



UNIVERSIDAD NACIONAL DEL COMAHUE

CENTRO UNIVERSITARIO REGIONAL ZONA ATLANTICA

PROGRAMA DE LA ASIGNATURA: Introducción a la Programación

CARRERA: Tecnicatura Universitaria en Desarrollo Web

CURSO: Primer año

ORDENANZA: Plan: 0885/12 - Modificatorias: 1074/13 - 1075/13

AÑO: 2024

CUATRIMESTRE: 1°

EQUIPO DE CATEDRA:

Profesor:

Master en Big Data y BI Carolina Juárez

Ayudante AYP 3:

Ingeniero en Informática Alberto Andría

1. FUNDAMENTACIÓN

Introducción a la Programación es una asignatura fundamental en la carrera de Tecnicatura Universitaria en Desarrollo Web, formando parte del conjunto de materias del primer cuatrimestre. Su objetivo es proporcionar a los alumnos los conocimientos básicos de programación, así como las técnicas y herramientas necesarias para abordar la resolución de problemas, como parte integral de su formación profesional que se desarrollará y profundizará en materias posteriores.

En este curso, los alumnos adquirirán habilidades para diseñar estrategias de resolución de problemas, desarrollar métodos de diseño de algoritmos y familiarizarse con el lenguaje de programación Python y el framework Django para la creación de aplicaciones web. La elección de esta combinación se debe a su reconocida eficacia en el desarrollo web, destacándose por su facilidad de uso, productividad, comunidad activa, escalabilidad y enfoque en la seguridad.

Esta combinación de herramientas proporcionará a los alumnos una base sólida para enfrentar los desafíos del desarrollo web y les permitirá construir aplicaciones robustas y de alta calidad en el futuro

2. OBJETIVOS SEGÚN PLAN DE ESTUDIOS

sin objetivos según plan de estudio

2.1 OBJETIVOS DEL PROGRAMA

El objetivo principal de esta cátedra es capacitar a los estudiantes en la resolución de problemas de complejidad moderada utilizando computadoras. Dado que existe una brecha significativa entre la formulación de un problema y la implementación de su solución en un programa informático, este objetivo se alcanza a través de las siguientes etapas:

1. Desarrollar habilidades para identificar situaciones problemáticas y plantear posibles enfoques de solución.
2. Traducir algoritmos diseñados en soluciones efectivas mediante la escritura de programas en Python.
3. Familiarizarse con Python, un lenguaje que permite crear aplicaciones completas de manera eficiente, facilitando así el aprendizaje desde cero.
4. Iniciarse en el uso del framework Django, como herramienta complementaria para el desarrollo web.

Además, se busca que los alumnos adquieran conocimientos básicos sobre computación y dominen la terminología esencial en el campo. Asimismo, se les desafía a resolver problemas que serán fundamentales en asignaturas futuras, al tiempo que se les enseña a desarrollar programas simples dentro del paradigma imperativo.

3. CONTENIDOS SEGÚN PLAN DE ESTUDIOS

Algoritmos. Resolución de problemas y clases de problemas con computadora. Programas.

Estructura General de un programa. Tipos de datos primitivos. Estructuras de control. Modularización.

4. CONTENIDOS ANALÍTICOS

Unidad 1: Resolución de Problemas

- Tipos de problemas y definiciones básicas.
- Representación de problemas: Descripción verbal, diagramas, matemática (ecuaciones, grafos, lógica), y geometría.
- Estrategias para resolver problemas.

Unidad 2: Algoritmos

- Definición y propiedades de algoritmos.
- Diseño de algoritmos y análisis: tratamiento de datos, entrada y salida de datos, variables, tipos y asignaciones.
- Metodologías de representación: diagramas de líneas y diagramas de flujo.
- Ejecución y verificación de algoritmos: traza de un algoritmo.

Unidad 3: Lenguaje de Programación

- Introducción al lenguaje y conceptos básicos de un programa.
- Estructura y sintaxis del lenguaje.
- Tipos de datos, variables y ámbito de las variables.
- Entrada y salida de datos, manejo de strings.

Unidad 4: Estructuras de control Condicionales

- Expresiones booleanas y alternativas en el análisis, diseño, implementación y verificación.
- Alternativas simples, dobles, múltiples y combinadas.

Unidad 5: Estructuras de control Repetitivas

- Condiciones booleanas y modularización/funciones.

Unidad 6: Estructura de Datos

- Listas y Tuplas: operaciones básicas, similitudes y diferencias
- Diccionarios y su uso en el almacenamiento de datos

5. BIBLIOGRAFÍA

Bibliografía básica

- Apuntes de cátedra.
- Polya, George. How to solve it. Editorial: Princeton University Press. Princeton, New Jersey. 1973
- Savitch, Walter, Carrano, Frank. Java: An Introduction to Problem Solving & Programming 5th Ed. Editorial: Pearson Prentice Hall. 2008..

Soporte en Internet:

- <https://www.python.org/>
- <https://docs.python.org/es/3/tutorial/>

Bibliografía de consulta

- J. Mason, B. Leone y K. Starey. Pensar Matemáticamente. Editorial: MEC, Labor Barcelona 1992.
- M. Lucas, J.P. Peyrin y P.C. Scholl. Algorítmica y Representación de Datos. Tomo I. Editorial Masson S.A. 1990.
- J.D. Warnier. Síntesis de Programación Lógica (Los tratamientos y sus datos). Editores Técnicos Asociados S.A. Barcelona. 1979.

6. PROPUESTA METODOLÓGICA

Se establecen foros para mantener a los estudiantes informados sobre novedades relacionadas con la materia, así como un foro por cada unidad para agrupar consultas sobre conceptos y trabajos prácticos.

Cada unidad del curso se aborda de manera teórico-práctica e incluye un trabajo práctico y cuestionario correspondiente. Tanto las teorías como los trabajos prácticos están disponibles en la plataforma virtual.

Hay un total de seis cuestionarios, uno por cada unidad temática del programa de Introducción a la Programación. Solo podrán rendir el examen parcial aquellos alumnos que hayan aprobado los cuestionarios correspondientes.

Como preparación previa a un examen parcial, se lleva a cabo un simulacro con el mismo nivel de dificultad que el examen real, permitiendo así que los alumnos se familiaricen con la complejidad de los ejercicios a evaluar. La cátedra proporciona la solución del simulacro, la cual se publica en la plataforma para el acceso de todos los estudiantes.

Dado que se espera que los alumnos adquieran habilidades para participar en proyectos de software de manera colaborativa y trabajar en equipos para el desarrollo de aplicaciones web, se proponen instancias de entrega de trabajos prácticos y evaluativos en grupo a lo largo del curso..

7. EVALUACIÓN Y CONDICIONES DE ACREDITACIÓN

Las condiciones que se detallan a continuación son aplicables tanto a los alumnos en modalidad **presencial, semipresencial y nodos:**

Acreditación de la asignatura como alumno regular:

El cursado de la asignatura tiene una validez de tres años. Para ser considerado alumno regular, se deben cumplir con los siguientes requisitos:

1. Cumplir con las fechas de entrega y obtener una calificación igual o superior a 60 en los cuestionarios y trabajos prácticos, tanto individuales como grupales.
2. Obtener una calificación igual o superior a 40 en dos parciales teórico-prácticos realizados

durante la cursada, uno a mitad y otro al final.

3. Aprobar el examen final presencial en la sede correspondiente con una calificación igual o superior a 40.

Al cumplir con los requisitos 1 y 2, el alumno obtiene la condición de alumno regular y puede rendir el examen final para aprobar en la mesa de exámenes final según el calendario académico.

Acreditación de la asignatura por promoción:

Este régimen de acreditación aplica para los alumnos que aprueben en primera instancia:

1. Seis cuestionarios teórico-prácticos (uno por cada unidad temática del programa).
2. Trabajos prácticos obligatorios.
3. Obtener una calificación de 70/100 en el primer y segundo parcial.
4. Realizar un coloquio presencial en la sede correspondiente, excepcionalmente la cátedra podrá autorizar un coloquio virtual con cámara y sonido activado por parte del alumno, presentando además un documento de identificación.

La aprobación de la materia se logra al cumplir con los cuatro requisitos mencionados anteriormente.

Acreditación de la asignatura como alumno libre:

La acreditación como alumno libre se rige por la ordenanza vigente. Para acceder a esta modalidad, además se debe cumplir con la siguiente condición previa:

- Aprobar, 15 días antes de la fecha del examen final, los seis cuestionarios teóricos/prácticos obligatorios disponibles en PEDCO.

Luego, y solo si se ha cumplido con la condición previa, se deberá aprobar el examen final teórico práctico, el mismo será presencial en la sede correspondiente.

8. PARCIALES, RECUPERATORIOS Y COLOQUIOS

Independientemente de la modalidad de cursado elegida, todos los estudiantes deben aprobar todos los cuestionarios y Trabajos Prácticos Obligatorios, así como también dos exámenes parciales.

Los cuestionarios tienen dos instancias de recuperación, con un tiempo mínimo de espera de 72 hs entre fechas. de reintento.

En caso de no aprobar los exámenes parciales en su primera instancia, se brinda una oportunidad de recuperarlos al finalizar el cuatrimestre.

Los coloquios se llevan a cabo de forma presencial en la sede correspondiente, excepcionalmente la cátedra podrá autorizar el coloquio fuera de sede, con la condición que el alumno utilice tecnología de cámara y micrófono, siempre acompañados de un documento que verifique la identificación del alumno.

9. DISTRIBUCIÓN HORARIA

Modalidad	Minutos *	Porcentaje
Presencialidad en el establecimiento	60	12.5%
Presencialidad remota	204	42.5%
EAD Asincronica - PEDCO	216	45%

* Minutos de dictado semanal

Observaciones

- Encuentros: Lunes y Miércoles 15 a 17 hs híbridos (alumnos presencial y virtual sincrónico)
- Lunes: Clase Teórica
- Miércoles: Clase Práctica

10. CRONOGRAMA TENTATIVO

Cuatrimestre				
Tiempo / Unidades	Marzo	Abril	Mayo	Junio
Unidad 1	X			
Unidad 2	X			
Unidad 3		X		
Unidad 4			X	
Unidad 5			X	
Unidad 6				X

11. PLANIFICACIÓN DE ACTIVIDADES EXTRACURRICULARES

- Trabajos de Investigación
- Análisis de Programas de Simple Complejidad publicados en Internet

Norma E. Carolina Juárez
Profesor a Cargo

Viedma, 16 de Marzo de 2024.