TEC | Tecnológico de Costa Rica

Programa del curso IC-1400

Elementos de Computación

Escuela de Computación

TEC | Tecnológico de Costa Rica

I parte: Aspectos relativos al plan de estudios

1 Datos generale

Nombre del curso:	Elementos de Computación
Código:	CA-2125
Tipo de curso:	Teórico – práctico
Nº de créditos:	3
Nº horas de clase por semana:	4
Nº horas extraclase por semana:	6
Ubicación en el plan de estudios:	
Requisitos:	Ninguno.
Correquisitos:	Ninguno
El curso es requisito de:	
Asistencia:	Obligatoria
Suficiencia:	No
Posibilidad de reconocimiento:	Sí.
Vigencia del programa:	II semestre 2024.

TEC | Tecnológico de Costa Rica

2 Descripción general

En este curso se desarrollan habilidades de análisis, diseño y programación de problemas básicos en ingeniería. Se utilizará el paradigma de orientación a objetos, diagramas (de flujo o actividad) y un lenguaje de alto nivel con el fin de llevar a la práctica los conocimientos obtenidos en clase.

3 Objetivos

Objetivo General

Al finalizar el curso el estudiante podrá analizar, diseñar y resolver ejercicios de problemas básicos de ingeniería mediante el uso del computador, utilizando el paradigma de orientación a objetos, alguna herramienta de diagramas y un lenguaje de programación de alto nivel.

4 Contenidos

I. Organización de un sistema computacional (1 semana)

i. Antecedentes históricos

ii. Arquitectura básica de un computador: memoria, unidad central de procesamiento, almacenamiento secundario.

iii. Componentes de software: Software del sistema, Programas de uso general, Desarrollo de aplicaciones

II. Resolución de problemas (2 semanas)

i. Definición del problema

ii. Entradas, salidas, proceso y limitaciones

iii. Algoritmos (diagrama flujo - seudocódigo)

iv. Verificación y análisis del algoritmo

v. Implementación del algoritmo

vi. Corrida ejemplo

vii. Validación del programa

III. Elementos básicos del lenguaje de programación (2 semanas)

i. Expresiones y sentencias

ii. Variables

iii. Tipos de datos

iv. Operadores

v. Comentarios

vi. Indentación

IV. Funciones (1 semana)

i. Definición de funciones

ii. Paso de parámetros

iii. Valores por omisión en los argumentos

iv. Funciones e

V. Herramientas de control de flujo (3 semanas)

i. Construcciones de selección (if, etc.)

ii. Construcciones de Iteración (for, while, etc.)

VI. Estructuras de datos compuestas (2 semanas)

i. Strings, Listas y Tuplas

ii. Diccionarios

VII. Clases y objetos (2 semanas)

i. Introducción a las clases

ii. Sintaxis de definición de clases

iii. Creación de objetos

iv. Clases y métodos

v. Herencia

VIII. Entrada y salida

vi. 1. Lectura y escritura de archivos

vii. 2. Métodos de los objetos archivo

IX. Manejo excepciones (1 semana)

viii. Estatutos Try y Except

ix. Clases de excepciones

II parte: Aspectos operativos

5 Metodología de enseñanza y aprendizaje

La metodología que se usará en el curso consistirá en clases de parte de el profesor y ejercicios de programación en laboratorio. Además, todas las semanas se desarrollarán ejercicios cortos de proyectos de

programación, los cuales consolidarán el aprendizaje de los temas vistos en clase

6 Evaluación

Se aplicarán se asignarán diferentes laboratorios con un valor total de 50% (aproximadamente 5); se realizarán tareas, con un valor de 10%, prácticas y trabajos en clase con un valor de un 10%, a lo largo del curso y 1 proyecto de 30%.

Aspecto	Valor
Laboratorios prácticos (5)	50%
Proyecto I (Asignación Semana 11 – Entrega Semana 16)	30%
Quices,Tareas,trabajo en clase	20%
TOTAL	**100%**

7 Bibliografía

- [Solano, 2012] Solano, Jaime. Introducción a la programación con Python. Editorial Tecnológica, TEC de Costa Rica. 2012.
- [Becerra, 1992] Becerra, César. Algoritmos: Conceptos básicos. 1 Edición. Colombia. Editorial Kimpres Ltda. 1992.
- [Deitel & Deitel, 2002] Deitel, Harvey, Deitel, Paul, Liperi, Jonathan. Python: How to Program. 1 Edición. New Jersey, USA: Prentice-Hall, 2002.
- [Downey, 2008] Downey, Allen. Think Python: an introduction to Software Design. 3 Edición. Massachusetts, USA: Green Tea Press., 2008.

- [Downey & Elkner & Meyers, 2002] Downey, Allen, Elkner, Jeffrey, Meyers, Chris. How to think like a Computer Scientist: Learning with Python. 1 Edición. Massachusetts, USA: Prentice-Hall, 2002.
- [González,] González Raúl. Python para todos. Creative Commons Reconocimiento 2.5, España
- [Joyanes, 1987] Joyanes, Luis. Metodología de la Programación: Diagramas de flujo, algoritmos y programación estructurada. México: McGraw-Hill., 1987
- [Hetland, 2005] Hetland, Magnus. Beginning Python: From novice to professional. 1 Edición. New York, USA: Springer-Verlag, 2005.
- [Lutz, 2008] Lutz, Mark. Learning Python: Powerful Object Oriented Programming. 3 Edition. California, USA: O'Reilly Media Inc, 2008.
- [Muñoz et al., 2002] Muñoz, Camelia, Niño, Alfonso, Vizcaíno, Aurora. Introducción a la programación con Orientación a Objetos. Madrid, España, Prentice-Hall, 2002.

8 Profesor

Jonathan Solis Parajeles
Ing. en Computación

Correo: josolis@itcr.ac.cr

Whatsapp: 8724-5688

Horario de consulta:

- Miércoles de 7:30 am a 11:00 am
- Viernes de 2:30 pm a 4:00 pm

TEC | Tecnológico de Costa Rica

Semana	Objetivo	Contenidos	Actividades de aprendizaje	Recursos necesarios	Actividades Extraclase	Evaluación y Medición
1	Explicar la organización de un sistema computacional para la compresión de su funcionamiento	**Unidad I. Organización de un sistema computacional** 1. Antecedentes históricos 2. Arquitectura básica de un computador: memoria, unidad central de procesamiento, almacenamiento secundario. 3. Componentes de software: Software del sistema, Programas de uso general, Desarrollo de aplicaciones	▪ El profesor y los estudiantes se presentan ▪ El profesor realiza la presentación del programa del curso (Objetivos, Contenidos, Evaluación y Cronograma) ▪ El profesor reúne a los estudiantes en grupos les asigna la lectura de un artículo para complementar los temas de la unidad ▪ Se revisan conceptos tanto con material de presentación como búsquedas propias en Internet. ▪ Observan videos sobre la temática de la clase	▪ Presentación ▪ Preguntas en mentimeter ▪ Cuestionario en Kahoot ▪ Videos		Diagnostica *Responde preguntas básicas sobre la organización de un sistema computacional para determinar su nivel de conocimiento sobre el tema Formativa *Responde preguntas intermedias sobre los conceptos vistos en clase Sumativa: **Actividad en clase 1:** *Elabora una **infografía** sobre una temática del tema de Organización de un sistema computacional *Presenta los resultados de la infografía
2	Discutir los fundamentos de la resolución de problemas utilizando para su interpretación la técnica de algoritmos	**Unidad II. Resolución de problemas** 1. Definición del problema 2. Entradas, salidas, proceso y limitaciones 3. Algoritmos (diagrama flujo - seudocódigo) 4. Verificación y análisis del algoritmo	▪ El profesor presenta los conceptos de problema y algoritmo ▪ En parejas, los estudiantes realizan actividades con el objetivo desarrollar la lógica de problemas.	▪ Presentación ▪ Listado de ejercicios ▪ Herramienta para DFD	▪ Repasar material de clase ▪ Resolver ejercicios adicionales sobre algoritmos en seudocódigo y diagramas de flujo	Formativa *Resuelve ejercicios de forma correcta sobre diagramas de flujo

			▪ El estudiante escribe los pasos necesarios para elaborar un producto con el objetivo de comprender el concepto de algoritmo. Posteriormente, El profesor selecciona estudiantes al azar para que retroalimentar sus resultados. ▪ El estudiante elabora algoritmos tanto en seudocódigo como en su equivalente diagrama de flujo para una serie de ejercicios prácticos ▪ El profesor revisa en clase algunos de los ejercicios, para retroalimentar a los estudiantes sobre la elaboración de algoritmos tanto en seudocódigo como en su equivalente diagrama de flujo.			
3	Discutir los fundamentos de la resolución de problemas utilizando para su	**Unidad II. Resolución de problemas** 1. Definición del problema 2. Entradas, salidas, proceso y limitaciones 3. Algoritmos (diagrama	▪ El profesor expone los conceptos relacionados a los ciclos usando DFD ▪ El profesor resuelve ejercicios en DFD sobre ciclos	▪ Presentación ▪ Listado de ejercicios	▪ Repasar material de clase ▪ Resolver ejercicios adicionales sobre algoritmos en	Formativa *Comprende el concepto de algoritmo mediante la resolución de ejercicios

	interpretación la técnica de algoritmos	flujo - seudocódigo) 4. Verificación y análisis del algoritmo	▪ Los estudiantes traducen de seudocódigo a DFD ▪ En grupos, los estudiantes resuelven ejercicios en DFD ▪ El profesor revisa en clase algunos de los ejercicios, para retroalimentar a los estudiantes sobre la elaboración de algoritmos tanto en seudocódigo como en su equivalente diagrama de flujo.		seudocódigo y diagramas de flujo	
4	Discutir los fundamentos de la resolución de problemas utilizando para su interpretación la técnica de algoritmos	**Unidad II. Resolución de problemas** 1. Implementación del algoritmo 2. Corrida ejemplo 3. Validación del programa	▪ El profesor divide los estudiantes en grupos para que desarrollen análisis de casos utilizando diagramas de flujo. Se comentan los resultados obtenidos. ▪ Los estudiantes intercambian las soluciones a los ejercicicios para su revisión. ▪ Al cierre de la clase, al menos dos grupos presentan los resultados obtenidos para el análisis y retroalimentación	▪ Presentación ▪ Listado de casos ▪ Guía de la tarea	▪ Bajar la guía de la tarea que se encuentra en el TEC-digital. ▪ Resolver los ejercicios propuestos ▪ Enviar la tarea antes de la hora estipulada	Formativa *Analiza casos de complejidad baja utilizando adecuadamente los diagramas de flujo Sumativa *Resuelve la **tarea 1** sobre ejercicios de DFD

5	Discutir los fundamentos de la resolución de problemas utilizando para su interpretación la técnica de algoritmos	**Unidad III. Elementos básicos del lenguaje de programación** 1. Expresiones y sentencias 2. Variables 3. Tipos de datos 4. Operadores 5. Comentarios 6. Indentación	▪ El profesor presenta las características básicas del lenguaje Python ▪ El profesor resuelve ejercicios básicos en Python ▪ El profesor convierte ejercicios en DFD a Python ▪ Los estudiantes resuelven ejercicios en Python	▪ Presentación ▪ Listado de casos ▪ Guía del laboratorio	▪ Repasar material de clase ▪ Resolver ejercicios adicionales	Formativa *Analiza casos de complejidad media utilizando adecuadamente los diagramas de flujo Sumativa *Resuelve el **laboratorio 1** sobre seudocódigo y diagramas de flujo
6	Describir los elementos básicos del lenguaje de programación de alto nivel en la solución de problemas básicos	**Unidad V. Herramientas de control de flujo** 1. Construcciones *de selección* (if, etc.) 2. Construcciones de Iteración (for, while, etc.)	▪ El profesor presenta los conceptos básicos de las herramientas de control de flujo de selección e iterativas ▪ El profesor muestra ejemplos de diagramas de flujo sobre herramientas de control de flujo de selección e iterativas ▪ El estudiante desarrolla en Python ejercicios en donde se utilicen herramientas de control de flujo de selección e iterativas ▪ El profesor revisa en clase algunos de los ejercicios, para retroalimentar a los estudiantes sobre la	▪ Presentación ▪ Listado de ejercicios ▪ Guía de la tarea	▪ Bajar la guía de la tarea que se encuentra en el TEC-digital. ▪ Resolver los ejercicios propuestos ▪ Enviar la tarea antes de la hora estipulada	Formativa *Reconoce los conceptos de sentencia, variable, tipo de dato, operadores e indentación en Python *Escribe código de forma guiada en el lenguaje de programación Python Sumativa *Resuelve el la **tarea 2** sobre elementos básicos del lenguaje de programación y operadores matemáticos y lógicos, cadenas y

			correcta utilización de las herramientas de control de flujo de selección.			expresiones booleanas
7	Describir los elementos básicos del lenguaje de programación de alto nivel en la solución de problemas básicos	**Unidad V. Herramientas de control de flujo** 1. Construcciones *de selección* (if, etc.) 2. Construcciones de Iteración (for, while, etc.)	▪ El profesor divide los estudiantes en grupos para que desarrollen análisis de casos utilizando python. Se comentan los resultados obtenidos. ▪ Los estudiantes intercambian las soluciones a los ejercicicios para su revisión. ▪ Al cierre de la clase, al menos dos grupos presentan los resultados obtenidos para el análisis y retroalimentación	▪ Presentación ▪ Listado de ejercicios ▪ Guía del laboratorio 2	▪ Repasar material de clase ▪ Resolver ejercicios adicionales	Formativa *Escribe código en el lenguaje de programación Python sobre estructuras de control de selección e iterativas Sumativa *Resuelve el **laboratorio 2** sobre elementos básicos del lenguaje de programación y operadores matemáticos y lógicos, cadenas y expresiones booleanas
8	Describir los elementos básicos del lenguaje de programación de alto nivel en la solución de problemas básicos	**Unidad VI. Estructuras de datos compuestas** 1. Strings y Listas	▪ El profesor presenta los conceptos básicos de estructuras de datos compuestas ▪ El profesor ejemplifica la utilización de las funciones propias de Python sobre Strings y listas ▪ El estudiante desarrolla en Python una serie de	▪ Presentación ▪ Listado de ejercicios ▪ Guía de la tarea	▪ Repasar material de clase ▪ Bajar la guía de la tarea que se encuentra en el TEC-digital. ▪ Resolver los ejercicios propuestos	Formativa *Escribe código en el lenguaje de programación Python sobre estructuras de control de selección e iterativas Sumativa *Resuelve **la tarea 3**

			ejercicios en donde utilice las listas ▪ El profesor revisa en clase algunos de los ejercicios, para retroalimentar a los estudiantes sobre la correcta utilización de las listas		▪ Enviar la tarea antes de la hora estipulada	sobre estructuras de control de selección e iterativas
9	Describir los elementos básicos del lenguaje de programación de alto nivel en la solución de problemas básicos	**Unidad VI. Estructuras de datos compuestas** 1. Strings y Listas	▪ El profesor realiza un repaso de los conceptos de la clase anterior ▪ El profesor expone sobre funciones definidas por el usuario ▪ El estudiante resuelve ejercicios sobre funciones definidas por el usuario	▪ Presentación ▪ Listado de ejercicios ▪ Guía del laboratorio	▪ Repasar material de clase ▪ Resolver ejercicios adicionales	Formativa *Escribe código en el lenguaje de programación Python sobre estructuras de control de selección e iterativas Sumativa: *Resuelve el **laboratorio 3** en Python sobre estructuras de control de selección e iterativas
10	Describir los elementos básicos del lenguaje de programación de alto nivel en la solución de problemas básicos	**Unidad VII. Clases y objetos** 1. Introducción a las clases 2. Sintaxis de definición de clases 3. Creación de objetos	▪ El profesor expone los conceptos básicos sobre clases y objetos ▪ El estudiante define atributos y métodos para un objeto vehículo ▪ En grupos, los estudiantes analizan diferentes ejemplos de	▪ Presentación ▪ Lista de ejercicios ▪ Guía del cuadro SQA	▪ Repasar material de clase ▪ Resolver ejercicios adicionales sobre clases ▪ Finalizar la elaboración	Formativa *Escucha con atención la explicación de el profesor. *Escribe código en el lenguaje de programación Python sobre listas

			representación de objetos y definen sus atributos y métodos ▪ El profesor implementa una clase en Python ▪ El estudiante define una clase de un objeto X, la implementa en python y crea tres instancias ▪ El profesor muestra la solución correcta de la clase de un objeto X, la implementación en python y las instancias ▪ El estudiante realiza un cuadro SQA sobre la temática de la clase		del cuadro SQA	Sumativa **Actividad en clase 2:** *Elabora un cuadro SQA sobre el tema de clases y objetivos desarrollado en la clase
11	Definir el paradigma de la orientación a objetos mediante el análisis, diseño y codificación de programas de cómputo para la resolución de problemas	**Unidad VII. Clases y objetos** 1. Introducción a las clases 2. Sintaxis de definición de clases 3. Creación de objetos	▪ El profesor orienta a los estudiantes sobre el proyecto I del curso ▪ Los estudiantes identifincan los objetos a partir de un problema determinado y especifican las clases ▪ El profesor define en Python una serie de clases	▪ Presentación ▪ Guía del proyecto	▪ Bajar la guía del proyecto que se encuentra en el TEC-digital. ▪ Resolver los ejercicios propuestos	Formativa *Comprende los conceptos de clase y objeto *Desarrolla adecuadamente una clase en Python Sumativa *Desarrolla el proyecto

12		**Unidad VII. Clases y objetos** 1. Introducción a las clases 2. Sintaxis de definición de clases 3. Creación de objetos	▪ El profesor expone las relaciones que se pueden definir entre las clases ▪ En grupos de 2 personas, redactan un problema que tenga mínimo 3 clases. Luego, elaboran el diagrama de clases que permita resolver dicho problema. ▪ El profesor expone como se implementan en Python las relaciones entre clases	▪ Presentación	▪ Resolver los ejercicios propuestos ▪ Trabaja en el proyecto ▪ Finaliza la elaboración del diagrama de clases	Formativa *Implementa adecuadamente las relaciones entre clases en Python Sumativa: **Actividad en clase 3:** *Elabora de manera correcta un diagrama de clases *Desarrolla el proyecto
13	Definir el paradigma de la orientación a objetos mediante el análisis, diseño y codificación de programas de cómputo para la resolución de problemas	**Unidad VII. Clases y objetos** 1. Introducción a las clases 2. Sintaxis de definición de clases 3. Creación de objetos	▪ El estudiante desarrolla en Python ejercicios en los que combina las clases con las listas ▪ El profesor revisa en clase algunos de los ejercicios, para retroalimentar a los estudiantes sobre la correcta utilización de las clases y objetos con las listas ▪ Al finalizar la clase, los estudiantes graban un video de máximo de 5 minutos donde explican lo más importante que han	▪ Presentación ▪ Listado de ejercicios ▪ Guía de la tarea	▪ Repasar material de clase ▪ Trabaja en el proyecto ▪ Finaliza la grabación del video	Formativa *Escribe código en el lenguaje de programación Python en el que combina clases con listas

			aprendido durante la clase.			
14	Definir el paradigma de la orientación a objetos mediante el análisis, diseño y codificación de programas de cómputo para la resolución de problemas	**Unidad VII. Clases y objetos** 1. Clases y métodos	▪ Los estudiantes desarrollan el Laboratorio ▪ El profesor expone sobre la herencia múltiple en Python ▪ Los estudiantes resuelven ejercicios sobre herencia múltiple	▪ Presentación ▪ Listado de ejercicios ▪ Guía del laboratorio	▪ Repasar material de clase ▪ Resuelve ejercicios adicionales ▪ Trabaja en el proyecto	Formativa *Comprende los pilares de la programación *Relaciona adecuadamente las clases Sumativa *Resuelve el **laboratorio 4** sobre listas *Desarrolla el proyecto
15	Aplicar el paradigma de la orientación objetos en la resolución de problemas utilizando de técnicas eficaces de análisis, diseño y codificación de programas de cómputo en un lenguaje de programación de alto nivel	**Unidad II. Estructuras de datos compuestas** 1. Strings, Listas y Tuplas 2. Diccionarios	▪ El profesor expone sobre Tuplas y Diccionarios ▪ Los estudiantes resuelven ejercicios sobre Tuplas y Diccionarios	▪ Presentación ▪ Listado de ejercicios	▪ Trabaja en el proyecto	Formativa *Resuelve ejercicios sobre Tuplas y Diccionarios Sumativa *Desarrolla el proyecto

16	Aplicar el paradigma de la orientación objetos en la resolución de problemas utilizando de técnicas eficaces de análisis, diseño y codificación de programas de cómputo en un lenguaje de programación de alto nivel		▪ Los estudiantes resuelven el laboratorio 5	▪ Guía del laboratorio	▪ Envía el proyecto finalizado	Sumativa *Desarrolla el proyecto *Resuelve el **laboratorio 5** sobre clases