



**Nombre del Cuso: Introducción a la Programación y Computación II**

<b>Código:</b>	771	<b>Créditos:</b>	5
<b>Escuela:</b>	CIENCIAS Y SISTEMAS	<b>Área a la que pertenece:</b>	Programación
<b>Pre requisito:</b>	Introducción a la Programación y Computación I (770) Matemática Intermedia (107) Lógica Matemática (795) Matemática de Computo 1 (960)	<b>Post requisito:</b>	Organización Computacional (964) Estructura de Datos (772) Org. Lenguajes y Compiladores 1 (777)
<b>Categoría:</b>	Obligatorio	<b>Semestre:</b>	2do. Semestre 2021
<b>Docente:</b>	Byron Rodolfo Zepeda Arévalo	<b>Auxiliar:</b>	Edwar Everaldo Zacarias Javier Estuardo Lima Abrego
<b>Edificio:</b>	-	<b>Sección:</b>	D
<b>Salón del curso:</b>	meet	<b>Salón de laboratorio:</b>	meet
<b>Horas por semana del curso:</b>	4	<b>Horas por semana del laboratorio:</b>	2
<b>Días que se imparte el curso:</b>	Jueves y viernes	<b>Días que se imparte el laboratorio</b>	Sábados
<b>Horario del curso:</b>	07:10 - 08:50	<b>Horario del laboratorio:</b>	8:50 - 10:30

**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

**Específicos**

1. Modelar problemas de forma estándar y profesional

- desarrollo, así como elementos de calidad que le ayuden a robustecer los entregables
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 periodos 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 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
<p><b>1. Estructura y manejo de la memoria</b></p>	<p><b>1. Modelos de administración de memoria</b></p> <ul style="list-style-type: none"> <li>1.1. Administración física de la memoria</li> <li>1.2. Administración lógica de la memoria                             <ul style="list-style-type: none"> <li>1.2.1. Reclamo de memoria</li> <li>1.2.2. Dellocation Controlada</li> <li>1.2.3. Manejo automático de la memoria</li> <li>1.2.4. Garbage Collector</li> </ul> </li> </ul> <p><b>2. Manejo de Memoria dinámica</b></p> <ul style="list-style-type: none"> <li>2.1. Apuntadores                             <ul style="list-style-type: none"> <li>2.1.1. Los índices y el apuntador simple</li> <li>2.1.2. El apuntador subíndice</li> <li><b>2.1.3. Almacenamiento</b></li> </ul> </li> <li>2.2. Tipos de datos abstractos (TDA's)                             <ul style="list-style-type: none"> <li>2.2.1. Concepto</li> <li>2.2.2. Listas simples</li> <li>2.2.3. Listas doblemente enlazadas</li> <li>2.2.4. Listas circulares</li> <li>2.2.5. Pilas</li> <li>2.2.6. Colas</li> <li>2.2.7. Listas ortogonales</li> <li>2.2.8. Listas n-encadenadas</li> <li><b>2.2.9. Ordenamientos</b></li> </ul> </li> </ul>
<p><b>2. Principios de Diseño de Software</b></p>	<p><b>1. Diseño con Objetos</b></p> <ul style="list-style-type: none"> <li>1.1. Frameworks orientados a objetos</li> <li>1.2. Diseños basados en objetos</li> <li><b>1.3. Diseños orientados a objetos</b></li> </ul> <p><b>2. Modelos de arquitecturas de despliegue del software</b></p> <ul style="list-style-type: none"> <li>2.1. Arquitectura Stand Alone</li> <li>2.2. Arquitectura Cliente-Servidor</li> <li>2.3. Arquitectura de N-Capas</li> <li><b>2.4. Arquitectura Cloud</b></li> </ul>
<p><b>3. Arquitecturas para soluciones de Software</b></p>	<p><b>1. Aplicaciones de escritorio</b></p> <p><b>2. Aplicaciones Cliente-Servidor</b></p> <p><b>3. Web Development</b></p> <ul style="list-style-type: none"> <li>3.1. Conceptos iniciales                             <ul style="list-style-type: none"> <li>3.1.1. Introducción al diseño web</li> <li>3.1.2. Introducción a HTML y CSS                                     <ul style="list-style-type: none"> <li>3.1.2.1. Principios básicos de HTML</li> <li>3.1.2.2. Principios básicos de CSS</li> </ul> </li> </ul> </li> </ul>



	<ul style="list-style-type: none"> <li>3.1.3. Entendimiento de servidores Web, Exploradores, HTTP y FTP                         <ul style="list-style-type: none"> <li>3.1.3.1. Funcionamiento de los servidores Web</li> <li>3.1.3.2. Web Servers</li> <li>3.1.3.3. Web Browsers / Exploradores</li> </ul> </li> <li>3.2. Front End Development                         <ul style="list-style-type: none"> <li>3.2.1. Programación del lado del Servidor</li> <li>3.2.2. Javascript</li> </ul> </li> <li>3.3. Back End Development                         <ul style="list-style-type: none"> <li>3.3.1. Programación del lado del servidor</li> <li>3.3.2. Introducción a las bases de datos y XML                                 <ul style="list-style-type: none"> <li>3.3.2.1. Conceptos de bases de datos</li> <li>3.3.2.2. Tipos de bases de datos</li> <li>3.3.2.3. XML y su uso en desarrollo web</li> </ul> </li> </ul> </li> </ul>
<p>4. Estrategias de Desarrollo de Software</p>	<ul style="list-style-type: none"> <li>1. <b>Ambientes de Software Configuración y Despliegue</b> <ul style="list-style-type: none"> <li>1.1. Tipos de ambientes                             <ul style="list-style-type: none"> <li>1.1.1. Desarrollo</li> <li>1.1.2. Quality Assurance (QA)</li> <li>1.1.3. Unit Acceptance Testing (UAT)</li> <li>1.1.4. Producción</li> </ul> </li> <li>1.2. Despliegue entre ambientes</li> <li>1.3. Consideraciones, aprobaciones y aseguramiento</li> </ul> </li> <li>2. <b>Metodologías de desarrollo</b> <ul style="list-style-type: none"> <li>2.1. Modelo cascada</li> <li>2.2. Scrum                             <ul style="list-style-type: none"> <li>2.2.1. Conceptos básicos</li> <li>2.2.2. Valores</li> <li>2.2.3. Equipo</li> <li>2.2.4. Eventos</li> </ul> </li> </ul> </li> <li>3. Principios básicos de versionamiento                         <ul style="list-style-type: none"> <li>3.1. Metodologías de numeración de versiones</li> <li>3.2. Criterio para modificación de versiones</li> <li>3.3. Versionado de productos complejos</li> <li>3.4. Clasificación para versiones no estables</li> <li>3.5. Versionado de parches</li> <li>3.6. Sistema de versionado de código centralizado</li> </ul> </li> </ul>
<p>5. Security &amp; Quality Assurance II</p>	<ul style="list-style-type: none"> <li>1. <b>Security OWASp Checklist</b> <ul style="list-style-type: none"> <li>1.1. Data protection</li> <li>1.2. File management</li> <li>1.3. Memory management</li> </ul> </li> <li>2. <b>Quality</b> <ul style="list-style-type: none"> <li>2.1. Desarrollo y plan de calidad</li> <li>2.2. Estrategias de pruebas de software                             <ul style="list-style-type: none"> <li>2.2.1. White box, black box testing</li> </ul> </li> <li>2.3. Diseño de casos de pruebas</li> <li>2.4. Pruebas de regresión</li> <li>2.5. Pruebas de rendimiento</li> <li>2.6. Pruebas de stress</li> </ul> </li> </ul>



---

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

## 8. Cronograma de actividades

Tema principal	Contenido a desarrollar	Fecha
Modelos de administración de memoria	Administración física de la memoria Administración lógica de la memoria	23-julio
Memoria dinámica	Apuntadores - Los índices y el apuntador simple - El apuntador subíndice - Almacenamiento	29-julio
	Tipos de datos abstractos (TDA's) - Concepto	30-julio
	- Listas simples	5-agosto
	- Listas doblemente enlazadas	
	- Listas circulares	6-agosto
	- Pilas	
- Colas	12-agosto	
- Listas ortogonales		
- Listas n-encadenadas	13-agosto	
- Ordenamientos		
Feriado local	15 y 16 de agosto	
<b>Primer parcial</b>	<b>20-agosto</b>	
Principios de diseño de software	Diseño con Objetos	19-agosto
	- Frameworks orientados a objetos - Diseños basados en objetos	26-agosto



	<ul style="list-style-type: none"> <li>- Diseños orientados a objetos</li> <li>Modelos de arquitecturas de despliegue del software</li> <li>- Arquitectura Stand Alone</li> <li>- Arquitectura Cliente-Servidor</li> <li>- Arquitectura de N-Capas</li> <li>- Arquitectura Cloud</li> </ul>	27-agosto 2-septiemb.
Arquitectura para aplicaciones WEB	<ul style="list-style-type: none"> <li>Aplicaciones de escritorio</li> <li>Aplicaciones Cliente-Servidor</li> <li>Web Development</li> <li>- Conceptos iniciales</li> <li>- Introducción al diseño web</li> <li>- Introducción a HTML y CSS</li> <li>- Entendimiento de servidores Web, Exploradores, HTTP y FTP</li> <li>- Front End Development</li> <li>- Programación del lado del Servidor</li> <li>- Back End Development</li> <li>- Programación del lado del servidor</li> <li>- Introducción a las bases de datos, XML y JSON</li> <li>- Conceptos de bases de datos</li> <li>- Tipos de bases de datos</li> <li>- XML y su uso en desarrollo web</li> <li>- Conceptos básico de JSON</li> </ul>	3-sept.  9-sept.  10-sept 16-sept 17-sept  23-sept 7-octubre 8-octubre 14-octubre
<b>Segundo parcial</b>	<b>24-septiembre</b>	
Día de la independencia	14 y 15 de septiembre	
Semana de congresos	27 de septiembre al 2 de octubre	
Estrategias de desarrollo de software	<ul style="list-style-type: none"> <li>Ambientes de Software Configuración y Despliegue</li> <li>- Tipos de ambientes</li> <li>- Despliegue entre ambientes</li> <li>- Consideraciones, aprobaciones y aseguramiento</li> <li>Metodologías de desarrollo</li> <li>- Modelo cascada</li> <li>- Scrum</li> <li>Principios básicos de versionamiento</li> </ul>	15-octubre 21-octubre  22-octubre 28-octubre
<b>Tercer parcial</b>	<b>29-octubre</b>	
Security & Quality Assurance II	<ul style="list-style-type: none"> <li>Security OWASp Checklist</li> <li>Quality</li> <li>- Diseño de casos de pruebas</li> <li>- Pruebas de regresión</li> <li>- Pruebas de rendimiento</li> <li>- Pruebas de stress</li> <li>- Pruebas Alpha y Beta</li> </ul>	4-nov. 5-nov.
Día de la revolución	20 de octubre	



Último día de clases	5 de noviembre	
Período de exámenes finales	Del 8 al 19 de 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 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

## 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 mantener encendida su cámara y estar conectados a la sesión de Google Meet durante todo el tiempo de evaluación.
- 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.