

## PROGRAMA GENERAL PARA ALCANZAR COMPETENCIAS EN “DISEÑO DE SISTEMAS CON PROCESADORES”

<b>COMPETENCIA 1:</b>	Representar cualquier dato numérico y alfabético mediante una codificación binaria.	<b>CAPÍTULO 1. CODIFICACIÓN DE BAJO NIVEL</b> 1.1. Sistemas de Numeración 1.2. Representación de Números en un Sistema Digital 1.3. Formatos Estándar IEEE-754 de Punto Flotante 1.4. Codificación Binaria 1.5. Secuencias de Conteo
<b>COMPETENCIA 2:</b>	Hacer ingeniería inversa a un código máquina para entender su código fuente con propósitos académicos, considerando sus implicaciones éticas.	<b>CAPÍTULO 2. PROGRAMACIÓN EN BAJO NIVEL</b> 2.1. Herramientas de Desarrollo en C 2.2. Control de Flujo en Bajo Nivel 2.3. Casos Especiales de Control de Flujo 2.4. Operaciones de la ALU: <i>Expresiones en C</i> 2.5. Tipos de Datos en Memoria: <i>Declaraciones en C</i> 2.6. Llamado con Retorno y Contexto en Pila: <i>Funciones en C</i>
<b>COMPETENCIAS 3:</b>	Analizar y diseñar un sistema de computador embebido.	<b>CAPÍTULO 3. METODOLOGÍAS DE DISEÑO</b> 3.1. Ingeniería de Software y de Hardware 3.2. Elaboración y Lectura de Diagramas 3.3. Sistemas de Computador Embebido 3.4. Configuración de Periféricos 3.5. Protocolos de Comunicación
<b>COMPETENCIAS 4:</b>	Diseñar, adaptar e implementar en software cualquier algoritmo mediante una “ <i>Finite State Machine</i> ”.	<b>CAPÍTULO 4. MÁQUINAS DE ESTADOS FINITOS</b> 4.1. Diagramas de Estado 4.2. Tipos de Máquinas de Estados Finitos 4.3. Implementación de Máquinas de Estados Finitos 4.4. Máquinas de Estados Finitos Concurrentes
<b>COMPETENCIA 5:</b>	Justificar la elección de una arquitectura de software embebido, acorde a los requerimientos del diseño.	<b>CAPÍTULO 5. INTRODUCCIÓN A RTOS</b> 5.1. Interrupciones Hardware 5.2. Arquitecturas de Software 5.3. Servicios de una Arquitectura RTOS 5.4. Otros Servicios del Sistema Operativo 5.5. Uso de un Diseño Básico con RTOS

## MATERIAL BIBLIOGRÁFICO DE APOYO EN CADA SECCIÓN DE “DISEÑO DE SISTEMAS CON PROCESADORES”

CAPÍTULO 1. CODIFICACIÓN DE BAJO NIVEL	MATERIAL BIBLIOGRÁFICO DE APOYO EN CADA SECCIÓN
<b>1.1. Sistemas de Numeración</b> 1.1.1. Historia de los sistemas de numeración 1.1.2. Sistema numérico posicional 1.1.3. Conversión entre bases numéricas	<b>Section 1.1:</b> Number System [Giraldo, 2023].
<b>1.2. Representación de Números en un Sistema Digital</b> 1.2.1. Precisión y rango dinámico 1.2.2. Representación de números negativos 1.2.3. Representación en punto fijo y punto flotante 1.2.4. Representación logarítmica 1.2.5. Aritmética binaria	<b>Section 1.1:</b> Number System [Giraldo, 2023]. <b>Section 1.4:</b> Number Systems [Harris & Harris, 2013], pp. 9-18
<b>1.3. Formatos Estándar IEEE-754 de Punto Flotante</b> 1.3.1. Justificación, tipos de formatos y actualizaciones 1.3.2. Codificación de los formatos 1.3.3. Valores y operaciones especiales 1.3.4. Operaciones aritméticas	<b>Section 1.2:</b> Real Number Representation [Giraldo, 2023]. <b>Section 5.3.2:</b> Floating-Point Number Systems [Harris & Harris, 2013], pp. 256-259 <b>Document:</b> IEEE Standard for Floating-Point Arithmetic, 1985. <b>Document:</b> IEEE Standard for Radix-Independent Floating-Point Arithmetic , 1987. <b>Document:</b> IEEE Standard for Floating-Point Arithmetic (Revision), 2008. <b>Document:</b> IEEE Standard for Floating-Point Arithmetic (Revision of IEEE 2008), 2019.
<b>1.4. Codificación Binaria</b> 1.4.1. Justificación y usos 1.4.2. Códigos ponderados, no-ponderados, auto-complementados 1.4.3. Códigos alfanuméricos 1.4.4. Códigos detectores y correctores de error	<b>Section 1.3:</b> Binary Codes [Giraldo, 2023].
<b>1.5. Secuencias de Conteo</b> 1.5.1. Secuenciadores hardware 1.5.2. Contadores binario y Gray 1.5.3. Contadores de anillo y Johnson 1.5.4. Contadores pseudo-aleatorios	<b>Section 1.4:</b> Counting [Giraldo, 2023].

CAPÍTULO 2. PROGRAMACIÓN EN BAJO NIVEL	MATERIAL BIBLIOGRÁFICO DE APOYO EN CADA SECCIÓN
<b>2.1. Herramientas de Desarrollo en C</b> 2.1.1. Sistema operativo por consola 2.1.2. El pre-procesador de C 2.1.3. Etapas de compilación en C 2.1.4. Herramientas software 2.1.5. Herramientas hardware	<b>Section 6.6.2:</b> Translating and Starting a Program [Harris & Harris, 2013], pp. 337-341 <b>Section 2.2.13:</b> The pre-processor [Harder et al., 2014], pp. 43-45 <b>Chapter 9:</b> Embedded Software Development Tools [Simon, 1999], pp. 261-281 <b>Chapter 10:</b> Debugging Techniques [Simon, 1999], pp. 283-328 <b>Chapter 5:</b> C Debugging Techniques [Anderson & Anderson, 1988], pp. 262-362
<b>2.2. Control de Flujo en Bajo Nivel</b> 2.2.1. Diagramas de flujo 2.2.2. Instrucciones, condiciones y secuencias 2.2.3. Bloque de instrucciones (delimitadores "{" y "}") 2.2.4. Secuencias lineales (separador ";") 2.2.5. Secuencias de selección ("if-else", "if") 2.2.6. Secuencias de repetición ("for", "while", "do-while")	Se seguirá material desarrollado por el Ing. Juan-Carlos Giraldo <b>Chapter 1:</b> C Refresher [Anderson & Anderson, 1988], pp. 1-46
<b>2.3. Casos Especiales de Control de Flujo</b> 2.3.1. Árboles de decisión 2.3.2. Selección con "switch-case" 2.3.3. Optimización de bucle 2.3.4. Saltos locales ("goto-label", "break", "continue", "return") 2.3.5. Saltos no-locales (setjmp.h) 2.3.6. Resumen: "Secuencias en Lenguajes de Ensamble"	Se seguirá material desarrollado por el Ing. Juan-Carlos Giraldo <b>Chapter 1:</b> C Refresher [Anderson & Anderson, 1988], pp. 1-46 <b>Chapter 4:</b> Static memory allocation [Harder et al., 2014], pp. 97-104 <b>Section 2.1.1:</b> Programming paradigms [Harder et al., 2014], pp. 14-15 <b>Section 2.2.11:</b> Switch statements versus functions pointers [Harder et al., 2014], pp. 42 <b>Section 4.3:</b> Set jump and long jump [Harder et al., 2014], pp. 102-103
<b>2.4. Operaciones de la ALU: <i>Expresiones en C</i></b> 2.4.1. Operandos y operadores 2.4.2. Precedencia 2.4.3. Asociación 2.4.4. Evaluación 2.4.5. Promoción 2.4.6. Puntos de secuencia garantizada	Se seguirá material desarrollado por el Ing. Juan-Carlos Giraldo <b>Chapter 1:</b> C Refresher [Anderson & Anderson, 1988], pp. 1-46

<b>2.5. Tipos de Datos en Memoria: <i>Declaraciones en C</i></b> 2.5.1. Atributos de tipos de datos 2.5.2. Tipos escalares, agregados y derivados 2.5.3. Apuntadores 2.5.4. Estructuras 2.5.5. Arreglos 2.5.6. Declaraciones complejas	<b>Chapter 3:</b> An Array of Choices [Anderson & Anderson, 1988], pp. 112-208 <b>Chapter 4:</b> A Closer look at C [Anderson & Anderson, 1988], pp. 217-230 <b>Section 2.2.10:</b> Bit-fields in C99 [Harder et al., 2014], pp. 41
<b>2.6. Llamado con Retorno y Contexto en Pila: <i>Funciones en C</i></b> 2.6.1. Parámetros y argumentos 2.6.2. Paso de parámetros por valor y por referencia 2.6.3. Tipos de almacenamiento ("auto", "static", "register", "extern") 2.6.4. Retorno de valores de la función 2.6.5. Uso del " <i>stack</i> " y el " <i>stack frame</i> "	<b>Chapter 2:</b> The Run Time Environment [Anderson & Anderson, 1988], pp. 47-111 <b>Chapter 4:</b> Static memory allocation [Harder et al., 2014], pp. 97-101

CAPÍTULO 3. METODOLOGÍAS DE DISEÑO	MATERIAL BIBLIOGRÁFICO DE APOYO EN CADA SECCIÓN
<b>3.1. Ingeniería de Software y de Hardware</b> 3.1.1. Métricas de complejidad del software 3.1.2. Fases de diseño de sistemas 3.1.3. Diseño " <i>top-down</i> ", " <i>bottom-up</i> " y diseño en "V" 3.1.4. Desarrollo iterativo e incremental 3.1.5. Metodologías " <i>Waterfall</i> " y " <i>Agile</i> "	<b>Chapter 1:</b> Software engineering in the UNIX/C environment [Frakes et al., 1991], pp. 1-9 <b>Chapter 2:</b> Real-time, embedded and operating-system programming languages [Harder et al., 2014], pp. 13-72 <b>Section 2.3:</b> Software engineering and development [Harder et al., 2014], pp. 43-45
<b>3.2. Elaboración y Lectura de Diagramas</b> 3.2.1. Diagramas de bloques 3.2.2. Diagramas de tiempo 3.2.3. Diagramas esquemáticos 3.2.4. Caso de estudio: "La Arquitectura de un Computador" 3.2.5. Aplicación: "Uso de Manuales de Especificaciones"	Se seguirá material desarrollado por el Ing. Juan-Carlos Giraldo <b>Chapter 2:</b> Hardware Fundamentals for the Software Engineer [Simon, 1999], pp. 13-44 <b>Chapter 3:</b> Computer organizations [Harder et al., 2014], pp. 73-96 <b>Chapter 4:</b> Architecture [Harris & Harris, 2013], pp. 295-370 <b>Section 6.8:</b> Real-World Perspective: x86 Architecture [Harris & Harris, 2013], pp. 347-356
<b>3.3. Sistemas de Computador Embebido</b> 3.3.1. Definición y usos 3.3.2. Atributos de calidad 3.3.3. Microprocesadores y buses 3.3.4. Metodología de diseño en capas 3.3.5. Casos de estudio: "Análisis de Sistemas Embebidos"	<b>Chapter 1:</b> A First Look at Embedded Systems [Simon, 1999], pp. 1-11 <b>Chapter 3:</b> Advanced Hardware Fundamentals [Simon, 1999], pp. 45-79 <b>Chapter 1:</b> Introduction to real-time systems [Harder et al., 2014], pp. 1-12 <b>Section 1.1:</b> What is a real-time system? [Harder et al., 2014], pp. 1-3 <b>Section 8.6:</b> Embedded I/O Systems [Harris & Harris, 2013], pp. 508-557
<b>3.4. Configuración de Periféricos</b> 3.4.1. Puertos paralelos 3.4.2. Temporizadores 3.4.3. Generadores " <i>Pulse-Width Modulation</i> " (PWM) 3.4.4. Conversores " <i>Analog-to-Digital</i> " (ADC) 3.4.5. Puertos seriales	<b>Section 8.6:</b> Embedded I/O Systems [Harris & Harris, 2013], pp. 508-557 <b>Section 8.6.4:</b> Timers [Harris & Harris, 2013], pp. 527-529 <b>Section 6.6.6:</b> Analog I/O [Harder et al., 2014], pp. 531-536 <b>Section 6.6.6:</b> Analog I/O - Pulse-Width Modulation [Harder et al., 2014], pp. 536-537 <b>Section 6.6.7:</b> Other Microcontroller Peripherals [Harder et al., 2014], pp. 537-558 Datasheets
<b>3.5. Protocolos de Comunicación</b> 3.5.1. Comunicaciones " <i>on-board</i> " 3.5.2. Comunicaciones entre dispositivos 3.5.3. Comunicaciones inalámbricas	<b>Section 8.6.3:</b> Serial I/O [Harris & Harris, 2013], pp. 515-527 Datasheets

CAPÍTULO 4. MÁQUINAS DE ESTADOS FINITOS	MATERIAL BIBLIOGRÁFICO DE APOYO EN CADA SECCIÓN
<b>4.1. Diagramas de Estado</b> 4.1.1. Teoría de grafos 4.1.2. Representación matricial de grafos 4.1.3. Aplicaciones y usos de grafos 4.1.4. Diagramas de estado como grafos 4.1.5. Diagramas de estado bien estructurados	Se seguirá material desarrollado por el Ing. Juan-Carlos Giraldo
<b>4.2. Tipos de Máquinas de Estados Finitos</b> 4.2.1. Definiciones formales 4.2.2. Taxonomía de autómatas según existencia de E/S 4.2.3. Taxonomía de máquinas traductoras 4.2.4. Taxonomía de máquinas según memoria 4.2.5. Taxonomía de máquinas según naturaleza determinística	Se seguirá material desarrollado por el Ing. Juan-Carlos Giraldo
<b>4.3. Implementación de Máquinas de Estados Finitos</b> 4.3.1. Implementaciones software vs. hardware 4.3.2. Implementación con "switch-case" 4.3.3. Implementación con "goto-label" 4.3.4. Implementación con declaraciones complejas	Se seguirá material desarrollado por el Ing. Juan-Carlos Giraldo
<b>4.4. Máquinas de Estados Finitos Concurrentes</b> 4.4.1. Preservación del contexto en tareas 4.4.2. Contexto privado con tipo de almacenamiento "static" 4.4.3. Contexto en estructuras externas pasadas por referencia	Se seguirá material desarrollado por el Ing. Juan-Carlos Giraldo

CAPÍTULO 5. INTRODUCCIÓN A RTOS	MATERIAL BIBLIOGRÁFICO DE APOYO EN CADA SECCIÓN
<b>5.1. Interrupciones Hardware</b> 5.1.1. Conceptos básicos sobre interrupciones 5.1.2. Problemas de datos compartidos 5.1.3. Latencia de interrupción	<b>Chapter 4:</b> Interrupts [Simon, 1999], pp. 81-112 <b>Chapter 8:</b> Hardware interrupts [Harder et al., 2014], pp. 235-249 <b>Section 8.6.4:</b> Timers [Harris & Harris, 2013], pp. 527-529
<b>5.2. Arquitecturas de Software</b> 5.2.1. "Round-robin" 5.2.2. "Round-robin" con interrupciones 5.2.3. "Function-Queue-Scheduling" 5.2.4. Sistema Operativo en Tiempo-Real (RTOS)	<b>Chapter 5:</b> Survey of Software Architectures [Simon, 1999], pp. 115-134 <b>Section 6.6.5:</b> Interrupts [Harder et al., 2014], pp. 529-531
<b>5.3. Servicios de una Arquitectura RTOS</b> 5.3.1. Tareas y estados de las tareas 5.3.2. Tareas y datos 5.3.3. Semáforos y datos compartidos	<b>Chapter 6:</b> Introduction to Real-Time Operating Systems [Simon, 1999], pp. 137-169
<b>5.4. Otros Servicios del Sistema Operativo</b> 5.4.1. "Message-Queues", "Mailboxes" y "Pipes" 5.4.2. Temporización de funciones 5.4.3. Eventos 5.4.4. Manejo de memoria 5.4.5. Rutinas de interrupción en un RTOS	<b>Chapter 7:</b> More Operating System Services [Simon, 1999], pp. 173-207
<b>5.5. Uso de un Diseño Básico con RTOS</b> 5.5.1. Encapsulamiento de semáforos y colas 5.5.2. Consideraciones de planeación para tiempo real duro 5.5.3. Ahorro de espacio de memoria 5.5.4. Ahorro de consumo de potencia	<b>Chapter 8:</b> Basic Design Using a Real-Time Operating System [Simon, 1999], pp. 215-260 <b>Chapter 9:</b> Synchronization [Harder et al., 2014], pp. 251-318

## Referencias:

- [Anderson & Anderson, 1988] Paul L. Anderson & Gail C. Anderson, *“Advanced C Tips and Techniques”*, Hayden Books, 1988.
- [Frakes et al., 1991] William B. Frakes, Christopher J. Fox, Brian A. Nejmeh, *“Software engineering in the UNIX/C environment”*, Published by Prentice Hall, Inc., 1991.
- [Giraldo, 2023] Juan-Carlos Giraldo, *“Low Level Programming in C”*, Pontificia Universidad Javeriana, 2023.
- [Harder et al., 2014] Douglas Wilhem Harder, Jeff Zarnett, Vajih Montaghmi, Allyson Giannikouris, *“A practical introduction to real-time systems for undergraduate engineering”*, University of Waterloo, 2014.
- [Harris & Harris, 2013] David Money Harris & Sarah L. Harris, *“Digital Design and Computer Architecture”*, Elsevier, 2013.
- [IEEE Std 754-1985] IEEE Computer Society, *“IEEE Standard for Floating-Point Arithmetic”*, 1985.
- [IEEE Std 854-1987] IEEE Computer Society, *“IEEE Standard for Radix-Independent Floating-Point Arithmetic”*, 1987.
- [IEEE Std 754-2008] IEEE Computer Society, *“IEEE Standard for Floating-Point Arithmetic (Revision)”*, 2008.
- [IEEE Std 754-2019] IEEE Computer Society, *“IEEE Standard for Floating-Point Arithmetic (Revision of IEEE 754-2008)”*, 2019.
- [ISO/IEC 9899] ISO (International Organization for Standardization) and IEC (the International Electrotechnical Commission), *“The C Language Standard”*, 2005.  
<http://www.open-std.org/JTC1/SC22/WG14/www/docs/n1124.pdf>



[Microchip, 2007]

Microchip Technology Incorporated, "*Introduction to Embedded Programming Using C*", 2007.

[Simon, 1999]

David E. Simon, "*An Embedded Software Primer*", by Pearson Education, Inc., 1999.