Books & Authors

**Database Programming with JDBC and Java, Second Edition**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  |  | |  |  | | --- | --- | |  | [[Print-friendly Version](http://java.sun.com/jsp_utils/PrintPage.jsp)Print-friendly Version](http://java.sun.com/jsp_utils/PrintPage.jsp) | |   [Books Index](http://java.sun.com/developer/Books/)  **Second Edition** by George Reese November 2000  Introduction | [Chapter 7: Distributed Application Architecture](http://java.sun.com/developer/Books/jdbc/ch07.pdf) | [Chapter 8: Distributed Component Models](http://java.sun.com/developer/Books/jdbc/ch08.pdf)   |  |  | | --- | --- | | Book cover | The JDC is pleased to present two chapters from *Database Programming with JDBC and Java, Second Edition* by George Reese, published by O'Reilly & Associates. |   **About the Book**  This book describes the standard Java interfaces that make portable object-oriented access to relational databases possible, and offers a robust model for writing applications that are easy to maintain. The second edition has been completely updated for JDBC 2.0, and includes reference listings for JDBC and the most important RMI classes. The book begins with a quick overview of SQL for developers who may be asked to handle a database for the first time, and goes on to explain how to issue database queries and updates through SQL and JDBC.  **About the Chapters**  [Chapter 7,](http://java.sun.com/developer/Books/jdbc/ch07.pdf) "Distributed Application Architecture," looks at the "big picture"--the landscape into which database programming falls in real world applications.  [Chapter 8,](http://java.sun.com/developer/Books/jdbc/ch08.pdf) "Distributed Component Models," covers the details of distributed computing that EJB lets you ignore, and shows you how to implement your own model when EJB cannot be used. | | | | | | | |

Cliente servidor

La arquitectura cliente/servidor es comúnmente una arquitectura que se basa en el concepto de procesamiento en dos o más máquinas. Una aplicación es cliente/servidor si se el almacenamiento de los datos se encuentra apartado de la presentación. En este caso, el servidor es un motor de base de datos que almacena datos y el cliente es el proceso que recupera o crea dichos datos. La idea detrás de esta arquitectura es que el servidor pueda permitir el acceso de múltiples usuarios a la misma fuente de datos.

La forma más simple de la arquitectura cliente servidor es llamada arquitectura de dos niveles. El término “dos niveles” significa que el procesamiento de la aplicación se divide entre el cliente y el servidor. Típicamente esta arquitectura cuenta con múltiples clientes cuya presentación es igual, y un servidor encargado de la persistencia de los datos.

Clientes pesados: La filosofía detrás de la arquitectura cliente/servidor supone que cada máquina realiza el procesamiento relevante para sus tareas. Los clientes son diseñados para proveer a los usuarios una interfaz gráfica lo que supone una presentación de los datos. Los servidores son diseñados para manejar la complejidad de los datos y las comunicaciones por lo cual sirven como almacenes de datos. Conforme los sistemas van creciendo, se van encontrando diferentes necesidades de procesamiento que no tiene un lugar claro en el modelo cliente/servidor. Por esto, los clientes ahora son más potentes, siendo capaces de soportar interfaces gráficas más robustas y procesamiento de información y los servidores cuentan con más capacidad de almacenamiento y procesamiento a la vez