

PROGRAMA DE LABORATORIO	
<b>Nombre del curso:</b>	Introducción a la programación y computación 1

<b>Código:</b>	0770	<b>Créditos</b>	4
<b>Escuela:</b>	Ciencias y Sistemas	<b>Área a la que pertenece:</b>	Desarrollo de Software
<b>Prerequisito</b>	33 créditos y 0103 Matemática Básica 2	<b>Post requisitos:</b>	0771 Introducción a la Programación y computación 2 0796 Lenguajes Formales y de programación
<b>Categoría:</b>	Obligatorio	<b>Vigencia:</b>	Primer Semestre 2023
<b>Catedrático:</b>	Ver anexo	<b>Auxiliar:</b>	Ver anexo
<b>Edificio:</b>	Virtual	<b>Secciones</b>	A, B, C, D, E, F, G
<b>Salon del curso:</b>	Meet	<b>Salon del laboratorio:</b>	Meet
<b>Horas por sema del curso magistral:</b>	4	<b>Horas por semana del laboratorio:</b>	2
<b>Horario del curso magistral:</b>	Martes y jueves	<b>Día que se imparte el laboratorio:</b>	Martes
<b>Horarios del curso magistral:</b>	07:10 - 08:50	<b>Horario del laboratorio:</b>	10:40 - 12:20

Descripción del curso
<p>El curso busca ser el acercamiento inicial del estudiante de la carrera de sistemas, al mundo de Desarrollo de Software mediante el uso de métodos, técnicas y metodologías especializadas. Se fundamenta en el concepto de algoritmo para la resolución de problemas de programación utilizando computadoras, enfatizando el uso del paradigma de Programación Orientado a Objetos. Se acerca al estudiante al conocimiento de los principales algoritmos de búsquedas y ordenamientos. Asimismo, el estudiante conocerá el lenguaje Java como el lenguaje oficial de programación del curso.</p>

Objetivos:
<p><b>General</b></p> <ul style="list-style-type: none"> <li>Adquirir, por parte del estudiante, la destreza de analizar, diseñar y codificar software de alta calidad independientemente de la plataforma y lenguaje de programación</li> </ul>

fundamentado en los conocimientos básicos de la programación utilizando el Paradigma Orientado a Objetos.

**Específicos:**

1. Integrar al estudiante a la tecnología de la computación.
2. Conocer las diferentes metodologías de software.
3. Analizar los problemas bajo la perspectiva de Programación Orientada a Objetos.
4. Diseñar soluciones elegantes basadas en el entendimiento de proceso de Análisis
5. Organizar soluciones utilizando un lenguaje de programación oficial y complementario.

**Metodología:**

- Clases Virtuales, Martes y Jueves, usando Meet-Usac con apoyo de la plataforma UEDI.
- Elaboración de investigaciones y tareas.
- Práctica de exámenes cortos y parciales.
- Laboratorio y talleres.
- Elaboración de proyectos de programación
- Elaboración de prácticas cortas de programación
- Cursos complementarios extra aula

**Requisitos:**

- El desarrollo de las actividades es de carácter individual. Todas las entregas serán evaluadas por copias entre secciones. Las copias parciales o totales tienen nota de 0 y reporte a la Escuela de Ciencias y Sistemas.
- El laboratorio se aprueba con 61 puntos.
- Las actividades por realizar en el laboratorio (tareas, prácticas, y proyectos) estarán coordinadas entre secciones.
- La forma de entrega de las actividades será vía UEDI, según la fecha y hora límite de entrega, indicada en el enunciado de cada actividad.
- Para la calificación de las actividades se tomará en cuenta la presentación, calidad, tolerancia.

**Aislamiento de rendimiento académico**

Descripción	Publicación	Entrega	Punteo
Practica 1: Arreglos y Ciclos	28/01/2023	10/02/2023	10
Practica 2: Recursividad e hilos	11/03/2022	26/03/2022	10
<b>Total de prácticas:</b>			<b>20</b>
Proyecto 1:	11/02/2023	10/03/2023	15
Proyecto 2:	26/03/2023	03/05/2023	25
<b>Total de proyectos:</b>			<b>40</b>

Tarea 1:	04/02/2023	09/02/2023	2.5
Tarea 2:	25/02/2023	01/03/2023	2.5
Tarea 3:	18/03/2023	23/03/2023	2.5
Tarea 4:	15/04/2023	20/04/2023	2.5
<b>Total tareas:</b>			<b>10</b>
Corto 1	—	26/02/2023	10
Corto 2	—	26/03/2023	10
<b>Total de Cortos:</b>			<b>20</b>
<b>Examen Final (30/04/2023)</b>			<b>10</b>
<b>Total:</b> El laboratorio se gana con 61 pts. de 100. <b>Para ganar el laboratorio se debe de contar con un 80% de asistencia.</b>			<b>100</b>

Contenido
<p><b>1. Fundamentos de Programación y JAVA- (21/01/2023 - 03/02/2023)</b></p> <ul style="list-style-type: none"> <li>1.1. ¿Qué es Java?</li> <li>1.2. Versiones y Ambiente de Java (JDK, JRE, JVM).</li> <li>1.3. Características de Java</li> <li>1.4. Tipos de programas en Java</li> <li>1.5. Componentes del lenguaje Java</li> <li>1.6. Introducción a la programación</li> <li>1.7. Elementos del lenguaje <ul style="list-style-type: none"> <li>1.7.1. Identificadores, Comentarios</li> <li>1.7.2. Tipos de datos, Constantes <ul style="list-style-type: none"> <li>1.7.2.1. Datos Primitivos</li> <li>1.7.2.2. Datos No Primitivos <ul style="list-style-type: none"> <li>1.7.2.2.1. Arreglos <ul style="list-style-type: none"> <li>1.7.2.2.1.1. Concepto</li> <li>1.7.2.2.1.2. Operaciones <ul style="list-style-type: none"> <li>1.7.2.2.1.2.1. Ordenamiento <ul style="list-style-type: none"> <li>1.7.2.2.1.2.1.1. Bruja</li> <li>1.7.2.2.1.2.1.2. Por inserción</li> <li>1.7.2.2.1.2.1.3. Por Selección</li> <li>1.7.2.2.1.2.1.4. Quick Sort</li> </ul> </li> </ul> </li> </ul> </li> </ul> </li> <li>1.7.2.2.2. Listas dinámicas</li> </ul> </li> </ul> </li> <li>1.7.3. Operadores, Prioridad de Operadores</li> </ul> <li>1.8. Input y output</li> <li>1.9. Condiciones</li> <li>1.10. Estructuras de Control <ul style="list-style-type: none"> <li>1.10.1. if - else</li> <li>1.10.2. switch</li> </ul> </li> <li>1.11. Ciclos <ul style="list-style-type: none"> <li>1.11.1. for</li> <li>1.11.2. while</li> <li>1.11.3. do - while</li> </ul> </li> <li>1.12. Procedimientos y Funciones</li>

1.13. Recursividad	1.13.1. Recursividad Simple	1.13.2. Recursividad Doble
1.14. Manejo de memoria	1.14.1. Memoria Estática	
1.15. HTML y CSS	1.15.1. Etiquetas Básicas	1.15.2. Frameworks
<b>2. Versionamiento - (04/02/2023 - 10/02/2023)</b>		
2.1. Introducción a versionamiento		
2.2. Herramientas de control de versiones	2.2.1. Git	2.2.2. Github, Gitlab
	2.2.3. Git Kraken	
2.3. Tareas básicas	2.3.1. Creación de repositorio	2.3.2. Commit
	2.3.2.1. Working directory	2.3.2.2. Staging area
	2.3.2.3. Repository	
2.3.3. Creación de ramas		
2.4. Colaboración en repositorios remotos	2.4.1. Clonación de repositorio	2.4.2. Resolución de conflictos
<b>3. Manejo de errores, debug y pruebas - (04/02/2023 - 10/02/2023)</b>		
3.1. Expresiones (Try-Catch, etc.)		
3.2. Debugging	3.2.1. Breakpoint	3.2.1.1. Inline
	3.2.1.2. Function	3.2.1.2.1. Entrar a función
		3.2.1.2.2. Salir de función
	3.2.2. Start	
	3.2.3. Pause	
	3.2.4. Continue	
	3.2.5. Stop	
3.3. Testing	3.3.1. Unit test	3.3.2. Functional test
	3.3.3. Integration test	
<b>4. Programación Orientada a Objetos (POO) - (11/02/2023 - 10/03/2023)</b>		
4.1. Concepto de abstracción y clasificación		
4.2. Clases y objetos		
4.3. Mensajes y métodos		
4.4. El principio el encapsulamiento		
4.5. Los miembros de una clase	4.5.1. Atributos	4.5.2. Métodos (operaciones)
	4.5.3. Constructores y destructores	
4.6. Modificadores de visibilidad	4.6.1. Privado	4.6.2. Público
	4.6.3. Protegido	
4.7. Relaciones entre clases y objetos	4.7.1. Asociación	4.7.2. Agregación y composición

4.7.3. Herencia
4.8. Polimorfismo
4.8.1. Sobrecarga de métodos
4.8.2. Virtualización
4.9. Construcciones abstractas
4.9.1. Clase abstracta
4.9.2. Interface
4.10. Conceptos avanzados
4.10.1. Miembros estáticos (static) y miembros de instancia
4.10.2. Referencia "this"
4.10.3. Clases paramétricas (plantilla de clases)
4.11. Principios básicos de UML (diagrama de clases)
4.11.1. Definición de clases y sus relaciones
4.11.2. Ámbito de las propiedades, Métodos
4.11.3. Diseño de programas
4.11.4. Asociaciones y restricciones, clases de asociaciones, multiplicidad, dependencia.
4.11.5. Relaciones múltiples (asociativas) y reflexivas
<b>5. Introducción a AWT y SWING - (11/03/2023 - 24/03/2023)</b>
5.1. Librerías de interfaz gráfica
5.2. Componentes de interfaz gráfica
5.3. Eventos
5.4. Construcción de aplicaciones con interfaz gráfica
<b>6. Manejo de Hilos en Java - (25/03/2023 - 14/04/2023)</b>
6.1. Concepto de Manejo de hilos
6.1.1. Condición de carrera
6.2. Hilos en Java
6.3. Animación usando hilos
<b>7. Manejo de Archivos - (15/04/2023 - 28/04/2023)</b>
7.1. Introducción al flujo de datos
7.2. Almacenamiento de datos
7.3. Archivos de texto plano
7.3.1. Abrir y cerrar
7.3.2. Lectura
7.3.3. Escritura
7.3.4. Eliminación
<b>8. Cloud Computing - (29/04/2023 - 05/05/2023)</b>
8.1. Redes
8.2. Protocolos de red
8.3. Antecedentes e Historia
8.4. Definición
8.5. Funcionamiento
8.6. Tipos de nube
8.6.1. Pública
8.6.2. Privada
8.6.3. Híbrida
8.7. Modelos en la nube
8.7.1. SaaS
8.7.2. PaaS
8.7.3. IaaS
8.8. Servicios en la nube
8.9. Proveedores de nube

### Cláusulas Restrictivas

El perfil del estudiante de la facultad de Ingeniería de la Universidad de San Carlos de Guatemala exige una alta calidad en la excelencia académica y ética profesional. Se establecen en este curso los siguientes lineamientos que regulan el comportamiento del estudiante:

- Copias en prácticas y proyectos.
- No hay prórrogas.
- No hay reposición de proyectos.

Es obligatorio ganar el laboratorio para tener derecho a evaluación total del curso.

### Puntos importantes a considerar:

- Para tener derecho a nota de laboratorio se debe cumplir con el 80% de asistencia a clase de laboratorio, a excepción de presentar una justificación y constancia.
- No se aceptarán entregas tarde sobre tareas, prácticas, exámenes cortos, exámenes finales y proyectos sin justificación. El tutor académico puede aplicar la penalización que considere apropiada.
- El medio de entrega oficial para las actividades es la plataforma UEDI de la facultad y formulario de Google proporcionado en el enunciado de cada actividad.
- Todo proyecto será verificado para validar la creación de este.
- Se realizará un seguimiento a las dudas planteadas en laboratorio sobre prácticas o proyectos.
- Copias obtendrán una nota de 0 y reportará a la Escuela de Ciencias y Sistemas.

### Bibliografía:

- JOYANES, L. y ZAHONERO, I. "**Programación en Java 2 (algoritmos, estructura de datos y programación orientada a objetos)**". España, McGraw-Hill / Interamericana de España, S. A. 2002, PP 725
- BUDD, Timothy. "**Introducción a la programación orientada a objetos**", EUA, Addison, Wesley, Iberoamericana, S. A. 1994, P. 409
- JOYANES, L. "**Programación en Turbo Pascal Versiones 5.5, 6.0, y 7.0**", (2da Edición), México, McGraw-Hill / Interamericana de España, S. A. 1995, PP. 914
- Deitel & Deitel. "**Cómo Programar en Java**" (7ma Edición), México, Prentice Hall 2008, PP. 1280
- McLaughlin, B.; Pollice, G. y West, D. "**Head First Object-Oriented Analysis & Design**", EUA, O'Reilly Media 2006, PP. 636
- Freeman, E.; Robson, E.; Bates, B. y Sierra, K. "**Head First Design Patterns**", EUA, O'Reilly
- Manuales de Referencia de Java, <<http://www.sun.com/java>>.
- Cualquier otro material (escrito o digital) entregado en clase.

Sección	Catedrático	Auxiliar
A	Marlon Fransisco Orellana Lopez	José Daniel Alvarado Fajardo
B	William Estuardo Escobar Argueta	Diego Fernando Cortez Lopez
C	Moises Eduardo Velasquez Oliva	José Alejandro Santizo Cotto
D	Herman Igor Veliz Linares	Salvador de Jesus López Bautista

		José Eduardo Morales García
<b>E</b>	Neftali de Jesus Calderon Mendez	Erick Daniel Antillon Chinchilla
<b>F</b>	William Estuardo Escobar Argueta	Freddy Alejandro Monterroso
<b>G</b>	Edgar Francisco Rodas Robledo	Christofer William Borrayo López

**Cronograma de actividades:**

	16/01	21/01	28/01	4/02	11/02	18/02	25/02	4/03	11/03	18/03	25/03	1/04	8/04	15/04	22/04	29/04
	20/01	27/01	3/02	10/02	17/02	24/02	3/03	10/03	17/03	24/03	31/03	7/04	14/04	21/04	28/04	5/05
Presentación																
Unidad 1																
Unidad 2																
Unidad 3																
Unidad 4																
Unidad 5																
Unidad 6																
Unidad 7																
Unidad 8																
Tarea 1																
Tarea 2																
Tarea 3																
Tarea 4																
Corto 1																
Corto 2																
Examen Final																
Práctica 1																
Proyecto 1																
Práctica 2																
Proyecto 2																