

# Informe II (Arquitectura del Computador)

Jesus Contreras (C.I: 31.021.750)  
Abraham Quintero (C.I:30.958.868)

July 14, 2025

## **1 ¿Qué diferencias existen entre registros temporales (\$t0--\$t9) y registros guardados (\$s0--\$s7) y cómo se aplicó esta distinción en la práctica?**

Los registros temporales (\$t0--\$t9) se usan para valores de uso breve y los registros guardados (\$s0--\$s7) se usan para conservar los valores entre funciones.

En el algoritmo de bubble sort, los registros temporales \$t0--\$t2 son usados para variables internas mientras que los registros guardados \$s0--\$s2 se usan para valores que van a persistir en el algoritmo.

## **2 ¿Qué diferencias existen entre los registros \$a0--\$a3, \$v0--\$v1, \$ra y cómo se aplicó esta distinción en la práctica?**

los registros \$a0--\$a3 se utilizan como parámetros de entrada, en cambio, los registros \$v0--\$v1 se usan como salida de funciones y \$ra es para indicar la dirección de retorno.

En la práctica las entradas para la impresión se pasó por \$a0 y se recibió resultado por \$v0. \$ra se usó en llamadas a funciones.

## **3 ¿Cómo afecta el uso de registros frente a memoria en el rendimiento de los algoritmos de ordenamiento implementados?**

Los registros se usan para comparar e intercambiar elementos eficientemente permitiendo mayor velocidad y evitando accesos lentos a memoria a diferencia de cuándo se hace el algoritmo directamente desde la memoria lo cual produce latencia y el programa termina siendo menos eficiente.

#### **4 ¿Qué impacto tiene el uso de estructuras de control (bucles anidados, saltos) en la eficiencia de los algoritmos en MIPS32?**

Los bucles anidados aumentan el número de instrucciones haciendo la ejecución mas lenta y con el uso de saltos eficientes se puede optimizar el algoritmo moviendonos hacia distintos procedimientos dependiendo de cómo se vaya desarrollando la ejecución del programa sin tener que ejecutar líneas de código innecesarias que suelen ser frecuentes en los bucles anidados.

#### **5 ¿Cuáles son las diferencias de complejidad computacional entre el algoritmo Bubble Sort y el algoritmo alternativo? ¿Qué implicaciones tiene esto para la implementación en un entorno MIPS32?**

Ambos algoritmos son de orden ( $n^2$ ) debido a que utilizan dos ciclos cada uno, sin embargo, el algoritmo de SelectionSort tiene menos interacciones debido a que por cada recorrido del vector, solamente realiza un intercambio a comparación del BubbleSort que puede hacer uno o más intercambios en cada recorrido del ciclo interno.

#### **6 ¿Cuáles son las fases del ciclo de ejecución de instrucciones en la arquitectura MIPS32 (camino de datos)? ¿En qué consisten?**

1. Fetch: Se obtiene la instrucción desde la memoria.
2. Decode: Se decodifica y se leen los registros fuente.
3. Execute: Se hace la operación en la ALU.
4. Memory Access: Acceso a la memoria si es que aplica.
5. Write-back: Se escribe el registro destino.

#### **7 ¿Qué tipo de instrucciones se usaron predominantemente en la práctica (R, I, J) y por qué?**

- Tipo R: operaciones aritméticas y comparaciones.

- Tipo I: acceso a memoria.
- Tipo J: saltos en bucles.  
En ambos algoritmos se ve más que todo operaciones tipo R debido a las constantes comparaciones e intercambios para ordenar los vectores usando operaciones aritméticas.

## **8 ¿Cómo se ve afectado el rendimiento si se abusa del uso de instrucciones de salto (j, beq, bne) en lugar de usar estructuras lineales?**

El uso de muchos saltos pueden volver confuso, fragmentar el flujo y reducir la eficiencia del algoritmo, a diferencia de usar estructuras secuenciales que mejoran el rendimiento y nos dan un resultado más ordenado y práctico.

## **9 ¿Qué ventajas ofrece el modelo RISC de MIPS en la implementación de algoritmos básicos como los de ordenamiento?**

Las ventajas que ofrece son un diseño simple que permite una ejecución rápida de instrucciones básicas, ideal para algoritmos con operaciones repetitivas como ordenamientos.

## **10 ¿Cómo se usó el modo de ejecución paso a paso (Step, Step Into) en MARS para verificar la correcta ejecución del algoritmo?**

Se utilizó para verificar línea por línea el código de nuestros algoritmos, nos ayudó a entender mejor los recorridos mientras usábamos los saltos, poder detectar específicamente dónde se encontraban los errores y validar los registros verificando que funcionen correctamente.

## **11 ¿Qué herramienta de MARS fue más útil para observar el contenido de los registros y detectar errores lógicos?**

El terminal o procesador porque nos indicaba dónde se encontraban los

errores que teníamos en los algoritmos.

## **12 ¿Cómo puede visualizarse en MARS el camino de datos para una instrucción tipo R? (por ejemplo: add)**

- Uso de ALU para sumar registros y guardar el resultado.
- MARS muestra cómo fluyen los datos dentro del procesador.

## **13 ¿Cómo puede visualizarse en MARS el camino de datos para una instrucción tipo I? (por ejemplo: lw)**

- Se usa base + desplazamiento para acceder a memoria.
- El contenido se transfiere a un registro destino.

## **14 Justificar la elección del algoritmo alternativo**

Optamos por utilizar el algoritmo de Selection Sort porque tiene menos intercambios, un menor uso de registros y mayor legibilidad en MIPS.