

CARRERAS DE DOS AÑOS

TECNICATURA EN PROGRAMACIÓN DE COMPUTADORAS

**PLAN DE ESTUDIOS Y
CONTENIDOS MÍNIMOS**

TECNICATURA EN PROGRAMACIÓN DE COMPUTADORAS

PLAN DE CARRERA

Ciclo Introdutorio

Código	Materia	Horas semanales
1	LÓGICA	5
2	PROGRAMACIÓN I	10
3	ORGANIZ.Y FUNCIONAM.COMPUTAD.	5
4	PROCESOS Y SISTEM.ORGANIZAC.	5
5	MATEMÁTICA DISCRETA	5
6	PROGRAMACIÓN II	10
7	AMB.FUNCIONES SISTEMA OPERAT.	5
8	REPRES.REQUISITOS DE SOFTWARE	5
	PRUEBA DE NIVEL DE INGLÉS TÉCNICO I	
	1RA. PRÁCTICA DE LABORATORIO	
	2DA. PRÁCTICA DE LABORATORIO	

Básico Uno

Código	Materia	Horas semanales
9	PROGRAMACIÓN III	10
10	ELEMENTOS COMPUTACIÓN EN RED	5
11	PROGRAM.AMBIENTES BASE DATOS	5
12	VERIF. DOCUMENTACIÓN PROGRAMAS	5
13	PROYECTO CONSTRUCCIÓN SOFTWARE	10
14	PROGRAM.AMBIENTES DE REDES	5
15	BASE DE DATOS	5
16	ELEMENTOS DE SEGURIDAD INFORM.	5
	PRUEBA DE NIVEL DE INGLÉS TÉCNICO II	
	3RA. PRÁCTICA DE LABORATORIO	
	4TA. PRÁCTICA DE LABORATORIO	
	5TA. PRÁCTICA DE LABORATORIO	
	6TA. PRÁCTICA DE LABORATORIO	
	TRABAJO PROFESIONAL SUPERVISADO	
	PRUEBA DE LECTO-COMPRESIÓN DE INGLÉS	

Título final: TECNICO EN PROGRAMACION DE COMPUTADORAS

CONTENIDOS MINIMOS DE LAS MATERIAS DE PLAN 2009

1er.año – 1er. cuatrimestre

Lógica

Teoría de Conjuntos. Funciones. Problemas de conteo, principio de Dirichlet. Lógica proposicional, conectivos, tablas de verdad, deducción y validez formal. Lógica de predicados, cuantificadores universales y existenciales. Nociones de implicación, negación y contradicción. Estructuras de pruebas formales. Inducción, inducción completa. Definiciones matemáticas recursivas.

Programación I

Introducción a la Programación Orientada a Objetos, Resolución de problemas y algoritmos. El paradigma de la programación orientada a objetos. Clases y objetos. Estructuras de control. Introducción a la recursividad. Tipos primitivos. Arreglos y cadenas de caracteres. Verificación de programas, construcción de juegos de prueba y pruebas de escritorio. Documentación. Codificación de algoritmos en un lenguaje orientado a objetos. Edición, compilación y ejecución de programas. Ejecución de juegos de prueba y análisis de resultados. Búsqueda, análisis y corrección de errores.

Organización y Funcionamiento de Computadoras

Evolución histórica del computador. Componentes de un sistema de computación. Representación de los datos a nivel de máquina y operaciones. Organización básica y funcionamiento de una máquina tipo von Neumann a nivel de Ensamblador. Sistemas de memoria, tecnología y jerarquía. Manejo de errores y confiabilidad. Concepto de memoria caché. Dispositivos de almacenamientos secundarios. Noción de memoria virtual. Buses. Dispositivos externos de entrada y salida.

Procesos y Sistemas de las Organizaciones

Operaciones típicas de distintos tipos de organizaciones: comprobantes y registros habitualmente utilizados. Concepto de proceso: físicos y con datos. Materiales y procesos de transformación. Circuitos administrativos: compras y pagos, ventas y pagos, producción, movimientos de fondos y valores, prestación de servicios.

Controles internos fundamentales. Auditoría interna. Estructura de las organizaciones y tipos de organización: en base a funciones, por negocios, por procesos.

1er.año – 2do. cuatrimestre

Matemática Discreta

Números enteros. Propiedades: divisibilidad y congruencia. Funciones recurrentes. Combinatoria (combinaciones, permutaciones, y variaciones). Resolución de relaciones de recurrencia. Árboles: representación y recorridos. Grafos no dirigidos y dirigidos: representación y recorridos. Algoritmos básicos. Grafos Expandidos. Coloración y estructuras de orden. Estrategias transversales.

Programación II

Objetos y abstracción de datos. Herencia, jerarquía de clases, polimorfismo, clases abstractas e interfaces. Excepciones. Clases de colección e iteradores. Archivos de texto. Interfaces gráficas de usuario. Herramientas de modelación: diagramas de clase, UML, tarjetas CRC. Lenguajes de programación y ambientes para el desarrollo de software. Diseño e implementación de algoritmos en diferentes lenguajes y ambientes de programación. Verificación y depuración de programas. Programación orientada a procesos, resolución de problemas utilizando un lenguaje apropiado.

Ambiente y Funciones del Sistema Operativo

Concepto de programa almacenado, lenguaje de programación y código ejecutable, mecanismos de traducción o interpretación. Sistema operativo: su evolución y funcionalidades que brinda al desarrollador y al usuario. Administración de recursos: memoria, procesador, entrada/salida. Multiprogramación y multiprocesamiento. Concepto de proceso: administración de procesos, manejo de excepciones, prioridades. Estado del usuario y del sistema. Administración de la memoria. Organización lógica y física de los datos en archivos. Recuperación de errores.

Representación de Requisitos de Software

El ciclo de vida del software y el problema de requisitos. Análisis del problema: definición, involucrados, límites de la solución, restricciones. Modelización del negocio: concepto, técnicas. Concepto de requisito, necesidad de acuerdo y documentación. Requisitos funcionales. Análisis de escenarios y de la interacción del usuario con el sistema. Casos de uso, representación y análisis. Modelización con UML. Análisis de los datos del sistema: modelo Entidad/Relación, diccionario de datos. Análisis estructurado de procesos: DFD. Análisis de estado.

2do.Año – 3er.Cuatrimestre

Programación III

Algoritmos y estructuras de datos. Análisis temporal y espacial de algoritmos iterativos. Medidas asintóticas. Algoritmos recursivos. Estrategia de dividir para conquistar, backtracking. Tipo de dato abstracto. Especificación, aplicación e implementaciones de pilas, colas, listas, árboles, tablas de dispersión y grafos. Análisis de las diferentes implementaciones. Archivos binarios y archivos índices. Árboles B. Implementación de las operaciones primitivas de distintos tipos de datos utilizando diferentes representaciones en memoria. Análisis del comportamiento de las diferentes implementaciones. Implementación de aplicaciones utilizando tipos de datos abstractos.

Programación en ambiente de Bases de Datos

Evolución de los sistemas de información que la almacenan y permiten recuperarla. Conceptos introductorios de Base de Datos, independencia de los datos. DBMS: funciones y componentes. Revisión del modelo conceptual de datos: modelo Relacional. Bases de datos relacionales: álgebra relacional y cálculo relacional. Lenguajes de consulta a base de datos: SQL; optimización de consultas; inserción de consultas y actualizaciones en lenguajes procedurales.

Elementos de Computación en Red

Antecedentes de comunicación entre computadores y desarrollo de Internet. Panorama de arquitecturas y topologías de redes. La web como un ejemplo de computación cliente servidor, protocolo HTML, formato de páginas web. Comprensión de datos. Redes y servicios: contraseñas y mecanismos para control de acceso, nombres de dominios y nombres de servicios.

Verificación y Documentación de Programas

Estrategias de prueba. Prueba de código, de especificaciones, práctica de prueba. Pruebas de programas, del sistema. La documentación de lo que se desarrolla, reglas de buenas prácticas. Recorridos y revisiones de programas. El plan de pruebas, especificación de datos de prueba, uso de bibliotecas de prueba. El empleo de herramientas automatizadas para la prueba.

2do. Año- 4º cuatrimestre

Proyecto de Construcción de software

Inmersión en una experiencia colectiva de programación sobre un sistema de complejidad moderada poniendo el énfasis en prácticas de ingeniería de software, incluyendo desarrollo incremental, verificación sistemática del software y depuración de artefactos de software. Revisión de elementos de programación, encapsulamiento, herencia, polimorfismo y estructuras de datos. Empleo de buenas prácticas de programación, documentación, manejo de excepciones y revisión cruzada de código. Validación unitaria e integral: planificación, preparación y pruebas, empleo de herramientas, estrategias de depuración. Administración de la configuración.

Programación en Ambiente de Redes

Internet como ejemplo de computación cliente – servidor. Capas de software que componen un sistema. Programación respondiendo a eventos. Procesamiento de transacciones, riesgos y mecanismos para asegurar la integridad del procesamiento.

Bases de Datos

Sistemas de administración de bases de datos: componentes, arquitectura de la base de datos: componentes, arquitectura de la base de datos e independencia de los datos. Diseño de una base de datos relacional: dependencia funcional, normalización, dependencia multivaluada, dependencia conjunta, teoría de representación. Conversión del esquema conceptual al esquema relacional, entidad, e integridad referencial. Procesamiento de transacciones, errores y recuperación, control de concurrencia. Bases de datos distribuidas: almacenamiento distribuido, procesamiento distribuido de consultas, modelo de transacciones distribuidas, soluciones, modelo cliente-servidor. Problemas conexos.

Elementos de seguridad informática

Paradigma de la Seguridad Informática. Vulnerabilidades. Amenazas y Riesgos Informáticos. Herramientas de Seguridad Informática. Modelos de Análisis. Elementos de Criptografía Aplicada. Organización y Control de la Seguridad Informática. Marco Legal, Reglamentario y Normativo para Argentina.

Obligaciones Académicas

Prueba de Inglés Técnico I

Consiste en una prueba para verificar el dominio del lenguaje técnico en inglés. Los alumnos podrán realizar un curso preparatorio no obligatorio que ofrece la universidad.

Prueba de Inglés Técnico II

Consiste en una prueba para verificar el dominio del lenguaje técnico en inglés. Los alumnos podrán realizar un curso preparatorio para la universidad.

Prácticas de Laboratorio

Los alumnos deberán realizar 6 prácticas de laboratorio con una carga de 32 horas cátedra cada una. Los contenidos se referirán respectivamente a las asignaturas: Programación I, Programación II, Programación III, Programación en Ambientes de Bases de Datos, Programación en Ambientes de Redes y Proyecto de Construcción de Software.

Prueba de Lectocomprensión de Idioma Inglés

Consiste en una prueba para verificar la capacidad de lectura y comprensión de textos específicos de inglés vinculados a las disciplinas de las carreras. Los alumnos podrán realizar un curso no obligatorio que ofrece la universidad para rendir el examen obligatorio de lectocomprensión de idioma inglés.

Trabajo Profesional Supervisado

Consiste en 300hs. Reloj de actividades de programación bajo supervisión que el estudiante tiene que realizar en ambientes de desarrollo de software. Las tareas a desarrollar tienen que ser, principalmente, de programación, pero también puede comprender actividades de especificación, diseño, documentación y verificación de programas en el contexto de proyectos de desarrollo de software. Estas actividades pueden ser desarrolladas en el marco de pasantías con programas definidos convenidas por la Facultad con empresas de sector; actividades homologables que realice el estudiante en el marco de su empleo, previo acuerdo entre el docente responsable y el supervisor de su actividad laboral; o desarrollo de programas realizados en el marco de proyectos de investigación u otros proyectos de desarrollo de software que realice la universidad. La dedicación del estudiante no podrá ser menor a los 20 ni mayor a las 40 horas semanales.