

PROGRAMA DE LABORATORIO

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS



LABORATORIO ARQUITECTURA DE COMPUTADORAS Y ENSAMBLADORES 1 - SECCIÓN A

CÓDIGO:	0778	PUNTEO NETO LABORATORIO:	26
ESCUELA DE INGENIERÍA EN:	CIENCIAS Y SISTEMAS	ÁREA A LA QUE PERTENECE:	CIENCIAS DE LA COMPUTACIÓN
PRE REQUISITO:	0796 Lenguajes Formales y de Programación 0964 Organización Computacional	POST REQUISITO:	0281 Sistemas Operativos 1 0779 Arquitectura de Computadores y Ensambladores 2 0970 Redes de Computadoras 1 2036 Prácticas Intermedias
CATEGORÍA:	OBLIGATORIO/OPTATIVO	VIGENCIA:	SEGUNDO SEMESTRE 2026

Descripción del Laboratorio

El laboratorio del curso “Arquitectura de Computadoras y Ensambladores 1” está diseñado para que el estudiante adquiera y consolide competencias prácticas en el uso de la arquitectura ARM, el lenguaje ensamblador y la interacción directa con hardware, orientadas al desarrollo de sistemas embebidos y aplicaciones IoT.

A través de actividades experimentales y prácticas guiadas, el estudiante analizará el funcionamiento interno de los microcontroladores ARM, incluyendo registros, modos de operación, instrucciones y estructuras de control, aplicando técnicas de programación a bajo nivel para la manipulación de memoria, control del flujo de ejecución y optimización del rendimiento.

Resumen de Ponderaciones y Tiempo de Auto-Aprendizaje

TIPO	CANTIDAD	PONDERACIÓN	HORAS DE AUTO-APRENDIZAJE
Actividades en Clase	10	10	5
Proyectos	2	70	60
Prácticas	1	10	10
Tareas	5	10	18
TOTAL	19	100	93

Desglose de Actividades

TIPO DE ACTIVIDAD	PONDERACIÓN	HORAS DE AUTO-APRENDIZAJE
Proyecto 1	30	25
Proyecto 2	40	35
Práctica Única	10	10
Actividad en Clase (10)	10 (1)	5
Tareas (4)	10 (2)	18
TOTAL	100	93

Equipo Académico

Coordinador del Área

Nombre: M.Sc. Luis Fernando Espino Barrios	Correo electrónico: usac.sistemas@gmail.com
---	--

Docente

Nombre del Docente OTTO RENE ESCOBAR LEIVA	Correo electrónico del Docente otto_escobar@yahoo.com
--	---

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Día		X		X		
Horario		10:40 - 12_20		10:40 - 12_20		
Lugar		T-3 312		T-3 312		

Tutor (es)

Nombre del Tutor	Carlos Alfredo Barrientos López	
Correo electrónico institucional	3016130280101@ingenieria.usac.edu.gt	

Tipo		Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Clase	Día						X
	Horario						13:50-15:30
	Lugar						T-3 209
Atención al Estudiante	Día						X
	Horario						20:00-22:00
	Lugar						Meet

Índice

Descripción del Laboratorio.....	1
Resumen de Ponderaciones y Tiempo de Auto-Aprendizaje.....	1
Desglose de Actividades.....	2
Equipo Académico.....	2
Coordinador del Área.....	2
Docente.....	2
Tutor (es).....	3
Competencias Vinculadas al Perfil del Egresado.....	5
Competencias Específicas.....	5
Competencias Generales.....	5
Competencias del Laboratorio.....	5
Competencia(s) Específica(s).....	5
Competencia(s) General(es).....	6
Diseño Didáctico.....	6
Sesión de Diagnóstico.....	6
Sesión No. 2, Unidad No. 1 - Arquitectura de Computadores.....	7
Sesión No. 3, Unidad No. 1 - Arquitectura de Computadores.....	8
Sesión No. 4, Unidad No. 1 - Arquitectura de Computadores.....	9
Sesión No. 5, Unidad No. 1 - Arquitectura de Computadores.....	10
Sesión No. 6, Unidad No. 2 - Lenguaje Ensamblador ARM-64 bits.....	11
Sesión No. 7, Unidad No. 2 - Lenguaje Ensamblador ARM-64 bits.....	12
Sesión No. 8, Unidad No. 2 - Lenguaje Ensamblador ARM-64 bits.....	13
Sesión No. 9, Unidad No. 2 - Lenguaje Ensamblador ARM-64 bits.....	14
Sesión No. 10, Unidad No. 3 - Programación Híbrida ARM-64 bits.....	15
Sesión No. 11, Unidad No. 3 - Programación Híbrida ARM-64 bits.....	16
Rúbrica de Evaluación.....	17
Normativa Académica y Ética del Curso.....	17
Bibliografía.....	18
E-Grafía.....	19

Competencias Vinculadas al Perfil del Egresado

Competencias Específicas

No.	Competencia
1	Aplica los conocimientos de su disciplina en la elaboración, fundamentación y defensa de argumentos para prevenir y resolver problemas complejos en su campo profesional, identificando y aplicando innovaciones.
2	Identifica oportunidades y riesgos para la innovación y adaptación de conocimientos y tecnologías orientadas a la resolución de problemas.
3	Toma decisiones profesionales con base en fundamentos teóricos, datos e información pertinente, válida y confiable.

Competencias Generales

No.	Competencia
1	Maneja e interpreta adecuadamente datos masivos, estructurados o no estructurados, facilitando su visualización e interpretación eficaz como apoyo para la toma de decisiones.
2	Aplica conocimientos tecnológicos con ética profesional, respetando y cuidando los recursos naturales, humanos y financieros.
3	Construye soluciones integrales mediante el trabajo colaborativo y propositivo en equipos interdisciplinarios, tanto de manera presencial como por medio de plataformas virtuales.

Competencias del Laboratorio

Competencia(s) Específica(s)

No.	Competencia	Nivel de Aprendizaje
1	El estudiante analiza los fundamentos y el modelo de ejecución de arquitecturas modernas, principalmente ARM y RISC-V, mediante el estudio de la CPU, memoria, buses, registros, instrucciones y modelos de arquitectura, para comprender cómo el hardware ejecuta los programas.	Analizar
2	El estudiante diseña soluciones de bajo nivel mediante el desarrollo de algoritmos en lenguaje ensamblador ARM64 y programación híbrida entre C y ARM, para resolver problemas computacionales de manera eficiente.	Crear
3	El estudiante evalúa el rendimiento de programas de bajo nivel mediante la comparación del tiempo de ejecución, ciclos de procesamiento, uso de registros y consumo de memoria, para seleccionar implementaciones eficientes de acuerdo con las características del problema.	Evaluar

4	El estudiante implementa sistemas embebidos orientados al Internet de las Cosas mediante el uso de Raspberry Pi, sensores digitales, actuadores, GPIO y mecanismos de comunicación como MQTT, para resolver necesidades de monitoreo y automatización.	Aplicar
5	El estudiante analiza la integración entre hardware y software mediante el estudio del flujo de instrucciones, la organización de la memoria, el acceso a dispositivos y el control del procesador, para comprender el funcionamiento de aplicaciones en arquitecturas modernas.	Analizar
6	El estudiante Identifica los componentes esenciales de un sistema IoT mediante el análisis de documentación técnica y catálogos de hardware para describir las funcionalidades básicas de cada elemento del sistema.	Recordar

Competencia(s) General(es)

No.	Competencia	Nivel de Aprendizaje
1	El estudiante desarrolla soluciones eficientes en sistemas computacionales mediante la integración de los fundamentos de arquitectura de computadores, programación de bajo nivel, sistemas embebidos y herramientas de análisis y simulación, para resolver problemas complejos relacionados con hardware, software e Internet de las Cosas.	Crear

Diseño Didáctico

Sesión de Diagnóstico

Evaluación de conocimientos previos

Se aplicará una actividad diagnóstica con el objetivo de identificar el nivel de conocimientos y habilidades que los estudiantes poseen al inicio del curso. No influye en la nota final, pero es obligatoria para todos los estudiantes.

Tipo de Actividad	Descripción
Cuestionario interactivo	Evaluación individual realizada a través de una plataforma digital interactiva tipo Kahoot, diseñada para medir el nivel de comprensión del estudiante sobre los temas desarrollados en clases anteriores, con registro automático de resultados.

Presentación del tutor

El tutor se presenta formalmente al grupo, compartiendo su formación académica, experiencia profesional y educativa, así como sus expectativas sobre el curso. También se abordan aspectos como normas de

convivencia, canales de comunicación, disponibilidad para consultas y métodos de acompañamiento.

Presentación de los estudiantes

Se escogen un grupo de estudiantes al azar. En su presentación, se les pedirá que compartan información básica como su nombre, intereses personales o profesionales, experiencias previas relacionadas con el curso y sus expectativas. Esta actividad busca promover la interacción, el reconocimiento entre pares y la construcción de un entorno participativo y respetuoso.

Presentación del programa del curso

Se presenta el contenido del programa del curso, se aclaran dudas y se fomenta el compromiso del estudiante con su aprendizaje.

Evaluación de conocimientos del laboratorio actual

Se realiza una evaluación o práctica que permite conocer el grado de familiaridad de los estudiantes con las herramientas, entornos o competencias técnicas necesarias para el laboratorio actual.

Tipo de Actividad	Descripción
Prueba evaluativa	Evaluación individual realizada a través de una plataforma digital interactiva tipo Kahoot, diseñada para medir el nivel de comprensión del estudiante sobre los temas que se desarrollan en el laboratorio, con registro automático de resultados.

Sesión No. 2, Unidad No. 1 - Arquitectura de Computadores

Área Actitudinal (Saber ser)

Nombre del valor: Curiosidad
Interés por comprender cómo funcionan los sistemas computacionales más allá de su utilización superficial, explorando su estructura interna y su comportamiento.

Área de Conocimiento (Saber)

Competencia(s)
El estudiante comprende e integra los fundamentos de la arquitectura ARM, el funcionamiento de programas en lenguaje ensamblador y la interacción con el hardware mediante el análisis de instrucciones, estructuras de control y manipulación de memoria, para desarrollar soluciones óptimas en sistemas embebidos y entornos IoT.
Analiza los fundamentos de arquitectura de computadores relacionando CPU, memoria, buses, conjuntos de instrucciones y modelos de ejecución, para comprender cómo el hardware ejecuta los programas.

Tema	Subtema
Organización del computador	CPU, memoria y buses
Arquitecturas clásicas	Von Neumann frente a Harvard
Tipos de arquitectura	RISC frente a CISC
Evolución de procesadores	x86 frente a ARM
Ejecución de programas	Ciclo de instrucción
Introducción al bajo nivel	Relación entre hardware y software
Aplicación	Uso de arquitecturas en sistemas reales

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante comprende e integra los fundamentos de la arquitectura ARM, el funcionamiento de programas en lenguaje ensamblador y la interacción con el hardware mediante el análisis de instrucciones, estructuras de control y manipulación de memoria, para desarrollar soluciones óptimas en sistemas embebidos y entornos IoT.	
Tipo de Actividad	Ponderación
Cuestionario	1

Sesión No. 3, Unidad No. 1 - Arquitectura de Computadores

Área Actitudinal (Saber ser)

Nombre del valor: Análisis
Capacidad de interpretar información técnica, relacionar conceptos y comprender el funcionamiento interno de un sistema computacional.

Área de Conocimiento (Saber)

Competencia(s)
Recuerda diferentes dispositivos de electrónica digital que le permitan dar soluciones a sistemas IoT analizando documentación técnica, catálogos de hardware que aporten un valor agregado a la solución de sistemas embebidos y su interacción con el entorno

Analiza el modelo de ejecución de arquitecturas modernas como ARM y RISC-V mediante la representación de datos, los registros, la memoria, el direccionamiento y las instrucciones.	
Tema	Subtema
Representación de datos	Sistemas binario y hexadecimal
Memoria	Jerarquía de memoria
Registros	Función de los registros en la CPU
Direccionamiento	Direcciones, contenido y acceso a memoria
Flujo de ejecución	Program Counter y secuencia de ejecución
Introducción a instrucciones	Qué es una instrucción de máquina
Aplicación	Simulación y ejecución en entornos reales o simulados

Área de Habilidades (Saber Hacer)

Competencia	
Comprende el funcionamiento de la arquitectura ARM, la interacción del hardware con el lenguaje a bajo nivel analizando instrucciones, estructuras de control y manejo de pines multipropósito permitiéndole desarrollar sistemas embebidos y entornos IoT	
Tipo de Actividad	Ponderación
Cuestionario	1

Sesión No. 4, Unidad No. 1 - Arquitectura de Computadores

Área Actitudinal (Saber ser)

Nombre del valor: Precisión
Capacidad de trabajar con exactitud al interpretar y desarrollar procesos técnicos relacionados con instrucciones, registros y operaciones de bajo nivel.

Área de Conocimiento (Saber)

Competencia(s)
Aplica instrucciones básicas de bajo nivel —aritméticas, lógicas, transferencia de datos y control de flujo— para representar algoritmos simples mediante registros y memoria.

Tema	Subtema
Instrucciones	Tipos de instrucciones
Operaciones	Instrucciones aritméticas y lógicas
Transferencia de datos	Modelo LOAD/STORE
Control de flujo	Salto, condicionales y ciclos
Ejecución	Secuencia de instrucciones
Lenguaje de bajo nivel	Introducción al lenguaje ensamblador
Aplicación	Algoritmos simples a nivel de registros

Área de Habilidades (Saber Hacer)

Competencia	
Recuerda diferentes dispositivos de electrónica digital que le permitan dar soluciones a sistemas IoT analizando documentación técnica, catálogos de hardware que aporten un valor agregado a la solución de sistemas embebidos y su interacción con el entorno	
Tipo de Actividad	Ponderación
Cuestionario	1

Sesión No. 5, Unidad No. 1 - Arquitectura de Computadores

Área Actitudinal (Saber ser)

Nombre del valor: Aplicación
Capacidad de trasladar los conocimientos teóricos a un entorno práctico de desarrollo y comprobar su funcionamiento mediante herramientas especializadas.

Área de Conocimiento (Saber)

Competencia(s)
Configura un laboratorio ARM64 reproducible en Linux utilizando QEMU, GNU Binutils, Make y GDB para compilar, ejecutar e inspeccionar programas mínimos.

Tema	Subtema
Laboratorio ARM64	Entorno Linux, WSL o máquina virtual
Toolchain	GNU Assembler, GNU Linker, Objdump y Readelf
Emulación	QEMU para ejecutar programas ARM64
Automatización	Makefile básico
Depuración inicial	GDB y ejecución paso a paso
Estructura de proyecto	Directorios src, build, Makefile y archivos .
Primer programa	Compilar, enlazar y ejecutar un binario mínimo

Área de Habilidades (Saber Hacer)

Competencia	
Aplica conceptos de electrónica para la construcción de prototipos IoT orientados a casos de uso del mundo real mediante placas de desarrollo como raspberry pi, sensores digitales y diseño de sistemas para la automatización de procesos logrando mayor eficiencia.	
Tipo de Actividad	Ponderación
Ejercicio práctico	1

Sesión No. 6, Unidad No. 2 - Lenguaje Ensamblador ARM-64 bits

Área Actitudinal (Saber ser)

Nombre del valor: Aplicación
Capacidad de utilizar los conocimientos fundamentales de arquitectura para comenzar a desarrollar programas reales en lenguaje ensamblador ARM64.

Área de Conocimiento (Saber)

Competencia(s)
Identifica los conceptos fundamentales de ARM64/AArch64, el vocabulario del lenguaje ensamblador, los tipos de datos, los registros y la estructura básica de un archivo .s.

Tema	Subtema
Introducción a ARM64	Ensamblador, ISA, ARMv8-A, AArch64 y A64
Introducción a ARM64	Vocabulario básico y tipos de datos
Modelo AArch64	Registros X/W, SP, LR, PC y PSTATE
GNU Assembly	Estructura de archivos .s, secciones y símbolos

Área de Habilidades (Saber Hacer)

Competencia	
Comprende el funcionamiento de la arquitectura ARM, la interacción del hardware con el lenguaje a bajo nivel analizando instrucciones, estructuras de control y manejo de pines multipropósito permitiéndole desarrollar sistemas embebidos y entornos IoT.	
Tipo de Actividad	Ponderación
Cuestionario	1

Sesión No. 7, Unidad No. 2 - Lenguaje Ensamblador ARM-64 bits

Área Actitudinal (Saber ser)

Nombre del valor: Precisión
Exactitud al escribir, revisar y ejecutar instrucciones a nivel de máquina.

Área de Conocimiento (Saber)

Competencia(s)	
Aplica el conjunto de instrucciones AArch64 utilizando movimientos, operaciones aritméticas, lógicas, carga, almacenamiento y modos de direccionamiento para manipular datos en registros y memoria.	
Tema	Subtema
Conjunto de instrucciones ARM	Movimiento de constantes y registros
Conjunto de instrucciones ARM	Instrucciones aritméticas

Conjunto de instrucciones ARM	Instrucciones lógicas y máscaras
Memoria y direccionamiento	Carga, almacenamiento, pre-indexing y post-indexing

Área de Habilidades (Saber Hacer)

Competencia	
Aplica instrucciones de movimiento, aritmética, lógica, carga y almacenamiento en programas funcionales escritos en ARM64.	
Tipo de Actividad	Ponderación
Actividad	1

Sesión No. 8, Unidad No. 2 - Lenguaje Ensamblador ARM-64 bits

Área Actitudinal (Saber ser)

Nombre del valor:
Capacidad de estructurar soluciones mediante condiciones, comparaciones y procesos de repetición.

Área de Conocimiento (Saber)

Competencia(s)	
Analiza el comportamiento de la arquitectura ARM utilizando registros, instrucciones y diagramas de bloques optimizando programas a bajo nivel en microprocesadores.	
Tema	Subtema
Operaciones sobre bits	Desplazamientos, rotaciones y extensiones
Banderas	NZCV y comparación mediante CMP
Selección condicional	Condiciones, CSEL y estructuras if/else
Branching y ciclos	B, BL, BR, ciclos y etiquetas

Área de Habilidades (Saber Hacer)

Competencia	
Analiza el comportamiento de la arquitectura ARM utilizando registros, instrucciones y diagramas de bloques optimizando programas a bajo nivel en microprocesadores.	
Tipo de Actividad	Ponderación
Actividad	1

Sesión No. 9, Unidad No. 2 - Lenguaje Ensamblador ARM-64 bits

Área Actitudinal (Saber ser)

Nombre del valor:
Comprensión de la manera en que un programa se comunica con el sistema operativo y utiliza sus servicios.

Área de Conocimiento (Saber)

Competencia(s)	
Analiza el comportamiento de la arquitectura ARM utilizando registros, instrucciones y diagramas de bloques optimizando programas a bajo nivel en microprocesadores	
Tema	Subtema
Excepciones	Exception Levels y SVC
Syscalls	Contrato de Linux: X8, X0–X5, valores de retorno y errores
GNU Assembly	Directivas, datos, literales y pseudoinstrucciones
Binarios ARM64	Ensamblado, enlazado, formato ELF y ejecución

Área de Habilidades (Saber Hacer)

Competencia	
Analiza el comportamiento de la arquitectura ARM utilizando registros, instrucciones y diagramas de bloques optimizando programas a bajo nivel en microprocesadores.	
Tipo de Actividad	Ponderación
Cuestionario	1

Sesión No. 10, Unidad No. 3 - Programación Híbrida ARM-64 bits

Área Actitudinal (Saber ser)

Nombre del valor:
La depuración con GDB, la construcción de funciones y el manejo del stack requieren pruebas constantes. El estudiante comprende que la calidad de un programa se alcanza mediante la revisión y corrección progresiva de su funcionamiento.

Área de Conocimiento (Saber)

Competencia(s)	
Crea crea soluciones innovadoras desarrollando algoritmos mediante aritmética, lógica y control del flujo que permitan resolver problemas computacionales orientado al análisis de datos de sistemas embebidos.	
Tema	Subtema
Depuración	GDB: breakpoints, registros, memoria y ejecución paso a paso
Funciones	BL, LR, RET y creación de funciones
STACK de ARM	SP, FP y stack frames
STACK de ARM	Prólogo, epílogo y variables locales

Área de Habilidades (Saber Hacer)

Competencia	
Construye y depura funciones reutilizables en ensamblador ARM64 aplicando paso de parámetros, valores de retorno, preservación de registros y manejo del stack.	
Tipo de Actividad	Ponderación
Cuestionario	1

Sesión No. 11, Unidad No. 3 - Programación Híbrida ARM-64 bits

Área Actitudinal (Saber ser)

Nombre del valor:
Capacidad de combinar lenguajes, herramientas y niveles de abstracción para resolver problemas reales.

Área de Conocimiento (Saber)

Competencia(s)	
Evalúa diferentes soluciones a bajo nivel en base a los recursos disponibles y encontrando la eficiencia determinando la alternativa más óptima según el entorno en donde se esté desarrollando.	
Tema	Subtema
Convenciones de llamada	Argumentos, valores de retorno y registros caller-saved y callee-saved
Compatibilidad de ARM Con C	Desarrollo de funciones ARM llamadas desde C
Compatibilidad de ARM Con C	Integración de bloques ARM en C
Compatibilidad de ARM Con C	Llamadas a funciones de C desde ARM y uso básico de libc

Área de Habilidades (Saber Hacer)

Competencia	
Integra funciones desarrolladas en ensamblador ARM64 con programas escritos en C, respetando las convenciones de llamada y el intercambio de parámetros y resultados.	
Tipo de Actividad	Ponderación
Cuestionario	1

Rúbrica de Evaluación

Cada una de las actividades del laboratorio (proyectos, prácticas, tareas y otras) cuenta con una rúbrica de evaluación específica, la cual está detallada en el documento que se entrega al estudiante al momento de asignar la actividad. Estas rúbricas describen los criterios de evaluación, niveles de desempeño esperados y la ponderación correspondiente de cada aspecto evaluado.

Es **responsabilidad del estudiante** leer detenidamente la rúbrica asignada antes de iniciar el desarrollo de la actividad. Comprender los criterios de evaluación no solo permite orientar adecuadamente el trabajo, sino también mejorar el desempeño académico y fomentar la autorregulación del aprendizaje.

En caso de no recibir la rúbrica al momento de la asignación, el estudiante **debe solicitarla directamente al tutor académico**, ya que constituye una herramienta esencial para el cumplimiento de los objetivos de aprendizaje y la evaluación transparente.

Normativa Académica y Ética del Curso

En concordancia con el perfil del estudiante de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, se espera un alto nivel de compromiso con la excelencia académica y la ética profesional. Por ello, que se establece los siguientes lineamientos de carácter obligatorio que regulan el comportamiento académico del estudiante:

Plagio y copias

- Todo proyecto será sometido a verificación para confirmar su autoría y originalidad, con la finalidad de evitar cualquier plagio, copia o que la actividad no haya sido realizada por el estudiante.
- Cualquier evidencia de lo antes descrito en las distintas actividades será sancionada con una calificación de 0 (cero) y el caso será reportado al Docente quien a su vez informará a la Escuela de Ciencias y Sistemas para su seguimiento institucional.

Prórrogas y reposiciones

- No se otorgarán prórrogas para entregas de actividades.
- No se permitirá la reposición de proyectos bajo ninguna circunstancia.

Requisitos para evaluación final del curso

- Es obligatorio aprobar el laboratorio para tener derecho a la evaluación final del curso.

- La calificación de prácticas, proyectos y otras actividades que se indique será asignada de forma presencial, en la fecha y hora establecidas por el tutor académico.

Asistencia

- Para obtener la nota del laboratorio, se requiere un mínimo del 80% de asistencia a las sesiones de laboratorio.
- En caso de inasistencia, sólo se aceptarán justificaciones válidas respaldadas por constancia oficial.

Entregas

- No se aceptarán entregas tardías de tareas, prácticas, exámenes cortos, exámenes finales o proyectos sin justificación.

Medio oficial de entrega

- La plataforma UEDI de la Facultad será el único medio oficial para la entrega de actividades del curso.

Bibliografía

- Patterson, D. A., & Hennessy, J. L. (2021).
Computer Organization and Design: The Hardware/Software Interface (ARM Edition).
2nd ed. Morgan Kaufmann.
ISBN: 978-0128203316
- Furber, S. (2020).
ARM System-on-Chip Architecture.
2nd ed. Addison-Wesley.
ISBN: 978-0201675192
- Sloss, A., Symes, D., & Wright, C. (2012).
ARM System Developer's Guide: Designing and Optimizing System Software.
Morgan Kaufmann.
ISBN: 978-1558608740
- Barrett, S. F., & Pack, D. J. (2016)
Microcontrollers Fundamentals for Engineers and Scientists.
Morgan & Claypool Publishers.
ISBN: 978-1627055078
- Monk, S. (2016).
Programming the Raspberry Pi: Getting Started with Python
McGraw-Hill Education.
ISBN: 978-1259587544
- Banzi, M., & Shiloh, M. (2022)
Getting Started with IoT.
Maker Media, Inc.
ISBN: 978-1680456924

E-Grafía

- ARM Ltd.
ARM Architecture Reference Manual for ARMv8-A.
<https://developer.arm.com/documentation>
(Documentación oficial sobre arquitectura ARM, instrucciones, registros y modos de operación)
- Raspberry Pi Foundation.
Raspberry Pi Documentation.
<https://www.raspberrypi.com/documentation/>
(Guías oficiales de hardware, GPIO, sensores y sistemas embebidos)
- GNU Project.
GNU Assembler (GAS) Documentation.
<https://sourceware.org/binutils/docs/as/>
(Referencia oficial del ensamblador utilizado en ARM/Linux)
- Free Software Foundation.
GDB: The GNU Project Debugger.
<https://www.gnu.org/software/gdb/documentation/>
(Uso de GDB para depuración de programas ARM)
- IBM Cloud Education.
What is the Internet of Things (IoT)?
<https://www.ibm.com/topics/internet-of-things>
(Conceptos fundamentales, arquitectura y aplicaciones de IoT)
- Linux Foundation.
Embedded Linux Wiki.
<https://elinux.org>
(Recursos técnicos sobre Linux embebido y desarrollo en ARM)