2024

SEMESTRE 2

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

Nombre del Curso: Introducción a la Programación y Computación 2			
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:**	2do. Semestre 2024
Docente:	Msc Ing. Estuardo Zapeta	**Auxiliar:**	Mario Cesar Moran Porras
Horas por semana del curso:	4	**Horas por semana del laboratorio:**	2
Días que se imparte el curso:	Jueves y viernes	**Días que se imparte el laboratorio**	Viernes
Horario del curso:	17:20 - 19:00	**Horario del laboratorio:**	15:40 - 17:20

1. Descripción del curso

Este curso se encuentra diseñado para que el estudiante entienda e implemente estructuras de datos por medio de memoria dinámica, que conceptualmente entienda el diseño de sistemas a través de la definición de una arquitectura y para finalizar la aplicación de versionamiento durante el desarrollo de 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. Implementar estructuras de datos haciendo uso de memoria dinámica.
2. Entender el diseño y análisis de arquitecturas de sistemas.
3. Aplicar buenas prácticas de versionamiento
4. Entender e implementar buenas prácticas de aseguramiento de calidad del software.

3. Metodología

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

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 90% 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 sí 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.

2024 SEMESTRE 2

6. Contenido temático del curso

Unidad	Tema
1. Estructura y manejo de la memoria	**1. Estructura y manejo de la memoria** **1.1. Gestión de la memoria** 1.1.1.1. Particiones estáticas 1.1.1.2. Particiones dinámicas 1.1.1.3. Paginación 1.1.1.4. Segmentación **1.2. Manejo de memoria dinámica** 1.2.1. Apuntadores 1.2.2. Tipos de datos abstractos (TDA's) 1.2.2.1. Concepto 1.2.2.2. Listas simples 1.2.2.3. Listas circulares 1.2.2.4. Listas doblemente enlazadas 1.2.2.5. Pilas 1.2.2.6. Colas
2. Principios de diseño de software	**2. Principios de diseño de software** 2.1. Modelo de análisis en un diseño de software **2.2. Diseño orientado a objetos** 2.2.1. Diagrama de clases 2.2.2. Representación de una clase 2.2.3. Relaciones entre clases 2.2.4. Visibilidad **2.3. Diseño orientado a la interfaz** 2.3.1. Diferentes perspectivas de la interfaz 2.3.2. Principios para el diseño de interfaces de usuario **2.4. Diseño orientado a la arquitectura** 2.4.1. Concepto 2.4.2. Componentes 2.4.3. Atributos de calidad 2.4.4. Ciclo de desarrollo de la arquitectura 2.4.5. Patrones y estilos arquitectónicos 2.4.6. Arquitectura en la nube **2.5. Diseño orientado a los datos** 2.5.1. Modelo de datos 2.5.2. Modelo relacional 2.5.2.1. Atributos 2.5.2.2. Dominio 2.5.2.3. Tipos de relación 2.5.2.4. Notación

3. Estrategias de desarrollo de software	**3. Estrategias de desarrollo de software** 3.1. Proyecto de software 3.2. Metodología para el desarrollo de software 3.3. Ciclo de vida para el desarrollo de software 3.4. Modelo cascada 3.5. Metodología SCRUM 3.5.1. Pilares fundamentales 3.5.2. Roles 3.5.3. Conceptos importantes 3.5.4. Artefactos 3.5.5. Eventos **3.6. Principios básicos de versionamiento** 3.6.1. Concepto 3.6.2. ¿Qué sucede si no hay versionamiento? 3.6.3. Términos clave
4. Aseguramiento de la calidad del software	**4. Aseguramiento de la calidad del software** 4.1. Calidad del software 4.2. ¿Cómo se asegura la calidad del software? 4.3. Funciones del equipo de aseguramiento 4.4. Asegurando calidad a través de las pruebas 4.5. Principios de las pruebas 4.6. Modelo “V” de pruebas 4.7. Clasificación de las pruebas 4.8. Herramientas de medición de calidad

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	04 pts.
	Prácticas	03 pts.
	Primer parcial	12 pts.
	Segundo parcial	13 pts.
	Tercer parcial	13 pts.
Total de clase		**45 pts.**
Laboratorio	Proyecto I	10 pts.
	Proyecto II	10 pts.
	Proyecto III	10 pts.
Total de laboratorio		**30 pts.**
Zona		75 pts.
Examen Final		25 pts.
Nota de promoción		100 pts.

8. Cronograma de actividades

Tema principal	Contenido a desarrollar	Fecha
Presentación del curso		18-julio
Gestión de la memoria	Particiones estáticas Particiones dinámicas	19-julio
	Paginación Segmentación	25-julio
Manejo de memoria dinámica	Apuntadores	26-julio
		01-agosto
	Tipos de datos abstractos (TDA's) - Concepto - Listas simples	02-agosto
	Listas circulares	08-agosto
		09-agosto
	Listas doblemente enlazadas	16-agosto
		22-agosto
	- Pilas - Colas	23-agosto
Primer parcial		**29-agosto**
Principios de diseño de software	Modelo de análisis en un diseño de software Diseño orientado a objetos - Diagrama de clases	30-agosto
	- Representación de una clase - Relaciones entre clases - Visibilidad	05-septiembre
	Diseño orientado a la interfaz - Diferentes perspectivas de la interfaz - Principios para el diseño de interfaces de usuario	06-septiembre
	Diseño orientado a la arquitectura - Concepto - Componentes - Atributos de calidad - Ciclo de desarrollo de la arquitectura	12-septiembre
	- Patrones y estilos arquitectónicos - Arquitectura en la nube	19-septiembre
	Diseño orientado a los datos - Modelo de datos	20-septiembre

	- Modelo relacional - Atributos - Dominio	03-octubre
	- Tipos de relación - Notación	04-octubre
Segundo parcial		**10-octubre**
Estrategias de desarrollo de software	Proyecto de software Metodología para el desarrollo de software Ciclo de vida para el desarrollo de software Modelo cascada	11-octubre
	Metodología SCRUM - Pilares fundamentales - Roles - Conceptos importantes	17-octubre
	- Artefactos - Eventos	17-octubre
	Principios básicos de versionamiento - Concepto - ¿Qué sucede si no hay versionamiento? - Términos clave	18-octubre
Aseguramiento de la calidad del software	Calidad del software ¿Cómo se asegura la calidad del software? Funciones del equipo de aseguramiento	24-octubre
	Asegurando calidad a través de las pruebas Principios de las pruebas Modelo “V” de Pruebas Clasificación de las pruebas	25-octubre
	Herramientas de medición de calidad	25-octubre
Tercer parcial		**31-octubre**

9. Bibliografía

- Python para informáticos Versión 2.7.2 Charles Severance
- 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 Sofware Development, CodingSans 2019
- El control de versions, Guillem Borrell, 2006.
- Software Design, David Budgen, 2ª. Edición, Pearson Addison Wesley.
- OWASP Application Security Verification Standard 4.0, 2019

10. Normas para la clase semipresencial

- Todas las comunicaciones con el profesor y los auxiliares deben ser por los correos electrónicos que se indiquen en clase.
- Las comunicaciones enviadas por correo electrónico serán atendidas en un máximo de 3 días hábiles.
- Los exámenes parciales y final serán en modalidad presencial.
- Durante las clases los estudiantes pueden hacer consultas por el chat del curso o habilitando su micrófono, según lo indique el profesor, teniendo el cuidado de ser respetuoso y mantener las reglas de cortesía durante la escritura.