

## **ANEXO 1**

### **Carreras de tecnicaturas informáticas.**

#### **Tecnicatura en programación de computadores.**

##### **Fundamentación de la carrera**

La actualidad de los sistemas productivos, relacionados con la producción de bienes y servicios se caracteriza por un vertiginoso avance constante y recurrente del campo tecnológico, en este contexto el ámbito de la informática requiere de un campo de actividades específicas.

En nuestro país, en forma acelerada las herramientas y procesos informáticos se han insertado en la cultura de las organizaciones públicas y privadas, dejando una fuerte impronta en el desarrollo de las actividades productivas, sociales e individuales.

En este contexto de cambio y adecuaciones permanentes, se plantea la necesidad de formar personal auxiliar en el área de sistemas y en este marco, desde el sistema educativo formal y más específicamente desde la educación superior universitaria, se generan instancias que promueven acciones para impulsar una mejora en el área de la informática.

Frente a la identificación de las necesidades descritas, y en relación a lo estipulado en el plan Nacional de apoyo a la informática, desarrollados por el ministerio de Educación, Ciencia y Tecnología, en vistas a mejorar la calidad de formación de técnicos, promover las reformas curriculares necesarias para adecuar los planes de estudio a las demandas de competencias requeridas en los mercados laborales e incrementar la cantidad de inscriptos y graduados en las carreras de formación de técnicos relacionados con la informática y servicios informáticos.

La facultad de ingeniería solicita la aprobación del título de preparado de técnico en programación de computadores.

Fundamentos dicho requerimiento desde múltiples aspectos, que darían cobertura a una demanda específica regional:

En lo académico, posibilitaría una articulación eficaz con titulaciones similares o afines otorgadas por instituciones educativas con reconocimiento oficial.

En el ámbito de las capacidades y posibilidades de una temprana y concreta inserción laboral, sería un instrumento para generar entradas en el mundo del trabajo con una titulación intermedia universitario.

En el ámbito institucional permitiría ampliar la oferta académica con un carácter inclusivo en las carreras relacionadas con las ciencias de la ingeniería.

En el ámbito socio-productivo, reforzaría la relación universidad-empresa. Al ofrecer una capacitación específica de corta duración los trabajadores que estén en condiciones de acceder a una carrera corta universitaria.

Por ello, la Facultad de Ingeniería de la Universidad Nacional de Lomas de Zamora en concordancia con lo antes mencionado y en el marco de una política institucional desde los campos de la ingeniería, así como toda otra actividad de impronta tecnológica, resulta de la aplicación de los fundamentos científicos formales y operacionales proporcionados por las ciencias básicas e informáticas, propone la creación de la presente Tecnicatura superior como aporte a la profesionalización de quienes deben desarrollar sistemas informáticos aplicables a áreas de gestión organizaciones y/o simulación y modelización de las propiedades de la materia y sus transformaciones en estructuras, maquinarias, productos, sistemas y procesos.

## **Título a otorgar.**

### **Técnico en programación de computadores.**

## **Requisitos de ingreso.**

Podrán ingresar a la tecnicatura en programación de computadores los aspirantes que acrediten titulación de Nivel Medio o Polimodal, otorgadas por instituciones públicas o privadas.

## **Objetivos.**

### **Que los cursantes:**

- Conozca y aprenda los lenguajes de programación de uso en PC mas actuales;
- Pueda instrumentar soluciones efectivas con su uso y aplicación;
- Adquiera manejo solvente de los entornos ,as requeridos en el mercado;
- Actúe con ética en el desempeño de sus funciones;

## **Perfil profesional.**

El técnico en programación de computadores estará capacitado para:

- Realizar programas o componentes de sistemas de computación;
- Interpretar especificaciones de diseño.
- Documentar los productos realizados;
- Verificar los componentes programados;
- Buscar causas de malfuncionamiento y corregir los programas o adaptarlo a cambios en las especificaciones desarrollando las actividades descriptas en el perfil profesional y cumpliendo con los criterios de realización establecidos para las mismas en el marco de un equipo de trabajo organizado por el proyecto.

## **Alcances del título.**

Interpretar especificaciones de diseño o requisitos de las asignaciones a programar, comprende validar la coherencia e integridad de las mismas y convalidar su propia interpretación con quienes la hayan realizado o provisto.

Analizar estrategias para desarrollar la asignación recibida en el contexto del proyecto y de la tecnología a utilizar. Implica investigar para refinar aspectos ambiguos o insuficientemente conocidos del diseño o los requisitos de la solución, resolver problemas de lógica que implican diseño o refinación de algoritmos o estructuras de datos que faciliten o permitan la solución, buscar componentes disponibles y adecuados para utilizar en la solución, bosquejar eventuales estrategias alternativas y evaluarlas para seleccionar la mas apropiada.

Producir el código que resolverá la asignación en el contexto de la tecnología asignada al proyecto. Esto incluye la modificación (agregado, reemplazo o eliminación) de código ya escrito, sea para corregir errores observados en pruebas o para cambiar funcionalidades o el comportamiento de productos con existencia previa.

Comprende la definición o instanciacion de clases, escritura de algoritmos, reestructuración de datos necesarios, o la incorporación y eventual adaptación de componentes obtenidos de bibliotecas o de otros programas, respetando estándares de buena practica y normas internas de la empresa o proyecto, así como identificando componentes o partes que puedan ser potencialmente reutilizados en el futuro.

Verificar unitariamente el producto desarrollado para asegurarse que cumple con las especificaciones recibidas. Implica planificar y documentar casos de prueba, preparar

datos y entornos de prueba (“testing”), generar código adicional para simular el entorno o activar las pruebas, analizar causas de comportamiento o resultados no previsto para correr el código incorrecto o preocuparse por mejorar la eficiencia (“tunning”) de la solución.

También comprende participar en la realización y control de pruebas de productos de otros.

Depurar el código de programas para decidir que hay que corregir. Implica revisar especificaciones y código de componentes unitarios APRA encontrar las partes o instrucciones que provocaran malfuncionamiento, incidentes reportados o ineficiencias, con el objeto de analizar sus causa y definir acciones correlativas.

Realizar, con otros programadores o con especialistas, revisiones cruzadas de código o de interfaces. Implica revisar el cumplimiento de especificaciones, de estándares y de buenas practicas, evaluar el uso eficiente de recurso y del cambien de desarrollo y aportar observaciones con propuestas de cambio tendientes a mejorar la calidad, mantenibilidad y eficiencia del producto.

Carga horaria:

- 1600 horas divididas en cinco cuatrimestres.

### **Organización del plan de estudios.**

El presente diseño curricular involucra aspectos del desarrollo académico y profesional en concordancia con el perfil planteado, tanto en lo relativo a la formación disciplinar, como a la epistemológica, metodológica y técnica, así como la actualización y profundización de los conocimientos de orden disciplinar.

En consideración a la orientación y a los objetivos del presente proyecto educativo, así como a los establecimientos contenidos en el perfil profesional, y sus consecuentes alcances, el presente diseño curricular se organiza a través de la definición de una línea disciplinar de conocimientos interrelacionados y con grado de complejidad creciente.

Año	Cuat.	Materia	Correlatividades
1	1	Introducción a la informática	
		Lógica y algoritmia	
		Programación I - C y C++	
	2	Arquitectura de computadoras	Introducción a la Informática
		Lectura de requisitos UML	Lógica y algoritmia - Programación I
		Programación II - Java	Programación I
2	3	Sistemas operativos	Arquitectura de Computadoras
		Seguridad	Arquitectura de Computadoras
		Programación III fundamentos del entorno .Net y C	Programación II
	4	Programación IV - ASP.Net (160 hs)	Programación III
		Aplicaciones web	Programación II
		Tecnología Java	Programación IV
3	5	Ética profesional	Programación IV - Aplicaciones Web
		PPS - Pasantía	Programación IV - Aplicaciones Web
		Taller integrador	Programación IV - Aplicaciones Web

**Cuadro de congruencia entre alcances y contenidos.**

Alcance del título	Contenidos relacionados		Formación practica
Formación básica e integral del profesional	Introducción a la informática. Arquitectura de computadores. Sistemas operativos Lógica y algorítmica	Entender y reconocer las estructuras internas de la computadora, Conceptos básicos actualizados sobre sistemas operativos. Lógica proposicional Análisis de algoritmos.	Links a documentos técnicos adicionales, disponibles en Internet. Biblioteca virtual con soporte electrónico. Herramientas y utilitarios de distribución libre y gratuita. Se realizan practicas de laboratorio en cada uno de los módulos, llevando casos de trabajo de la realidad para su discusión y resolución en grupo
Interpretar, en el contexto del proyecto, especificaciones de diseño o requisitos de las asignaciones a programar Analizar estrategias para desarrollar la asignación	Lectura de Requisitos UML		Interacción con objetos, mensajes, asociación y composición, Análisis y diseño orientado a objetos, utilizando lenguaje UML.

recibida		Uso de diagramas UML, casos de uso, conceptos de clases especificación, secuencia actividad
Producir el código Verificar unitariamente el producto desarrollado. Depurar el código.	Programación I C Y C++ Programación II JAVA Fundamentos del entorno .Net Tecnología Java para aplicación Web. Programación III MS-ASP.net	Programar aplicando instrumentos para la resolución de problemas en forma algorítmica. Programación orientadas a objeto java: encapsulación herencia y polimorfismo. Configuración del layer de servicios. Diseño de interfases Web. Testing Debugging. Desarrollo de aplicaciones
Instrumentar soluciones efectivas con su uso y aplicación	Taller integrador, Practica profesional supervisada	Diseño de una aplicación distribuida y multicapas orienta a una intranet. Discutir y concluir en metodologías y mejores prácticas aplicadas al diseño, implementación y mantenimiento.

### **Contenidos mínimos de las asignaturas.**

#### **INTRODUCCION A LA INFORMATICA**

- Lograr que el alumno conozca la estructura interna y los principios de funcionamiento del computador. Familiarizar al alumno con el vocabulario técnico y específico de la profesión. Introducir al alumno en el área de conocimientos
- Definición de computador. Sus características, estructuras y aplicaciones. Generaciones y tipos de computadores. Hardware. Introducción a la arquitectura. Representación de caracteres. Operaciones. Variables análogas y discretas. Códigos. Relaciones entre variables (álgebra de Boole) Compuertas. Circuitos lógicos. Sistemas combinacionales y secuenciales.

#### **LOGICA Y ALGORITMIA**

- Lógica proposicional. Razonamiento proposicionales.
- Análisis de algoritmos y estructuras de datos básicas. Arrays (listas y tablas).
- Estructuras y uniones. Diseño de algoritmos. Recursividad, ordenación y búsquedas. Estructuras de datos fundamentales (abstracción de datos, listas, pilas, colas y tablas hash) Estructura de datos avanzadas (árboles y grafos)

## **PROGRAMACIÓN I-C YC ++**

- Programar aplicando instrumentos para la resolución de problemas en forma algorítmica.
- Acentuar el pasaje de las capacidades analíticas a aquellas proyectuales.
- Indecencia entre la herramienta y la implementación.
- Implementación exitosa de algoritmos básicos utilizando como herramienta un lenguaje de programación.
- El lenguaje C: conceptos básicos de C. Entrada y salida de datos. Preparación y ejecución de un programa en C. Instrucciones de control. Funciones. Estructura de un programa. Formaciones. Punteros. Estructuras y uniones. Archivo de datos. Programación a bajo nivel.
- Características adicionales de C.
- El lenguaje de programación C++: de C a C++. Extensiones orientadas a objetos.

## **ARQUITECTURA DE COMPUTADORES.**

- Entender y reconocer las estructuras internas de la computadora.
- Manejar el flujo de información mediante los periféricos.
- Permitir la introducción en la arquitectura y la organización de las computadoras.
- Sistemas numéricos. Operaciones aritméticas. Lógica electrónica. Compuertas. Circuitos combinatorios y secuenciales. Unidades aritméticas. Lógica electrónica. Compuertas. Circuitos combinatorios y secuenciales. Unidades aritméticas. Comparadores. Sumadores. Conversores de código. Decodificadores. Flip`'s-Flop`'s. Contadores. Registros. Representación de los datos a nivel de máquina. Lenguaje ensamblador. Jerarquía de memoria. Organización funcional. Máquinas algorítmicas.
- Representación de datos y aritmética del computador. Organización del computador. Diseño del procesador. Segmentación. Paralelismo entre instrucciones. Sistemas de memoria.

## **LECTURA DE REQUISITOS –UML**

- Programación orientada a objetos.
- Conceptos de orientación a objetos: clase, objeto, mensaje, relaciones, herencia, clase abstracta. Tipos genéricos. Asignación estática y dinámica de memoria. Polimorfismo.
- Interacción con objetos, mensajes, asociación y composición. Análisis y diseño orientado a objetos, utilizando lenguaje UML. Uso de diagramas UML, casos de Uso, conceptos de clases especificación, secuencia y actividad.

## **PROGRAMACIÓN II- JAVA**

- Funciones de programación orientada a objetos java: encapsulación, herencia y polimorfismo.
- Tipos de datos y expresiones en java.
- Construcciones del control de flujo java.
- Matrices y otros grupos de datos.

- Técnicas de gestión de errores utilizando el manejo de excepciones. Interfaces gráficas controladas por eventos, paneles, botones, etiquetas, campos y cuadros de texto.
- Funcionalidades de E/S. Multithreads.

## **SISTEMAS OPERATIVOS**

- Adquirir el dominio de conceptos básicos y actualizados sobre sistemas operativos.
- Introducir los lineamientos generales de nuevos desarrollos tecnológicos en estos temas.
- Generar una concepción global y un enfoque selectivo para las soluciones algorítmicas de los diferentes problemas que ocurren dentro de un computador y la correcta utilización de los mismos.
- Introducción a los sistemas operativos. Administración del proceso. Planificación del proceso. Concurrency de ejecución.
- Interbloques. Administración de recursos compartidos, sincronización y comunicación entre procesos.
- Administración de la memoria. Administración de recursos compartidos, sincronización y comunicación entre procesos.
- Administración de la memoria. Administración de dispositivos de entrada/salida. Sistemas de archivos. Protección y seguridad. Mediciones del sistema.
- Gestión de archivos y directorios. Seguridad y protección. Sistemas distribuidos. Algo de LINUX/WINDOWS.

## **SEGURIDAD**

- El entorno de red. Topologías. Dispositivos de conexión. Fundamentos de seguridad en redes. Routers, Firewall y Proxy servers. Funcionamiento básico.
- Definición de roles. Procesos involucrados. Introducción al servidor Proxy. Configuración de Proxy.
- Opciones ASP.net para punto net.

## **PROGRAMACIÓN III: FUNDAMENTOS DEL ENTORNO.NET. C\***

- Vb.net: El entorno de programación, el lenguaje y la sintaxis.
- Orientación a objetos.
- Testing y debugging.
- Desarrollo de aplicaciones.

## **PROGRAMACION IV-ASP.net**

- ASP.NET: introducción. Tipos de datos. Eventos. Propiedades. Librerías. Clases. Threads.
- ADO.NET
- XML: Editor XML. Niveles. Creación de datos y esquemas, Net framework, XML dentro del Framework, opciones de display.

## **APLICACIÓN WEB**

- Programación orientada a objetos (POO) en PHP.
- El patrón de diseño MVC (Model View Controller) para aplicaciones Web.
- Conceptos básicos de arquitectura orientada a servicios (SOA)
- Persistencia- El patrón de diseño DAO (Data access objects)

## **TECNOLOGIA JAVA**

- Java beans, servlets y Java Server pages. Aplicaciones distribuidas. Componentes FB. Servicio de mensajería java. Persistencia. Reusabilidad de componentes.
- Configuración del layer de servicios. Diseño de interfases Web. Compilación en tiempo y ejecución.

## **ETICA PROFESIONAL**

- Nociones básicas y aplicadas de la ética en la formulación y seguimiento de proyectos y en la conformación de los equipos de proyecto.

## **TALLER INTEGRADOR.**

- Asistencia para el diseño de una aplicación distribuida y multicapas orientada a una intranet.
- Aplicación del concepto de reutilización de clases y configuración de la plataforma para la implementación de una aplicación (java o puntonet)

## **PRACTICA PROFESIONAL SUPERVISADA.**

- Le permite al estudiante complementar su formación integral y lograr una mejor capacitación para responder a las necesidades de la industria.
- Lograr una mayor aproximación entre los objetivos y contenidos de formación propuesta por la facultad y las necesidades reales de las empresas del país.
- Permitirles a los estudiantes observar la importancia de la aplicación de una metodología de análisis, investigación, desarrollo y control, en los establecimientos industriales, y confrontarla con la aplicación del conocimiento específico.
- Lograr que el estudiante se enfrente a las responsabilidades reales y cotidianas que se viven en el ambiente empresarial.

## Metodología de trabajo y sistemas de regularidad y evaluación

### Metodología de trabajo

La definición de la estructura curricular y su gestión académica se inscriben en la concepción pedagógica del aprendizaje significativo, entendiéndose por significativo el aprendizaje que establece enlaces y vinculaciones sustantivas entre los conocimientos previos del sujeto, y los nuevos contenidos desarrollados por las acciones educativas.

Consecuentemente, la metodología de trabajo deberá enmarcarse en un enfoque globalizador e interdisciplinario que posibilite el mayor número posible de relaciones entre los múltiples y variados conocimientos, disponibles y nuevos. La organización y secuenciación de los contenidos de aprendizaje deberá considerar los aspectos de interrelación otros contextos del saber, a los efectos de facilitar la producción del sentido y la transferencia de los mismos a situaciones nuevas.

En tal sentido, la enseñanza deberá contemplar tanto la inclusión de facilitadores como de adecuados recursos didácticos para la comprensión de los contenidos de aprendizaje y su generalización y aplicación.

En lo específico disciplinar, la formación relativa supone el abordaje de conceptos y de procedimientos aplicables a las ciencias informáticas, lo que implica la integración



teórico- práctica acorde a los requerimientos y a las particularidades de cada espacio contenido en cada línea curricular previamente consignada.

#### Sistema de Evaluación

El sistema de evaluación previsto para el plan de estudios de la carrera deberá comprometer, en cada espacio curricular, las instancias evaluativas que seguidamente se detallan.

Diagnostica: para producir datos de partida de la situación potencial de aprendizaje de los cursantes y adecuar secuencialmente las estrategias didácticas tendientes a la producción de anclajes significativos.

De procesos: para monitorear los avances en la construcción de los aprendizajes y detectar los desvíos potenciales, sobre los cuales se redefinirán las propuestas pedagógico- didácticas.

De resultado o final: para medir los aprendizajes totales alcanzados y su integración, y verificar así los rendimientos a los efectos de la posible aprobación y promoción de cada asignatura en cuestión.

Por lo expuesto, las modalidades de evaluación diagnostican y de proceso podrán contemplar producciones escritas, orales y/o multimediales, de resolución individual y/o grupal, dependiendo de las características de cada asignatura y de los contenidos en ella involucrados. Su desarrollo podrá ser áulico o extra áulico.

Las evaluaciones parciales y finales deberán ser presenciales, orales y/o escritas, de resolución individual o grupal, acorde a la modalidad establecida por cada cátedra de conformidad a las exigencias de tratamiento de los contenidos de aprendizaje implicados.

Cabe señalar que de optarse por la resolución grupal de estas evaluaciones, se establece en ellas la responsabilidad individual de cada integrante del grupo. Por tanto el carácter de esta modalidad de evaluación será “grupal con responsabilidad individual”, lo que implica la examinación en las dos dimensiones.

La implementación de estas instancias de evaluación se desarrollará acorde al calendario académico y a los requisitos reglamentarios de la facultad de ingeniería de la Universidad Nacional de Lomas de Zamora.

A los efectos de la ponderación, se utilizará la calificación numérica comprendida en una escala de UNO (1) a DIEZ (10), estableciéndose la aprobación con un mínimo de CUATRO (4) puntos.

#### Graduación.

Se alcanzará el derecho a las titulaciones cuando el cursante:

Apruebe la totalidad de las asignaturas que conforman los ciclos lectivos,