

Planificación de la Asignatura: Fundamentos de Programación - Bioinformática

Fecha: 23/10/2024 13:02

Código: L1305

Carrera: Licenciatura en Bioinformática

Departamento Académico: Informática

Docente a cargo:

Correo del docente a cargo: analia.cherniz@uner.edu.ar

Régimen de Dictado: Cuatrimestral doble oferta

Carga Horaria Semanal: 5 horas semanales

Carga Horaria Total: 70 horas

Contenidos Mínimos:

Algoritmos computacionales. Programación: diseño de programas para resolver problemas empleando un lenguaje de programación estándar: tipos de datos, constantes y variables, estructuras de control, estructuras de datos básicas (arreglos, registros), funciones, archivos. Introducción a la programación Orientada a Objetos: prueba y depuración de programas.

Competencias Genéricas:

CT 1. Identificación, formulación y resolución de problemas de la disciplina Bioinformática. Nivel de dominio 1.

CT 4. Utilización de técnicas y herramientas de aplicación en la disciplina Bioinformática. Nivel de dominio 1.

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

Como materia de primer año, Fundamentos de Programación aporta a la primera competencia genérica tecnológica: "Identificación, formulación y resolución de problemas" (en particular en la identificación y resolución) mediante el uso de código computacional y con un nivel de dominio inicial, que permite el desarrollo de conocimientos disciplinares, habilidades y destrezas que serán el fundamento para la adquisición de esta competencia y de las competencias específicas.

Debido a que Fundamentos de Programación es parte del descriptor Informática, la asignatura aporta también a la cuarta competencia genérica tecnológica: "Utilización de técnicas y herramientas de aplicación en la disciplina Bioinformática", pues se trata de una asignatura con fuerte impronta en la utilización de recursos computacionales. Para el desarrollo de las actividades prácticas se utilizan lenguajes informáticos y entornos de trabajo usados también en el ámbito profesional y científico. Se trata de una primera aproximación al campo profesional, mediante el desarrollo de actividades basadas en situaciones problemáticas, por lo cual se trabaja en un primer nivel de dominio de la competencia.

Correlativas Regulares para cursar:

No posee

Correlativas Aprobadas para cursar:

No posee

Correlativas Aprobadas para promocionar o rendir el examen final:

No posee

Insercion de la Asignatura en el plan de Estudios:**Justificación**

El objetivo de la carrera es la formación de un graduado que intervenga en problemáticas relacionadas con las ciencias biomédicas aplicando la informática en el análisis, modelado y simulación de las estructuras y fenómenos observados en los seres vivos en los distintos niveles de organización. La actividad del Licenciado en Bioinformática está relacionada, también, a los programas de secuenciación y análisis de genomas de distintas especies, esta especialización cumple un importante papel en el análisis de cualquier tipo de datos biológicos, ecológicos, sistemáticos y bioquímicos con el soporte principal de programas informáticos (software). Dentro de este contexto la programación cumple un rol formador de la capacidad de abstracción y la creatividad del futuro profesional e instruye en la utilización de la herramienta computacional para que el Bioinformático desarrolle sus actividades.

Integración curricular

Fundamentos de Programación junto con Informática Básica, Programación Avanzada y Algoritmos y Estructuras de Datos son las primeras materias que conforman el tronco principal en la formación informática del futuro graduado, ya que aportan los conocimientos teórico-prácticos para el diseño y desarrollo de programas, fomentando la tarea creativa como una actividad metódica y sistemática, exigiendo la definición del problema, modelo de resolución, diseño e implementación de la solución e interpretación de los resultados.

Al aplicar estos conocimientos en la solución de problemas de otras disciplinas, permite la interrelación e integración entre las distintas áreas curriculares. De este modo, los docentes de cada área podrán planificar tareas y trabajos en donde se estimule al alumno a aplicar herramientas de programación para la resolución de problemas.

Objetivo General:

Que el alumno logre:

Integrar el recurso informático a su proceso de formación básica, científica- técnica.

Comprender y utilizar los conceptos de programación de computadoras.

Resolver problemas a través de programas, empleando un lenguaje de programación estándar de alto nivel.

Desarrollar el pensamiento lógico y crítico dentro de un contexto de trabajo colaborativo.

Objetivos Particulares:

Que el alumno logre:

Dominar el concepto de algoritmo y su importancia en la resolución de problemas.

Dominar y aplicar los conceptos de programación: diseño, edición, compilación, depuración y ejecución de programas.

Utilizar con destreza un lenguaje estándar de alto nivel.

Emplear un entorno de programación y conocer y utilizar sus principales comandos.

Adquirir la destreza necesaria para seleccionar estructuras de datos básicas y los tipos de datos adecuados para organizar la información en los programas.

Diseñar algoritmos de solución para distintos problemas y su programa correspondiente.

Aplicar el modelo de la programación modular y estructurada a la resolución de problemas.

Familiarizarse con el hardware, manuales y lenguaje técnico propios de los elementos empleados en el desarrollo de la asignatura.

Conocer las bases conceptuales del modelo de la programación orientada a objetos.

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.

Esta asignatura es de cursado obligatorio en todas las carreras de grado, durante el segundo cuatrimestre de primer año.

Formación Práctica:

Si bien durante la teoría y el coloquio se trabajan en actividades de formación práctica, la instancia principal que profundiza sobre esta actividad es la clase de práctica en los laboratorios.

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. Además de situaciones problemáticas genéricas, se plantean problemas relacionados a la Bioingeniería o a las ciencias médicas, Bioinformática o el Transporte, los cuales se adaptan al nivel del alumno de primer año.

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. Antes de dar inicio a la clase, el docente realiza una breve evaluación a fin de valorar los conocimientos del tema a trabajar, los cuales fueron desarrollados anteriormente en las clases de teoría y coloquio. Esto se realiza a fin de estimular al estudiante a revisar sus apuntes o las guías de estudio, como medio para aumentar su rendimiento.

Listado de Actividades de Formación Práctica:

Listado de actividades prácticas:

Introducción a Python, variables y operadores

Estructuras de control de flujo

Cadenas en Python

Estructuras de datos

Funciones 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: 35 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 las 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:

Primer Examen Parcial: 09 de Septiembre de 2024

Segundo Examen Parcial: 04 de Noviembre de 2024

Recuperatorio 01: 23 de Septiembre de 2024

Recuperatorio 02: 22 de Noviembre de 2024

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:

Considerando una media cuatrimestral de 100 estudiantes inscriptos para cursar la asignatura (cursado obligatorio), distribuidos en 1 comisión de teoría, 2 comisiones de coloquio y 6 comisiones de práctica, se sugiere el siguiente personal docente y su distribución de tareas:

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

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: Analía CHERNIZ.

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: Yanina ATUM.

3 JTP (DP): responsables de hasta 2 clases prácticas cada uno y seguimiento de actividades en la PVA: Iván PERALTA, Rodrigo PERALTA, Ricardo RETTORE.

2 Aux. Alumnos (DS): colaboradores en las clases prácticas: Laura OSUNA, Maximiliano COLAZZO.

* Personal afectado a otras actividades académicas dentro de la FI-UNER, que no disponen de la totalidad de su dedicación en esta asignatura.

Actividades de Investigación Gestión y Extensión:

De la Mg. Analía Cherniz:

Integra el Laboratorio de Investigación del Movimiento Humano (LIMH) de la FIUNER y el proyecto PID "Desarrollo e implementación de herramientas de análisis del movimiento humano para asistir en la rehabilitación motriz en personas con discapacidad", que dirige la Dra. Paola Catalfamo.

Integra el proyecto PID "Prácticas educativas mediatizadas en la Universidad Nacional de Entre Ríos. Inclusión de tecnologías digitales e innovación pedagógica en la post-pandemia", que dirige el Dr. Gonzalo Andrés y la Dra. Ileana Tossolini.

Integra el Grupo de Investigación y Desarrollo en Enseñanza de la Ingeniería (GIDEI).

Es Subdirectora del Departamento Informática y miembro de la Comisión Directiva del Departamento.

Es miembro del Comité Académico de la carrera de posgrado "Especialización en Sistemas Embebidos".

Es Coordinadora de la carrera de posgrado "Maestría en Enseñanza de la Ingeniería" y Tutora en dicha carrera.

Es Referente del Programa "UNER Saludable" para la sede FIUNER.

Se ha incorporado a la Comisión de Accesibilidad y Discapacidad de la FIUNER.

De la Mg. Yanina Atum:

Integra el Centro de Investigación "Centro de Ingeniería en Rehabilitación e Investigaciones Neuromusculares y Sensoriales" (CIRINS) de la FIUNER, bajo la dirección del Dr. Rubén Acevedo.

Integra el Grupo de Investigación y Desarrollo en Enseñanza de la Ingeniería (GIDEI).

Integra la Comisión directiva del Departamento Informática de la FIUNER.

Integra la Comisión de Seguimiento de Planes de Estudio de la FIUNER.

Es Tutora de la Maestría en Enseñanza de la Ingeniería de la FIUNER.

Es Directora de la Maestría en Ingeniería Biomédica de la FIUNER.

Requisitos de admisión para alumnos oyentes:

Los alumnos que deseen cursar la asignatura como alumnos oyentes deberán informar de su situación al profesor responsable de la asignatura y, además, al departamento alumnado o a Secretaría académica. Su admisión quedará supeditada a la disponibilidad de cupo dentro del grupo de trabajo, priorizándose la asistencia de los alumnos inscriptos a las carreras de la FIUNER, cuya condición sea regular o libre.

Infraestructura, equipamiento y recursos necesarios:

Para el dictado de teoría deberá contarse con un aula con una capacidad de entre 40 y 100 alumnos (dependiendo del cuatrimestre de cursado). Para las clases de coloquio se necesitarán aulas con capacidad para 50 alumnos aproximadamente.

Las clases de prácticas se desarrollaran en los Laboratorios de computación, los cuales tienen capacidad para 25 alumnos. Será necesario que las computadoras tengan instalado un compilador de C++ editor de el uso de Computadora de escritorio con sistema operativo Linux instalado y componentes informáticos.

Se utilizará cañón para proyección de diapositivas durante las clases de teoría y para el desarrollo de alguna de las clases prácticas de laboratorio o coloquio.

Otros: