**Trabajo Interciclo**

Carlos Castillo,Dario Molina, Edison Morocho & Andres Yascaribay.

Mayo 2019.

Universidad Politécnica Salesiana.

Ingeniería en Sistemas.

Aplicaciones Distribuidas. **Abstract**

En el siguiente documento analizaremos la plataforma Java Enterprise Edition, las ventajas y desventajas de su uso y la relación que mantiene con la arquitectura dividida en capas.

**Tabla de Contenidos**

[**Capítulo 1  
Introducción e información general**](#_wrwpc1vh12do) **5**

[Introducción:](#_30j0zll) 5

[Especificaciones](#_2et92p0) 5

[Ventajas](#_erhn88wgw5lu) 5

[Desventajas​​:](#_58pvocxt2jvo) 6

[Modelo:](#_jd1pl5xlk1ks) 7

[Relación De La Arquitectura Jee Con La Arquitectura N-capas](#_lhfmv4yampk3) 8

[Relación que existe entre Java Enterprise Edition y la arquitectura n-capas](#_1ksv4uv) **11**

[**Lista de referencias**](#_irxkq9cc4ge7) **12**

**Lista de figuras**

# 

[**Capítulo 1  
Introducción e información general**](#_wrwpc1vh12do) **5**

[**Introducción:**](#_30j0zll) **5**

[**Especificaciones**](#_2et92p0) **5**

[**Ventajas**](#_erhn88wgw5lu) **5**

[**Desventajas​​:**](#_58pvocxt2jvo) **6**

[**Modelo:**](#_jd1pl5xlk1ks) **7**

[**Relación De La Arquitectura Jee Con La Arquitectura N-capas**](#_lhfmv4yampk3) **8**

[**Relación que existe entre Java Enterprise Edition y la arquitectura n-capas**](#_1ksv4uv) **11**

[**Lista de referencias**](#_irxkq9cc4ge7) **12**

# 

# 

# 

# 

# 

# 

# Capítulo 1 Introducción e información general

## Introducción:

“Java Platform, Enterprise Edition (Java EE) es el estándar en software empresarial impulsado por la comunidad. Java EE se desarrolla utilizando e​l ​proceso de la comunidad Java , con contribuciones de expertos de la industria, organizaciones comerciales y de código abierto, grupos de usuarios de Java e innumerables personas.” (Mendez, 2016)

Progresivamente conforme pasa el tiempo va adoptando nuevas características logrando adaptarse a las necesidades del mercado y permitiendo mejorar la portabilidad de las aplicaciones solidificando de mejor manera la productividad del desarrollador.

## Especificaciones

Java EE contiene algunas especificaciones de API (Interfaz de Programación de Aplicaciones), unas de ellas son JDBC, RMI, e-mail, JMS, Servicios Web, XML, etc. y precisa cómo ordenarlos. Java EE tiene especificaciones únicas para Java EE para componentes. Estas incluyen Enterprise JavaBeans, servlets, portlets; esto con el fin de poder crear aplicaciones de modo que estas sean portables entre plataformas y escalables, y que de este modo sea posible combinarlas con tecnologías anteriores.

## Ventajas

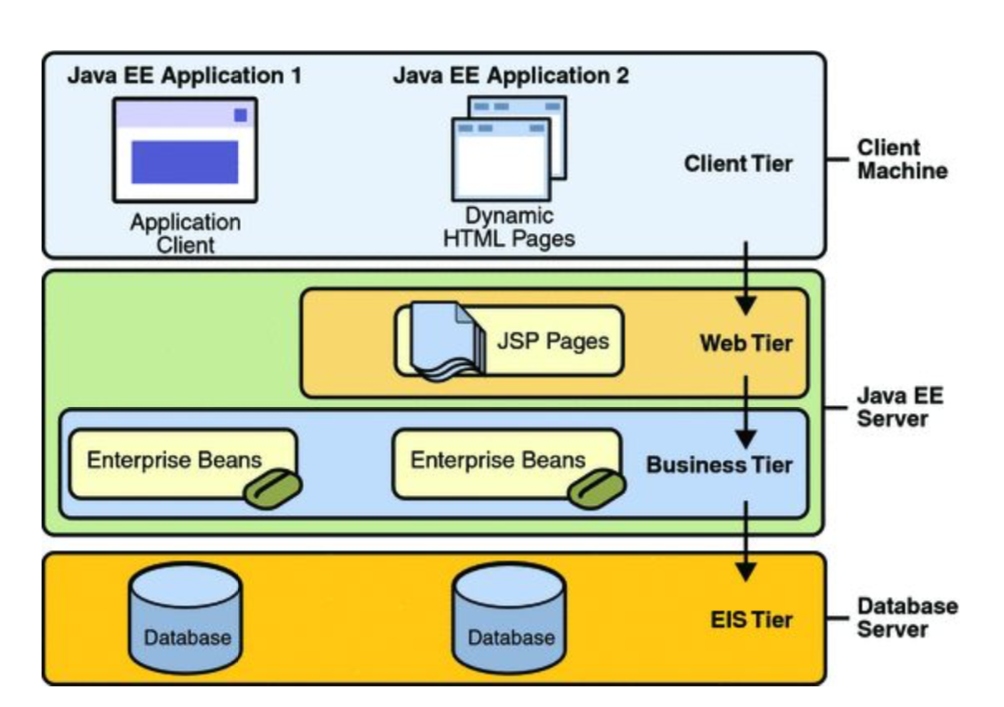
* ● Las implementaciones de J2EE pueden adquirirse a distintas compañías.
* ● Debido al proceso evolutivo de los productos de Microsoft, y en muchos casos, por motivos de compatibilidad la seguridad frente a virus informáticos de los productos de Microsoft es menor que los basados en Java, pues desde un comienzo Java se fundamentó en un estricto modelo de seguridad.
* ● Las aplicaciones Java pueden correr en una amplia gama de sistemas operativos (desde sistemas empresariales como Windows 2000, OS/390, Solaris, HP-UX, IRIX u otras versiones de Unix hasta en sistemas orientados más a ordenadores personales como Mac OS, Windows 9x ó Linux ,y en sistemas operativos para dispositivos móviles) y de arquitecturas hardware.
* ● La tecnología Java es una tecnología abierta (en el sentido de que el código de la plataforma completa puede ser obtenido, revisado y estudiado por cualquiera que esté interesado).
* ● Aunque Java fue creado originalmente por una compañía: Sun MicroSystems, lo cierto es que J2EE es ahora el producto de la colaboración de más de 400 empresas y organizaciones de todo tipo (público, privado sin ánimo de lucro, privado con ánimo de lucro, y de normalización en ámbitos nacionales e internacionales).

## Desventajas​​:

* ● J2EE sólo trabaja con el lenguaje de programación de java.
* ● Las herramientas de desarrollo de implementa J2EE son más complicadas y poco intuitivas que las que utiliza .NET.
* ● La plataforma de J2EE no se ha diseñado pensando en los servicios Web.
* Ahora viendo las ventajas, que son mucho mayores que las desventajas, es posible darse cuenta que J2EE es una plataforma guiada en aplicaciones bastante fiable y segura.

## Modelo:

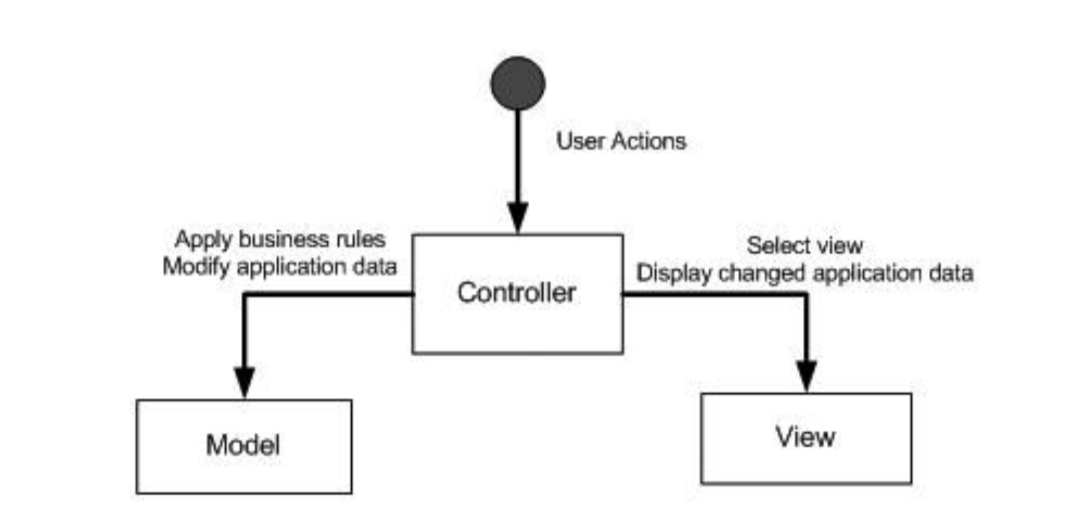
El modelo clásico de capas en la arquitectura de las aplicaciones Java EE se divide en las siguientes:

* ● Cliente: normalmente se trata de un navegador pero puede ser un teléfono inteligente o una computadora de escritorio, incluso otra aplicación. Es la que presenta la información al usuario.
* ● Capa web: se comunica con el cliente y la capa de negocio. Obtiene los datos y los transforman al formato adecuado al cliente generalmente HTML o JSON.
* ● Capa de negocio: proporciona y persiste los datos de la capa cliente y contiene la lógica de negocio de la aplicación. Se ejecuta en un servidor de aplicaciones o contenedor de servlets.Sistemas de información: donde se persisten los datos de la aplicación, puede ser una base de datos relacional como Oracle, MySQL o PostgreSQL o una base de datos NoSQL como Redis o MongoDB u otros sistemas como Elasticsearch.
* 
* *“Figura 1”.* Java y su distribución en capas.

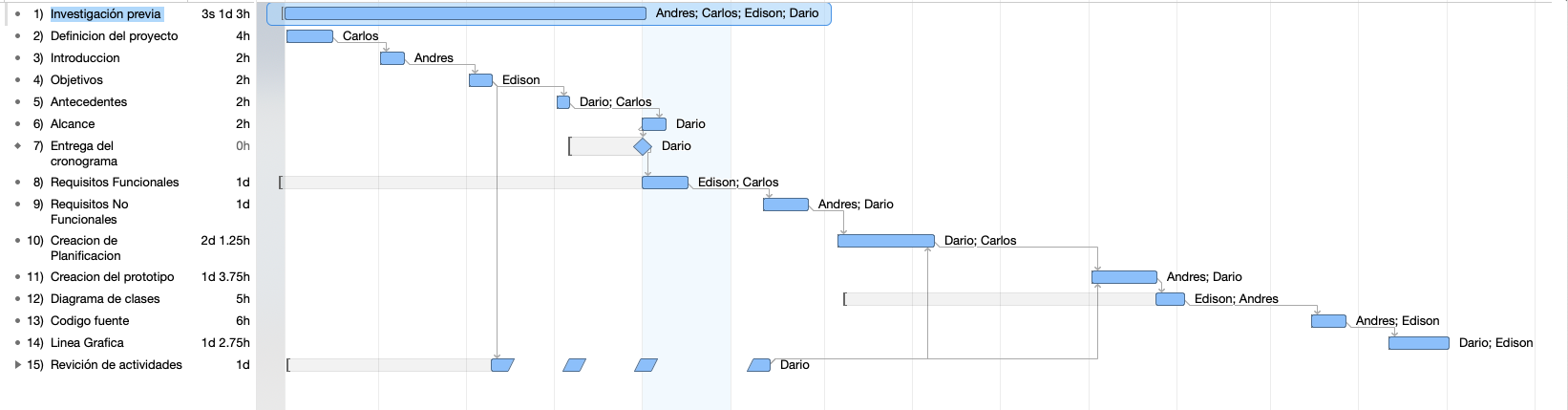
## Relación De La Arquitectura Jee Con La Arquitectura N-capas

El desarrollo de aplicaciones distribuidas de n niveles es un trabajo complejo y desafiante. La distribución del procesamiento en niveles separados conduce a una mejor utilización de los recursos. También permite la asignación de tareas a expertos que están mejor capacitados para trabajar y desarrollar un nivel particular. Los diseñadores de páginas web, por ejemplo, están más equipados para trabajar con la capa de presentación en el servidor web. Los desarrolladores de bases de datos, por otro lado, pueden concentrarse en desarrollar procedimientos y funciones almacenados. Sin embargo, mantener estos niveles como silos aislados no tiene ningún propósito útil. Deben estar integrados para lograr un objetivo empresarial más grande. Es imperativo que esto se haga aprovechando el protocolo más eficiente; de lo contrario, esto conduce a una grave degradación del rendimiento.  
 Además de la integración, una aplicación distribuida requiere varios servicios. Debe poder crear, participar o administrar transacciones mientras interactúa con sistemas de información dispares. Esto es una necesidad absoluta para garantizar la concurrencia de los datos de la empresa. Dado que se accede a las aplicaciones de nivel n a través de Internet, es imperativo que estén respaldadas por servicios de seguridad sólidos para evitar el acceso malicioso.

Los eventos desencadenados por cualquier acción del usuario son interceptados por el controlador. Dependiendo de la acción, el controlador invoca el modelo para aplicar reglas comerciales adecuadas que modifican los datos de la aplicación. El controlador selecciona un componente de vista para presentar los datos de la aplicación modificados al usuario final. Por lo tanto, verá que MVC proporciona pautas para una separación clara de responsabilidades en una aplicación. Debido a esta separación, varias vistas y controladores pueden trabajar con el mismo.

* *“Figura 2”.* Descripción del funcionamiento del controlador cuando el usuario genera una acción.**Capítulo 2  
  Cronograma.**

Las diferentes tareas han sido propuestas para que cada integrante pueda tener un control de

****

* *“Figura 3”. Arquitectura* JAVA EE.

# Relación que existe entre Java Enterprise Edition y la arquitectura n-capas

Según lo que se ha investigado para el presente trabajo se ha visto una clara relación que existe entre Java EE y la arquitectura N-capas ya que Java EE para permitir que se tenga las ventajas antes mencionadas, divide en 4 capas sus componentes, estas capas son cliente, web , negocio y datos las cuales ya se explicaron con anterioridad en este documento, la arquitectura vista en clase de N-capas divide a la aplicación así mismo en componentes pero a diferencia de Java EE esta divida en 3 componentes usuario, negocios y datos. Pero como podemos observar cada una de estas capas tienen la misma finalidad tanto en Java EE como en la arquitectura N-capas.

Tabla 1

*Relacion N-Capas y JEE*

|  |  |
| --- | --- |
| **N-Capas** | **Java EE** |
| **Cliente** | **Usuario** |
| **Web** | **Negocio** |
| **Negocio** |
| **Datos** | **Datos** |

Podemos observar cada una de estas capas tienen la misma finalidad tanto en Java EE como en la arquitectura N-capas

# Lista de referencias

Méndez,J (2016) Java™ EE at a Glance. California, E.E.U.U.

[Viklund](http://andreasviklund.com/), A. (2006). Java y su distribución en capas. [Figura 1]. Recuperado de <http://www.jtech.ua.es/j2ee/2006-2007/jee.html>

Kayal,D.(2009). Descripción del funcionamiento del controlador cuando el usuario genera una acción. [Figura 2]. Recuperado de <https://www.developer.com/design/article.php/10925_3808106_4/Introducing-Enterprise-Java-Application-Architecture-and-Design.htm>

Barrios.J (2003).Arquitectura JAVA EE.. [Figura 3].Recuperado de <https://users.dcc.uchile.cl/~jbarrios/J2EE/node14.html>

# 

# 