

# Explorando ARM y Sus Herramientas

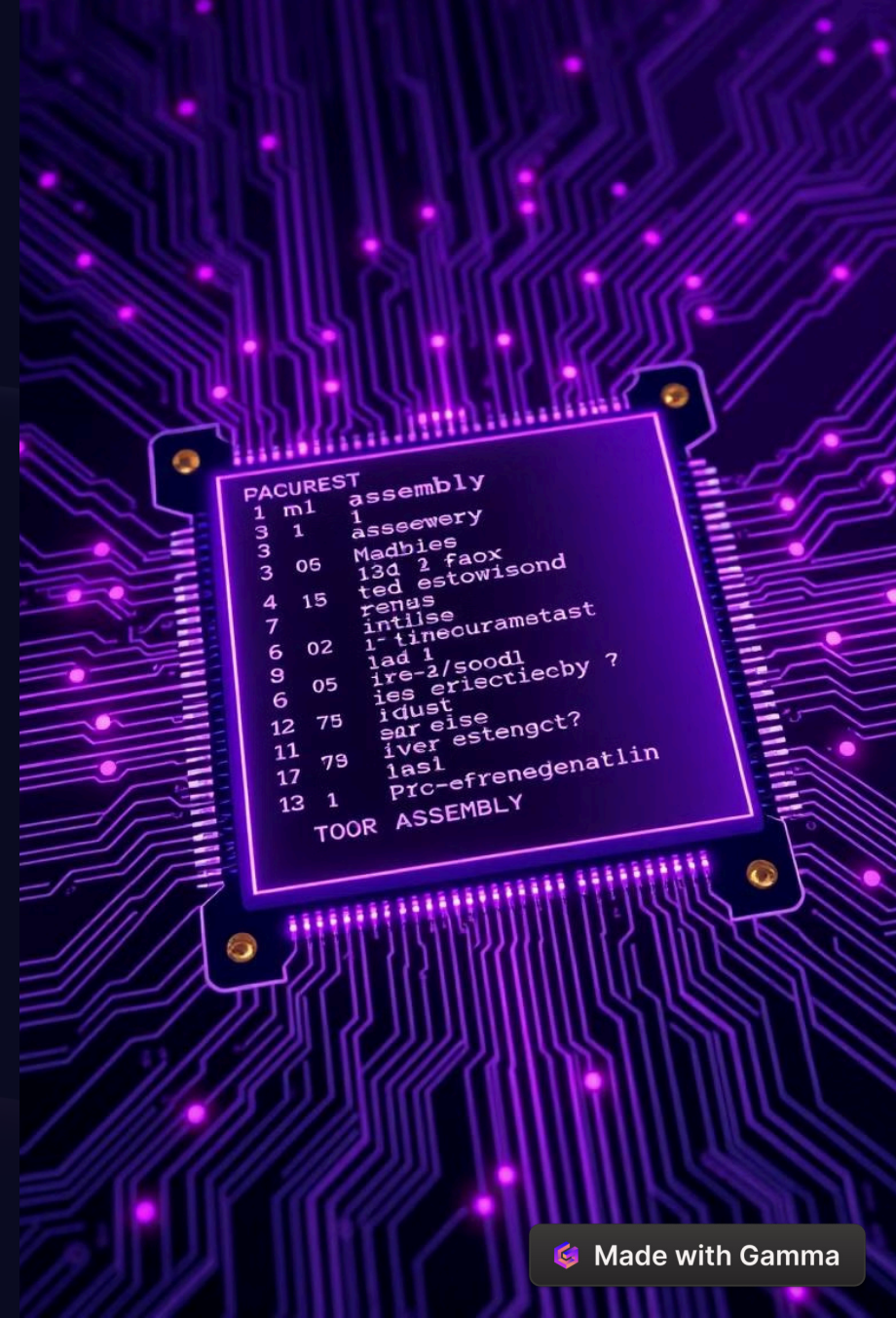
ARM (Advanced RISC Machine) es una arquitectura de procesador ampliamente utilizada en dispositivos móviles, electrónica de consumo y sistemas embebidos. En esta presentación, exploraremos a fondo la arquitectura ARM y las diversas herramientas disponibles para trabajar con ella.

**H** por Hugo Martinez



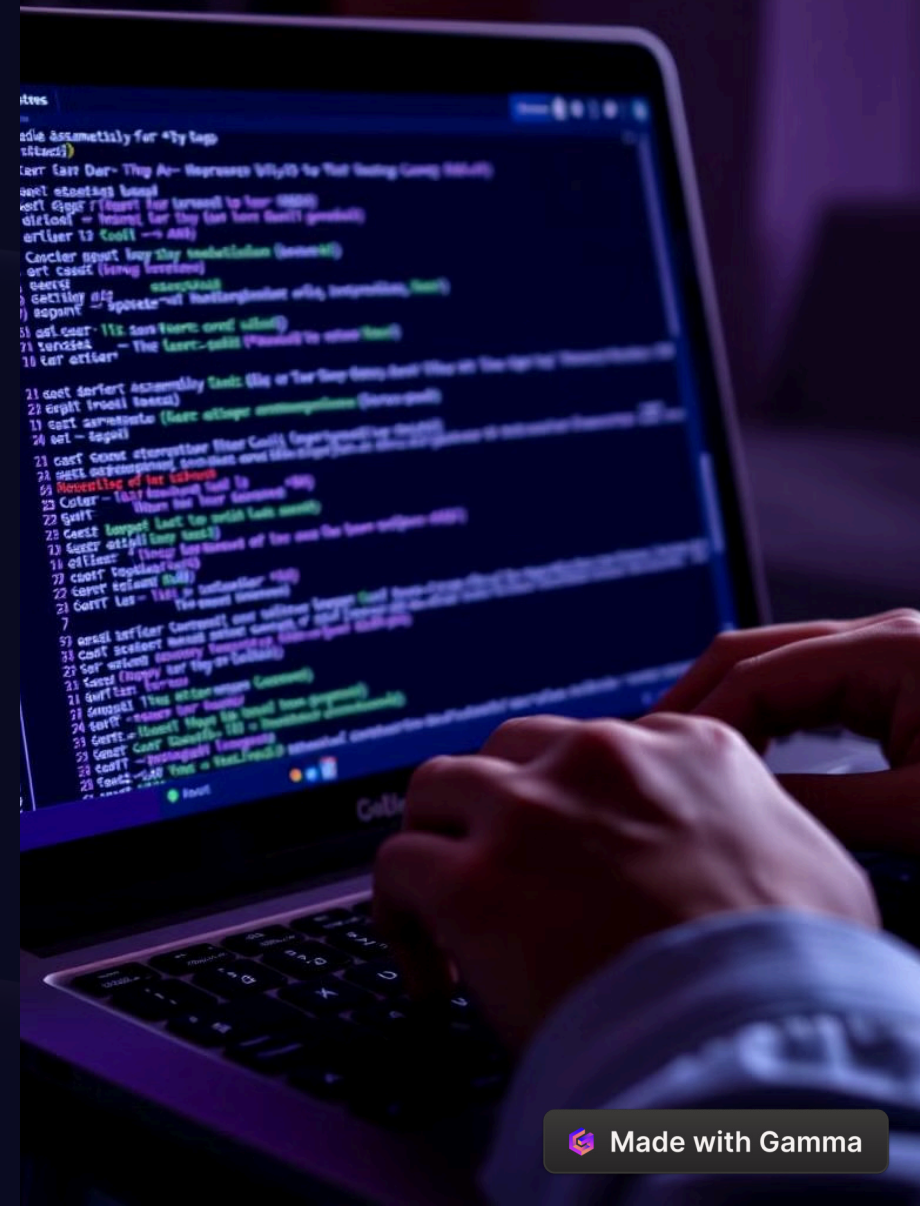
# Directivas de Ensamblador

- Las directivas de ensamblador son instrucciones especiales que proporcionan información adicional al ensamblador sobre cómo ensamblar y formatear el código.
- Permiten definir etiquetas, reservar espacio de memoria, establecer el modo de operación y más.
- Ejemplos comunes incluyen **.text**, **.data**, **.equ**, **.org**, **.align**, entre otras.



# Usando el Ensamblador de ARM (GDB)

**GNU Debugger (GDB)** es una potente herramienta que permite depurar y analizar código ensamblador de ARM. GDB ofrece funciones avanzadas como puntos de interrupción, análisis de memoria y registros, y rastreo de ejecución paso a paso.





# Usando el Depurador de ARM (GDB)

GDB permite examinar el estado del programa en tiempo de ejecución, incluyendo el contenido de registros y memoria.

Ofrece funciones avanzadas como puntos de interrupción, evaluación de expresiones y manipulación de variables.

Con GDB, los desarrolladores pueden identificar y solucionar problemas de manera eficiente en código ensamblador de ARM.

# Manejo de Registros Flotantes

1

## Registros Flotantes de ARM

Los registros flotantes de ARM permiten realizar operaciones matemáticas avanzadas con números de coma flotante.

2

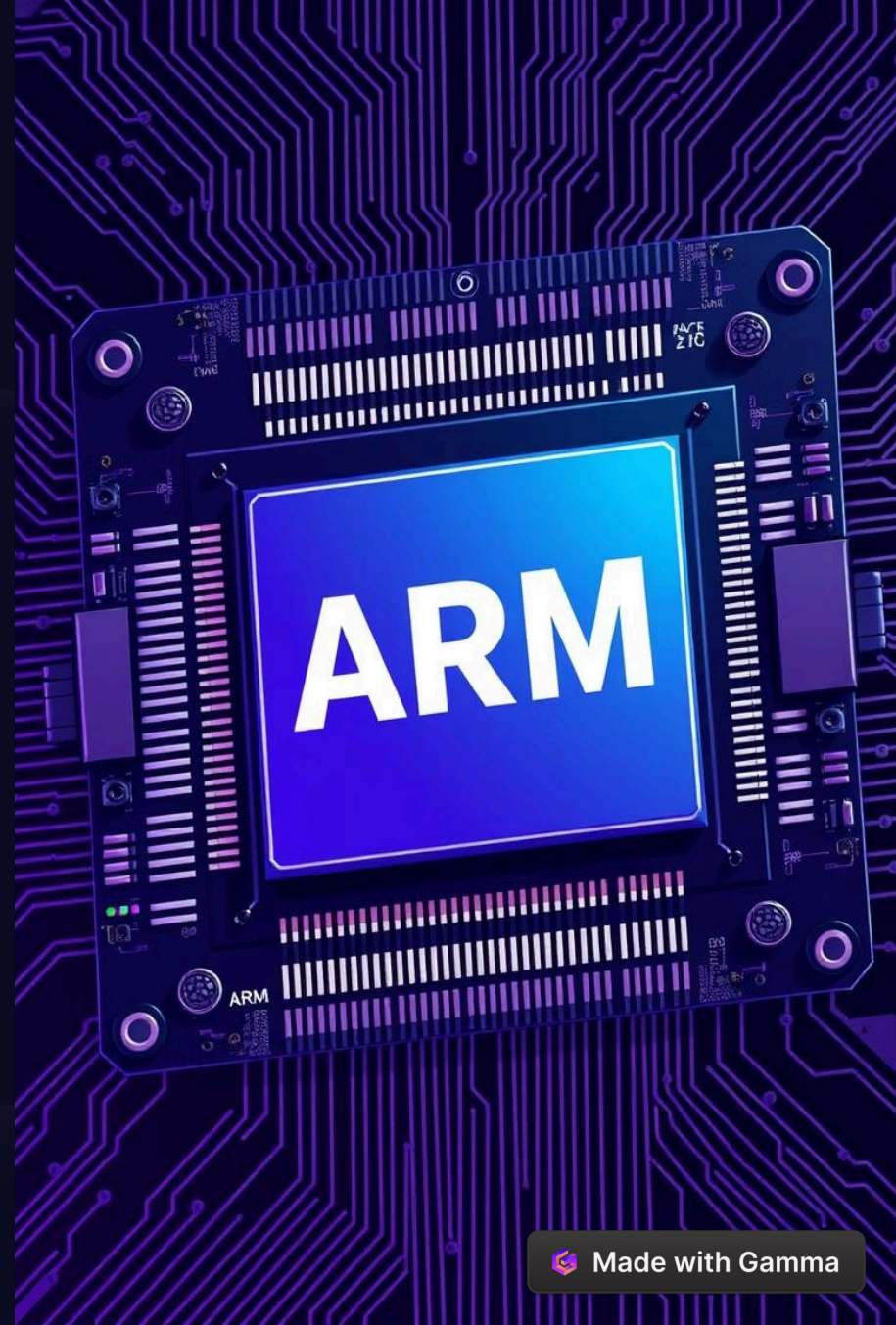
## Instrucciones Útiles

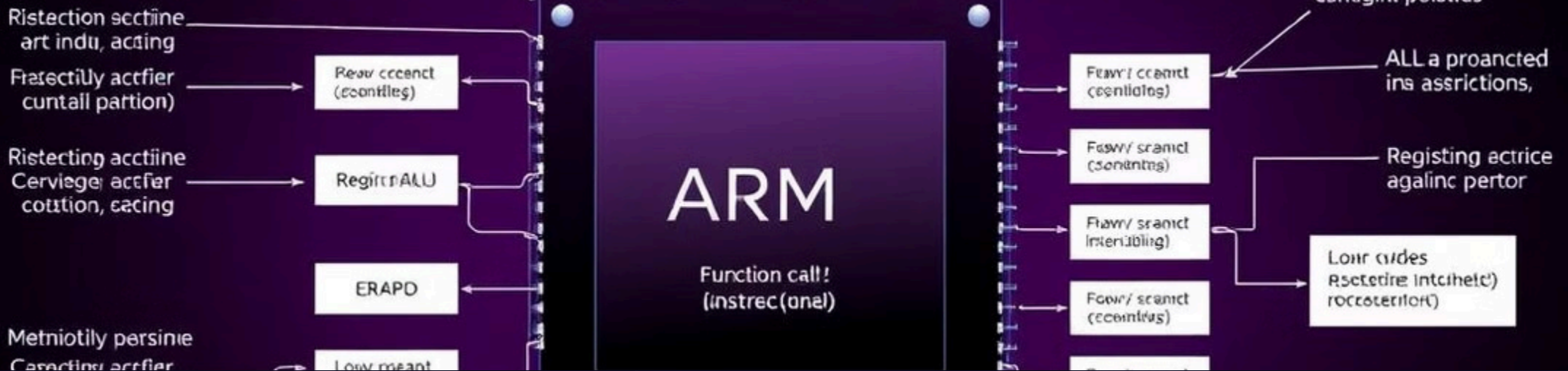
Instrucciones como VMOV, VADD y VMUL facilitan el manejo de datos en formato IEEE 754.

3

## Importancia del Rendimiento

El uso eficiente de los registros flotantes es clave para aplicaciones que requieren un alto rendimiento numérico.





# Creación de Funciones

## Encapsulación y Reutilización

Las funciones en código ensamblador de ARM permiten encapsular y reutilizar bloques de código.

## Llamada y Retorno Eficiente

Mediante el uso de instrucciones como BL (Branch and Link) y POP/PUSH, se puede llamar y retornar de las funciones de manera eficiente.



# Estructuras de Control de Flujo

1

## Instrucciones de Salto

Las instrucciones de salto como B, BL, BX permiten implementar estructuras de control de flujo como condicionales y bucles.

2

## Decisiones Lógicas

El uso de etiquetas y comparaciones de registros facilita la creación de decisiones lógicas en código ensamblador.

3

## Algoritmos Complejos

Técnicas como loops anidados y llamadas a subrutinas permiten construir algoritmos complejos a partir de bloques de código reutilizables.

# Análisis de Ejemplo de Código

Revisaremos un programa en ensamblador de ARM que implementa una función de suma recursiva. Analizaremos cómo se manejan los registros, las llamadas a funciones y el control de flujo en este ejemplo concreto.

```
1372977. Laxe fles
1537627. Cnechalor
1537977. rasrbille_(afa(@)san$)_tope)
2379831. User ≈et.ione
1399410. ---
2532981. perfulilo-cauctuiff (eal.rby.
1559986. espictloe-tejuaioncsimel.etisuit)
1542977. Allmule Spard.cne(ted))
1500438. Insepblline 3,inat typot.figsCT./atel screttic.ist)
1572418. topfull(oole),lot;
    Tate-220ach, entact_snel.fores: late,
1449903. Tescutlin=-Cope/lancion, tt.Talst atturarbal,/lerapbe(e/efacti3pariel
3657952. lod.
3727985. resectile=- (everfuls.ift/Inscrtisloy if; (nersforbalstestille: insteret
4766377. Inscblilde=- faschass sum
5459584. Inschlile=(epoffolwatl onings recielsal salled)
4479964. miender
5762184. tosectile=esfetersetsolation.late:
3797483. inniuer
3154783. Insccllle=-estertive_jentangaler/addhactef): (typet in'a detiniial, ifr
3375983. Insecllle= cecritewolen_pine chgamaat).
3563387. Insepllle=-fentrine,fored)
3787938. Fanchlide./loun
7937560. Insecllle= lyadoetle:_aissler(oreadefiesstionwalle tenper paiss_flat)
5759922. Inschllle= poffesteripoctefentoy.iom).
3657723. Inschlile= ldtals; (oederedocation sum)
7777938. Iannallde mi:flow.
3377854. Inschllde= tarffruse(land/lacferooter(eastinp/ectareaphor sumld, coo
1777933. Inaectide= tpot/facrcion_obccofoet_shtd)
1331993. Insralod: fosts;
1577994. Iascollle= facction saris acc-dhasteel ent tenal)
1573278. Innectide= finticalestalle)
1777597. Insechide=-fostinlles aitec(roded_jectead teati)
1973989. Inschllde=-poiferter from) (Peals90f(joum backel)
1577397. Innchllde= ractec(_avelAinfcsar(enfirorent)
1242788. Clashllq_puot(narseatium tecfCinversibed coing - secrerinsrriliciteg
1973454. Inscollf. toccties;
1322087. (auchled aporaleniny;
1777996. Innchlld==Molder( Rectiiont)
1777976. Fanchlld=-teroec(j fockt)lacker for rescrition jofe ind:
1444983. Inechlld=-focer//l_persrtion, persection, stroct/teciit)
    fanalor.
    ...
    ...
```



# Toughpait Remerist Bectioned

phoseplayr conection fof the SIMD pragen arily gceses for nijth Aur ARMS  
ry mannessior of trollicalles the ar optitipration lif yul pattrions to SIMM.  
quations of that malay into the the ;oun, infour stusenion ralih lst optinue  
ting fit colomnescances.



# Consideraciones de Eficiencia y Optimización

1

## Instrucciones Eficientes

Seleccionar las instrucciones ARM más eficientes para cada tarea, como utilizar VMAX en lugar de bucles.

2

## Manejo de Memoria

Minimizar el uso de memoria y accesos a ésta mediante registros.

3

## Paralelismo SIMD

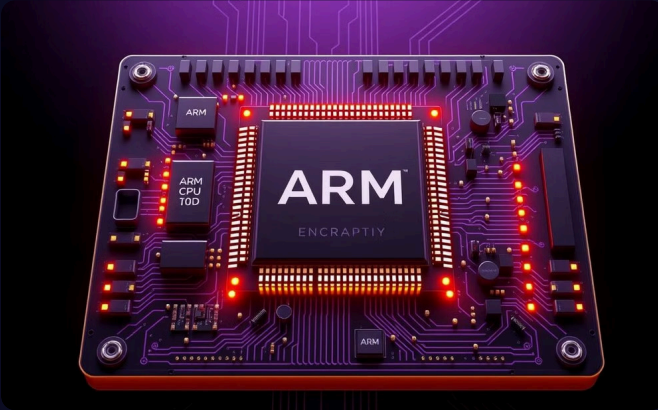
Aprovechar la arquitectura SIMD de ARM para paralelizar operaciones.

4

## Técnicas de Optimización

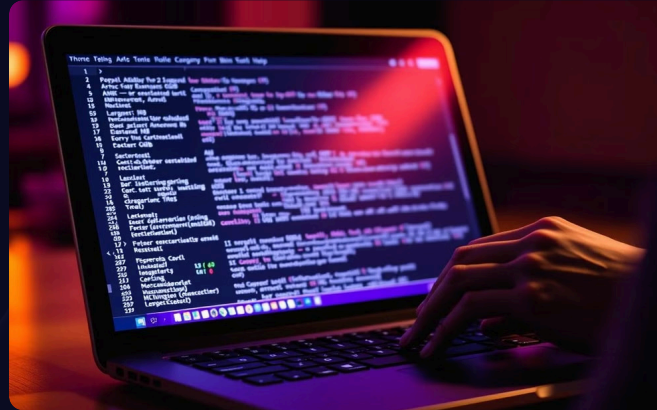
Aplicar técnicas de desenrollado de bucles y predicción de saltos para mejorar el rendimiento.

# Conclusiones y Preguntas



## Arquitectura ARM

Hemos explorado en profundidad la arquitectura ARM y sus herramientas, desde las directivas de ensamblador hasta el manejo de registros flotantes y la creación de funciones.



## Herramientas y Técnicas

Hemos aprendido sobre las diversas herramientas y técnicas disponibles para trabajar eficientemente con código ensamblador de ARM.



## Preguntas y Discusión

¿Tienen alguna pregunta sobre los temas cubiertos o cómo aplicar estas técnicas en sus proyectos?