**Ficha: 2694679**

**NOMBRE COMPLETO - SI/NO PRESENTO – MINUTOS/TIEMPO QUE LE FALTO – ARTÍCULO.**

1. Jhon Alexander Corredor Medina – No Presento
2. Yanuard Steven Montealegre Bonilla – No presento
3. Heyder Santiago Rodriguez Galviz – No presento
4. Johan Calderón Perdomo – Si presento – 7 minutos/ 2:30 minutos – **Artículo 1**
5. Maryury Bonilla González – Si presento – 7 minutos/ 1:20 minutos – **Artículo 5**
6. Carlos Andres Pantoja Jaramillo – Si presento – 7 minutos/ 2:10 minutos - **Artículo 8**
7. Hector Fabian Cardoso Morales – No presento
8. Stefanny Nikoll Hidalgo Urrea – Si presento – 7 minutos/ 1:50 minutos – **Artículo 16**
9. Juan David Cerquera Salazar – Si presento – 7 minutos/ 20 minutos – **Artículo 7**
10. Erick Daniel Peña Cedeño – Si presento – 7 minutos/ 2 minutos – **Artículo 2**
11. David Mauricio Florez Quintero – No presento
12. Jose Alejandro Osorio Ramirez – No presento
13. Laura Camila Sanchez Villegas – No presento
14. Marlon Estiven Torres Medina – Si presento – 7 minutos/ 13 minutos – **Artículo 8**

**Resúmenes**

**Johan Calderón Perdomo**  
**Artículo 1:**   
El resumen expone cómo la arquitectura de software organiza un sistema a nivel general, definiendo los elementos y sus relaciones para facilitar el desarrollo. Destaca el uso de arquitecturas de referencia para reutilizar componentes y cómo estas han evolucionado, guiando el diseño y comprensión de sistemas complejos a través de patrones, enfoques estructurales y escenarios.

**Maryury Bonilla González**  
**Artículo 5:**   
Se compararon las arquitecturas monolíticas y de microservicios, analizando cuándo son más efectivas. También se presentaron casos reales, como los de Amazon y eBay, que migraron entre estas arquitecturas, destacando los desafíos y beneficios en términos de escalabilidad, mantenimiento y desarrollo eficiente.

**Carlos Andrés Pantoja Jaramillo**  
**Artículo 8:**   
Se exploraron dos variantes del patrón Modelo-Vista-Controlador (MVC) en aplicaciones web, evaluando su efecto en el tiempo de respuesta y la escalabilidad. La implementación clásica (llamada y respuesta) se comparó con una alternativa basada en tuberías y filtros con Python.

**Stefanny Nikoll Hidalgo Urrea**  
**Artículo 16:**  
La exposición detalló una arquitectura para un robot móvil, organizada en tres niveles: uno para los componentes básicos, otro para librerías que permiten crear aplicaciones de control, y el último que incluye una interfaz con un panel de control y simulador 3D para programar y monitorear el robot. Además, se verificó su funcionamiento con pruebas, y todo fue desarrollado en C#.

**Juan David Cerquera Salazar**  
**Artículo 7:**   
Se analizó un estudio dirigido a profesionales de software sobre las arquitecturas más usadas y sus características principales:

* **Arquitectura en la nube:** Seguridad y flexibilidad.
* **MVC:** Mantenibilidad, rendimiento, velocidad y memoria.
* **Microservicios:** Mantenibilidad, rendimiento, seguridad y flexibilidad.
* **MVP:** Modificabilidad, rendimiento, flexibilidad y modularidad.  
  El estudio también clasificó las arquitecturas según los dispositivos a los que están orientadas, destacando aplicaciones web, móviles y de escritorio.

**Erick Daniel Peña Cedeño**  
**Artículo 2:**  
Se describió cómo los microservicios ofrecen una solución más flexible y moderna frente a las limitaciones de las arquitecturas monolíticas. Esta arquitectura permite que los componentes del sistema sean autónomos e independientes, mejorando la eficiencia en el desarrollo y mantenimiento de aplicaciones.

**Marlon Estiven Torres Medina**  
**Artículo 8:**  
El patrón Flyweight fue presentado como una técnica para optimizar el uso de memoria al compartir datos entre objetos similares. Se mostraron ejemplos de su aplicación en áreas como los videojuegos y sistemas gráficos, junto con sus ventajas, limitaciones y consideraciones para implementarlo.