



TECNICATURA en DESARROLLO DE SOFTWARE

Título al que se accede: Técnico Desarrollador de Software

Nivel de la carrera: Tecnicatura

La Tecnicatura Desarrollador de Software está diseñada para responder a las necesidades de actualización y apropiación de nuevos modelos, procesos, estándares y lenguajes, relacionados con el desarrollo de Software, utilizando los enfoques avanzados de calidad, metodologías y técnicas.

Se nos plantea la necesidad de incorporar en la carrera materias con intensa práctica y específicamente conocimientos básicos de la disciplina que nos permitan desarrollar capacidades para la resolución de problemas.

El egresado en la carrera de la presente Tecnicatura, es un programador que se dedica a una o más facetas del proceso de desarrollo de software. Puede contribuir a la visión general del proyecto en su nivel de aplicación no solo en el plano de sus componentes o tareas de programación individuales. Su función consiste en trasladar las especificaciones del analista en código ejecutable por la computadora.

El cambio de denominación está orientado a la formación de técnicos profesionales especializados en el desarrollo de software y a la utilización de las metodologías necesarias para ese fin. Se considera así las definiciones expresadas en el documento creado en el Instituto Nacional de Educación Técnica - INET y visto por la Cámara Empresaria de Software y Servicios Informáticos - CESSI

OBJETIVOS GENERALES DE LA CARRERA

Es objetivo de esta carrera la formación de técnicos especializados en Desarrollo de Software que puedan atender adecuadamente las demandas del mercado laboral. A través de este proceso la Universidad tiene como objetivo fortalecer el desarrollo de la industria del Software regional, provincial y nacional, de tal manera que alcance niveles de competitividad a escala internacional. Los egresados de este programa podrán afrontar el reto de desarrollar, validar, interpretar la especificidad de diseño y documentar los productos realizados participando activamente.



OBJETIVOS ESPECÍFICOS

- Propiciar un espacio para la actualización y la reflexión acerca del desarrollo de Software tomando como referencia los métodos y tecnologías modernas de desarrollo y los principios y procesos definidos en los modelos internacionales de calidad de software.
- Impulsar por medio de proyectos la definición de estrategias y acciones concretas que mejoren la práctica diaria del desarrollo de software en las organizaciones y el mercado.
- Conocer, evaluar y aplicar los lenguajes, modelos y tecnologías modernas, que permitan el desarrollo de aplicaciones distribuidas, flexibles y reutilizables.
- Evaluar el proceso de desarrollo de software e implantar principios de mejora continua en la calidad de software, de acuerdo a los atributos de calidad establecidos en el producto software y alineado a los estándares y procedimientos definidos en el modelo de calidad que se propone.
- Adquirir principios, modelos y herramientas para la planeación de proyectos de desarrollo de software.
- Comprender el concepto de redes de computadoras y de seguridad informática, para poder desenvolverse correctamente en el proceso de desarrollo de software y tender a un software seguro.

PERFIL DEL EGRESADO

Se pretende que el alumno como profesional de esta carrera esté capacitado para enfrentar con mayor propiedad el reto de desarrollo y evaluación de acuerdo a las exigencias de las organizaciones, pudiéndose desempeñar para:

- Realizar programas o componentes de sistemas de computación utilizando técnicas avanzadas de desarrollo.
- Interpretar especificaciones de diseño.
- Documentar los productos realizados.
- Verificar y validar componentes de software.
- Crear, corregir o adaptar programas.

Además será un técnico que podrá cumplir con las actividades y criterios definidos en el marco de un equipo de trabajo organizado y en las buenas prácticas.

Requisitos para el ingreso a la carrera:

Poseer el título de nivel medio o polimodal y haber cumplido con los requisitos del curso de nivelación.

Organización del Plan de Estudio

La Tecnicatura Desarrollador de Software está estructurada en dieciocho (18) asignaturas y las mismas están distribuidas en seis áreas de integración de formación en ciencias básicas, específicas



Universidad
del Chubut

y la práctica profesional/trabajo integrador. La duración es de dos años y un cuatrimestre. Cada año con dos cuatrimestres.

N°	Asignatura	Año	Cuat.	Áreas	Carga Horaria Total	Carga Horaria Semanal	Para Cursar Regular	Para Cursar Aprobada	Para Rendir Aprobada
1	Introducción a la Programación	1	1	D. P.	128	8			
2	Laboratorio de Programación I	1	1	D.P.	128	8			
3	Matemática Aplicada I	1	1	C.B.	112	7			
4	Inglés I	1	1	I.	64	4			
5	Técnicas Avanzadas de Programación	1	2	D.P.	96	6	1 - 2		1-2
6	Estructura de Datos y Algoritmos	1	2	D.P.	96	6	1 - 2		1-2
7	Paradigmas de Programación	1	2	L.B.D	112	7	1-2		1-2
8	Matemática Aplicada II	1	2	C.B.	112	7	3		1-3
9	Inglés II	1	2	I.	64	4	4		2,4
10	Laboratorio Avanzado de Programación II	2	1	D.P.	112	7	5	1-2	5-7
11	Bases de Datos I	2	1	L.B.D	128	8	6	1-2	5-6-7
12	Programación Web I	2	1	D.P.	128	8	7	1- 2	6-7
13	Redes y Seguridad Informática	2	1	A.S.O	80	5	9	1- 2-8	5-6-7-9
14	Ingeniería de Software I	2	2	I.S.	112	7	10-11	7	10-11-12
15	Programación Web II	2	2	D.P.	128	8	12	5	12
16	Bases de Datos II	2	2	L.B.D	128	8	11		11-6
17	Gestión de proyectos de software	2	2	DP	96	6	13		11-12-13
18	Practica Profesional/Trabajo Integrador.	3	1		260	16	14-15-16-17	10-11-12-13	14-15-16-17
	TOTAL HORAS				2.084				

Áreas de Integración Curricular

La propuesta curricular está basada en seis áreas de integración :

- Ciencias básicas
- Arquitectura y Sistemas Operativos
- Lenguajes y Bases de Datos
- Diseño y Programación
- Ingeniería de Software
- Idiomas

Nota: para acceder a la Práctica Profesional/Trabajo Integrador el alumno deberá tener las asignaturas del 2º año cursadas y aprobadas las del 1º año completo y el 1º cuatrimestre del 2º año.

Diseño Curricular de acuerdo a los trayectos de formación

	Ciencias Básicas	Arquitectura y Sistemas Operativos	Lenguajes y Bases de Datos	Diseño y Programación	Ingeniería de Software	Formación General
PRIMER AÑO	Matemática Aplicada I Matemática Aplicada II		Paradigmas de Programación	*Introducción a la Programación *Laboratorio I *Técnicas Avanzadas de Programación *Estructura de Datos y Algoritmos		Inglés I Inglés II
SEGUNDO AÑO		Redes y seguridad Informática	Base de Datos I Base de Datos II	*Laboratorio Avanzado de Programación II *Programación Web I *Programación Web II	Ingeniería de Software	
TERCER AÑO	Práctica Profesional/ Trabajo Integrador					



Contenidos mínimos de las asignaturas

Introducción a la Programación

Modelización de problemas del mundo real. Búsqueda de Estrategias para resolución de Problemas.

Concepto de programación imperativa. Algoritmo, proceso, programa. Estructura de control: secuencia, selección, iteración. Variables simples y estructuradas.

Diagramas de flujos de Datos. Metodología. Dibujo de diagramas. Diagrama de contexto. El diccionario de datos.

Estrategias de resolución: top-down, bottom-up, divide-y-conquista. Algoritmos de búsqueda, inserción, eliminación, ordenamiento. Características de la programación estructurada: modularización, legibilidad, mantenibilidad, reusabilidad, eficiencia. Documentación.

Técnicas de testing y debugging de aplicaciones.

Laboratorio de Programación I

Fundamentos del Lenguaje C. Estructura de un programa C. El entorno de desarrollo.

Declaraciones de constantes y variables. Tipos de datos simples y sus operadores.

Tipos de datos estructurados: arreglos, punteros y registros. Operaciones.

Entrada y salida de datos por consola.

Sentencias de control y decisión. Declaración definición e invocación de funciones, pasaje de argumentos por valor y por referencia.

Variables globales y locales. Ventajas y desventajas. Hábitat, ámbito y tiempo de vida de una variable.

Matemática aplicada I

Lógica Simbólica. Operaciones Lógicas. Tablas de Verdad.

Teoría de Conjuntos. Operaciones entre Conjuntos.

Relaciones Binarias y Funciones. Graficación e Interpretación de Gráficas.

Conceptos Indispensables de Precálculo. Manejo de Software Específico de Asistencia Matemática (Maple). Proyecto de Implementación.

Ingles I

Gramática, sintaxis, vocabulario, pronunciación y conversación.

Palabras estructurales: artículos, preposiciones, pronombres, conectores.

El verbo be.

Información personal. El presente simple. Palabras interrogativas.



Palabras conceptuales: sustantivos, adjetivos, verbos. Tiempos continuos y perfectos. Tiempos condicionales.

Descripción de lugares y personas.

Verbos no conjugados. Voz pasiva. Tiempos futuros.

Be going to y presente continuo usados haciendo referencia al futuro.

Técnicas avanzadas de programación

Recursión directa e indirecta, ventajas y desventajas.

Estructuras de datos lineales: Métodos de ordenación y búsqueda

Estructura de datos no lineales: árboles. Árboles binarios. Árboles binarios ordenados. Algoritmos de búsqueda, inserción, eliminación y ordenamiento. Noción de eficiencia.

Estructuras de Datos y Algoritmos

Tipo Abstracto de Datos. Estructuras de datos lineales: pilas, listas, colas, dicolas: propiedades, implementación mediante arreglos y listas enlazadas. Operaciones.

TAD en disco: Archivos secuenciales y aleatorios. Algoritmos de ordenación de archivos: MergeSort. Algoritmos.

Técnicas de diseño de algoritmos. Algoritmos voraces, divide y vencerás, algoritmos de Greedy.

Paradigmas de programación

Paradigma funcional, lógico, imperativo características fundamentales.

Paradigma orientado a objetos: Generalidades. Concepto de Objeto. Clases, definición, Miembros datos y funciones. Clases derivadas, conceptos de Herencia simple y múltiple.

El Polimorfismo, sobrecarga de operadores binarios. Patrones. Implementación de los conceptos en Lenguaje C++.

Matemática Aplicada II

Conceptos Indispensables de Cálculo I: Límites Laterales. Límite Funcional. Límites Simples. Continuidad de Funciones. Derivada. Interpretación Geométrica. Derivadas Elementales. Puntos Críticos. Integral. Integrales Inmediatas. Aplicaciones. Manejo de Software Específico de Asistencia Matemática (Maple). Proyecto de Implementación.

Ingles II

Voz pasiva. Interpretación y traducción de la terminación ing, ed, to infinitivo. Construcciones impersonales. Prefijos y sufijos. Estructura de una compañía de software. Tiempos condicionales. Factual conditionals. Predictive conditionals. Hypothetical conditionals. Counterfactual



conditionals. Descripción de lugares y personas. Tiempos futuros. Clausulas relativas. Who, which, that, where, when, why. Usos de diccionarios en la web y bancos terminológicos.

Laboratorio Avanzado de programación II

Utilización y manejo de software de desarrollo Visual. Lenguaje C#
Tipos de Datos, Operadores, Funciones y Procedimientos, Tipos de Bucles, Comandos de Entrada y Salida, Solución y Proyectos, Espacio de Nombre. Clases. Objetos, Herramientas de trabajo. Propiedades de los Componentes de: Controles comunes, Contenedores, Menues, Datos, Impresiones, Diálogos, Eventos.
Conectividad con Base de Datos.

Base de Datos I

Conceptos básicos. Componentes y funciones del DBMS.
Modelo de Entidad / Relación. Conceptos básicos: entidades, conjunto de entidades, relaciones, conjunto de relaciones, atributos. Cardinalidad. Claves. Roles. Diagramas de E/R. Traducción del modelo de entidades/relaciones a tablas.
Modelo relacional Conceptos básicos: relación, atributos, dominios. Clave: superclave, clave candidata, clave primaria.
Diseño de Bases de Datos Relacionales: Descomposición de relaciones. Propiedades deseables de la descomposición: pérdida de información, pérdida de dependencia funcional. Formas Normales. Proceso de descomposición sin pérdida de información.
Bases de Datos Transaccionales / SQL. Lenguajes de Consulta: creación de la base de datos. Acceso y modificación de datos La instrucción SELECT, UPDATE, INSERT, DELETE. Tipos de datos, operadores: aritméticos de bit, lógicos, unarios, Consultas. Cláusulas where, having, from.

Programación Web I

Historia de las técnicas y herramientas de Programación Web. Conceptos de programación Internet/Intranet. Conceptos de Sistemas Cliente-Servidor. Concepto de páginas estáticas y páginas dinámicas. Programación Web vs. Tradicional.
Protocolo de transferencia de HiperTexto (HTTP). Desarrollo de Páginas Estáticas. Lenguajes de Hipertextos HTML, DHTML, XML. Sintaxis de HTML.
Concepto de Forms. Referencia a Forms en Javascript. Eventos y Handlers. Métodos de Form. Objetos Form. Utilización de propiedades y métodos de los objetos.
Desarrollo de Páginas Dinámicas. Lenguajes de programación JavaScript.
Conceptos de programación Dinámica. Conceptos de desarrollo con ASP.NET y PHP. Conceptos de Web Services.



Redes y seguridad informática

Redes de área local (LAN). Redes de área metropolitana (MAN). Redes de área mundial (WAN). Internet. Intranet. Tipos de conectividad. Dispositivos de redes. Protocolos. Capas ISO/OSI. Protocolo TCP/IP. Puertos. Ejemplos de protocolos Telnet, SSH, FTP, SFTP, HTTP, SMTP, SNTP. Direcciones IP. Concepto de direcciones IPv6. Arquitectura cliente/servidor. Servidores. Introducción a la seguridad informática. Disponibilidad, confidencialidad, integridad. Amenazas y vulnerabilidades. Diferentes tipos de ataques. Concepto de software seguro. Tendencias técnicas que afectan la seguridad del software. Bugs. Autenticación y autorización. Criptografía. Técnicas clásicas de cifrado. Algoritmos y funciones más usados. Firma digital.

Ingeniería de software

Conceptos del ciclo de desarrollo de software. Modelo code and Fix. Modelo en cascada. Concepto de Prototipos. Modelo en espiral. Proceso Unificado. Introducción a UML. Diagramas. Modelos. Modelos estáticos y dinámicos. Diagramas de Clases. Introducción al Proceso Unificado. Dirigido por Casos de Usos. Centrado en la Arquitectura. Iterativo e Incremental. Modelo de Casos de Usos. Captura de requisitos. Modelo del Dominio. Diagrama de Interacción. UML Avanzado. Introducción a patrones de diseño. Conceptos. Descripción. Selección de un patrón de Diseño. Utilización. Problema. Solución. Consecuencia.

Programación Web II

Introducción a XML, Documentos XML, Formato de datos y Validación. Conceptos de parsing de XML. Desarrollo de páginas con PHP. Diferencias entre ASP.NET y PHP. Sintaxis PHP, Variables, Operadores, Estructuras de control, Arrays, Manejo de sesiones en PHP, Cookies, Utilizando includes en PHP, Funciones PHP de acceso a bases de datos, Diferencias entre la programación orientada a objetos y la programación procedural. Concepto de objeto como nueva herramienta para el desarrollo de aplicaciones con PHP 5. Archivos de configuración. Seguridad de aplicaciones PHP. Prácticas de programación con PHP. Entornos de desarrollo PHP (Symfony, Zend).

Bases de datos II

Utilización y manejo de un motor de Base de Datos: Administración de Usuarios, Inicios de Sesión, Funciones, Privilegios.



Administración de Bases: Copia de Seguridad, Plan de Mantenimiento. Diagramas, Tablas, Vistas, Funciones definidas por Usuarios, Procedimientos Almacenados, Bases del Sistema, Conectividad: ADO y ODBC.

Exportación e Importación de datos.

Diferencias entre los modelos convencionales de bases de datos y el modelo Orientado a Objetos. Modelización de objetos vs. Modelización de entidades. Semántica del esquema orientado a objetos. Bases de datos orientadas a objetos como una parte integral de la tecnología de objetos. Característica de los lenguajes de consultas orientados a objetos; un lenguaje tipo OSQL; procesamiento de consultas.

Gestión de Proyectos de Software y calidad

Gestión de proyectos. Estimaciones. Técnicas de descomposición. Tamaño del software. Objetivos de la planificación. Ámbito del software.

Métricas del proceso. Métricas orientadas al tamaño, a la función. Métricas para la calidad. Integración.

Garantía de calidad del software (SQA). Actividades de SQA. Revisiones de software. Fiabilidad del Software. Estándar de calidad.

Gestión de las configuraciones (GCS). Control de versiones. Control de cambios. Herramientas CASE para GCS.

Ingeniería del Software del comercio electrónico. Introducción. Sistemas distribuidos. Software intermedio. Arquitecturas estratificadas. Protocolos IP, POP3, ICMP. Comercio electrónico. Diseño de sistemas distribuidos.

Ingeniería de Software asistida por computadora. Que significa CASE. Taxonomía de herramientas CASE. Entornos CASE integrados. Arquitectura de integración. Repositorio CASE.

PRACTICA PROFESIONAL/TRABAJO INTEGRADOR (1 cuatrimestre):

Desarrollo de un proyecto o práctica laboral rentada coordinada y supervisada desde la ULP, que implique para el alumno la aplicación concreta de los conocimientos adquiridos integrando temas de Lenguajes, Ingeniería de Software y Base de Datos.

Análisis de Congruencia

Objetivo de la carrera	Alcances del Título	Perfil del Egresado	Asignaturas
Formar Profesionales Técnicos especializados en Desarrollo de Software, capaces de responder a las necesidades de actualización y apropiación de	Los ámbitos típicos para el desenvolvimiento de los profesionales en el desarrollo de software, en empresas privadas o administración pública en donde impera la necesidad del software como servicio, o	Un profesional de esta área estará capacitado para enfrentar con mayor propiedad el reto de desarrollo del software, de acuerdo a las exigencias de las organizaciones,	Las materias de la carrera están organizadas en seis trayectorias: Ciencias básicas, arquitectura y sistemas operativos, Diseño y programación, Lenguajes y Bases de



Universidad
del Chubut

nuevos modelos, procesos, estándares, lenguajes, métodos y tecnologías relacionados con el desarrollo de software y atender adecuadamente las demandas del mercado laboral.	aquellas que operen directamente en la producción del mismo.	pudiéndose desempeñar idóneamente.	Datos, Ingeniería del Software e Idiomas.
---	--	------------------------------------	---

Diseño Curricular de acuerdo a los trayectos de formación

	Ciencias Básicas		Arquitectura y Sistemas Operativos	Lenguajes y Bases de Datos	Diseño y Programación	Ingeniería de Software	Formación General
PRIMER AÑO	Matemática Aplicada I Matemática Aplicada II			Paradigmas de Programación	*Introducción a la Programación *Laboratorio I *Técnicas Avanzadas de Programación *Estructura de Datos y Algoritmos		Inglés I Inglés II
SEGUNDO AÑO			Redes y seguridad Informática	Base de Datos I Base de Datos II	*Laboratorio Avanzado de Programación II *Programación Web I *Programación Web II	Ingeniería de Software	



Universidad
del Chubut

TERCE R AÑO		Práctica Profesion al/ Trabajo Integrado r						