

# Primeros Pasos Con ARM64

La arquitectura ARM64 ofrece una plataforma poderosa y eficiente para el desarrollo de sistemas embebidos y de alto rendimiento. En esta presentación, exploraremos los fundamentos clave para comenzar a programar en este entorno.

 por Hugo Martinez

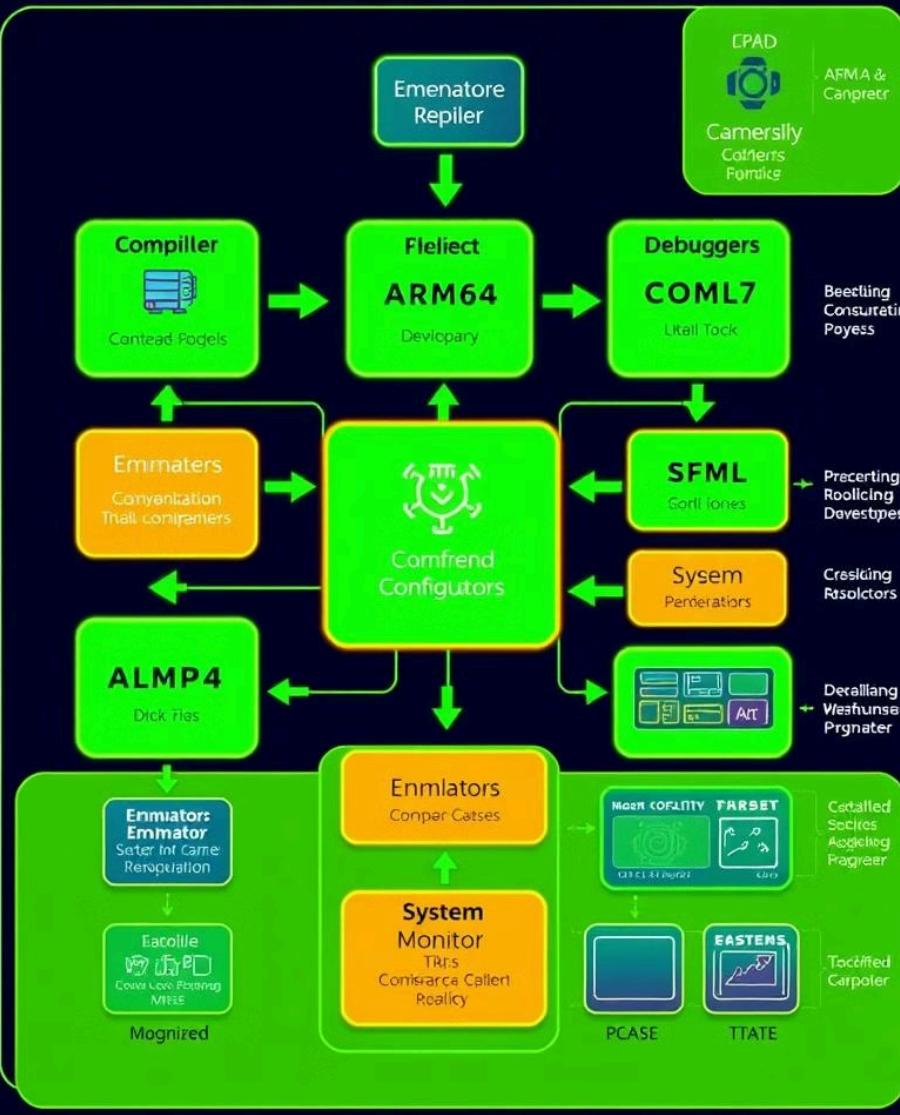




# Introducción a la Arquitectura **ARM64**

**ARM64** es una arquitectura de 64 bits con una variedad de modelos y capacidades para diversos usos. Ofrece un conjunto de instrucciones optimizado para el bajo consumo de energía y el alto rendimiento. Esta arquitectura se utiliza ampliamente en dispositivos móviles, IoT, servidores y sistemas embebidos por sus ventajas de eficiencia y versatilidad.

# ARM64



## Configuración del Entorno de Desarrollo

- Instala un compilador y herramientas de desarrollo compatibles con ARM64, como GCC o Clang.
- Configura el sistema de construcción y las opciones de compilación para aprovechar las características de la arquitectura.
- Utiliza un emulador o una plataforma de desarrollo física para probar y depurar tus aplicaciones ARM64.
- Integra herramientas de perfilado y análisis para optimizar el rendimiento de tu código.

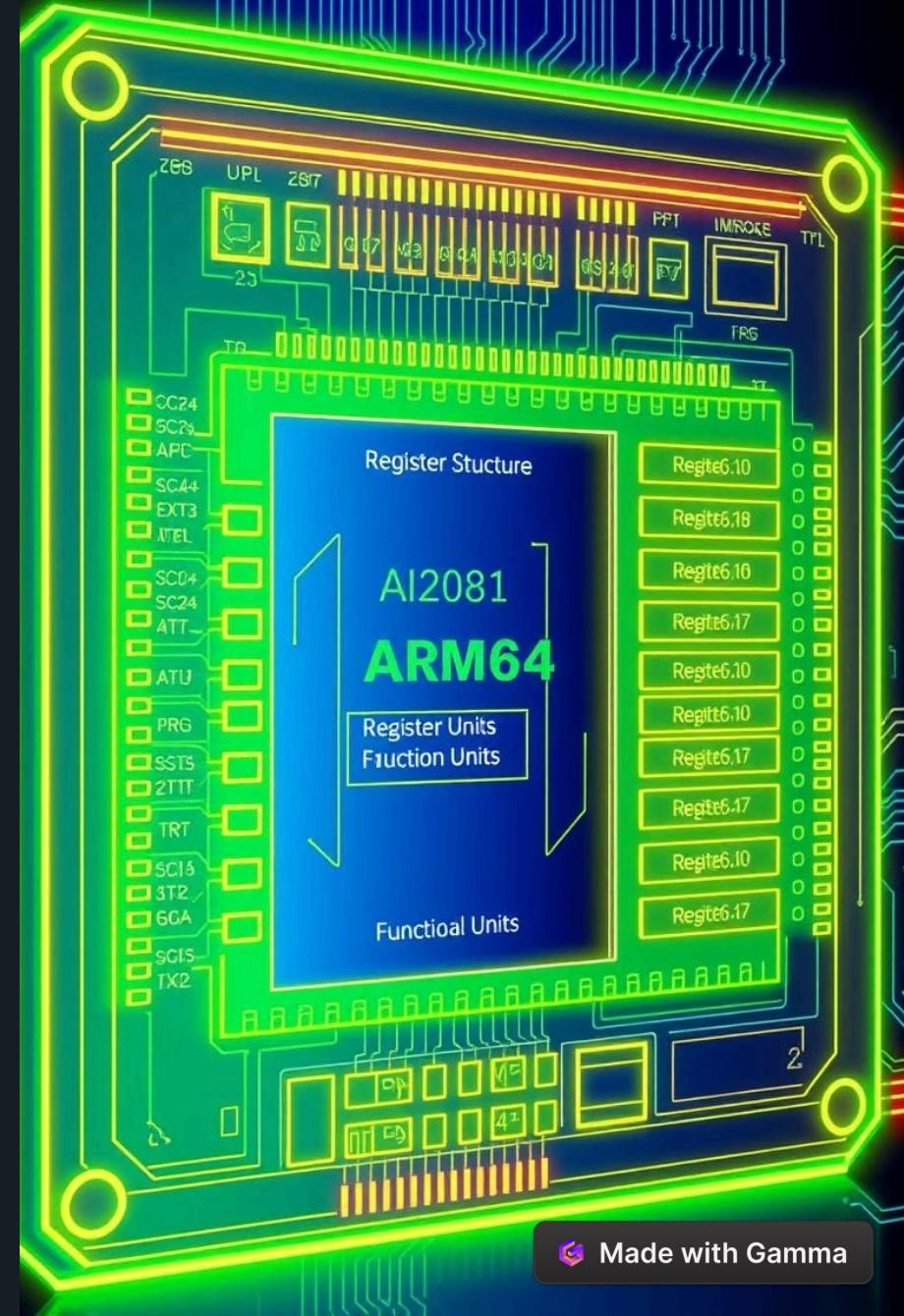
# Conceptos Básicos de Programación en ARM64

## 1 Registros

La arquitectura ARM64 cuenta con un conjunto de 31 registros de propósito general de 64 bits.

## 2 Instrucciones

El conjunto de instrucciones ARM64 incluye operaciones aritméticas, lógicas, de movimiento de datos y de control de flujo.



# **Pre y Post Indexing**

## **Pre-indexing**

El pre-indexing permite cargar o almacenar datos en memoria antes de actualizar el registro base.

## **Post-indexing**

El post-indexing actualiza el registro base después de la carga o almacenamiento de datos en memoria.



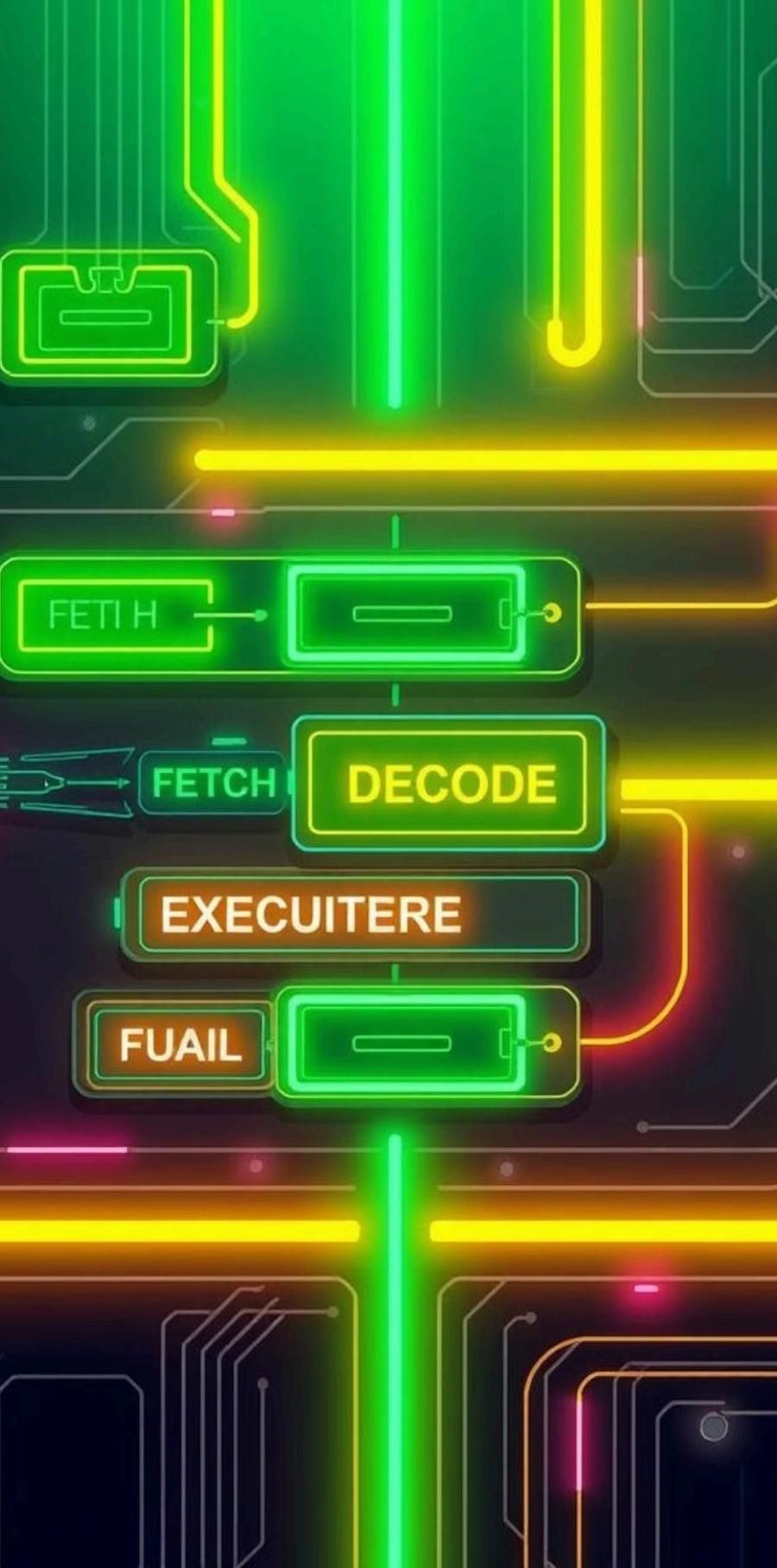
# Índices Basados en Registros

## Cálculo Dinámico

Los índices basados en registros permiten calcular dinámicamente la dirección de memoria a la que se accede.

## Acceso a Estructuras

Esta funcionalidad es especialmente útil para acceder a estructuras de datos y arreglos complejos.



# Ciclos de Instrucciones en ARM64

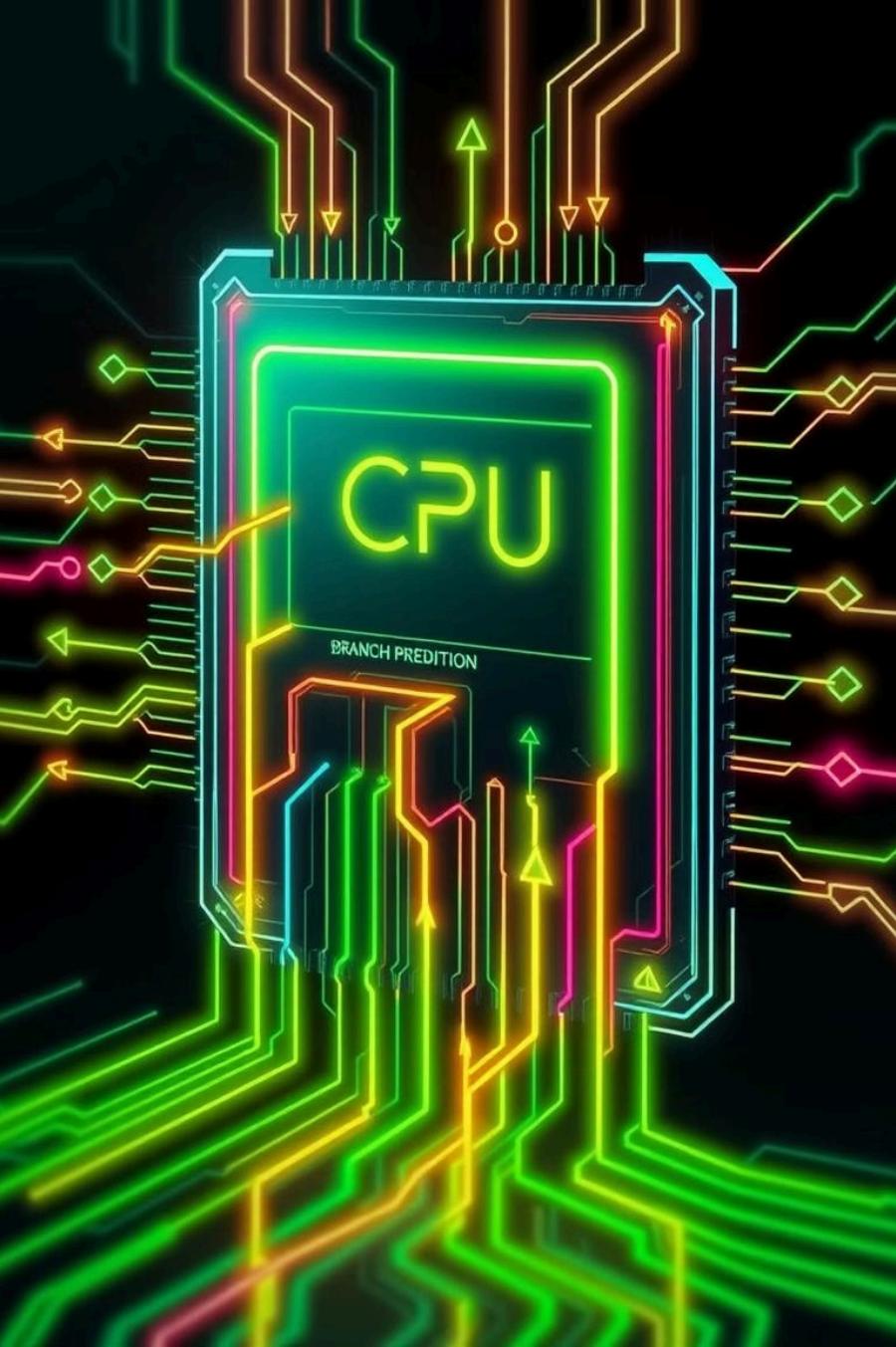
- 1 Etapas de Ejecución**

Las instrucciones en ARM64 se ejecutan en una serie de etapas, como búsqueda, decodificación, ejecución y escritura de resultados.
- 2 Pipeline de Instrucciones**

La pipeline de instrucciones permite procesar múltiples instrucciones simultáneamente, mejorando el rendimiento.
- 3 Unidades Funcionales**

La arquitectura cuenta con diferentes tipos de unidades funcionales para paralelizar la ejecución de instrucciones.
- 4 Optimización**

El rendimiento puede optimizarse mediante el desenrollado de bucles y la predicción de saltos.



# Estructuras de Control de Flujo

- 1
- 2
- 3

## Instrucciones Condicionales

ARM64 proporciona instrucciones condicionales para implementar control de flujo eficiente.

## Estructuras de Control

Bucles y condicionales permiten ejecutar código de manera selectiva y optimizada.

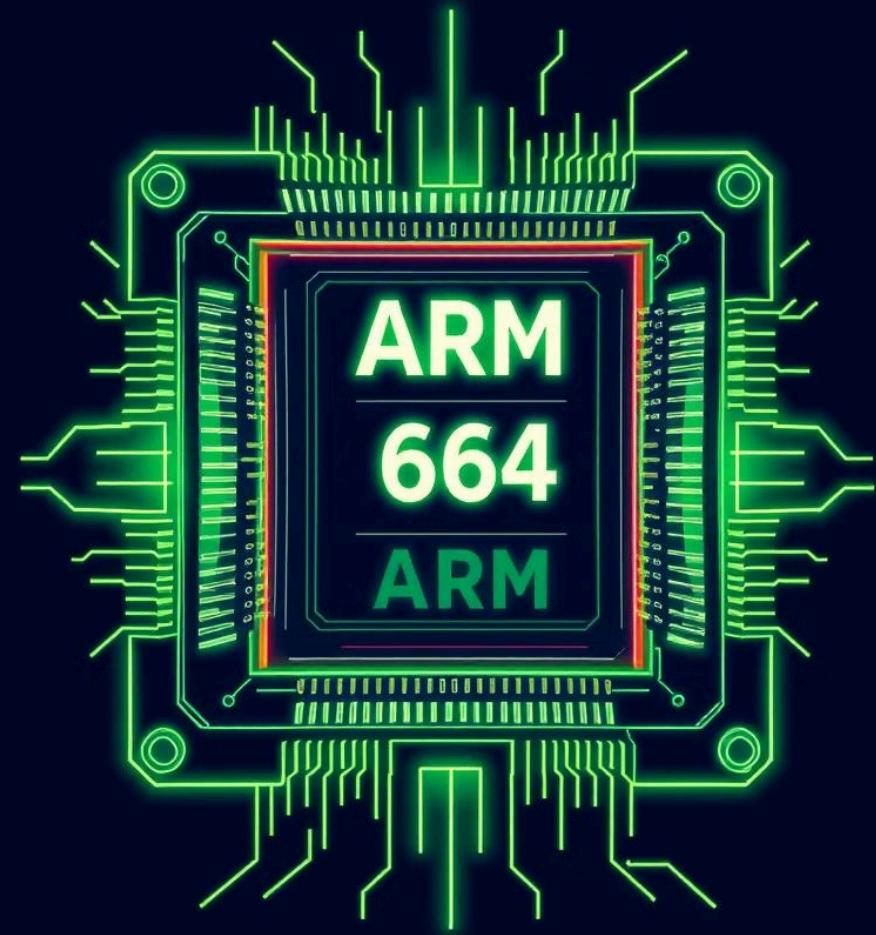
## Saltos y Ramificaciones

Las instrucciones de salto y ramificación permiten implementar el control de flujo requerido.

# Desplazamiento s y Rotaciones

La arquitectura ARM64 permite realizar desplazamientos lógicos y aritméticos de bits para realizar operaciones de enmascaramiento, extracción y ajuste eficientes.

Las instrucciones de rotación permiten manipular de forma circular los bits de los registros, lo que es útil para cálculos y codificaciones.



# Optimización de Código ARM64

- Utiliza técnicas de desenrollado de bucles y predicción de saltos para aprovechar al máximo la pipeline de instrucciones.
- Aplica optimizaciones a nivel de instrucción, como el uso de instrucciones SIMD y de bajo consumo, para mejorar el rendimiento y la eficiencia energética.

