

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>	0103 Matemática Básica 2 0147 Física Básica 0960 Matemática para Computación 1	<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 2027
<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>Salón del curso:</b>	Meet	<b>Salón 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>	A, F: Lunes y Miércoles B,C,D,E: Martes y Jueves G: Sábado	<b>Día que se imparte el laboratorio:</b>	C, E, G: Jueves A, B, D, F: Viernes
<b>Horarios del curso magistral:</b>	A,B,C,D,E: 07:10 - 08:50 F: 11:30 -13:10 G: 13:00 - 16:20	<b>Horario del laboratorio:</b>	DEPENDE POR SECCIÓN

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:****General**

- Desarrollar en el estudiante la capacidad de analizar, diseñar y codificar software de alta calidad, aplicando los principios fundamentales de la Programación Orientada a Objetos, sin depender de una plataforma o lenguaje de programación específico

**Específicos:**

1. Fomentar la integración del estudiante en el ámbito de las tecnologías computacionales, promoviendo su familiarización con herramientas y conceptos clave en el área.
2. Explorar y comprender diferentes metodologías de desarrollo de software, evaluando su aplicación en diversos contextos y proyectos.
3. Aplicar la perspectiva de la Programación Orientada a Objetos (POO) para analizar y resolver problemas de manera estructurada y eficiente.
4. Diseñar soluciones innovadoras y efectivas, basadas en un profundo entendimiento del proceso de análisis y desarrollo.
5. Dominar el uso de un lenguaje de programación oficial, complementado por otros lenguajes relevantes para ampliar las capacidades de desarrollo y solución de problemas.

**Metodología:**

- Aulas virtuales (días, ver inicio).
- Elaboración de ejercicios.
- Exámenes cortos.
- Elaboración de proyectos de programación.
- Cursos complementarios extra aula.
- Conferencia complementaria.

**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 (ejercicios, exámenes cortos 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 y puntualidad.

**Aislamiento de rendimiento académico**

Descripción	Publicación	Entrega	Punteo
Práctica 1 (Consola)	07/02/2025	14/02/2025 (Prorroga posible: 16/02/2025)	5
Práctica 2 (Hilos/SWIG)	07/03/2025	21/03/2025(Prorroga posible: 23/03/2025)	5
<b>Total de prácticas:</b>			<b>10</b>

Proyecto 1 (SWIG)	17/02/2025	07/03/2025(Prorroga posible: 09/03/2025)	30
Proyecto 2 (SWIG/Hilos)	21/03/2025	19/04/2025(Prorroga posible: 21/04/2025)	40
<b>Total de proyectos:</b>			<b>70</b>
Corto 1	—	25-26-27/02/2025	5
Corto 2	—	25-26-27/03/2025	5
<b>Total de Cortos:</b>			<b>10</b>
<b>Examen Final (06-07-08/05/2025)</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>

Práctica (Python)	Extra	21/04/2025	28/04/2025	5
<b>Total de prácticas:</b>				<b>5</b>

<b>Contenido</b>	
<b>1. Fundamentos de Programación</b>	<ul style="list-style-type: none"> <li>1.1. Algoritmos</li> <li>1.2. Diagramas de Flujo</li> <li>1.3. Pseudocódigo</li> </ul>
<b>2. Versionamiento</b>	<ul style="list-style-type: none"> <li>2.1. Introducción a versionamiento</li> <li>2.2. Herramientas de control de versiones <ul style="list-style-type: none"> <li>2.2.1. Git</li> <li>2.2.2. Github</li> <li>2.2.3. Github Desktop</li> </ul> </li> <li>2.3. Tareas básicas <ul style="list-style-type: none"> <li>2.3.1. Creación de repositorio</li> <li>2.3.2. Commit</li> <li>2.3.3. Release</li> <li>2.3.4. Push</li> <li>2.3.5. Pull</li> </ul> </li> <li>2.4. Colaboración en repositorios remotos <ul style="list-style-type: none"> <li>2.4.1. Clonación de repositorio</li> <li>2.4.2. Resolución de conflictos</li> </ul> </li> </ul>
<b>3. Introducción a JAVA</b>	<ul style="list-style-type: none"> <li>3.1. ¿Qué es Java?</li> <li>3.2. Versiones y Ambiente de Java <ul style="list-style-type: none"> <li>3.2.1. JDK</li> <li>3.2.2. JRE</li> <li>3.2.3. JVM</li> </ul> </li> <li>3.3. Características de Java</li> <li>3.4. Comentarios de una línea y multilínea</li> </ul>

- 3.5. Variables
- 3.6. Tipos Primitivos y No Primitivos
- 3.7. Casteos Implícitos y Explícitos
- 3.8. Operadores Aritméticos, Relacionales y Lógicos
- 3.9. Input y output
- 3.10. Estructuras de Control
  - 3.10.1. if, else if, else
  - 3.10.2. switch
- 3.11. Ciclos
  - 3.11.1. for
  - 3.11.2. while
  - 3.11.3. do - while
- 3.12. Try-Catch-Finally
- 3.13. Procedimientos y Funciones
- 3.14. Debugging
  - 3.14.1. Breakpoint
    - 3.14.1.1. Inline
    - 3.14.1.2. Function
      - 3.14.1.2.1. Entrar a función
      - 3.14.1.2.2. Salir de función
  - 3.14.2. Start
  - 3.14.3. Pause
  - 3.14.4. Continue
  - 3.14.5. Stop

#### **4. Manejo de memoria y archivos de texto plano**

- 4.1. Arreglos
- 4.2. Matrices
- 4.3. Listas Dinámicas
- 4.4. Metodos de Ordenamiento:
  - 4.4.1. Burbuja
  - 4.4.2. Por inserción
  - 4.4.3. Por Selección
  - 4.4.4. Quick Sort
- 4.5. Recursividad
  - 4.5.1. Estrategia "divide y vencerás"
  - 4.5.2. Pila de ejecución
  - 4.5.3. Recursividad Simple
  - 4.5.4. Recursividad de Cola
- 4.6. Archivos de texto plano
  - 4.6.1. Creación
  - 4.6.2. Lectura
  - 4.6.3. Escritura
  - 4.6.4. Eliminación

#### **5. Programación Orientada a Objetos (POO)**

- 5.1. Concepto de abstracción y clasificación
- 5.2. Clases y objetos
- 5.3. Mensajes y métodos
- 5.4. Pilares de POO
  - 5.4.1. Abstracción
  - 5.4.2. Encapsulamiento
  - 5.4.3. Polimorfismo
  - 5.4.4. Herencia
- 5.5. Los miembros de una clase
  - 5.5.1. Atributos
  - 5.5.2. Métodos (operaciones)
  - 5.5.3. Constructores y Destruyores
- 5.6. Modificadores de visibilidad

5.6.1. Privado
5.6.2. Público
5.6.3. Protegido
5.7. Construcciones abstractas
5.7.1. Clase abstracta
5.7.2. Interface
5.8. Conceptos avanzados
5.8.1. Miembros estáticos (static) y miembros de instancia
5.8.2. Referencia "this"
5.9. Serialización de objetos en archivos
<b>6. Interfaces Gráficas en JAVA</b>
6.1. Librerías de interfaz gráfica AWT y SWING
6.2. Componentes de interfaz gráfica
6.3. Disparadores de Eventos
6.4. Drag and Drop
<b>7. Diagrama de Clases UML</b>
7.1. Definición de clases y sus relaciones
7.1.1. Asociación
7.1.2. Agregación y composición
7.2. Ámbito de las propiedades, Métodos
7.3. Diseño de programas
7.4. Asociaciones y restricciones, clases de asociaciones, multiplicidad, dependencia.
7.5. Relaciones múltiples (asociativas) y reflexivas
<b>8. Patrones de Diseño</b>
8.1 MVC
8.1.1 Definición de MVC y sus componentes
8.1.2 Modelo
8.1.2 Vista
8.1.3 Controlador
<b>9. Hilos</b>
9.1 Definición de Hilos
9.2 Ciclo de Vida de un Hilo
9.3 Hilos y sus prioridades
9.4 Creación de Hilos
<b>10. Cloud Computing</b>
10.1. Ventajas y Desventajas
10.2. Tipos de nube
10.2.1. Pública
10.2.2. Privada
10.2.3. Híbrida
10.3. Modelos en la nube
10.3.1. SaaS
10.3.2. PaaS
10.3.3. IaaS
10.4. Servicios en la nube
10.5. Proveedores de nube

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:

- Calificación de prácticas y proyectos serán presenciales de acuerdo a la fecha y hora establecidas por el tutor académico.
- 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.
- 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
- 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 Francisco Orellana López	Anthony Alexander Aquino Santiago
B	William Estuardo Escobar Argueta	Sebastian Alejandro Velasquez Bonilla
C	Moises Eduardo Velasquez Oliva	Juan Francisco Urbina Silva
D	Herman Igor Veliz Linares	Lesther Kevin Federico López Miculax
E	Neftali de Jesus Calderon Mendez	Douglas Alexander Soch Catalán
F	William Estuardo Escobar Argueta	Zenaida Irazema Chacón García
G	Edgar Francisco Rodas Robledo	Max Rodrigo Durán Canteo