2022

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 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:**	2do. Semestre 2022
Docente:	Msc Ing. Estuardo Zapeta	**Auxiliar:**	José Carlos Estrada García
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:**	19:00-20:40

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 **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. 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 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.

2022 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. Modelos de administración de memoria** 1.1.1. Administración física 1.1.2. Administración lógica 1.1.2.1. Reclamo de memoria 1.1.2.2. Dellocation Controlada 1.1.2.3. Manejo automático de la memoria 1.1.2.4. Garbage Collector **1.2. Manejo de Memoria dinámica** 1.2.1. Apuntadores 1.2.1.1. Los índices y el apuntador simple 1.2.1.2. El apuntador subíndice 1.2.1.3. Almacenamiento 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 doblemente enlazadas 1.2.2.4. Listas circulares 1.2.2.5. Pilas 1.2.2.6. Colas 1.2.2.7. Listas ortogonales 1.2.2.8. Listas n-encadenadas 1.2.2.9. Ordenamientos
2. Principios de diseño de software	**2. Principios de Diseño de Software** **2.1. Diseño con Objetos** 2.1.1. Frameworks orientados a objetos 2.1.2. Diseños basados en objetos 2.1.3. Diseños orientados a objetos **2.2. Modelos de arquitecturas de despliegue del software** 2.2.1. Arquitectura Stand Alone 2.2.2. Arquitectura Cliente-Servidor 2.2.3. Arquitectura de N-Capas 2.2.4. Arquitectura Cloud
3. Arquitecturas para soluciones de software	**3. Arquitecturas para soluciones de Software** **3.1. Aplicaciones de escritorio** **3.2. Aplicaciones Cliente-Servidor** **3.3. Web Development** 3.3.1. Conceptos iniciales 3.3.1.1. Introducción al diseño web 3.3.1.2. Introducción a HTML y CSS

	3.3.1.2.1. Principios básicos de HTML 3.3.1.2.2. Principios básicos de CSS 3.3.1.3. Servidores Web, Exploradores, HTTP y FTP 3.3.1.3.1. Funcionamiento de los servidores Web 3.3.1.3.2. Web Servers 3.3.1.3.3. Web Browsers / Exploradores 3.3.2. Front End Development 3.3.2.1. Programación del lado del Servidor 3.3.2.2. Javascript 3.3.3. Back End Development 3.3.3.1. Programación del lado del servidor 3.3.3.2. Introducción a las bases de datos y XML 3.3.3.2.1. Conceptos de bases de datos 3.3.3.2.2. Tipos de bases de datos 3.3.3.2.3. XML y su uso en desarrollo web
4. Estrategias de desarrollo de software	**4. Estrategias de Desarrollo de Software** **4.1. Ambientes de Software Configuración y Despliegue** 4.1.1. Tipos de ambientes 4.1.1.1. Desarrollo 4.1.1.2. Quality Assurance (QA) 4.1.1.3. Unit Acceptance Testing (UAT) 4.1.1.4. Producción 4.1.2. Despliegue entre ambientes 4.1.3. Consideraciones, aprobaciones y aseguramiento **4.2. Metodologías de desarrollo** 4.2.1. Modelo cascada 4.2.2. Scrum 4.2.2.1. Conceptos básicos 4.2.2.2. Valores 4.2.2.3. Equipo 4.2.2.4. Eventos **4.3. Principios básicos de versionamiento** 4.3.1. Metodologías de numeración de versiones 4.3.2. Criterio para modificación de versiones 4.3.3. Versionado de productos complejos 4.3.4. Clasificación para versiones no estables 4.3.5. Versionado de parches 4.3.6. Sistema de versionado de código centralizado
5. Security & Quality Assurance	**5. Security & Quality Assurance** **5.1. Security OWASp Checklist** 5.1.1. Data protection 5.1.2. File management 5.1.3. Memory management **5.2. Quality**

	5.2.1.	Desarrollo y plan de calidad
	5.2.2.	Estrategias de pruebas de software
	5.2.2.1.	White box, black box testing
	5.2.3.	Diseño de casos de pruebas
	5.2.4.	Pruebas de regresión
	5.2.5.	Pruebas de rendimiento
	5.2.6.	Pruebas de stress
	5.2.7.	Pruebas Alpha y Beta

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		21-julio
Modelos de administración de memoria	Administración física de la memoria Administración lógica de la memoria	22-julio
Memoria dinámica	Apuntadores - Los índices y el apuntador simple - El apuntador subíndice - Almacenamiento	28-julio
	Tipos de datos abstractos (TDA's) - Concepto - Listas simples	29-julio
	Listas doblemente enlazadas	04-agosto
	Listas circulares	05-agosto
	- Pilas - Colas	11-agosto
	- Listas ortogonales	12-agosto
	- Ordenamientos	18-agosto
Primer parcial		**19-agosto**
Principios de diseño de software	Diseño con Objetos - Frameworks orientados a objetos	25-agosto

	- Diseños basados en objetos - Diseños orientados a objetos	
	Modelos de arquitecturas de despliegue del software - Arquitectura Stand Alone - Arquitectura Cliente-Servidor - Arquitectura de N-Capas - Arquitectura Cloud	01-septiembre
Arquitectura de aplicaciones web	Aplicaciones de escritorio Aplicaciones cliente-servidor	
	Web Development - Conceptos iniciales - Introducción al diseño web - Introducción a HTML y CSS - Entendimiento de servidores Web, Exploradores, HTTP y FTP	02-septiembre
	- Front-end development	
	- Back-end development	
	- Introducción a las bases de datos, XML y JSON	08-septiembre
	- Modelos de datos	09-septiembre
	- Tipos de bases de datos	16-septiembre
	- XML y su uso en desarrollo web	22-septiembre
	- Conceptos básicos de JSON	
Segundo parcial		**23-septiembre**
Estrategias de desarrollo de software	Ambientes de Software Configuración y Despliegue	06-octubre
	- Tipos de ambientes - Despliegue entre ambientes	07-octubre
	- Consideraciones, aprobaciones y aseguramiento	13-octubre
	Metodologías de desarrollo - Modelo cascada	14-octubre
	- Scrum	21-octubre
	Principios básicos de versionamiento	27-octubre
Security & Quality Assurance	- Security OWASp Checklist - Quality - Diseño de casos de pruebas	
	- Pruebas de regresión - Pruebas de rendimiento	28-octubre
	- Pruebas de stress - Pruebas Alpha y Beta	03-noviembre
Tercer parcial		**04-noviembre**

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 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 los exámenes los estudiantes deben mantenerse conectados por Meet.
- 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.