

CI-0202 PRINCIPIOS DE INFORMÁTICA

Profesor: Mag. Melvin Araya González

Créditos: 4

I Ciclo 2024

I Generalidades

Horario: miércoles de 08:00 a 11:50.

Horario de consulta: jueves de 08:00 a 10:00

Correo: melvin.arayagonzalez@ucr.ac.cr

Modalidad del entorno virtual: Virtual para las clases sincrónicas, lecturas propuestas, actividades propuestas (individuales o grupales) de desarrollo o ejecución, entrega de material didáctico.

Las clases cuentan con modalidad virtual (sincrónicas o asincrónicas), por lo que al ingresar usted como usuario puede estar siendo grabado, lo mencionado se realiza con fines educativos, en aras de generar material audiovisual para estudiar y de repaso para los estudiantes inscritos de este curso.

El realizar grabaciones de las sesiones o clases del curso queda a criterio del docente a cargo.

II Descripción del curso

El curso de Principios de Informática es un curso básico de programación para estudiantes el área de ingeniería y afines.

En el curso se introduce al estudiante al concepto del pensamiento abstracto en la resolución de situaciones informáticas, mediante el uso ordenado y adecuado de resolución y construcción de problemas.

El estudiante aprende a diseñar, construir y reconocer la utilización de las estructuras de datos básicas de la actividad informática para lograr el diseño e implementación de algoritmos.

III Objetivo General

Proveer formación básica en programación utilizando técnicas actuales para lograr la resolución de problemas mediante la construcción de programas orientados al área de ingeniería y afines.

IV Objetivos Específicos

Al finalizar este curso el o la estudiante será capaz de:

- Diseñar, organizar e implementar algoritmos para resolver problemas específicos
- Usar un ambiente de programación para la edición, prueba y depuración de programas
- Reutilizar componentes de software existentes en una plataforma abierta
- Aplicar a nivel básico buenas prácticas de construcción de software

V Contenido del curso

Tema 1: Fundamentos de la Programación

- Lenguajes de programación: concepto de programación, lenguaje máquina, lenguaje ensamblador, lenguaje de alto nivel, máquina virtual, compilador y paradigmas.
- Ciclo de vida de un programa: problema, análisis, diseño, implementación y prueba
- Algoritmo: concepto, primitivas y ejemplos

Tema 2: Introducción a la Programación Orientada a Objetos

- Paradigma: clases e instancias, atributos y métodos, abstracción y reutilización
- Análisis y diseño: modelaje de clases e instancias (UML)
- Compilación y ejecución en lenguaje Java

Tema 3: Tipos de Datos

- Tipos de datos: primitivos (enteros, reales, booleano y caracteres) e hileras
- Precisión y literales: entero (*byte*, *short*, *int* y *long*), real (*float* y *double*), *booleano*, *carácter* e *hilera* (secuencias de escape)

Tema 4:

Definición de Variables

- Declaración: tipo, identificador y dirección
- Inicialización: tipo primitivo (valor), instancia (referencia) y estado de memoria

Utilización de Variables

- Atributos de clase: declaración, ciclo de vida y ocultamiento (encapsulamiento)
- Variables locales: declaración y ciclo de vida
- Estáticas y constantes: declaración y ciclo de vida

Tema 5:

Entrada y Salida Básica

- Entrada: parámetros de línea de comandos y diálogo
- Salida: línea de comandos y diálogo

Manejo de Excepciones

- Concepto, ejemplos y definición (clase), lanzamiento y atrape

Tema 6: Expresiones y Operadores

- Aritméticos binarios (multiplicativos y aditivos) y unarios (negación y posfijos)
- Relacionales (comparación e igualdad), lógicos (binarios y unarios) y asignación
- Evaluación y prioridad (orden de precedencia)

Tema 7: Instrucciones y Estructuras de Control

- Estructuras de secuenciación (*{}*)
- Estructuras de selección o bifurcación (*if/else* y *switch*)
- Estructuras de repetición o iteración (*while*, *do* y *for*)

Tema 8:

Métodos: Fundamentos

- Conceptos: modularización y reutilización, declaración e invocación
- Componentes: encabezado (identificador, parámetros y tipo de retorno) y cuerpo
- Métodos estáticos (funciones) y ejemplos
- Sobrecarga: declaración, firmas y resolución de llamados

Métodos: Funcionamiento

- Paso de argumentos: por valor y por referencia
- Estado de memoria estática, memoria dinámica y pila de llamados
- Reglas de alcance o ámbito en identificadores

Constructores

- Concepto y utilización: declaración e invocación
 - Orden de llamados

Tema 9: Arreglos o Vectores: Fundamentos

- Concepto, estructura y estado de memoria
- Tipos de Datos: primitivos e instancias
- Declaración e inicialización
- Acceso a celdas y recorrido
- Parámetros de tipo arreglo y paso de argumentos

Tema 10: Matrices

- Concepto, estructura y estado de memoria
- Declaración e inicialización
- Acceso a celdas y recorrido

VI Metodología

El curso se base en clases de 1 lección virtual impartidas por el profesor (teoría y práctica)

- Según lo establecido en las resoluciones VD-R-8458-2009 y VD-R-9374-2016, se utilizará un entorno virtual de aprendizaje específico para el curso, en la plataforma oficial institucional Mediación Virtual (<https://mv1.mediacionvirtual.ucr.ac.cr>). El entorno será utilizado para la entrega del programa del curso, material, enunciados de evaluaciones, entre otros, por parte del profesor, así como para entrega y/o realización de evaluaciones y otros entregables por parte del estudiantado cuando corresponda. Según lo establecido en la resolución R-2664-2012, que establece el correo institucional con el dominio @ucr.ac.cr como la herramienta oficial para las comunicaciones de toda la comunidad universitaria, se utilizará el correo institucional como medio oficial de comunicación entre profesores y estudiantes, por lo cual el estudiantado debe tenerlo activo y revisarlo continuamente.
- En este curso es muy importante la participación y la práctica por parte del estudiante, para lograr construir una visión clara de los conceptos de programación con el propósito de que pueda ser empleado en su campo de acción profesional
- En las lecciones magistrales se utiliza los recursos audiovisuales y Zoom, para mostrar los conceptos, usos e implementaciones de las diferentes estructuras de datos en programación
- El IDE de programación que se utilizará será DrJava. El uso de este será acompañado con Jeliot y BlueJ. El estudiante debe buscar una computadora que pueda utilizar fuera de horas de clases para hacer las asignaciones y prácticas. En ella, el estudiante es libre de escoger cualquier herramienta que desee programar
- La nota del curso se divide en:

Exámenes: El estudiante debe resolver un conjunto de ejercicios de programación en un tiempo delimitado, generalmente 2 horas. En cada uno de ellos, se evaluarán los conceptos vistos hasta el momento. La materia será acumulativa centrada en los temas que el profesor indique.

Laboratorios: en todas las clases prácticas, los estudiantes deberán resolver una asignación que consistirá en uno o varios problemas de programación. Los mismos deben ser entregados al finalizar el laboratorio en Mediación Virtual.

Proyecto: consistirá en asignación basada en la materia vista o en laboratorios, pero de mayor complejidad. También pueden contener algún componente de investigación, es decir, algún tema que no se haya visto completamente en clase y el estudiante deba investigar cómo resolverlo por sus medios.

Quices: pequeñas evaluaciones de la materia para mantener al estudiante al día y que sirvan de práctica para el examen.

Tareas programadas: Las prácticas programadas tienen una complejidad mucho mayor. Consistirán en dar solución a un problema que abarque todos los temas vistos en el curso hasta el momento. El profesor brindará las pautas para la entrega de las tareas programadas.

VII Reglas del curso

Las siguientes son reglas que se aplicarán en el curso:

- Las tareas cortas y programadas deben ser entregados el día y la hora especificados en el enunciado de cada tarea.
- Solo se reponen evaluaciones en los casos contemplados en el artículo 24 de Reglamento de Régimen Académico Estudiantil de la UCR, utilizando el procedimiento allí descrito.
- Los quices podrán ser realizados sin previo aviso, en cualquier momento de la clase y pueden ser de comprobación de tareas cortas, programadas, laboratorios o sobre materia vista previamente.
- Habrá tareas individuales y grupales. Este criterio debe ser respetado para la entrega.
- Todos los documentos deben ser entregados en **formato de texto libre** (se recomienda el uso de *.odt de OpenOffice que lo pueden descargar de openoffice.org. No se aceptarán documentos en formatos propietarios (por ejemplo, *.doc ó *.docx de Microsoft). O bien lo pueden entregar en un formato solicitado por el/la docente.
- Todo estudiante debe estar debidamente inscrito en Mediación Virtual. Será el medio de difusión oficial y deben estar atentos a los correos con la cuenta institucional que contendrán correcciones o actualizaciones importantes a fechas o asignaciones.
- Cualquier fraude o intento del mismo será evaluada con una nota de “cero” en los casos que corresponda y castigado con el reglamento universitario.

VIII Recursos didácticos

- Equipo audiovisual con presentaciones referentes a cada tema, donde se explica gráficamente los conceptos a desarrollar y se ofrecen los elementos teóricos básicos para el conocimiento de las distintas estructuras y algoritmos.
- Pizarra para explicar en más detalle el funcionamiento de los algoritmos y características de las estructuras. Es un recurso importante para la clarificación de dudas prácticas y presentación de ejercicios prácticos.
- Prácticas de ejercicios y problemas redactados por el profesor o de libros de referencia.
- Laboratorio para conocer los conceptos básicos de una herramienta visual.

IX Evaluación

Descripción	Porcentaje
Examen Parcial 1	25%
Examen Parcial 2	25%
Proyecto	20%
Quices	10%
Tareas Programadas	10%
Laboratorios	10%

TOTAL 100%

Nota: los criterios de evaluación, las guías metodológicas y los puntos a cumplir del proyecto final, serán entregado en un documento aparte de manera detallada, en el medio o entorno oficial.

Consideraciones sobre la evaluación

- Todas las evaluaciones se califican de 1 a 100, nota que luego será ponderada a su respectivo porcentaje. Cada evaluación posee una distribución distinta de varios aspectos. Los valores serán especificados en la presentación de cada uno. Además, se hará uso de rúbricas según corresponda en las evaluaciones
- Las evaluaciones individuales o grupales pueden incluir una comprobación mediada en la plataforma de Zoom con la activación (video, audio) cuando así lo indique el/la docente.
- El proyecto final combina un informe escrito o auto video (voz, video) como sustento final.

- Los trabajos deben ser entregados puntualmente en la fecha y hora indicadas en mediación virtual.
- No se aceptarán trabajos que no se hayan elaborado, revisado y corregido durante el semestre.
- Cuando un estudiante no pueda asistir personalmente a una clase o presentación de proyectos, debe hacer llegar sus trabajos en la fecha y horas establecidas para su entrega y comunicarlo con anticipación al profesor.
- Cuando un(a) estudiante goce de una incapacidad médica o enfrente una situación personal que amerite ser considerada, debe comunicarlo al profesor a la mayor brevedad posible.
- Los trabajos que no sean entregados durante el período lectivo correspondiente reciben una calificación de "0". No obstante, si son presentadas posteriormente son objeto de revisión y crítica, si el estudiante así lo desea.
- La aplicación de talleres o dinámicas durante las clases estarán relacionados a los temas del curso, estos se evaluarán sustituyendo las actividades dirigidas o individuales de la lección, se aplicarán de forma esporádica.

X Bibliografía

Barea Fernández, A. (2015). Análisis de las prestaciones de la placa Micro Python board v.1.0. Universitat Politècnica de Catalunya.

Fernández De Sevilla Vellón, María, and María Jesús Algar Díaz. Introducción Práctica a La Programación Con Python. Alcalá De Henares: Editorial Universidad De Alcalá, 2019. Textos Universitarios. Tecnología. Web.

Guzdial, M., & Ericson, B. (2013). Introducción a la computación y programación con Python. Distrito Federal: Pearson Educación.

Nolasco Valenzuela, J. (2018). Python. Madrid: RA-MA Editorial.

OZ, E. ; SOUSA, K. J. Administración de los sistemas de información (7a. ed.). ed. México D.F: Cengage Learning, 2015. 562 p.

Turkel, W., & Crymble, A. (2017). Introducción a Python e Instalación. Programming Historian En Español, 1(1), Programming Historian en español, 2017-03-15, Vol.1 (1).

Vasconcelos Santillán, J. (2018). Introducción a la computación. Ciudad de México: Grupo Editorial Patria.

Schmidt. (2022). *Arduino: curso completo*. Ra-Ma.

XI Cronograma

N°	Semana	Temas
1	11 al 16 de marzo Semana B	Tema 1: Fundamentos de la Programación
2	18 al 23 de marzo	Tema 2: Introducción a la Programación Algoritmos
3	25 al 30 de marzo	Semana Santa
4	01 al 06 de abril	Tema 2: Introducción a la Programación Algoritmos
5	08 al 13 de abril	Tema 3: Tipos de Datos y I Quiz

6	15 al 20 de abril	Tema 5: Entrada y Salida Básica; Manejo de Excepciones (Se ve junto al tema 6 y 7) I Laboratorio
7	Semana U 22 al 27 de abril	Tema 6: Expresiones y Operadores (se ve junto al tema 7)
8	29 de abril al 04 mayo	Tema 4: Definición de Variables, Utilización de Variables (Se ve junto al tema 7) y II Quiz
9	06 al 11 de mayo	Tema 7: Instrucciones y Estructuras de Control
10	13 al 18 de mayo	I Examen y Tarea programada
11	20 al 25 de mayo 0es3	Tema 8: Métodos: Fundamentos
12	27 de mayo al 01 de junio	Tema 8: Métodos: Fundamentos y II Laboratorio
13	03 al 08 de junio	Tema 9: Arreglos o Vectores: Fundamentos
14	10 al 15 de junio	Tema 10: Matrices
15	17 al 22 de junio	Tema 9: Arreglos o Vectores: Fundamentos
16	24 al 29 de junio	II Examen y entrega de proyecto
17	01 al 06 de julio	Entrega de Notas
18	08 al 13 de julio	Examen de Ampliación