de 1997 Microsoft anunció las clases de Microsoft Application Foundation Classes (AFC) para la creación de interfaces de usuario en Java, para el 07 de octubre del mismo año se lanzó el SDK (software development kit) 2.0 para Java, que incluía el AFC de Microsoft.⁴³

8.2 CARACTERÍSTICAS TÉCNICAS

AFC es el acrónimo de Application Foundation Classes, es un rico conjunto de gráficos, interfaz de usuario y capacidades multimedia, escrita completamente en Java, que se acumulan en la parte superior de la biblioteca básica de Java AWT para proporcionar un potente conjunto de bloques de construcción para los desarrolladores de aplicaciones de Internet y Java.

Hay dos versiones de AFC, una que trabaja con la máquina virtual Java de Microsoft (IE y JDK 1.1) y la otra que trabaja con JDK 1.02, la versión que optimizada para Microsoft debe brindar un mejor uso de los recursos.⁴⁴

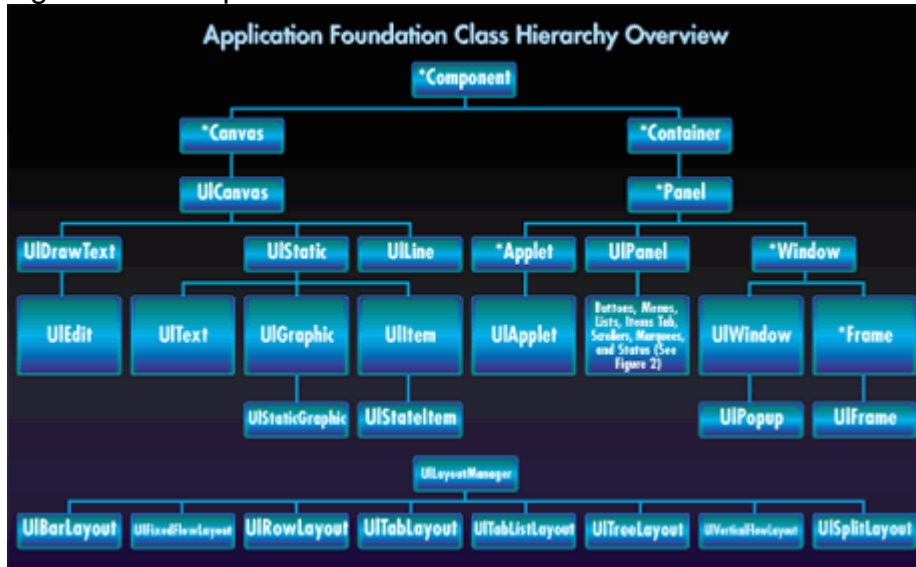
Application Foundation Classes se proporciona en tres paquetes:

- `com.ms.ui`: El paquete de interfaz de usuario, que proporciona el marco básico para applets, aplicaciones y controles de interfaz de usuario.
- `com.ms.fx`: El paquete FX proporciona controles gráficos avanzados en sus aplicaciones como el control sobre los colores, las fuentes, curvas, texto y texturas.
- `com.ms.resource`: El paquete de recursos proporciona acceso a Win-32 recursos, incluidos los cuadros de diálogo y tablas de cadenas, de las aplicaciones Java.

⁴³ Historia de AFC [en línea]. [citado el 15 de Junio de 2013]. Disponible en: <http://www.extropia.com/tutorials/java/java_afc_jfc.html >

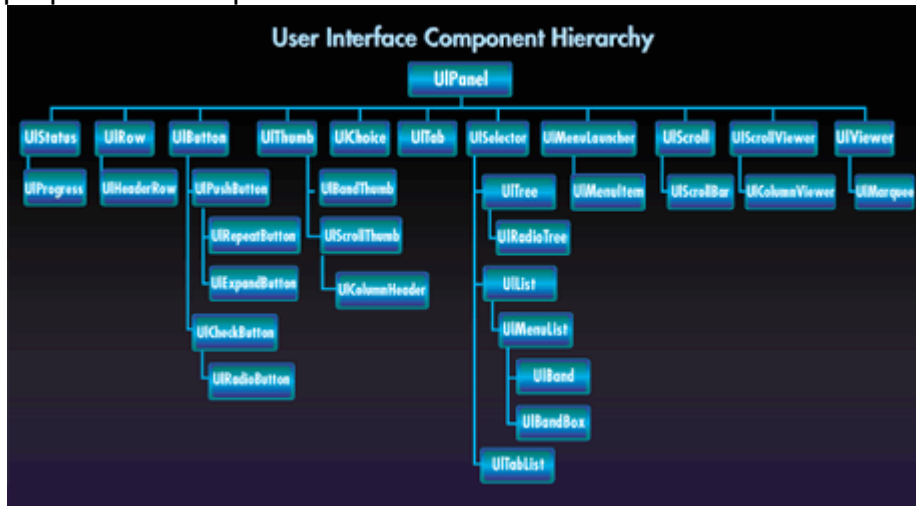
⁴⁴ APPLICATION FOUNDATION CLASSES [en línea]. [citado el 15 de Junio de 2013]. Disponible en: <<http://www2.sys-con.com/itsg/virtualcd/java/archives/0207/fronkowiakp14/index.html> >

Figura 6. Jerarquía de AFC



Fuente: <http://www2.sys-con.com/itsg/virtualcd/java/archives/0207/fronkowiakp14/index.html>

Figura 7. Jerarquía de clases de los elementos de la interfaz de usuario proporcionadas por las AFC.



Fuente: <http://www2.sys-con.com/itsg/virtualcd/java/archives/0207/fronkowiakp14/index.html>

8.3 USABILIDAD

Para la documentación sobre AFC se tomó un tiempo de 30 horas en total, buscando información sobre sus características y funcionalidades, de diferentes fuentes y autores, pero no fue posible encontrar mucha información.

Las pruebas de usabilidad de este framework se realizaron en el sistema operativo Windows 7, para ello fue necesario instalar:

- Java SDK 2.0
- Visual J++

Luego, se intento realizar una prueba básica para verificar el funcionamiento de AFC, con base en el tutorial del siguiente enlace:

<http://www2.sys-con.com/itsg/virtualcd/java/archives/0207/fronkowiakp14/index.html>

Después de muchos intentos, no se pudo completar el ejemplo anterior dado que la documentación al respecto que se encuentra en internet sobre este framework es muy limitada, se utilizó el tiempo máximo de duración para configurar el framework y hacer funcionar el ejemplo, pero no fue posible completar esta actividad, por tal motivo tampoco fue posible realizar el ejercicio práctico planteado. Los resultados concretos son los siguientes:

- Instalación: Alta.
- Ejemplo: Alta.
- Actividad: Alta
- IDE: Si.

8.4 SOPORTE

Los resultados de la evaluación de las pruebas de soporte de AFC fueron los siguientes:

- Cantidad de dominios en español dedicados: se encuentra en el rango de 0, se pueden encontrar en el anexo **A. Pruebas de Soporte.**
- Cantidad de dominios en ingles dedicados: Se encuentra en el rango de 1-5, se pueden encontrar en el anexo **A. Pruebas de Soporte.**
- Tipo de Soporte: Sin Soporte.

- Comunidad dedicada: No.
- Website Oficial: No.

8.5 COMPATIBILIDAD

- Sistemas Operativos: Windows.
- Lenguajes: 1 - Java.

8.6 INTEROPERABILIDAD

Otros frameworks: Solo con sí mismo.

8.7 ACCESIBILIDAD

- Licencia: Privativo, comercial.
- Requisitos de software: Java SDK 2.0, Visual J++.

9. BONOBO

9.1 HISTORIA

*Bonobo es un modelo de componente para la creación de componentes de software reusable y composición de documentos. Fue creado por la compañía Ximian (adquirida por Novell).*⁴⁵

9.2 CARACTERÍSTICAS TÉCNICAS

Bonobo es un "framework" para crear componentes e interfaces para las aplicaciones de de GNOME. Esta construido sobre el estándar de la industria Common Object Request Broker Architecture (CORBA) , el que Bonobo esté construido sobre la CORBA le proporciona muchas ventajas como que sus componentes se puedan ejecutar en procesos diferentes o que se pueda escribir en diferentes lenguajes necesitando solo interfaces IDL para comunicarse.⁴⁶

Todos los componentes de Bonobo heredan de la interfaz Unknown, pues es la que permite controlar y reconocer los componentes en tiempo real.

Los tipos de componentes de Bonobo son los siguientes:

- Componentes no visuales: estos serían el equivalente a servidores CORBA normales, que pueden ser usados para definir servicios fácilmente accesibles desde otras aplicaciones.
- Controles: son el equivalente a los ActiveX de Microsoft. Son simples controles (widgets de pantalla) que están implementados en procesos independientes, y que pueden ser insertados dentro de otros controles en otras aplicaciones.
- Componentes "empotrables": éstos son componentes que pueden ser insertados (empotrados) en contenedores Bonobo, y que al mismo tiempo se incrustan dentro de la ventana contenedora permitiendo la edición 'inline'.

⁴⁵ BONOBO AND CORBA [en línea]. [citado el 15 de Junio de 2013]. Disponible en: <<https://developer.gnome.org/platform-overview/2.32/bonobo-corba.html.en> >

⁴⁶ La arquitectura de componentes Bonobo: Una introducción práctica [en línea]. [citado el 15 de Junio de 2013]. Disponible en: < <http://acsblog.es/articulos/trunk/ProgramacionActual/bonobo/html/c25.html> >

- Contenedores: estos son los componentes visuales encargados de embeber los componentes "empotrables". Además, permiten la implementación de 'monikers', que es un sistema mediante el cual otra aplicación puede acceder a un elemento específico dentro de un componente embebido en un contenedor.

9.3 USABILIDAD

Para el Framework BONOBO no se realizaron las pruebas de forma habitual, ya que este ha sido oficialmente desaprobado por GNOME desde su versión 2.4 y la documentación es demasiada escasa.

9.4 SOPORTE

Los resultados de la evaluación de las pruebas de soporte de BONOBO fueron los siguientes:

- Cantidad de dominios en español dedicados: se encuentra en el rango de 6- 10, se pueden encontrar en el anexo **A. Pruebas de Soporte**.
- Cantidad de dominios en inglés dedicados: Se encuentra en el rango 1 - 5, se pueden encontrar en el anexo **A. Pruebas de Soporte**.
- Tipo de Soporte: Libre.
- Comunidad dedicada: No.
- Website Oficial: No.

9.5 COMPATIBILIDAD

- Sistemas Operativos: Linux.
- Lenguajes: 7 - C, C + +, Perl, Python, Java, Cobol y Lisp.

9.6 INTEROPERABILIDAD

Otros frameworks: Con aquellos que interpreten interfaces IDL. Corba, EJB, KDE, XpCom.

9.7 ACCESIBILIDAD

- Licencia: GPL (Licencia Pública General).
- Requisitos de software: Compilador del lenguaje a trabajar.

10. COM – COMPONENT OBJECT MODEL

10.1 HISTORIA

El Component Object Model (COM), es una tecnología de Microsoft para desarrollar componentes de software que se puedan comunicar entre ellos sin importar el lenguaje en que son creados.⁴⁷

Entre los personajes más destacados en la creación de la arquitectura COM, se encuentra Anthony Williams, quien promovió el concepto de componentes de software en sus Libros de “Object Architecture: Dealing With the Unknown”, “Type Safety in a Dynamically Extensible Class” de 1988 y “On Inheritance: What It Means and How To Use It” de 1990.⁴⁸

Inicialmente COM fue creado como un modelo de objetos subyacente de OLE (object linking and embedding), juntos hacían frente a los componentes de software en general, pero poco a poco las tecnologías de OLE fueron renombradas para COM.

10.2 CARACTERÍSTICAS TÉCNICAS DE COM (COM, DCOM, COM+ Y ACTIVE X)

La familia de tecnologías COM incluye COM +, COM distribuido (DCOM) y controles ActiveX. Tecnologías que son usadas principalmente en la familia de aplicaciones Microsoft como los es office. Algunas de las características de los componentes COM es que pueden ser creado en diversos lenguajes, pueden estar dentro de un mismo proceso o diferentes, he incluso en equipos remotos.⁴⁹

Además Microsoft proporciona interfaces COM para muchas interfaces de programación de aplicaciones de Windows, como DirectShow, Fundación de Medios, API Packaging, Windows Animation Manager, dispositivos portátiles de Windows y Microsoft Active Directory (AD).

DCOM en español Modelo de Objetos de Componentes Distribuidos, es una tecnología propietaria de Microsoft para desarrollar componentes software distribuidos sobre varios ordenadores y que se comunican entre sí.

⁴⁷ COM: Component Object Model Technologies [en línea]. [citado el 15 de Junio de 2013]. Disponible en: <<http://www.microsoft.com/com/default.msp>>

⁴⁸ Historia COM [en línea]. [citado el 15 de Junio de 2013]. Disponible en: <<http://www.thefreelibrary.com/Component+Object+Model-a+history.-a0163334583>>

⁴⁹ Características técnicas de COM [en línea]. [citado el 15 de Junio de 2013]. Disponible en: <<http://msdn.microsoft.com/es-es/library>>

COM + es una evolución del modelo de objetos componentes de Microsoft (COM) y Microsoft Transaction Server (MTS)⁵⁰. Quitándole peso de encima al desarrollador ya que se encarga de varias de las tareas que este antes debía programar por sí mismo. Como la seguridad o la administración de hilos.

Características que ofrece COM+:

- Controles de acceso de nivel de aplicación habilitada de forma predeterminada.
- Application Pooling
- Reciclaje de aplicaciones
- Particiones COM +
- Servicios sin componentes COM +
- Servicio (Simple Object Access Protocol) SOAP COM +
- Niveles de aislamiento configurables
- Creación de componentes privados
- Configuración de seguridad de DTC (Distributed Transaction Coordinator)
- Activación de compuerta con Baja Memoria
- Mover y copiar componentes COM
- Acceso a la red
- Pausar y deshabilitar aplicaciones
- Proceso de vertimiento
- Ejecución de aplicaciones COM + con Servicios de NT
- Ensamblaje por ambos lados
- Reporte de errores de Windows

ActiveX es un entorno para definir componentes de software reusables de forma independiente del lenguaje de programación. Las aplicaciones de software pueden ser diseñadas por uno o más de esos componentes para así proveer su correspondiente funcionalidad.

10.3 USABILIDAD

Para el Framework COM no se realizaron las pruebas de forma habitual, a pesar de que COM sigue siendo una tecnología viable y que Microsoft no ha dado indicios de suspenderla, este framework tiende a ser reemplazado por la apuesta Microsoft .Net, por lo que la evaluación de usabilidad no se llevó a cabo.

⁵⁰ Características técnicas de COM [en línea]. [citado el 15 de Junio de 2013]. Disponible en: < <http://msdn.microsoft.com/es-es/library> >

10.4 SOPORTE

Los resultados de la evaluación de las pruebas de soporte de COM fueron los siguientes:

- Cantidad de dominios en español dedicados: se encuentra en el rango de 11 -15, se pueden encontrar en el anexo **A. Pruebas de Soporte.**
- Cantidad de dominios en inglés dedicados: Se encuentra en el rango 16 - 20, se pueden encontrar en el anexo **A. Pruebas de Soporte.**
- Tipo de Soporte: De pago.
- Comunidad dedicada: Si (<http://msdn.microsoft.com/en-us/library/windows/desktop/ms694363%28v=vs.85%29.aspx>).
- Website Oficial: Si (<http://www.microsoft.com/com/default.aspx>).

10.5 COMPATIBILIDAD

- Sistemas Operativos: Windows.
- Lenguajes: 3 - C, C++, Visual Basic, y varios de los lenguajes de script implementados en la plataforma Windows.

10.6 INTEROPERABILIDAD

Otros frameworks: .Net y GCF.

10.7 ACCESIBILIDAD

- Licencia: Privativo, comercial.
- Requisitos de software: Compilador MIDL y del lenguaje en el que se vaya a trabajar.

11. CORBA – COMMON OBJECT REQUEST BROKER ARCHITECTURE

11.1 HISTORIA

En el año 1991 fue lanzada la primera versión de CORBA, que contiene el modelo de objetos, el IDL (Lenguaje de especificación de interfaces), y el conjunto básico de interfaces de programación de aplicaciones (API). En el año 1996 salió la versión 2.0 en la cual se incluye principalmente la arquitectura de interoperabilidad (GIOP, IIOp, DCE CIOP), por último, en el 2002 salió la versión actual 3.0.⁵¹

11.2 CARACTERÍSTICAS TÉCNICAS.

CORBA (Common Object Request Broker Architecture) usa el protocolo IIOp estándar, el cual permite a sus aplicaciones comunicarse entre sí en cualquier ordenador, sistema operativo, lenguaje de programación y red.

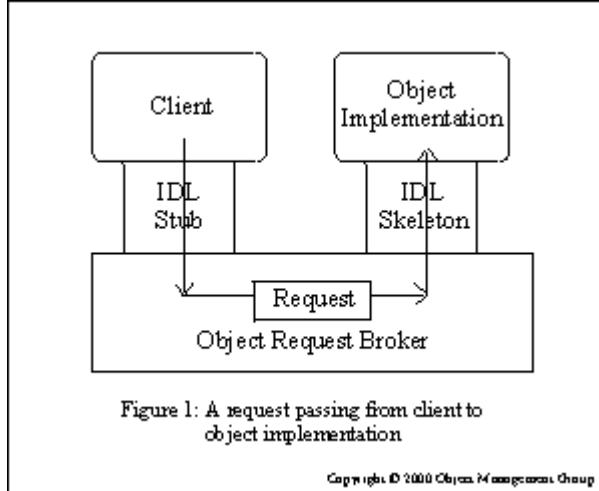
Las aplicaciones desarrolladas en CORBA se comunican por medio de interfaces IDL, la definición de la interfaz IDL no es independiente del lenguaje de programación, sino que se asigna a todos los lenguajes de programación a través de los estándares OMG: OMG tiene asignaciones estandarizadas de IDL a C, C + +, Java, COBOL, Smalltalk, Ada, Lisp, Python, y IDLscript.⁵²

En la figura 8 se puede observar el proceso que se lleva a cabo cuando un cliente hace una petición a un objeto, el cliente realiza la petición por medio de la idl, la cual es procesada por el ORB (Agente de petición de objetos) y se comunica al objeto por medio de su propio idl.

⁵¹ Historia de CORBA [en línea]. [citado el 15 de Junio de 2013]. Disponible en: < http://www.omg.org/gettingstarted/history_of_corba.htm >

⁵² CORBA® BASICS [en línea]. [citado el 15 de Junio de 2013]. Disponible en: < <http://www.omg.org/gettingstarted/corbafaq.htm> >

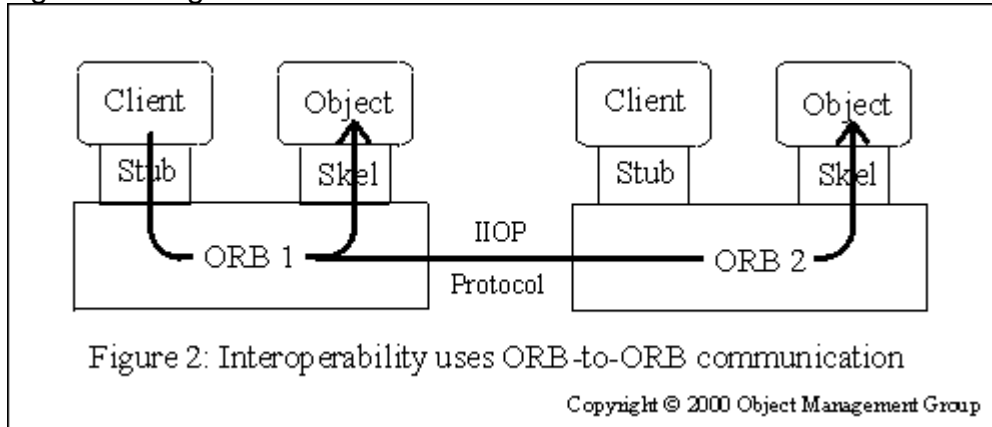
Figura 8. Petición de un Cliente a un Objeto



Fuente: <http://www.omg.org/gettingstarted/corbafaq.htm>

En la figura 9 se encuentra definido el proceso que se lleva a cabo al realizarse una invocación remota, en la cual los ORB se comunican entre sí por medio del protocolo IIOP para peticiones a objetos por fuera de la máquina local. Para mayor información acerca del ORB visite el siguiente enlace: http://www.omg.org/gettingstarted/orb_basics.htm

Figura 9. Diagrama de una invocación remota



Fuente: <http://www.omg.org/gettingstarted/corbafaq.htm>

Las características más destacadas de la declaración de componentes son las siguientes:

- Especifica el nombre del componente.
- Puede especificar una lista de interfaces que el componente soporta.
- Soporta herencia simple de otras definiciones de componentes.

11.3 USABILIDAD

Para la documentación sobre Corba se tomó un tiempo de 22 horas en total, leyendo sobre sus características y funcionalidades, de diferentes fuentes y autores, pero tomando como base principal la página oficial de dicho framework.

Las pruebas de usabilidad de este framework se realizaron en el sistema operativo Windows 7 y Ubuntu 12.04, para ello fue necesario instalar:

- Java SE Development Kit 7u25.

Lo cual tomó 4 horas por la descarga e instalación, pero ocurrieron algunos problemas con la configuración que se solucionaron al cabo de otras 4 horas investigando cuál era el inconveniente, para un tiempo total de 8 horas.

Se realizó el siguiente ejemplo usando CORBA y java:

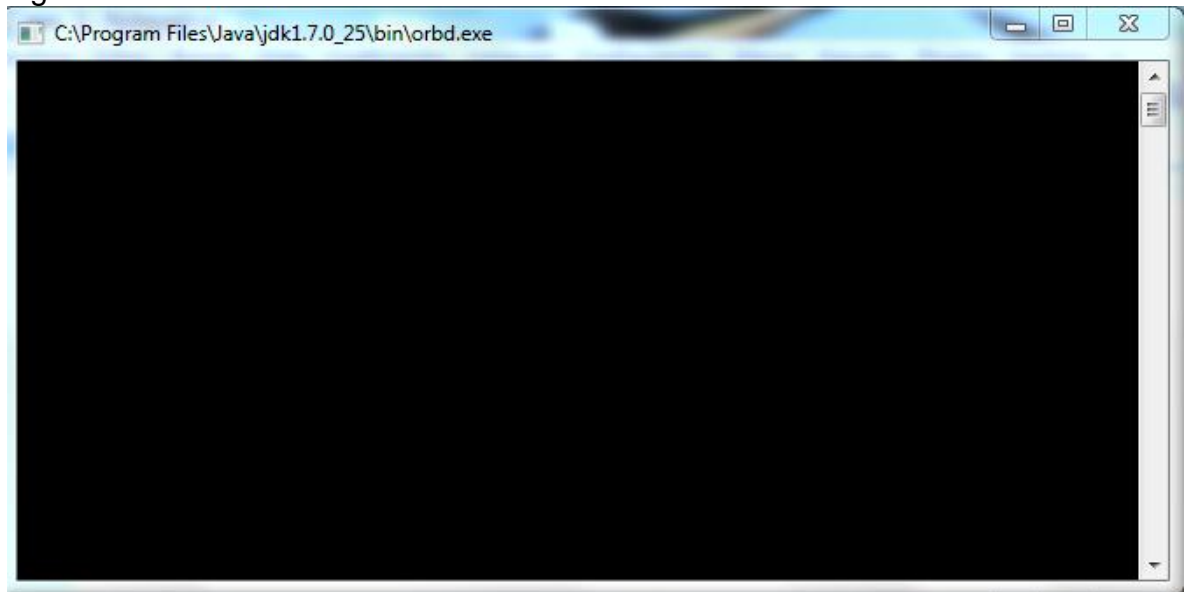
<http://ivaras.blogspot.com/2009/02/usando-corba-y-netbeans.html>

El cual tomó un tiempo de 6 horas para completarlo, ya que no se explicaba detalladamente y fue necesario buscar información adicional para compilarlo y ejecutarlo.

Después de comprobar el funcionamiento de CORBA, se procedió a realizar el ejercicio práctico planteado para evaluar la usabilidad desarrollando el componente de la calculadora básica, la cual se pudo realizar correctamente con un tiempo de 7 horas. Para este caso el cliente le pregunta al servidor sobre la operación suma entre los números 3 y 4, los resultados concretos son los siguientes:

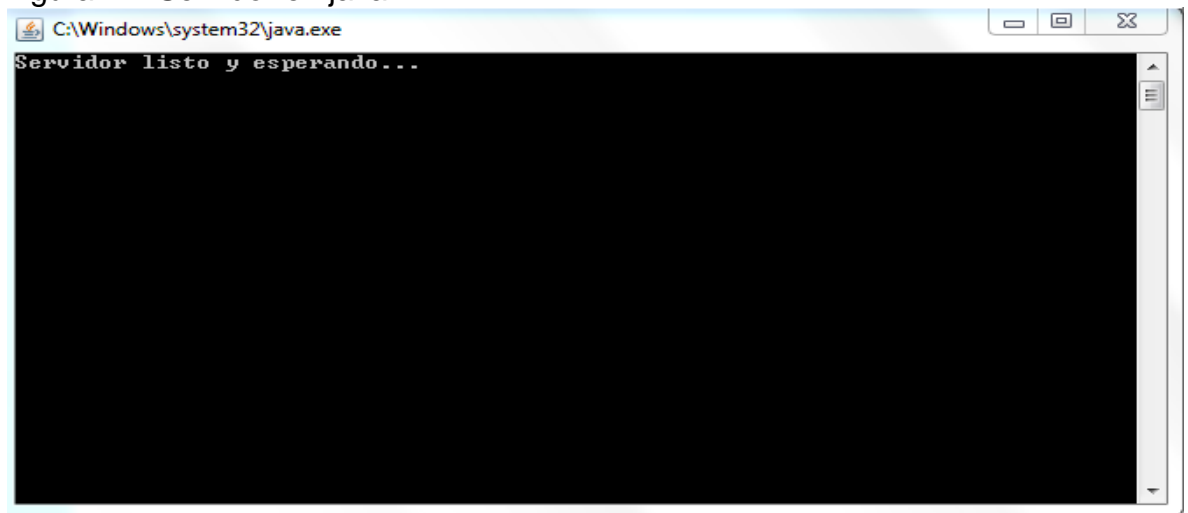
- Instalación: Media.
- Ejemplo: Media
- Actividad: Media.
- IDE: Si.

Figura 10. ORBD de Corba



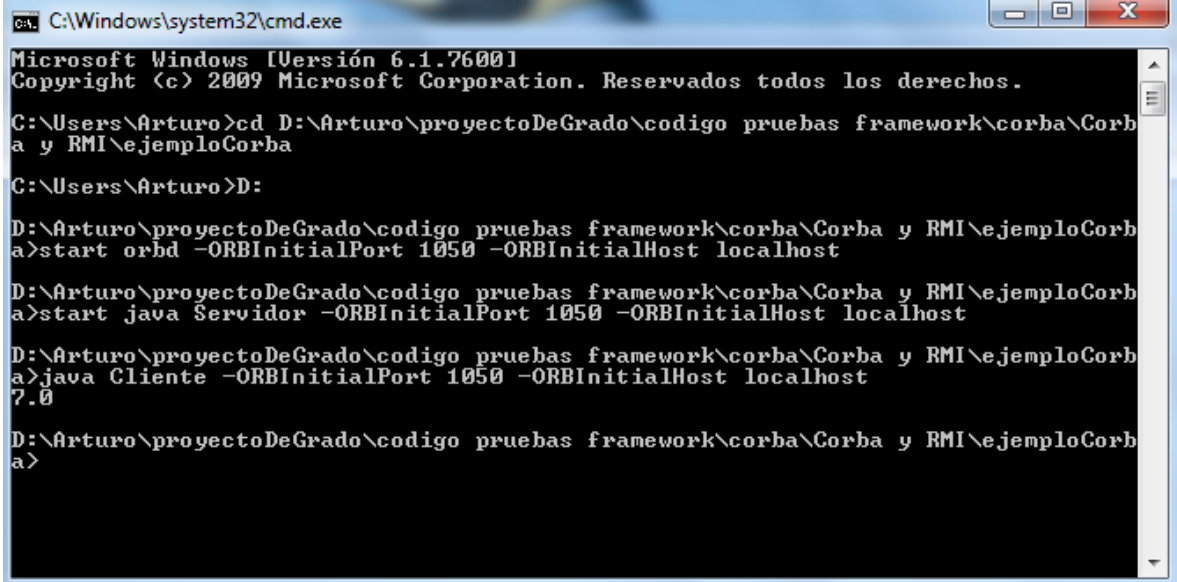
Fuente: Los autores.

Figura 11. Servidor en java



Fuente: Los autores.

Figura 12. Consola donde se invoca el cliente en java



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Arturo>cd D:\Arturo\proyectoDeGrado\codigo pruebas framework\corba\Corba y RMI\ejemploCorba
C:\Users\Arturo>D:
D:\Arturo\proyectoDeGrado\codigo pruebas framework\corba\Corba y RMI\ejemploCorba>start orbd -ORBInitialPort 1050 -ORBInitialHost localhost
D:\Arturo\proyectoDeGrado\codigo pruebas framework\corba\Corba y RMI\ejemploCorba>start java Servidor -ORBInitialPort 1050 -ORBInitialHost localhost
D:\Arturo\proyectoDeGrado\codigo pruebas framework\corba\Corba y RMI\ejemploCorba>java Cliente -ORBInitialPort 1050 -ORBInitialHost localhost
7.0
D:\Arturo\proyectoDeGrado\codigo pruebas framework\corba\Corba y RMI\ejemploCorba>
```

Fuente: Los autores.

11.4 SOPORTE

Los resultados de la evaluación de las pruebas de soporte de EJB fueron las siguientes:

- Cantidad de dominios en español dedicados: se encuentra en el rango de 6-10, se pueden encontrar en el anexo **A. Pruebas de Soporte**.
- Cantidad de dominios en ingles dedicados: Se encuentra en el rango 11 - 15, se pueden encontrar en el anexo **A. Pruebas de Soporte**.
- Tipo de Soporte: De pago.
- Comunidad dedicada: Si (http://www.omg.org/welcome_to_the_members_area.htm).
- Website Oficial: Si (<http://www.corba.org/>).

11.5 COMPATIBILIDAD

- Sistemas Operativos: Linux, Windows y Mac.
- Lenguajes: 9 - C, C + +, Java, COBOL, Smalltalk, Ada, Lisp, Python, y IDLscript.

11.6 INTEROPERABILIDAD

Otros frameworks: Compatible con frameworks que utilicen el protocolo IIOP de Corba. Bonobo, COM, EJB, XpCom.

11.7 ACCESIBILIDAD

- Licencia: GPL v2 (Licencia Pública General de GNU).
- Requisitos de software: Se necesita tener instalado el JDK, se puede correr, compilar y ejecutar mediante la consola de comandos.

12. EJB - ENTERPRISE JAVA BEANS

12.1 HISTORIA

La especificación EJB ha ido evolucionando a la par que lo hacía la propia especificación J2EE. Las diferentes versiones que han existido hasta la fecha son.⁵³

- *EJB 1.0: la especificación original*
- *EJB 1.1: la primera incluida dentro de J2EE*
- *EJB 2.0: incluida en J2EE 1.3, añadía las interfaces Locales.*
- *EJB 2.1: incluida en la última revisión de J2EE, la 1.4.*
- *EJB 3.0: Ahora con Clúster y está incluida en JEE 5.1*
- *EJB 3.1: incluida en lo nuevo de JavaEE 6 en diciembre de 2009*

12.2 CARACTERÍSTICAS TÉCNICAS

EJB es una plataforma para construir aplicaciones de negocio portables, reusables y escalables usando el lenguaje de programación Java. Desde el punto de vista del desarrollador, un EJB es una porción de código que se ejecuta en un contenedor EJB, no es más que un ambiente especializado (runtime) que provee determinados componentes de servicio.⁵⁴

*Cada EJB debe facilitar una clase de implementación Java y dos interfaces Java. El contenedor EJB creará instancias de la clase de implementación Java para facilitar la implementación EJB. Las interfaces Java son utilizadas por el código cliente del EJB. El cliente invoca un método en los proxy generados que a su vez sitúa los argumentos método en un mensaje y envía dicho mensaje al servidor EJB. Los proxy usan **RMI-IIOP** para comunicarse con el servidor EJB. El servidor llamará a un método correspondiente a una instancia de la clase de implementación Java para manejar la llamada del método remoto.⁵⁵*

⁵³ Historia EJB [en línea]. [citado el 15 de Junio de 2013]. Disponible en:
< <https://code.google.com/p/fap-devel/wiki/EJB3> >

⁵⁴ Introducción a los JavaBeans [en línea]. [citado el 15 de Junio de 2013]. Disponible en:
< <http://www.sc.ehu.es/sbweb/fisica/cursoJava/applets/javaBeans/fundamento.htm> >

⁵⁵ Ibid

Tipos de EJB's:

- EJB de Entidad (Entity EJBs): Se encarga de encapsular los objetos del lado del servidor que almacena los datos, presentan la característica fundamental de la persistencia:
 - Persistencia gestionada por el contenedor (CMP), es decir, por el servicio.
 - Persistencia gestionada por el bean (BMP), es decir, por el componente.
- EJB de Sesión (Session EJBs): gestionan el flujo de la información en el servidor. Generalmente sirven a los clientes como una fachada de los servicios proporcionados por otros componentes disponibles en el servidor. Puede haber dos tipos:
 - Con estado (statefull). se limita el acceso a un solo cliente.
 - Sin estado (stateless). puede ser accedido por varios clientes.
- EJB dirigidos por mensajes (MDB o Message driven Beans): Tienen funcionamiento asíncrono.

Servicios:

- Integración: permite acoplar en tiempo ejecución distintos componentes, por medio de configuración en XML.
- Pooling: El contenedor de EJBs crea instancias de los componentes para ser utilizados por los clientes con el fin de optimizar la memoria, se aplica a los EJBs de sesión y los dirigidos por mensaje (MDB).
- Thread-safely: El programador puede escribir componentes del lado del servidor como si estuviera trabajando en una aplicación sencilla con un solo thread (hilo).
- Administración de Estados: El contenedor maneja y almacena los estados de un Bean.
- Mensajería: Con ayuda de los Beans dirigidos por mensaje se puede lograr la comunicación de dos componentes de forma asíncrona.
- Transacciones: El contenedor se encarga de gestionar las transacciones llamadas por los componentes.
- Seguridad: soporta integración con la Java Authentication and Authorization Service (JAAS) API, para todos los Bean de sesión.
- Interceptores: añade un pequeño framework para el manejo de la programación orientada a aspectos, además el desarrollador puede agregar sus propios aspectos.
- Acceso Remoto: Permite acceder remotamente a los Beans de sesión

- Web Services: Un bean de sesión sin estado puede publicar sus métodos como servicios web.
- Persistencia: EJB 3 provee la especificación JPA para el mapeo de objetos (Entities) a tablas.
- Catching and Performance: manejo fácil y persistente de cache proporcionado por JPA.

12.3 USABILIDAD

Para la documentación sobre EJB se tomó un tiempo de 16 horas en total, leyendo sobre sus características y funcionalidades, de diferentes fuentes y autores, pero tomando como base principal la página oficial de dicho framework.

Las pruebas de usabilidad de este framework se realizaron en el sistema operativo Windows 7, para ello fue necesario instalar:

- Java SE Development Kit 7u25.
- NetBeans IDE 7.3.1, este trae integrado el JEE con el respectivo EJB y también incluye el servidor Glassfish.

Lo cual tomó 4 horas en total.

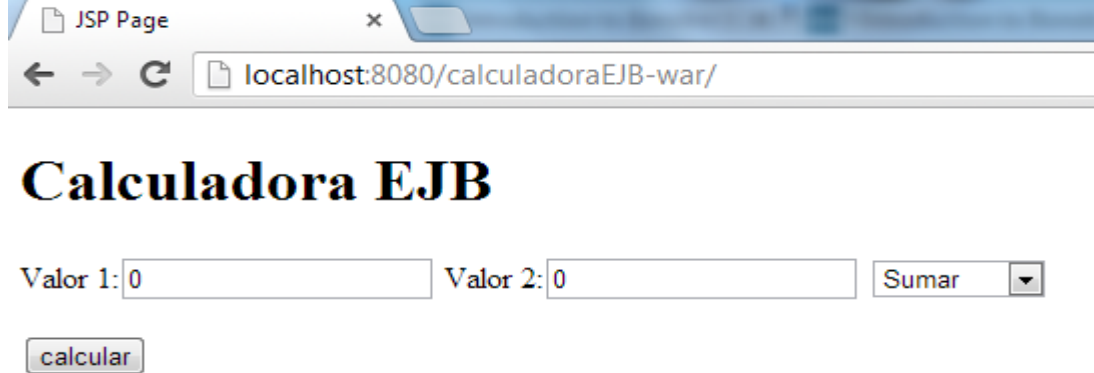
Luego, se realizó una prueba básica para verificar el funcionamiento de EJB, para ello se ejecutó unos de los ejemplos de muestra que trae el netbeans:

Se selecciono la opción File-> New Project y en la sección **categories** se eligió la opción **samples** y luego en **Java EE** se eligió el proyecto **Lottery Annotation (Java EE 5)**. Con este ejemplo se pudo comprobar satisfactoriamente la instalación y el funcionamiento del framework Enterprise Java Beans, con un tiempo de duración de 3 horas.

Después de comprobar el funcionamiento de EJB, se procedió a realizar el ejercicio práctico planteado para evaluar la usabilidad desarrollando el componente de la calculadora básica, la cual se pudo realizar correctamente en un tiempo de 8 horas, ya que fue necesario investigar acerca de la comunicación interna de los componentes EJB. Los resultados concretos son los siguientes:

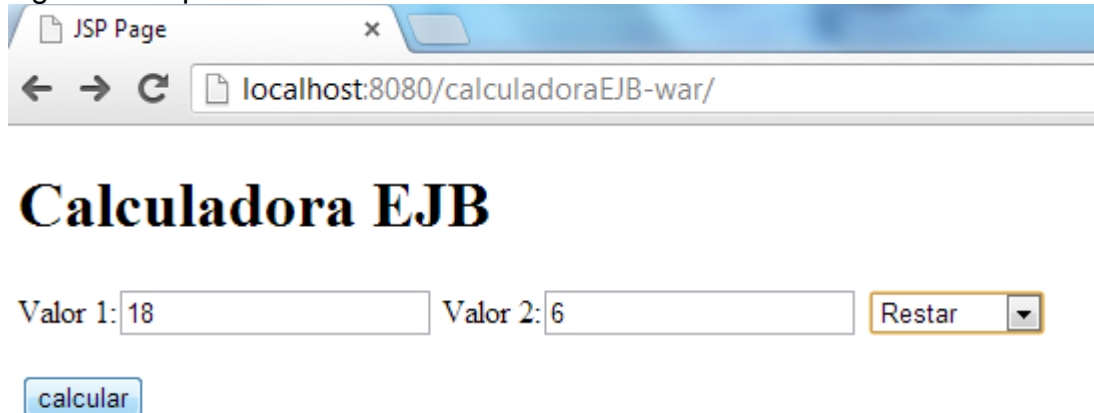
- Instalación: Baja.
- Ejemplo: Baja.
- Actividad: Media.
- IDE: Si.

Figura 13. Aplicación Web de una calculadora en EJB



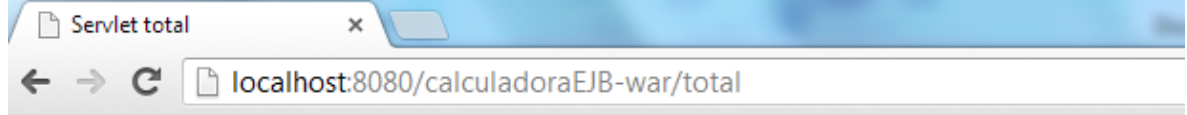
Fuente: Los autores.

Figura 14. Operación "Restar"



Fuente: Los autores.

Figura 15. Resultado de la resta



Operación: Restar

Resultado: 12

Fuente: Los autores.

12.4 SOPORTE

Los resultados de la evaluación de las pruebas de soporte de EJB fueron los siguientes:

- Cantidad de dominios en español dedicados: se encuentra en el rango de 16-20, se pueden encontrar en el anexo **A. Pruebas de Soporte**.
- Cantidad de dominios en inglés dedicados: Se encuentra en el rango >20, se pueden encontrar en el anexo **A. Pruebas de Soporte**.
- Tipo de Soporte: De pago.
- Comunidad dedicada: Si
(https://forums.oracle.com/community/developer/english/java/java_enterprise_%26_remote_computing - <https://community.java.net/community/java-enterprise>).
- Website Oficial: Si
(<http://www.oracle.com/technetwork/java/javaee/ejb/index.html>).

12.5 COMPATIBILIDAD

- Sistemas Operativos: Linux, Windows y Mac.
- Lenguajes: 1. Solo Java.

12.6 INTEROPERABILIDAD

Otros frameworks: Bonobo, Corba, y XpCom.

12.7 ACCESIBILIDAD

- Licencia: CDDL v1.0 (Licencia de distribución y Desarrollo Común) y GPL v2 (Licencia Pública General de GNU).
- Requisitos de software: Se necesita tener instalado el JDK y un IDE para aplicaciones de java, como por ejemplo eclipse o netbeans, adicionalmente debe tener instalado el JEE para que pueda utilizar EJB, y también el servidor Glassfish.

13. GCF - GENERIC COMPONENT FRAMEWORK

13.1 HISTORIA

GCF comenzó como un módulo para VTK Designer 2. El objetivo principal en ese momento era proporcionar una arquitectura de aplicaciones para VTK Designer 2 que permite ofrecer aplicaciones, nuevas funciones y correcciones de errores de copias desplegados del software.⁵⁶

13.2 CARACTERÍSTICAS TÉCNICAS

GCF es una estructura de componentes a la medida usando Qt 4.4.x. para el desarrollo de aplicaciones como un conjunto de componentes para minimizar el mantenimiento y la actualización de interfaz de usuario.⁵⁷

GCF ofrece soluciones para el desarrollo basado en componentes, que ayuda a:

- Carga (UI y no UI) componentes de bibliotecas compartidas
- GUI Fusión y configuración de componentes
- Descubrimiento Componente / Objeto
- La comunicación y las dependencias entre componentes
- IPC - La comunicación entre los objetos a través de las aplicaciones y el sistema de límites
- integración con ActiveX y KParts

*GCF ofrece una arquitectura de aplicación preparada.*⁵⁸

- *Para gestionar la carga, inicialización y descarga de componentes.*
- *Para configurar componentes utilizando archivos XML.*
- *Fusionar GUI ofrecidos por diferentes componentes en un solo widget de contenedor de aplicaciones.*
- *Tener componentes descubrir fácilmente la funcionalidad ofrecida por otros componentes.*

⁵⁶ Historia GCF [en línea]. [citado el 15 de Junio de 2013]. Disponible en:
< <http://www.vcreatelogic.com/products/gcf/highlights> >

⁵⁷ Generic Component Framework (GCF) [en línea]. [citado el 15 de Junio de 2013]. Disponible en:
< <http://www.vcreatelogic.com/docs/gcf-2.5.0/html> >

⁵⁸ Arquitectura GCF [en línea]. [citado el 15 de Junio de 2013]. Disponible en:
< <http://www.prashanthudupa.com/?s=gcf> >

- *Para descargar e instalar las actualizaciones de componentes a partir de uno o más servidores de actualización automática,*
- *Para comunicarse con los componentes de otras aplicaciones dentro y fuera de la computadora host (IPC),*
- *Para crear conexiones de señal / slot entre los objetos que residen en diferentes procesos (IPC)*
- *Cargar KParts y componentes ActiveX con facilidad*
- *Para cargar los componentes del GCF en Application Server para el acceso remoto de HTML web-applications que se ejecutan en un navegador.*

El modelo de componentes GCF se basa en los siguientes conceptos clave:⁵⁹

- Componente GCF
- Aplicación GCF donde se cargan los componentes
- ApplicationXML, archivo XML que contiene los parámetros de configuración clave para los módulos de IPC y la actualización de software.
- GUIXML, archivo XML donde van todos los widgets y como van unidos.
- Objetos, aparatos, menús, barras de herramientas y acciones que figuran en un archivo GUIXML son llamados objetos "exposed".

13.3 USABILIDAD

La documentación sobre GCF se tomó 24 horas en total, leyendo sobre sus características y funcionalidades, de diferentes fuentes y autores, pero tomando como base principal la página oficial de dicho framework.

Las pruebas de usabilidad de este framework se realizaron en el sistema operativo Windows 7, para ello es necesario instalar:

- Qt >= 4.7.4C
- Qt-Creator con MinGW o Visual Studio >= 2008

Lo cual no fue posible, ya que la extensión para crear componentes GCF no se pudo instalar en el Qt-Creator ni en Visual Studio en el tiempo máximo de duración para esta actividad.

Debido a esto, no se pudo realizar la prueba básica para verificar el funcionamiento de GCF, con base en el siguiente tutorial:

⁵⁹ Generic Component Framework (GCF) [en línea]. [citado el 15 de Junio de 2013]. Disponible en: < <http://www.vcreatellogic.com/docs/gcf-2.5.0/html> >

<http://www.vcreatelogic.com/docs/gcf-2.6.0/html/gettingstarted.html>

Dado que la documentación al respecto que se encuentra en internet sobre este framework es muy limitada, por tal motivo tampoco fue posible realizar el ejercicio práctico planteado para evaluar la usabilidad.

Los resultados concretos son los siguientes:

- Instalación: Alta.
- Ejemplo: Alta.
- Actividad: Alta.
- IDE: Si.

13.4 SOPORTE

Los resultados de la evaluación de las pruebas de soporte de kparts fueron las siguientes:

- Cantidad de dominios en español dedicados: se encuentra en el rango de 0, se pueden encontrar en el anexo **A. Pruebas de Soporte**.
- Cantidad de dominios en ingles dedicados: Se encuentra en el rango de 11-15, se pueden encontrar en el anexo **A. Pruebas de Soporte**.
- Tipo de Soporte: Libre o De Pago.
- Comunidad dedicada: Si (<http://www.vcreatelogic.com/forum> - <http://blog.vcreatelogic.com/>).
- Website Oficial: Si (<http://www.vcreatelogic.com/products/gcf>).

13.5 COMPATIBILIDAD

- Sistemas Operativos: Linux, Windows y Mac.
- Lenguajes: 1. Solo C++.

13.6 INTEROPERABILIDAD

Otros frameworks: con componentes KDE y Active X mediante servidores. COM.

13.7 ACCESIBILIDAD

- Licencia: GPL y Privativa Comercial.
- Requisitos de software: Qt \geq 4.7.4C, Qt-Creator con MinGW o Visual Studio \geq 2008.

14. KDE

14.1 HISTORIA

KDE fue fundado el 14 de octubre del año 1996. Y actualmente se encuentra en su versión 4.3.5.⁶⁰

14.2 CARACTERÍSTICAS TÉCNICAS

KDE es una herramienta de código abierto, para entorno gráfico de Linux/Unix basado en un conjunto de herramienta Qt para aplicaciones gráficas. KParts es un marco para componentes KDE que utiliza el modelo de QObjects programados en el lenguaje c++.⁶¹

QObject proporciona la funcionalidad común a todos los objetos: la relación de nombres, padre / hijo, señales y slots, modelo de eventos y propiedades. Se define un conjunto muy simple de clases: Part, Plug-in, MainWindow y Part Manager.⁶²

- Un **Part** es el nombre de un componente gráfico KDE.
- Un **plug-in** es una pieza no grafica, pero que proporciona funcionalidad a la interfaz.
- Una **MainWindow** es una ventana de KDE que permite agregar soporte para plugins, y da acceso al título de la ventana y la barra de estado.
- Un **Part Manager** es un objeto que se encarga de gestionar los **Parts**.

Para el manejo de las interfaces, kparts utiliza un framework llamado XmlGui, el cual tiene como objetivo definir interfaces gráficas de usuario mediante ficheros con estructura XML.⁶³

⁶⁰ Historia KDE [en línea]. [citado el 15 de Junio de 2013] Disponible en:
< <http://www.kde.org/community/history> >

⁶¹ Características técnicas kde [en línea]. [citado el 15 de Junio de 2013] Disponible en:
< <http://www.ibm.com/developerworks/library/l-kparts> >

⁶² Ibid

⁶³ Ibid

14.3 USABILIDAD

Para la documentación sobre KDE se tomó un tiempo de 25 horas en total, leyendo sobre sus características y funcionalidades, leyendo de diferentes fuentes y autores, pero tomando como base principal la página oficial de dicho framework.

Las pruebas de usabilidad de este framework se realizaron en el sistema operativo Ubuntu 12.04, para ello fue necesario instalar:

- kdelibs-devel >= 4.3
- Qt-devel >= 4.5.2
- CMake >= 2.6.2
- g++ >= 4.0

Lo cual tomó un tiempo de 20 horas debió a que además de los requisitos anteriores era necesario instalar software adicional básico para el correcto funcionamiento de los requerimientos básicos.

Luego, se realizó una prueba básica para verificar el funcionamiento de KDE, con base en el siguiente tutorial:

- http://techbase.kde.org/Development/Tutorials/First_program.

Completar este ejemplo tomó 10 horas debido a los inconvenientes con los complementos en el momento de compilar y ejecutar la aplicación.

Luego de comprobar el funcionamiento de KDE, se procedió a investigar más sobre su funcionamiento, y se encontraron los siguientes tutoriales que ilustran de una manera un poco más avanzada el procedimiento para crear y utilizar componentes:

- <http://kde-qt.blogspot.com/>
- http://techbase.kde.org/Development/Tutorials/Using_Qt_Creator
- http://techbase.kde.org/Development/Tutorials/Using_Qt_Designer

Con base en el ejemplo realizado y los demás tutoriales se procedió a realizar la actividad planteada, pero esta no pudo completarse en el máximo de tiempo destinado.

Los resultados concretos son los siguientes:

- Instalación: Alta.
- Ejemplo: Media.
- Actividad: Alta.

- IDE: Si.

14.4 SOPORTE

Los resultados de la evaluación de las pruebas de soporte de kparts fueron las siguientes:

- Cantidad de dominios en español dedicados: se encuentra en el rango de 1-5, se pueden encontrar en el anexo **A. Pruebas de Soporte**
- Cantidad de dominios en ingles dedicados: Se encuentra en el rango de 6-10, se pueden encontrar en el anexo **A. Pruebas de Soporte**
- Tipo de Soporte: Libre
- Comunidad dedicada: Si (<http://forum.kde.org/viewforum.php?f=218>)
- Website Oficial: Si (<http://techbase.kde.org/Development/Architecture/KDE4/KParts>)

14.5 COMPATIBILIDAD

- Sistemas Operativos: Linux, Windows y Mac
- Lenguajes: 1. Solo C++

14.6 INTEROPERABILIDAD

Otros frameworks: solo con otros que usen DCOP (Protocolo de Comunicación de Escritorio) mediante XParts. Bonobo y GCF

14.7 ACCESIBILIDAD

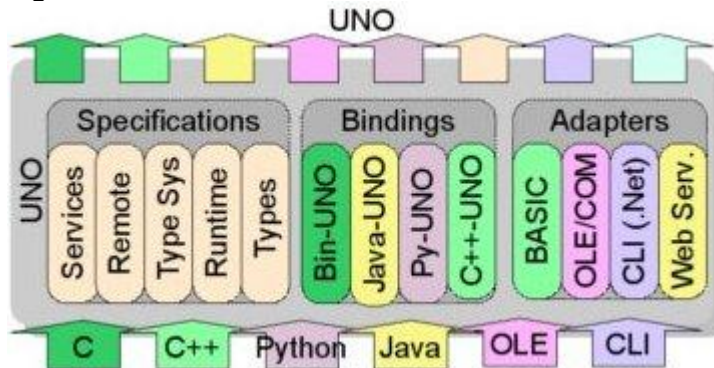
- Licencia: LGPL 2.1 (Licencia Pública General Reducida de GNU)
- Requisitos de software: kdelibs-devel \geq 4.3, Qt-devel \geq 4.5.2, Boost MultiIndex library \geq 1.35, g++ \geq 4.0 (o compatible) Para Linux, msvc \geq 8 Para sistemas Windows, GNU make (Para Linux), CMake \geq 2.6.2, KDevelop Development Platform \geq 1.0.0, Grantlee \geq 0.1.7

15. UNO - UNIVERSAL NETWORKING OBJECTS

15.1 CARACTERÍSTICAS TÉCNICAS

Universal Networking Objects (Uno) es el modelo de componentes de OpenOffice.org.⁶⁴

Figura 16. UNO



Fuente: <http://www.openoffice.org/udk/>

Las interfaces de UNO (UNO-interfaces) estas especificadas en IDL y sus objetos se controlan mediante referencias globales. UNO garantiza la identidad del objeto, la identidad hilo, y la secuencia de las llamadas:

- La identidad del objeto: Referencias Dos interfaces que se pueden comparar por la igualdad.
- Identidad del subproceso: En UNO cada hilo es nombrado por un identificador de hilo único a nivel global.
- secuencia de llamadas: UNO permite declarar un método unidireccional (o asíncrono).

*Una secuencia de llamadas de sentido único se puede transferir y ejecutar muy rápidamente a través de una conexión entre procesos. El protocolo entre procesos UNO está optimizado para conexiones con poco ancho de banda.*⁶⁵

Los objetos UNO se definen por servicios. Un servicio puede incluir varios servicios e interfaces. Una interfaz define un único aspecto de un servicio y por lo tanto es más pequeño en su alcance.

⁶⁴ Universal Network Objects (UNO) [en línea]. [citado el 15 de Junio de 2013] Disponible en: <http://www.openoffice.org/udk/common/man/uno.html>

⁶⁵ Ibid

15.2 USABILIDAD

Para la documentación sobre UNO se tomó un tiempo de 14 horas en total, leyendo sobre sus características y funcionalidades, leyendo de diferentes fuentes y autores, pero tomando como base principal la página oficial de dicho framework.

Las pruebas de usabilidad de este framework se realizaron en el sistema operativo Windows 7, para ello fue necesario instalar:

- Apache OpenOffice - Software Development Kit (SDK)

Lo cual tomó un tiempo de 4 horas, para la descarga e instalación del framework.

Luego, se realizó una prueba básica para verificar el funcionamiento de UNO, con base en el siguiente tutorial:

- http://wiki.openoffice.org/wiki/Tutorial_UNO_Library

Después de realizar múltiples intentos se utilizó el tiempo máximo para la instalación mas el tiempo del ejemplo buscando una configuración para hacerlo, pero no fue posible completarlo, dado que la documentación al respecto que se encuentra en internet sobre este framework es muy limitada, por tal motivo tampoco fue posible realizar el ejercicio práctico planteado para evaluar la usabilidad.

Los resultados concretos son los siguientes:

- Instalación: Alta.
- Ejemplo: Alta.
- Actividad: Alta.
- IDE: Si.

15.3 SOPORTE

Los resultados de la evaluación de las pruebas de soporte de UNO fueron las siguientes:

- Cantidad de dominios en español dedicados: se encuentra en el rango de 1-5, se pueden encontrar en el anexo **A. Pruebas de Soporte.**
- Cantidad de dominios en inglés dedicados: Se encuentra en el rango de 1-5, se pueden encontrar en el anexo **A. Pruebas de Soporte.**

- Tipo de Soporte: Libre.
- Comunidad dedicada: Si
(<http://forum.openoffice.org/es/forum/viewforum.php?f=50&sid=41497ae7637d4140ea274d7b3aa0ab30>).
- Website Oficial: Si (<http://www.openoffice.org/udk/common/man/uno.html>).

15.4 COMPATIBILIDAD

- Sistemas Operativos: Linux, Windows y Mac.
- Lenguajes: 3 - C, C++, java.

15.5 INTEROPERABILIDAD

Otros frameworks: Sólo con sí mismo.

15.6 ACCESIBILIDAD

- Licencia: GPL (Licencia Pública General).
- Requisitos de software: Apache OpenOffice - Software Development Kit (SDK).

16. XPCOM - CROSS PLATFORM COMPONENT OBJECT MODEL

16.1 HISTORIA

*XPCOM es una plataforma que permite a los desarrolladores partir los proyectos de software monolíticos en piezas modulares más pequeñas. Estas piezas, conocidas como componentes son ensamblados juntas nuevamente en tiempo de ejecución.*⁶⁶

16.2 CARACTERÍSTICAS TÉCNICAS

XPCOM suministra las herramientas necesarias para la carga y manipulación de componentes, y ayudan a desarrollador a escribir componentes multiplataforma. Usando XPCOM, los desarrolladores crean componentes que pueden ser reutilizados en diferentes aplicaciones o pueden ser reemplazados para cambiar la funcionalidad de aplicaciones existentes.⁶⁷

*XPCOM no solamente soporta el desarrollo de componentes de software, también provee gran parte de la funcionalidad de una plataforma de desarrollo, como.*⁶⁸

- *gestión de componentes*
- *abstracción de archivos*
- *paso de mensajes objeto*
- *manejo de memoria*

16.3 USABILIDAD

Para la documentación sobre XpCom se tomó un tiempo de 11 horas en total, leyendo sobre sus características y funcionalidades, leyendo de diferentes fuentes y autores, pero tomando como base principal la página oficial de dicho framework.

Las pruebas de usabilidad de este framework se realizaron en el sistema operativo Windows 7, para ello fue necesario instalar:

⁶⁶ XPCOM [en línea]. [citado el 15 de Junio de 2013] Disponible en:
< <https://developer.mozilla.org/es/docs/XPCOM> >

⁶⁷ Ibid

⁶⁸ Ibid

- Mozilla Firefox => 3.0
- Gecko SDK => 1.9.0

El proceso de instalación y configuración del framework tomó 11 horas, que incluyen principalmente la investigación sobre cómo integrar los componentes al explorador Mozilla Firefox.

Luego, se realizó una prueba básica para verificar el funcionamiento de XPCOM, con base en el siguiente tutorial:

- https://developer.mozilla.org/es/docs/C%C3%B3mo_crear_un_componente_XPCOM_en_JavaScript

Se utilizó el tiempo máximo para el ejemplo práctico, pero la prueba no se pudo realizar con éxito, por lo cual tampoco fue posible llevar a cabo la actividad planteada, pero se pudo comprobar su funcionamiento mediante ejemplos alojados en el siguiente link:

- <https://developer.mozilla.org/es/demos/?menu>

Los resultados concretos son los siguientes:

- Instalación: Alta.
- Ejemplo: Alta.
- Actividad: Alta.
- IDE: Si.

16.4 SOPORTE

Los resultados de la evaluación de las pruebas de soporte de kparts fueron las siguientes:

- Cantidad de dominios en español dedicados: se encuentra en el rango de 1-5, se pueden encontrar en el anexo **A. Pruebas de Soporte**.
- Cantidad de dominios en inglés dedicados: Se encuentra en el rango de 6-10, se pueden encontrar en el anexo **A. Pruebas de Soporte**.
- Tipo de Soporte: Libre.
- Comunidad dedicada: Si
(<https://forums.mozilla.org/addons/viewforum.php?f=13>).
- Website Oficial: Si (<https://developer.mozilla.org/es/docs/XPCOM>).

16.5 COMPATIBILIDAD

- Sistemas Operativos: Linux, Windows y Mac.
- Lenguajes: 7 - C, JavaScript, C++, Python, Perl, CLI, Ruby.

16.6 INTEROPERABILIDAD

Otros frameworks: con cualquier aplicación a través de mozilla. .Net, Bonobo, COM, Corba y EJB.

16.7 ACCESIBILIDAD.

- Licencia: LGPL 2.1 (Licencia Pública General Reducida de GNU).
- Requisitos de software: Mozilla Firefox => 3.0, Gecko SDK => 1.9.0.

17. SÍNTESIS DE LOS RESULTADOS

Con el resultado de las evaluaciones de cada framework se sintetizan los resultados para una mejor comparación. Cada tabla representa una variable, en la cual se enfrenta cada framework contra los indicadores de dicha variable. Para cada variable se resaltarán los 3 frameworks mejor evaluados.

17.1 USABILIDAD

En la tabla 1 se observan los resultados de cada framework con respecto a los indicadores de usabilidad. En la primera columna se posicionan los frameworks y en la segunda fila los indicadores a evaluar.

Las celdas con N.A. (No Aplica) fueron los frameworks en los cuales no se evaluó el respectivo ítem.

Las celdas del tiempo que aparecen Sin resultado indican que el ejercicio propuesto a realizar no se pudo completar, a causa de una escasa documentación y/o una curva de aprendizaje muy alta.

Tabla 1. Usabilidad

Framework	Indicadores			
	Instalación	Ejemplo	Actividad	IDE
.Net	Baja	Media	Baja	Si
AFC	Alta	Alta	Alta	Si
Bonobo	N.A	N.A	N.A	No
COM	N.A	N.A	N.A	Si
Corba	Media	Media	Media	Si
EJB	Baja	Baja	Media	Si
GCF	Alta	Alta	Alta	Si
KDE	Alta	Alta	Alta	Si
UNO	Alta	Alta	Alta	Si
XPCOM	Alta	Alta	Alta	Si

Fuente: Los autores.

Tabla 2. Puntuación usabilidad

Frameworks	Indicadores de usabilidad				
	INSTALACIÓN	EJEMPLO	ACTIVIDAD	IDE	TOTAL
.NET	7	3	4	3	17
AFC	1	1	1	3	6
BONOBO	N.A	N.A	N.A	1	1
COM	N.A	N.A	N.A	3	3
CORBA	3,5	3	2	3	11,5
EJB	7	6	2	3	18
GCF	1	1	1	3	6
KDE	1	3	1	3	8
UNO	1	1	1	3	6
XPCOM	1	1	1	3	6

Fuente: Los autores.

Los frameworks mejor evaluados para esta variable son:

- EJB
- .Net
- Corba

17.2 SOPORTE

En la tabla 3 se observan los resultados de cada framework con respecto a los indicadores de soporte. En la primera columna se posicionan los frameworks y en la segunda fila los indicadores a evaluar.

El término CDED resume Cantidad de Dominios en Español Dedicados.
 El término CDID resume Cantidad de Dominios en Ingles Dedicados.

Tabla 3. Soporte

Framework	indicadores				
	CDED	CDID	Tipo Soporte	Comunidad Dedicada	Web Site Oficial
.Net	6-10	16-20	de Pago	Si	Si
AFC	0	1-5	Sin Soporte	No	No
Bonobo	6-10	1-5	Libre	No	No
COM	11-15	16-20	de Pago	Si	Si
Corba	6-10	11-15	de Pago	Si	Si
EJB	16-20	> 20	de Pago	Si	Si
GCF	0	11-15	libre - De Pago	Si	Si
KDE	1-5	6-10	Libre	Si	Si
UNO	1-5	1-5	Libre	Si	Si
XPCOM	1-5	6-10	Libre	Si	Si

Fuente: Los autores.

Tabla 4. Puntuación Soporte

Framework	Indicadores de soporte					
	CDED	CDID	tipo de soporte	web site oficial	Cominidad Dedicada	Total
.NET	2	4	1,5	4	3	14,5
AFC	0	1	0	0	0	1
BONOBO	2	1	3	0	0	6
COM	3	4	1,5	4	3	15,5
CORBA	2	3	1,5	4	3	13,5
EJB	4	5	1,5	4	3	17,5
GCF	0	3	2	4	3	12
KDE	1	2	3	4	3	13
UNO	1	1	3	4	3	12
XPCOM	1	2	3	4	3	13

Fuente: Los autores.

Nota: El framework COM no se tendrá en cuenta como uno de los mejores ya que tiende a ser reemplazado por .Net.

Los frameworks mejor evaluados para esta variable son:

- EJB
- .Net
- Corba

17.3 COMPATIBILIDAD

En la tabla 5 se observan los resultados de cada framework con respecto a los indicadores de compatibilidad. En la primera fila se posicionan los frameworks y en la segunda columna los indicadores a evaluar.

Tabla 5. Compatibilidad

Frameworks		.Net	AFC	Bonobo	COM	Corba	EJB	GCF	KDE	UNO	XPCOM	
sistemas operativos	linux			X		X	X	X	X	X	X	
	mac					X	X	X	X	X	X	
	windows	X	X		X	X	X	X	X	X	X	
Indicadores	Lenguajes	C#	X									
		VB	X			X						
		Ruby										X
		C			X	X	X				X	X
		Smalltalk					X					
		F#	X									
		Ada					X					
		Perl	X		X							X
		Python	X		X		X					X
		Java Script										X
		Fortran	X									
		Prolog	X									
		CLI										X
		Java		X	X		X	X			X	
		J#	X									
		COBOL			X		X					
		Lisp			X		X					
Delphi	X											
C++	X		X	X	X		X	X	X	X		
IDLScript						X						

Fuente: Los autores.

Tabla 6. Puntuación Compatibilidad

Framework	Indicadores de Compatibilidad		
	Sistemas operativos	Lenguajes de Programación	Total
.NET	2	7	9
AFC	2	0,7	2,7
BONOBO	2	4,9	6,9
COM	2	2,1	4,1
CORBA	6	6,3	12,3
EJB	6	0,7	6,7
GCF	6	0,7	6,7
KDE	6	0,7	6,7
UNO	6	2,1	8,1
XPCOM	6	4,9	10,9

Fuente: Los autores.

Los frameworks mejor evaluados para esta variable son:

- Corba
- XpCom
- .Net

17.4 INTEROPERABILIDAD

En la tabla 7 se observan los resultados de cada framework con respecto a los indicadores de interoperabilidad. En la primera columna se posicionan los frameworks y en la segunda fila los indicadores a evaluar.

Tabla 7. Interoperabilidad

Framework	indicadores									
	.Net	AFC	Bonobo	COM	Corba	EJB	GCF	KDE	UNO	XPCOM
.Net	X			X						X
AFC		X								
Bonobo			X		X	X		X		X
COM	X			X	X		X			X
Corba			X	X	X	X				X
EJB			X		X	X				X
GCF				X			X			
KDE			X					X		
UNO									X	
XPCOM	X		X	X	X	X				X

Fuente: Los autores.

Tabla 8. Puntuación Interoperabilidad

Frameworks	Interoperabilidad con otros Frameworks
.net	6
AFC	2
BONOBO	10
COM	10
CORBA	10
EJB	8
GCF	4
KDE	4
UNO	2
XPCOM	12

Fuente: Los autores.

A pesar de que Bonobo y COM cuentan con una alta puntuación, son frameworks que tienden a desaparecer, por lo tanto no se tomarán en cuenta entre los mejores. Los frameworks mejor evaluados para esta variable son:

- XpCom
- Corba
- EJB

17.5 ACCESIBILIDAD

En la tabla 9 se observan los resultados de cada framework con respecto a los indicadores de accesibilidad. En la primera columna se posicionan los frameworks y en la segunda fila los indicadores a evaluar.

Tabla 9. Accesibilidad

Framework	indicadores		
	Licencia		Requisitos
	CopyLeft	CopyRight	
.Net		X	Visual Studio
AFC		X	Java SDK >= 2,0 y Visual J++
Bonobo	X		Compilador del Lenguaje a trabajar
COM		X	Compilador MIDL y del lenguaje a trabajar
Corba	X		IDL y Compilador del Lenguaje a trabajar
EJB	X		JDK y Servidor Virtual
GCF	X	X	Qt, Qt Creator con MinGW o Visual Studio >= 2008
KDE	X		kdelibs-devel >= 4.3, Qt-devel >= 4.5.2, Boost MultiIndex library >= 1.35, g++ >= 4.0 (o compatible) para Linux, msvc >= 8 para sistemas Windows, GNU make (para Linux), CMake >= 2.6.2, KDevelop Development Platform >= 1.0.0, Grantlee >= 0.1.7
UNO	X		Apache OpenOffice - Software SDK
XPCOM	X		Mozilla Firefox >= 3.0 y Gecko SDK >=1.9.0

Fuente: Los autores.

Tabla 10. Puntuación Accesibilidad

Frameworks	Indicadores de accesibilidad		
	licencia	requisitos	total
.Net	6	8	14
AFC	6	6	12
BONOBO	12	8	20
COM	6	6	12
CORBA	12	6	18
EJB	12	6	18
GCF	9	6	15
KDE	12	1	13
UNO	12	6	18
XPCOM	12	6	18

Fuente: Los autores.

Los frameworks mejor evaluados para esta variable son:

- EJB
- Corba
- XpCom
- Uno

18. CONCLUSIONES

- Existen muchos criterios de evaluación para la calidad de software que van orientados a las aplicaciones finales, sin embargo las variables elegidas para el instrumento de medición fueron aquellas que se acoplaban mejor a la evaluación de un framework de desarrollo.
- Con base en las tablas de Usabilidad, Soporte, Compatibilidad, Interoperabilidad y Accesibilidad, encontramos que los frameworks más destacados fueron: EJB, Corba, .Net y XpCom, pues obtuvieron las mejores calificaciones en las 5 variables que fueron evaluados.
- El framework EJB destaco por su gran soporte en la web y fácil usabilidad, reflejando una curva de aprendizaje baja, su mayor desventaja es que solo ofrece la posibilidad de trabajar en el lenguaje de programación java.
- El framework CORBA destaco por gran compatibilidad e interoperabilidad, su mayor desventaja es la complejidad para su uso e integración, lo que una curva de aprendizaje muy alta.
- El framework .Net destaco por su soporte y fácil usabilidad, sus mayores desventajas fueron la interoperabilidad y accesibilidad, sin embargo su fácil uso lo posiciona como una buena opción para el desarrollo de componentes.
- El framework XpCom destaco por Su gran interoperabilidad, compatibilidad y acceso, su mayor desventaja es la usabilidad ya que su curva de aprendizaje es alta.
- Al concluir de manera incompleta la actividad en algunos de los frameworks bajo las condiciones especificadas en el instrumento de medición, nos indica que sus curvas de aprendizaje son más elevadas, debido a la falta de documentación, ejercicios prácticos y/o complejidad del mismo.

BIBLIOGRAFÍA

Aguilar, Luis Joyanes. Programación Orientada a Objetos, 1996

Bertoa, Manuel F., et al. Aspectos de Calidad en el Desarrollo de Software Basado en Componentes.

SZYPERSKI, Clemence. Component Software. Beyond Object -Oriented Programming, 1998

FREEMAN, Edwar .Strategic Management: A Stakeholder Approach. Boston, 1984.

FONTÁN, Emmanuel. Metodología de desarrollo de software basado en componentes, 2010

MORALES MACHUCA, Carlos. Andrés. (s.f.). Estado del Arte: Servicios Web, 2010

Rojas, M. A., & García, J. C. INTRODUCCIÓN Y PRINCIPIOS BÁSICOS DEL DESARROLLO DE SOFTWARE BASADO EN COMPONENTES, 2004

ROJAS, M. A., & GARCÍA, J. C. JERARQUÍA Y GRANULARIDAD DE COMPONENTES DE SOFTWARE PARA PYMES EN BOGOTÁ. Bogotá, 2005

VÉLEZ SERRANO, José F. et al. Diseñar y programar, todo es empezar: Una introducción a la programación orientada a objetos usando UML y java. Madrid: DYKYNSON, 2011

Constitución Política de Colombia, artículo 61, 1991

ISO/IEC 14598-1:1999 (E)

ISO/IEC FDIS9126-1:2000(E)

Reporte del Caos [en línea]. The Standish Group, 1994 [citado el 10 de Febrero del 2013]. Disponible en:

<<http://www.materiabiz.com/mbz/ityoperaciones/nota.vsp?nid=29672> >

Comportamiento de proyectos de TI [en línea]. [Citado el 10 de Febrero de 2013]. Disponible en:

<<http://www.alejandrobarrros.com/content/view/691759/Comportamiento-de-proyectos-TI-Estan-en-deuda.html> >

Reporte del Caos [en línea]. The Standish Group, 2009 [citado el 10 de Febrero de 2013]. Disponible en:

<http://www1.standishgroup.com/newsroom/chaos_2009.php >

Framework [en línea]. [Citado el 21 de Julio de 2013]. Disponible en:

<<http://whatis.techtarget.com/definition/framework> >

Paradigma de Programación [en línea]. 2009 [citado el 21 de Julio de 2013]. Disponible en:

<<http://labsys.frc.utn.edu.ar/ppr-2009/Unidad%20I%20-%20POO%20Avanzada/Apunte/Unidad%20I%20-%20POO%20Avanzada.pdf>>

Repositorio [en línea]. [Citado el 21 de Julio de 2013]. Disponible en:

<http://bibliotecabiologia.usal.es/tutoriales/catalogos-repositorios-bibliosvirtuales/repositorios_digitales.html >

Widget [en línea]. [Citado el 21 de Julio de 2013]. Disponible en:

<<http://www.alegsa.com.ar/Dic/widget.php> >

IIOP [en línea]. [Citado el 21 de Julio de 2013]. Disponible en:

<<http://searchcio-midmarket.techtarget.com/definition/IIOP> >

IDL [en línea]. [Citado el 21 de Julio de 2013]. Disponible en:

<<http://whatis.techtarget.com/definition/IDL-interface-definition-language>>

GUI [en línea]. [Citado el 21 de Julio de 2013]. Disponible en:

<http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Interfaces/enlightment/definiciones_1.html>

IDE [en línea]. [Citado el 21 de Julio de 2013]. Disponible en:

<<http://curso-sobre.berlios.de/introsobre/2.0.1/sobre.html/sec-ide.html> >

ISO [en línea]. [Citado el 21 de Julio de 2013]. Disponible en:

<<http://www.iso.org/iso/home.htm> >

IEC [en línea]. [Citado el 21 de Julio de 2013]. Disponible en:

<<http://www.iec.ch/index.htm>>

Website [en línea]. [Citado el 21 de Julio de 2013]. Disponible en:

<http://diseno.ciberaula.com/articulo/que_es_un_sitio_web/ >

Copyleft [en línea]. [Citado el 21 de Julio de 2013]. Disponible en:

<<http://www.gnu.org/philosophy/categories.es.html#CopyleftedSoftware> >

Api [en línea]. [Citado el 21 de Julio de 2013]. Disponible en:
<<http://www.elwebmaster.com/referencia/api-interface-de-programacion-de-aplicaciones> >

Ipc [en línea]. [Citado el 21 de Julio de 2013]. Disponible en:
<<http://carocols.files.wordpress.com/2011/10/modos-de-transmisi3b3n.pdf> >

Plugin [en línea]. [Citado el 21 de Julio de 2013]. Disponible en:
<<http://informaticandotf.blogspot.com/2010/08/plugin-definicion.html> >

Vera, Fredy Humberto & Rojas, Fernando. Propuesta de un Proceso de Desarrollo de Componentes Software Reutilizables. En: Gerencia Tecnológica Informática, 2008, vol. 7, no. 19. [En Línea]. [Citado el 18 de Mayo de 2013]. Disponible en:
<<http://revistas.uis.edu.co/index.php/revistagti/article/view/160/870> >

Historia de .Net [en línea]. [Citado el 15 de Junio de 2013] Disponible en:
<<http://blogs.msdn.com/b/warnov/archive/2010/07/04/breve-historia-del-framework-net.aspx> >

Introducción a .NET Framework [en línea]. [Citado el 15 de Junio de 2013]. Disponible en:
<<http://msdn.microsoft.com/es-es/library/vstudio/hh425099> >

Biblioteca de clases .Net [en línea]. [Citado el 15 de Junio de 2013]. Disponible en:
<<http://msdn.microsoft.com/es-es/library/vstudio/zw4w595w> >

Historia de AFC [en línea]. [Citado el 15 de Junio de 2013]. Disponible en:
<http://www.extropia.com/tutorials/java/java_afc_jfc.html >

APPLICATION FOUNDATION CLASSES [en línea]. [Citado el 15 de Junio de 2013]. Disponible en:
<<http://www2.sys-con.com/itsg/virtualcd/java/archives/0207/fronkowiakp14/index.html> >

BONOBO AND CORBA [en línea]. [Citado el 15 de Junio de 2013]. Disponible en:
<<https://developer.gnome.org/platform-overview/2.32/bonobo-corba.html.en> >

La arquitectura de componentes Bonobo: Una introducción práctica [en línea]. [Citado el 15 de Junio de 2013]. Disponible en:
<<http://acsblog.es/articulos/trunk/ProgramacionActual/bonobo/html/c25.html> >

COM: Component Object Model Technologies práctica [en línea]. [Citado el 15 de Junio de 2013]. Disponible en: <<http://www.microsoft.com/com/default.mspx> >

Características técnicas de COM práctica [en línea]. [Citado el 15 de Junio de 2013]. Disponible en: < <http://msdn.microsoft.com/es-es/library> >

Historia de CORBA [en línea]. [Citado el 15 de Junio de 2013]. Disponible en: < http://www.omg.org/gettingstarted/history_of_corba.htm >

CORBA® BASICS [en línea]. [Citado el 15 de Junio de 2013]. Disponible en: < <http://www.omg.org/gettingstarted/corbafaq.htm> >

Historia EJB [en línea]. [Citado el 15 de Junio de 2013]. Disponible en: < <https://code.google.com/p/fap-devel/wiki/EJB3> >

Introducción a los JavaBeans [en línea]. [Citado el 15 de Junio de 2013]. Disponible en: <<http://www.sc.ehu.es/sbweb/fisica/cursoJava/applets/javaBeans/fundamento.htm>>

Historia GCF [en línea]. [Citado el 15 de Junio de 2013]. Disponible en: < <http://www.vcreatelogic.com/products/gcf/highlights> >

Generic Component Framework (GCF) [en línea]. [Citado el 15 de Junio de 2013]. Disponible en: < <http://www.vcreatelogic.com/docs/gcf-2.5.0/html> >

Arquitectura GCF [en línea]. [Citado el 15 de Junio de 2013]. Disponible en: < <http://www.prashanthudupa.com/?s=gcf> >

Historia KDE [en línea]. [Citado el 15 de Junio de 2013] Disponible en: < <http://www.kde.org/community/history> >

Características técnicas kde [en línea]. [Citado el 15 de Junio de 2013] Disponible en: < <http://www.ibm.com/developerworks/library/l-kparts> >

Universal Network Objects (UNO) [en línea]. [Citado el 15 de Junio de 2013] Disponible en: <<http://www.openoffice.org/udk/common/man/uno.html>>

XPCOM [en línea]. [Citado el 15 de Junio de 2013] Disponible en: < <https://developer.mozilla.org/es/docs/XPCOM> >

ANEXO A. PRUEBAS DE SOPORTE

Se realizaron en google búsquedas avanzadas en español y en ingles, las comillas en google indican que la palabra debe estar contenida en resultado de la búsqueda, el símbolo “+” (más) añade condiciones a la búsqueda, de esta forma evitamos que la búsqueda incluya conectores gramaticales, teniendo resultados más limpios.

Las consultas se construyeron de la siguiente manera, siempre empezando con el nombre del framework y filtrando los resultados solo en español:

“nombre del framework” + “componente”
“nombre del framework” + “framework” + “Empresa Desarrolladora”
“nombre del framework” + “tutorial”
“nombre del framework” + “documentación”
“nombre del framework” + “Desarrollo”

Para la búsqueda en ingles se uso la opción que otorga google para filtrarlos solo en ingles, y uso la misma estructura

“nombre del framework” + “component”
“nombre del framework” + “framework” + “Empresa Desarrolladora”
“nombre del framework” + “tutorial”
“nombre del framework” + “development”
“nombre del framework” + “Documentation”

se consultaron 100 diferentes paginas por cada idioma mostrados en los resultados de google, para los resultados de la investigación solo se toman cuenta las páginas que contengan información relevante sobre el framework y sin contar paginas repetidas, es decir paginas dentro del mismo sitio

A.1 .NET

Español

[http://msdn.microsoft.com/es-es/library/ms166014\(v=sql.105\).aspx](http://msdn.microsoft.com/es-es/library/ms166014(v=sql.105).aspx)

<https://www.dotnetclubs.com/forums/p/2064/5261.aspx>

<http://geoavila.com/about/>

<http://www.microsoft.com/es-es/download/details.aspx?id=26>

http://codigofacilito.com/videos/tutorial_visual_basic_net_parte_curso_vbnet_3

<http://www.microsoftvirtualacademy.com/training-courses/desarrollo-net>

<http://ingenieriasystems.blogspot.com/2013/04/introduccion-net-desarrollo-net-mva.html>

Ingles

<http://ironpython.net/documentation/dotnet/>

<http://blogs.msdn.com/b/dotnet/archive/2013/02/19/net-documentation-improvements.aspx>

[http://msdn.microsoft.com/en-us/library/ms166014\(v=sql.105\).aspx](http://msdn.microsoft.com/en-us/library/ms166014(v=sql.105).aspx)

<http://forums.asp.net/t/1829396.aspx/1>

<http://www.microsoft.com/en-us/download/details.aspx?id=17851>

<http://www.functionx.com/vbnet/>

<http://mottishaked.com/training/dot-net-tutorial>

<http://www.visualbuilder.com/dotnet/tutorial/>

<http://visualbasic.about.com/od/learnvbnet/a/blecvbsbs0101.htm>

<http://www.hsd.com.au/services/dotnetdevelopment>

<http://forums.devshed.com/net-development-87/>

<http://forums.asp.net/t/1908804.aspx/1>

A.2 AFC

Español

Ninguno

Ingles

<http://www.os3.it/labs/afc/html/intro/intro.html>

[http://msdn.microsoft.com/en-us/library/aa239173\(v=vs.60\).aspx](http://msdn.microsoft.com/en-us/library/aa239173(v=vs.60).aspx)

A.3 BONOBO

Español

<http://www.es.gnome.org/>

<http://zarza.usal.es/~fgarcia/docencia/poo/04-05/Trabajos/Bonobo.pdf>

<http://paginaspersonales.deusto.es/porduna/documentacion/mono/julio-2005/transparencias/referencias.faltan.txt>

<http://www.calcifer.org/documentos/librognome/bonobo.html>

<http://acsblog.es/articulos/trunk/ProgramacionActual/bonobo/html/x76.htm>

|

Ingles

<http://www.jirka.org/pong-documentation/upl-custom-bonobo-controls.html>

<http://cutebugs.net/files/corba/book.html>

<http://www.djcbsoftware.nl/projecten/nluug-bonobo/>

<http://www.ibm.com/developerworks/webservices/library/co-bnbo2/index.html>

A.4 COM

Español

http://infoplcn.net/files/documentacion/comunicaciones/infoPLC_net_Tknika_Actividad_ad_6 OPC_Conceptos_.pdf

<http://www.slideshare.net/jabenitez88/3creacion-de-componentes-visuales>

<http://dSPACE.unach.edu.ec/bitstream/123456789/66/1/FI-ESC-40A002.pdf>

<http://es.scribd.com/doc/11608048/4/COMPONENT-OBJECT-MODEL-COM>

<http://msdn.microsoft.com/es-es/library/vstudio/dd984046%28v=vs.100%29.aspx>

<http://www.waldyd.8m.com/main.htm>

<http://www.fpress.com/revista/num9701/ene97.htm>

http://digeset.ucol.mx/tesis_posgrado/Pdf/Miguel%20Angel%20Miranda%20Guerreiro.pdf

<http://www.ica.luz.ve/cstufano/CursoDeRedes/DCOM.pdf>

<http://www.php.net/manual/es/book.com.php>

Ingles

<http://www.novell.com/documentation/extendas35/docs/help/books/pgCOM.html>

<http://pro.openecry.com/api/api/index.html>

<http://www.tenouk.com/visualcplumfc/visualcplumfc23d.html>

<http://www.microsoft.com/com/default.mspx>

<http://graphcomp.com/info/specs/com/comch02.htm>

<http://www.codeproject.com/Articles/633/Introduction-to-COM-What-It-Is-and-How-to-Use-It>
<http://www.codeguru.com/cpp/com-tech/activex/tutorials/article.php/c5567/Step-by-Step-COM-Tutorial.htm>
<http://www.tenouk.com/visualcplusmfc/visualcplusmfc23.html>
<http://www.dreamincode.net/forums/topic/165742-microsoft-working-with-component-object-model-objects-part-i/>
<http://www.peterindia.net/COMLinks.html>
<http://atlcomtutorial.blogspot.com/>
<http://www.cs.umd.edu/~pugh/com/>
<http://support.sas.com/documentation/cdl/en/appdevgd/62709/HTML/default/viewer.htm#p0zfyef0erc1cvn1qb2twqfrpz53.htm>
<http://edndoc.esri.com/arcobjects/9.0/ArcGISDevHelp/DevelopmentEnvs/COM/IntroToCOM.htm>
<http://www.cs.utah.edu/flux/oskit/html/oskit-wwwch4.html>
http://webserver.ignou.ac.in/virtualcampus/BIT_StudyMaterials/LabManuals/component.htm#1

A.5 CORBA

Español

<http://www.programatium.com/corba.htm>
http://www.lawebdelprogramador.com/cursos/Corba/6488-Tutorial_de_CORBA_desde_Java.html
<http://www.ramonmillan.com/tutoriales/corba.php>
<http://acsblog.es/articulos/trunk/LinuxActual/CursoCORBA/>
<http://www.calcifer.org/documentos/librognome/corba.html>
http://www.programacion.com/articulo/curso_practico_de_corba_en_gnu-linux_162/4#corba_prof

Ingles

<http://www.omg.org/gettingstarted/corbafaq.htm>
<http://www.ece.rutgers.edu/~parashar/Classes/ece451-566/slides/lecture-dist-obj-corba-liu.pdf>
<http://people.inf.elte.hu/toke/CORBA%20distr-object-management/OMG-current-doc-CORBA3.1/CORBA-08-01-08-part3.pdf>
<http://www-cdfonline.fnal.gov/daq/CORBAXXX/tutorial.html>
<http://docs.oracle.com/javase/7/docs/technotes/guides/idl/jidlExample.html>
<http://www.yolinux.com/TUTORIALS/CORBA.html>
<http://www.ois.com/Products/what-is-corba.html>
<http://ovir.icp.ac.ru/corba/books/Teach14/fm/fm.htm>
<http://publib.boulder.ibm.com/infocenter/rsdp/v1r0m0/topic/com.ibm.help.download.rhapsody.doc/pdf/corba.pdf>

A.6 GCF

Español
Ninguno

Inglés

<http://www.kdab.com/kdab-products/kdcf-overview/>

<http://www.vcreatelogic.com/products/gcf/>

<http://www.prashanthudupa.com/?s=gcf>

[http://www.training-](http://www.training-classes.com/programs/04/70/47099_generic_component_framework_gcf.php)

[classes.com/programs/04/70/47099_generic_component_framework_gcf.php](http://www.training-classes.com/programs/04/70/47099_generic_component_framework_gcf.php)

http://foss.in/2007/register/slides/Component_Software_Design_Development_430.pdf

A.7 EJB

Español

<http://www.sc.ehu.es/sbweb/fisica/cursoJava/applets/javaBeans/fundamento.htm>

http://www.programacion.com/articulo/javabeans_enterprise_84/2

<http://www.slideshare.net/SiegfriedG/guia-ejb-deshabdig>

<http://www.help400.com/asp/scripts/nwart.asp?Num=131&Pag=28&Tip=M>

<http://code.google.com/p/fap-devel/wiki/EJB3>

<http://holamundojava.blogspot.com/2006/11/enterprise-javabeans.html>

http://www.lawebdelprogramador.com/foros/EJB_Enterprise_Java_Beans/index1.html

<http://www.apuntesdejava.com/2007/11/tutorial-ejb-30-enterprise-beans-con.html>

<http://www.roseindia.net/javabeans/javabeans.shtml>

<http://www.dosideas.com/noticias/java.html>

<http://www.osgg.net/omarsite/resources/proceedings/EJB.pdf>

<http://javaejb.osmosislatina.com/curso/basico.htm>

[http://www.fing.edu.uy/inco/cursos/tsi/TSI2/2010/teorico/03-](http://www.fing.edu.uy/inco/cursos/tsi/TSI2/2010/teorico/03-EnterpriseJavaBeans.pdf)

[EnterpriseJavaBeans.pdf](http://www.fing.edu.uy/inco/cursos/tsi/TSI2/2010/teorico/03-EnterpriseJavaBeans.pdf)

<https://cwiki.apache.org/GMOxDOC11/aplicacin-ejb-ejemplo.html>

<http://sistemas.uniandes.edu.co/~isis2503/dokuwiki/lib/exe/fetch.php?media=principal:isis2503-ejb3.pdf>

<http://www.davidmarco.es/blog/entrada.php?id=239>

<http://tomavideotutoriales.blogspot.com/2013/01/java-ejb-tutorial.html>

<http://skerzo.dyndns.org/web/Libros/Colecciones/600LibrosTecnicos/C%20U%20R%20S%20O%20S/04%20Curso%20JavaBeans%20-%20Sp.pdf>

http://sistemas.uniandes.edu.co/~csof5302/dokuwiki/lib/exe/fetch.php?media=principal:logica_de_negocio-ejbs.pdf

http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp?topic=%2Fcom.ibm.websphere.base.doc%2Fae%2Fcontainer_ejb_developing3beans.html&lang%3Des
<http://ccia.ei.uvigo.es/docencia/SCS/1011/transparencias/Tema5-2.pdf>
http://ocw.uc3m.es/ingenieria-telematica/software-de-comunicaciones-1/UDs_JEE/jee-unidad-2-nivel-de-negocio-en-j2ee-enterprise-java-beans
http://www.it.uc3m.es/~celeste/docencia/ittt/swc0405/traspas/2_Enterprise_JavaBeans.pdf

Ingles

<http://www.jboss.org/ejb3>
http://portal.aauj.edu/portal_resources/downloads/programming/enterprise_java_beans_developers_guide.pdf
<http://www.freejavaguide.com/ejb.pdf>
<https://cwiki.apache.org/GMOxDOC11/aplicacin-ejb-ejemplo.html>
<http://www.conceptgo.com/gsejb/resources.html>
<http://www.masslight.com/Chapter6.html>
<http://static.springsource.org/spring/docs/3.0.x/spring-framework-reference/html/ejb.html>
<http://www.dosideas.com/component/finder/search.html?q=EJB&Itemid=113>
http://docstore.mik.ua/oreilly/java-ent/jenut/ch07_01.htm
<http://www.oracle.com/technetwork/java/javaee/overview/index-jsp-135147.html>
<http://code.google.com/p/fap-devel/wiki/EJB3>
<http://publib.boulder.ibm.com/infocenter/radhelp/v8/index.jsp?topic=%2Fcom.ibm.javaee.doc%2Ftopics%2Fcejb3vejb21.html>
<https://netbeans.org/kb/docs/javaee/javaee-entapp-ejb.html>
<http://www.davidmarco.es/blog/entrada.php?id=239>
<http://www.mastertheboss.com/ejb-3>
<http://www.oxsus.net/tutorials/j2ee/entity-beans>
<http://www.roseindia.net/javabeans/index.shtml>
<http://www.redbooks.ibm.com/redbooks/SG246819/wwhelp/wwhimpl/js/html/wwhelp.htm>
<http://help.eclipse.org/indigo/index.jsp?topic=%2Forg.eclipse.jst.ejb.doc.user%2Ftopics%2Fcover.html>
<http://www.math.iitb.ac.in/resources/manuals/VAJ/VAJavaBeans.pdf>
<http://download.java.net/jdk8/docs/technotes/guides/rmi-iiop/interop.html>
<http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc38033.0520/html/easfg/BABEFEGE.htm>
http://help.sap.com/saphelp_nw73ehp1/helpdata/en/4c/41ed7f0660697de10000000a42189b/frameset.htm
http://www.cs.ait.ac.th/~on/O/oreilly/java-ent/jenut/ch07_01.htm

A.8 KDE

Español

[https://techbase.kde.org/Development/Tutorials_\(es\)](https://techbase.kde.org/Development/Tutorials_(es))

<http://developer.kde.org/~larrosa/es/tutorial/p4.html>

http://www.ukuug.org/events/linux2004/programme/paper-JPedersen-2/index_es.html

Inglés

http://techbase.kde.org/Development/Tutorials/Using_KParts

<http://api.kde.org/4.x-api/kdelibs-apidocs/kparts/html/>

<http://www.kde-forum.org/artikel/1982/easy-example-on-using-kparts.html>

<http://www.ibm.com/developerworks/library/l-kparts/>

<http://kde.6490.n7.nabble.com/>

<http://tinf2.vub.ac.be/~dvermeir/manual/KDE20Development-html/ch12.html>

A.9 UNO

Español

https://help.libreoffice.org/Common/Integrating_new_UNO_components/es

Inglés

<http://www.openoffice.org/udk/common/man/uno.html>

http://wiki.openoffice.org/wiki/Documentation/BASIC_Guide/UNO

http://wiki.openoffice.org/wiki/Tutorial_UNO_Library

<http://api.libreoffice.org/docs/tools.html>

A.10 XPCOM

Español

<https://developer.mozilla.org/es/docs/XPCOM>

http://www.inecc.gob.mx/descargas/calaire/2008_inf_ind_meteoro.pdf

<http://www.dspace.espol.edu.ec/bitstream/123456789/9036/1/D-39818.pdf>

<http://materias.fi.uba.ar/7500/zavolinsky-tesisdegradoingenieriainformatica.pdf>

<http://es.slideshare.net/bijolianabhi/firefox-extension-development>

Ingles

http://kb.mozillazine.org/Implementing_XPCOM_components_in_JavaScript

<http://www.ibm.com/developerworks/webservices/library/co-xpcom/index.html>

<http://www.activestate.com/komodo-ide>

<http://books.mozdev.org/html/mozilla-chp-8.html>

<http://benjamin.smedbergs.us/tests/XPCOM-intro.pdf>

<http://www.codeproject.com/Articles/87465/A-Simple-XPCOM-Tutorial>

http://www.rdsquared.net/2009/08/12/Creating_XPCOM_Components.pdf

<http://www.ors.uoguelph.ca/docs/xul/xultu/xpcom.html>