

| | | | |
|-------------------------------|---|--|-----------------------------------|
| Nombre Asignatura: | Diseño de Sistemas con Procesadores | | |
| Créditos Académicos | 4 | | |
| ID Curso: | ID 31581 | | |
| Semestre | 2 | | |
| Grado Académico: | <input checked="" type="checkbox"/> Pregrado | <input type="checkbox"/> Posgrado | |
| Énfasis | No Aplica | | |
| Componentes: | <input type="checkbox"/> Teórico | <input checked="" type="checkbox"/> Teórico - Práctico | <input type="checkbox"/> Proyecto |
| Condiciones de Inscripción | Asignatura Abierta | <input checked="" type="checkbox"/> Si <input type="checkbox"/> No | |
| | Prerrequisitos: Introducción a la Programación Embebida (de Electrónica) ó Introducción a la Programación (de Sistemas) | | |
| Fecha de actualización: | 01/01/2023 | | |

1 DESCRIPCIÓN

La asignatura "Diseño de Sistemas con Procesadores" ofrece al estudiante de Ingeniería Electrónica las herramientas conceptuales básicas para solucionar problemas que involucren la concepción, diseño e implementación de programas en el Lenguaje de Programación C, para ser usados en sistemas de computador embebido. El estudiante estará en capacidad de formular algoritmos en forma de Diagramas de Flujo o Máquinas de Estados Finitos acorde con los requerimientos del problema. Adicional a ello, el estudiante estará en capacidad de implementarlos, ejecutarlos y probarlos, mediante herramientas de diseño asistido por computador y tarjetas de desarrollo con microcontroladores como elemento fundamental de diseño.

2 COMPETENCIAS DISCIPLINARES (CONTENIDOS NUCLEARES)

- A. Ingeniería de requerimientos y definición de especificaciones.
- B. Metodología de descomposición "top-down" y niveles de abstracción.
- C. Arquitecturas de Microcontrolador y dimensionamiento de soluciones con base en sistemas microcontrolados
- D. Tipos de periféricos en Microcontroladores y sus características.
- E. Atención apropiada de entradas/salidas y periféricos.
- F. Diseño de sistemas de procesador embebido de propósito específico.
- G. Máquinas de estados finitos

3 COMPETENCIAS NO DISCIPLINARES

2.5 CAPACIDADES PROFESIONALES Y ACTITUDES

- Estado actual del mundo de la ingeniería (CDIO 2.5.6)

3.2 COMUNICACIÓN

- Comunicación Escrita (CDIO 3.2.3)
- Comunicación Gráfica (CDIO 3.2.5)
- Comunicación Oral (CDIO 3.2.6)

4.1 CONTEXTO SOCIAL Y EXTERNO

- Roles y responsabilidad de los ingenieros (CDIO 4.1.1)
- Impacto de la ingeniería en la sociedad (CDIO 4.1.2)
- Regulación social de la ingeniería (CDIO 4.1.3)
- Contextos cultural e histórico (CDIO 4.1.4)
- Asuntos contemporáneos y valores (CDIO 4.1.5)
- Desarrollo de una perspectiva global (CDIO 4.1.6)

4.3 CONCEBIR Y APLICAR INGENIERÍA A LOS SISTEMAS

- Fijación de las metas y requerimientos (CDIO 4.3.1)
- Definición de la función, concepto y arquitectura (CDIO 4.3.2)
- Modelado de un sistema y fijación de metas alcanzables (CDIO 4.3.3)

4.4 DISEÑO

- El proceso de diseño (CDIO 4.4.1)
- Utilización del conocimiento en diseño (CDIO 4.4.3)

4.5 IMPLEMENTACIÓN

- Integración hardware – software (CDIO 4.5.4)
- Pruebas, verificación, validación, y certificación (CDIO 4.5.5)

4 OBJETIVO DE FORMACIÓN DE LA ASIGNATURA

El estudiante estará en capacidad de solucionar un problema, aplicando un enfoque metodológico que permite la Concepción, Diseño, Implementación y Operación del *software* de un sistema de computador, ‘embebido’ en un sistema mayor para realizar algunas de sus funciones y cuyo control de eventos externos por ‘*polling*’ se implementa mediante algoritmos descritos mediante Diagramas de Flujo y Máquinas de Estados Finitos.

5 RESULTADOS DE APRENDIZAJE ESPERADO (RAE)

- 5.1 Concebir la solución a un problema utilizando un sistema de computador embebido siguiendo una metodología de diseño *top-down* (CDIO 4.3.3) (NUCLEAR A-B)
- 5.2 Describir las especificaciones de un sistema basado en microcontrolador a partir de los requerimientos de una problemática (CDIO 4.3.1) (NUCLEAR A)
- 5.3 Identificar las funciones y características de cada módulo y periférico a partir de las especificaciones del sistema (CDIO 4.3.2) (NUCLEAR C)
- 5.4 Evaluar el comportamiento en el tiempo de una solución digital basada en microcontrolador. (CDIO 4.5.5) (NUCLEAR E)
- 5.5 Diseñar sistemas basados en microcontrolador integrando sensores y actuadores con programas escritos en el lenguaje de programación C utilizando máquinas de estados finitos (CDIO 4.4.3, 4.5.4) (NUCLEAR D-F-G)
- 5.6 Ilustrar, utilizando diagramas de bloques, tiempos, secuencia y estados, la arquitectura, características y comportamiento de un sistema de computador embebido. (CDIO 3.2.5, 4.3.2) (NUCLEAR A-F)
- 5.7 Describir por medio de presentaciones orales e informes técnicos la metodología, desarrollo y resultados de una solución basada en sistemas de computador embebido. (CDIO 3.2.3, 3.2.6, 4.4.1) (NUCLEAR A-F)

NOTA: Aunque no estén explícitas las competencias 4.1 CONTEXTO SOCIAL Y EXTERNO y 2.5.6 ESTADO ACTUAL DEL MUNDO DE LA INGENIERÍA en los objetivos, ésta debe ser trabajada por medio de los problemas presentados.

6 RÚBRICAS DE VALORACIÓN DE LOS RESULTADOS

| | | | |
|--|--|---|--|
| Objetivo 5.1. Concebir la solución a un problema utilizando un sistema de computador embebido siguiendo una metodología de diseño top-down | | | |
| <i>Indicador de Desempeño</i> | <i>Valoración menos aceptable - insuficiente</i> | <i>Valoración Desempeño medio</i> | <i>Valoración ejemplar - excelente</i> |
| Identifica los subsistemas que componen la solución del problema sobre un sistema digital basado en microcontrolador y las dependencias entre ellos. | Los subsistemas descritos no solucionan el problema propuesto, la funcionalidad no es correctamente trazable entre los subsistemas de un microcontrolador. | Los subsistemas propuestos son adecuados pero las funcionalidades, y dependencias no están claramente descritas. | Los subsistemas propuestos son adecuados y las funcionalidades y dependencias están correctamente descritas. |
| Objetivo 5.2. Describir las especificaciones de un sistema basado en microcontrolador a partir de los requerimientos de una problemática | | | |
| <i>Indicador de Desempeño</i> | <i>Valoración menos aceptable - insuficiente</i> | <i>Valoración Desempeño medio</i> | <i>Valoración ejemplar - excelente</i> |
| Describe la solución a un problema identificando las entradas, salidas y funcionalidades. | El sistema solución no tiene coherencia entre funciones y las entradas/salidas respecto al problema. | El sistema descrito no contiene todas las funcionalidades necesarias para la solución del problema. | La solución es coherente tanto en funciones como en la definición de entradas/salidas respecto al problema. |
| Define atributos del sistema como casos de uso de la solución sobre un sistema digital basado en microcontrolador. | Define atributos que cumplen parcialmente los requerimientos de una solución y que no son viables sobre el microcontrolador asignado. | Define atributos que cumplen algunos de los requerimientos pero con un alcance no acotado al microprocesador asignado | Define correctamente cada uno de los requerimientos a través de atributos con un alcance acotado |
| Objetivo 5.3. Identificar las funciones y características de cada módulo y periférico a partir de las especificaciones del sistema. | | | |
| <i>Indicador de Desempeño</i> | <i>Valoración menos aceptable - insuficiente</i> | <i>Valoración Desempeño medio</i> | <i>Valoración ejemplar – excelente</i> |
| Define bibliotecas para el funcionamiento de un periférico. | Utiliza el periférico ineficientemente al no definir funciones para rutinas repetitivas en el manejo de tareas sobre el periférico. | Define librerías que definen funciones para tareas repetitivas, pero sin definir la documentación de uso de las funciones para el periférico. | Define librerías para funciones repetitivas del periférico con su respectiva documentación. |
| Objetivo 5.4. Evaluar el comportamiento en el tiempo de una solución digital basada en microcontrolador. | | | |
| <i>Indicador de Desempeño</i> | <i>Valoración menos aceptable - insuficiente</i> | <i>Valoración desempeño medio</i> | <i>Valoración ejemplar – excelente</i> |
| Desarrolla y evalúa algoritmos sobre un microcontrolador que interactúan con un entorno dinámico. | La interacción mediante señales controladas en el tiempo por el algoritmo no generan una solución al fenómeno | La interacción mediante señales controladas en el tiempo por el algoritmo soluciona algunos requerimientos que describen el fenómeno. | La interacción mediante señales controladas en el tiempo por el algoritmo genera una solución al fenómeno. |
| Objetivo 5.5. Diseñar sistemas basados en microcontrolador integrando sensores y actuadores con programas escritos en el lenguaje de programación C utilizando máquinas de estados finitos. | | | |
| <i>Indicador de Desempeño</i> | <i>Valoración menos aceptable -</i> | <i>Valoración Desempeño medio</i> | <i>Valoración ejemplar – excelente</i> |

| | | | |
|--|---|---|---|
| | <i>insuficiente</i> | | |
| Implementa máquinas de estados finitos para microcontrolador sobre lenguaje de alto nivel que interactúan con sensores y actuadores. | La máquina de estados finitos desarrollada presenta errores en la lógica de descripción sobre lenguaje de alto nivel, de forma que los sensores y actuadores no cumple con el requerimiento de la problemática. | El software desarrollado está correctamente descrito en lenguaje de alto nivel, y permite la interacción parcial entre sensores y actuadores según el requerimiento de la problemática. | El software desarrollado está correctamente descrito en lenguaje de alto nivel, y permite la interacción entre sensores y actuadores según el requerimiento de las problemática. |
| Objetivo 5.6. Ilustrar, utilizando diagramas de bloques, tiempos, secuencia y estados, la arquitectura, características y comportamiento de un sistema de computador embebido. | | | |
| <i>Indicador de Desempeño</i> | <i>Valoración menos aceptable - insuficiente</i> | <i>Valoración Desempeño medio</i> | <i>Valoración ejemplar - excelente</i> |
| Aplica principios fundamentales para comunicar ideas de ingeniería de forma gráfica mediante el uso de diferentes tipos de herramientas visuales. | Mediante los entregables presentados no permite la extracción de las características y comportamiento del sistema. | Mediante los entregables presentados no alcanza a describir de manera completa un proyecto de ingeniería de manera gráfica. | Mediante los entregables sintetiza de manera visual el desarrollo de un proyecto mediante, diagramas de bloques, diagramas de tiempos, diagramas de flujo, máquinas de estados finitos y demás herramientas gráficas para presentar resultados. |
| Objetivo 5.7. Describir por medio de presentaciones orales e informes técnicos la metodología, desarrollo y resultados de una solución basada en sistemas de computador embebido. | | | |
| <i>Indicador de Desempeño</i> | <i>Valoración menos aceptable - insuficiente</i> | <i>Valoración Desempeño medio</i> | <i>Valoración ejemplar - excelente</i> |
| Tiene la capacidad para hacer una presentación técnica bajo unas restricciones de tiempo a una audiencia | Al desarrollar la presentación técnica se desvía del guion escrito y no demuestra manejo de tiempo ni dominio de la audiencia. | Al desarrollar la presentación técnica comunica sus ideas pero tiene falencias en el manejo de los elementos para que sea una presentación efectiva. | Desarrolla una presentación técnica reconociendo la audiencia mediante una clara introducción, desarrollo y cierre con las respectivas conclusiones. |
| Mediante un informe escrito demuestra la aplicación de la metodología para diseño de un sistema basado en microcontroladores. | En su informe técnico de diseño no se aprecia la apropiación de una metodología rigurosa para presentar un proyecto y el lenguaje usado no es apropiado para este propósito. | En su informe técnico, aunque usa un lenguaje y estilo apropiados no demuestra rigor en la apropiación ni aplicación metodológica para llevar a cabo una metodología de diseño. | Está en capacidad de producir un escrito técnico y/o de ingeniería de alta calidad. Considera aspectos éticos y de estilo en su escritura imparcial y precisa de todos los aspectos de un proyecto. |

7 ESTRATEGIAS PEDAGÓGICAS

En el desarrollo del proyecto se abordan diferentes estrategias como: **Aprendizaje por Proyectos**, donde se propone la elaboración un producto que responda a una necesidad o problema. Permite organizar y planificar el trabajo, identificando roles y responsabilidades para buscar una solución. **Aprendizaje Colaborativo**, donde a través del grupos de trabajo se realice un intercambio de conocimientos e instrucción por pares para reforzar el aprendizaje significativo. **Aprendizaje por discusión**, que bajo un problema propuesto el estudiante plantee una alternativa de solución susceptible a controversia mediante la discusión en grupos de trabajo se construye conocimiento bajo el reconocimiento de diferentes puntos de vista.

8 ACTIVIDADES DE EVALUACIÓN

Las estrategias de evaluación están centradas en la valoración de los resultados de aprendizaje esperado de la asignatura; las cuales pueden ser formativas que suscitan la comprensión y construcción de conocimiento, y sumativas que incluyen porcentajes de evaluación. Estas últimas son:

| | COMPONENTE | FECHA | PORCENTAJE |
|----|------------------------------|------------------|------------|
| | Valoración Individual | | |
| 1. | Primer Examen Parcial | Semana 06 | 20% |
| 2. | Segundo Examen Parcial | Semana 12 | 20% |
| 3. | Evaluación Final | Semana 18 | 30% |
| | Trabajo en Equipo | | |
| 4. | Primer proyecto de diseño | Semana 10 | 10% |
| 5. | Segundo proyecto de diseño | Semana 16 | 10% |
| | Trabajo Extra-clase | | |
| 6. | Trabajos asignados | Todo el semestre | 10% |

9 Horarios:

Asignatura de 4 Créditos implica 192 Horas de trabajo semestrales divididas así:

48 Horas de clase presenciales (3 H/Semana)

16 Horas de trabajo práctico (1 H/Semana)

128 Horas de trabajo fuera del aula de clase (8 H/semana)

10 PROGRAMA DEL CURSO

A. CODIFICACIÓN DE BAJO NIVEL

10.A.1 *Sistemas de Numeración*

10.A.2 *Representación de Números en un Sistema Digital*

10.A.3 *Formatos Estándar IEEE-754 de Punto Flotante*

10.A.4 *Codificación Binaria*

10.A.5 *Secuencias de Conteo*

B. PROGRAMACIÓN EN BAJO NIVEL

10.B.1 *Herramientas de Desarrollo en C*

10.B.2 *Control de Flujo en Bajo Nivel*

10.B.3 *Casos Especiales de Control de Flujo*

10.B.4 *Operaciones de la ALU: Las Expresiones en C*

10.B.5 *Tipos de Datos en Memoria: Las Declaraciones en C*

10.B.6 *Llamado con Retorno y Contexto en Pila: Las Funciones en C*

- C. METODOLOGÍAS DE DISEÑO
- 10.C.1 *Ingeniería de Software y de Hardware*
 - 10.C.2 *Elaboración y Lectura de Diagramas*
 - 10.C.3 *Sistemas de Computador Embebido*
 - 10.C.4 *Configuración de Periféricos*
 - 10.C.5 *Protocolos de Comunicación*
- D. MÁQUINAS DE ESTADOS FINITOS
- 10.D.1 *Diagramas de Estado*
 - 10.D.2 *Tipos de Máquinas de Estados Finitos*
 - 10.D.3 *Implementación de Máquinas de Estados Finitos*
 - 10.D.4 *Máquinas de Estados Finitos Concurrentes*
- E. INTRODUCCIÓN A RTOS
- 10.E.1 *Interrupciones Hardware*
 - 10.E.2 *Arquitecturas de Software*
 - 10.E.3 *Servicios de una Arquitectura RTOS*
 - 10.E.4 *Otros Servicios del Sistema Operativo*
 - 10.E.5 *Uso de un Diseño Básico con RTOS*

| | |
|--------------------|--|
| Semana 1 | Presentación del Curso – Metodología de Trabajo |
| | Introducción al Curso Presentación del syllabus Metodología Forma de calificación |
| | CAPÍTULO 1. CODIFICACIÓN DE BAJO NIVEL |
| | 1.1. Sistemas de Numeración 1.1.1. Historia de los sistemas de numeración 1.1.2. Sistema numérico posicional 1.1.3. Conversión entre bases numéricas |
| | 1.2. Representación de Números en un Sistema Digital 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 |
| Semana 2 | Formatos IEEE-754, Codificación y Herramientas de Desarrollo |
| | 1.3. Formatos Estándar IEEE-754 de Punto Flotante 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 |
| Estudio individual | 1.4. Codificación Binaria 1.4.1. Justificación y usos 1.4.2. Códigos ponderados, no-ponderados, auto-complementados |

| | |
|------------|--|
| | <p>1.4.3. Códigos alfanuméricos</p> <p>1.4.4. Códigos detectores y correctores de error</p> <p>1.5. Secuencias de Conteo</p> <p>1.5.1. Secuenciadores hardware</p> <p>1.5.2. Contadores binario y Gray</p> <p>1.5.3. Contadores de anillo y Johnson</p> <p>1.5.4. Contadores pseudo-aleatorios</p> |
| | CAPÍTULO 2. PROGRAMACIÓN EN BAJO NIVEL |
| | <p>2.1. Herramientas de Desarrollo en C</p> <p>2.1.1. Sistema operativo por consola</p> <p>2.1.2. El pre-procesador de C</p> <p>2.1.3. Etapas de compilación en C</p> <p>2.1.4. Herramientas software</p> <p>2.1.5. Herramientas hardware</p> |
| Semana 3 | Control de Flujo |
| | <p>2.2. Control de Flujo en Bajo Nivel</p> <p>2.2.1. Diagramas de flujo</p> <p>2.2.2. Instrucciones, condiciones y secuencias</p> <p>2.2.3. Bloque de instrucciones (delimitadores "{" y "}")</p> <p>2.2.4. Secuencias lineales (separador ";")</p> <p>2.2.5. Secuencias de selección ("if-else", "if")</p> <p>2.2.6. Secuencias de repetición ("for", "while", "do-while")</p> |
| | <p>2.3. Casos Especiales de Control de Flujo</p> <p>2.3.1. Árboles de decisión</p> <p>2.3.2. Selección con "switch-case"</p> <p>2.3.3. Transformaciones de "loop"</p> <p>2.3.4. Saltos locales ("goto-label", "break", "continue", "return")</p> <p>2.3.5. Saltos no-locales (setjmp.h)</p> <p>2.3.6. Resumen: "Secuencias en Lenguajes de Ensamble"</p> |
| Semana 4 | Expresiones en Lenguaje C |
| | <p>2.4. Operaciones de la ALU: <i>Expresiones en C</i></p> <p>2.4.1. Operandos y operadores</p> <p>2.4.2. Precedencia</p> <p>2.4.3. Asociación</p> <p>2.4.4. Evaluación</p> <p>2.4.5. Promoción</p> <p>2.4.6. Puntos de secuencia garantizada</p> |
| | Resolución en clase del taller de evaluación expresiones |
| Semana 5 | Tipos de Datos y Uso de la Memoria |
| | <p>2.5. Tipos de Datos en Memoria: <i>Declaraciones en C</i></p> <p>2.5.1. Atributos de tipos de datos</p> <p>2.5.2. Tipos escalares, agregados y derivados</p> <p>2.5.3. Apuntadores</p> <p>2.5.4. Estructuras</p> <p>2.5.5. Arreglos</p> <p>2.5.6. Declaraciones complejas</p> |
| Febrero 24 | <p>2.6. Llamado con Retorno y Contexto en Pila: <i>Funciones en C</i></p> <p>2.6.1. Parámetros y argumentos</p> |

| | |
|--------------------|---|
| | <p>2.6.2. Paso de parámetros por valor y por referencia</p> <p>2.6.3. Tipos de almacenamiento ("auto", "static", "register", "extern")</p> <p>2.6.4. Retorno de valores de la función</p> <p>2.6.5. Uso del "stack" y el "stack frame"</p> |
| Semana 6 | Semana de Evaluación e Introducción a Metodologías de Diseño |
| | Primera Evaluación Parcial (20%) |
| | CAPÍTULO 3. METODOLOGÍAS DE DISEÑO |
| | <p>3.1. Ingeniería de Software y de Hardware</p> <p>3.1.1. Métricas de complejidad del software</p> <p>3.1.2. Fases de diseño de sistemas</p> <p>3.1.3. Diseño "top-down", "bottom-up" y diseño en "V"</p> <p>3.1.4. Desarrollo iterativo e incremental</p> <p>3.1.5. Metodologías "Waterfall" y "Agile"</p> |
| Semana 7 | Diagramas en Ingeniería y Uso de Manuales Técnicos |
| | <p>3.2. Elaboración y Lectura de Diagramas</p> <p>3.2.1. Diagramas de bloques</p> <p>3.2.2. Diagramas de tiempo</p> <p>3.2.3. Diagramas esquemáticos</p> |
| | <p>3.2.4. Caso de estudio: "La Arquitectura de un Computador"</p> <p>3.2.5. Aplicación: "Uso de Manuales de Especificaciones"</p> |
| Semana 8 | Introducción a los Sistemas de Computador Embebido |
| | <p>3.3. Sistemas de Computador Embebido</p> <p>3.3.1. Definición y usos</p> <p>3.3.2. Atributos de calidad</p> <p>3.3.3. Microprocesadores y buses</p> |
| | <p>3.3.4. Metodología de diseño en capas</p> <p>3.3.5. Casos de estudio: "Diagramas Esquemáticos de Varios Sistemas"</p> |
| Semana 9 | Configuración de Periféricos y Protocolos de Comunicaciones |
| | <p>3.4. Configuración de Periféricos</p> <p>3.4.1. Puertos paralelos</p> <p>3.4.2. Temporizadores</p> <p>3.4.3. Generadores "Pulse-Width Modulation" (PWM)</p> |
| | <p>3.4.4. Conversores "Analog-to-Digital" (ADC)</p> <p>3.4.5. Puertos seriales</p> |
| Estudio individual | <p>3.5. Protocolos de Comunicación</p> <p>3.5.1. Comunicaciones "on-board"</p> <p>3.5.2. Comunicaciones entre dispositivos</p> <p>3.5.3. Comunicaciones inalámbricas</p> |
| Semana 10 | Entrega de Proyectos e Introducción a Máquinas de Estados Finitos |
| | Entrega y sustentación del primer proyecto (10%) |
| | CAPÍTULO 4. MÁQUINAS DE ESTADOS FINITOS |
| | <p>4.1. Diagramas de Estado</p> <p>4.1.1. Teoría de grafos</p> <p>4.1.2. Representación matricial de grafos</p> <p>4.1.3. Aplicaciones y usos de grafos</p> <p>4.1.4. Diagramas de estado como grafos</p> <p>4.1.5. Diagramas de estado bien estructurados</p> |

| | |
|---------------------------|--|
| Semana Santa | |
| Semana 11 | Tipos de Máquinas de Estados Finitos y su Implementación |
| | 4.2. Tipos de Máquinas de Estados Finitos 4.2.1. Definiciones formales 4.2.2. Taxonomía de autómatas según existencia de entrada/salida 4.2.3. Taxonomía de máquinas traductoras 4.2.4. Taxonomía de máquinas según memoria 4.2.5. Taxonomía de máquinas según naturaleza determinística |
| | 4.3. Implementación de Máquinas de Estados Finitos 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 |
| Semana 12 | Máquinas Concurrentes |
| | 4.4. Máquinas de Estados Finitos Concurrentes 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 |
| | Segunda Evaluación Parcial (20%) |
| Semana 13 | Interrupciones Hardware |
| | CAPÍTULO 5. INTRODUCCIÓN A RTOS |
| | 5.1. Interrupciones Hardware 5.1.1. Conceptos básicos sobre interrupciones 5.1.2. Problemas de datos compartidos 5.1.3. Latencia de interrupción |
| | Pre-entrega del segundo proyecto (5%) |
| Semana 14 | Arquitecturas Software y Servicios de un RTOS |
| Exposiciones individuales | 5.2. Arquitecturas de Software 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) |
| Exposiciones individuales | 5.3. Servicios de una Arquitectura RTOS 5.3.1. Tareas y estados de las tareas 5.3.2. Tareas y datos 5.3.3. Semáforos y datos compartidos |
| Semana 15 | Servicios de un RTOS y Uso de un Diseño Básico |
| Exposiciones individuales | 5.4. Otros Servicios del Sistema Operativo 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 |
| Exposiciones individuales | 5.5. Uso de un Diseño Básico con RTOS 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 |
| Semana 16 | Entregas Finales de Proyectos |

| | |
|-----------|--|
| | Entregas finales y sustentaciones del segundo proyecto (5%) |
| | Entregas finales y sustentaciones del segundo proyecto (5%) |
| Semana 17 | Inicio de evaluaciones finales |
| | Exámenes Finales de otras asignaturas |
| Semana 18 | Cierre del curso |
| | Evaluación Final de DISPRO (30%) |

11 BIBLIOGRAFIA

- A. Paul L. Anderson & Gail C. Anderson, ***“Advanced C Tips and Techniques”***, Hayden Books, 1988.
- B. William B. Frakes, Christopher J. Fox, Brian A. Nejmeh, ***“Software engineering in the UNIX/C environment”***, Published by Prentice Hall, Inc., 1991.
- C. Juan-Carlos Giraldo, ***“Low Level Programming in C”***, Pontificia Universidad Javeriana, 2023.
- D. Douglas Wilhem Harder, Jeff Zarnett, Vajih Montaghmi, Allyson Giannikouris, ***“A practical introduction to real-time systems for undergraduate engineering”***, University of Waterloo, 2014.
- E. David Money Harris & Sarah L. Harris, ***“Digital Design and Computer Architecture”***, Elsevier, 2013.
- F. IEEE Computer Society, ***“IEEE Standard for Floating-Point Arithmetic”***, 1985.
- G. IEEE Computer Society, ***“IEEE Standard for Radix-Independent Floating-Point Arithmetic”***, 1987.
- H. IEEE Computer Society, ***“IEEE Standard for Floating-Point Arithmetic (Revision)”***, 2008.
- I. IEEE Computer Society, ***“IEEE Standard for Floating-Point Arithmetic (Revision of IEEE 754-2008)”***, 2019.
- J. 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>
- K. Microchip Technology Incorporated, ***“Introduction to Embedded Programming Using C”***, 2007.
- L. David E. Simon, ***“An Embedded Software Primer”***, by Pearson Education, Inc., 1999.

12 DECLARACIÓN DE LOS REGLAMENTOS

En esta sección se deben colocar las declaraciones que registrarán sobre los casos de copia, los cuales deben estar basados en los artículos 113d - 114b – 114e - 117 y 118 del reglamento de estudiantes

A. Para parciales:

Objetos permitidos: lápiz, lapicero, minas, portaminas, borrador, (calculadora no programable en caso que aplique, previa autorización expresa del profesor). Nota: todo objeto diferente a los permitidos se considerará material no autorizado, deberá permanecer en los morrales y por lo tanto el estudiante no debe estar en posesión del mismo durante el parcial, ejemplo:

teléfonos celulares, relojes inteligentes, apuntes, cuadernos, hojas, entre otros. En caso de posesión de material no autorizado durante el examen, independientemente se esté manipulando o no, el profesor retirará el parcial y le impondrá la calificación de 0.0. en la evaluación. Así mismo, el caso será reportado a la Dirección de Carrera para proceder de acuerdo a lo estipulado en el Reglamento de Estudiantes de la Pontificia Universidad Javeriana

B. Para proyectos de desarrollo de código:

El proyecto es un trabajo individual y original de cada grupo. No es permitido reproducir total o parcialmente el código de otro estudiante o código de otras fuentes bibliográficas o de internet.