

CRITERIOS DE EVALUACIÓN DE PLATAFORMAS DE DESARROLLO DE APLICACIONES EMPRESARIALES PARA AMBIENTES WEB

MARIA HELENA TREJOS ARROYAVE
DIEGO FERNANDO ZAMORA CARDONA

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA
2012

CRITERIOS DE EVALUACIÓN DE PLATAFORMAS DE DESARROLLO DE
APLICACIONES EMPRESARIALES PARA AMBIENTES WEB

MARIA HELENA TREJOS ARROYAVE
DIEGO FERNANDO ZAMORA CARDONA

Monografía para optar al título de Ingeniero de Sistemas y computación

Asesor
Ph. D JULIO CESAR CHAVARRO PORRAS

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA
2012

Nota de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

DEDICATORIA

A Dios todopoderoso mi compañía y mi confidente de todas las alegrías y angustias de este camino, A mi familia por ser apoyo incondicional durante toda la carrera, por darme ánimos cada día, por escuchar mis alegrías y aguantar todas mis quejas, a mis Padres por estar siempre atentos y sentir como su propio triunfo cada escalón que subía para llegar a mi Meta porque no me sentía sola, a mi esposo por trasnochar conmigo y por ser un aliciente para continuar.

Maria Helena Trejos Arroyave.

A Dios, mi señor y guía, mi proveedor de vida y alegría que nunca me deja desfallecer, A mis padres que me apoyaron incondicionalmente y me dieron su mano en todo momento para poder cumplir está, una de mis grandes metas y aspiraciones, por la paciencia que me han tenido al no poder compartir más tiempo con ellos por cumplir mi sueño, A mi familia, por darme la energía y el ánimo para seguir adelante.

Diego Fernando Zamora Carmona

AGRADECIMIENTOS

Al Ingeniero Julio Cesar Chavarro, tutor del trabajo de grado por su gran aporte y ayuda desinteresada en la elaboración adecuada del documento.

A los ingenieros Carlos Meneses y Omar Iván Trejos por su apoyo y por darnos siempre ánimo para cumplir con esta Meta.

A todos los compañeros y profesores por haber sido parte de los nuevos amigos que se forman durante todo el proceso de aprendizaje y que han estado presentes para ayudarnos a seguir adelante cada vez que lo necesitamos.

CONTENIDO

	Pág.
1. GENERALIDADES	10
1.1 INTRODUCCION.....	10
1.2 PLANTEAMIENTO DEL PROBLEMA.....	10
1.3 JUSTIFICACIÓN.....	12
1.4 OBJETIVO GENERAL	13
1.5 OBJETIVOS ESPECÍFICOS.....	13
1.6 MARCO REFERENCIAL	13
1.6.1 Marco Teórico	13
1.6.2 Marco Conceptual	14
2. DESCRIPCION DE LOS LENGUAJES Y FAMILIAS TECNOLOGICAS QUE SOPORTAN LAS PLATAFORMAS WEB MÁS EXTENDIDAS EXISTENTES ACTUALMENTE	25
2.1 LENGUAJES QUE SOPORTAN PROGRAMACION WEB.....	25
2.1.1 JAVA:	25
2.1.2 RUBY ON RAILS.....	26
2.1.3 PHP	27
2.1.4 ASP .Net	28
2.1.5 C#.....	28
2.1.6 COMPARACION	29

3. PLATAFORMAS DE DESARROLLO MULTILENGUAJE Y AMBIENTES INTEGRADOS DE DESARROLLO CON SOPORTE WEB	32
3.1 NETBEANS.....	32
3.2 ECLIPSE.....	33
3.3 JDEVELOPER	33
3.4 VISUAL STUDIO.....	33
3.5 DELPHI.....	34
3.6 GENEXUS	34
3.7 COMPARACION.....	34
4. PRINCIPALES PLATAFORMAS UTILIZADAS EN LA EJECUCIÓN DE APLICACIONES EMPRESARIALES WEB.....	36
4.1 J2EE	36
4.2 CORBA	38
4.3 WEBSPHERE	39
4.4 .NET.....	41
4.5 COMPARACION.....	42
5. CRITERIOS DE COMPARACIÓN DE LAS PLATAFORMAS DE DESARROLLO	43
6. CONCLUSIONES	48
BIBLIOGRAFÍA.....	50

LISTA DE ILUSTRACIONES

	Pág.
Ilustración 1. Estilo arquitectural - Cliente/ Servidor.....	17
Ilustración 2. Estilo arquitectural - Basado en componentes.....	18
Ilustración 3. Estilo arquitectural - arquitectura en capas(N-Layer).....	19
Ilustración 4. Estilo arquitectural - presentación desacoplada	19
Ilustración 5. Estilo arquitectural - arquitectura en capas (N-tier).....	20
Ilustración 6. Estilo arquitectural - Arquitectura orientada al dominio (DDD)....	21
Ilustración 7. Estilo arquitectural - orientado a objetos (OO).....	21
Ilustración 8. Estilo arquitectural - Bus de servicio (Mensajes)	22
Ilustración 9: Elementos de la plataforma java.....	36
Ilustración 10. Modelo de aplicación empresarial en J2EE	38
Ilustración 11. El modelo OMA(object Management Architecture) definido por la OMG, muestra los componentes agrupados en función del nivel de reutilización de los mismos.	39
Ilustración 12. Componentes Websphere	40
Ilustración 13 . Plataforma de ejecución intermedia en .NET.....	42

LISTA DE TABLAS

Tabla 1: Tabla para evaluación de los criterios de acuerdo a cada plataforma y lenguaje.	47
---	----

1. GENERALIDADES

1.1 INTRODUCCION

Las aplicaciones web han estado tomando fuerza en los últimos años, esto debido a la practicidad de las mismas, entre otras razones, por la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales y son accesibles desde cualquier lugar del mundo gracias a la red de redes, Internet.

Los mecanismos de desarrollo de aplicaciones Web, recogen elementos comunes al desarrollo de aplicaciones empresariales, pero tienen características propias en análisis, diseño, e implementación. Estos elementos serán independientes del estilo arquitectónico que se decida implementar y también de la arquitectura de software.

Para las empresas desarrolladoras de software al igual que para las que tienen su departamento pero el desarrollo de software no es su especialidad, siempre sus ingenieros de sistemas se tendrán que hacer la pregunta, ¿Cuál es la mejor plataforma de desarrollo empresarial para mis aplicaciones web?, es un tema que presenta cierta independencia de la arquitectura que se va a usar, del modelo de datos y en sí, de todo el proceso de ingeniería de los proyectos.

El proyecto propone un conjunto de criterios que deben servir de guía al momento de seleccionar una de las diferentes plataformas de desarrollo empresarial para aplicaciones web, que existen actualmente en el mercado.

1.2 PLANTEAMIENTO DEL PROBLEMA

El inminente crecimiento de las empresas y con ellas el de la tecnología obliga a las empresas a dar un ordenamiento diferente a sus sistemas de información y con ello a la búsqueda de herramientas y plataformas que le den el sustento para manejar los volúmenes de información que diariamente se le presentan. Esto lleva a las empresas a buscar soluciones que permitan interconectar sus sistemas, o darles una estructura homogénea a sus sistemas.

Cualquier solución debe incorporar los dos elementos. La interconexión de sus sistemas es abordada por los estilos arquitecturales y las arquitecturas de referencia del sistema de información para las empresas o para el dominio de ellas.

Por otra parte, la homogenización de sus sistemas, requiere de la búsqueda de una integridad de las tecnologías y herramientas que les brindaran a los usuarios del proceso de desarrollo.

Plataformas para el desarrollo de sistemas hay muchas, pero en la actualidad tenemos opciones que son especializadas en el desarrollo empresarial, pues estas le brindan diferentes características en cuanto al rendimiento, la escalabilidad, el diseño, y en general en las características necesarias para el día a día de una gran empresa.

Por tanto, es necesario poderlas comparar y poder responder a la pregunta: ¿Cuál plataforma de desarrollo empresarial es más adecuada para un proyecto particular?, Una respuesta adecuada debe conllevar a una selección objetiva. Con lo cambiantes que son los sistemas y las tecnologías, documentarse bien para poder tomar esta importante decisión, costara buen tiempo, puesto que podría tener consecuencias exitosas o por el contrario tener consecuencias desastrosas para la empresa.

En síntesis, el problema es que no existe un conjunto de criterios que ayuden a evaluar una plataforma para el desarrollo de aplicaciones empresariales para ambientes web y facilitar el proceso de escoger la que mejor se acomode a las necesidades de la empresa.

1.3 JUSTIFICACIÓN

La economía del país ha venido creciendo cada año. En el 2011 tuvo un crecimiento superior al 5%, y en el 2012 se prevé un crecimiento del 4.3%¹. Esto jalona todo el desarrollo del país, obligando a las empresas a solidificarse y buscar soluciones tecnológicas que le permitan estar presente activamente en el mercado.

Teniendo en cuenta el constante crecimiento de las empresas, y la cantidad de información que cada una de ellas debe manejar cada día de forma ágil, segura, confiable, se hace necesario pensar en soluciones empresariales robustas, integradas y de fácil acceso para todos los usuarios, dando un panorama diferente a los departamentos de sistemas y a los Ingenieros que los administran.

Dados los niveles de información y la necesidad de que esta esté segura, las empresas utilizan modelos arquitectónicos para sus sistemas y esto se acompaña de la necesidad de escoger una plataforma de desarrollo empresarial que permita agilidad, buen desempeño, soluciones eficientes y que se acoja a la arquitectura seleccionada, pero al momento de entrar a evaluar cual escoger, es ahí donde un Ingeniero puede tener inconvenientes, pues un error en esta decisión puede llevar a una empresa a la cola en el índice de crecimiento.

Es por ello que se ve la necesidad de estudiar las diferentes plataformas para desarrollo empresarial para ambientes web, de tal forma que permita establecer criterios de evaluación y comparación que le permitan a un Ingeniero de sistemas tomar la decisión correcta sobre la herramienta que debe elegir para desarrollar las aplicaciones que le darán a su empresa la solución que necesita.

¹ <http://www.portafolio.co/economia/economia-colombiana-crecera-del-5-al-6-al-cierre-del-2011>

1.4 OBJETIVO GENERAL

Definir y describir los criterios para la evaluación de las plataformas de desarrollo de aplicaciones empresariales para ambientes web.

1.5 OBJETIVOS ESPECÍFICOS

- a. Identificar las plataformas de desarrollo empresarial para ambientes web de mayor utilización y las que actualmente ofrecen mayor penetración en el mercado Colombiano.
- b. Estudiar las características generales y específicas de las plataformas de desarrollo empresarial para ambientes web más reconocidas en el mundo.
- c. Describir cada uno de los criterios de comparación de las plataformas de desarrollo empresarial para ambientes web más usadas en el mercado.
- d. Determinar cuáles serían los principales criterios de evaluación y comparación, así como los mecanismos de ponderación, que se deben tener en cuenta para elegir cual plataforma empresarial para ambientes web que se debería usar.

1.6 MARCO REFERENCIAL

1.6.1 Marco Teórico

La creación de aplicaciones empresariales para ambientes web incorpora cada día un mayor nivel de complejidad, por ejemplo en aspectos como calidad y cumplimiento de cronograma de trabajo; esto implica contar con equipos de desarrollo de excelentes habilidades y con una plataforma de desarrollo que permita llevar a la realidad todo el diseño de la aplicación con sus características especiales de acuerdo a la arquitectura solicitada.

Una de las actividades necesarias para entender la arquitectura es su modelamiento, para lo cual se ha convertido el lenguaje UML en un estándar de facto, que es enriquecido con estereotipos y artefactos para las diferentes áreas que cubre el desarrollo de software. Por ejemplo, existe extensión para modelado de negocios que por medio de sus diagramas de casos de uso permiten una mejor visión de las entidades y procesos del negocio.

Cada modelo representa el sistema desde un punto de vista particular, para esto debemos tener en cuenta cuales modelos usar de acuerdo a la necesidad, modelo de casos de uso, de implementación, de procesos, de seguridad, de interfaz de usuarios, modelo del sistema.

Todo lo anterior, entrega unas pautas claras a tener en cuenta para el desarrollo de las aplicaciones. Tal es el caso de las aplicaciones web, que nos plantea la siguiente inquietud, ¿en qué plataforma se debe implementar?, esta es una decisión que no es fácil de tomar y para ello es necesario contestar a otros varios interrogantes que permitan evaluar las necesidades y validar con las plataformas existentes en el mercado.

Una empresa puede incluso haber utilizado alguna plataforma de desarrollo en el pasado pero no tener la documentación que condense las características que posee y que le llevaron a tomar la decisión de usarla, cada nuevo desarrollo trae consigo nuevas decisiones a tomar y conceptos a evaluar tales como:

- Nivel de complejidad de la plataforma
- Necesidad de soluciones de software libre
- Se tiene personal especializado en un lenguaje específico
- Necesidad de compatibilidad con diferentes sistemas operativos.

1.6.2 Marco Conceptual

Para tener una idea clara del proyecto es necesario conocer los conceptos que se relacionaran directamente con el tema, estos son:

Aplicación empresarial: Es un paquete de software diseñado para ayudar a agilizar las tareas administrativas de la empresa, esta reproduce y automatiza los procesos de trabajo del negocio aumentando el rendimiento y reduciendo los costos. Esta puede ser instalada en un servidor de internet para permitir el acceso desde cualquier sucursal o desde cualquier ubicación donde se tenga conexión a internet.

Aplicación Web: (Web application, webapp). Es cualquier aplicación que puede ser accedida vía web por una red como internet o una intranet y tiene una base de datos asociada y que permite mayor interacción con el usuario. También se llaman así los programas que son ejecutados a través del navegador y se pueden ejecutar en múltiples plataformas.

En general, el término también se utiliza para designar aquellos programas informáticos que son ejecutados en el entorno del navegador (por ejemplo, un applet de Java) o codificado con algún lenguaje soportado por el navegador (como JavaScript, combinado con HTML); confiándose en el navegador web para que reproduzca (renderice) la aplicación.

Una de las ventajas de las aplicaciones web cargadas desde internet (u otra red) es la facilidad de mantener y actualizar dichas aplicaciones sin la necesidad de distribuir e instalar un software en, potencialmente, miles de clientes. También la posibilidad de ser ejecutadas en múltiples plataformas.

Características de las aplicaciones web:

- El usuario puede acceder fácilmente a estas aplicaciones empleando un navegador web (cliente) o similar.
- Si es por internet, el usuario puede entrar desde cualquier lugar del mundo donde tenga un acceso a internet.
- Pueden existir miles de usuarios pero una única aplicación instalada en un servidor, por lo tanto se puede actualizar y mantener una única aplicación y todos sus usuarios verán los resultados inmediatamente.
- Emplean tecnologías como Java, JavaFX, JavaScript, DHTML, Flash, Ajax, entre otros, que dan gran potencia a la interfaz de usuario.
- Emplean tecnologías que permiten una gran portabilidad entre diferentes plataformas. Por ejemplo, una aplicación web flash podría ejecutarse en un dispositivo móvil, en una computadora con Windows, Linux u otro sistema, en una consola de videojuegos, etc.

Ajax es un ejemplo de una tecnología de desarrollo web que le da gran poder de interactividad a las aplicaciones web.²

En pocas palabras una aplicación web es una herramienta muy útil, pues tiene una cantidad potencial de clientes exponencial gracias a internet. Con este tipo de soluciones solo es necesaria una conexión a internet para acceder a las aplicaciones, funciona con los mínimos requisitos de hardware, es fácil de usar, facilita la colaboración entre grupos de trabajo gracias a los beneficios que ha incorporado el cloud-computing, no requieren de instalación, la actualización se hace en un solo lugar, el servidor. Todas estas bondades son la que hacen que las soluciones web tengan más acogida cada día.

Aplicaciones empresariales web: Además de todas las cualidades que tiene una aplicación web, cuando hablamos de aplicaciones web empresariales además que el acceso no esté restringido por el sistema operativo y que sea de fácil utilización, que tiene un mercado potencial muy alto. Es importante también tener en cuenta los clientes internos, pues si bien, un usuario externo es importante, no se debe dejar de lado los demás.

² <http://www.alegsa.com.ar/Dic/aplicacion%20web.php>

En la actualidad las herramientas web de colaboración entre usuarios, email, entornos de capacitación, ayudas para la selección de personal, entre muchas otras cosas, potencializan el rendimiento de los colaboradores de cualquier organización y aun mas con las facilidades que brinda hoy el internet, estas son cosas que toda empresa debe tener en mente en estos momentos. Además hay otro factor que no se ha considerado y que tiene un gran impacto en los últimos días, los smarth phones y los ipads que tienen ya acceso a la web y que hoy en día han estado excluyendo los equipos portátiles y de escritorio, hoy aun no se puede hacer todo en estos dispositivos, pero mañana se puede estar seguro que ellos no tendrán limitante alguno, pues la web cada día más nos facilita el uso de aplicaciones en la web, por lo que no será necesario equipos con altos requerimientos de hardware para poder trabajar.

La gran inmersión en el mercado de los dispositivos y equipos con la capacidad de conectarse a internet, hace inevitable la incursión de sitios web completos donde los clientes internos y externos a la organización puedan conocer más de ella y trabajar para ella. Entonces podríamos deducir que una aplicación empresarial no es más que una aplicación web, pero con toda la funcionalidad que requiere la administración de la empresa para mantener y optimizar la productividad de la misma.

Arquitectura empresarial: Es necesario tomar los conceptos de procesos y flujos y generar tal como lo haría un arquitecto, un diseño en estructuras y capas para soportar estos procesos desde los niveles estratégicos de negocio hasta los niveles físicos de implementación. El objetivo es definir una forma ordenada de proveer a todos los niveles de la empresa un marco de trabajo definido y claro, un rayado de cancha donde todos los actores de la empresa están considerados, y donde cada nivel participa, centrándose en los procesos y finalmente apoyando las estrategias y metas del negocio.³

La arquitectura empresarial es un esquema de representación de los procesos empresariales y su interacción que al ser implementada permite cantidad de beneficios, algunos de ellos son:

- Conocer el estado ideal al que se desea llegar en la organización y cómo la tecnología va a soportar los procesos clave de negocio necesarios para alcanzar dicho estado.
- Optimizar los procesos de la organización.
- Permite responder rápida, eficaz y positivamente a las oportunidades y desafíos presentados por los cambios de mercado, consolidaciones del sector y avances tecnológicos.⁴

³ <http://www.misbytes.com/wp/2006/09/19/%20arquitecturas-empresariales-bpm-y-soa/>

⁴ <http://www.ubiquando.com.co/index.php/Productos-y-Servicios/arquitectura-empresarial.html>

Arquitectura de software: Conjunto de elementos estáticos propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente como al comportamiento del software en un tiempo de ejecución, naturalmente este diseño arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto.

Contenedor web: Es lo que comúnmente denominamos servidor web. Es la parte visible del servidor de aplicaciones. Utiliza los protocolos HTTP y SSL (seguro) para comunicarse.

Entorno de desarrollo integrado: También llamado IDE por sus siglas en Inglés (Integrated Development Environment), “están formados por el conjunto de instrumentos que facilitan o automatizan las actividades de desarrollo”⁵. En términos generales, un entorno de desarrollo es una herramienta integrada que ofrece facilidades al momento de realizar un desarrollo.

Estilo arquitectónico: Un estilo arquitectónico define una familia de sistemas en términos de un patrón de organización estructural. En particular, de acuerdo a los autores, un estilo arquitectónico define tanto un vocabulario de tipos de componentes y conectores como un conjunto de restricciones sobre cómo combinar esos componentes y conectores.

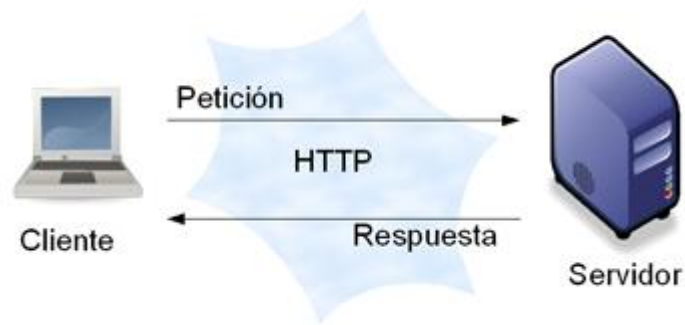
Estilo arquitectural: Los estilos arquitecturales describen las arquitecturas basadas en sus componentes y definen a alto nivel un aspecto de la aplicación.

Los principales aspectos son: **Comunicación, despliegue, dominio, interacción, relación y estructura**. Lo normal en una arquitectura es no basarse en un solo estilo arquitectural, sino que combine varios para obtener las ventajas existentes de cada uno.

- **Cliente/Servidor:** Define una relación entre dos aplicaciones en las cuales una de ellas (cliente) envía peticiones a la otra (servidor y fuente de datos).

Ilustración 1. Estilo arquitectural - Cliente/ Servidor

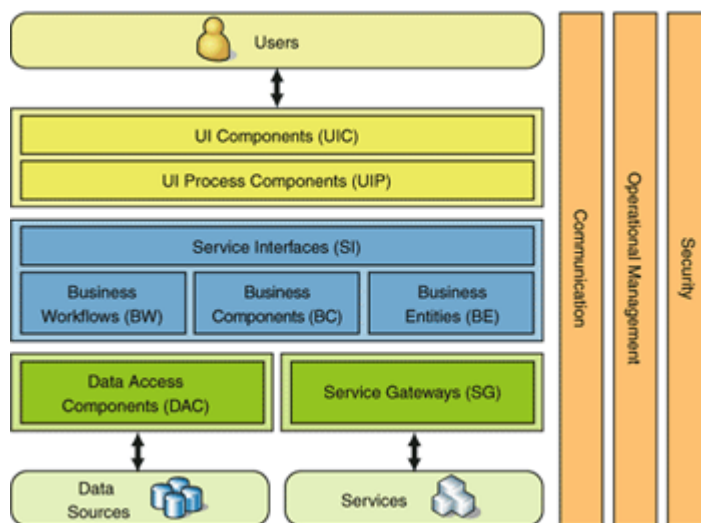
⁵ <http://lml.ls.fi.upm.es/ep/0708/entornos.html>



Fuente: <http://www.josecuellar.net/arquitectura-de-software/estilos-arquitecturales-en-el-diseno-de-un-sistema>

- **Basado en componentes:** Conjunto de componentes que exponen interfaces bien definidas y que colaboran entre sí para resolver el problema. Diseñado de forma que puedan ser reutilizados en distintos escenarios en distintas aplicaciones aunque algunos componentes son diseñados para tareas específicas.

Ilustración 2. Estilo arquitectural - Basado en componentes



Fuente: <http://www.josecuellar.net/arquitectura-de-software/estilos-arquitecturales-en-el-diseno-de-un-sistema>

- **Arquitectura en capas (N-Layer):** Distribución jerárquica de los roles y las responsabilidades para proporcionar una división efectiva de los problemas a resolver. Los roles indican el tipo y forma de interacción con otras capas y las responsabilidades la funcionalidad que implementan.

Ilustración 3. Estilo arquitectural - arquitectura en capas(N-Layer)



Fuente: <http://www.josecuellar.net/arquitectura-de-software/estilos-arquitecturales-en-el-diseno-de-un-sistema>

- **Presentación desacoplada:** Indica cómo debe realizarse el manejo de las acciones del usuario, la manipulación de la interfaz y los datos de la aplicación. Separación de componentes de la interfaz del flujo de datos y de la manipulación.

Ilustración 4. Estilo arquitectural - presentación desacoplada

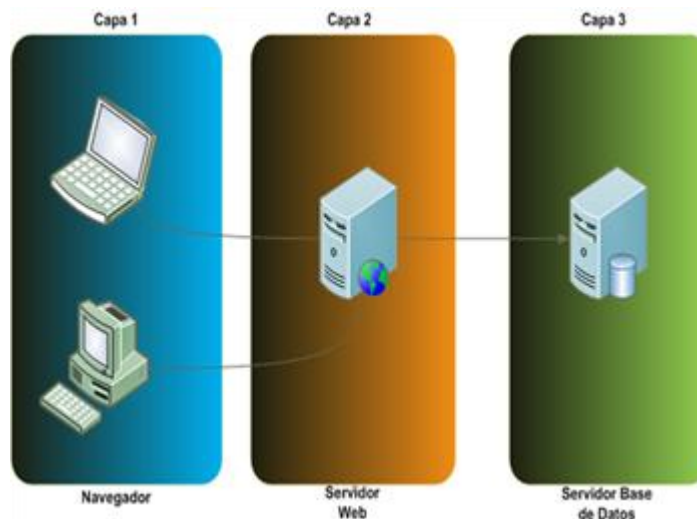


Fuente: <http://www.josecuellar.net/arquitectura-de-software/estilos-arquitecturales-en-el-diseno-de-un-sistema>

- **Arquitectura en capas (N-Tier):** Conceptualmente igual que la arquitectura en capas (n-layer), aunque se define la separación de la

funcionalidad en segmentos físicos separados (Tier). Normalmente la separación física se realiza en servidores diferenciados por razones de escalabilidad, seguridad, o simplemente necesidad.

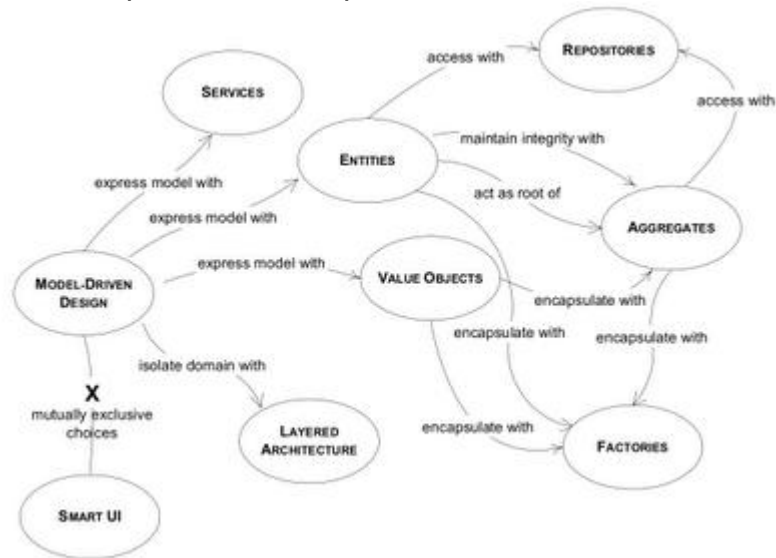
Ilustración 5. Estilo arquitectural - arquitectura en capas (N-tier)



Fuente: <http://www.josecuellar.net/arquitectura-de-software/estilos-arquitecturales-en-el-diseno-de-un-sistema>

- **Arquitectura Orientada al Dominio (DDD):** Orientado para diseñar e implementar aplicaciones empresariales complejas donde es fundamental definir un Modelo de Dominio expresado en el propio lenguaje de los expertos del dominio de negocio real (llamado Lenguaje Ubicuo).
 - Arquitectura N-Layer.
 - Patrones de diseño:
 - Repository
 - Entity
 - Aggregate
 - Value-Object
 - Unit of Work
 - Services
 - Desacoplamiento entre componentes pertenecientes al diseño.
- Todo el equipo de desarrollo deben tener contacto con los expertos del dominio (expertos funcionales) para modelar correctamente el Dominio. El corazón del software es el Modelo del Dominio el cual es una proyección directa de dicho lenguaje acordado entre todos los miembros del equipo (lenguaje Ubicuo).

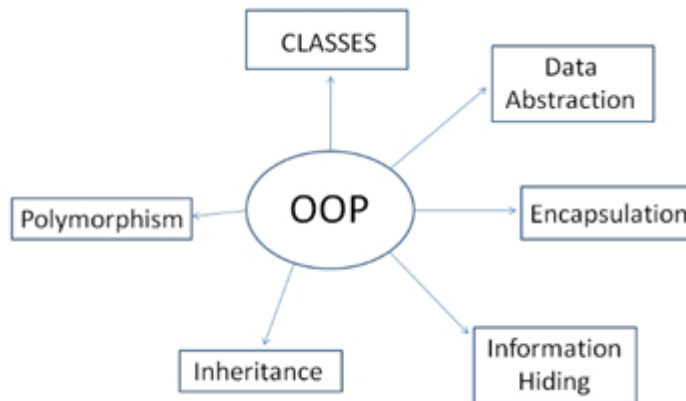
Ilustración 6. Estilo arquitectural - Arquitectura orientada al dominio (DDD)



Fuente: <http://www.josecuellar.net/arquitectura-de-software/estilos-arquitecturales-en-el-diseno-de-un-sistema>

- **Orientado a Objetos (OO):** Conjunto de objetos que cooperan entre sí en lugar de cómo un conjunto de procedimientos. Los objetos son discretos, independientes y poco acoplados, se comunican mediante interfaces y permiten enviar y recibir peticiones.

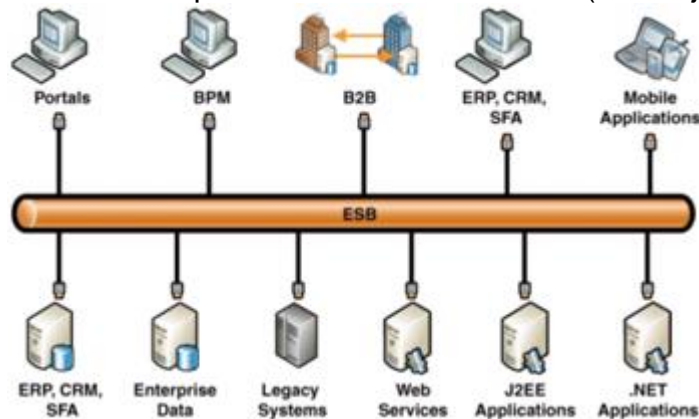
Ilustración 7. Estilo arquitectural - orientado a objetos (OO)



Fuente: <http://www.josecuellar.net/arquitectura-de-software/estilos-arquitecturales-en-el-diseno-de-un-sistema>

- **Bus de Servicios (Mensajes):** Define un sistema de software que puede enviar y recibir mensajes usando uno o más canales de forma que las aplicaciones pueden interactuar sin conocer detalles específicos la una de la otra. Interacción entre aplicaciones a través del paso de mensajes por un canal de comunicación común (bus). Se implementa a menudo usando un sistema de mensajes como MSMQ.

Ilustración 8. Estilo arquitectural - Bus de servicio (Mensajes)



Fuente: <http://www.josecuellar.net/arquitectura-de-software/estilos-arquitecturales-en-el-diseno-de-un-sistema>

Familia tecnológica: La familia tecnológica está formada por un grupo de lenguajes, IDEs, aplicaciones y todos los productos pertenecientes a una misma marca de soluciones. Como la familia de tecnologías java que está integrada por entre otros.

- XML es una familia de tecnologías. El estándar define qué son las etiquetas y los atributos, pero alrededor están creciendo una serie de módulos que proporcionan etiquetas, atributos y guías para realizar tareas específicas:
 - **XPointer**, **XBase** y **XLink** describen un modo estándar de añadir hiperenlaces a un documento XML.
 - **XPath** define la sintaxis para referirse a partes de un documento XML.
 - **CSS**, el lenguaje de estilo de HTML se puede aplicar a XML.
 - **XSL** es el lenguaje de estilo avanzado para XML, y está basado en **XSLT**, que es un lenguaje de transformación que permite reordenar, añadir o eliminar etiquetas y atributos.
 - **DOM** y **SAX** son un conjunto estándar de llamadas a funciones que permiten manipular documentos XML (y HTML) desde programas.
 - **XML Namespaces** describe cómo se puede asociar una URL con cada etiqueta y atributo de un documento XML.
 - **XML Schemas** proporcionan un modo de definir la estructura, el contenido y la semántica de los documentos XML.
 - **XEncryption**, **XSignature** y **XSecurity** tratan los aspectos de seguridad y autenticación en el intercambio de documentos XML.
 - **XQuery** proporciona un modo flexible de consulta para extraer datos de los documentos en el Web.

Frameworks: Se refiere a “ambiente de trabajo, y ejecución”, por ejemplo “.Net” es considerado un “framework” para desarrollar aplicaciones (Aplicaciones sobre Windows). En general los framework son soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución).”⁶ Con componentes que actúan interactivamente con requerimientos de otros componentes u objetos para generar salidas.

Ingeniería de software: “La Ingeniería del Software es una disciplina o área de la informática o ciencias de la computación, que ofrece método y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo. Hoy día es cada vez más frecuente la consideración de la Ingeniería del Software como un nueva área de la ingeniería, y el Ingeniero del Software comienza a ser una profesión implantada en el mundo laboral internacional, con derechos, deberes y responsabilidades que cumplir, junto a una, y reconocida consideración social en el mundo empresarial y, por suerte, para esas personas con brillante futuro.”⁷

Lenguaje de programación: Un lenguaje de programación es una herramienta formada por conjunto de reglas semánticas y sintácticas que definen una técnica estándar de comunicación para darle instrucciones específicas al computador sobre lo que se quiere que sea el producto final o software, cabe aclarar que algunos programas escritos en determinados lenguajes necesitan pasar por el proceso de compilación (traducción del código a un lenguaje que entienda la computadora llamado lenguaje de máquina, este se guarda como ejecutable para uso posterior) o un proceso de interpretación (traducción del código a lenguaje de maquina a medida que se recorre sin guardarlo para uso posterior), para que pueda ser ejecutado por un computador.

Cuando un lenguaje es dependiente de la maquina y muy cercano a la arquitectura hardware es llamado de bajo nivel aquí se encuentra el lenguaje de máquina que es una combinación de 1 y 0 (código binario) y el lenguaje ensamblador que está formado por abreviaturas de letras y números que generaron la creación de los programas traductores para poder llevar estos a lenguaje maquina y cuando se utilizan expresiones más cercanas al lenguaje humano se llaman lenguaje de alto nivel y son independientes de la arquitectura del computador.

El lenguaje de programación se puede clasificar así:

⁶ <http://www.soaagenda.com/journal/articulos/que-son-los-frameworks/>

⁷ <http://www.rodolfoquispe.org/blog/que-es-la-ingenieria-de-software.php>

Lenguajes Declarativos: Aquellos lenguajes donde las instrucciones se dan por medio de expresiones regulares que permiten realizar consultas

Lenguajes imperativos: Aquellos lenguajes donde las instrucciones se entregan por medio de un conjunto de pasos o instrucciones que se ejecutan en forma secuencial.

Lenguaje de programación web: Es un lenguaje de marcas hipertextuales que es usado para crear páginas web en internet, para este tipo de programación se debe conocer el URL identificador de la ubicación de los recursos, el HTTP que será el protocolo encargado de transportar la información por la red y por último el HTML que es lenguaje necesario para representar la información que contiene una página web, Este es el lenguaje de programación web más conocido pero existen otros como JavaScript, DHTML, PHP, ASP, ASP.NET, Python, JSP y Ruby entre los más conocidos.

Plugins: Pequeños módulos de programas que se agregan a otros programas para proporcionar funciones adicionales específicas sin afectar la aplicación original, no son actualizaciones, algunas de las funciones son reproducir ficheros multimedia, visualizar diferentes formatos gráficos.

Programación web: Es el tipo de programación necesaria para crear una página web, para ello se deben conocer los siguientes conceptos:

URL (Uniform Resource Locators), es un sistema que permite ubicar un recurso (página web, servicio, etc) en la red a través del nombre que se le da al mismo indicando el camino del recurso.

HTTP: (Hyper Text Transfer Protocol), es el protocolo de comunicación que se encarga de transportar el contenido de la página web por toda la red de internet.

HTML:(Hyper Text Markup Lenguaje), es el lenguaje que permite representar la información que se encontrara en la página web en forma de lenguaje común, esto se logra conectando dos elementos entre si y colocándoles unas etiquetas o marcas para mostrar cómo se quiere visualizar el documento.

La programación web en su inicio era cargada y tenía interacción con el usuario, ahora es más sencillo, pero es necesario conocer otros lenguajes de programación como JavaScript,

Servidor de aplicaciones: Proporciona servicios que soportan la ejecución y disponibilidad de las aplicaciones desplegadas. Es el corazón de un gran sistema distribuido.

2. DESCRIPCION DE LOS LENGUAJES Y FAMILIAS TECNOLOGICAS QUE SOPORTAN LAS PLATAFORMAS WEB MÁS EXTENDIDAS EXISTENTES ACTUALMENTE

Para la selección de las plataformas más extendidas es necesario conocer los lenguajes de programación web que más auge tienen en este momento, luego, la selección de la plataforma estará basada en si soporta o no los lenguajes de programación más extendidos.

2.1 LENGUAJES QUE SOPORTAN PROGRAMACION WEB

Los lenguajes de programación marcan un paso fundamental al momento de seleccionar la herramienta para desarrollar las aplicaciones web empresariales. De estos dependerá el rendimiento tanto de la aplicación como el rendimiento del programador a la hora de desarrollarla. En algunos casos, con la selección del lenguaje se delimitaran algunas funcionalidades y capacidades de procesamiento. Por ejemplo, imágenes raster, georreferenciación, entre otros.

En este numeral, se incluyen factores como el paradigma de programación del lenguaje, aunque es de anotar que en la actualidad el paradigma orientado a objetos es el que mayor auge tiene con una incidencia en el mercado del 57,6 %, seguido de los lenguajes procedimentales con un 36,1 %⁸; otro aspecto es la flexibilidad que ofrece el lenguaje para hacer todo lo que sea necesario.

Los lenguajes que encontramos disponibles y que son de gran uso hoy en día son lenguajes de alto nivel que ayudan mucho en el desarrollo de una aplicación, claro que no debemos confundir esto con el IDE que se este usando, es muy diferente, pues el IDE lo que hace es facilitarle al programador una serie de herramientas que le ayudarán a manejar el proyecto, a realizar los formularios y le brindaran todo lo necesario para que el programador se pueda concentrar en el problema de programación real.

Ahora desarrollaremos una descripción de lenguajes para desarrollo web que tienen mayor auge en el momento:

2.1.1 JAVA:

⁸ <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

“El lenguaje de programación Java, fue diseñado por la compañía Sun Microsystems Inc, con el propósito de crear un lenguaje que pudiera funcionar en redes computacionales heterogéneas (redes de computadoras formadas por más de un tipo de computadora, ya sean PC, MAC's, estaciones de trabajo, etc.), y que fuera independiente de la plataforma en la que se vaya a ejecutar. Esto significa que un programa de Java puede ejecutarse en cualquier máquina o plataforma”⁹

El gran fuerte del lenguaje java es su portabilidad, pues un programa desarrollado en cualquier sistema operativo, se puede ejecutar en cualquier otra sistema operativo que lo soporte, gracias a la maquina virtual de java JVM.

De acuerdo a lo anterior nos damos cuenta que cuando compila java no se genera un código de máquina, sino que se genera algo que se llama bytecode, que es lo que el JVM interpreta y lo lleva al lenguaje de máquina para ejecutar la instrucción. Esto puede ser un punto en contra para este lenguaje.

Gran parte de la sintaxis de este lenguaje está basada en C y C++, con la diferencia que es un lenguaje de alto nivel y simplifica el modelo de orientación a objetos.

Esta es quizá una de las herramientas más usadas actualmente para desarrollo web, gracias a los diferentes frameworks que tiene que le dan la posibilidad al usuario de hacer lo que se imagine.

2.1.2 RUBY ON RAILS

Este lenguaje aparece en el mercado en 1995, es un lenguaje inspirado en los lenguajes Perl, Smalltalk, Python, Lisp, Eiffel, Scheme, Ada, Dylan y CLU. Es un lenguaje multiparadigma y multiplataforma. Su creador es Yukihiro "Matz" Matsumoto y su nombre obedece a la comparación con Perl (perla).

“Ruby es un lenguaje de código abierto, amigable y simple para los desarrolladores que se ha convertido en el lenguaje de mayor

⁹ <http://www.monografias.com/trabajos/lengprog/lengprog.shtml>

crecimiento de los últimos años y en el elegido por el 80% de las nuevas startups mundiales.”¹⁰

En 2008 Ruby alcanzó el más grande puntaje de su historia, cuando alcanzó un 4% en la calificación de TOIBE, con la posición 8, de ahí en adelante ha venido en decadencia y en los resultados de Febrero de 2012 llegó a la posición 13 con un 1,41%¹¹.

“Un lenguaje de programación dinámico y de código abierto enfocado en la simplicidad y productividad. Su elegante sintaxis se siente natural al leerla y fácil al escribirla.”¹²

Un lenguaje dinámico y de código abierto (libre en todo sentido, gratuito), simple y productivo, eso es lo que cualquier empresario necesita para su organización.

2.1.3 PHP

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.¹³

¹⁰http://www.palermo.edu/ingenieria/eventos/presentacion_del_lenguaje_de_programacion_ruby.html

¹¹ <http://www.tiobe.com/index.php/paperinfo/tpci/Ruby.html>

¹² <http://www.ruby-lang.org/es/>

¹³ <http://www.desarrolloweb.com/articulos/392.php>

2.1.4 ASP .Net

El lenguaje ASP (Active Server Pages), es un lenguaje de programación de servidores para generar páginas Web dinámicamente. Se conocen cuatro versiones de este lenguaje las 1.0, 2.0, 3.0 y la ASP.NET que se la conoce como la ASP Clásica.

El lenguaje de programación ASP nace aproximadamente en el año 1996, lo que ofrecía de nuevo este lenguaje era que se podía crear una página web en la que se pudiese programar para que nos ofreciera unos determinados datos. Esto era una gran ventaja porque en aquella época solo se podía dibujar una tabla e incluir unos pocos datos.

Posteriormente se crea el lenguaje ASP.Net que es un lenguaje mucho más complejo que el original ASP. Este lenguaje nos permite separar en las páginas webs la parte de diseño que contiene la página, no interviniendo para nada el código HTML. Así el trabajo de los diseñadores y programadores es mucho más sencillo. Cada cual se ocupa de su parte del trabajo dentro de la página web sin interferir en la parte de otro.

El ASP es un lenguaje de programación para servidores, es adecuado para acceso a bases de datos, lectura de ficheros, etc. Se vale de dos lenguajes de Script, como son el VBScript y el JavaScript para que lo que programemos con el ASP sea visible.¹⁴

2.1.5 C#

Microsoft ha desarrollado este nuevo lenguaje de programación que combina los mejores elementos de diferentes lenguajes que han tenido acogida por los usuarios.

“C# combina los mejores elementos de múltiples lenguajes de amplia difusión como C++, Java, Visual Basic o Delphi. De hecho, su creador Anders Heljsberg fue también el creador de muchos otros lenguajes y entornos como Turbo Pascal, Delphi o Visual J++. La idea principal detrás del lenguaje es combinar la potencia de lenguajes como C++ con la sencillez de lenguajes como Visual Basic, y que además la migración

¹⁴ <http://www.larevistainformatica.com/ASP.htm>

a este lenguaje por los programadores de C/C++/Java sea lo más inmediata posible.”¹⁵

C# se ha ubicado en el ranking de TOIBE en el tercer (3er) lugar en febrero de 2012 con un 8,24%.¹⁶

2.1.6 COMPARACION

Evaluar un lenguaje de programación para web empresariales puede ser tan complejo como la necesidad que se tenga.

Los lenguajes de programación web empresariales tienen una serie de parámetros por los cuales se puede medir, entre ellos está la dificultad que el mismo presente para el aprendizaje, lo que técnicamente se le dice, curva de aprendizaje del lenguaje. Así como este existe una gran variedad de elementos como son la persistencia, la escalabilidad, la seguridad, la disponibilidad, entre otros. Pero no solo son de rendimiento y por aprendizaje, sino que también se tienen en cuenta otros factores no tecnológicos como que tan popular es.

Si bien es cierto, todo lo que se quiera desarrollar se puede hacer con cualquier herramienta, pero hay cosas que a las empresas les preocupa más como son la seguridad que brinda el lenguaje, la estructuración del mismo, la presentación que se convierte en un factor fundamental.

Conectividad: La conectividad marca uno de los papeles más importantes en la selección de la herramienta. Los lenguajes actuales no tienen restricciones muy marcadas con el acceso a datos, pero si tienen una mejor relación con determinado motor de base de datos, Ejemplo, Java tiene una gran fortaleza en la conexión a bases de datos Oracle y PHP funciona de forma muy integrada con MySQL.

Curva de aprendizaje: Desde el surgimiento, Java se ha destacado por tener una curva de aprendizaje superior a la de los demás lenguajes. Por ejemplo el sencillo programa hola mundo con Ruby se hace en solo una línea, mientras que con Java son mínimo 4.

¹⁵ <http://www.desarrolloweb.com/articulos/561.php>

¹⁶ <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

Por otra parte Java tiene a favor que es un lenguaje que actualmente tiene popularidad y por lo tanto se encuentra mucha documentación y elementos de ayuda para cerrar la brecha de la curva de aprendizaje, pues en la Internet se encuentra solución a la mayoría de los inconvenientes que se presentan.

PHP y Ruby por otra parte se han caracterizado por ser lenguajes muy intuitivos y fáciles de aprender, posibilitando así un desarrollo rápido y en corto tiempo.

Documentación disponible: Sin lugar a dudas, el internet ha revolucionado este factor. Con la posibilidad de acceso a la información en los diferentes blogs y las diferentes comunidades que se han creado en el mundo en calidad de colaboración al desarrollo de la tecnología en los diversos lenguajes y plataformas. Sin embargo aquí ya empieza a jugar un papel importante la calidad de la información que se encuentre en este medio, y por ello se hace más importante aun tener un grupo sólido detrás de esa herramienta que seleccionemos. Es el caso de Java, que tiene a Oracle y Sun Microsystem soportando esta plataforma.

Escalabilidad: En este criterio .Net, Java y Ruby están favorecidos, sin embargo si se hace un diseño y se ejecuta en una arquitectura bien elaborada, cualquier lenguaje soporta la escalabilidad necesaria. En las aplicaciones web este criterio depende más del servidor web, de la habilidad del programador y del diseño que tenga la aplicación.

Otra de las grandes fortalezas de java es su variedad en Frameworks de Persistencia, de control y de Interfaz.

Flexibilidad: La facilidad y la flexibilidad de este lenguaje a hecho que este permanezca dentro de los primeros lenguajes de programación web, esto lo ha logrado gracias a Ruby on Rails, que ha potencializado la creación de páginas web dinámicas fácilmente.

Portabilidad: Todos los lenguajes expuestos tienen la virtud que son multiplataforma en lo que tiene que ver con su ejecución, claro que es una propiedad exclusiva de las aplicaciones web. Por otra parte esta el desarrollo que también la debemos considerar, y ahí pierde puntaje .Net ya que solo es funcional en Windows.

Robustez(solidez): Java ha demostrado ser un lenguaje muy completo, y aun más para empresas con la puesta en el mercado de la plataforma J2EE, que es la versión para desarrollo empresarial que tiene toda la robustez para soportar las necesidades de estas.

Rendimiento: El rendimiento en aplicaciones Web es un punto que hoy en día esta de segunda mano, pues con el crecimiento de la capacidad de los equipos actuales, este criterio lo sacrifican un poco por presentación y disponibilidad.

Respaldo: Java tiene un gran grupo que se encuentra detrás de este lenguaje, su creador es Sun Microsystem, un robusto de la informática que ahora fue absorbido por Oracle, uno de los grandes de las bases de datos. Otro es .Net, respaldado por Microsoft. Los demás están bajo grupo y/o comunidades tecnológicas que son las que los administran. Ese es uno de los inconvenientes que presenta .Net, esta herramienta obliga a tener el servidor web Windows con un servicio exclusivo de ellos, mientras que los otros lenguajes se implementan en cualquier plataforma.

Tipo de licencia: Otro factor determinante es que PHP, Ruby on Rails y Java son grandes y robustos, además de gratuitos, como también sus frameworks y herramientas para desarrollo, mientras que .Net con su plataforma Visual Studio tienen un costo significativo en el mercado. Esto toma fuerza en la era actual debido a que la mayoría de software del mundo está desarrollado en lenguajes y plataformas gratuitas.

Popularidad: Evaluando los lenguajes anteriormente expuestos JAVA, RUBY, PHP, ASP .NET. Tenemos que PHP por ejemplo, es un lenguaje que ha tenido mucho auge en los últimos días, tiene gran desempeño y es fácil de aprender.

De acuerdo al número de líneas de código escritas, Java es el lenguaje más usado en la actualidad según TIOBE (Programming Community Index, indicador de popularidad de lenguajes de programación), el cual se lleva el 17 % seguido de un 16.5 % de C. Esta hace que haya una tendencia a escoger este lenguaje.

Tendencias del mercado: Por las tendencias del mercado Java, PHP y .Net están liderando. Estos lenguajes tienen gran acogida uno por su robustez y los otros por su facilidad.

3. PLATAFORMAS DE DESARROLLO MULTILENGUAJE Y AMBIENTES INTEGRADOS DE DESARROLLO CON SOPORTE WEB

Las herramientas que permiten tener todos los componentes necesarios para el desarrollo de aplicaciones son los Framework, sus beneficios a nivel general son los siguientes:

- Consistencia a través de muchas plataformas
- Mejores condiciones para cumplir estándares y políticas
- Mejor manera de reutilizar componentes
- Empleados de menores habilidades puede efectuar el trabajo de forma eficiente.
- Produce aplicaciones que son más robustas, rápidas, estables y consistentes.

Sin pretender ser exhaustivos en el estudio de los frameworks, se han seleccionado algunos que permiten desarrollar aplicaciones con soporte web y sobre los que se enfoca este capítulo:

- Netbeans
- Eclipse
- JDeveloper
- Visual Studio
- Delphi
- Genexus

3.1 NETBEANS

La plataforma Netbeans es un amplio entorno de desarrollo y ejecución inicialmente desarrollado para el lenguaje de programación java, creado por Sun Microsystems como proyecto de código abierto en Junio del 2000. Este contiene Apis que simplifican el manejo de ventanas, acciones, archivos y muchos otros elementos necesarios para el desarrollo de aplicaciones.

El IDE de Netbeans es una de las tantas aplicaciones basadas en esta plataforma. Un modulo de Netbeans es un grupo de clases de java que proveen una característica específica a una aplicación como lo haría un plugin, esto hace que sus aplicaciones sean fáciles de extender agregando mas módulos.

Netbeans es multiplataforma es decir funciona con todos los sistemas operativos y tiene licencia GNU.

3.2 ECLIPSE

Es una plataforma de desarrollo y ejecución de programas en java, programas en C++ y aplicaciones de cliente enriquecido. También es un IDE de código abierto.

Este se lanzo originalmente en noviembre del 2001 por IBM cuando formo Eclipse Consortium, con el objetivo de desarrollar una plataforma industrial robusta para el desarrollo de herramientas integradas con todas la características y calidad necesarias, que se enfoco en tres proyectos principales : Proyecto eclipse responsable de crear el banco de trabajo (workbench) del IDE de Eclipse, el JDT (herramientas de desarrollo de java) y el PDE (ambiente de desarrollo de plugins) usado para extender la plataforma.

El proyecto de herramientas eclipse debe crear las herramientas para esta plataforma, algunas de ellas son: IDE cobol, IDE C/C++ y la herramienta de modelado EMF. El proyecto de tecnología de Eclipse está enfocado en las investigaciones tecnológicas y la educación utilizando esta plataforma. Eclipse soporta los lenguajes de programación Java, C/C++ y cobol, es multiplataforma y tiene licencia publica de Eclipse.

3.3 JDEVELOPER

Es un entorno de desarrollo y ejecución (IDE) que ofrece características para desarrollo en los lenguajes Java, XML, SQL, HTML, JavaScript, UML y PHP entre otros. Creado por Oracle Corporation es su principal plataforma para desarrollo de sus herramientas, su primera versión se lanzo en el 98 y estaba basada en Jbuilder, de Borland pero a partir del 2001 en su versión 9i está basado en Java, tiene licencia de propietario pero es gratuito, es multiplataforma.

3.4 VISUAL STUDIO

Entorno de desarrollo de aplicaciones para los sistemas operativos de Windows, desarrollado en el 98 por Microsoft, permite desarrollar en los lenguajes Visual C++/C#/J#, ASP. Net y visual basic entre otros. Permite generar aplicaciones Web ASP.NET, servicios web XML y aplicaciones de escritorio y móviles en cualquier entorno que soporte .NET, tiene licencia de software propietario.

3.5 DELPHI

Es un entorno de desarrollo flexible y un lenguaje de programación, también es un intérprete de un lenguaje llamado Object Pascal; permite crear ejecutables binarios para Windows y Linux, y para la plataforma .net . Creado por Borland Software Corporation basado en el lenguaje pascal, Delphi le permite crear programas estándar de Windows, aplicaciones ISAPI (aplicaciones web), Web Services, Servicios de NT, aplicaciones de consola, los DLL, controles ActiveX y los objetos COM, es multiplataforma y tiene licencia de software propietario

3.6 GENEXUS

Herramienta de desarrollo que permite al analista enfocarse en capturar procesos, conocimientos y las necesidades de las personas, dejando que GeneXus se encargue de la programación de bajo nivel.¹⁷ Creado en el 89 por la compañía Artech basado conocimiento y lenguaje declarativo, soporta los lenguajes Java, C#, cobol , RPG, Visual Foxpro y Ruby .tiene licencia de software propietario. Poderosa herramienta para el diseño y desarrollo de software multiplataforma. Permite el desarrollo incremental de aplicaciones críticas de negocio de forma independiente de la plataforma.

Usada para desarrollar complejos sistemas de misión crítica con grandes bases de datos que comprenden desde sistemas centralizados, distribuidos hasta aplicaciones Web. Estos sistemas integran módulos de data warehouse, web services, portales corporativos y mucho más.

3.7 COMPARACION

Todos los IDE y/o Frameworks mencionados adicional a que son entornos muy completos para desarrollo de aplicaciones web cumplen con la característica de utilizar lenguajes enfocados también a desarrollo de aplicaciones web y de ser multiplataforma; Netbeans, Eclipse y JDeveloper pueden ser usados por aquellos que gustan más del software libre y para quienes tienen la posibilidad de pago tienen disponibles Visual Studio, Delphi y Genexus.

El Desarrollo de aplicaciones no limita el uso de herramientas o de framework a uno ni a dos, es posible utilizar varios de ellos en un desarrollo debido a que cada uno está especializado en agregar una característica o componente a la aplicación, Así es necesario utilizar

¹⁷ <http://www.genexus.com/Institucional/fundadores?es>

usar no solo los mencionados sino que se deben agregar otros mas de acuerdo a cada necesidad.

De acuerdo con una mayoría de público que se inclina por el uso de software gratuito estos mismos soportan el lenguaje Java, PHP y C, el lenguaje COBOL aunque poco usado en este momento es soportado por Eclipse y Genexus, y todos manejan arquitectura cliente /servidor y Netbeans y JDeveloper soportan SOA.

La curva de aprendizaje es mínima en los IDE de Netbeans, Eclipse, JDeveloper, Visual studio y Genexus se dice que su manejo es sencillo y muy intuitivo, en cambio es más alta para Delphi, de acuerdo a los aportes de usuarios en foros la mayoría se inclina por Netbeans seguido de Eclipse entre otras por la variedad de lenguajes soportados lo que permite versatilidad, se encuentra que Delphi y Genexus no están incluidos en los IDEs más usados de acuerdo a las diferentes listas de IDEs mas usados.

4. PRINCIPALES PLATAFORMAS UTILIZADAS EN LA EJECUCIÓN DE APLICACIONES EMPRESARIALES WEB

El impacto, transversalidad y popularidad de los sistemas basados en Web, han sido acompañados de una multiplicidad de tecnologías que soportan el desarrollo de aplicaciones empresariales orientadas a Web. En este documento se tratarán las plataformas de desarrollo de aplicaciones empresariales orientadas a la web más conocidas y utilizadas, como lo son J2EE, Corba, Websphere y .NET.

4.1 J2EE

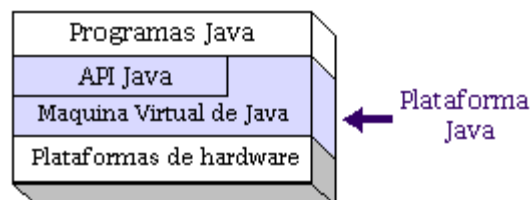
Java como plataforma desde su surgimiento en 1995 ha tenido un crecimiento impensable. La plataforma Java tiene 2 componentes:

La maquina virtual de Java (JVM)

La interfaz de programación de aplicaciones Java (API)

En la siguiente imagen podremos ver los elementos de la plataforma java:

Ilustración 9: Elementos de la plataforma java



Fuente: http://profesores.fi-b.unam.mx/carlos/java/java_basico1_2.html

La plataforma java nos provee una serie de tecnologías las cuales podemos usar de acuerdo a nuestra necesidad. Encontramos por ejemplo Java ME, que es la tecnología especializada para desarrollo de aplicaciones para móviles, hasta J2EE que es la tecnología especializada en desarrollo de aplicaciones empresariales, que es lo que realmente interesa en este trabajo.

Java 2, Edición empresarial, es una plataforma para soluciones empresariales que define el estándar para el desarrollo de aplicaciones empresariales de varios niveles haciéndolas más simples, basándolas en componentes modulares y estandarizados, ofrece cantidad de servicios para estos componentes, y facilita la tarea del programador automatizando detalles de comportamiento de la aplicación.

Componentes:

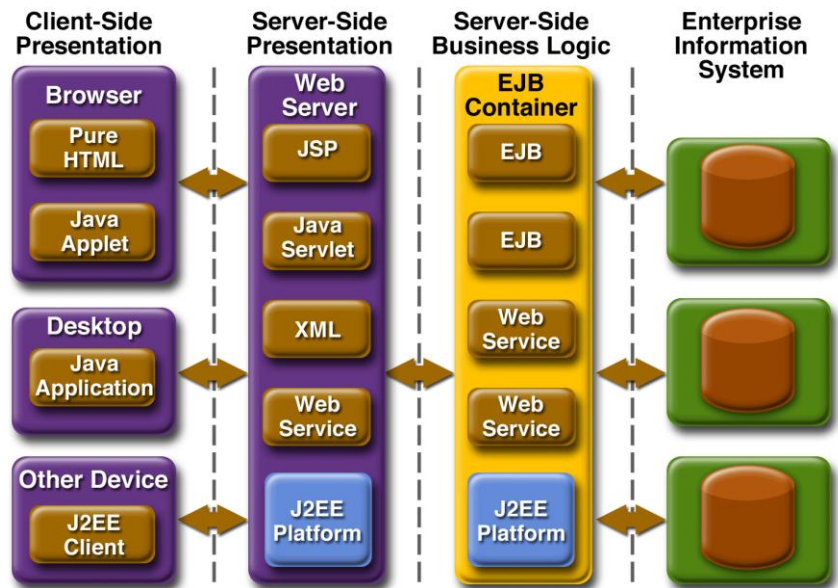
- Este modelo divide las aplicaciones empresariales en tres partes fundamentales:
 - Componentes: Núcleo principal para los desarrolladores
 - Conectores: Están bajo la plataforma definiendo un servicio portable API que se comunica con las ofertas existentes en los proveedores, Promueven la flexibilidad con variedad de implementaciones de servicios específicos, En especial implementan contratos de servicios de mensajería que permiten la comunicación bidireccional entre los componentes de la plataforma y los sistemas empresariales.
 - Contenedores: Interceden ante los clientes y componentes, prestando servicios transparentes a ambos incluyendo soporte de transacción y agrupación de recursos, esto permite implementar el comportamiento de estos en la implementación.

Principales Características

- Utiliza muchas características de la plataforma de Java 2, Edición Estándar (J2SE), como “Escríbelo una vez, córrelo en cualquier parte”, la portabilidad, el API JDBC para la base de datos de Access, la tecnología CORBA para la interacción con los recursos existentes, y un modelo de seguridad que protege los datos incluso en las aplicaciones de internet.
- Agrega soporte completo para las aplicaciones con componentes de JavaBeans, Java Servlets API, Java Server pages y tecnología XML.
- Incluye especificaciones completas y test de cumplimiento para asegurar la portabilidad de la aplicación en todos los sistemas empresariales existentes que soportan esta plataforma.
- Asegura la interoperabilidad de servicios web por medio de la inclusión del perfil WS-I Basic, es decir que se puede construir aplicaciones sobre esta plataforma como servicios web que interactúan con los servicios web desde entornos no compatibles con J2EE.
- Integra la simplicidad, portabilidad, escalabilidad, e integración con sistemas heredados.
- El modelo de la aplicación encapsula las capas de funcionalidad en funcionalidad en tipos específicos de componentes. La lógica de negocio se encapsula en componentes Enterprise Java Beans (EJB). La Interacción

con el cliente puede darse a través de simples páginas web HTML, a través de las páginas web impulsadas por los applets, Java Servlets, tecnología JavaServer Pages, o a través de aplicaciones independientes de Java. Los componentes se comunican de forma transparente usando varios estándares: HTML, XML, HTTP, SSL, RMI, IIOP, entre otros.

Ilustración 10. Modelo de aplicación empresarial en J2EE



Fuente: <http://java.sun.com/j2ee/appmodel.html>

4.2 CORBA

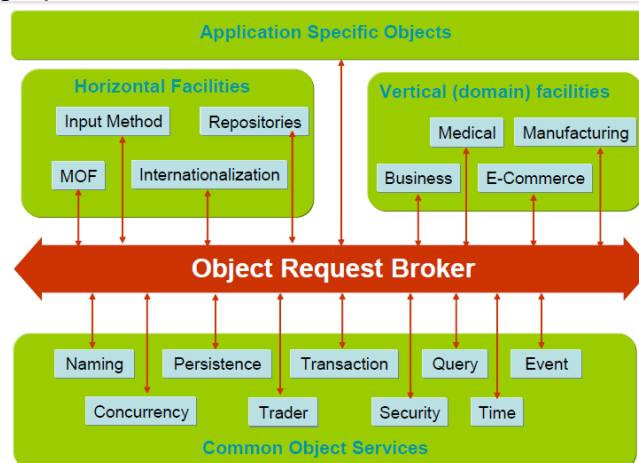
Acrónimo de Common Object Request Broker Architecture, estándar definido por la organización OMG (Object Management Group), este modelo especifica una área para la creación de objetos plug and play basado en un mecanismo intermediario de comunicación: el bróker, fue originalmente creado para crear aplicaciones de gestión y negocio que proporciona un middleware flexible para la integración de aplicaciones.

Componentes

- **Horizontal Facilities**, utilizables como servicios completos en un amplio rango de aplicaciones.
- **Common Object Services**, utilizables como bloques elementales de construcción de aplicaciones ofrecen servicios pre construidos garantizados que simplifican e desarrollo de aplicaciones complejas.

- **Domain Facilities**, que proporcionan componentes reutilizables en un dominio concreto de aplicación (por ejemplo procesos continuo, fabricación discreta, aviónica o sistemas médicos)
- **Application Specific Objects**: Objetos especialmente construidos para una aplicación concreta.
-

Ilustración 11. El modelo OMA(object Management Architecture) definido por la OMG, muestra los componentes agrupados en función del nivel de reutilización de los mismos.



Fuente: Corba: una plataforma software para los sistemas de control del futuro, http://www.isa-spain.org/images/biblioteca_virtual/hrtc.pdf

Características principales

- Se puede integrar con otras tecnologías basadas en objetos como Enterprise Java Beans (EJB).
- Permite la comunicación e interacción de forma transparente entre componentes que pueden ser creados en distintos lenguajes y ejecutados en diferentes plataformas con diferentes sistemas operativos.
- Tiene una arquitectura abierta e independiente enfocada hacia los siguientes objetivos: Orientación a objetos, transparencia de localización, independencia del lenguaje de programación e interoperabilidad.
- Este modelo encapsula la creación, el ciclo de vida y los eventos de los objetos permitiendo indagar todos los detalles de los objetos, esto reduce la curva de aprendizaje.
- Herramienta para la integración de sistemas distribuidos.

4.3 WEBSPHERE

Plataforma modular de integración de Software de IBM que ofrece software de infraestructura middleware para crear e integrar aplicaciones de la nueva generación de negocios e-business como Business to Business (B2B) usando plataformas de red y las tecnologías de la web,

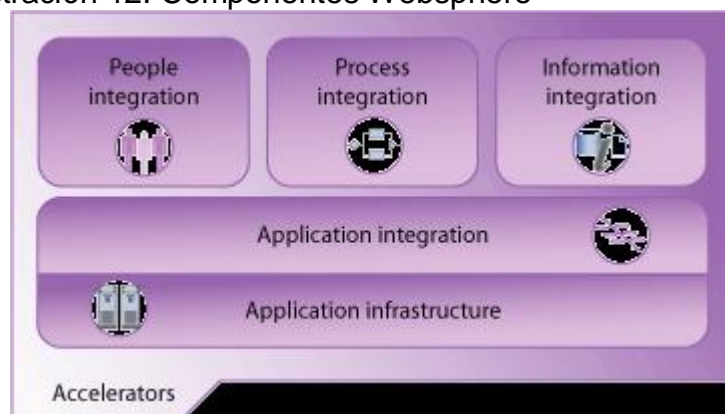
Componentes

- WAS, Runtime o ambiente de ejecución
- Websphere application server, servidor de aplicación sobre el cual funciona todo.
- Websphere Enterprise service bus, es la base para las aplicaciones modulares con arquitectura orientada a SOA
- Websphere Business Modeler, herramienta para el diseño de procesos del negocio
- Websphere Designer, herramienta para integrar el modelo del negocio en las aplicaciones existentes.
- Websphere process server, Websphere Monitor, herramientas para ejecutar y monitorizar los procesos del negocio.
- Aplicaciones de integración y conectividad
- Websphere Commerce, provee interacción directa para hacer negocios con el cliente, o con otras empresas.

Principales características

- Integración, Escalabilidad, Flexibilidad y Compatibilidad
- Se puede usar para desarrollar y monitorear una infraestructura que da soporte a negocios on demand (donde los procesos del negocio están integrados con los socios, proveedores y clientes) .
- Soporta Entornos SOA y los que no lo son.
- Está basada en estándares abiertos soportados por la industria.
- Es soportada por plataformas, como AIX, HP Unix, i5/OS, Linux, Sun Solaris, Windows y z/OS.

Ilustración 12. Componentes Websphere



<http://www.rational.com.ar/webspherespanish/webspheresoftwareplatform.html>

4.4 .NET

Es una plataforma de desarrollo y ejecución que ofrece todas las herramientas y servicios para la creación de aplicación empresariales con una ejecución óptima. Es la evolución de COM (Component Object Model) basada en desarrollo de aplicación de visual basic.

Componentes:

- Runtime o entorno de ejecución, que es el componente que permite ejecutar la aplicación (iniciarla y detenerla) e interactuar con el sistema operativo para proveer a la aplicación servicios y recursos.
- Bibliotecas que ofrecen controles y funciones con componentes listos para usar con otras aplicaciones.
- Lenguajes de programación de alto nivel, junto con compiladores para permitir el desarrollo de aplicaciones en la plataforma.
- Utilitarios y herramientas de desarrollo encargados de simplificar las tareas comunes al programador
- Documentación de las mejores prácticas de diseño, organización, desarrollo, prueba e instalación de las aplicaciones.

Principales Características

- Basada en el paradigma de orientación a objetos
- Ejecución intermedia, Es decir que la ejecución de las aplicaciones se hace por medio del runtime no del sistema operativo.
 - Es multilenguaje, ofrece una lista de lenguajes para codificar los programas no exige un lenguaje específico.
 - Permite la creación y ejecución de aplicaciones empresariales grandes y complejas para la operación de distintos tipos de organizaciones.
 - . Net fue diseñado de manera tal de poder proveer un único modelo de programación, uniforme y consistente, para todo tipo de aplicaciones (ya sean de formularios Windows, de consola, aplicaciones Web, aplicaciones móviles, etc.) y para cualquier dispositivo de hardware (PC's, Pocket PC's, Teléfonos Celulares Inteligentes, también llamados "SmartPhones", Tablet PC's, etc.). Esto representa un gran cambio con respecto a las plataformas anteriores a .NET, las cuales tenían modelos de programación, bibliotecas, lenguajes y herramientas distintas según el tipo de aplicación y el dispositivo de hardware¹⁸

¹⁸ <http://wiki.elhacker.net/programacion/dotnet/introduccion/-que-es-net>

- Hace posible integrar las aplicaciones con otras creadas en al plataforma anterior COM, en plataformas de Microsoft, en otras plataformas de software, sistemas operativos o lenguajes de programación, haciendo uso de estándares globales como XML, HTTP, SOAP, WSDL y UDDI.

Ilustración 13 . Plataforma de ejecución intermedia en .NET



Fuente: <http://wiki.elhacker.net/programacion/dotnet/introduccion/-que-es-net>

4.5 COMPARACION

Las plataformas de desarrollo de aplicaciones empresariales presentan características similares, Todas están orientadas a Objetos, y buscan generar una independencia del lenguaje, plataforma de hardware y sistema operativo, cabe anotar que J2EE solo permite trabajar con lenguaje Java, también poseen aspectos importantes como integración, escalabilidad, flexibilidad y compatibilidad.

Java es quizás la familia tecnológica más seria y completa para el desarrollo de aplicaciones web empresariales.

Es importante que el IDE de la plataforma sea rápido e intuitivo,

5. CRITERIOS DE COMPARACIÓN DE LAS PLATAFORMAS DE DESARROLLO

Para la elección de las plataformas de desarrollo es necesario hacer una revisión exhaustiva de las principales características que las postula como candidatas de evaluación para elegirla.

Se puede hablar que las principales características de las plataformas de desarrollo que se mencionaron en capítulos anteriores son:

Conectividad: Para este criterio se debe tener en cuenta cuales motores de base de datos soporta la plataforma.

Medición: Colocar los diferentes motores de base de datos que este soporta, y en el caso que tenga problemas de conectividad en determinado aspecto indicarlo.

Curva de aprendizaje: Cada lenguaje de programación tiene un nivel de complejidad diferente, lo cual va directamente a afectar el rendimiento inicial de un proyecto, es como decir que aun niño le enseñaremos primero los numero en ingles y luego en español, podría demorarse más que si los aprende primero en español y luego en ingles. En el caso de un programador el cambio de PHP a Java podría significar un tiempo de aprendizaje alto, y el rendimiento se vería notablemente afectado hasta que la persona obtenga ese aprendizaje necesario para tener el mismo rendimiento.

Documentación disponible: En el desarrollo de sistemas de información es vital que la herramienta en la que se esté trabajando tenga información disponible en la cual se puedan resolver dudas de programación en un momento determinado.

Escalabilidad: Es la propiedad de un sistema para agregar características a sí mismo, para incrementar sus capacidades de trabajo sin que tenga que rediseñarse y simplemente aprovecha las adiciones hechas.

Estabilidad: Propiedad de un sistema de mantenerse trabajando correctamente cuando su nivel de fallos disminuye por debajo de un determinado umbral de la plataforma, que varía dependiendo de la estabilidad que se requiera, es una prioridad de los sistemas.

Flexibilidad: Posibilidad de desarrollar una parte del sistema según unos requerimientos y luego más adelante agregarle nuevas funcionalidades.

Portabilidad: Portabilidad se refiere a la capacidad muy particular en algunos programas, que es poder ejecutarse independiente de la plataforma en la que estemos trabajando, así poder “librarse” un poco de la dependencia del

hardware en el que estemos trabajando. Todo esto se hace sin que se hagan mayores modificaciones en el código del programa. Este programa para poder ejecutarse en cualquier plataforma, debe cumplir con un prerrequisito, que es la abstracción generalizada entre la aplicación lógica y las interfaces del sistema.

Robustez (solidez): Es la característica de un sistema que puede ejecutar diversos procesos de manera simultánea sin generar fallos o bloquearse (colgarse). Se refiere a la capacidad del software de defenderse de las acciones anormales que llevan al sistema a un estado no deseado o por lo menos no previsto, causando un comportamiento inesperado, indeseado y posiblemente erróneo.

Rendimiento: Expresa la manera o la eficiencia con que un sistema cumple sus metas, puede hablarse de números de trabajos atendidos por unidad de tiempo, tiempo de respuesta, tiempo de reacción, capacidad de ejecución.

Respaldo: Es necesario evaluar que empresa, empresas, grupos de investigación o entidades están soportando la plataforma en la cual se desea trabajar, en aras de tener un poco de confianza en el producto que se va a usar y de tener la seguridad que esta herramienta no va ser discontinuada en el corto tiempo, provocando otro retraso por tener que escoger una nueva herramienta en la cual se debe entrenar el equipo y desarrollar habilidades.

Tipo de licencia: Este tema es importante revisarlo con lupa, pues tiene disposiciones legales que pueden acarrear millonarias multas. Por lo tanto se presenta en diferentes definiciones así:

Licencia: contrato entre el desarrollador de un software sometido a propiedad intelectual y a derechos de autor y el usuario, en el cual se definen con precisión los derechos y deberes de ambas partes. Es el desarrollador, o aquél a quien éste haya cedido los derechos de explotación, quien elige la licencia según la cual distribuye el software [2].

Patente: conjunto de derechos exclusivos garantizados por un gobierno o autoridad al inventor de un nuevo producto (material o inmaterial) susceptible de ser explotado industrialmente para el bien del solicitante por un periodo de tiempo limitado [6].

Derecho de autor o *copyright*: forma de protección proporcionada por las leyes vigentes en la mayoría de los países para los autores de obras originales incluyendo obras literarias, dramáticas, musicales, artísticas e intelectuales, tanto publicadas como pendientes de publicar [6].

Software libre [3]: proporciona la libertad de:

- Ejecutar el programa, para cualquier propósito;
- Estudiar el funcionamiento del programa, y adaptarlo a sus necesidades;
- Redistribuir copias;
- Mejorar el programa, y poner sus mejoras a disposición del público, para beneficio de toda la comunidad.

Software de fuente abierta [2]. Sus términos de distribución cumplen los criterios de:

- Distribución libre;
- Inclusión del código fuente;
- Permitir modificaciones y trabajos derivados en las mismas condiciones que el software original;
- Integridad del código fuente del autor, pudiendo requerir que los trabajos derivados tengan distinto nombre o versión;
- No discriminación a personas o grupos;
- Sin uso restringido a campo de actividad;
- Los derechos otorgados a un programa serán válidos para todo el software redistribuido sin imponer condiciones complementarias;
- La licencia no debe ser específica para un producto determinado;
- La licencia no debe poner restricciones a otro producto que se distribuya junto con el software licenciado;
- La licencia debe ser tecnológicamente neutral.

Estándar abierto: según Bruce Perens, el basado en los principios de

- Disponibilidad;
- Maximizar las opciones del usuario final;
- Sin tasas sobre la implementación;
- Sin discriminación de implementador;
- Permiso de extensión o restricción;
- Evitar prácticas predatorias por fabricantes dominantes [4].

Software de dominio público: aquél que no está protegido con *copyright* [1].

Software con *copyleft*. software libre cuyos términos de distribución no permiten a los redistribuidores agregar ninguna restricción adicional cuando lo redistribuyen o modifican, o sea, la versión modificada debe ser también libre [1].

Software semi libre: aquél que no es libre, pero viene con autorización de usar, copiar, distribuir y modificar para particulares sin fines de lucro [1].

Freeware: se usa comúnmente para programas que permiten la redistribución pero no la modificación (y su código fuente no está disponible) [1].

Shareware: software con autorización de redistribuir copias, pero debe pagarse cargo por licencia de uso continuado.

Software privativo: aquél cuyo uso, redistribución o modificación están prohibidos o necesitan una autorización.

Software comercial: el desarrollado por una empresa que pretende ganar dinero por su uso. ¹⁹

A continuación se muestra una tabla de sugerencia para la evaluación de los diferentes criterios en cada plataforma de acuerdo al lenguaje que soporta, teniendo en cuenta que solo se están mencionando los lenguajes más relevantes para desarrollo de aplicaciones web, la calificación se debe dar de acuerdo a la experiencia y a la información disponible en el momento del estudio, de acuerdo a los factores de medición explicados en cada criterio.

Estos criterios deben ser calificados utilizando una escala de 0 a 10, donde 0 representa el mínimo valor y este criterio no tienen ningún peso para la evaluación y el 10 representa el máximo valor que puede obtener, con lo que se considera que es muy bueno el desempeño de esta característica dentro del lenguaje o plataforma, al final se puede totalizar y este valor puede representar una idea en forma muy general cual plataforma se ajusta mas a las necesidades de la empresa y ser una base solida para la toma de la decisión.

¹⁹ <http://www.informatica.us.es/~ramon/articulos/LicenciasSoftware.pdf>

PLATAFORMA/LENGUAJES/ CRITERIOS	NETBEANS		ECLIPSE	JDEVELOPER		VISUAL STUDIO		DELPHI	GENEXUS	
	JAVA	PHP	JAVA	JAVA	PHP	ASP.NET	C#	DELPHI	JAVA	RUBY
CONECTIVIDAD										
CURVA DE APRENDIZAJE										
DOCUMENTACIÓN DISPONIBLE										
ESCALABILIDAD										
FLEXIBILIDAD										
PORTABILIDAD										
ROBUSTEZ										
RENDIMIENTO										
RESPALDO										
TIPO DE LICENCIA										
TOTALES										

Tabla 1: Tabla para evaluación de los criterios de acuerdo a cada plataforma y lenguaje.

6. CONCLUSIONES

Siendo objetivo al momento de seleccionar una herramienta en la cual deseo desarrollar las aplicaciones web para la organización, debería estar dado por la conveniencia y necesidad de la organización, y no por el capricho del programador.

Otro punto que es importante tener en cuenta, es que las empresas de hoy deberían concentrarse en tener bien definidos los procesos, de la esencia, de su razón de ser y dejarle este tipo de decisiones a las empresas dedicadas a ello, así no tendrá que preocuparse por si tiene el personal suficientemente capacitado, cuánto tiempo se demoraran sus ingenieros en aprender el lenguaje, si este es robusto, en fin, solo se preocupa por asegurarse de que el proveedor sea bueno y exigirle aplicaciones con calidad, seguridad y estabilidad.

Los criterios sugeridos son una selección entre muchos que se deben tener presentes y que pueden variar de acuerdo a la experiencia de cada gerente de tecnología y sus ingenieros de desarrollo. No significa esto que sean los únicos que se deban tener en cuenta, ya se sabe que en informática y tecnología todo puede variar.

Las herramientas para la creación de aplicaciones empresariales para ambientes web se encuentran en grandes cantidades en el mercado actual, la selección de las mismas depende no solo del resultado de la evaluación de sus características especiales, sino también de las necesidades específicas del desarrollo recopiladas en los requisitos del cliente e incluso, de las inclinaciones personales del desarrollador encargado del proyecto o hasta de la gerencia, por una determinada herramienta o lenguaje.

Se hace necesaria la presencia de personal con conocimiento claro e indispensablemente basado en la experiencia para realizar la calificación de los criterios, sería adecuado consultar usuarios que hayan utilizado más de una plataforma para que compartan sus puntos de vista y ponderen cada una con la mayor objetividad posible.

Es importante el estudio y análisis comparativo de los aspectos más relevantes para el buen desempeño de la aplicación, teniendo claro los lenguajes soportados por cada uno, los IDE que soporta, el acceso a los datos, los servidores de aplicación que usan, los estándares que manejan y la interoperabilidad que permita interconectar sistemas distribuidos en múltiples plataformas de software y arquitecturas de hardware.

Como resultado de la evaluación de los criterios puede obtenerse que la mejor solución no es una sola herramienta o plataforma, puede ser que a la organización le convenga tener un híbrido de varias de ellas para satisfacer sus necesidades.

BIBLIOGRAFÍA

BARROS V, Oscar, La ingeniería de Negocios y Enterprise architecture, 2006, 56 p. trabajo, Universidad de Chile. Facultad de ciencias físicas y matemáticas.

CAMACHO, Érica, “Arquitecturas de Software”. Internet: (<http://prof.Usb.ve/Imendoza/Documentos/PS-6116/Guia%20Arquitectura%20v.2.Pdf>)

CUELLAR, José, “Estilos arquitecturales en el diseño de un sistema”. Internet (<http://www.josecuellar.net/arquitectura-de-software/estilos-arquitecturales-en-el-diseno-de-un-sistema>)

CUESTA, Carlos E, “Arquitectura de Software”. Internet: (http://www.Kybele.etsii.urjc.es/docencia/DAS_GIS_M/2011-2012/Material/Tema%201%20-%20Arquitectura%20de%20Software.pdf)

DAVILA SANTOS, Rubén, “Introducción a Ruby on Rails ”. Internet: (<http://www.maestrosdelweb.com/editorial/rubyonrails/>)

GARLAN David, SHAW Mary, “An Introduction to Software Architecture”. Internet: (www.bredemeyer.com/pdf_files/ArchitectureIntroduction.PDF)

MENDEZ ROMERO, Jhon Jairo, Análisis comparativo de las plataformas J2EE y .NET aplicado al desarrollo de servicios web. Pamplona, 2008. 201 p. Trabajo de Grado, Ingeniero de sistemas. Universidad de Pamplona. Facultad de Ingenierías y Arquitectura.

MOLPECERES TOURIS, Alberto,” Arquitectura empresarial y software libre, J2EE”. Internet: (<http://ccia.ei.uvigo.es/docencia/SCS/articulo-j2ee.pdf>)

NEAT, Adams, “Java JamsM Five IDEs test”. Internet: (<http://www.builderau.com.au/program/java/soa/Java-jams-five-IDEs-tested/0,339024620,339174040,00.htm>)

LUJAN MORA, Sergio, Programación de aplicaciones web: historia, principios básicos y clientes web. Alicante, 2002. 65 p.

PEREZ VALDES, Damián, “Los diferentes lenguajes de programación para web” . Internet: (<http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>)

RENE, Rene, "Eclipse I- Historia y toma de contacto". Internet:
(http://www.programacion.com/articulo/eclipse_-_i_-_historia_y_toma_de_contacto_288)

REYNOSO, Carlos Billy, "Introducción a la arquitectura del Software". Internet:
(<http://carlosreynoso.com.ar/archivos/arquitectura/Introduccion.PDF>)

REYNOSO, Carlos, KICCILLOF, Nicolas, "Estilos y Patrones en la Estrategia de Arquitectura de Microsoft". Internet:
(<http://cic.puj.edu.co/wiki/lib/exe/fetch.php?media=materias:estiloypatron.pdf>)

URRUTIA, Javier, "Arquitecturas Empresariales (BMP y SOA)",
Internet:(<http://www.alegsa.com.ar/Dic/aplicacion%20web.php>)

VALENCIA, Alejandra. FERRO GONZALES, Mauricio. Documentación Y Análisis Crítico De Algunas Arquitecturas De Software En Aplicaciones Empresariales. Pereira, 2011. 91 p. Monografía de Investigación, ingeniero de sistemas. Universidad Tecnológica de Pereira. Facultad de ingenierías. Ingeniería de Sistemas.

8 LENGUAJES DE PROGRAMACIÓN QUE DEBERÍAS APRENDER
<http://www.tufuncion.com/diferentes-lenguajes-programacion>

ACTUALIDAD DE LA EMPRESA, GoLive apuesta por las aplicaciones web en el mundo de la empresa
<http://www.onegolive.com/noticias-las-apliaciones-web-en-el-mundo-de-empresarial.aspx>

ARQUITECTURA EMPRESARIAL, Derribe las barreras para cambiar: haga de su arquitectura empresarial un vehículo para la innovación
<http://www-01.ibm.com/software/co/info/itsolutions/enterprisearchitecture>

COMUNIDAD RUBY, A Programmer's Best Friend.
<http://www.ruby-lang.org/es/about/>

DICCIONARIO DE INFORMATICA, Definición de aplicación web.
<http://www.alegsa.com.ar/Dic/aplicacion%20web.php>

OTHER ARCHITECTURES AND FRAMEWORKS
<http://pubs.opengroup.org/architecture/togaf8-doc/arch/chap37.html>

SOLUCION JAVA Aplicaciones empresariales
http://www.solucionjava.com/apl_emp.php