

Actividad de trabajo autónomo

1. What is distributed computing?

a. Distributed computing refers to a model where computational tasks are divided and spread across multiple machines (or nodes) that communicate over a network to achieve a common goal. Each node in the system contributes its resources (CPU, memory, etc.) to help process parts of the overall task. Distributed systems improve efficiency, scalability, and fault tolerance, as they don't rely on a single central machine. The Internet and cloud computing are popular examples of distributed computing systems. (Deitel & Deitel, 2018).

i. In the context of Java, distributed computing often involves using technologies such as Sockets, RMI (Remote Method Invocation), and Web Services to enable communication between different systems.

2. What are Sockets used for?

a. Sockets are endpoints for communication between two machines over a network. They allow processes on different systems (or even on the same machine) to send and receive data. Sockets use underlying communication protocols like TCP (Transmission Control Protocol) or UDP (User Datagram Protocol) to establish a connection and facilitate data exchange. (Deitel & Deitel, 2018).

i. In Java:

1. Client sockets are used to initiate a connection to a server.

2. Server sockets listen for incoming connection requests from clients.

ii. Sockets are a fundamental component in network programming, enabling direct communication between machines for activities like file transfer, chat applications, and real-time data streaming.

3. What is the difference between UDP and TCP?

a. TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are two of the most commonly used transport layer protocols. Here's how they differ:

i. TCP is a connection-oriented protocol. It establishes a reliable connection between the sender and receiver before data transfer. TCP guarantees that the data packets will arrive in the correct order, and

ensures that lost or corrupted packets are retransmitted. It is used for applications where data integrity and reliability are crucial, like web browsing (HTTP), email (SMTP), and file transfer (FTP).

- ii. UDP is a connectionless protocol. It sends data packets (called datagrams) without establishing a connection first. There is no guarantee that the packets will arrive in order, or even at all. UDP is faster than TCP because it doesn't have the overhead of maintaining a connection, but it sacrifices reliability. It is commonly used in applications where speed is more critical than accuracy, such as video streaming, online gaming, or Voice over IP (VoIP). (Deitel & Deitel, 2018).

4. What is RMI and JNDI? And how are they related to Sockets?

- i. RMI (Remote Method Invocation): RMI is a Java API that allows an object in one Java virtual machine (JVM) to invoke methods on an object located in another JVM, potentially on a different physical machine. It provides a way for Java programs to communicate with each other over a network, abstracting the complexity of underlying socket communication. RMI uses TCP/IP for data transmission under the hood, meaning that at a low level, it relies on sockets for communication.
 - ii. JNDI (Java Naming and Directory Interface): JNDI is an API used for accessing naming and directory services, such as DNS, LDAP, and RMI registry. It provides a way to look up objects and services in a distributed environment. Although JNDI itself doesn't use sockets directly, it can access services that are accessed over a network using sockets. (Deitel & Deitel, 2018).
- b. In short, both RMI and JNDI abstract the use of sockets in Java by providing higher-level mechanisms for distributed object communication (RMI) and resource lookups (JNDI).

5. What is a Web Service?

- a. A Web Service is a standardized way for applications to communicate with each other over a network, often the internet. Web services allow different systems, written in different programming languages or running on different

platforms, to exchange data using standard protocols such as HTTP, SOAP (Simple Object Access Protocol), or REST (Representational State Transfer).

- b. Web services typically use XML or JSON to encode the data being transmitted. They provide a way for applications to interact with each other without needing to know the details of the underlying implementation. For example, a Java application can communicate with a web service written in Python, and they can exchange information in a language-neutral format like JSON or XML. (Deitel & Deitel, 2018)

- i. There are two main types of web services:

- 1. SOAP Web Services: These are based on the SOAP protocol and usually involve more overhead with XML-based messaging. They are used in enterprise applications that require more formal standards for security and transactions.
- 2. RESTful Web Services: These are based on the simpler REST architecture and use standard HTTP methods (GET, POST, PUT, DELETE). RESTful services are widely used due to their simplicity and scalability, particularly in web applications and mobile apps.

Preguntas orientadoras

- 1. ¿Cuáles fueron los aprendizajes obtenidos al realizar esta guía?, lista como mínimo 3 aprendizajes y relaciónelos con su futuro quehacer profesional.
 - a. **Conocimiento sobre Computación Distribuida:** Aprendí cómo dividir tareas entre varias máquinas para mejorar la eficiencia y escalabilidad, lo cual es esencial en el desarrollo de aplicaciones en la nube.
 - b. **Diferencias entre TCP y UDP:** Comprendí cuándo usar cada protocolo según las necesidades de una aplicación, lo que es crucial para desarrollar sistemas que requieren velocidad (UDP) o confiabilidad (TCP).
 - c. **Implementación de RMI y Sockets en Java:** Entender el uso de RMI y sockets me preparó para crear aplicaciones que se comuniquen de forma efectiva en redes, algo fundamental para proyectos de software distribuidos.

2. ¿Dónde presentó mayor dificultad resolviendo la guía? y ¿cómo lo resolvieron?
¿Cuáles fueron las estrategias de solución?

- a. La mayor dificultad fue comprender la configuración y el uso de RMI en Java, ya que requiere una correcta configuración de la red y la coordinación entre las diferentes máquinas virtuales Java para que la comunicación sea efectiva.

Estrategias de solución:

- a. Búsqueda de documentación y tutoriales: Utilicé la documentación oficial de Java y tutoriales en línea para entender cómo configurar y ejecutar RMI, lo que me ayudó a comprender mejor los pasos necesarios.

Bibliografía

Deitel, P., & Deitel, H. (2018). Cómo programar Java (7ª ed.). Pearson Educación.