

Planificación de la Asignatura: Informática Básica - TUPED

Fecha: 23/10/2024 13:02

Código: T1628

Carrera: Tecnicatura Universitaria en Procesamiento y Explotación de Datos

Departamento Académico: Informática

Docente a cargo:

Correo del docente a cargo: sin datos

Régimen de Dictado: Cuatrimestral 1º Cuatrimestre

Carga Horaria Semanal: 5 horas semanales

Carga Horaria Total: 70 horas

Contenidos Mínimos:

Interacción con sistemas operativos. Elementos y características de los lenguajes de programación.

Entornos de programación. Tipos de datos, variables y constantes. Algoritmos computacionales. Estructuras de control. Estructuras de datos: arreglos y listas. Funciones. Manipulación de archivos.

Competencias Genéricas:

No corresponde

Competencias Específicas:**Argumentación de aportes marcados en la matriz de competencias:**

Correlativas Regulares para cursar:

No posee

Correlativas Aprobadas para cursar:

No posee

Correlativas Aprobadas para promocionar o rendir el examen final:

No posee

Inserción de la Asignatura en el plan de Estudios:

Dentro de la carrera de Tecnicatura Universitaria en Procesamiento y Explotación de Datos de la Facultad de Ingeniería, de la Universidad Nacional de Entre Ríos, Informática Básica se dicta como asignatura curricular en el primer cuatrimestre del primer año. Forma parte del bloque denominado Formación de Fundamento (FF), y presenta una carga horaria semanal de 5 (cinco) horas reloj, y una carga horaria total de 75 (setenta y cinco) horas reloj. Y en relación a asignaturas posteriores, es correlativa con las asignaturas de primer año “Probabilidad y Estadística”, “Algoritmos y Estructuras de Datos” y “Aspectos Legales del Uso de la Información” y con las del segundo año “Bases de Datos”, “Modelado Estadístico”, “Exploración de Datos Multivariados”, “Espacio Integrador II” e “Inglés I” de la carrera. Todo lo anterior según el Plan de Estudios vigente, Res. “C.S.” 035/19.

Objetivo General:

Que los estudiantes logren:

- Integrar el recurso informático al proceso de formación de fundamento, científica, técnica y tecnológica.
- Comprender y utilizar con criterio los conceptos fundamentales de diseño de algoritmos computacionales y su programación en computadoras digitales.
- Resolver problemas a través de programas de computadora, empleando un lenguaje de programación interpretado de alto nivel.
- Desarrollar el pensamiento lógico y crítico dentro de un contexto de trabajo colaborativo.
- Alcanzar un nivel de pensamiento que le permita extender autodidácticamente sus conocimientos en el área abarcada por la asignatura.

Objetivos Particulares:

Al finalizar el cursado y aprobar la asignatura Informática Básica, el estudiante será capaz de:

- Dominar el concepto de algoritmo computacional y su importancia para resolver problemas de programación utilizando computadoras digitales modernas.
- Diseñar y desarrollar algoritmos computacionales y sus correspondientes programas para resolver distintos problemas de complejidad baja a media.
- Dominar y aplicar los conceptos de programación para el diseño, edición, interpretación, depuración y ejecución de programas.
- Adquirir la destreza necesaria para seleccionar tipos de datos adecuados para organizar apropiadamente soluciones a problemas de baja complejidad.
- Emplear un entorno de desarrollo integrados (IDE, en inglés) utilizando sus principales características de soporte al desarrollo de programas informáticos.
- Reconocer y reutilizar las principales bibliotecas y módulos de código en un lenguaje interpretado de alto nivel.
- Comprender y aplicar criteriosamente las estructuras de datos conocidas como arreglos y listas a la resolución de problemas de programación tomando en cuenta sus principales características.
- Comprender, crear y aplicar estructuras de datos básicas combinadas: registros y listas.
- Comprender y aplicar estructuras de control en un lenguaje de programación de alto nivel.
- Comprender y programar funciones en un lenguaje de programación de alto nivel.
- Diseñar y desarrollar programas de computadora en un lenguaje interpretado de alto nivel para manipular archivos digitales en dispositivos de almacenamiento secundario para creación, acceso,

modificación, movimiento, borrado y verificaciones.

• Aplicar criterios de buenas prácticas de programación.

• Trabajar colaborativamente para resolver problemas de programación.

Programa Analítico:**Unidad 1: Introducción a la Programación**

Concepto de algoritmo y algoritmos computacionales. Programa. Depuración y documentación de programas. Lenguajes de Programación: tipos y características. Paradigmas de programación.

Unidad 2: Introducción al Lenguaje Python

Estructura de un programa en Python y su ejecución por parte del Intérprete. Tokens del lenguaje. Variables. Ámbito de validez de las variables. Flujos de entrada y salida.

Unidad 3: Expresiones y Lógica de Control

Operadores en Python. Jerarquía de los operadores. Construcción de expresiones. Estructuras de control: condicional (if, elif, else) iterativas (while, for). Uso de las declaraciones pass, break, continue. Función range(). Módulo random. Resolución de problemas mediante programas en Python. Edición, prueba y depuración de programas empleando un Intérprete de Python.

Unidad 4: Estructuras de datos

Listas: creación, recorrido, añadir y quitar elementos, métodos integrados. Listas por comprensión. Tuplas. Problemas de programación empleando datos estructurados.

Unidad 5: Funciones y módulos

Diseño e implementación de funciones en Python. Argumentos, argumentos por defecto, argumentos arbitrarios. Funciones integradas en Python. Importación de módulos y paquetes. Modularidad: creación de módulos con funciones. Problemas de programación empleando funciones. Recursividad

Unidad 6: Módulo String

Definición y creación, eliminación, modificación, recorrido de objetos tipo String. Métodos integrados. Operadores y operaciones. Problemas de programación empleando cadenas de caracteres.

Unidad 7: Archivos de texto

Apertura. Lectura. Escritura. Cierre. Funciones integradas vinculadas al manejo de archivos. Archivos de texto estructurados: CSV

Metodología Didáctica:

Las clases se imparten en tres modalidades:

Teoría: será desarrollada en 1 hora y 30 minutos de duración; y a cargo del docente responsable de la cátedra. Serán utilizadas para el desarrollo de: conceptos, criterios de diseño de algoritmos y programas, análisis crítico. Debido al escaso tiempo disponible, las clases serán expositivas, empleando material didáctico adecuado (slides, proyector) para poder promover la comprensión y discusión de conceptos.

Práctica: Se desarrollarán en aula y laboratorio. Se orientará al alumno para diseñar las soluciones y se implementarán en computadora los programas y proyectos correspondientes a la ejercitación propuesta en las guías de trabajos prácticos. Estarán a cargo de los profesores y jefes de trabajos prácticos. En estas clases está prevista la presencia de ayudantes alumnos y/o pasantes para colaborar con el docente.

Duración: 2,5 Hs. por clase.

Coloquios: Se implementarán a cargo del profesor adjunto de la asignatura en 1 hora inicial, previa al desarrollo de la práctica. Tiene por objeto realizar un seguimiento del proceso enseñanza-aprendizaje y subsanar aspectos relacionados con la posible no comprensión de conceptos teóricos y/o de la aplicación de los mismos en la resolución de problemas.

Blended Learning: Los alumnos disponen de un sitio web basado en un software para e-learning o plataforma virtual de aprendizaje (Moodle), donde pueden acceder al material de la cátedra y a varios recursos disponibles que pueden ayudarlo en su proceso de aprendizaje: noticias, foros, sitios de interés, material adicional, videos. Algunas actividades a distancia serán promovidas por los docentes de la asignatura. Desde el sitio podrán comunicarse con los profesores y entre pares. Cada Jefe de Trabajos Prácticos debe habilitar a sus alumnos requiriendo un nombre de usuario, clave y correo electrónico para acceder al sitio.

Consultas: Se coordinarán con los alumnos. Los profesores propondrán días y horarios en la plataforma (Moodle) en base a los requerimientos de los estudiantes.

Formación Práctica:

Durante la teoría y el coloquio se trabajan actividades de formación práctica, pero la instancia principal que profundiza sobre esta actividad es la clase de práctica en los laboratorios donde el alumno desarrollará su capacidad de saber hacer.

Al inicio de cada clase de laboratorio, los docentes a cargo de las mismas retoman los conceptos desarrollados durante las clases de teoría y coloquio mediante la resolución de uno de los problemas planteados en la guía de actividades prácticas del tema. Este ejemplo se analiza y explica detalladamente a fin de repasar los conceptos principales del tema a desarrollar. Luego, el docente planteará los problemas a resolver durante la clase.

Para cada uno de los temas de la asignatura se cuenta con guías, donde se proponen diferentes tipos de actividades, en su mayoría problemas abiertos, que presentan un nivel creciente de complejidad. Se comienza trabajando sobre problemas clásicos y ejercicios simples, con el objetivo de afirmar los conceptos vistos en teoría, para luego avanzar hacia problemas de complejidad creciente, en los cuales se integran, además, los temas ya abordados anteriormente.

Durante la clase, los estudiantes trabajan en grupos de 2 o 3 integrantes para resolver y codificar en lenguaje Python la solución a los problemas planteados. Luego, utilizando la computadora deben ejecutar y depurar el programa realizado. Durante este proceso podrán consultar sus dudas al docente a cargo de la práctica o al docente ayudante, quienes actuarán como orientadores y estimuladores en la tarea de resolución de los problemas. Asimismo, los alumnos socializarán las soluciones planteadas describiendo en el pizarrón el código propuesto y se discutirán posibles alternativas o escenarios de solución, según las diferentes propuestas de solución de cada uno de ellos.

Una vez finalizada la clase el docente propondrá uno o más problemas de la guía para que los alumnos resuelvan en sus hogares por medio del uso de la computadora o proponiendo la solución en papel.

Debido al carácter eminentemente práctico de la asignatura, la asistencia a la misma es obligatoria.

Listado de Actividades de Formación Práctica:

Introducción a Python, variables y operadores

Estructuras de control de flujo

Estructuras de datos

Funciones en Python

Cadenas en Python

Entrada y Salida: archivos de texto

Intensidad de la formación práctica

Detalle de la carga horaria total prevista para cada una de las siguientes actividades:

Actividades prácticas que aportan a las competencias específicas en el Nivel de dominio 1: 0 horas

Actividades prácticas que aportan a las competencias específicas en el Nivel de dominio 2: 0 horas

Actividades prácticas que aportan a las competencias específicas en el Nivel de dominio 3: 0 horas

Horas totales de actividades de formación práctica: 40 horas

Metodología de Evaluación Durante el cursado:**Evaluaciones parciales**

Se realizarán 2 evaluaciones parciales de 2 hs de duración. En la primera se evaluarán los temas hasta la unidad 3 de la asignatura. Al final del cuatrimestre se tomará un examen integrador final de la totalidad de los temas dictados.

Ambas instancias de evaluación se pueden recuperar.

La nota para aprobar las evaluaciones es 6 o superior.

Metodología de Evaluación en Exámenes Finales:**Examen final para alumnos regulares:**

Se realizará una evaluación de 2 o 3 problemas a resolver mediante código en Python y preguntas conceptuales. Tiempo estimado de resolución: 2 Hs. Calificación mínima para la aprobación: 6.

Examen final para alumnos libres:

El alumno en condición de libre debe presentar la resolución de un trabajo práctico. Se brindarán con antelación el enunciado y los archivos para procesar. El día de la mesa, el estudiante deberá presentar un código funcional, que resuelva las actividades planteadas, las cuales se evaluarán mediante un cuestionario. Una vez aprobada esta instancia, pasará a rendir el mismo examen que el alumno regular. Calificación mínima para la aprobación: 6 en ambas instancias (examen escrito y trabajo práctico).

Condiciones de Regularidad :

Condiciones de Regularidad

Para poder regularizar se exige que el alumno apruebe el primer examen parcial y el examen integrador final con nota igual o superior a 60 puntos y tenga un 75% de asistencia a las clases de práctica.

Condiciones de promoción

La asignatura puede ser aprobada (promovida) durante el cursado, evitando la instancia del examen final.

La condición para promover es tener 75% de asistencia en la práctica y nota igual o superior a 75 puntos en el examen integrador final.



Cronograma de parciales durante el primer Cuatrimestre:

Primer Examen Parcial: 15 de Abril de 2024

Segundo Examen Parcial: 10 de Junio de 2024

Recuperatorio 01: 29 de Abril de 2024

Recuperatorio 02: 24 de Junio de 2024

Cronograma de parciales durante el segundo Cuatrimestre:

Bibliografía Principal:

El tutorial de Python. Python Software Foundation. Disponible online en: <https://docs.python.org/es/3/tutorial/>

La Biblioteca Estándar de Python. Python Software Foundation. Disponible online en:

<https://docs.python.org/es/3/library/index.html#library-index>

Referencia del Lenguaje Python. Python Software Foundation. Disponible online en:

<https://docs.python.org/es/3/reference/index.html#reference-index>

Bibliografía Complementaria:

Tutorial de Python: Guía básica de Python en español. Juan José Lozano Gómez. Disponible en:

<https://j2logo.com/python/tutorial/>

Programación en Python - Nivel básico. Repositorio de manuales y recursos del entrenamiento realizado por la empresa Covantec R.L. Disponible en:

<https://entrenamiento-python-basico.readthedocs.io/es/latest/index.html>.

Introducción a la programación con Python. Apuntes del Curso de iniciación a la programación en Python, del módulo Lenguaje de Marcas y Sistemas de Gestión de la Información del ciclo formativo Administración de Sistemas Informáticos en Red (ASIR), IES Abastos de Valencia (España), 2020/2021. Disponible en:

<https://www.mclibre.org/consultar/python/>.

Python Tutorial. Bernd Klein. Disponible en: <https://python-course.eu/python-tutorial/>

Lógica de Programación. Fundación Carlos Slim. Disponible en:

https://aprende.org/pages.php?r=cfc_course&tagID=7929

Equipo de Cátedra:

1 Profesor Titular (DS): a cargo de teoría.

1 Profesor Adjunto (DE): a cargo de 2 coloquios, diseño de actividades en la PVA (Moodle) y seguimiento de las mismas. A cargo del dictado de Informática Básica, asignatura de la carrera de Licenciatura en Bioinformática.

1 JTP (DE): a cargo de 2 comisiones de práctica y seguimiento de actividades en la PVA. A cargo de las actividades prácticas de Informática Básica, asignatura de la carrera de Licenciatura en Bioinformática.

3 JTP (DP): responsables de hasta 2 clases prácticas cada uno y seguimiento de actividades en la PVA.

2 Aux. Alumnos (DS): colaboradores en las clases prácticas.

Todos los docentes están afectados a otras actividades académicas dentro de la FI-UNER, por lo que no disponen de la totalidad de su dedicación en esta asignatura.

Actividades de Investigación Gestión y Extensión:

Requisitos de admisión para alumnos oyentes:

Que exista disponibilidad de lugar físico

Infraestructura, equipamiento y recursos necesarios:

Para desarrollar las actividades de práctica se deberá contar con laboratorios equipados con al menos una computadora cada dos alumnos, e idealmente una para cada alumno. Cada computadora deberá tener instalado y funcionando el software Spyder (o IDE similar que soporte Python).

Las aulas para desarrollar teoría y coloquio deberán contar con proyector y espacio disponible para el total de estudiantes.

Otros: