



## Practica 3 Núcleo Digital

1. ¿Por qué el lenguaje de máquina es importante para definir la arquitectura computacional?

- El lenguaje de máquina es básicamente el conjunto de instrucciones que una computadora puede ejecutar directamente. Cada tipo de procesador, o CPU, tiene su propio lenguaje de máquina, que está hecho a medida para esa arquitectura en particular. Esto es súper importante porque define cómo el procesador entiende y ejecuta las instrucciones. ¿Por qué es importante?

**Especificidad del Hardware:** El lenguaje de máquina marca cómo se organizan y funcionan los componentes internos del procesador, como los registros y las unidades aritmético-lógicas.

**Optimización:** Ayuda a los diseñadores a ajustar el rendimiento del sistema. Por ejemplo, algunos lenguajes de máquina permiten hacer operaciones más rápido que otros.

**Compatibilidad:** Asegura que el software y el hardware puedan comunicarse de manera efectiva.

2. ¿Qué diferencia ven entre arquitectura computacional, arquitectura de software y arquitectura del sistema? Justifique su respuesta.

	<b>Arquitectura Computacional</b>	<b>Arquitectura de Software:</b>	<b>Arquitectura del Sistema</b>
<b>¿Qué es?</b>	Es la estructura y organización del hardware de una computadora. Incluye la CPU, la memoria y los buses de comunicación.	Es el diseño y la organización del software, incluyendo cómo están estructurados los componentes de software y cómo interactúan entre sí. Se centra en los patrones de diseño y la estructura del código.	Es una combinación de la arquitectura computacional y la arquitectura de software. Es el diseño general que incluye tanto el hardware como el software y cómo interactúan entre ellos.
<b>¿En qué se enfoca?</b>	Se trata de cómo se organizan y conectan los componentes físicos para que puedan ejecutar instrucciones de manera eficiente.	Se enfoca en cómo se organizan las aplicaciones y los sistemas operativos para que sean eficientes y fáciles de mantener.	Se trata de cómo el hardware y el software trabajan juntos para formar un sistema completo y funcional.

La arquitectura computacional se centra en optimizar el diseño del hardware para mejorar el rendimiento y es esencial cuando se está construyendo o ajustando el hardware de una computadora. La arquitectura de software organiza y estructura el software para facilitar su desarrollo y mantenimiento, siendo crucial en el diseño de aplicaciones y sistemas operativos. La arquitectura del sistema integra ambos aspectos, hardware y software, para asegurar que trabajen juntos de manera eficiente y es clave para el diseño de sistemas completos y funcionales.

**Bonus:** Como informático o computista: ¿La arquitectura computacional o la arquitectura del sistema no tiene en cuenta igualmente la arquitectura de software? Justifique su respuesta.

- Como informático, entiendo que tanto la arquitectura computacional como la arquitectura del sistema necesitan tener en cuenta la arquitectura de software. Aunque la arquitectura computacional se enfoca en el diseño del hardware, como la CPU y la memoria, para que funcione bien con el software, el hardware debe estar preparado para soportar las necesidades del software. Por otro lado, la arquitectura del sistema integra tanto el hardware como el software, por lo que aquí la arquitectura de software es clave para asegurar que todo funcione juntos de manera eficiente. En resumen, la arquitectura de software siempre juega un papel importante en ambas áreas para lograr un sistema bien integrado.

## REFERENCIAS

Hennessy, J. L., & Patterson, D. A. (2019). *Computer Architecture: A Quantitative Approach* (5ª ed.). Morgan Kaufmann. Recuperado de [https://acs.pub.ro/~cpop/SMPA/Computer%20Architecture%20A%20Quantitative%20Approach%20\(5th%20edition\).pdf](https://acs.pub.ro/~cpop/SMPA/Computer%20Architecture%20A%20Quantitative%20Approach%20(5th%20edition).pdf)

Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice* (3ª ed.). Addison-Wesley. Recuperado de [https://edisciplinas.usp.br/pluginfile.php/5922722/mod\\_resource/content/1/2013%20-%20Book%20-%20Bass%20%20Kazman-Software%20Architecture%20in%20Practice%20%281%29.pdf](https://edisciplinas.usp.br/pluginfile.php/5922722/mod_resource/content/1/2013%20-%20Book%20-%20Bass%20%20Kazman-Software%20Architecture%20in%20Practice%20%281%29.pdf)

Sommerville, I. (2016). *Software Engineering* (10ª ed.). Pearson. Recuperado de [https://gc.scalahed.com/recursos/files/r161r/w25469w/ingdelsoftwarelibro9\\_compressed.pdf](https://gc.scalahed.com/recursos/files/r161r/w25469w/ingdelsoftwarelibro9_compressed.pdf)