



Microprogramación

2 créditos teóricos, 2 créditos prácticos



A. Información del profesor

Nombre del profesor

Teoría: Ing. Cristian Estuardo Roldán Rodríguez
Laboratorio: Inga. Karen Elisabeth Liska Lima

Correo electrónico

Teoría: ceroldanr@correo.url.edu.gt
Laboratorio: karenliska@gmail.com

Campus o sede

Campus Central

Horario

Teoría: lunes 19:30 – 21:00 horas, Salón T-306
Laboratorio: viernes 19:30 – 21:00 horas, Salón T-305



B. Información general

Descripción

El curso permite la interacción con los lenguajes de bajo nivel y la abstracción del estilo de programación en un entorno ligado al lenguaje de máquina (conjunto de '0' y '1' que son los únicos comprensibles por los microprocesadores, en su interpretación como pulsos eléctricos), por el contrario de los lenguajes de alto nivel acostumbrados, más ligados al usuario. Enfoca su contenido en la programación de microprocesadores en base a su set de instrucciones, parámetros de interrupciones y registros de sistema, específicamente de la familia Intel 80x86.

Modalidad

Presencial.



C. Malla curricular

COMPETENCIAS GENÉRICAS



El egresado landivariano se identifica por:

Pensamiento
lógico, reflexivo y
analógico

Pensamiento
crítico

Resolución de
problemas

Habilidades de
investigación

Uso de TIC y
gestión de la
información

Comunicación
efectiva, escrita y
oral

Comprensión
lectora

Compromiso
ético y
ciudadanía

Liderazgo
constructivo

Aprecio y respeto
por la diversidad e
interculturalidad

Creatividad

COMPETENCIAS ESPECÍFICAS

Competencia 1

Utiliza la conversión entre sistemas numéricos para la representación lógica del almacenamiento de datos en la memoria principal.

Competencia 2

Diferencia los principios de los lenguajes de programación de bajo nivel en microprocesadores en base a su set de instrucciones.

Competencia 3

Genera programas de bajo nivel sobre arquitecturas de 16, 32 y 64 bits para el aprovechamiento óptimo de los recursos de hardware.



METODOLOGÍA

Este curso se desarrollará a través de los siguientes métodos de aprendizaje-enseñanza:



Aprendizaje Basado en Proyectos

Consiste en realizar un proyecto, individual o grupal, que puede basarse en uno o varios temas de clase. El profesor debe garantizar que el problema que deben desarrollar en el proyecto ha sido claramente entendido por los estudiantes. Además, debe asegurarse que se cuenta con todos los recursos necesarios para el desarrollo de las partes del proyecto.

El Aprendizaje Basado en Proyectos permite a los estudiantes desarrollar competencias complejas como el pensamiento crítico, la comunicación, la colaboración o la resolución de problemas a través de la elaboración de proyectos que den respuesta a problemas de la vida real, partiendo de un problema concreto.



PROGRAMACIÓN

COMPETENCIA 1

Utiliza la conversión entre sistemas numéricos para la representación lógica del almacenamiento de datos en la memoria principal.

Saber conceptual (contenido temático)

Sistemas numéricos

- Sistema de numeración binario
- Sistema de numeración decimal
- Sistema de numeración hexadecimal

Saber procedimental (habilidades y destrezas)

- Relaciona los sistemas de numeración en base a sus patrones de conversión en forma algorítmica.

Saber actitudinal (conductas observables)

- Utiliza las mejores prácticas de programación para resolver problemas optimizando recursos.
- Presenta soluciones completas y realizadas con técnicas ordenadas y eficientes.

Indicador de logro 1 (resultado):

Convierte datos numéricos entre los diferentes sistemas mediante métodos manuales y programados a bajo nivel.



COMPETENCIA 2

Diferencia los principios de los lenguajes de programación de bajo nivel en microprocesadores en base a su set de instrucciones.

Saber conceptual (contenido temático)

Introducción al Lenguaje Ensamblador

- Lenguajes de programación de bajo nivel
- Ensamblador
- Lenguaje ensamblador
- Turbo Assembler
- Macro Assembler
- Otros ensambladores

Organización de la Máquina

- Arquitectura del Computador
- Memoria Principal
- Registros
- Segmentos de Memoria

Saber procedimental (habilidades y destrezas)

- Implementa programas con diferentes ensambladores según el set de instrucciones correspondiente al hardware de la computadora.

Saber actitudinal (conductas observables)

- Utiliza las mejores prácticas de programación para resolver problemas optimizando recursos.
- Presenta soluciones completas y realizadas con técnicas ordenadas y eficientes.

Indicador de logro 1 (resultado):

Codifica un programa de bajo nivel que utilice el ensamblador acorde a la arquitectura del computador sobre la que será ejecutado.

Indicador de logro 2 (resultado):

Utiliza las características de un ensamblador específico para acceder a los elementos de la memoria principal en programas de bajo nivel ejecutados en arquitecturas de 32 o 64 bits.



COMPETENCIA 3

Genera programas de bajo nivel sobre arquitecturas de 16, 32 y 64 bits para el aprovechamiento óptimo de los recursos de hardware.

Saber conceptual (contenido temático)

Programación en Lenguaje Ensamblador

- Estructura del programa
- Modelos del programa
- Directivas

Instrucciones en Lenguaje Ensamblador

- Transferencia de datos
- Operaciones
- Saltos

Servicios de Interrupción

- Eventos
- Tabla de Interrupciones

Macros

- Librerías de Macros

Arreglos en Memoria

- Mapeo lexicográfico

Saber procedimental (habilidades y destrezas)

- Emplea la nomenclatura básica del lenguaje ensamblador para generar programas de bajo nivel.
- Expande el alcance de los programas de bajo nivel con el uso de transferencia de datos, operadores y saltos.
- Complementa la programación a bajo nivel con el uso de los servicios de interrupción, Macros y arreglos en memoria.

Saber actitudinal (conductas observables)

- Utiliza las mejores prácticas de programación para resolver problemas optimizando recursos.
- Presenta soluciones completas y realizadas con técnicas ordenadas y eficientes.

Indicador de logro 1 (resultado):

Aplica código de bajo nivel compilado con Turbo Assembler para generar un programa sobre una arquitectura de 16 bits.

Indicador de logro 2 (resultado):

Utiliza código de bajo nivel compilado con un ensamblador específico para generar un programa sobre una arquitectura de 32 bits o superior.



EVALUACIÓN

a. Estrategias de evaluación sumativa

Estrategias	Puntaje
Teoría	
Actividades de seguimiento	10
Evaluaciones de seguimiento	20
Evaluación Final	20
Laboratorio	
Prácticas de desarrollo	20
Proyectos de aplicación	30
TOTAL	100

b. Estrategias de evaluación formativa

Técnicas	Procedimiento
Retroalimentación	Comentarios pertinentes en la entrega de prácticas de desarrollo y proyectos de aplicación.
Foros	Participación en clase, resolución de dudas asincrónicas.
Trabajo colaborativo	Resolución de dudas en ambientes colaborativos.
Citas individuales	Resolución de dudas en ambientes personalizados.



CALENDARIO DE REFERENCIA POR TEMAS

Fecha	Tema	Actividad de evaluación
Semana 1	<p>Sistemas Numéricos</p> <ul style="list-style-type: none"> Sistema de numeración binario, decimal y hexadecimal <p>Introducción al Lenguaje Ensamblador</p> <ul style="list-style-type: none"> Lenguajes de programación Perspectiva histórica Lenguajes de bajo nivel Ensamblador Lenguaje Ensamblador 	Práctica de desarrollo
Semana 2 – 3	<p>Organización de Máquina</p> <ul style="list-style-type: none"> Arquitectura del computador Memoria principal Registros Segmentos de memoria 	Presentaciones grupales
Semana 4	<p>Programación en Lenguaje Ensamblador</p> <ul style="list-style-type: none"> Turbo Assembler Estructura de programa Modelos de programa Directivas 	Práctica de desarrollo
Semana 5 – 6	<p>Instrucciones del Lenguaje Ensamblador</p> <ul style="list-style-type: none"> Transferencia de datos Operaciones Salto condicionales y no condicionales 	Práctica de desarrollo Evaluación de selección múltiple
Semana 7	<p>Servicios de Interrupciones</p> <ul style="list-style-type: none"> Tipos de Interrupción Eventos en una Interrupción Tabla de Servicios de Interrupción 	Práctica de desarrollo
Semana 8	<p>Procedimientos</p> <ul style="list-style-type: none"> Definiciones Llamadas 	Práctica de desarrollo
Semana 9 – 11	<p>Introducción a otros ensambladores</p> <ul style="list-style-type: none"> Macro Assembler Turbo Assembler vs Macro Assembler Otros ensambladores 	Práctica de desarrollo Proyecto de aplicación
Semana 12 – 13	<p>Macros</p> <ul style="list-style-type: none"> Definiciones Bibliotecas de Macros 	Práctica de desarrollo Evaluación de selección múltiple
Semana 14	<p>Arreglos en Memoria</p> <ul style="list-style-type: none"> Mapeo lexicográfico 	Práctica de desarrollo Proyecto de aplicación Evaluación final



Vo. Bo.

Ing. Adolfo Galán

Director

Departamento de Ingeniería en Informática y Sistemas
Facultad de Ingeniería