

# **Una Introducción al UML**

## **El Modelo Físico**

Autor: Geoffrey Sparks, Sparx Systems, Australia

Traducción: Fernando Pincioli (Solus S.A., Argentina) y Aleksandar Orlic (Craftware Consultores Ltda., Chile)

[www.sparxsystems.com.ar](http://www.sparxsystems.com.ar) - [www.sparxsystems.cl](http://www.sparxsystems.cl)

## Tabla de Contenidos

<b>TABLA DE CONTENIDOS</b> .....	<b>2</b>
<b>EL MODELO FÍSICO</b> .....	<b>3</b>
INTRODUCCIÓN AL UML .....	3
EL MODELO DE DESPLIEGUE .....	3
INSTANCIAS DE NODOS .....	4
LOS COMPONENTES .....	5
LA COMUNICACIÓN .....	6
LAS DEPENDENCIAS .....	7
EL MODELADO DE NEGOCIOS Y LOS DIAGRAMAS DE IMPLEMENTACIÓN.....	7
LECTURA RECOMENDADA .....	9

## El Modelo Físico

El modelo físico en UML describe los componentes, de hardware y de software, que se desplegarán en el ambiente seleccionado. Describe elementos tales como plataformas de hardware, denominadas **nodos** en UML, conectividad de redes, componentes de software, procesadores, sistemas operativos y herramientas de terceras partes.

Los diagramas de despliegue son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema. Este artículo ilustrará la notación que se emplea en los diagramas de despliegue y algunos ejemplos de cómo se utilizan.

### *Introducción al UML*

El Lenguaje Unificado de Modelado (UML) es, tal como su nombre lo indica, un lenguaje de modelado y no un método o un proceso. El UML está compuesto por una notación muy específica y por las reglas semánticas relacionadas para la construcción de sistemas de software. El UML en sí mismo no prescribe ni aconseja cómo usar esta notación en el proceso de desarrollo o como parte de una metodología de diseño orientada a objetos.

El UML soporta un conjunto rico en elementos de notación gráficos. Describe la notación para clases, componentes, nodos, actividades, flujos de trabajo, casos de uso, objetos, estados y cómo modelar la relación entre esos elementos. El UML también soporta la idea de extensiones personalizadas a través de elementos estereotipados.

El UML provee beneficios significativos para los ingenieros de software y las organizaciones al ayudarles a construir modelos rigurosos, trazables y mantenibles, que soporten el ciclo de vida de desarrollo de software completo.

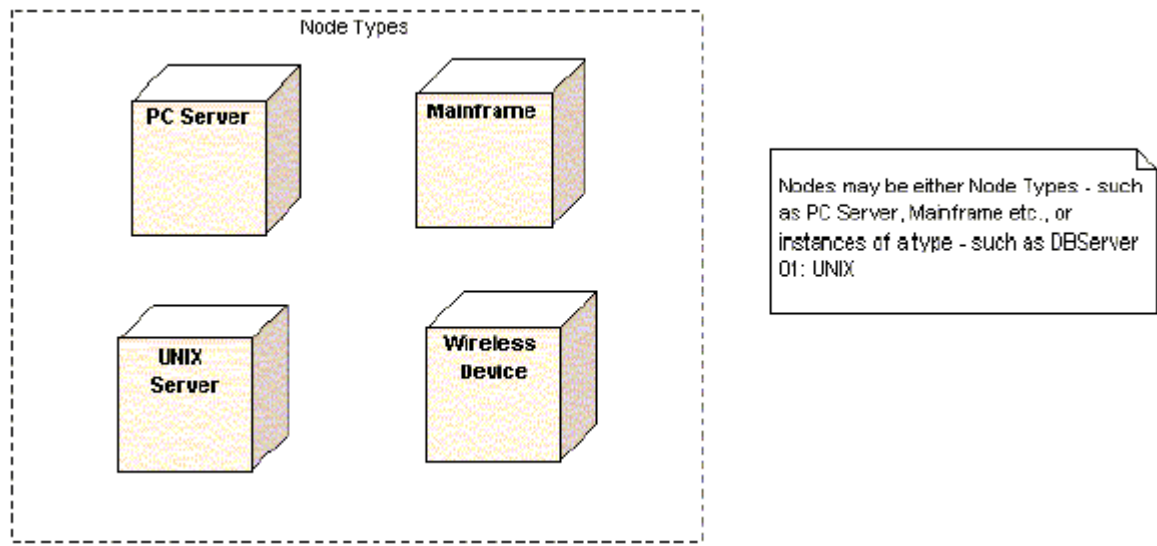
Este artículo se enfoca en la representación funcional de los requisitos en UML utilizando Casos de Uso.

En los libros mencionados en la sección de lectura recomendada se puede encontrar más información sobre el UML y de los documentos de especificación del UML que se pueden encontrar en las páginas de recursos de UML del OMG (*Object Management Group*) [www.omg.org/technology/uml/](http://www.omg.org/technology/uml/) y [www.omg.org/technology/documents/formal](http://www.omg.org/technology/documents/formal).

### *El Modelo de Despliegue*

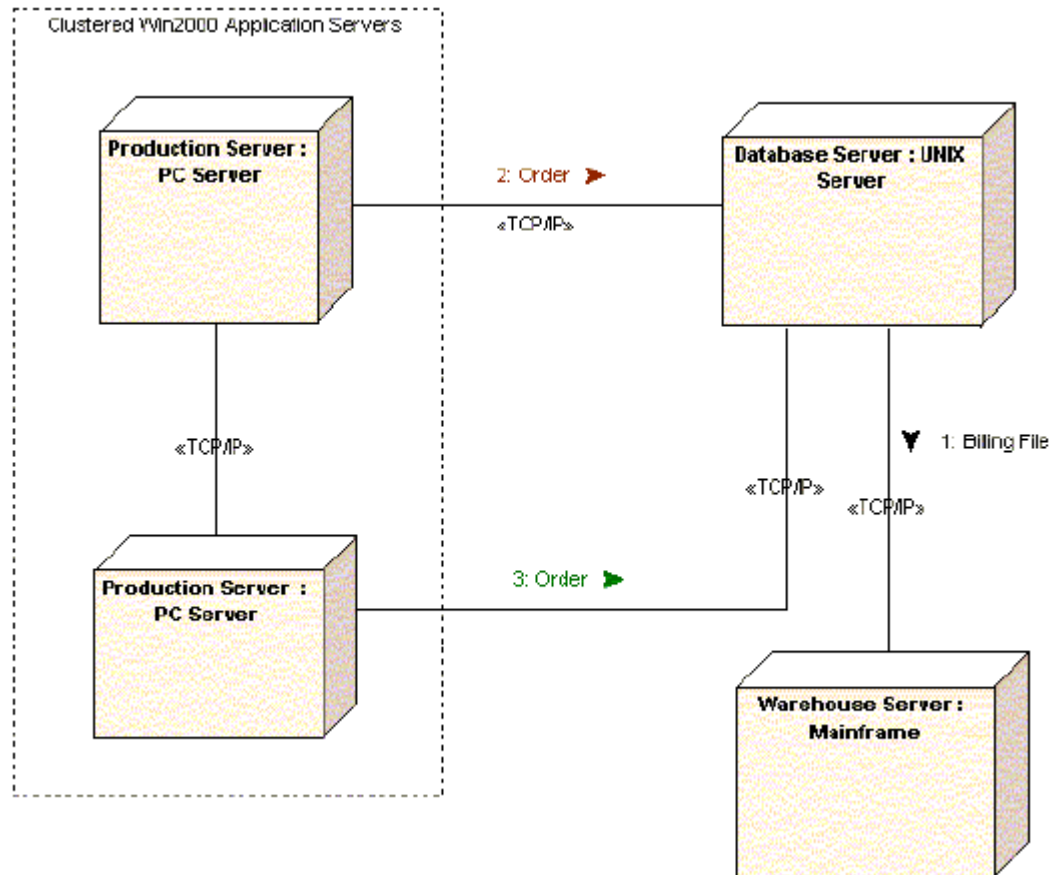
Los diagramas de despliegue representan los nodos y sus relaciones. Típicamente, los nodos son conectados por asociaciones de comunicación -tales como enlaces de red, conexiones TCP/IP, microondas, etc-.

La imagen de abajo muestra una variedad de tipos de nodos. Estos son tipos abstractos que pueden ser implementados en tiempo de ejecución por instancias físicas. Es la misma forma en la que se usan una clase y un objeto para modelar la definición abstracta y la instancia en tiempo de ejecución respectivamente.



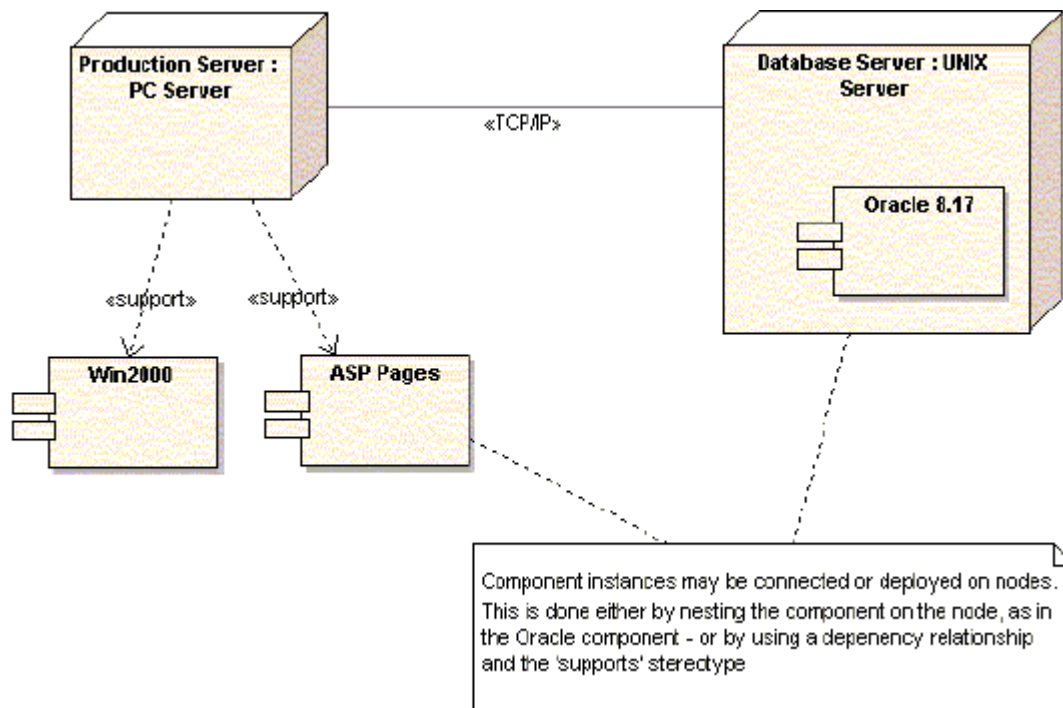
## ***Instancias de Nodos***

Una que se definieron los tipos de los nodos se puede crear el diagrama equivalente de un diagrama de colaboración, usando las instancias de los nodos tal como serán desplegadas en tiempo de ejecución. El ejemplo de más abajo muestra cómo se pueden crear y vincular las instancias de los tipos de nodos ya definidos con asociaciones de comunicación. Los mensajes de colaboración también se utilizaron para representar el flujo de información o los objetos discretos entre los nodos, tales como el archivo de cobranza (*Billing File*) y la información de la solicitud (*Order*).



## Los Componentes

Las instancias de los componentes se usan para representar los ejecutables, las aplicaciones y otros componentes de software que se despliegan en los nodos. El diagrama siguiente muestra esto:

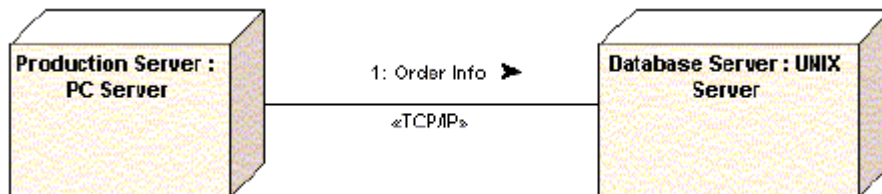


Una instancia es un ejemplo físico en tiempo de ejecución de un tipo abstracto

## La Comunicación

La comunicación entre los nodos se representa usando asociaciones, que se pueden estereotipar para ilustrar el protocolo exacto que se usó para la comunicación -tal como TCP/IP o SNA-.

La imagen abajo muestra dos instancias de nodos -una PC servidor llamada “*Production Server*” y un servidor UNIX llamado “*Database Server*”-. Estas dos instancias están conectadas por una conexión red con TCP/IP. Uno de propósitos principales de este enlace es enviar la información de la solicitud desde el servidor de aplicaciones de producción al servidor de la base de datos, por lo que se agregó un mensaje de colaboración para modelar esto.

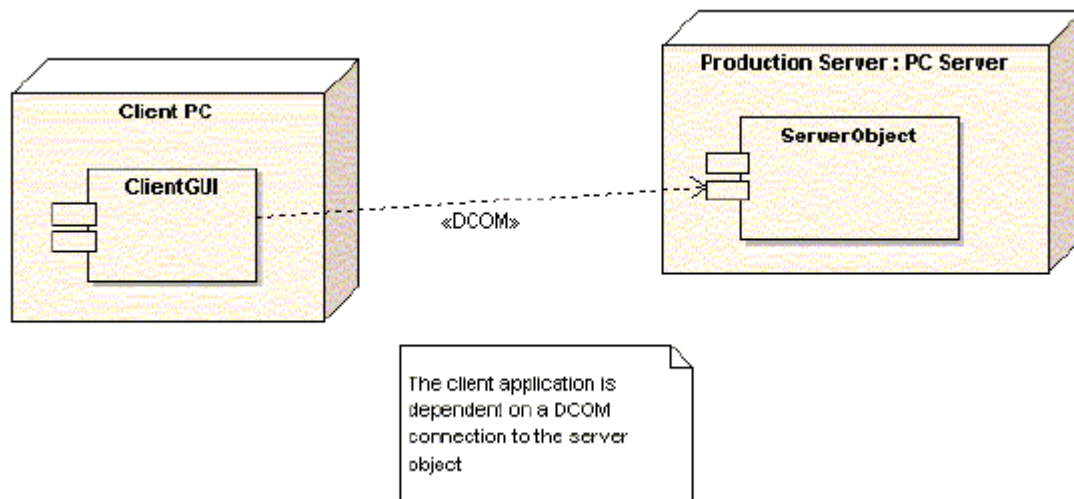


La comunicación entre dos nodos puede tomar varias formas:

- Conectividad de redes
- TCP/IP
- SNA
- Microondas
- Infrarrojo
- Protocolo inalámbrico

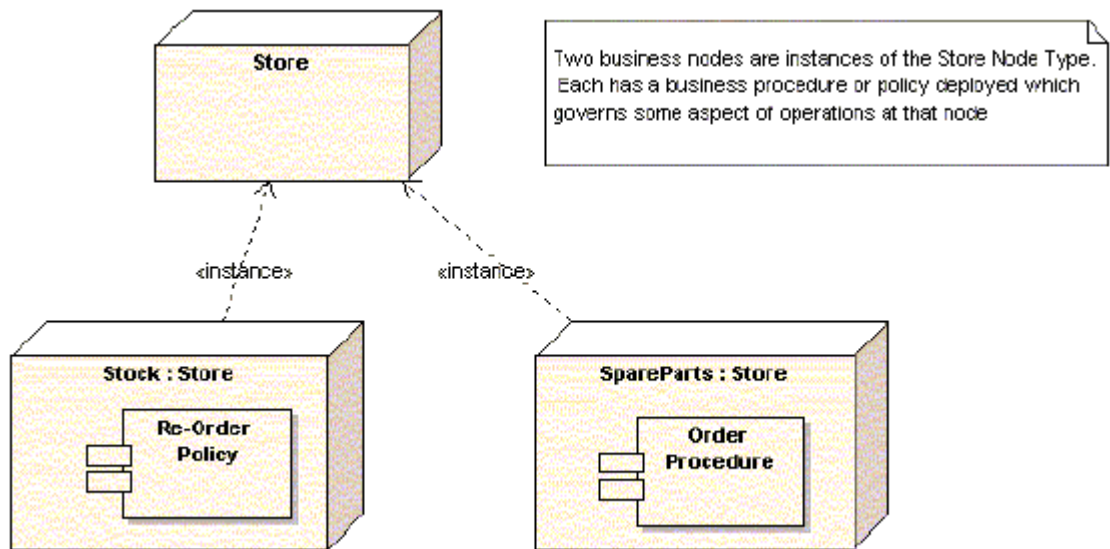
### ***Las Dependencias***

Las dependencias se usan en los diagramas de despliegue para ilustrar las relaciones de dependencia entre los artefactos. Éstas se pueden estereotipar para indicar la relación exacta –por ejemplo se puede indicar <<DCOM>> como un protocolo distribuido dependiente para conectar un cliente y un servidor de objetos.



### ***El Modelado de Negocios y los Diagramas de Implementación***

Los diagramas de despliegue también pueden modelar las entidades y los sistemas no computacionales dentro de una organización -tales como el Local, la Recepción, la Oficina principal y la Sucursal de repuestos. Los componentes de negocio se pueden usar para modelar cosas tales como reglas de negocio, procedimientos y documentos. Estos componentes se pueden desplegar en los nodos de negocio. El ejemplo siguiente muestra esta idea:



---

El Modelado de Negocio usando los nodos y los componentes es una medio efectivo para capturar los procesos y las entidades no computacionales. Esto se puede realizar muy tempranamente en la fase de análisis para complementar el modelo de los casos de uso y otras formas de modelado de negocio.

---



### ***Lectura Recomendada***

Sinan Si Alhir, **UML in a NutShell**.

ISBN: 1-56592-448-7. Publisher: O'Reilly & Associates, Inc

Doug Rosenberg with Kendall Scott , User Interface Driven Object Modeling with UML

ISBN: 0-201-43289-7. Publisher: Addison-Wesley

Geri Scheider, Jason P. Winters, **Applying User Interfaces**

ISBN: 0-201-30981-5. Publisher: Addison-Wesley

Ivar Jacobson, Martin Griss, Patrik Jonsson, **Software Reuse**

ISBN: 0-201-92476-5. Publisher: Addison-Wesley

Hans-Erik Eriksson, Magnus Penker, **Business Modeling with UML**

ISBN: 0-471-29551-5. Publisher: John Wiley & Son, Inc

Peter Herzum, Oliver Sims, **Business Component Factory**

ISBN: 0-471-32760-3 Publisher: John Wiley & Son, Inc