**INFORME SIMULATOR DE PIPELINE Y MEMORIAS EN PYTHON**

**UNIVERSIDAD SERGIO ARBOLEDA**

**PROYECTO FINAL DE SEMESTRE – ARQUITECTURA DE COMPUTADORES**

**JUAN VASQUEZ**

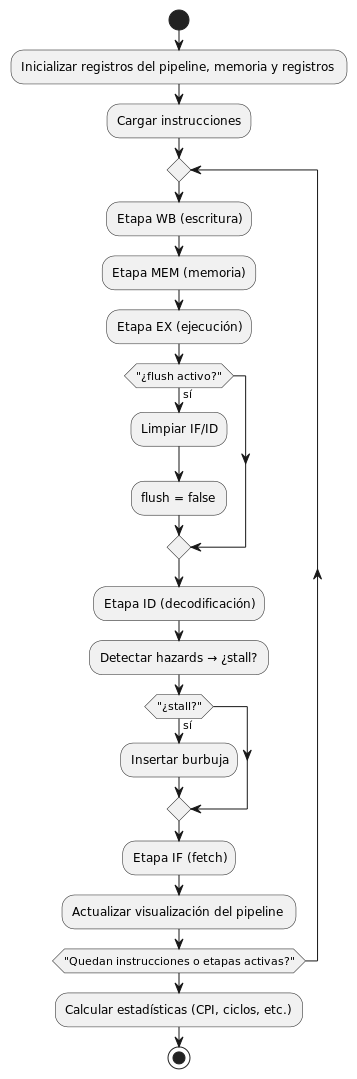
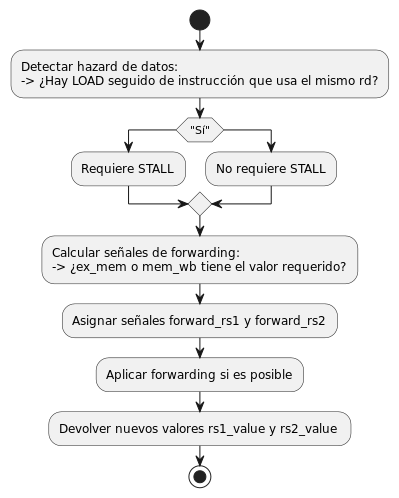
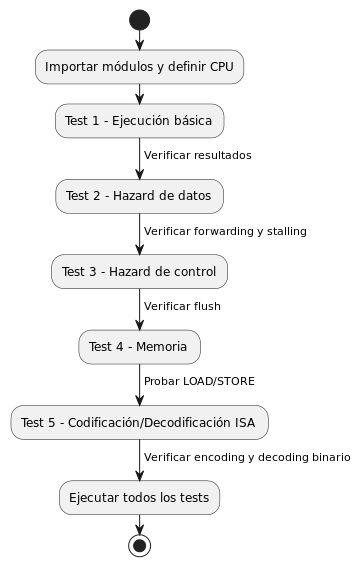
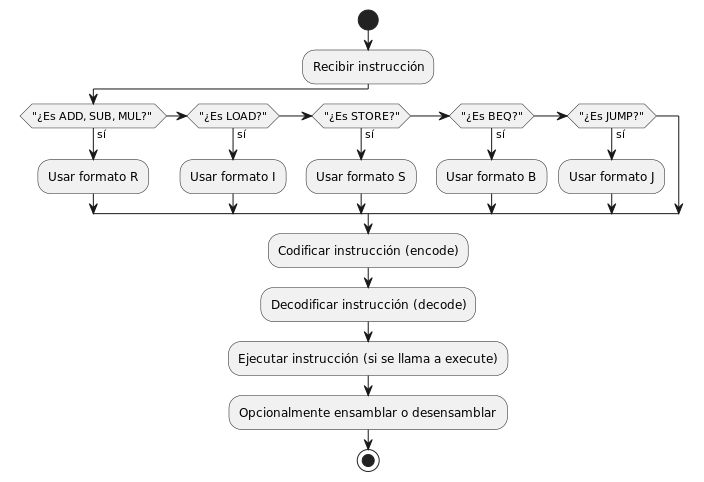
**MIGUEL CELIS**

**JERONIMO LINARES**

**SANTIAGO RODRIGUEZ**

**DOCENTE: OSCAR ANDRÉS ARIAS**

**26 DE MAYO DE 2025**

**DIAGRAMA PIPELINE****DIAGRAMA HAZARD**

**SIMULACION DE DIAGRAMA PIPELINE**

**El simulador utiliza un pipeline de 5 etapas de un procesador RISC, este diseño fue escogido debido al paralelismo y la detección de riesgos que se realiza en la segunda etapa, permitiendo modelar varias instrucciones y simular riesgos de datos y control, en la simulación se utiliza la instrucción LD frecuentemente para poder visualizar el funcionamiento y la interacción de los componentes del sistema, esto dado que es una de las instrucciones que tiene menor complejidad.**

**Durante el flujo de las instrucciones en el pipeline se van ejecutando acciones según la etapa en la que se encuentra, en IF (Fetch) se obtiene la instrucción de la memoria, en ID (Decode) se identifican operandos, en este caso el registro de destino (rd), y el inmediato que representa una dirección de memoria, en EX (Execute) se realizan las operaciones matemáticas o en este caso se busca la dirección de memoria, en MEM (Memory Access) se accede a direcciones de memoria y se guarda el valor, finalmente en WB (Write Back) el valor guardado anteriormente es almacenado en la memoria cache**

**La detección de riesgos o Hazards simples se da al comprobar si el registro fuente de la instrucción en ID es el mismo que se encuentra en la de EX, en caso de que esto suceda se produce un stall para evitar una lectura incorrecta del registro, lo que resulta en la detención de la ejecución de la etapa EX y las anteriores a esta para permitir que EX finalice su operación y se pueda escribir el resultado de esta.**

**ESPECIFICACIONES DE LA ISA:**

**El simulador utiliza una ISA (instruction Set Architecture) para una arquitectura RISC, en esta arquitectura se presentan instrucciones simples de longitud fija, la ISA implementada facilita dividir la ejecución durante las diferentes etapas del pipeline y el realizar la ejecución por ciclos.**

**La isa implementada almacena el código de la operación a realizar (Opcode), un registro de destino (rd), un primer registro fuente (rd1), un segundo registro fuente (rd2), un valor inmediato (imm) y un valor de salida (result), esto permite representar instrucciones de diferente tipo, y lo hace compatible con la arquitectura RISC**

**A computer screen shot of a diagram

AI-generated content may be incorrect.**

**El simulador de memoria caché implementa dos tipos de organización:**

1. **Caché de Mapeo Directo: Cada bloque de memoria principal solo puede ubicarse en una única línea de caché específica.**
2. **Caché Asociativa por Conjuntos (2-way): Cada bloque de memoria principal puede ubicarse en cualquiera de las dos vías dentro de un conjunto específico.**

**Ambas implementaciones son parametrizables en términos de tamaño de bloque y número de líneas/conjuntos.**