



Nombre del Cuso: Introducción a la Programación y Computación II			
Código:	771	Créditos:	5
Escuela:	CIENCIAS Y SISTEMAS	Área a la que pertenece:	Programación
Pre requisito:	Introducción a la Programación y Computación I (770) Matemática Intermedia (107) Lógica Matemática (795) Matemática de Computo 1 (960)	Post requisito:	Organización Computacional (964) Estructura de Datos (772) Org. Lenguajes y Compiladores 1 (777)
Categoría:	Obligatorio	Semestre:	Segundo Semestre 2026
Docente:	Ing. Fernando José Paz Gonzales	Auxiliar:	Carlos Manuel Lima Y Lima
Edificio:		Sección:	P
Salón del curso:	Meet	Salón de laboratorio:	Meet
Horas por semana del curso:	4	Horas por semana del laboratorio:	2
Días que se imparte el curso:	Lunes y Miércoles	Días que se imparte el laboratorio:	Martes
Horario del curso:	17:20 – 19:00	Horario del laboratorio:	17:20 – 19:00

1. Descripción del curso

Este curso está diseñado para que el estudiante inicie el proceso de modelado de sistemas de software utilizando los conceptos de la programación orientada a objetos, los ambientes en que estas soluciones son dispuestas, temas orientados a las bases de desarrollo sobre la web, Cloud, así como los temas asociados al manejo de versiones y control del software.

2. Objetivos

General

Lograr que el estudiante expanda sus conocimientos de desarrollo de software usando elementos que le brinden una visión general de los procesos de desarrollo, así como elementos de calidad que le ayuden a robustecer los entregables

Específicos

1. Modelar problemas de forma estándar y profesional
2. Entender metodologías de desarrollo para construir aplicaciones de software aplicando buenas prácticas orientada a la calidad de este.
3. Organizar soluciones en diferentes ambientes en el proceso del desarrollo del



software usando arquitecturas claves como stand-alone, cliente servidor y web.

3. Metodología

1. El curso se impartirá a través de clases magistrales **virtuales** dos días por semana, con duración de dos periodos cada día.
2. El laboratorio se impartirá **de manera virtual** una vez por semana, con duración de dos períodos cada día.
3. Durante el semestre, se asignarán tres proyectos de programación a realizarse de manera individual; así como tareas, ejercicios y pruebas cortas.
4. Las evaluaciones parciales y el examen final serán de manera **presencial**.

4. Competencias terminales

Al finalizar el curso el estudiante desarrolla las siguientes competencias:

- Capacidad para aplicar metodologías de programación y desarrollo de aplicaciones de software.
- Capacidad de aplicar los temas de calidad en el desarrollo de su software bajo el uso de prácticas estándares.
- Conocimiento de los aspectos claves de seguridad y calidad en el desarrollo del software.
- Dominio en el manejo de la memoria dinámica y los TDA's básicos requeridos en el curso de estructura de datos.
- Conocimiento de los ambientes necesarios para desarrollar software y garantizar su buen funcionamiento ante los usuarios finales.
- Conocimiento de la gestión de versiones del software y releases correspondientes.

5. Observaciones

1. Es obligatorio acumular el 80% de asistencia antes de cada parcial (de lo contrario no se tendrá derecho a examen).
2. El laboratorio se calificará sobre 100, y será equivalente a 30 puntos de zona.
3. Habrá 3 proyectos de programación.
4. El catedrático revisará las notas obtenidas en el curso y el laboratorio. Podrá decidir si es necesaria una segunda revisión a cada proyecto y considerar nuevamente la ponderación obtenida en cada proyecto.
5. Las notas de cada proyecto serán publicadas por el catedrático del curso en el transcurso del semestre, el estudiante tendrá 8 días como máximo para pedir revisión de proyecto.
6. El laboratorio debe aprobarse con 61 puntos sobre 100.
7. Es obligatorio ganar el laboratorio para tener derecho a evaluación final del curso.
8. No habrá proyecto de retrasada, ni reposición de nota de laboratorio.
9. El curso se aprueba con 61 puntos.



6. Contenido temático del curso

Unidad	Tema
1. Otros algoritmos y memoria dinámica aplicada (3 clases)	- Análisis y diseño de algoritmos - Algoritmos y eficiencia - Tiempo de ejecución $T(n)$ - Análisis de casos - Medidas asintóticas - Notación O grande - Análisis de la complejidad de algoritmos recursivos - Análisis de la eficiencia de algoritmos de ordenamiento - Métodos burbuja, inserción, selección y Quick Sort - Eficiencia comparativa - Paradigma Divide y Vencerás - Definición de pasos del paradigma - Método mergesort - Método de búsqueda binaria - Tipos de datos abstractos (TDA's) - Tipos de datos abstractos (TDA's) Lineales y no lineales - Matrices dispersas (Ortogonales)
2. Tipos de datos abstractos NO lineales (5 clases)	- Estructuras base no lineales - Árboles - Definición y terminología - Representación gráfica - Modelado orientado a objetos aplicados a árboles - Diagramas de clases para árboles - Relaciones entre nodos - Diseño de estructuras dinámicas - Árboles binarios de búsqueda - Operaciones (inserción, búsqueda, eliminación) - Recorridos (Inorden, preorden, posorden) - Árboles AVL - Definición - Operaciones - Rotaciones (simples, dobles)
3. Arquitecturas para soluciones WEB (12 clases)	- Arquitecturas de despliegue - Modelado UML para despliegue de aplicaciones Diagramas de componentes, Diagramas de despliegue - Arquitectura de despliegue stand-alone - Arquitectura de despliegue Cliente-Servidor - Arquitectura de despliegue N-Capas - Arquitectura Cloud - Web Development - Conceptos iniciales Introducción al diseño web Introducción a HTML y CSS



	Principios básicos de HTML Principios básicos de CSS Entendimiento de servidores Web, Exploradores, HTTP/HTTPS y FTP Funcionamiento de los servidores Web Web Servers Web Browsers / Exploradores 2.2. Front End Development Programación del lado del cliente Javascript 2.3. Back End Development Programación del lado del servidor JSON files XML files Uso de marcadores de hipertexto del lado de la capa de datos 2.4. Patrón MVC Conceptos iniciales Modelo Vista Controlador 2.5. Modelado base aplicado al MVC Diagramas de clases. Regla de navegabilidad en capas: Vista > Controlador > Modelo Diagrama de secuencias. Flujo típico: Usuario > Vista > Controlador > Modelo > Vista Diagrama de componentes. Representar MVC como componentes conectados por interfaces. Ejemplos prácticos 2.6. Aplicación Web básica Formularios Manejo de datos Navegación entre páginas 2.7. Integración de componentes Comunicación entre capas Flujo de información 2.8. Ambientes de Software, Configuración y Despliegue Tipos de ambiente (Desarrollo, Quality Assurance (QA) Unit Acceptance Testing (UAT) y producción. Despliegue entre ambientes
3. Calidad y seguridad del Software (5 clases)	1. Introducción a la calidad en pruebas de Software 1.1. Conceptos de pruebas de software 1.2. Objetivos 1.3. Casos de prueba 1.4. Validación de entradas y manejo de errores 1.5. Introducción a pruebas en entornos ágiles 2. Introducción al testing automatizado 2.1. Qué es la automatización de pruebas 2.2. Beneficios de la automatización 2.3. Flujo básico de pruebas automatizadas



-
- | | |
|--|---|
| | <ul style="list-style-type: none">2.4. Validación funcional de aplicaciones Web2.5. Introducción a IA aplicada a QA y Testing con agentes3. Seguridad en el Software<ul style="list-style-type: none">3.1. Seguridad en el ciclo de vida del software3.2. Shift left Security3.3. Buenas prácticas de desarrollo seguro3.4. Seguridad en aplicaciones Web y móviles<ul style="list-style-type: none">Security OWASp Checklist – Data protection, file management, memory managementOWASp top 10 |
|--|---|
-

7. Evaluación del rendimiento académico

Según el Reglamento General de Evaluación y Promoción del Estudiante de la Universidad de San Carlos de Guatemala, la zona tiene valor de 75 puntos, la nota mínima de promoción es de 61 puntos y la zona mínima para optar a examen final es de 36 puntos.

Procedimiento de evaluación		Ponderación
Clase	Tareas y/o cortos	05.25 pts.
	Prácticas	03 pts.
	Primer parcial	12 pts.
	Segundo parcial	13 pts.
	Tercer parcial	13 pts.
Total de clase		46.25 pts.
Laboratorio	Proyecto I	8 pts.
	Proyecto II	8 pts.
	Proyecto III	8 pts.
	Examen final (Evaluación de 1 punto por proyecto)	4.75 pts.
		.
Total de laboratorio		28.75 pts.
Zona		75 pts.
Examen Final		25 pts.
Nota de promoción		100 pts.

8. Cronograma de actividades

Tema principal	Contenido a desarrollar	Fecha
Análisis y diseño de algoritmos	Algoritmos y eficiencia Tiempo de ejecución $T(n)$ Análisis de casos Medidas asintóticas Notación O grande Análisis de la complejidad de algoritmos recursivos	Julio
Análisis de la eficiencia de algoritmos de ordenamiento	Métodos de burbuja Método de inserción Selección Quick Sort Eficiencia comparativa	Julio
Paradigma Divide y Vencerás	Definición de pasos del paradigma Método mergesort Método de búsqueda binaria	Julio
Tipos de datos abstractos (TDA's)	Introducción a los tipos de datos abstractos (TDA's) TDAs lineales y no lineales Matrices dispersas (Ortogonales)	Julio
Estructuras base no lineales	Árboles Definición y terminología	Julio



	Representación gráfica	
Modelado orientado a objetos aplicados a árboles	Diagramas de clases para árboles Relaciones entre nodos Diseño de estructuras dinámicas	Julio
Árboles binarios de búsqueda	Operaciones (inserción, búsqueda, eliminación) Recorridos (Inorden, preorden, posorden)	Agosto
Árboles AVL	Definición Operaciones Rotaciones (simples, dobles)	Agosto
Primer parcial		Agosto
Arquitecturas de despliegue	Modelado UML para despliegue de aplicaciones Diagramas de componentes, Diagramas de despliegue Arquitectura de despliegue stand-alone Arquitectura de despliegue Cliente-Servidor Arquitectura de despliegue N-Capas Arquitectura Cloud	Agosto
Web Development	Conceptos iniciales Introducción al diseño web Introducción a HTML y CSS Principios básicos de HTML Principios básicos de CSS Entendimiento de servidores Web, exploradores, HTTP/HTTPS y FTP Funcionamiento de los servidores Web Web Servers Web Browsers / Exploradores Front End Development Programación del lado del cliente Javascript Back End Development Programación del lado del servidor JSON files XML files Uso de marcadores de hipertexto del lado de la capa de datos	Septiembre
Patrón MVC	Conceptos iniciales (Modelo, Vista, Controlador) Modelado base aplicado al MVC Diagramas de clases. Regla de navegabilidad en capas: Vista > Controlador > Modelo Diagrama de secuencias. Flujo típico: Usuario > Vista > Controlador > Modelo > Vista Diagrama de componentes. Representar MVC como componentes conectados por interfaces. Ejemplos prácticos.	Septiembre
Segundo parcial		Septiembre

2

0

2

6

SEMESTRE

2



Aplicación Web básica	Formularios Manejo de datos Navegación entre páginas Integración de componentes Comunicación entre capas Flujo de información	Septiembre
Semana de Congresos	28/09 a 02/10	
Ambientes de software	Configuración y Despliegue Tipos de ambiente - Desarrollo - Quality Assurance (QA) - Unit Acceptance Testing (UAT) - Producción. Despliegue entre ambientes	Octubre
Introducción a la calidad en pruebas de Software	Conceptos de pruebas de software Objetivos Casos de prueba Validación de entradas y manejo de errores Introducción a pruebas en entornos ágiles	Octubre
Tercer parcial		Octubre
Introducción al testing automatizado	Qué es la automatización de pruebas Beneficios de la automatización Flujo básico de pruebas automatizadas Validación funcional de aplicaciones Web Introducción a IA aplicada a QA y Testing con agentes	Octubre
Seguridad en el Software	Seguridad en el ciclo de vida del software Shift left Security Buenas prácticas de desarrollo seguro Seguridad en aplicaciones Web y móviles	Octubre
Security OWASP	Security OWASp Checklist - Data protection, - file management, - memory management OWASp top 10	Octubre
Período de exámenes finales	6 a 16 /10	



9. Bibliografía

- Introducción al análisis y diseño de algoritmos (2014). Maria Gomez Fuentes
- Python para informáticos Versión 2.7.2 Charles Severance
- UML 2.0 Y Patrones, Craig Larman.
- Fundamentos de programación, Algoritmos, Estructuras de Datos y Objetos, Luis Joyanes Aguilar, Cuarta Edición, McGraw-Hill
- The Scrum Guide™ Ken Schwaber and Jeff Sutherland. 2017
- Craig Larman; Introducción al análisis y diseño orientado a objetos. Prentice Hall
- Software Quality Assurance, From Theory to Implementation, Daniel Galin 2004.
- State of Software Development, CodingSans 2019
- El control de versiones, Guillem Borrell, 2006.
- Software Design, David Budgen, 2ª. Edición, Pearson Addison Wesley.
- OWASP Application Security Verification Standard 4.0, 2019
- Metrics for software conceptual models, Marcela Genero, Mario Piattini, Coral Calero.
- Java y el patrón Modelo Vista y Controlador, 2019, Sonia Alexandra Pinzo Núñez (ISBN 978-958-787-491-4)
- International Software Testing Qualifications Board, 2023, Certified Tester Foundation Level syllabus version 4.0. ISTQB. <https://www.istqb.org/>
- International Software Testing Qualifications Board, 2021. Agile Tester syllabus. ISTQB. <https://www.istqb.org/>
- Agile testing: A practical guide for testers and agile teams. Addison-Wesley, Crispin, L., & Gregory, J., 2009.
- OWASP Foundation, 2021, OWASP Top 10: The ten most critical web application security risks <https://owasp.org/>

10. Normas para la clase virtual

- Todas las Comunicaciones con el profesor y los auxiliares deben ser por los correos electrónicos que se indiquen en clase.
- En toda comunicación escrita se debe mostrar respeto y no utilizar mensajes en mayúsculas.
- Las comunicaciones enviadas por correo electrónico serán atendidas en un máximo de 3 días hábiles.
- Durante las clases los estudiantes deben encender su cámara siempre que el profesor o el auxiliar les hagan una pregunta directa, o bien, cuando el estudiante realice alguna consulta.
- Durante las clases los estudiantes pueden hacer consultas por el chat del curso o por la opción de Questions / Answers, según lo indique el profesor, teniendo el cuidado de ser respetuoso y mantener las reglas de cortesía durante la escritura.